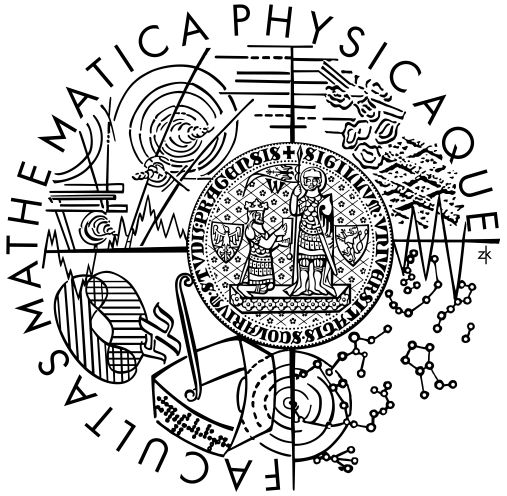
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE




Jan Krajíček

## Tvorba spektrogramů a jejich zpětná syntéza

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Bálek

Studijní program: Obecná informatika

2010

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne Jan Krajíček

# Obsah

Název práce: Tvorba spektrogramů a jejich zpětná syntéza
Autor: Jan Krajíček
Katedra: Katedra aplikované matematiky
Vedoucí bakalářské práce: Mgr. Martin Bálek
e-mail vedoucího: martin.balek@mff.cuni.cz

Abstrakt: V práci je popsána tvorba grafů čas-frekvence-intenzita (spektrogramů) ze zvukových nahrávek pomocí Fourierovy transformace. Dále jsou zhodnoceny teoretické možnosti a omezení zpětné syntézy zvuku ze spektrogramu a popsány dvě praktické metody syntézy, založené na rekonstrukci z čistých tónů a z náhodného šumu. Tvorba spektrogramů a obě popsané metody syntézy jsou implementovány v podobě programu s grafickým uživatelským rozhraním, které umožňuje pohodlné nastavení příslušných parametrů.

Klíčová slova: spektrogram, syntéza, Fourierova transformace, zvuk

Title: Spectrogram generation and their synthesis
Author: Jan Krajíček
Department: Department of Applied Mathematics
Supervisor: Mgr. Martin Bálek
Supervisor's e-mail address: martin.balek@mff.cuni.cz

Abstract: The thesis describes generation of time-frequency-intensity graphs (spectrograms) of audio recordings using the Fourier transform. The theoretical possibilities and limitations of spectrogram synthesis (audio reconstruction of a given spectrogram) are discussed and two practical synthesis methods are described, based on reconstruction using pure tones and random noise. Spectrogram generation and both described synthesis methods are implemented in the form of a computer program with a graphical user interface, allowing straightforward configuration of relevant parameters.

Keywords: spectrogram, synthesis, Fourier transform, sound

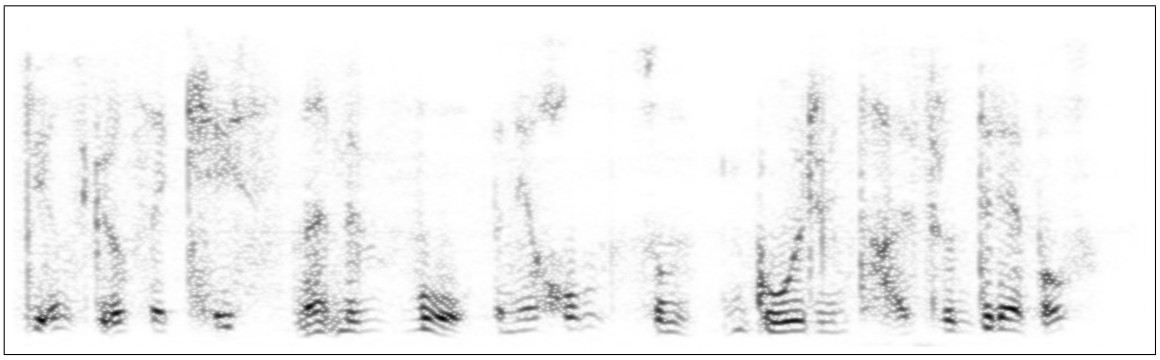
# Kapitola 1

# Úvod

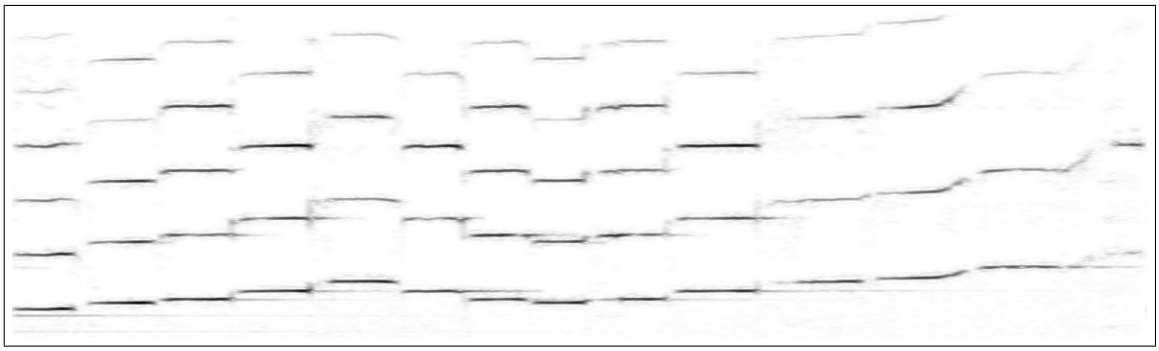
## 1.1 Co je to spektrogram

Spektrogram je graf, zobrazující intenzitu frekvenčních složek zvukového signálu v čase, kde čas je typicky na vodorovné ose, frekvence na svislé ose a intenzita je zobrazena zbarvením daného bodu. Jde o velice názornou reprezentaci zvuku v obrázkové podobě, ze které je možné nejen poznat charakter zvuku (zda jde o řeč, hudbu se zpěvem či bez zpěvu, zvuky přírodního nebo umělého původu), ale také obsahuje dost informací pro převod zpět do zvuku.

Spektrogram nezachovává všechny informace z původní nahrávky, proto je převod do grafu i zpět nevyhnutelně ztrátový. Ze zpětně vytvořeného zvuku je přesto možno identifikovat konkrétní hudební skladbu nebo porozumět slovům zaznamenaným původní nahrávkou.



Obrázek 1.1: Ukázka spektrogramu lidské řeči (4.4 s).



Obrázek 1.2: Ukázka spektrogramu krátkého záznamu hraní na housle (5.8 s). Mimo základních tónů jsou vidět alikvótní tóny vyšších frekvencí.

## 1.2 Motivace

Spektrogramy jsou základním nástrojem analýzy zvuku s řadou praktických využití. Jako názorná vizualizace se používají například pro studium ptačího zpěvu [1] nebo křiku novorozeňat [2]. Spektrogramy mohou být využity lidmi s poruchou sluchu pro učení přirozené řeči [3]. Poskytují totiž obrázkovou zpětnou vazbu, kterou je možno srovnat se spektrogramem zachycujícím správnou výslovnost. Podobným způsobem jsou využívány také pro vývoj algoritmů rozpoznávání [4] a syntézy řeči [5]. S tréninkem je dokonce možno rozeznávat slabiky řeči a přímo ze spektrogramu číst slova, která zachycuje [6]. Spektrogramy lze také použít pro přepis hudby do not [7]. Každá nota odpovídá určité frekvenci, jejíž přítomnost je na spektrogramu snadno vidět (obrázek 1.2).

Podrobnější informace a další příklady využití spektrogramů jsou popsány např. v [8].

Syntéza spektrogramů zpět do zvuku poskytuje další možná využití. Spektrogram je možné upravovat jako obrázek a využít tak možnosti obrázkových editorů pro manipulaci se zvukem. Jednoduchým „natažením“ spektrogramu a zpětnou syntézou lze změnit rychlost nahrávky bez frekvenčního posunu, který nastane při podobné úpravě provedené přímo na zvuku. S využitím různých grafických filtrů a úprav lze dosáhnout zajímavých zvukových efektů. Ve spektrogramu je také možné poměrně snadno oddělit jednotlivé nástroje či hlas z hudební nahrávky a zpětnou syntézou získat samostatné přesně izolované stopy.

Spektrogram samozřejmě nemusí vycházet ze zvukové nahrávky, kreslením do prázdného obrázku lze přímo vytvořit umělé, nebo přírodu napodobující zvuky „z ničeho“. Jakýkoli obrázek je nakonec možné interpretovat jako spektrogram a převést na zvuk. Takto si člověk může poslechnout například svůj oblíbený fraktál. Spektrogramy vycházející ze skutečného zvuku či hudby ale mají poměrně charakteristickou podobu, které se běžné obrázky většinou příliš nepodobají, proto z jsou nich vzniklé zvuky obvykle chaotické a nelze v nějakém obrázku očekávat objev skutečných hudebních kvalit. Je ale

možné obrázek tímto způsobem převést na zvuk, přenést analogovým médiem a s určitou ztrátou kvality znovu zrekonstruovat vytvořením spektrogramu.

