

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Scrolling Song Book: Zpěvník s automatickým posunem textu

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Hana KALIVODOVÁ**
Osobní číslo: **A18B0228P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Scrolling Song Book – zpěvník s automatickým posunem textu**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se rámcově s problematikou rozpoznávání tónů a akordů v digitalizovaném zvukovém záznamu. Vyberte vhodnou knihovnu pro zpracování digitalizovaného zvuku, kterou bude možné snadno integrovat do vyvíjené aplikace.
2. Navrhněte a implementujte aplikaci, která bude průběžně analyzovat zvuk a podle rozpoznávaných akordů v záznamu posouvat souhlasně text/notový/tabulovaný zápis písně na monitoru.
3. Otestujte (např. testy na nepoučených uživatelích) použitelnost vyvinuté aplikace, zhodnoťte úspěšnost rozpoznávání tónů/akordů.
4. Vše důkladně popište v průvodním dokumentu práce.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Kamil Ekštein, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **4. října 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 14. října 2021

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů. V bakalářské práci jsou použity názvy programových produktů, firem apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

V Plzni dne 5. května 2022

Hana Kalivodová

Poděkování

Chtěla bych poděkovat Ing. Kamilu Ekšteinovi, Ph.D. za vstřícnost a ochotu při vedení bakalářské práce a věcné připomínky při jejím vypracování.

Abstract

The aim of this work is to develop an application that will display song lyrics with chords using a clear simple user interface. The application will use spectral analysis of the incoming sound to evaluate the chords in the played music. Based on the evaluated chords, the application will scroll the song lyrics synchronized with the chord accompaniment.

Abstrakt

Cílem práce je vývoj aplikace, která bude pomocí přehledného jednoduchého uživatelského rozhraní zobrazovat texty písní s akordy. Aplikace by měla s využitím spektrální analýzy umět vyhodnocovat akordy zaznamenaného zvuku. Na základě vyhodnocených akordů by měla aplikace posouvat text ve shodě s akordickým doprovodem.

Obsah

1	Úvod	9
2	Zpracování signálu	10
2.1	Podvzorkování signálu	10
2.1.1	Nyquist-Shannonův vzorkovací teorém	10
2.1.2	Podvzorkování v časové doméně	11
2.1.3	Podvzorkování ve frekvenční doméně	11
2.2	Výpočet frekvence	11
2.2.1	Časová doména	11
2.2.2	Frekvenční doména	12
2.3	Fourierova transformace	13
2.3.1	Matematika a Fourierova transformace	13
2.3.2	Diskrétní Fourierova transformace	14
2.3.3	Určení koeficientů a_k, b_k	14
2.4	Funkce okna (Windowing)	15
2.4.1	Hannova funkce okna	15
2.4.2	Hammingova funkce okna	15
2.4.3	Blackmanova funkce okna	16
2.5	Emfáze	16
2.6	Zero Crossing	16
2.7	Střední hodnota energií signálu	17
2.8	Lineární konvoluční filtr	17
2.9	Savitzky-Golayův vyhlazovací filtr	17
3	Vyhodnocování akordů	19
3.1	Zvuk	19
3.2	Oktáva, stupnice, tónina	20
3.3	Akord	21
3.4	Obraty akordu	21
3.5	Hledání vrcholů	22
3.6	Podmíněná pravděpodobnost	23
4	Návrh implementace	24
4.1	Účel a požadavky aplikace	24
4.2	Příprava akordů	25
4.3	Překrývání rámců	25

4.4	Velikost rámce	26
4.5	Počet zpracovávaných oken	26
4.6	Emfáze	26
4.7	Výběr rámců	27
4.8	Funkce okna (windowing)	27
4.9	Savitzky-Golayův filtr	28
4.10	Konvoluce	29
4.11	Průměr spekter	29
4.12	Váhování spektra logaritmem	30
4.13	Střední hodnota spektra	30
4.14	Určení akordu	32
4.15	Posun textu	33
5	Implementace	34
5.1	Využité technologie	34
5.2	Využité knihovny a jejich funkce	34
5.2.1	PyQt5	35
5.2.2	PyAudio	35
5.2.3	NumPy	35
5.2.4	math, statistics	36
5.2.5	SciPy	36
5.2.6	Matplotlib	36
5.3	Struktura projektu	37
5.4	Popis implementace	38
6	Testování aplikace	41
7	Správnost vyhodnocení akordů	44
8	Závěr	46
	Literatura	47
A	Seznam zkratk	50
B	Testovací scénář	51
B.1	Ověření funkčnosti aplikace	51
B.2	Hodnocení aplikace	52
C	Uživatelská dokumentace	53
D	Obsah přiloženého ZIP souboru	55

1 Úvod

Hrajete alespoň amatérsky na nějaký hudební nástroj a hrajete podle zpěvníku nebo z not? Pak se vám jistě stalo, že jste se dostali na konec stránky a potřebovali jste otočit na další stránku. V tom případě jste museli přerušit svou hru. Aby nebylo nutné přerušit hru, je potřeba osoby, která stránku včas otočí. Další možností je využití různých aplikací, které umožňují nastavit rychlost posunu textu. Pokud se však ve skladbě vyskytne například mezihra nebo repetice, která je většinou zapsaná akordy na jedné řádce, nepomůže ani to. Nastavení rychlosti zajišťuje pouze posun textu o řádku po nastaveném čase bez ohledu na obsah řádky.

