

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

DIGITÁLNÍ HUDEBNÍ EFEKT ZALOŽENÝ NA
WAVELETOVÉ TRANSFORMACI JAKO PLUG-IN MODUL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

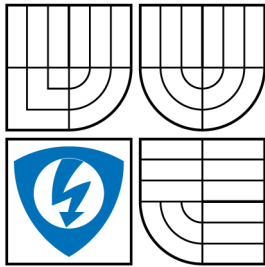
AUTOR PRÁCE
AUTHOR

MARTIN MIKL

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

DIGITÁLNÍ HUDEBNÍ EFEKT ZALOŽENÝ NA
WAVELETOVÉ TRANSFORMACI JAKO PLUG-IN MODUL
DIGITAL AUDIO EFFECT BASED ON WAVELET TRANSFORMATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN MIKL

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. PAVEL RAJMIC, PH.D.

BRNO 2010

ABSTRAKT

Bakalářská práce pojednává o problematice digitálního zpracování hudebního signálu. V úvodní části je rozebrána funkce a využití standartu VST. Stručně je popsána historie vzniku rozhraní VST, jeho uplatnění a použití v hostitelských editačních programech, zmíněny jsou i konkurenční technologie a licenční podmínky šíření této technologie. Dále je zde stručně popsán princip waveletové transformace, včetně několika základních matematických vztahů. Důraz je kladen především na konečnou diskrétní waveletovou transformaci a Mallatův pyramidový algoritmus.

S využitím znalostí o waveletové transformaci bylo navrženo blokové schéma digitálního hudebního efektu, ze kterého bylo vycházeno při tvorbě digitálního hudebního efektu standartu VST plugin. V závěrečné části práce je popsáno řešení tohoto efektu, včetně rozboru jeho dílčích funkcí. Je zde popsáno předávání důležitých parametrů i ovládání efektu.

KLÍČOVÁ SLOVA

VST plugin; Konečná diskrétní waveletová transformace; Mallatův pyramidový algoritmus; Digitální hudební efekt; Změna frekvenční charakteristiky.

ABSTRACT

The Bachelor's thesis deals about problems of digital signal processing. The first part includes function and use of VST Plug-In standard. It describes the history of the VST Plug-In interface, its use in hosting programs, there are a few words about another similar technologies and license terms of use. It is also briefly described the principle of wavelet transform, including some basic mathematical equations. Emphasis is placed on the Discrete-time wavelet transform and Fast wavelet transform.

The knowledge of the wavelet transform led into concept of a block diagram with digital audio effect. From this concept has been made digital audio effect in VST Plug-In standart. The final part describes the solution of this effect, including analysis of its component functions. It describes the transmission of the important variables and its controls.

KEYWORDS

VST Plug-In; Discrete-time wavelet transform; Fast wavelet transform; Digital audio effect; Change in frequency response.

MIKL M. *Digitální hudební efekt založený na waveletové transformaci jako plug-in modul: Bakalářská práce.* Brno: FEKT VUT v Brně, Ústav Telekomunikací, 2010. Počet stran 31. Vedoucí práce byl Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „DIGITÁLNÍ HUDEBNÍ EFEKT ZALOŽENÝ NA WAVELETOVÉ TRANSFORMACI JAKO PLUG-IN MODUL“ 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 bakalářské 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)

OBSAH

Úvod	7
1 Technologie VST plugin	8
2 Waveletová transformace	10
2.1 Úvod	10
2.2 Vývoj a princip waveletové transformace	10
2.3 Integrovaná waveletová transformace	12
2.4 Mnohoměřítková analýza	13
2.5 Konečná diskretní waveletová transformace	15
2.6 Mallatův pyramidový algoritmus	16
2.7 Kmitočtová charakteristika waveletových filtrů a aliasing	18
3 Řešení plugin modulu	21
3.1 Úvod	21
3.2 Konvoluce signálů	21
3.3 Blokové schéma hudebního efektu	22
3.4 Šablona VST Plug-In Template	23
3.5 Grafické rozhraní VST pluginy	24
3.6 Kvadraturní zrcadlové filtry	25
3.7 Dopředná a zpětná waveletová transformace	27
3.8 Frekvenční úpravy detailních koeficientů	27
4 Závěr	29
Literatura	30
Seznam symbolů, veličin a zkratk	31

SEZNAM OBRÁZKŮ

1.1	Funkce VST	9
2.1	Haarův wavelet	11
2.2	Dilatace (změna měřítka). Původní signál má modrou barvu, červený signál je odvozen z prvního pomocí změny měřítka	13
2.3	Translace (posunutí v čase). Původní signál má modrou barvu, červený signál je posunutý oproti původnímu	14
2.4	První krok při waveletové transformaci pomocí Mallatova pyramidového algoritmu	16
2.5	Princip Mallatova pyramidového algoritmu	17
2.6	Jeden krok inverzní waveletové transformace (rekonstrukce)	17
2.7	Frekvenční charakteristiky ideálních filtrů g (modrá barva), h (červená barva)	18
2.8	Příklad frekvenčních charakteristik filtrů g (modrá barva), který se překrývá s filtrem h (červená barva)	19
2.9	Rozdělení frekvenčního pásma na oktávy	19
3.1	Ilustrace principu konvoluce	21
3.2	Blokové schéma hudebního efektu, využívajícího Mallatův algoritmus s hloubkou dekompozice 4	23
3.3	Náhled na grafické rozhraní VST puginy	25

ÚVOD

Digitalizace při zpracování hudebních signálů je dnes již značně rozšířená mezi profesionálními nahrávacími studii i mezi širokou veřejností, věnující se záznamu a úpravě zvukových signálů. Svou oblibu si získala především díky široké škále jednoduchých editačních možností, které nabízí.

Digitální záznam zvuku spočívá v přivedení analogového zvukového signálu na A/D převodník, kde je signál navzorkován, kvantován a převeden na binární kód. Takto upravený signál lze zpracovat na osobním počítači, nebo jakémkoliv jiném přístroji, který umožňuje práci s digitálním zvukovým záznamem. Standartem kvality pro hudební CD je vzorkovací frekvence 44,1 kHz a 16bitové kódování. Kvalita digitálního záznamu zvuku záleží především na velikosti kvantizačního zkreslení, způsobeného zaokrouhlovací chybou při kvantování navzorkovaného signálu, a na kvalitě vstupního signálu. Tímto se lze vyhnout přeslechům mezi stopami, šumění a zkreslení magnetofonu při analogovém zpracování hudebních signálů. Mezi mnohými umělci i zvukovými režiséry je však toto jemné zkreslení oblíbené a žádoucí. Proto některá prestižní nahrávací studia používají kombinaci analogové i digitální technologie.

Spolu s rozšířením digitálních technologií záznamu zvuku započal i vývoj editačního software. Postupně bylo naprogramováno nepřehledné množství zvukových efektů za pomoci matematického rozkladu zvukového signálu, bylo vyvinuto několik rozhraní na bázi zásuvných modulů (tzv. plugin), lišících se především v kompatibilitě s hostitelskými editačními programy od různých výrobců.

V této práci je rozebrána problematika okolo digitálního zpracování hudebního signálu pomocí waveletové transformace, prostřednictvím VST plugin modulu. První část práce je věnována rozboru technologie VST z hlediska jejího vývoje, popisu funkce, licenčních podmínek jejího šíření a uplatnění v praxi. V druhé části je uveden stručný úvod do problematiky waveletové transformace a základní matematické principy, vedoucí k popisu Mallatova pyramidového algoritmu a uvedení kvadraturních zrcadlových filtrů. Třetí část pojednává o postupu technického řešení, které vyplývá v návrh blokového schématu waveletové transformace a jeho implementaci v projektu VST plugin.