Většina těchto využití je nicméně limitována kvalitou syntézy a ztrátovostí samotného převodu do spektrogramu.

## 1.3 Struktura práce

Ve druhé kapitole jsou definovány základní pojmy, diskutován jejich vztah k řešenému problému a popsány existující řešení. Třetí kapitola popisuje teorii, na které je založeno dále navržené řešení. Metody tvorby a syntézy spektrogramů použité v programu, který byl vytvořen v rámci této práce, jsou popsány ve čtvrté kapitole. Dokumentace k vytvořenému programu se nachází v páté kapitole. V závěru jsou diskutovány dosažené výsledky a navrženy možná budoucí rozšíření.

# Kapitola 2

# Analýza problému

## 2.1 Krátký úvod do digitálního zvuku

Spektrogram lze obecně vytvořit z libovolného diskrétního signálu, nemusí jít o záznam zvuku, ale toto použití je nejobvyklejší a velmi názorné.

Zvuk, jakožto oscilace tlaku v čase, je převeden na elektrický signál pomocí mikrofonu. Pro převod z analogové podoby do digitální se používá tzv. *pulzně kódová modulace* (PCM, pulse-code modulation), která spočívá v odečítání hodnoty signálu v pravidelných intervalech (vzorkování) a převodu odečtených hodnot do konečné binární podoby (kvantizace). Digitální záznam zvuku je tedy posloupnost zaokrouhlených hodnot (vzorků) s danou vzorkovací frekvencí, běžně 48 000 Hz (vzorků za sekundu).

Jedním ze základních výsledků teorie informace je Nyquistův-Shannonův teorém [9]:

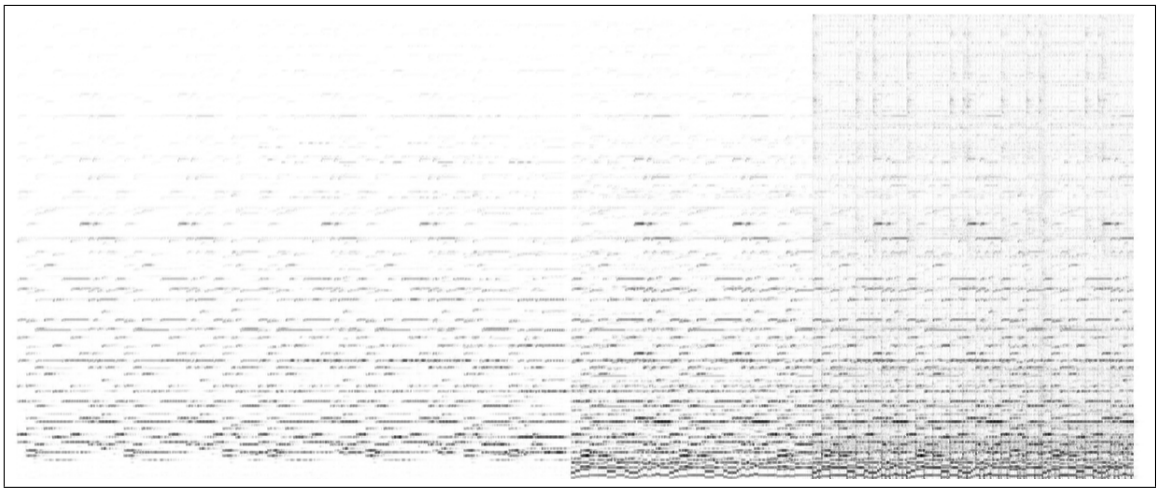
**Věta 1** (Nyquistův-Shannonův teorém) *Přesná rekonstrukce spojitého, frekvenčně omezeného signálu z jeho vzorků je možná tehdy, pokud byl vzorkován frekvencí alespoň dvakrát vyšší, než je maximální frekvence rekonstruovaného signálu.*

Při nedostatečném vzorkování dochází k takzvanému *aliasingu*, kdy jsou vyšší frekvence reprezentovány chybně a vedou ke znatelnému zkreslení signálu. Pro předejití aliasingu se ze signálu odstraní vysoké frekvence pomocí dolní propusti (low-pass filter).

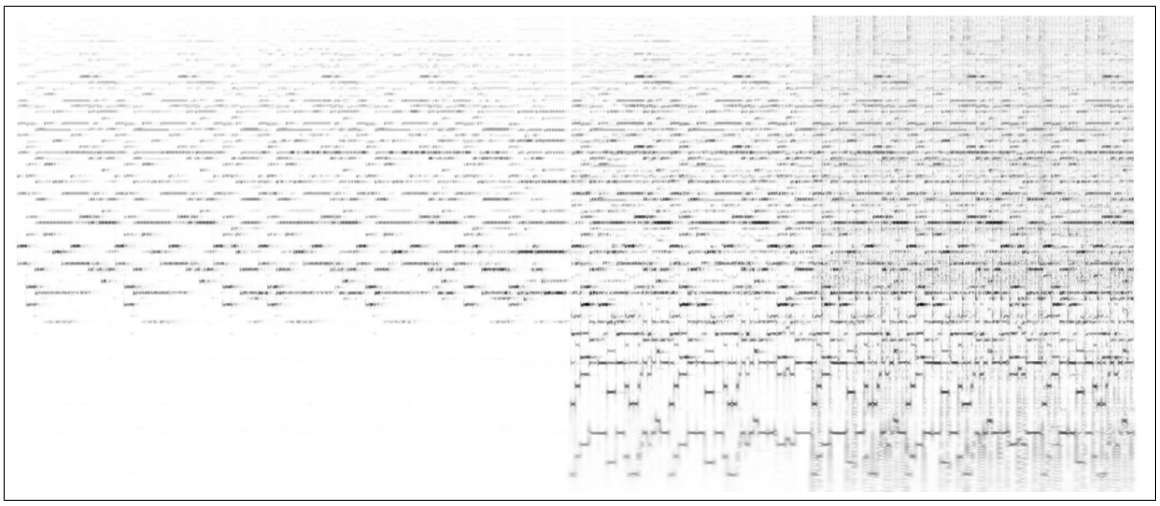
*Nyquistova frekvence* je polovina vzorkovací frekvence diskrétního signálu. Odpovídá horní mezi frekvencí, které jsou z tohoto signálu dokonale rekonstruovatelné.

## 2.2 Logaritmické a lineární spektrogramy

V této práci budou často rozlišovány dva základní druhy spektrogramů – logaritmické a lineární. Toto označení se vztahuje na frekvenční (svislou) osu spektrogramu. V lineárním spektrogramu zobrazuje každý vodorovný pruh stejně velký interval frekvencí



Obrázek 2.1: Lineární spektrogram části hudební skladby (2 min).



Obrázek 2.2: Logaritmický spektrogram části hudební skladby (2 min).

měřených v Hertzech. V logaritmickém spektrogramu se intervaly frekvencí zobrazené jedním vodorovným pruhem zvětšují exponenciálně se základem 2. Každý pruh logaritmického spektrogramu zobrazuje konstantní počet *centů*, přičemž cent je definován jako $1/1\,200$ oktávy.

Logaritmický spektrogram zobrazuje zvukový záznam blíže tomu, jak jej vnímá lidské ucho. Citlivost lidského ucha na zvyšující se frekvence totiž, jak popisuje např. [10], postupně klesá přibližně logaritmickou křivou až do cca. 1 000 Hz, kolem 2 000 Hz je lokálně zvýšená citlivost na frekvence kolem kterých se vyskytuje lidský hlas a nad 4 000 Hz pak citlivost kolísavě stoupá až do maximální slyšitelné frekvence, která je přibližně 20 000 Hz (se stářím člověka se postupně snižuje).

Z těchto informací vyplývá, že jsou pro lidský sluch mnohem významnější nižší frekvence, kterým je v logaritmickém spektrogramu přiřazeno výrazně více prostoru než

v lineárním, kde jsou při velkém zobrazovaném rozsahu „stlačeny“ ve spodní části, jak ilustrují obrázky 2.1 a 2.2, které zobrazují stejný rozsah frekvencí (od nuly do 6 000 Hz).

Pro zobrazení menšího rozsahu frekvencí, jako například záznamu lidského hlasu, může být lineární spektrogram také vhodný. Příkladem takového použití je obrázek 1.1.

## 2.3 Existující implementace

Existuje řada programů umožňujících vytvořit spektrogram zvukové nahrávky. Velmi pokročilý je například Baudline[1], který kromě spektrogramů nabízí celou sadu nástrojů pro analýzu signálů. Program SFS/RTGRAM[2] se zaměřuje na tvorbu spektrogramů lidské řeči v reálném čase. Existují také sady skriptů pro MATLAB, které umí tvořit různé druhy spektrogramů. Mnoho z těchto programů ale nemá pohodlné uživatelské rozhraní nebo nabízí jen omezené možnosti nastavení parametrů spektrogramů.