Cílem této bakalářské práce je vytvoření aplikace s jednoduchým uživatelským prostředím, do které by uživatel nahrál text písně s akordy a která by rozpoznávala právě hraný akord a na základě této informace průběžně posouvala text. K vyhodnocení akordů je potřeba zaznamenávat a zpracovávat informaci o příchozím zvukovém signálu, upravovat jej do potřebné formy a získávat z něj potřebné akordy, kterými se řídí posun textu. Rozpoznávání akordů může být zajištěno spektrální analýzou nebo umělou neuronovou sítí, kterou lze pomocí trénovacích dat (data, u kterých známe správný výsledek) naučit tak, aby z příchozích úseků zaznamenávaného zvukového signálu vyhodnotila akord. V této bakalářské práci je k rozpoznávání akordů použita spektrální analýza.

Spektrální analýza je proces skládající se z několika kroků. Například zaznamenávaný zvukový signál je potřeba vzorkovat se zadanou frekvencí a rozdělovat do úseků stejné velikosti, tzv. rámců. Z rámců pomocí Fourierovy transformace lze získat odhad výkonové spektrální hustoty, který je možné vyhladit Savitzky-Golayovým filtrem. Následně je možné určit frekvence jednotlivých tónů a vyhodnotit hraný akord.

Jednotlivé kroky spektrální analýzy jsou popsány v teoretické části bakalářské práce. V návrhu implementace jsou uvedeny a vysvětleny konkrétní zvolené hodnoty a parametry v krocích spektrální analýzy. Popis implementace zahrnuje použité technologie a strukturu projektu. Výsledná aplikace je testována na nepoučených uživateli, kteří se řídí testovacím scénářem. Na závěr je uvedena statistika úspěšnosti vyhodnocovaných akordů.

2 Zpracování signálu

Zvukový signál, který slyšíme, se šíří vzduchem v podobě spojitého kmitání hmotného prostředí. Spojitý signál je potřeba převést do digitální podoby. Digitální signál je diskrétní, tj. je nespojitý v čase i amplitudě, a je popsateľný celými čísly, tzv. vzorky. Převod spojitého signálu na diskrétní dnes automaticky provádí A/D převodník, elektronická součástka v počítači. A/D převodníky spojitý signál nejprve vzorkují a následně kvantují. Při vzorkování signálu je signál rozdělen v ose x na úseky stejné velikosti, na hranách úseků získáme jednotlivé vzorky signálu. Takto vzniklý vzorkovaný signál je kvantován. Při kvantizaci je signál rozdělen na hladiny v ose y . Hodnoty diskrétního signálu jsou určeny průniky hladin na ose y se vzorky na ose x . Na diskrétní digitální signál je možné použít spektrální analýzu. Spektrální analýza se skládá z matematických operací, jako je například Fourierova transformace. Některé matematické operace jsou popsány v této kapitole.

2.1 Podvzorkování signálu

Podvzorkování signálu je proces, při kterém se snižuje vzorkovací frekvence (počet vzorků za jednotku času) periodickým vyřazováním některých vzorků signálu. Vzorkovací frekvence je snižována o celočíselný faktor. Pokud podvzorkujeme signál $s[n]$ o celočíselný faktor Q , zbyde každý Q -tý vzorek signálu $s[n]$ (ostatní vzorky ignorujeme). Vzorkovací frekvence je pak $1/Q$ z původní vzorkovací frekvence signálu $s[n]$. Signál lze podvzorkovat v časové i frekvenční doméně [2].

2.1.1 Nyquist-Shannonův vzorkovací teorém

Podle Nyquist-Shannonova vzorkovacího teorému je přesná rekonstrukce spojitého frekvenčně omezeného signálu teoreticky možná, pokud je vzorkovací frekvence vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu, viz 2.1.

$$f_{vz} \geq 2f_{max} , \quad (2.1)$$

kde f_{vz} je vzorkovací frekvence a f_{max} je frekvence nejvyšší harmonické složky vzorkovaného signálu.

Při podvzorkování signálu na nižší vzorkovací frekvenci může dojít k porušení Nyquistova-Shannonova vzorkovacího teorému, což může vést k nežádoucímu jevu překrývání spekter, tzv. *aliasingu*. Tento jev se může projevit

v časové i frekvenční oblasti [8]. Abychom aliasingu předešli, použijeme filtr s dolní propustí. Tento filtr propustí pouze vzorky s frekvencí stejnou nebo nižší, než je nová snížená vzorkovací frekvence. Díky použití filtru s dolní propustí zachováme Nyquistův-Shannonův vzorkovací teorém [2].

2.1.2 Podvzorkování v časové doméně

Je-li šířka pásma signálu $s[n]$ menší nebo rovna $1/Q$ původní vzorkovací frekvence signálu $s[n]$, pak podvzorkování dosáhneme výběrem vzorků s indexy $k \cdot Q$, kde $k \in N$. Signál lze rovněž vynásobit posloupností $p[n]$ s periodou Q a následně ze signálu odstranit vzorky s nulovou hodnotou [2].

2.1.3 Podvzorkování ve frekvenční doméně

Součin signálu $s[n]$ a posloupnosti $p[n]$ s periodou Q v časové doméně ovlivňuje konvoluci mezi spektry $S[k]$ a $P[k]$. Spektrum $P[k]$ je tvořeno Q impulzy na intervalech mezi vzorky $k = 0$ a celočíselných násobcích $k = \pm N/Q$, kde N je celkový počet vzorků. Vzniklé části spektra jsou dále zmenšeny o faktor Q . Amplituda spekter je určena podílem N/Q . Důsledkem konvoluce vzniká Q spektrálních kopií $S[k]$ na intervalech mezi vzorky $k = 0$ a $k = \pm N/Q$. V poslední fázi procesu podvzorkování zmenšíme šířku pásma signálu o faktor Q , ponecháme pouze N/Q vzorků na intervalu v okolí vzorku $k = 0$ [2].