1 TECHNOLOGIE VST PLUGIN

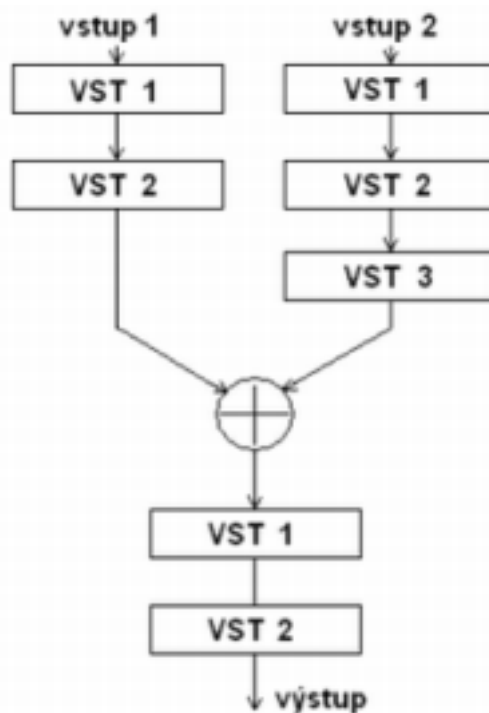
Virtual Studio Technology (VST) je rozhraní vyvinuté firmou Steinberg v roce 1996. Dnes se již hojně využívá při syntéze audio signálů (VSTi) a při úpravě audio signálů pomocí virtuálních hudebních efektů. Na trhu je již velké množství VST plugin, a to komerčních i freeware [4].

VST pluginy nabízejí velké množství funkcí a možností úpravy zvukového signálu. Některé jsou určeny pouze k analýze a sledování časových průběhů signálu, nebo jejich frekvenčních či směrových charakteristik. Jiné vstupní signál upraví podle volitelných nebo statických parametrů a pošlou na výstup upravený signál. Lze takto realizovat velké množství virtuálních hudebních efektů, jako chorus, reverb, delay, kompresor, ekvalizace frekvenční charakteristiky a spousta dalších.

VST instruments (VSTi) slouží ke generování zvukového signálu pomocí virtuálního syntezátoru a sampleru. Mnohé z těchto virtuálních instrumentů dosahují kvalit externích hardwarových samplerů, jsou však flexibilnější. Lze takto například získat kvalitní náhradu za nástroj, který chceme mít v nahrávce, ale je obtížné jej sehnat [5].

Pro práci s VST pluginy je nutné pracovat s programem, určeným pro úpravu hudebního signálu, který podporuje technologii VST. V uživatelském prostředí hostitelského programu lze libovolný plugin modul vložit do jakékoliv dílčí stopy, nebo jej lze vložit na konec řetězce zpracování hudebního signálu těsně před výstup programu, a tím ovlivnit vlastnosti výsledného zvuku jako celku (viz Obrázek 1.1).

Co se týče realizace digitálního zpracování hudebních efektů, tak konkurentem plugin technologií jsou samostatně fungující aplikace, které ke svému provozu nevyžadují hostitelský editační program. Jedná se o aplikace v režimu stand-alone. Tyto aplikace jsou plnohodnotnými spustitelnými programy, které vyžadují předchozí instalaci. Někteří výrobci produkují jak stand-alone, tak i plugin verze svých aplikací. Nejobvyklejší je to u virtuálních hudebních nástrojů, kde je tímto uživatelům umožněno používat stejný nástroj při hudební produkci na koncertech, i při nahrávání ve studiu a následném mixu hudebních nahrávek. Pro vyžití PC (nebo MAC) jako efektové jednotky je potřeba, aby měly stand-alone aplikace možnost měnit vstupy a výstupy programu ze zvukových software ovladačů na fyzické vstupy a výstupy zvukové karty. U plugin modulů se o komunikaci s vnějším prostředím stará hostitelská aplikace.



Obr. 1.1: Funkce VST

Jednotlivé VST pluginy lze sériově (i paralelně) skládat za sebe tak, že upravený signál z předešlé VST pluginy se dostane na vstup následující pluginy. Tyto pluginy mohou pracovat a přepočítávat signál v reálném čase. Je tedy potřeba dbát na pořadí těchto plugin. Je tedy například potřeba předem zvážit pořadí efektů pracujících s frekvenčními charakteristikami a rozhodnout se zda vstupní signál první ekvalizovat a poté zkreslit, či naopak.

Na trhu je nepřehledné množství hostitelských aplikací od různých výrobců. Přední světoví výrobci těchto aplikací často prosazují své vlastní plugin standarty, které jsou bohužel mnohdy vzájemně nekompatibilní. Krom VST plugin, které prosazují výrobky firmy Steinberg (Cubase, Nuendo, a další), jsou zde i jiná rozhraní pracující s digitální úpravou hudebního signálu. Jejich použití a oblíbenost se odvíjí především od oblíbenosti různých hostitelských aplikací. Mezi nejrozšířenější alternativy k VST pluginům patří především AU (rozhraní plugin efektů vytvořené firmou Apple), RTAS (vyvinuto firmou Digidesign, která je výrobcem oblíbeného editačního programu Pro Tools) a DX (postaveno na technologii Microsoft Direct X). [8]

Standart VST plugin umožňuje vývojářům, kteří získají licenci od firmy Steinberg, vytvářet své vlastní plugin moduly. V prostředí C++ lze vytvořit požadovaný efekt v podobě plugin modulu a exportovat jej jako knihovnu *.dll, kterou může vyvolat v hostitelké aplikaci. Je obvyklé shromažďovat vlastní VST pluginy ve společném adresáři. Umožní to rychlejší a přehlednější přístup k dostupným VST pluginům.

2 WAVELETOVÁ TRANSFORMACE

2.1 Úvod

V následujícím textu budou uvedeny základní poznatky a principy waveletové transformace. V druhé části problematiky waveletů se budu věnovat historii vzniku waveletové transformace a jejím principům. Dále se pokusím osvětlit základní poznatky ohledně rozkladu spojitého signálu do bazových funkcí pomocí integrální waveletové transformace a jeho obdobu pro navzorkovaný signál, tzv. diskretní waveletovou transformaci.

Ve čtvrté části nastíním princip mnohoměřítkové analýzy a v páté části se budu věnovat konečné diskretní waveletové transformaci, která je obdobou waveletové transformace pro signály s diskretním časem. V šesté části uvedu Mallatův pyramidový algoritmus, který vychází z konečné waveletové transformace, a v závěrečné části této problematiky osvětlím kmitočtové charakteristiky waveletových filtrů, jejich vlastnosti a vznik aliasingu. Pomocí Mallatova pyramidového algoritmu rozeberu blokové schéma hudebního efektu a stejný princip použiji v jádře programu, který bude přepočítávat hudební signál.

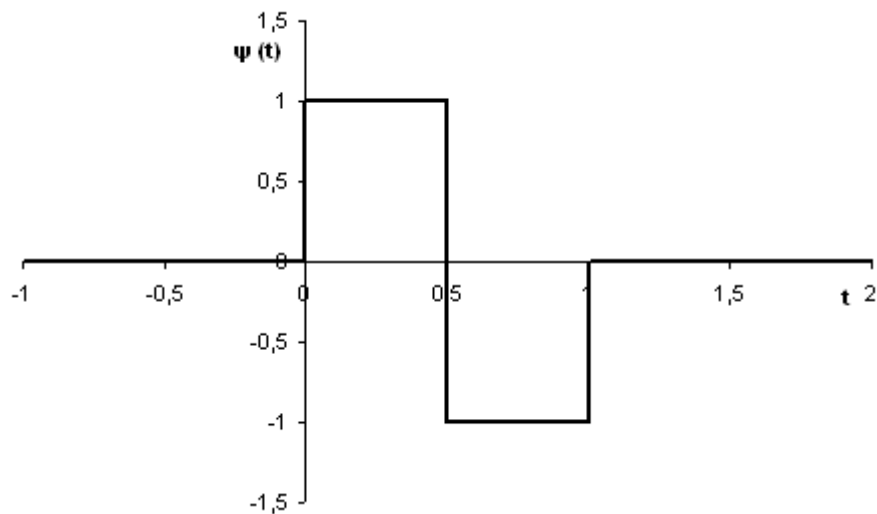
2.2 Vývoj a princip waveletové transformace

Waveletová (vlnková) transformace má své počátky na začátku 20. století, kdy se této problematice začal věnovat maďarský matematik Alfréd Haar. V roce 1909 pomocí posloupnosti

$$\psi(t) = \begin{cases} 0 & -\infty \leq t < 0 \\ 1 & 0 \leq t < 0,5 \\ -1 & 0,5 \leq t < 1 \\ 0 & 1 \leq t < \infty \end{cases} \quad (2.1)$$

demonstroval výpočet ortonormálního systému pomocí plošné integrace, v reálné ose. Tato posloupnost dostala název Haarův wavelet. Je to historicky první a zároveň nejjednodušší wavelet.