Zpětnou syntézou spektrogramů se zabývají pouze dva programy. ARSS[3], který již není dále vyvíjen, umí pracovat jen s formáty wav a bmp a poskytuje pouze konzolové rozhraní, což jej činí poněkud nepraktickým. Program ARSS přešel v komerční Photosounder[4], který se zaměřuje na úpravy zvuku pomocí přímé editace spektrogramů.

[1] Webová stránka Baudline: `http://www.baudline.com/`
[2] Webová stránka SFS/RTGRAM: `http://www.phon.ucl.ac.uk/resource/sfs/rtgram/`
[3] Webová stránka ARSS: `http://arss.sourceforge.net/`
[4] Webová stránka Photosounderu: `http://photosounder.com/`

# Kapitola 3

# Teoretické základy

## 3.1 Fourierova transformace

Základním nástrojem pro práci se signály je Fourierova transformace. Pro tuto práci nás bude zajímat především použití jednorozměrné Diskrétní Fourierovy Transformace (DFT) pro převod reálné posloupnosti (signálu) z časové oblasti do oblasti frekvenční.

Matematicky může být zapsána následovně:

$$Y_k = \sum_{j=0}^{n-1} X_j \cdot e^{\frac{-2\pi i j k}{n}} \tag{3.1}$$

Kde $i$ je imaginární jednotka, $X$ je vektor reálných čísel délky $n$ a $Y$ je výsledný vektor komplexních čísel délky $n$, kde polovina hodnot je redundantní díky symetrii $Y_k = \overline{Y_{n-k}}$ ($\bar{x}$ značí komplexně sdružené číslo), která je pro reálný vstupní vektor zaručena. Zajímat nás tedy budou hodnoty $Y_0$ až $Y_{\lfloor n/2 \rfloor}$.

Inverzní Fourierova transformace je definována takto:

$$X_k = \frac{1}{n} \sum_{j=0}^{n-1} Y_j \cdot e^{\frac{2\pi i j k}{n}} \tag{3.2}$$

Kde $Y$ je komplexní vektor splňující výše popsanou symetrii a $X$ pak zpětně získaný vektor reálných čísel.

Převod z časové do frekvenční oblasti znamená, že signál popsaný svou hodnotou v čase je analyzován vzhledem k čistě sinusovým složkám různých frekvencí a fázových posunů, ze kterých lze složit. Komplexní čísla ve vektoru získaném transformací představují po převedení do polárních souřadnic právě amplitudu a fázový posun sinusoid jednotlivých frekvencí, určených indexem. Dříve zmíněný Nyquistův-Shannonův teo-

rém (1) říká, že diskrétní signál vzorkovaný frekvencí $f$ půjde reprezentovat a dokonale rekonstruovat pomocí $\lfloor f/2+1 \rfloor$ frekvenčních složek.

## 3.2 Krátkodobá Fourierova transformace

Jednoduchý lineární spektrogram lze vytvořit přímočarou aplikací Fourierovy transformace na po sobě jdoucí krátké časové intervaly. Taková transformace se nazývá krátkodobá Fourierova transformace (Short-Time Fourier Transform, STFT). Výsledné transformované (komplexní) vektory v absolutní hodnotě pak představují svislé pruhy vzniklého spektrogramu. Tato metoda je ve výsledku prakticky ekvivalentní s metodou popsanou dále v této práci. Její výhodou je možnost provádět převod přímočaře v reálném čase, tedy na signálu, který není celý znám už na začátku. Nevýhodou je složitější tvorba logaritmických spektrogramů.

Běžně používaný způsob, jak vylepšit vzhled a informativnost STFT spektrogramu je aplikace transformací na překrývající se intervaly časových vzorků s utlumením překrytých okrajů pomocí okenní funkce, místo přímé transformace disjunktních, po sobě jdoucích intervalů. Tímto se dosáhne lepšího rozlišení, plynulejších náběhů a doběhů zobrazovaných tónů a zabrání se artefaktům, které by jinak vznikaly na ostrých přechodech mezi jednotlivými intervaly. Obdobný postup je použit i v metodě popsané dále v této práci, pouze aplikovaný na frekvenční oblasti místo časové.

## 3.3 Informace ve spektrogramu ztracené

Jak již bylo naznačeno, reprezentace signálu spektrogramem není bezztrátová, proto není ani teoreticky možné z něj původní signál rekonstruovat dokonale.

Pro čtení a rekonstrukci spektrogramu je především nutné znát parametry, se kterými byl vytvořen, zejména pak měřítka časové a frekvenční osy. Samotný spektrogram tyto informace neobsahuje, je proto nutné je přidat v podobě popisků nebo metadat. Bez těchto informací není možné zjistit jak je zachycený signál dlouhý či jaké frekvence zobrazuje.

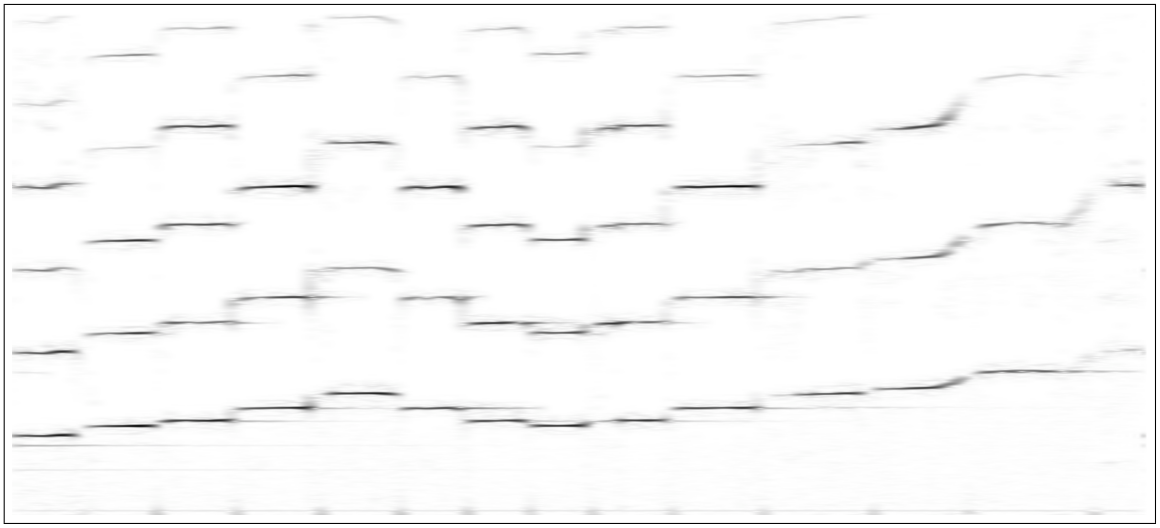
Významná část informace o původním signálu je fázový posun jednotlivých frekvenčních složek. Tento není ve spektrogramu nijak zachycen. Byly popsány způsoby aproximace původního fázového posunu za přijmutí určitých předpokladů o výchozím signálu, např. v [11]. Metody popsané v této práci se o rekonstrukci fázového posunu nepokouší. Předchozí práce jako [12] ukazují, že lze dosáhnout uspokojivých výsledků i bez rekonstrukce fázového posunu.

Základní vlastností spektrogramu je vztah rozlišení frekvenční a časové osy. Zatímco pro spojitý signál by šla přesnost obou os zvolit libovolně, pro signál s omezeným počtem

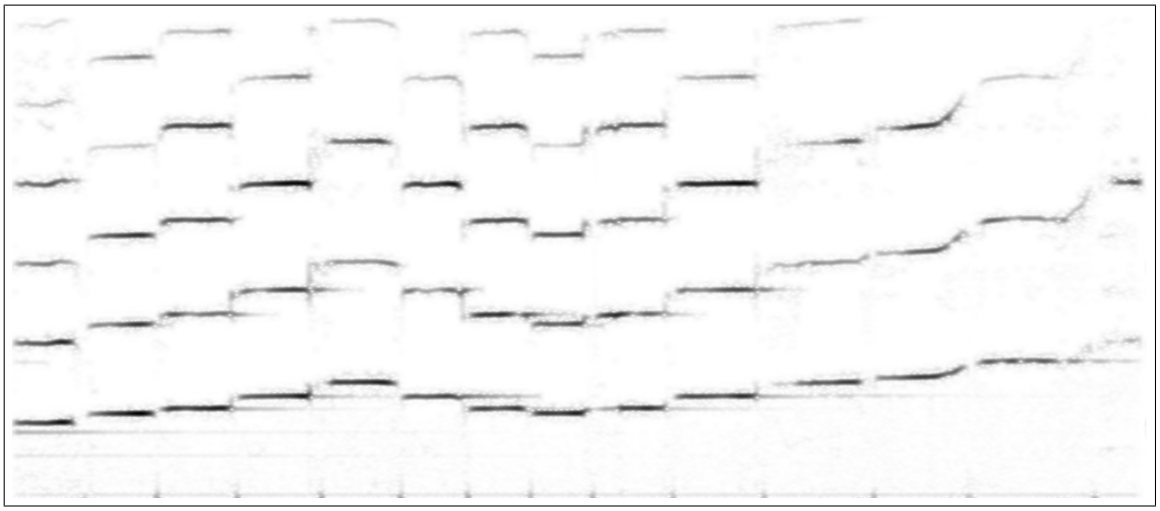
vzorků je rozlišení jedné osy pevně dané a nepřímo úměrné rozlišení druhé. Při tvorbě spektrogramu je proto nutné se rozhodnout pro vhodný kompromis. Tato volba závisí na charakteru analyzovaného signálu a účelu tvořeného spektrogramu. Obrázky 3.1, 3.2 a 3.3 ilustrují její důsledky. U obrázku 3.1 jsou frekvence hraných tónů určeny přesněji než na obrázku 3.3, kde je možné podrobněji vidět, jak frekvence a intenzity některých tónů lehce kolísají. Dva extrémy této volby jsou čistě časová reprezentace, což je nezměněný původní signál (maximální rozlišení časové osy, ale pouze jeden „pruh“ pokrývající všechny frekvence) a čistě frekvenční reprezentace, což je původní signál po Fourierově transformaci (maximální rozlišení frekvencí bez lokalizace v čase).