2.2 Výpočet frekvence

Fourierova transformace převádí diskrétní signál z časové domény do frekvenční [21].

2.2.1 Časová doména

Vzorkovací frekvence

Vzorkovací frekvence F_s vyjadřuje počet vzorků signálu získaných za sekundu. Fyzikální rozměr frekvence je s^{-1} , jednotkou je 1 Hz [21]. Podle Nyquist-Shannonova vzorkovacího teorému potřebujeme signál vzorkovat minimálně dvojnásobkem nejvyšší harmonické složky vzorkovaného signálu, viz 2.1, abychom mohli signál přesně rekonstruovat.

Velikost rámce

Velikost rámce může být dána počtem vzorků N , které jsou použity jako vstup pro Fourierovu transformaci, nebo dobou trvání T , která je dána vztahem 2.2 [21]:

$$T = \frac{N}{F_s}. \quad (2.2)$$

2.2.2 Frekvenční doména

Šířka pásma

Šířka pásma F_{max} je určena maximální frekvencí, kterou lze zachytit v daném odhadu výkonové spektrální hustoty. Z Nyquistova kritéria (viz vzorec 2.1) vyplývá, že šířka pásma je polovinou vzorkovací frekvence, viz 2.3 [21]:

$$F_{max} = \frac{F_s}{2}. \quad (2.3)$$

Velikost odhadu výkonové spektrální hustoty

Výstupem Fourierovy transformace je odhad výkonové spektrální hustoty (PSD). Velikost odhadu PSD d_{psd} je analogií velikosti rámce v časové doméně, tedy je dána počtem pásem. Velikost odhadu PSD je rovna polovině velikosti rámce, viz 2.4 [21]:

$$d_{psd} = \frac{N}{2}. \quad (2.4)$$

Šířka frekvenčního pásma

Šířka frekvenčního pásma Δf je dána rozdílem hodnot dvou sousedních pásem odhadu PSD. Šířka frekvenčního pásma je rámci odhadu PSD konstantní a je rovna podílu celkové šířky pásma a velikosti odhadu PSD, viz 2.5 [21]:

$$\Delta f = \frac{F_{max}}{d_{psd}}. \quad (2.5)$$

Vztahy 2.2, 2.4, 2.5 lze vzájemně propojit, viz 2.6 [21]:

$$\Delta f = \frac{1}{T} = \frac{F_{max}}{d_{psd}} = \frac{F_s}{N}. \quad (2.6)$$

Výslednou frekvenci daného pásma získáme součinem šířky frekvenčního pásma a indexu daného pásma v odhadu PSD, viz 2.7:

$$f = i\Delta f, \quad i = 0, \dots, d_{psd} - 1. \quad (2.7)$$

2.3 Fourierova transformace

Základním nástrojem pro zpracování signálu je Fourierova transformace, která rozkládá signál na komponenty, které lze vyjádřit funkcemi sinus a cosinus. Nahrávaný zvukový signál je signál složený, protože zachycuje více tónů s různými frekvencemi, kterými je nahrávaný akord tvořený. Abychom mohli hraný akord určit, je potřeba nejprve získat ze signálu frekvence jednotlivých tónů, což umožňuje právě Fourierova transformace [5].

2.3.1 Matematika a Fourierova transformace

Složený signál je možné zobrazit v časové doméně. Představíme-li si signál jako graf, na ose y bude amplituda signálu a osa x bude představovat čas. Fourierova transformace převádí diskrétní signál z časové domény do frekvenční. Tedy na ose x budou frekvence, osa y bude představovat energii.

Libovolný spojitý signál v časové doméně lze jednoznačně reprezentovat nekonečným součtem sinusoid. Ke každému signálu, který je generován funkcí $x(t)$, lze nalézt funkci $f(t)$, viz vztah 2.8, která je součtem nekonečné řady sinusoid. Tyto sinusoidy budou přibližně popisovat signál [5].

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos 2\pi kt + b_k \sin 2\pi kt), \quad (2.8)$$

kde a_0 je první člen nekonečné řady, a_k , b_k jsou koeficienty určující tvar signálu.

K učení koeficientů a_k , b_k využijeme Fourierovu transformaci, jejímž výsledkem je řada koeficientů $X(F)$. Fourierovy koeficienty získáme vztahem 2.9 [5].

$$X(F) = \int_{-\infty}^{\infty} x(t)e^{-2\pi i F t} dt, \quad (2.9)$$

kde $X(F)$ jsou Fourierovy koeficienty, $x(t)$ funkce generující signál, i imaginární jednotka. Komplexní číslo z má tvar 2.10 a je složeno z reálné části a a imaginární části bi . Výsledkem integrálu 2.9 jsou komplexní čísla, jejichž složky a , b odpovídají koeficientům a_k , b_k , uvedeným ve vztahu 2.8 [5].

$$z = a + bi \quad (2.10)$$

2.3.2 Diskrétní Fourierova transformace

Spojité signál lze vzorkováním v čase a kvantováním převést na diskrétní. Diskrétní signál lze popsat posloupností čísel, viz 2.11:

$$\{x_i\}_{i=0}^{N-1}, \quad (2.11)$$

ve které číslo x_i představuje dílčí vzorek, N je počet vzorků signálu.

Libovolný diskrétní (vzorkovaný) signál v časové doméně je možné jednoznačně reprezentovat konečnou řadou sinusoid. Diskrétní Fourierova transformace umožňuje získat z posloupnosti vzorků, viz 2.11, Fourierovy koeficienty, viz vztah 2.12, což jsou komplexní čísla odpovídající koeficientům Fourierovy řady [5].

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i(kn)}{N}}, \quad (2.12)$$

kde X_k je k -tý koeficient, x_n n -tý vzorek posloupnosti (2.11).