V roce 1975 nastínil fyzik George Zweigh principy integrální waveletové transformace, které v roce 1982 formulovali Jean Morlet a Alex Grossmann. Od roku 1989, kdy Stéphane Mallat vynalezl mnohoměřítkovou analýzu, se staly wavelety a waveletová transformace rychle se rozšiřujícím odvětvím. V dnešní době stále pokračuje vývoj dalších metod waveletové transformace a své využití nalézá v mnoha



Obr. 2.1: Haarův wavelet

odvětvích, zejména při zpracování a analýze signálů [6].

Wavelet (vlnka) je matematická funkce, která třídí informace do různých frekvenčních složek, a pak analyzuje tyto složky podle měřítka waveletové funkce. Obrovskou výhodou waveletové transformace oproti Fourierově transformaci je schopnost řešit matematické a fyzikální situace, kde je signál nespojitý, nebo má ve svém průběhu jiné ostré hrany.

Různé wavelety se používají ve specifických odvětvích jako je matematika, kvantová fyzika, elektrotechnika nebo seismologie. Zejména užitečné jsou při kompresi obrazu, v radiolokační technologii, meteorologii, zpracování zvukového signálu a při oddělování signálu od rušivých prvků.

Wavelet má tvar oscilace, která začíná a končí vždy v nule. Jsou tvořeny tak, aby byly jejich specifické parametry užitečné při zpracování daného signálu. Lze měnit jejich měřítko posunovat je v čase vůči analyzovanému signálu.

Slovo wavelet pochází z francouzštiny, poprvé ho se objevilo v pracích fyziků Jeana Morleta a Alexe Grossmanna. Použili výraz „ondelette” (vlnka), který byl později převzat do angličtiny záměnou „onde” za „wave.”

2.3 Integrální waveletová transformace

Všechny funkce, které lze analyzovat waveletovou transformací musí náležet prostoru $L^2(\mathbb{R})$ daném množinou všech komplexních funkcí definovaných na \mathbb{R} . Waveletem pak nazýváme funkce $\psi(\omega)$, které náleží tomuto prostoru, a splňují podmínku přípustnosti, která definuje inverzní integrální waveletovou transformaci. Podmínka přípustnosti je dána vztahem

$$\int_{-\infty}^{\infty} \frac{|\psi_f(\omega)|}{|\omega|} d\omega < \infty, \quad (2.2)$$

kde $\psi_f(\omega)$ představuje Fourierovu transformaci funkce $\psi(\omega)$ [1]. Pro wavelety ψ musí také platit podmínka nulové průměrné hodnoty

$$\int_{-\infty}^{\infty} \psi(t) dt = 0. \quad (2.3)$$

Z předchozích vztahů vyplývá, že wavelet je funkce s osciloskopickým charakterem, kdy se kmity tlumí směrem k $\pm\infty$

$$\lim_{t \rightarrow \infty} \psi(t) = 0, \quad (2.4)$$

$$\lim_{t \rightarrow -\infty} \psi(t) = 0. \quad (2.5)$$

Integrální waveletová transformace (CWT) funkce f se spojitým časem je dána následujícím vztahem

$$CWT(a, b) = |a^{-0,5}| \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt, \quad (2.6)$$

kde a je reálné číslo větší a různé od nuly. Změnou parametru a měníme měřítko waveletové funkce ψ . Parametr b patří do množiny reálných čísel a jeho změnou docílíme posunutí waveletové funkce po časové ose. Koeficient $|a^{-0,5}|$ má za úkol zachování energie soustavy.

Existenci inverzní waveletové transformace lze dokázat také pro vybrané parametry $a = 2^j$, $b = 2^j k$, kterých je spočetně mnoho. Pomocí těchto parametrů můžeme definovat diskrétní waveletovou transformaci (DWT), která je navzorkovanou obdobou integrální waveletové transformace

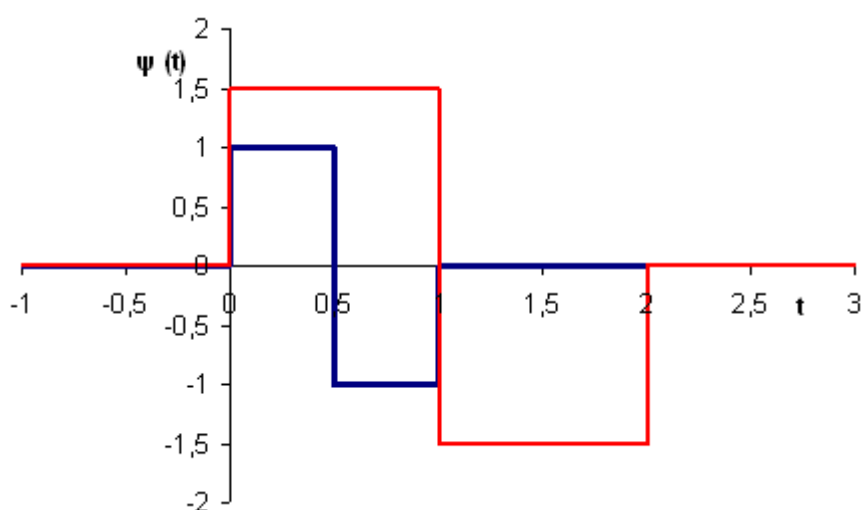
$$DWT(j, k) = \sqrt{|a^{-j}|} \int_{-\infty}^{\infty} f(t) \overline{\psi(2^{-j}t - k)} dt. \quad (2.7)$$

Parametry j, k patří do množiny celých čísel [1].

Některé wavelety ψ mají vlastnost, že množina $\{\psi_{j,k}\}$ je úplnou ortonormální

bázi $L^2(\mathbb{R})$, jedná se o takzvané mateřské wavelety. Obdobně jakou u Fourierovy transformace, kde lze signál rozložit na dílčí harmonické signály pomocí obměny parametrů frekvence a fázového posunu, lze také libovolný signál rozložit na waveletovou řadu pomocí změn parametrů j, k .

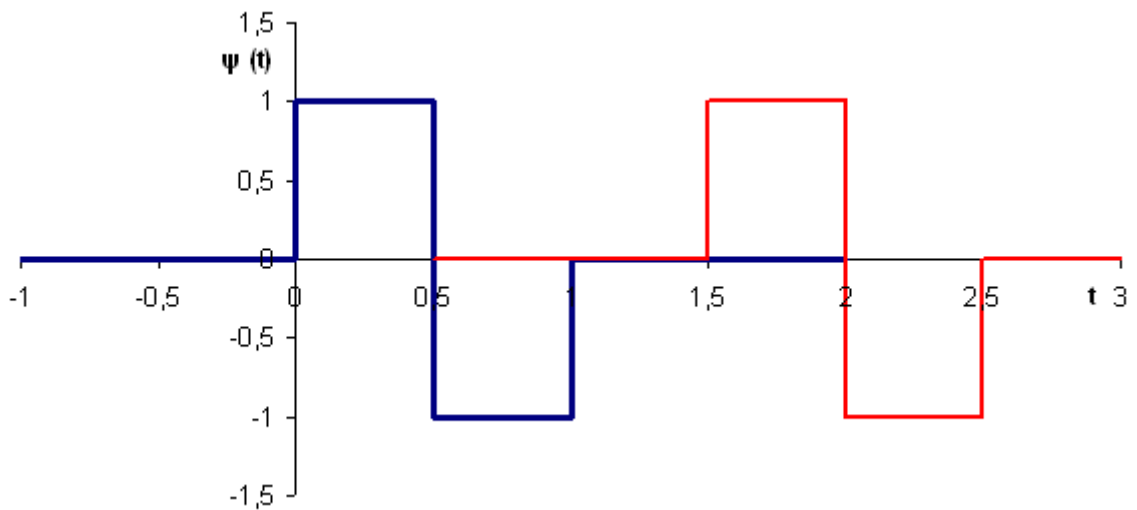
Zvláštní případy waveletů jsou takové, kdy má oscilační průběh pouze v ohraničeném intervalu a mimo něj je waveletová funkce nulová. Těmto waveletům se říká, že mají kompaktní nosič. Velkou výhodou těchto waveletů je jejich lokální působnost vzhledem k analyzované funkci. Pomocí změny měřítka (Obrázek 2.2) a posunutím (Obrázek 2.3) waveletu vzhledem k analyzované funkci se při analýze můžeme soustředit na libovolnou oblast této funkce.