Tato vlastnost je důsledkem faktu, že pro získání podrobnějšího rozlišení na frekvenční ose, pro jednoduchost například v lineárním STFT spektrogramu, potřebuji Fourierovou transformací získat delší komplexní vektor (který bude představovat svislý pruh spektrogramu) a musím proto transformaci aplikovat na delší intervaly vzorků z časové oblasti. Celkový počet intervalů proto bude menší a tím se zhorší rozlišení časové osy. Jde o jev podobný Heisenbergovu principu neurčitosti týkající se přesnosti určení polohy a hybnosti elementární částice v kvantové fyzice, resp. nemožnosti určit obě tyto veličiny současně s libovolnou přesností.

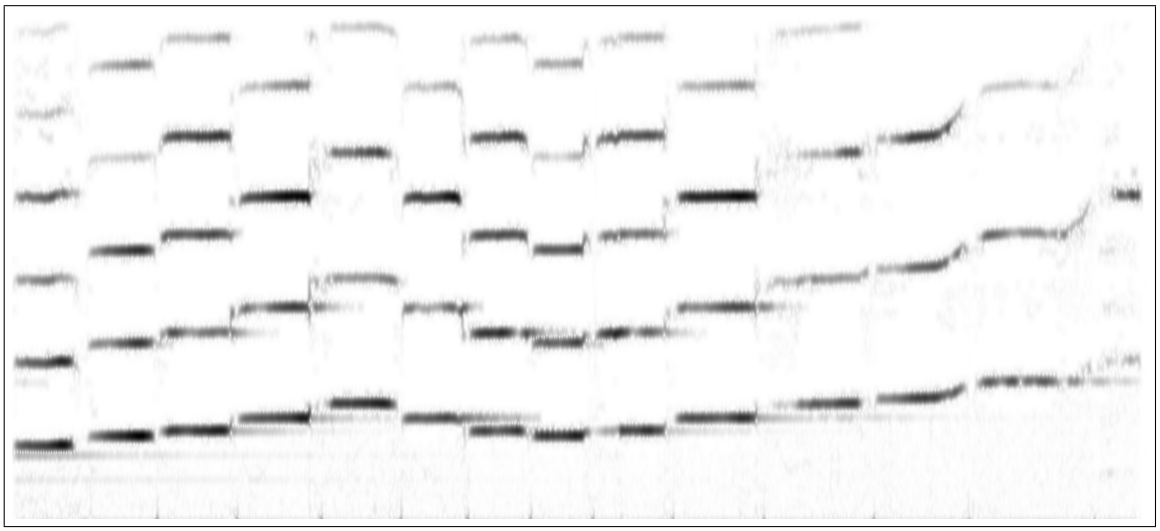
Spektrogram dále způsobí zaokrouhlovací chyby z důvodu agresivnější kvantizace a vzorkování. Zatímco původní digitální záznam reprezentuje hodnotu každého vzorku číslem délky například 16 bitů (běžný nekomprimovaný soubor ve formátu wav), ve spektrogramu je přesnost záznamu intenzity pixelu omezena počtem barev použitých pro jeho vykreslení. Pro černobílý spektrogram je to typicky 256 odstínů šedi, tedy 8 bitů. Zaokrouhlení časové informace převzorkováním se pak nejvíce projeví u vysokých frekvencí v logaritmickém spektrogramu. Fourierova transformace aplikovaná na signál sice vrátí právě dostatečný počet vzorků potřebných pro jeho rekonstrukci, ale šířka spektrogramu by byla příliš veliká, pokud by měl zobrazovat všechny získané vzorky u nejvyšších frekvencí a nižší frekvence by naopak byly zbytečně roztaženy. Zvolením rozumné šířky spektrogramu se proto připravíme o možnost určité frekvence rekonstruovat přesně.



Obrázek 3.1: Spektrogram s vysokým rozlišením frekvenční osy.



Obrázek 3.2: Spektrogram s kompromisní volbou rozlišení os.



Obrázek 3.3: Spektrogram s vysokým rozlišením časové osy.

# Kapitola 4

# Popis metod

## 4.1 Metody tvorby spektrogramu

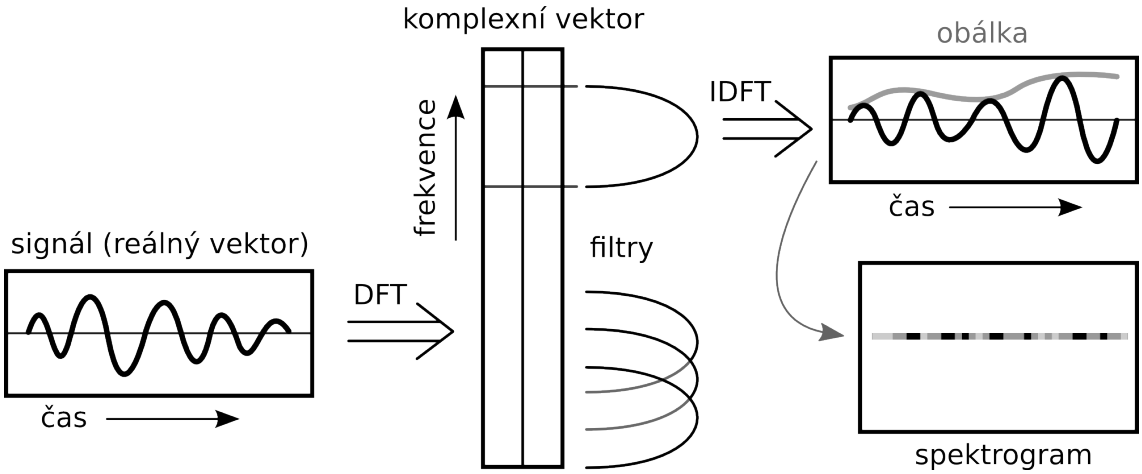
Program vypracovaný v rámci této práce používá pro tvorbu spektrogramu metodu převedení celého signálu do frekvenční oblasti a rozdělení frekvenčního spektra na intervaly, které jsou jednotlivě převedeny zpět do časové oblasti. Jejich obálky pak tvoří vodorovné pruhy spektrogramu. Schématicky je tento proces znázorněn na obrázku 4.1. Oproti výše popsané STFT metodě je tato podobná spíše analogovému postupu, kdy je signál rozdělen pomocí frekvenčních pásmových propustí a vodorovné pruhy spektrogramu jsou tvořeny obálkami jednotlivých částí.

Tato metoda poskytuje naprostou volnost v rozložení a velikosti jednotlivých frekvenčních intervalů, jde proto snadno použít pro tvorbu jak lineárních, tak logaritmických spektrogramů.