Modifikací diskrétní Fourierovy transformace je rychlá Fourierova transformace, která se v praxi využívá častěji z důvodu nižší výpočetní složitosti. Výpočetní složitost diskrétní Fourierovy transformace je $O(N^2)$, u rychlé Fourierovy transformace je to $O(N \log_2 N)$ [5].

2.3.3 Určení koeficientů a_k , b_k

Pro určení koeficientů a_k , b_k využijeme Eulerovu formuli, viz 2.13 a vztah pro diskrétní Fourierovu transformaci 2.12 [5]:

$$e^{i\varphi} = \cos \varphi + i \sin \varphi. \quad (2.13)$$

Ve vztahu 2.12 nahradíme $e^{-\frac{2\pi i(kn)}{N}}$ funkcemi \sin a \cos podle Eulerovy formule. Dále využijeme, že $\cos(-\varphi) = \cos(\varphi)$ a $\sin(-\varphi) = -\sin(\varphi)$. Nakonec upravíme rovnici do součtového tvaru a vznikne vztah 2.14 [5].

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi(kn)}{N}\right) - i \sum_{n=0}^{N-1} x_n \sin\left(\frac{2\pi(kn)}{N}\right) \quad (2.14)$$

Výsledný vztah (2.14) má tvar komplexního čísla, viz 2.10, jehož složky a , b odpovídají hledaným koeficientům a_k , b_k [5].

Z reálné a imaginární části komplexních čísel, které vzniknou Fourierovou transformací, lze vypočítat hodnoty odhadu výkonové spektrální hustoty jako velikost komplexního čísla, viz 2.15.

$$|z| = \sqrt{a^2 + b^2}, \quad (2.15)$$

kde z je komplexní číslo, a jeho reálná složka, b jeho imaginární složka.

2.4 Funkce okna (Windowing)

Periodický signál lze zpracovat například Fourierovou transformací. Je-li potřeba zpracovávat neperiodický signál, rozdělíme ho na rámce (frame) stejné délky, stejného počtu vzorků. Předpokládejme, že vně rámce je signál periodický. Při dělení signálu na rámce vznikají na hranách rámců nespojitosti, které ve spektru vytvářejí nežádoucí vysoké frekvence, tzv. prosakování. Abychom prosakování omezili, vynásobíme každý rámeček signálu funkcí okna, tedy provedeme tzv. windowing [7].

Signál v daném rámci bude ohraničen zvolenou funkcí okna, která zajistí nulové krajní hodnoty rámce a jeho vnějšího okolí a utlumí signál na vnitřních krajích rámce. Mezi známé a často využívané funkce okna patří např. Hannova, Hammingova a Blackmanova funkce. Tyto funkce se od sebe liší prakticky jen mírou zúžení u krajů rámce. Obecným cílem funkcí oken ve frekvenční doméně je vysoký útlum postranních laloků (na krajích rámce) a malá šířka hlavního laloku (uprostřed rámce) [20]. V následujících vztazích 2.16, 2.17, 2.18 předpokládáme, že délka rámce je rovna počtu vzorků v rámci a je značena písmenem N .

2.4.1 Hannova funkce okna

Hannova funkce okna, známá také jako kosinový kužel, je dána vztahem 2.16.

$$w(n) = 0,5 - 0,5 \cos\left(\frac{2\pi n}{N}\right), \text{ pro } n = 0, 1, 2, \dots, N-1. \quad (2.16)$$

Počáteční a koncová hodnota Hannovy funkce okna je 0, postupně roste až k hodnotě 1 a následně plynule klesá zpět k hodnotě 0. Maximální hodnoty 1 dosáhne v polovině rámce [20].

2.4.2 Hammingova funkce okna

Hammingova funkce okna je zúžením Hannovy funkce okna a je dána vztahem 2.17.

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N}\right), \text{ pro } n = 0, 1, 2, \dots, N-1. \quad (2.17)$$

Průběh Hammingovy funkce okna je velmi podobný Hannově funkci, liší se pouze počáteční a koncovou hodnotou. Hammingova funkce okna začíná a končí na hodnotě 0,08. V polovině rámce dosáhne maximální hodnoty 1 [20].

2.4.3 Blackmanova funkce okna

Blackmanova funkce okna využívá funkce kosinus a je dána vztahem 2.18.

$$w(n) = 0,42 - 0,5 \cos\left(\frac{2\pi n}{N}\right) + 0,08 \cos\left(\frac{4\pi n}{N}\right), \text{ pro } n = 0, 1, 2, \dots, N-1. \quad (2.18)$$

Ve srovnání s výše uvedenými funkcemi okna má Blackmanova funkce větší šířku hlavního laloku (nežádoucí), ale větší útlum postranních laloků (žádoucí) [20].

2.5 Emfáze

Emfáze se skládá z *pre-emfáze* na straně odesílatele a *de-emfáze* na straně příjemce. Pre-emfáze je způsob, jak zdůraznit vyšší frekvence signálu. Často se implementuje jako filtr s konečnou impulzivní odezvou (FIR filtr) v časové doméně s jedním voleným parametrem [1]. De-emfáze je opak pre-emfáze, pomocí níž lze zdůraznit nižší frekvence signálu.

2.6 Zero Crossing

Pojem *zero crossing* lze definovat jako počet průniků signálu s osou x ($y = 0$). Výpočet této hodnoty získáme z definice 2.19

$$ZC = \sum_{i=0}^{N-2} f_{ZC}(x_i, x_{i+1}), \quad (2.19)$$

kde ZC označuje hodnotu příznaku zero crossing, N je počet vzorků signálu. Funkce f_{ZC} nabývá hodnoty 1, pokud došlo k přechodu signálu přes nulu, v opačném případě nabývá hodnoty 0. Funkci f_{ZC} lze definovat různými způsoby, dále jsou uvedeny jen některé z nich.