Obr. 2.2: Dilatace (změna měřítka). Původní signál má modrou barvu, červený signál je odvozen z prvního pomocí změny měřítka

2.4 Mnohoměřítková analýza

Mnohoměřítkovou analýzu poprvé použili a zavedli matematici Stéphane Mallat a Yves Meyer. Její název je odvozen z angličtiny (multi-resolution analysis). Změnami koeficientů j, k lze analyzovat jakoukoliv část signálu, jak bylo uvedené výše. Pro větší koeficienty j , které zvětšují měřítko 2^j násobně, analyzujeme charakter průběhu signálu. S malými koeficienty j zase sledujeme podrobné chování signálu. Z tohoto poznatku mnohoměřítková analýza vychází.



Obr. 2.3: Translace (posunutí v čase). Původní signál má modrou barvu, červený signál je posunutý oproti původnímu

Prostor $L^2(R)$, ve kterém se wavelet nachází, lze rozložit na dílčí podprostory

$$L^2(R) = \dots \oplus W_2 \oplus W_1 \oplus W_0 \oplus W_{-1} \oplus \dots, \quad (2.8)$$

kde jednotlivá W značí prostory funkcí [1]. Pro vysvětlení principu mnohoměřítkové analýzy je nutné zavést pojem měřítková funkce. Je to funkce definována v prostoru $L^2(R)$, množinu podprostorů, které do ni náleží budeme značit $\{V_j\}$, kde j jsou celá čísla. Značíme ji ϕ . Měřítková funkce má na starosti změnu měřítka u waveletové funkce

$$V_j = V_{j+1} \oplus W_{j+1}. \quad (2.9)$$

Dosazením do předchozího vztahu lze upravit na

$$L^2(R) = V_j \oplus W_j \oplus W_1 \oplus W_{j-1} \oplus W_{j-2} \oplus \dots \quad (2.10)$$

Díky mnohoměřítkové analýze lze jakoukoliv funkci, která náleží prostoru $L^2(R)$, rozložit na součet dílčích vzájemně ortogonálních funkcí. Základem mnohaměřítkové analýzy jsou dva vztahy, a to měřítková rovnice

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi(2t - k) \quad (2.11)$$

a waveletová rovnice [1]

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} g_k \phi(2t - k). \quad (2.12)$$

Parametry h a g jsou předem dané posloupnosti souřadnic. Měřítková a waveletová funkce jsou tedy jakousi lineární kombinací měřítkové funkce $\phi(2t)$, která je o jednu polovinu zúžená oproti těmto funkcím. Vektory g, h jsou různé pro každý wavelet. Pokud je to možné, upřednostňujeme wavelety s kompaktním nosičem. Každý wavelet má už z definice minimálně jeden nulový bod. Počet těchto nulových bodů je jeden z charakteristických rysů každého waveletu. Pokud platí

$$\int_{-\infty}^{\infty} \psi(t)t^i dt = 0 \quad \text{pro } i = 0, 1, 2, \dots, u-1, \quad (2.13)$$

pak u je počet nulových bodů daného waveletu ψ .

Pokud bude řád analyzovaného polynomu $u-1$, pak bude platit, že všechny detailní waveletové koeficienty $c_{j,k}$ budou nulové pro $j \geq 0$

$$c_{j,k} = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt = 0. \quad (2.14)$$

Pokud by se ale v průběhu analyzovaného signálu objevila nespojitost, nebo jiná singularita, tak by měly waveletové koeficienty při vyšším stupni dekompozice výrazně nenulové hodnoty v tomto místě.

2.5 Konečná diskretní waveletová transformace

Ve většině oborů, které využívají waveletovou transformaci je zapotřebí analyzovat diskretní signály, které mají konečnou délku. Z integrální waveletové transformace lze odvodit její obdobu pro diskretní signály, která získala pojmenování konečná diskretní waveletová transformace. Tuto konečnou diskretní waveletovou transformaci (DTWT - discrete-time wavelet transform) můžeme vyjádřit čtvercovou ortogonální maticí W řádu $n \times n$, kterou podle speciálního algoritmu zaplní prvky z vektorů g, h .

Pokud provedeme transformaci vstupního vektoru $y = \{y_0, \dots, y_{n-1}\}$, kde n je délka vektoru, pak dostaneme výsledek v podobě vektoru $d = \{d_0, \dots, d_{n-1}\}$, který má stejný počet prvků jako vstupní vektor

$$d = Wy. \quad (2.15)$$

Díky linearitě waveletové transformace způsobené ortogonalitou matice W , lze vektor y zpětně vypočítat z vektoru d pomocí inverzní waveletové transformace

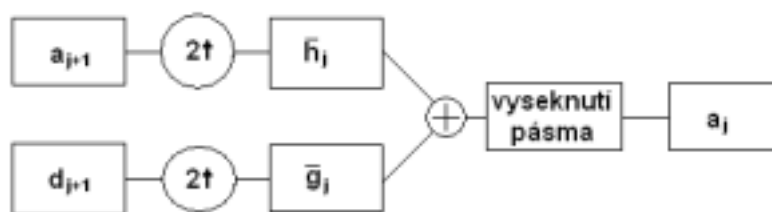
$$y = W^{-1}d = W^T d. \quad (2.16)$$

2.6 Mallatův pyramidový algoritmus

Mallatův pyramidový algoritmus byl pojmenován po svém autorovi, kterým byl francouzský matematik Stéphane Mallat, dále se pro tento algoritmus používá název rychlá waveletová transformace (FWT - fast wavelet transform). Mallat je mimo jiné i autorem mnohoměřítkové analýzy. Při tvorbě pyramidového algoritmu využil pevně daného řádu v ortogonální matici W a nahradil ji dvěma vektory g a h . Tyto vektory mají charakter filtrů horní propust - g a dolní propust - h .

Filtry g a h mají přímou souvislost s waveletovou a měřítkovou rovnicí, které byly zavedené v kapitole 2.4. Figurují ve vztazích, které řeší výpočet těchto dvou funkcí (rovnice 2.11 a 2.12). Jedná se o kvadraturní zrcadlové filtry.

Při analýze signálu y pomocí Mallatova pyramidového algoritmu podrobíme vstupní signál konvoluci s těmito filtry. Získané posloupnosti poté decimujeme (tj. vypustíme každý druhý vzorek).



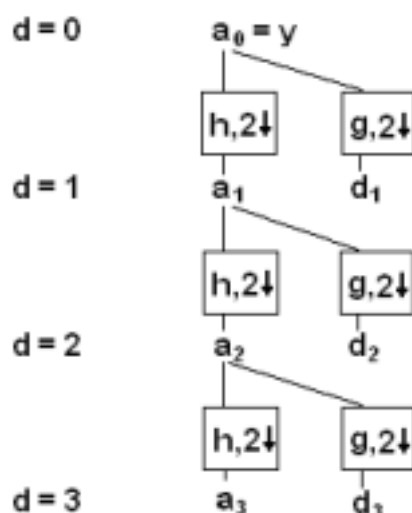
Obr. 2.4: První krok při waveletové transformaci pomocí Mallatova pyramidového algoritmu

Pokud je n délkou vstupního vektoru y a m je délka filtrů g a h , tak nově získané posloupnosti budou mít po decimaci délku

$$\frac{n + m + 1}{2}. \quad (2.17)$$

Koeficienty, které získáme filtrací vektorem g , nazýváme detailními waveletovými koeficienty a koeficienty, které získáme filtrací vektorem h , nazýváme aproximačními waveletovými koeficienty vstupního signálu.