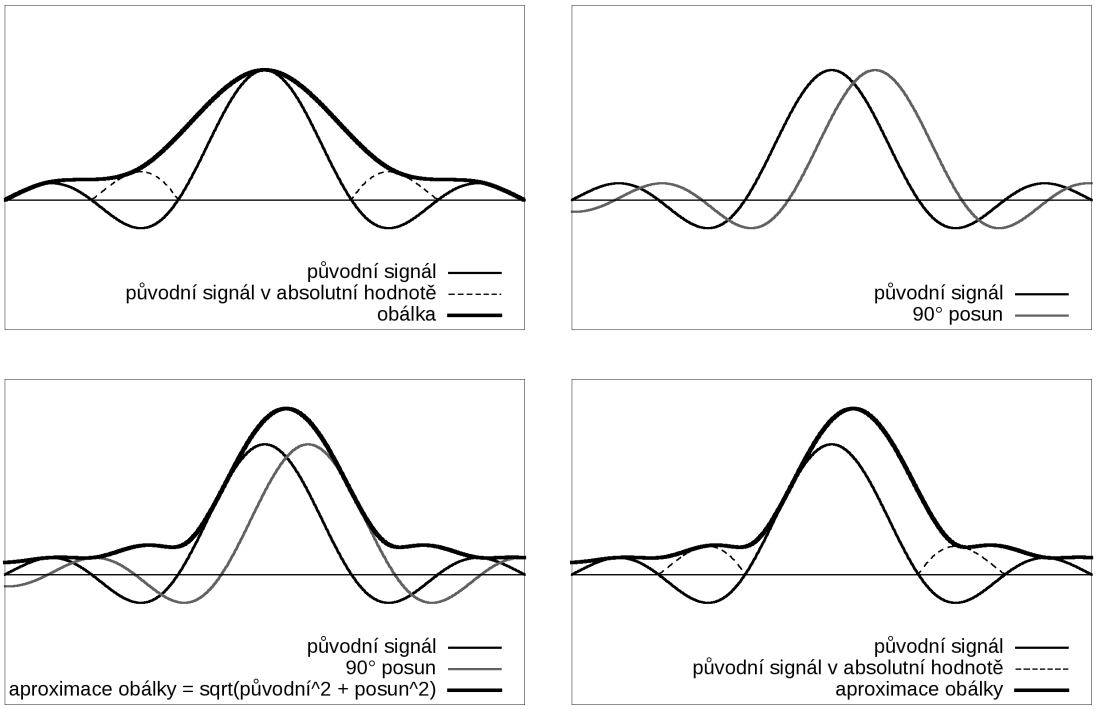
Vstupem je reálný vektor délky $n$ představující zkoumaný signál vzorkovaný známou frekvencí $f$. Tento je Fourierovou transformací převeden do frekvenční oblasti. Získáme komplexní vektor délky $\lfloor n/2+1 \rfloor$, jehož prvky představují jednotlivé frekvenční složky od 0 Hz do $(f/2)$ Hz, hranice dané Nyquistovou-Shannonovou větou (1).

Vektor představující frekvenční oblast je potom rozdělen na intervaly (filtry), jejichž počet je dán požadovaným rozlišením frekvenční osy spektrogramu a zadanou mírou jejich překrytí. Pro lineární spektrogram mají filtry konstantní velikost a všechny tedy odpovídají stejné šířce pásma v Hz. Pro logaritmický spektrogram se filtry zvětšují se stoupající frekvencí tak, aby pokrývaly konstantní počet oktáv (zadaných v centech). V obou případech se hranice každého frekvenčního intervalu v snadno přepočítají na odpovídající indexy vektoru.

V případě lineárního spektrogramu je počátek prvního filtru přímo dán *základní frekvencí*, specifikovanou uživatelem. V případě logaritmického spektrogramu odpovídá základní frekvence středu prvního filtru. Pro optimální zachycení hudby v logaritmickém




Obrázek 4.1: Schéma tvorby spektrogramu.




Obrázek 4.2: Aproximace obálky signálu.

spektrogramu je vhodné, když jsou filtry vystředěny na frekvencích, které odpovídají hudebním notám. Výchozí hodnota základní frekvence je proto 55 Hz, odpovídající $A_1$. Frekvence pod 30 Hz nejsou příliš důležité, protože ani lidské ucho na ně není citlivé. Další filtry budou správně vystředěny právě tehdy, pokud je šířka jednotlivých filtrů násobkem 1/12 oktávy, tedy 100 centů.

Na každý získaný interval je dále aplikována uživatelem zvolená okenní funkce a inverzní Fourierovou transformací je převeden zpět do časové oblasti. Pro zakreslení do spektrogramu je třeba získat obálku tohoto signálu. Použitá metoda detekce obálky přičte ke čtvercům hodnot zpětně transformovaného signálu čtverce hodnot jeho kopie s fázovým posunem o 90° vyšším (tento posun je ve frekvenční oblasti efektivní operace). Z výsledku lze pak obálku aproximovat odmocninou jeho hodnot. Tento proces ilustruje obrázek 4.2.

V případě logaritmického spektrogramu jsou výsledné obálky dílčích signálů různě dlouhé, jako by byly vzorkované s různou frekvencí, protože každý dílčí signál zachycuje jinak velký rozsah frekvencí. Spektrogram má ovšem konstantní šířku, proto jsou dílčí obálky převzorkovány podle uživatelem zadaného počtu pixelů za sekundu.

Výsledné obálky, které představují vodorovné pruhy spektrogramu, jsou do něj postupně zakresleny podle uživatelem zvolené palety barev. Čím více různých barev paleta obsahuje, tím méně jsou hodnoty signálu zaokrouhleny zobrazením na ni.

## 4.2 Metody zpětné syntézy spektrogramu

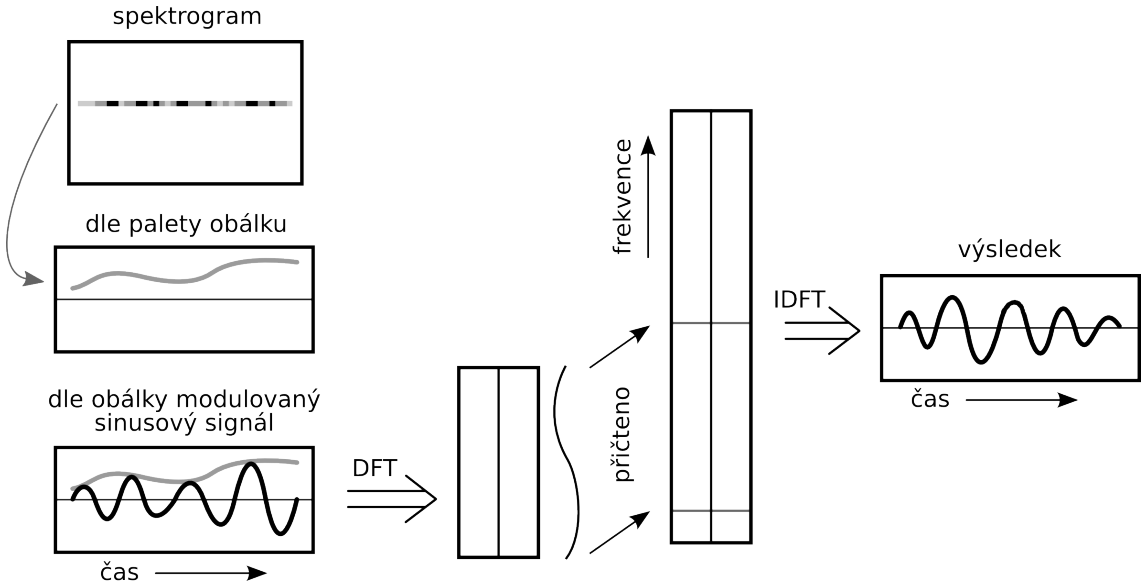
### Sinusová syntéza

Sinusová metoda využívá sinusové signály modulované obálkami z výchozího spektrogramu. Schématicky ji znázorňuje obrázek 4.3.

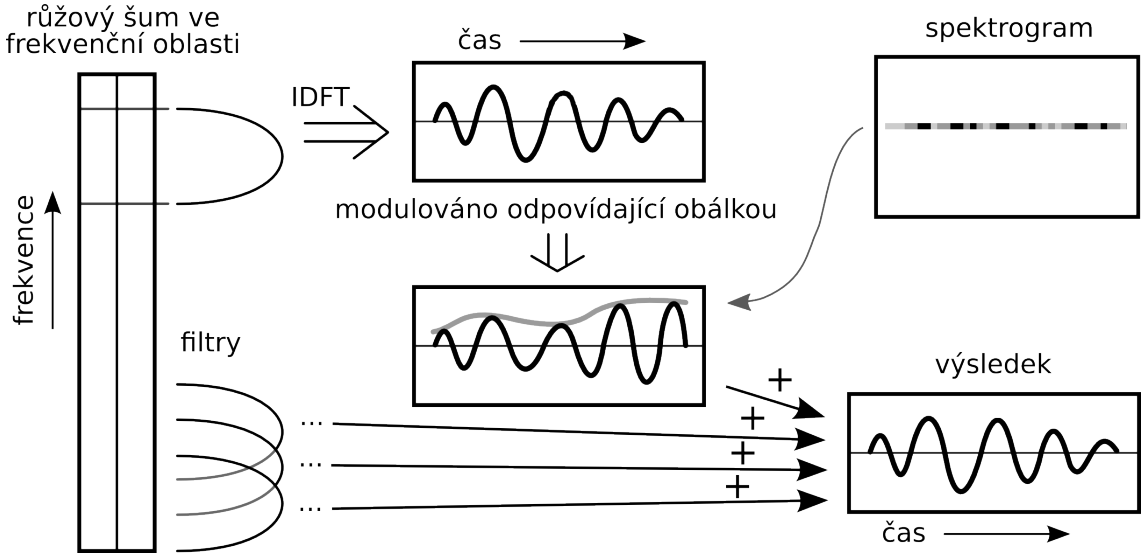
Zvuk je rekonstruován postupně ve frekvenční oblasti. Na počátku je vytvořen komplexní vektor odpovídající frekvenční oblasti výsledného zvuku. Rekonstrukce probíhá po vodorovných pruzích spektrogramu, které odpovídají intervalům frekvencí, jejichž střední frekvence jsou určeny parametry spektrogramu.