Nejjednodušší metodou je porovnávání dvou po sobě jdoucích vzorků signálu. Pokud je jeden vzorek kladný a druhý záporný, jedná se o zero crossing. Funkci f_{ZC} lze popsat vztahem 2.20 [23].

$$f_{ZC} = \begin{cases} 1, & x_i > 0 \text{ a zároveň } x_{i+1} < 0 \\ & \text{nebo } x_i < 0 \text{ a zároveň } x_{i+1} > 0 \\ 0, & \text{jinak.} \end{cases} \quad (2.20)$$

Dalším způsobem je součin dvou po sobě jdoucích prvků signálu. Pokud dojde k přechodu přes nulu, je výsledný součin záporný a funkce f_{ZC} vrací

hodnotu 1, v ostatních případech hodnotu 0, viz 2.21 [23].

$$f_{zC} = \begin{cases} 1, & x_i \cdot x_{i+1} < 0 \\ 0, & \text{jinak.} \end{cases} \quad (2.21)$$

Tento způsob lze doplnit o funkci signum, která umožňuje porovnávat jen celá čísla místo reálných, viz 2.22 [23].

$$f_{zC} = \begin{cases} 1, & \text{sgn}(-x_i \cdot x_{i+1}) > 0 \\ 0, & \text{jinak.} \end{cases} \quad (2.22)$$

2.7 Střední hodnota energií signálu

Střední hodnotu energií signálu $E(X)$ lze určit jako podíl součtu druhých mocnin hodnot v každém vzorku signálu a počtu vzorků, viz vzorec 2.23

$$E(X) = \frac{\sum_{i=0}^N x_i^2}{N}, \quad (2.23)$$

kde N je počet všech vzorků signálu, x_i hodnota daného vzorku.

2.8 Lineární konvoluční filtr

Lineární konvoluční filtr lze aplikovat na signál (funkci) k jeho tzv. vyhlazení. Vyhlazením signálu lze odstranit drobné kmity. K vyhlazení můžeme využít průměr hodnot. Hodnotu x vyjádříme jako průměr Q sousedních hodnot hodnoty x a hodnoty x , kde Q je námi zvolená konstanta udávající velikost okolí hodnoty x . Následně na místo hodnoty x přiřadíme získanou průměrnou hodnotu.

2.9 Savitzky-Golayův vyhlazovací filtr

V analytické chemii, spektroskopii nebo zpracování signálu je pro vyhlazení signálu často používaným nástrojem Savitzky-Golayův filtr [6]. Pomocí Savitzky-Golayova filtru lze vyhladit spektrum od drobných kmitů, které se v něm nacházejí.

Savitzky-Golayův filtr je dán šířkou filtrovacího okna w a řádem polynomu n , kterým je signál prokládán. Tvar křivky je vždy vypočten pro dané filtrovací okno, které je posouváno podél signálu [6]. Šířka filtrovacího okna je zadána jako celé liché číslo, liché z důvodu možné symetrie, neboť filtr vytváří odhad proložením polynomu středem každého filtrovacího okna. Šířka filtrovacího okna by měla být větší nebo rovna řádu polynomu zvýšenému

o 1, tedy $w \geq (n + 1)$. Pokud je $w > (n + 1)$ dochází k vyhlazování signálu, v případě rovnosti nedojde k žádné změně původního signálu [6].

3 Vyhodnocování akordů

Tato kapitola seznamuje čtenáře v první části s některými základními pojmy hudební teorie, které jsou dále použité v bakalářské práci. Jedná se především o pojmy tón (hudební zvuk), stupnice, oktáva, tónina a akord. U tónu jsou uvedeny i některé jeho vlastnosti. Akord je zde popsán včetně svých tzv. obrátů. V druhé části jsou popsány metody pro hledání vrcholů odhadu výkonové spektrální hustoty a pojmy použité k vyhodnocování akordů.

3.1 Zvuk

Zvuk je mechanické vlnění hmotného prostředí. Zdrojem zvuku tedy může být libovolné chvějící se těleso, například hudební nástroje, hlasivky, součástky strojů, motory a další. U hudebních nástrojů jsou zdrojem zvuku většinou struny, membrány a píšťaly. Rozlišujeme dva druhy zvuků. Zvuky nehudební (neperiodické) nazýváme hluky nebo šумы. Naopak zvuky hudební (periodické) nazýváme tóny. Pojmem jednoduchý tón je označován nejjednodušší hudební zvuk, který lze reprezentovat jedinou sinusoidou [10].

Výška tónu

Výška jednoduchého tónu je přímo závislá na jeho frekvenci, čím vyšší frekvence, tím vyšší výška. Člověk je schopný slyšet zvukové vlnění s frekvencí přibližně od 16 Hz do 16 kHz. Slyšitelnost konkrétních frekvencí je velmi individuální, zaleží například na zdravotním stavu a věku jedince. Vzhledem k obtížnosti přesného určení absolutní výšky tónu sluchem je výška dána poměrem frekvencí daného tónu a tónu základního. Jako základní tón byl stanoven tón s frekvencí 440 Hz, který je označován jako komorní a [11].