V dalších krocích namísto vstupního signálu dále filtrujeme vektor aproximačních waveletových koeficientů, který jsme získali z předchozího kroku. Vektor detailních waveletových koeficientů uchováme a poté analyzujeme. Hloubka dekompozice je zvolený počet těchto kroků. Na zvolené hloubce dekompozice závisí kvalita a rychlost waveletové transformace. Volíme ji podle konkrétního účelu aplikace, ve které waveletovou transformaci využíváme.



Obr. 2.5: Princip Mallatova pyramidového algoritmu

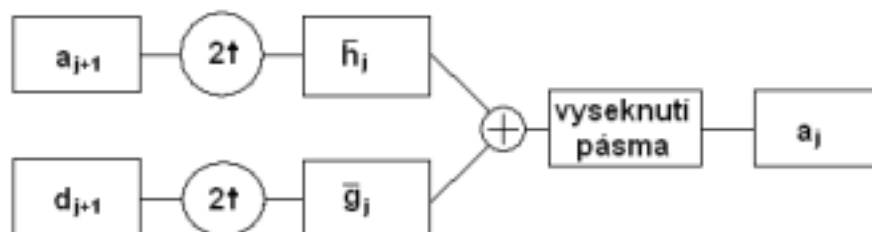
Maximální hloubka dekompozice závisí na délce vstupního signálu a značíme ji D_{MAX} . Vztah pro její výpočet je

$$D_{MAX} \leq \log_2 s, \quad (2.18)$$

kde s je délkou vstupního vektoru [1].

Při inverzní (zpětné) waveletové transformaci, která rekonstruuje analyzovaný signál je postup obdobný jako u dekompozice. Do aproximačních i detailních waveletových koeficientů vkládáme mezi každý vzorek nulu, čímž je nadvzorkujeme. Takto upravené posloupnosti poté filtrujeme inverzními filtry k filtrům g a h .

Z takto získané nové posloupnosti musíme vyloučit okrajové části a použít jen užitečnou střední část. Hloubka rekonstrukce je stejná jako hloubka dekompozice, provádíme tedy stejný počet kroků, jako v případě dekompozice signálu. Po provedení všech kroků získáme zrekonstruovaný vstupní signál.



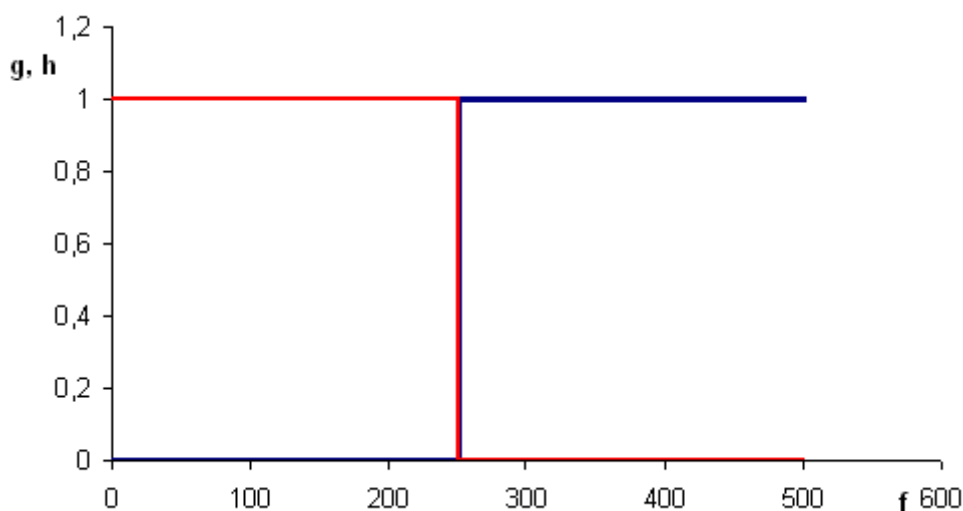
Obr. 2.6: Jeden krok inverzní waveletové transformace (rekonstrukce)

Není tedy potřeba při rekonstrukci signálu násobit vektory waveletových koeficientů inverzní maticí W^T . Tímto dosáhneme podstatného snížení výpočetní náročnosti celé waveletové transformace. Místo kvadratických výpočtů spojených s maticí W^T , použijeme pouze sadu lineárních operací při filtrování signálu. Použitím Mallatova pyramidového algoritmu také výrazně snížíme nároky na paměť.

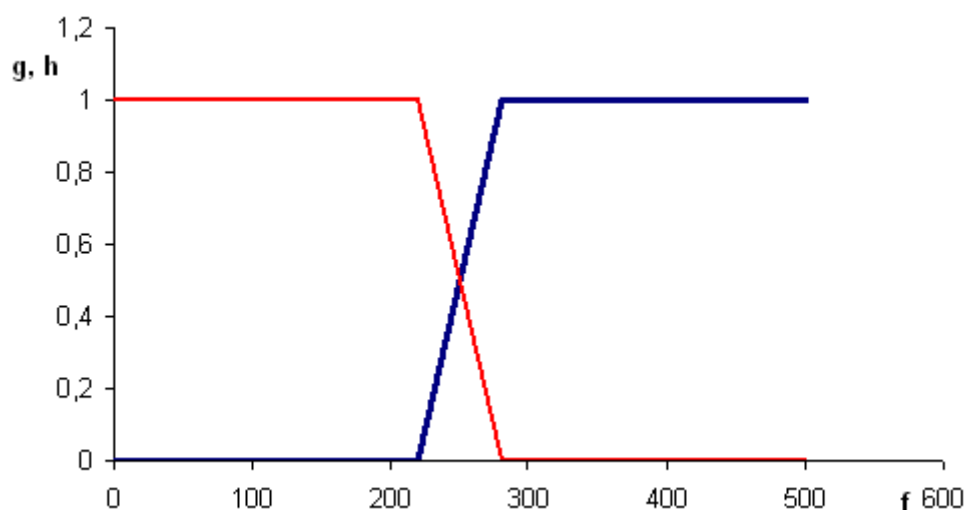
2.7 Kmitočtová charakteristika waveletových filtrů a aliasing

Filtry g a h vybíráme podle typu použitého waveletu. Rozdílné filtry použijeme v případě waveletu s kompaktním nosičem a waveletu bez ohraničeného nosiče. Pokud použijeme wavelet s neohraničeným nosičem, tak budou mít tyto filtry nekonečný počet hodnot, zatímco při použití waveletu s kompaktním nosičem bude počet prvků ve vektoru filtru konečný. Kmitočtová charakteristika těchto filtrů závisí na jejich koeficientech.

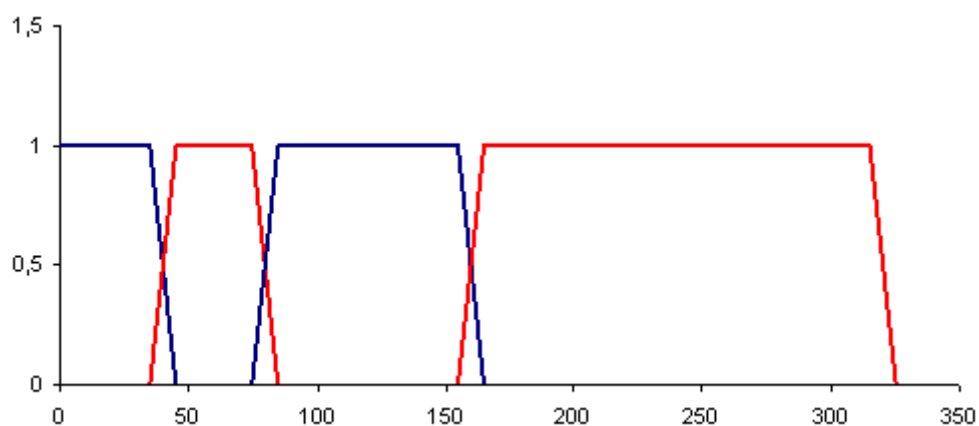
Filtry s vyšším řádem mají více koeficientů a jejich frekvenční charakteristika je mnohem strmější než u filtrů s nižším řádem. V ideálním případě má filtr frekvenční charakteristiku obdélníkového tvaru. V tomto případě by měl filtr okamžitý náběh a sestup. (viz Obrázek 2.7).