Každý pruh spektrogramu je dle známé palety barev převeden na vektor reálných hodnot popisující obálku dílčího frekvenčně omezeného signálu v časové oblasti. Obálkou je pak modulován sinusový signál s náhodným fázovým posunem (jak je uvedeno v teorii, posun obecně nelze ze spektrogramu rekonstruovat). Jsou použity čtyři vzorky na jednu periodu fázově posunuté sinusoidy – frekvence není podstatná, protože ve výsledku bude určená zařazením na příslušnou pozici ve frekvenční oblasti rekonstruovaného signálu.

Modulovaný signál je pak převeden Fourierovou transformací do frekvenční oblasti. Stejným způsobem jako při generování spektrogramu se vypočte střední frekvence daného




Obrázek 4.3: Schéma sinusové syntézy.




Obrázek 4.4: Schéma šumové syntézy.

vodorovného pruhu a index této frekvence v dílčím signálu. Se středem v odpovídající frekvenci ve frekvenční oblasti výsledku je do něj spektrum dílčího signálu začleněno.

Až jsou takto zpracovány všechny pruhy spektrogramu, je výsledek převeden inverzní Fourierovou transformací do časové oblasti. Tím je získán výsledný rekonstruovaný signál.

## Šumová syntéza

Šumová metoda používá pro rekonstrukci náhodný šum, který je, podobně jako čisté tóny v sinusové metodě, po částech modulován odpovídajícími obálkami z výchozího spektrogramu. Schématicky ji znázorňuje obrázek 4.4.

Výsledný zvuk je zde rekonstruován přímo v časové oblasti. Nejprve je vygenerován náhodný růžový šum (tedy šum se stejnou energií v každé oktávě) ve frekvenční oblasti. Tento šum je zpracován sadou frekvenčních filtrů se stejnými parametry, jako při tvorbě rekonstruovaného spektrogramu.

Inverzní Fourierovou transformací aplikovanou na vyfiltrované frekvenční intervaly je pro každý vodorovný pruh spektrogramu získán zvukový „nosič“ v časové oblasti.

Stejně jako v sinusové syntéze je pak každý pruh spektrogramu převeden dle známé palety na obálku odpovídajícího dílčího signálu. Touto obálkou je modulován odpovídající nosič získaný z růžového šumu.

Výsledný signál je rekonstruován sečtením všech modulovaných dílčích signálů.

## Srovnání

Nedostatkem sinusové metody je, že nezachovává spojitost signálu na frekvenční ose, protože mnoho frekvencí, které zachycuje jeden vodorovný pruh spektrogramu, je rekonstruováno pouze pomocí sinusového signálu střední frekvence daného pruhu. Pro tóny s konstantní frekvencí se tento nedostatek příliš neprojeví, ale pro „rušné“ zvuky jako úder činelu či kolísající frekvence je reprodukce méně věrná. Šumová syntéza tímto nedostatkem netrpí, ale nastává u ní problém zašumění výsledného zvuku, jelikož části náhodného šumu jsou použity jako základ rekonstrukce.

# Kapitola 5

# Dokumentace

## 5.1 Uživatelská dokumentace

### Instalace

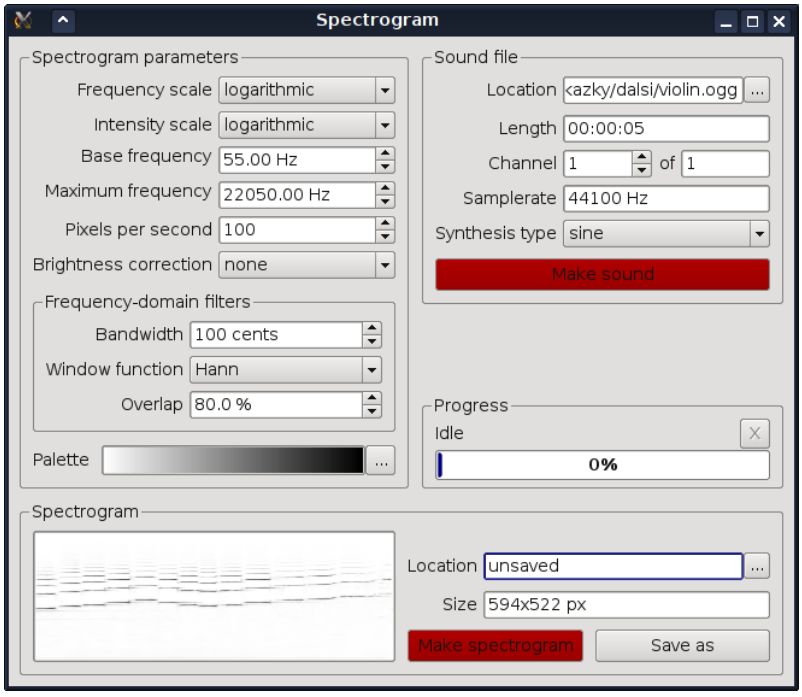
Pro platformu Windows je program přiložen v podobě přímo spustitelného souboru a není třeba jej instalovat. Pro platformu Linux jsou přiloženy kompletní zdrojové kódy. Návod pro jejich překlad je v programátorské části dokumentace.

### Použití

Veškerá funkcionalita programu je k dispozici přímo z hlavního okna (obr. 5.1), které je zobrazeno po spuštění. V tomto okně lze nastavit parametry pro generování spektrogramů a jejich syntézu a zvolit vstupní data. Průběh déle trvajících operací je indikován na pravé straně okna s možností operaci přerušit stiskem tlačítka označeného „X“.

Pro vytvoření spektrogramu nejprve zvolte vstupní zvukový soubor zadáním cesty do kolonky „Location“ v části „Sound file“, případně ho vyberte stiskem tlačítka napravo od kolonky pro cestu. Po vybrání platného souboru je zobrazena jeho délka, vzorkovací frekvence a je možno vybrat kanál, který bude použit pro tvorbu spektrogramu. Ve stereo souborech odpovídá první kanál levé zvukové stopě a druhé pravé. Pak nastavte požadované parametry spektrogramu na levé straně okna. Význam jednotlivých voleb:

- Frequency scale – Zde je možno zvolit tvorbu lineárního nebo logaritmického spektrogramu.
- Intensity scale – Lze nastavit lineární nebo logaritmické zobrazení intenzity. Obecně je vhodné ponechat na výchozí volbě (logarithmic).




Obrázek 5.1: Hlavní okno programu.

- Base frequency – Základní frekvence označuje v logaritmickém spektrogramu střed prvního frekvenčního filtru (vodorovného pruhu). Pro analýzu hudebních nahrávek je vhodné hodnotu nastavit na frekvenci hudební noty, např. výchozích 55 Hz odpovídá $A_1$. V lineárním spektrogramu označuje nejnižší zobrazenou frekvenci.
- Maximum frequency – Frekvence nad zadanou mez nebudou ve spektrogramu zachyceny.
- Pixels per second – Ovlivňuje šířku výsledného spektrogramu. Čím vyšší hodnota, tím podrobněji je zobrazena časová osa. Pro zpětnou syntézu je vhodné spektrogram vytvořit s hodnotou alespoň 150 pps.
- Brightness correction – Některé spektrogramy mohou být příliš tmavé i s logaritmickou osou intenzity. Použití korekce usnadní čtení spektrogramu, ale ovlivní zpětnou syntézu.
- Bandwidth – Určuje velikost intervalů frekvencí představujících vodorovné pruhy spektrogramu. Menší hodnota znamená vyšší rozlišení ve frekvenční oblasti, ale menší rozlišení v oblasti časové.
- Window function – Umožňuje zvolit okenní funkci použitou jednotlivými frekvenčními filtry.

- Overlap – Udává míru překrytí jednotlivých frekvenčních filtrů. Pokud není použita okenní funkce (volba „none“), je možno ji nastavit na nulu, jinak je doporučeno překrytí alespoň 60%. Čím vyšší hodnota, tím podrobnější je zobrazení frekvenční osy.

- Palette – Určuje paletu barev, která bude použita pro vykreslení spektrogramu. Intenzita stoupá směrem do prava. Je možné zvolit vlastní paletu barev z obrázku, jako paleta je použit jeho první řádek. Pro syntézu je vhodné, aby se barvy v paletě neopakovaly. Několik ukázkových palet je přiloženo.

Jakmile jsou požadované parametry nastaveny, spektrogram lze vytvořit kliknutím na červené tlačítko „Make spectrogram“. Výsledek je možno uložit tlačítkem „Save as“.

Syntéza se provádí obdobně jako tvorba spektrogramu, jen místo zvukového souboru je nejprve načten spektrogram (v části „Spectrogram“). Pokud byl spektrogram vytvořen tímto programem, jsou jeho parametry automaticky vyplněny. V opačném případě je nutno je vyplnit ručně pokud možno co nejblíže těm se kterými byl daný spektrogram vytvořen. Program nabízí dvě metody syntézy, které je možno vybrat v nabídce „Synthesis type“:

- sine – Sinusová syntéza je velmi rychlá a dává obecně dobré výsledky.