Barva tónu

Díky barvě tónu jsme schopni od sebe odlišit tóny stejné výšky pocházející z různých zdrojů, např. hudební nástroje, lidské hlasy. Podle Fourierovy teorie je možné zvuk rozložit na tzv. harmonické složky. Libovolnou periodickou funkci s periodou T lze zapsat jako lineární kombinaci harmonických funkcí s periodami T/n , kde n je přirozené číslo. Koefficienty lineární kombinace udávají barvu tónu. Vyšší harmonické frekvence jsou tvořeny přirozenými násobky základní frekvence, která odpovídá frekvenci $n = 1$. Amplitudy

vyšších harmonických frekvencí jsou tedy obvykle menší než amplituda základní frekvence. Díky vyšším harmonickým frekvencím odlišujeme jednotlivé zdroje zvuku. Barvu tónu lze u hudebních nástrojů ovlivnit zejména tvarem a materiálem [12].

3.2 Oktáva, stupnice, tónina

Oktáva je tvořena 12 půltóny. Půltón v evropské hudbě označuje nejmenší interval mezi tóny. Pokud začneme tónem C , bude oktáva tvořena tóny C , Cis , D , Dis , E , F , Fis , G , Gis , A , B , H . V hudbě je určováno 9 různých oktáv, viz tab. 3.1, které zahrnují rozsah frekvencí přibližně od 16 Hz do 8 kHz. Každý hudební nástroj má omezený rozsah tónů, které na něj lze zahrát. Největší rozsah z hudebních nástrojů mají varhany, jejichž nejhlubší tón (16,5 Hz) má frekvenci na prahu slyšitelnosti a u nejvyššího tónu (8370 Hz) už nejsme schopni rozlišit barvu tónu. Základní tón, komorní a o frekvenci 440 Hz, se nachází v tzv. jednočárkované oktávě. Frekvence tónů lze vypočítat jako násobky frekvencí tónů z jednočárkované oktávy, $a_2 = 2 \cdot 440 \text{ Hz} = 880 \text{ Hz}$, $a = 0,5 \cdot 440 \text{ Hz} = 220 \text{ Hz}$ [13]. Jednotlivé frekvence tónů u temperovaného ladění lze vypočítat z komorního $a_1 = 440 \text{ Hz}$. Na základě poznatků, že jedna oktáva je tvořena 12 půltóny a frekvence tónu o oktávu vyšší je dvojnásobek frekvence původního tónu, lze určit vztah mezi jednotlivými půltóny. Podíl frekvencí sousedních půltónů odpovídá hodnotě $p = \sqrt[12]{2} \approx 1,06$. Z tónu, jehož frekvence je známa, lze určit frekvenci libovolného tónu. Pro příklad jsou uvedeny výpočty frekvencí tónů g_1 a c_2 z frekvence tónu a_1 , viz 3.1 [15].

$$g_1 = \frac{a_1}{p^2}, \quad c_2 = a_1 \cdot p^3 \quad (3.1)$$

Pojmem *stupnice* je označovaná řada sedmi po sobě jdoucích tónů. Tóny jedné stupnice tvoří tóninu, např. stupnici C dur odpovídá tónina C dur. Oktáva je tvořena 12 tóny, existuje tedy 12 základních stupnic a tónin. Stupnice dělíme podle toho, jestli mají tvrdý nebo měkký *tónorod*. Tónorod určujeme podle postavení tercie ve stupnici. Velká tercie, vzdálenost mezi tóny je 4 půltóny, udává tvrdý tónorod. Malá tercie, která odpovídá vzdálenosti 3 půltónům, udává měkký tónorod. Tvrdý tónorod mají všechny stupnice dur a stupnice durového charakteru, naopak měkký tónorod mají stupnice moll a stupnice mollového charakteru [16].

Oktáva	Označení	Frekvence prvního tónu v Hz
subkontra	C_2	16,5
kontra	C_1	33
velká	C	66
malá	c	132
jednočárkovaná	c_1	264
dvoučárkovaná	c_2	528
tříčárkovaná	c_3	1056
čtyřčárkovaná	c_4	2112
pětičárkovaná	c_5	4224

Tabulka 3.1: Přehled určovaných oktáv v hudbě, jejich označení a frekvence prvního tónu [13].

3.3 Akord

Akordy vycházejí ze stupnic, tedy máme 12 základních akordů. Akord je tvořen konkrétními tóny stupnice. Základní tvar akordu, zvaný *kvintakord* tvoří první, třetí a pátý tón stupnice, které dohromady tvoří souzvuk. Podle tónorodu lze stejně jako stupnice dělit akordy na durové a mollové. Základní akordy jsou tedy děleny na 12 durových a 12 mollových akordů. Akordy lze různě modifikovat. Nejčastější modifikací jsou tzv. septakordy, které jsou tvořeny čtyřmi tóny a značené číslicí 7. Přidáme-li k základní trojici kvintakordu (první, třetí a pátý tón stupnice) ještě půltón mezi šestým a sedmým tónem stupnice, vytvoříme septakord. Další modifikace akordů jsou označovány dim, maj, sus4 a další.

3.4 Obraty akordu

Základní tvar akordu, tedy první, třetí a pátý tón stupnice, kdy první tón je nejhlubší tón akordu, se nazývá *kvintakord*. Kvintakord C dur je tvořen tóny C , E , G , přičemž tón C je nejhlubší. Obrat akordu získáme přesunutím nejhlubšího tónu akordu o oktávu výše. Vzniklý akord je nazýván *sextakord* nebo 1. obrat. V případě akordu C dur je složen z tónů E , G , C , kde tón E je nejhlubší. 2. obrat, tzv. *kvartsextakord*, vznikne opět přesunutím nejhlubšího tónu o oktávu výše. U akordu C dur je tedy složen z nejhlubšího tónu G , následují tóny C a E [9].

Akord je i v obrazech tvořen stále stejnými tóny. Obraty akordu však mohou znít odlišně, což je způsobeno změnou nejhlubšího tónu. Obraty akordu mohou pozměnit vyznění písně, ale jsou používány rovněž z důvodu snazšího

střídání akordů. Často díky obrátům stačí změnit jen jeden tón a změní se celý akord [9].