Obr. 2.7: Frekvenční charakteristiky ideálních filtrů g (modrá barva), h (červená barva)



Obr. 2.8: Příklad frekvenčních charakteristik filtrů g (modrá barva), který se překrývá s filtrem h (červená barva)



Obr. 2.9: Rozdělení frekvenčního pásma na oktávy

Bohužel reálné filtry mají své náběhy pozvolné, což se při překrývání filtrů projeví aliasingem. Čím strmější bude mít filtr své hrany, tím se aliasing projeví v menší míře (viz Obrázek 2.8). Kvadraturní zrcadlové filtry, které se během waveletové transformace používají jsou však speciálně navrženy tak, aby se vznik aliasingu neprojevil. Aliasing se sice při dekompozici objeví, ale při zpětné rekonstrukci naopak zmizí.

Při waveletové transformaci se kmitočtové pásmo díky použití filtrů typu horní a dolní propust během Mallatova pyramidového algoritmu rozpůlí na vyšší a nižší

kmitočty. Tímto způsobem se od zbytku kmitočtového pásma oddělí nejvyšší kmitočty. V následujících krocích se oddělují další vyšší kmitočty, a tímto způsobem se celé kmitočtové pásmo rozdělí na několik pásem, která svým charakterem a rozsahem přibližně odpovídají rozdělení akustického pásma na oktávy. Proto se těmto filtrům občas říká oktávové filtry (viz Obrázek 2.9) [6].

3 ŘEŠENÍ PLUGIN MODULU

3.1 Úvod

Tato kapitola je věnovaná technické realizaci bakalářské práce. Na počátku stručně osvětlím princip konvoluce signálů, kterého využiji při filtrování vstupního signálu. V další části se budu věnovat návrhu blokového schématu hudebního efektu, který bude využívat waveletovou transformaci a v závěrečné části bude následovat rozbor VST plugin modulu a popis jeho funkce.

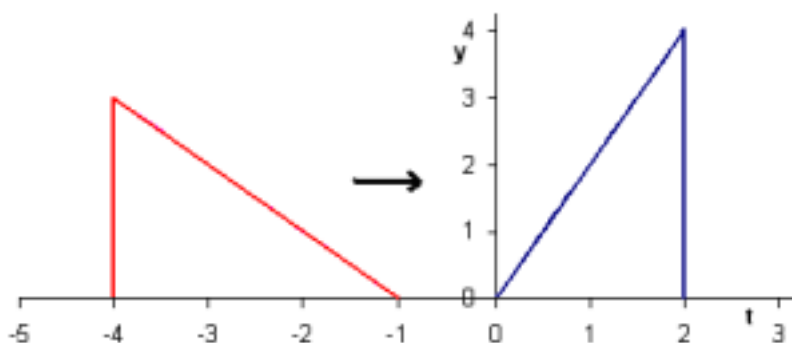
3.2 Konvoluce signálů

Konvoluce je matematická operace, která může zpracovávat dvě funkce a na jejím výstupu vznikne nová funkce. Jde v podstatě o překrytí funkce f s funkcí g , která je zrcadlově otočená podle osy y . Operátorem je hvězdička $*$. Spojitou konvoluci definujeme jako integrál součinů dvou funkcí $f(t)$ a $g(t)$, z nichž jedna je obrácená a posunutá po časové ose [3]

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(\tau - t)dt. \quad (3.1)$$

Diskrétní konvoluce je definována obdobně jako spojitá konvoluce, pro konečný počet hodnot

$$f[n] * g[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m]. \quad (3.2)$$



Obr. 3.1: Ilustrace principu konvoluce

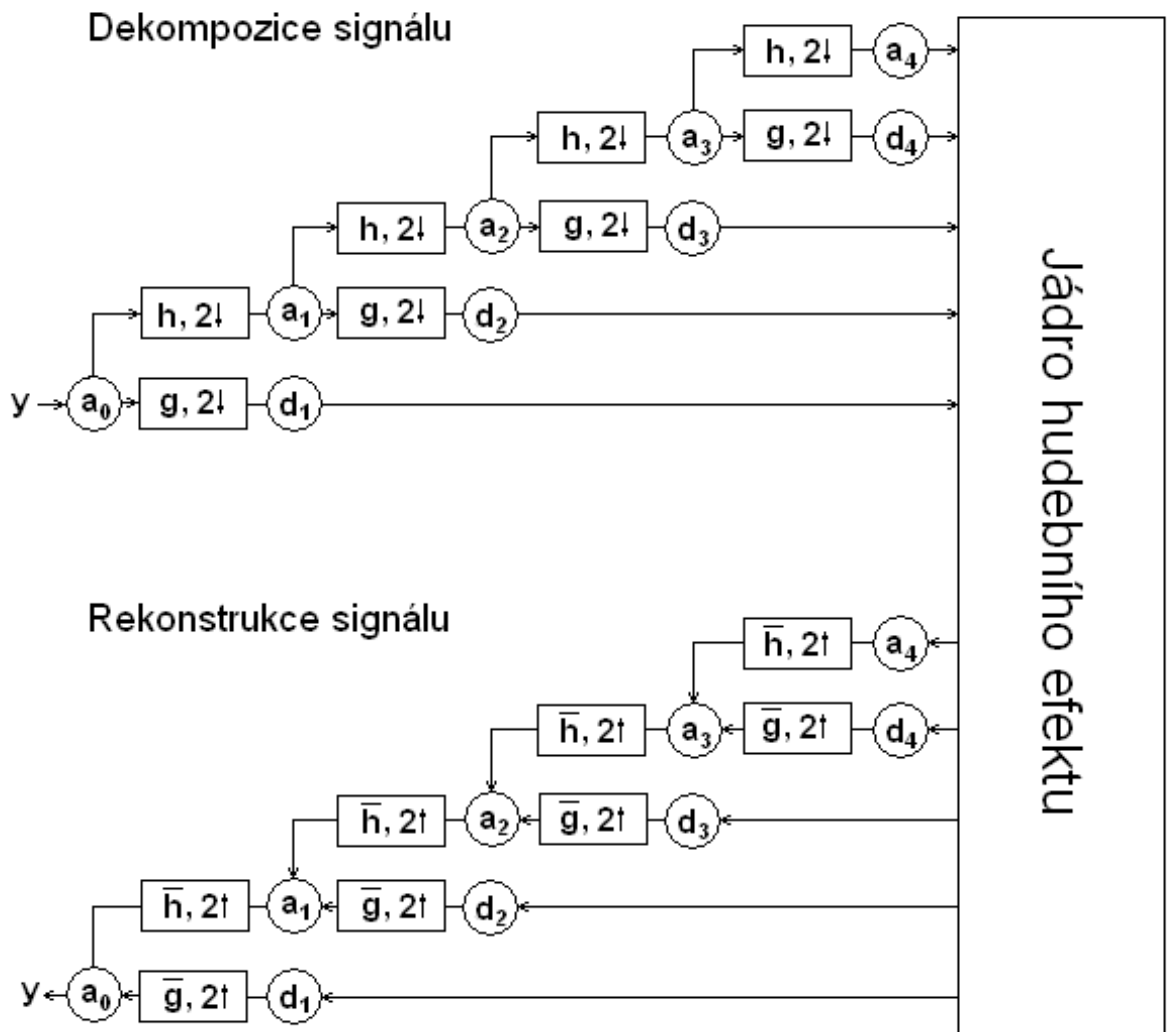
3.3 Blokové schéma hudebního efektu

Při návrhu blokového schématu jsem vyšel z Mallatova pyramidového algoritmu. Vstupní signál y je vektor navzorkovaného hudebního signálu a odpovídá výchozímu vektoru aproximačních waveletových koeficientů a_0 . Tento vektor podrobíme konvoluci s filtrem g , z čehož nám po decimaci vyjde vektor detailních waveletových koeficientů d_1 , který si uchováme. Vektor a_0 zároveň necháme projít konvolucí s filtrem h , čímž získáme další vektor aproximačních waveletových koeficientů a_1 , který je o řád výš. V dalších krocích opakujeme postup filtrování nově získaných aproximačních waveletových koeficientů, dokud nedosáhneme požadovaného stupně dekompozice. V blokovém schématu (obrázek 3.2) jsou graficky znázorněny 4 stupně dekompozice.