- noise – Šumová syntéza je pomalejší, ale pro náročnější spektrogramy (např. hudba o více nástrojích) může vytvořit věrnější rekonstrukci než sinusová metoda.

Když je vše nastaveno, syntéza se spustí tlačítkem „Make sound“. Po dokončení je nabídnuta možnost výsledný zvukový soubor uložit.

## Doporučené parametry

Volba vhodných parametrů pro spektrogram záleží především na účelu spektrogramu a na charakteru analyzovaného signálu. Obecně je vhodné omezit maximální frekvenci podle vstupního signálu. Pokud jde například o záznam lidské řeči, není nutné volit maximální frekvenci nad 8 000 Hz, pravděpodobně bude stačit i méně. Vhodná volba lze poznat podle toho, že v horní části spektrogramu není zbytečně velká prázdná oblast. Naopak při příliš nízké volbě je na spektrogramu viditelné, že je z něj horní část „odříznuta“. Pro signály s menším rozsahem frekvencí, jako tomu je u lidské řeči, může lineární spektrogram vypadat lépe než logaritmický, který je dobrou volbou například pro hudbu. V případě hudby je také vhodné volit šířku pásma frekvenčních filtrů („Bandwidth“) v násobcích 100 centů a jako základní frekvenci zvolit notu $A_1$ (55 Hz) nebo $A_0$ (27,5 Hz).

U spektrogramů určených pro názornou vizualizaci signálu obvykle stačí menší hodnota počtu pixelů za sekundu, např. 50, což má také výhodu v tom, že výsledný spektrogram

nebude příliš široký. Touto volbou se zhorší zobrazení podrobností na časové ose, proto je možné si dovolit lepší rozlišení na frekvenční ose a tím i větší výšku spektrogramu a příznivější poměr jeho stran. Dobrá hodnota šířky pásma frekvenčních filtrů je proto např. 50 centů nebo 75 Hz pro lineární spektrogram. Pro zvýraznění méně patrných detailů je vhodné zkusit povolit korekci zesvětlení („Brightness correction“). S touto volbou bude spektrogram světlejší a budou čitelné i oblasti s menší intenzitou.

Spektrogramy určené pro úpravy a zpětnou syntézu naopak vyžadují větší počet pixelů za sekundu, nejlépe 200 a více. Vyplatí se také dát přednost rozlišení časové osy před osou frekvenční s šířkou pásma frekvenčních filtrů 100, 200 nebo 400 centů. Pro kompenzaci zmenšení výšky spektrogramu je možno zvýšit míru překrytí filtrů („Overlap“) např. na 95%.

### Podporované formáty

Program podporuje většinu běžně používaných zvukových formátu, jako např. mp3, wav, flac či ogg. U některých mp3 souborů (s proměnným datovým tokem) se nemusí zobrazit jejich správná délka, ale pro tvorbu spektrogramů je lze bez problémů použít.

Podporována je také většina běžných formátů obrázků, např. png, bmp, tiff, xpm, jpg (pro čtení), gif (pro čtení).

## 5.2 Programátorská dokumentace

### Překlad kódu

Program, který byl vytvořen v rámci této práce, je napsán v jazyce C++ s grafickým rozhraním využívajícím knihoven Qt4 a systémem CMake pro automatizovaný překlad a konfiguraci. Kód i použité knihovny jsou multiplatformní a je možné je přeložit na platformách Windows i Linux.

Překlad lze provést pomocí překladače g++ na platformě Linux nebo pomocí MinGW na platformě Windows. Pro překlad jsou potřeba následující závislosti:

- CMake (systém pro překlad):
  `http://www.cmake.org`
- Qt4 (pro grafické rozhraní):
  `http://www.qtsoftware.com/products`
- FFTW (single-precision verze, použitá pro Fourierovou transformaci):
  `http://www.fftw.org`

- SRC (neboli libsamplerate, pro převzorkování signálů):
  `http://www.mega-nerd.com/SRC/`
- libsndfile (pro podporu mnoha zvukových formátů – pro podporu formátů flac a ogg je potřeba libsndfile verze alespoň 1.0.18 se zabudovanými odpovídajícími moduly):
  `http://mega-nerd.com/libsndfile`
- MAD (pro podporu formátu mp3):
  `http://www.underbit.com/products/mad/`

Na platformě Debian je možné použít následující příkaz pro nainstalování všech závislostí:

```
apt-get install cmake libqt4-dev libfftw3-dev libsndfile-dev
libsamplerate-dev libmad0-dev
```

Pro překlad na platformě Linux přejděte do adresáře „build“ ve stromu zdrojových kódů a spusťte příkaz „`cmake ..`“. Pokud nedojde k žádné chybě (obvykle chybějící knihovny), je možno program sestavit příkazem „`make`“. Spustitelný soubor „spectrogram“ bude vytvořen v adresáři „build“.

Pro překlad na platformě Windows je třeba mít kromě uvedených závislostí ještě MinGW a MSYS (`http://www.mingw.org/`). Jakmile jsou přeloženy a nainstalovány všechny závislosti, spusťte program „cmake-gui“, zvolte adresář se zdrojovými kódy a použijte adresář „build“ pro umístění binárních souborů. Potom zmáčkněte tlačítko „Configure“ a použijte generátor „MSYS Makefile“. Pro každou knihovnu která nebyla automaticky nalezena zadejte její cestu a znovu proveďte konfiguraci. Když je konfigurace hotová, zmáčkněte tlačítko „Generate“. Pro překlad programu přejděte v MSYS konzoli do adresáře „build“ a spusťte překlad příkazem „`make`“.

## Orientace v kódu

Hlavní třídy a funkce jsou zdokumentovány pomocí systému Doxygen. Vygenerovaná dokumentace je přiložena na CD (viz příloha A).

Nejzajímavější třídou programu je třída *spectrogram*. Tato drží parametry a implementuje metody tvorby a syntézy spektrogramu. Tvorbu spektrogramu provádí metoda *make_image()*, syntézu pak *sine_synthesis()* a *noise_synthesis()*. Pro uložení parametrů do metadat obrázku spektrogramu se používá metoda *serialized()*, která parametry převede na řetězec, ze kterého je lze zpětně načíst metodou *deserialize()*.

Tvorba i syntéza spektrogramu využívá abstraktní třídy *Filterbank*, jejíž účelem je obecné dělení frekvenční oblasti (resp. vektoru který ji představuje) na vhodné intervaly

podle zadaných parametrů. Dvě její implementace jsou *LinearFilterbank*, která provádí dělení pro lineární spektrogram a *LogFilterbank* pro spektrogram logaritmický. Tyto třídy obsahují metodu *get_band()*, která pro index frekvenčního filtru vrátí indexy vektoru frekvenční oblasti odpovídající příslušnému (vypočítanému) frekvenčnímu intervalu (dvojici hodnot od-do). Pomocí metody *get_center()* umí také vypočítat index vektoru frekvenční oblasti, který odpovída středu daného frekvenčního intervalu.

Převod hodnot intenzity na barvy spektrogramu a zpět řeší třída *Palette*. Obsahuje vektor barev, které jsou použity pro kresbu spektrogramu. Metodou *get_color()* vybírá vhodnou barvu pro zadanou intenzitu a opačně *get_intensity()* převede barvu zpět na intenzitu pro účely syntézy. Tato třída také zajistí vytvoření plátna (prázdného obrázku) pro spektrogram metodou *make_canvas()*. Podle počtu barev, které paleta obsahuje, je pro obrázek zvolen buď indexovaný mód (úspornější, pokud je barev méně než 256), nebo RGB mód.

Kód, který se týká grafického rozhraní programu se nachází v souborech mainwindow.cpp a mainwindow.hpp. Třída *MainWindow* reprezentuje hlavní okno programu, umožňuje uživateli zadat vstupní data a prezentuje výstupy. Rozvržení hlavního okna je vytvořeno pomocí programu Qt Designer v souboru mainwindow.ui. Toto rozvržení je v třídě *MainWindow* dostupné přes člen *ui*. Parametry spektrogramu jsou dostupné přes ukazatel *spectrogram*, samotný obrázek spektrogramu reprezentuje člen *image* a zvukový soubor člen *soundfile*. Pro zvukový soubor a obrázek spektrogramu jsou důležité metody *chooseSoundfile()*/*chooseImage()*, které nabídnou uživateli okno pro výběr vstupního souboru a zavolají *loadSoundfile()*/*loadImage()*, které daný soubor načtou a zkontrolují jeho platnost. Na platný soubor je pak zavoláno *updateSoundfile()*/*updateImage()*, které načtou informace o daném souboru (v případě zvuku jeho délku, vzorkovací frekvenci atd., v případě obrázku jeho velikost) a zobrazí ho v hlavním okně. Zda uživatel zvolil platný zvukový soubor pro analýzu nebo obrázek spektrogramu pro syntézu je možno zkontrolovat metodami *soundfileOk()* a *imageOk()*. Obnovy výchozího stavu lze dosáhnout zavoláním *resetSoundfile()* a *resetImage()*.