3.5 Hledání vrcholů

Vrcholem je v této práci míněno lokální maximum odhadu výkonové spektrální hustoty, dále jen spektra. Jedná se tedy o hledání všech lokálních maxim spektra. Vrcholy udávají pásma s nejvyšší energií, pro které předpokládáme, že jsou nejvýznamnější pro určení akordu. K nalezení lokálních maxim je možné využít různé metody.

Nejjednodušší metodou je porovnávání hodnot sousedních pásem spektra. Metoda postupně porovnává hodnoty pásem. Pokud jsou hodnoty pásem na indexech spektra $i - 1$ a $i + 1$ nižší než hodnota pásma na indexu i , je pásmo na indexu i označeno jako lokální maximum. Pokud se ve spektru vyskytuje pásmo, jehož hodnota je nepatrně vyšší oproti sousedním pásmům, je touto metodou označen za vrchol, přestože hledaným vrcholem není. Pravděpodobnost takovýchto případů lze snížit použitím vyhlazovacího filtru.

U spekter, kde je hodnota hledaných pásem výrazně vyšší než u ostatních pásem spektra, lze pro určení vrcholů využít střední hodnotu spektra. Metoda prochází spektrum a hledá pásmo s nejvyšší hodnotou nad střední hodnotou spektra. Pokud je některá hodnota menší než střední hodnota spektra, vyhodnotí uloženou nejvyšší hodnotu jako lokální maximum. Tento proces opakuje, dokud metoda neprojde všechna pásma spektra [4].

Pokud ze signálu nelze vrcholy jednoznačně určit (v okolí vrcholu se vyskytují pásma s podobnou hodnotou), lze určit vrcholy na základě klouzavého průměru spektra. Průměr je vypočten pro každé pásmo na základě okolních pásem. Jedná se tak vlastně o vyhlazení spektra lineárním konvolučním filtrem, viz kapitola 2.8. Pro následné určení vrcholů vyhlazeného spektra lze například použít jeho střední hodnotu [4].

K nalezeným vrcholům vypočteme jim odpovídající frekvence a přiřadíme jim tón. Z odhadu výkonové spektrální hustoty získáme po provedení dříve popsaných kroků seznam tónů. Na základě počtu shodných tónů v akordu a seznamu tónů, můžeme u každého akordu určit pravděpodobnost, že se jedná o právě hraný akord. Pravděpodobnosti jednotlivých akordů odpovídají podmíněné pravděpodobnosti, kdy podmíněným jevem je seznam tónů. Akord s nejvyšší pravděpodobností pak vyhodnotíme jako právě hraný akord.

3.6 Podmíněná pravděpodobnost

Podmíněnou pravděpodobnost lze využít v případě, kdy chceme znát pravděpodobnost daného jevu za určité podmínky. Obvykle se uvažuje, že je známa pouze částečná informace výsledku experimentu [24]. Podmíněnou pravděpodobnost jevu A za podmínky, že nastal jev B , lze spočítat podle následujícího vzorce 3.2:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{|A \cap B|}{|B|}, \quad (3.2)$$

kde $P(A \cap B)$ je pravděpodobnost průniku jevů, tj. oba jevy nastanou současně. $P(B)$ je pravděpodobnost jevu B , pro kterou předpokládáme, že $P(B) > 0$. Podmíněnou pravděpodobnost lze také vyjádřit jako podíl počtu společných prvků množin jevů A a B (velikost průniku obou množin) a počtu všech prvků množiny jevu B (velikost množiny jevu B).

4 Návrh implementace

Kapitola popisuje účel vytvářené aplikace a jednotlivé části spektrální analýzy a vyhodnocování akordů, v nichž parametry funkcí byly často určovány experimentálně. Veškeré obrázky uvedené v této kapitole jsou vytvořeny v programovacím jazyce Python pomocí modulu **pyplot** z knihovny **matplotlib**, viz 5.2.6. V celé kapitole je pojmem *spektrum* míněn odhad výkonové spektrální hustoty.

4.1 Účel a požadavky aplikace

Aplikace vytvářená v rámci bakalářské práce by měla umožňovat posun textu s akordy na základě nahrávaného zvukového signálu a vyhodnocování akordů.

Uživatelské rozhraní aplikace by mělo uživateli umožnit čtení textu s akordickým doprovodem ve formátu *ChordPro* ze souboru. Dále by mělo umožňovat nastavení velikosti, fontu a barvy písma. Pokud umožníme změnu barvy písma, bylo by vhodné umožnit také nastavení barvy pozadí. Pokud například uživatel bude hrát za tmy nebo šera, zvolí raději teplejší barvy, které jsou příjemnější pro oči. Další důležitou funkcí, kterou by aplikace měla obsahovat, je možnost transpozice. Transpozice je převod akordů z původní tóniny do tóniny, kterou si zvolí uživatel, aby se mu píseň lépe hrála nebo zpívala. Aby nedocházelo ke zbytečnému nahrávání zvukového signálu a případnému vyhodnocování akordů, měla by aplikace obsahovat tlačítka pro spuštění a zastavení záznamu zvuku. Tlačítko **start** dává impuls k vyhodnocování akordů a posouvání textu, tlačítko **stop** naopak ukončuje záznam zvuku a předpokládá, že uživatel píseň dohrál, text bude tedy posunut zpět na začátek. Aby bylo možné pozastavit posun textu a po spuštění pokračovat tam, kde uživatel přestal, bylo by vhodné zavést tlačítko **pause**, které to bude umožňovat. Uvedené funkce by měly být umístěny v tzv. toolbaru, který bude možné přesunout na libovolnou stranu okna.