Detailní waveletové koeficienty získané dekompozicí signálu se dostanou na vstup jádra hudebního efektu, kde budou podle určitých algoritmů, specifických dle typu hudebního efektu, upraveny a převedeny na výstup. Poté čeká upravené detailní waveletové koeficienty proces rekonstrukce.

Vektor detailních waveletových koeficientů nejvyššího řádu (v našem případě 4. řád) se nadvzorkuje pomocí přidání jedné nuly mezi každý vzorek. Takto upraven se prožene filtrem inverzním k filtru g , poté se spojí s vektorem aproximačních waveletových koeficientů stejného řádu, který prošel obdobným procesem jako vektor detailních waveletových koeficientů, pouze s tím rozdílem, že byl filtrován inverzním filtrem k filtru h . Poté se z takto nově vzniklé posloupnosti vysekne užitečná střední část a získáme tak aproximační waveletové koeficienty nižšího řádu. Poté opakujeme tento proces obdobným způsobem, dokud nedosáhneme koeficientu a_0 . Dojde tímto k ukončení rekonstrukce a tato posloupnost je převedena na výstup programu.

Pokud by v jádře hudebního efektu nedošlo k žádným změnám v hodnotách detailních waveletových koeficientů (pokud tedy bude vstup do jádra bez jakéhokoliv zásahu převeden rovnou na výstup), pak by se měl v ideálním případě vstupní signál y rovnat rekonstruovanému signálu. Teoreticky záleží především na hloubce dekompozice.



Obr. 3.2: Blokové schéma hudebního efektu, využívajícího Mallatův algoritmus s hloubkou dekompozice 4

3.4 Šablona VST Plug-In Template

Tvůrce standartu VST plugin, firma Steinberg, umožňuje třetí straně se podílet na dalším vývoji této technologie. Pro udělení licence je nutné se zaregistrovat na webových stránkách této společnosti a souhlasit s licenčními podmínkami. Společnost Steinberg má tři typy licencí, a to freeware plugin, shareware plugin a pluginy pro komerční využití.

Po udělení licence je vývojářům umožněno stažení balíčku VST SDK, poslední nabízenou verzí ke dni 30. 5. 2010 je VST SDK 3. Tento balíček je souborem několika základních tříd a hlavičkových souborů, které jsou vodítkem pro další vývoj VST pluginy. Dále balíček obsahuje podmínky licenčních ujednání v textovém souboru, několik grafických variant loga VST, které je dle licence nutné na VST plugin umístit pro její volné šíření, a dále pomocný program pro testování VST plugin.

Soubory zdrojového kódu balíčku VST SDK však řeší jen základní principy komunikace s prostředím hostitelských aplikací a vše ostatní je ponecháno na konkrétních vývojářích. Při tvorbě VST pluginy jsem měl možnost navázat na šablonu VST Plug-In Template, která byla vyvinuta z balíčku VST SDK verze 2.4. Autorem této šablony je pan Ing. Jiří Schimmel, Ph.D.

Šablona VST Plug-In Template 1.2, ze které jsem ve své práci vycházel je projektem Microsoft Visual Studio C++, který kromě svých souborů obsahuje také přehledný popis souborů určených k editaci, užitečných funkcí a návod pro ladění a kompilaci konečné VST pluginy. V následujícím textu se soustředím na programovou nástavbu, která vychází z této šablony.

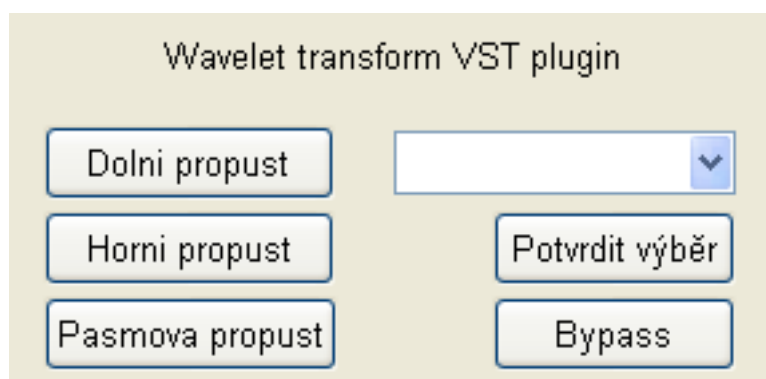
3.5 Grafické rozhraní VST pluginy

K jednomu z dílčích úkolů při realizaci projektu byla nutnost vytvořit nové ovládání VST pluginy. Považuji za důležitou součást projektu možnost měnit uživatelem některé klíčové parametry použité během waveletové transformace i následné operace s frekvenčním spektrem. V původním projektu se nacházely dva ovládací prvky, posuvné lišty „Gain” a „Bypass”, od kterých jsem upustil, protože jsem pro ně nenašel využití v mé práci.

Prvotním cílem mé práce bylo vytvořit jádro programu provádějící dopřednou a zpětnou waveletovou transformaci. Pro demonstraci funkce waveletové transformace je umožněno uživatelům zadat do tohoto procesu více různých kvadratických zrcadlových filtrů a porovnat tak rozdíl mezi rychlejšími fitry s menším množstvím hodnot a fitry s větším množstvím hodnot, které mají ostřejší vzájemné přechody. Jako zástupce kvadratických zrcadlových filtrů jsem vybral 3 filtry typu Daubechies, a to Daubechies 4, Daubechies 8 a Daubechies 16. Tyto filtry lze jednotlivě zadat ve výběrovém prvku ComboBox a potvrdit tlačítkem.

Další částí zadání je provedení úprav frekvenčního spektra vstupního signálu. Zde jsem uživatelům umožnil pouze výběr typu efektu. Jedná se o dolní, horní a pásmovou propust. Každý z těchto efektů je přiřazen jednomu konkrétnímu tlačítku, stisknutím tlačítka „Bypass” je zrušen výběr těchto efektů a signál je přepočítáván

pouze pomocí waveletové transformace.



Obr. 3.3: Náhled na grafické rozhraní VST puginy

Deklarace a přiřazení funkce jednotlivým ovládacím prvkům se děje ve třídě „Tform.“ Hodnoty, které vzejdou z této třídy jsou dále distribuovány v hlavní části programu.

3.6 Kvadrurní zrcadlové filtry

Jako kvadrurní zrcadlové filtry jsem zvolil široce využívané a oblíbené filtry typu Daubechies. Ty se vyznačují tím, že je předem známý počet nulových momentů. Filtr Daubechies 4 má čtyři koeficienty:

$$\begin{aligned} &0,4829629, \\ &0,8365163, \\ &0,2241439, \\ &-0,1294095. \end{aligned} \tag{3.3}$$

Tento filtr se vyznačuje nízkou strmostí u horní i dolní propusti a zratelnými zákmity, což způsobuje překrývání kmitočtových pásem, reprezentovaných detailními waveletovými koeficienty na výstupu waveletové transformace. Následné úpravy frekvenčního spektra u určitého pásma mají tedy vliv i na okolní pásma. Díky malému množství koeficientů filtr ale vyniká nízkou výpočetní náročností.