Protože tvorba spektrogramu a jeho syntéza jsou dlouho trvající operace, jsou pro zachování použitelnosti rozhraní během jejich běhu a poskytnutí možnosti je přerušit prováděny ve zvláštním vlákně. Toto je řešeno použitím frameworku QtConcurrent, konkrétně instancemi QFuture a QFutureWatcher, které řeší nutnou synchronizaci. Kód, který provádí samotný výpočet, průběžně posílá zprávy o stavu výpočtu pomocí signálů *progress()* a *status()* a zároveň kontroluje, zda nebyl signálem *cancel()* výpočet přerušen.

Třída Soundfile se v programu používá pro vysokoúrovňovou manipulaci se zvukovými soubory. Po načtení souboru metodou *load()* umí především přečíst celý jeden kanál pro účel jeho analýzy. Podrobnosti o načteném souboru jsou dostupné skrz metodu *data()*.

Tato vrací instanci třídy *SoundfileData*, která představuje společné abstraktní rozhraní nad knihovnami, které podporují různé zvukové formáty. Dvě poskytované implemenetace jsou *MP3Data*, která využívá knihovnu MAD pro čtení mp3 souborů a *SndfileData*, která využívá libsndfile pro podporu mnoha dalších formátů. Přes rozhraní *SoundfileData* je možné přistupovat k informacím o délce zvuku (metoda *length()*), jeho vzorkovací frekvenci (metoda *samplerate()*) atd. nezávisle na formátu. Další implementací tohoto rozhraní by šla snadno přidat podpora dalších zvukových formátů. O zápis zvuku vytvořeného zpětnou syntézou se stará statická metoda *Soundfile::writeSound()*, která využívá knihovnu libsndfile.

Převzorkování signálů je implementováno pomocí knihovny SRC. Její rozhraní je obaleno funkcí *resample()*. Knihovna bohužel neumožňuje převzorkování v poměru menším než 1/256 nebo větším než 256, které může být potřebné zejména v případě šumové syntézy. Taková převzorkování jsou pak prováděna ve více bězích, což tuto operaci značně zpomaluje.

Pro rychlou implementaci v programu hojně využívané diskrétní Fourierovy transformace a její inverze je použita knihovna FFTW3. Její komplikované rozhraní pro jazyk C je v tomto programu obaleno funkcemi *padded_FFT()* a *padded_IFFT()*. Tyto funkce zároveň zajistí, že velikost vstupních vektorů bude rozšířena na čísla ve tvaru $2^x \cdot 3^y \cdot 5^z$. Knihovna FFTW3 umí sice transformace obecně dlouhých vektorů, ale pro velikosti skládající se z malých prvočísel je transformace výrazně rychlejší.

# Kapitola 6

# Závěr

## 6.1 Výsledky

V této práci byl vytvořen program se snadno použitelným uživatelským rozhraním, který umožňuje i laikovi hlouběji nahlédnout do podstaty zvuku a poskytuje netradiční způsob manipulace s ním.

Program umožňuje pohodlně tvořit spektrogramy s podrobnými možnostmi nastavení jejich parametrů. Funkce zpětné syntézy pak nabízí možnost v této názorné obrázkové reprezentaci provádět úpravy a dosáhnout zvukových efektů, které nejsou tradičními způsoby dosažitelné.

Kvalita syntézy bohužel nedosahuje úrovně, kdy by nebylo možné snadno poznat, že jde o umělou rekonstrukci. Šumová syntéza, která dává v mnoha případech lepší výsledky než sinusová, poněkud trpí svými časovými nároky. Je nicméně možné z rekonstruovaných zvuků například bez problémů porozumět řeči, dokonce i tehdy, byl-li její spektrogram nakreslen zcela ručně dle předlohy.

Ve srovnání s jediným dalším volně dostupným řešením, tedy nedokončeným programem ARSS, nabízí v rámci této práce vytvořený program navíc příjemné uživatelské rozhraní, pečlivější základ v teorii a dobře čitelný a zdokumentovaný multiplatformní kód.

## 6.2 Možná budoucí rozšíření

V rámci další práce na programu se nabízí implementovat některé další vlastnosti, které by zlepšily jeho použitelnost, ale přesahují rámec této práce či na ně nezbyl čas.

Pro lepší prezentovatelnost vytvořených spektrogramů by se program mohl postarat o automatické vytvoření popisků a měřítek os, tedy zakreslit kóty času, frekvencí a měřítko intenzit. Měly by být zakresleny tak, aby nebránily zpětné syntéze, tedy aby šly od spektrogramu snadno automaticky odříznout.

Program by mohl také poskytovat konzolové rozhraní, umožňující snadné dávkové provádění jeho funkcí. Současné grafické rozhraní je v kódu programu důsledně odděleno od výpočetní části, proto by vytvoření konzolového rozhraní byl poměrně přímočarý úkol.

Zobrazení intenzit ve vytvořených spektrogramech lze ovlivnit volbou palety, ale pro pokročilejší analýzu signálů by bylo vhodné nabídnout podrobnější možnosti nastavení osy intenzit, například možnost restrikce zobrazovaných intenzit na konkrétní interval hlasitostí.

V programu by mohly být implementovány další metody tvorby a syntézy spektrogramů, případně obdoby využívající například obecnějších vlnkových transformací místo Fourierovy transformace. Pro úpravy zvuků pomocí spektrogramů lze navrhnout obdobu šumové syntézy s použitím původního zvuku ve frekvenční oblasti jako „nosiče“ místo šumu. Tím by se odstranil problém zašumění rekonstrukce a umožnily úpravy téměř bez ztráty kvality. Pro rekonstrukci pouze ze spektrogramu by ale taková metoda použitelná nebyla.

# Literatura


[1] Collias N. (1963): A Spectrographic Analysis of the Vocal Repertoire of the African Village Weaverbird, University of California, Los Angeles
http://elibrary.unm.edu/sora/Condor/files/issues/v065n06/p0517-p0527.pdf

[2] Michelsson K., Christensson K., Rothgänger H., Winberg J. (1996): Crying in separated and non-separated newborns: sound spectrographic analysis, University of Helsinki, Finland; Department of Women and Child Health, Karolinska Institute, Stockholm, Sweden; Humboldt University, Berlin, Germany

[3] Ertmer D., Maki E. (2000): A Comparison of Speech Training Methods with Deaf Adolescents: Spectrographic versus Noninstrumental Instruction, Journal of Speech, Language, and Hearing Research, Vol. 43 (Dec 2000)

[4] Kingsbury B., Morgan N., Greenberg S. (1998): Robust speech recognition using the modulation spectrogram, Speech Communication Journal, vol. 25 (Aug 1998), pp. 117–132

[5] Dutoit T. (1997): An introduction to text-to-speech synthesis, Springer, ISBN: 9780792344988

[6] Carmell T., Hosom J.-P., Cole R. (1999): A computer-based course in spectrogram reading, In Proceedings of the ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education.
http://cslu.cse.ogi.edu/toolkit/pubs/ps/carmell_MATISSE_99.ps

[7] Bello J. P., Monti G., Sandler M. (2000): Techniques for automatic music transcription, Proceedings of the 1st Annual International Symposium on Music Information Retrieval
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.9745&rep=rep1&type=pdf

[8] Truxal J. G. (1990): The Age of Electronic Messages, 113–117. MIT press, ISBN: 9780262200745

[9] Nyquist H. (1928): Certain topics in telegraph transmission theory, In Proceedings of the IEEE, Vol. 90, No. 2. (2002), pp. 280–305.

[10] Robinson D. W. et al. (1956), A re-determination of the equal-loudness relations for pure tones, British Journal of Applied Physics **7**
http://www.iop.org/EJ/abstract/0508-3443/7/5/302

[11] Ezzat T., Bouvrie J., Poggio T. (2006): Max-Gabor Analysis and Synthesis of Spectrograms, ICSLP 2006, Pittsburgh
http://web.mit.edu/jvb/www/papers/maxgabor_icslp06.pdf

[12] Arai T., Yasu K., Goto T. (2006): Digital pattern playback: Converting spectrograms to sound for educational purposes, Acoustical Science and Technology, Vol. 27, No. 6, pp. 393–395.
http://www.jstage.jst.go.jp/article/ast/27/6/27_393/_article

[13] Rouzic M. (2008): Program ARSS a jeho webové stránky
http://arss.sourceforge.net

[14] M. Slaney (1995): Pattern playback from 1950 to 1995, Proc. IEEE Int. Conf. Systems, Man and Cybernetics Conf., Vol. 4, pp. 3519–3524

[15] Smith S. W. (1997): The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing, 1997, ISBN: 0-9660176-3-3

# Příloha A

# Obsah CD

Přiložené CD obsahuje následující adresáře:

- `dokumentace` – obsahuje Doxygen dokumentaci (v angličtině) v pdf verzi a html verzi (`dokumentace/html/index.html`)
- `priklady` – obsahuje několik ukázkových zvukových souborů a spektrogramů
- `spustitelne` – obsahuje program ve verzi pro Windows
- `text` – obsahuje tuto práci ve formátu pdf a obrázky
- `zdrojovy_kod` – obsahuje zdrojové kódy k programu