Filtr Daubechies 8 již má mnohem ostřejší přechod na hranici propustí a méně znatelné zákmity než u předchozího případu. Jeho koeficienty jsou:

$$\begin{aligned}
 &0,2303778, \\
 &0,7148466, \\
 &0,6308808, \\
 &-0,0279838, \\
 &-0,1870348, \\
 &0,0308414, \\
 &0,0328830, \\
 &-0,0105974.
 \end{aligned}
 \tag{3.4}$$

Posledním zvoleným filtrem je Daubechies 16. Díky svým šestnácti hodnotám má velice ostrý přechod na hranici propustí a prakticky žádné zákmity. Kmitočtová pásma se tedy překrývají minimálně, což je žádoucí jev. S vyšším počtem hodnot ale také roste výpočetní náročnost filtru, což by mohlo být v některých aplikacích citlivých na navýšení výpočetní náročnosti problémem. Koeficienty tohoto filtru jsou uvedeny zde:

$$\begin{aligned}
 &0,0544158, \\
 &0,3128716, \\
 &0,6756307, \\
 &0,5853547, \\
 &-0,0158291, \\
 &-0,2840155, \\
 &0,0004725, \\
 &0,1287474, \\
 &-0,0173693, \\
 &-0,0440883, \\
 &0,0139810, \\
 &0,0087461, \\
 &-0,0087035, \\
 &-0,0003917, \\
 &0,00067545, \\
 &-0,00011748.
 \end{aligned}
 \tag{3.5}$$

Všechny zde uvedené koeficienty jsou oproti hodnotám z některých jiných zdrojů již upraveny pro použití ve waveletové transformaci, vydělením hodnot konstantou $\sqrt{2}$. Dále byly pro větší přehlednost v textu částečně zaokrouhleny. Úplné koeficienty všech tří filtrů jsou uvedeny v hlavičkovém souboru „filtry.h.” [7]

3.7 Dopředná a zpětná waveletová transformace

Přepočítání hudebního signálu pomocí waveletové transformace je jádrem celé bakalářské práce. Samotná dopředná i zpětná waveletová transformace je součástí funkce „processReplacing,” která je deklarována ve třídě „CVSTTemplate.” Funkce „processReplacing” má tři parametry, pomocí kterých komunikuje s okolím, a to pole vstupních hodnot „inputs,” pole výstupních hodnot „outputs” a počet vzorků „SampleFrame.”

Před samotnou waveletovou transformací je potřeba zavést ukazatele na vstupní a výstupní pole pomocí funkcí „getInputBuffer” a „getOutputBuffer.” Pro práci s vektory je zde deklarováno 5 typů polí hodnot. Vstupní a výstupní pole hodnot „in” a „out,” pole reprezentující typ vybraného filtru „F” a pole aproximačních a detailních waveletových koeficientů „a_coefs” a „d_coefs,” které jsou klíčovými datovými poli, pro následné frekvenční úpravy vstupního signálu.

Po těchto úpravách lze již přistoupit k samotné waveletové transformaci prostřednictvím dvou konvolucí. První konvoluce probíhá mezi vstupními daty „in” a filtrem typu horní propust. Dalším vstupním parametrem je počet dekompozičních kroků, tedy počet provedených cyklů. Výstupem této konvoluce bude pole hodnot detailních koeficientů.

Druhá konvoluce je vyvolána bezprostředně po první, na jejím vstupu zůstává pole vstupních hodnot a stupeň dekompozice, který musí zůstat stejný jako u předchozí konvoluce. Změnou na vstupu funkce je ale jiný druh filtru, dolní propust. Výstupem této funkce je pole aproximačních koeficientů.

3.8 Frekvenční úpravy detailních koeficientů

Pro práci s frekvenčním spektrem dále využíváme pole detailních waveletových koeficientů, které získáme konvolucí vstupních hodnot s horní propustí. Tyto detailní koeficienty jsou rozříděny dle stupně dekompozice a příslušně opatřeny indexem. Koeficienty z prvního kroku dekompozice odpovídají nejvyšším frekvencím a s každým dalším krokem dekompozice tyto koeficienty reprezentují stále nižší část frekvenčního spektra.

Jako zástupce jednoduchých úprav frekvenčního spektra jsem zvolil dolní, horní a pásmovou propust. [2] Pro realizaci dolní propusti je zapotřebí vymazat ty koeficienty, které vznikly filtrováním v prvních krocích waveletové transformace a pak takto upravenou množinu detailních koeficientů podrobit zpětné waveletové transformaci. U realizace horní propusti je naopak zapotřebí vynulovat koeficienty vzniklé

v posledních krocích waveletové transformace. U pásmové propusti se jedná o kombinaci obou typů frekvenčních filtrů, je potřeba vynulovat počáteční koeficienty, i ty s nejvyšším indexem. Pro zpětnou waveletovou transformaci tak zbyde pouze pásmo někde ve středu kmitočtového spektra.

4 ZÁVĚR

Během tvorby bakalářské práce se mi podařily uplatnit a prohloubit nashromážděné informace ohledně waveletové transformace a její aplikace na analýzu signálu. Teoretickou část této práce jsem z velké části převzal a částečně rozšířil z mého semestrálního projektu, který je předchůdcem této práce.

Použití waveletové transformace ve VST pluginech umožnilo obrovský rozmach této technologie. Tento princip, nenáročný na výpočetní výkon, umožňuje přepočítávání zvukového signálu včetně změn jeho parametrů v reálném čase. Za několik let od uvedení této technologie na světlo světa vzniklo velké množství nejrůznějších plugin, které umí zpracovat téměř cokoliv od zvýraznění či potlačení vybraného frekvenčního pásma až po změnu výšky a délky tónu.

V hudebním průmyslu již před několika lety zavládl rozmach digitální nahrávací techniky, která se neustále zdokonaluje. Stále více nahrávacích studií upouští od tradičního analogového způsobu zvukového záznamu, díky výhodám, které digitální zpracování zvuku nabízí. S rozvojem výpočetní techniky se zpřístupnila možnost vytvořit si zvukovou nahrávku širokému spektru lidí. Díky přehledným editačním programům a jednoduchosti obsluhy VST plugin si může vytvořit nahrávku téměř kdokoliv, v kvalitě přiměřené jeho zkušenostem, hudebnímu citění a vlastnictvím dalšího nezbytného vybavení, jako jsou mikrofony, převodníky a podobně.

Jako hlavní produkt této bakalářské práce je VST plugin pojmenovaný „Wavelet transform,” zkomplikovaný do podoby *.dll knihovny. VST plugin byl projektován v prostředí MS Visual Studio 2008 C++ a výsledný produkt byl laděn v prostředí hostitelské aplikace Sony Vegas 6.0. Spolu s elektronickou verzí tohoto textu je tento projekt součástí přiloženého CD disku.

LITERATURA

- [1] RAJMÍČ, 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.
- [2] KÁŇA, L., SCHIMMEL, J.: *Studiová a hudební elektronika*. Elektronická skripta VUT v Brně, Brno, 2002.
- [3] WEISSTEIN, Eric W.: *Convolution* [online]. 1999, poslední revize 15. 1. 2010. Dostupné z: <<http://mathworld.wolfram.com/Convolution.html>>.
- [4] STEINBERG Media Technologies: *Company technologies* [online]. 2010, poslední revize 25. 1. 2010. Dostupné z: <http://www.steinberg.net/en/company/steinberg_technology.html>.
- [5] Wikipedia: *Virtual Studio Technology* [online]. 2010, poslední revize 25. 1. 2010. Dostupné z: <http://en.wikipedia.org/wiki/Virtual_Studio_Technology>.
- [6] Wikipedia: *Discrete Wavelet transform* [online]. 2009, poslední revize 15. 12. 2009. Dostupné z: <http://en.wikipedia.org/wiki/Discrete_wavelet_transform>.
- [7] WICKERHAUSER M. V.: *Adapted Wavelet Analysis from Theory to Software*, IEEE Press, 1994 A K Peters, Ltd
- [8] JIRSÁK, M.: *VST, AU, RTAS, DX, a jim podobní*. Muzikus. 2009, XIX, Prosinec 2009, s. 42-56. ISSN 1210-1443.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D Analogově digitální převodník

VST Virtual studio technology

VSTi Virtual studio technology instrument

PC osobní počítač

MAC Produkt firmy Apple

OS Operační systém

AU Audio units

DX Microsoft Direct X

RTAS Real-time audio suite

CWT Continuous wavelet transform

DWT Discrete wavelet transform

DTWT Discrete-time wavelet transform

FWT Fast wavelet transform