Text s akordy by měl být zobrazován v podobě řádky s akordy a řádky odpovídající textu, přičemž akordy by měly být uvedeny nad textem v místě, kde má být akord změněn. Pokud předpokládáme, že rozpoznávání akordů může být chybné, bylo by vhodné uvádět akord, který aplikace očekává. Proto bude ve spodní části aplikace řádek (StatusBar), který bude zobrazovat akord, na který aplikace čeká.

Nahrávání zvuku ani vyhodnocování akordů by nemělo omezovat inter-

akci uživatele s aplikací. Z tohoto důvodu by mělo nahrávání a zpracovávání zvuku probíhat paralelně. Paralelní běh je možné zajistit pomocí vláken.

4.2 Příprava akordů

V anechoické místnosti na FAV ZČU v Plzni za pomoci vedoucího bakalářské práce Ing. Kamila Ekšteina, Ph.D. byly na akustickou kytaru nahrány všechny akordy, které by aplikace měla rozpoznávat. Jednalo se o základní akordy durové a mollové, septakordy durových akordů a dim akordy. Dim akordy jsou pouze tři, *C* dim, *D* dim a *G* dim. Díky tomu, že jsou tvořeny tóny, které jsou od sebe přesně 3 půltóny, obsáhnou jejich obraty všechny tóny. Každý akord byl nahráván ve dvou verzích. První verze trvala přibližně jednu minutu a hra na kytaru měla doprovodný, rytmický charakter. Druhá verze trvala přibližně 30 sekund a zaznamenávala vydrnkávání akordů. Akordy byly nahrány s vzorkovací frekvencí 44100 Hz, poté byly programově podvzorkovávány na poloviční vzorkovací frekvenci, tj. 22050 Hz. Tyto akordy jsem využila při implementaci spektrální analýzy a určování parametrů v jejich krocích. Nahrané akordy byly také použity pro výslednou statistiku k určení míry přesnosti vyhodnocovaných akordů, která je uvedena v kapitole 7.

4.3 Překrývání rámců

Nahrávaný signál je vzorkován a ukládán po úsecích stejné délky, tzv. rámcích. Velikost rámce je dána počtem vzorků, které se v daném úseku signálu nacházejí. Z velikosti rámce a vzorkovací frekvence signálu lze určit délku trvání rámce (viz vztah 2.2). Na hranách, tedy na začátku a konci, rámce může vlivem dělení signálu na jednotlivé úseky docházet ke zkreslení signálu. Zkreslení lze do značné míry omezit pomocí funkce okna, tzv. windowing (viz 2.4). Aby nedocházelo ke ztrátě informace na krajích rámců, lze využít překrývání rámců. K překryvu dochází v polovině rámce, tedy rámec je složen z poloviny dat předešlého rámce a poloviny dat nových.

Pokud aplikace bude nahrávat zvuk po rámcích bez využívání překryvů, je zapotřebí zajistit překryv explicitně. Pokud byly například načteny tři rámce, po překrytí jich vznikne celkem pět. Takto vzniklé rámce jsou pak dále využívána k analýze.

4.4 Velikost rámce

Dle dokumentace knihovny **Numpy** [17] je pro funkci `fft.fft`, která implementuje rychlou Fourierovu transformaci, optimální, pokud je počet vstupních vzorků mocnina 2. Dále využijeme faktu, že velikost odhadu výkonové spektrální hustoty získaný Fourierovou transformací odpovídá polovině počtu vstupních vzorků. Zaznamenávaný zvuk v jednom rámci musí mít dostatečnou délku, aby bylo možné akord vyhodnotit.

Pokud například jeden rámec bude mít délku 200 ms, lze rozdělit 1 sekundu nahrávaného zvuku do 5 oken. Předpokládáme-li vzorkovací frekvenci 22050 Hz, tak 200 ms odpovídá 4410 vzorků. Z tohoto důvodu byl nejprve zvolený rámec s 4096 vzorky, a tedy spektrum o velikosti 2048 frekvenčních pásem. Výsledky akordů však téměř neodpovídaly nahraným akordům. Pro zvýšení přesnosti byly vstupní vzorky doplněny nulami na velikost 8192 vzorků. Výsledné spektrum obsahovalo 4096 frekvenčních pásem. Pro získání ještě větší přesnosti nakonec byl zvolený rámec o velikosti 8192 vzorků (nedoplňované nulami).

4.5 Počet zpracovávaných oken

Délka rámce se pohybuje řádově kolem stovek milisekund. V případě této práce, kdy je signál vzorkován frekvencí 22050 Hz a rámec obsahuje 8192 vzorků, je délka rámce přibližně 372 ms. Na základě vlastního zkoumání, ve kterém jsem zjišťovala, za jak dlouho je člověk schopný rozpoznat akord, jsem došla k závěru, že nejkratší možná délka akordu, za kterou jej lze bezpečně rozeznat, je přibližně 450 ms. Z tohoto důvodu jsou v práci k vyhodnocení akordu použity tři překryté rámce. K hodnotě 450 ms jsem došla poslechem nahrávaných akordů a postupnému zmenšování intervalu, dokud jsem nebyla schopna akordy odlišit.

4.6 Emfáze

Jednotlivé akordy jsou vyhodnocovány na základě nižších frekvencí (podrobněji v kapitole 4.12). Pre-emfáze zdůrazňuje vyšší frekvence, vhodnější by tedy byla de-emfáze, která naopak zdůrazňuje nižší frekvence. Vzhledem k malému vlivu na výsledné vyhodnocování akordů nebyl nakonec žádný typ emfáze použit.

4.7 Výběr rámců

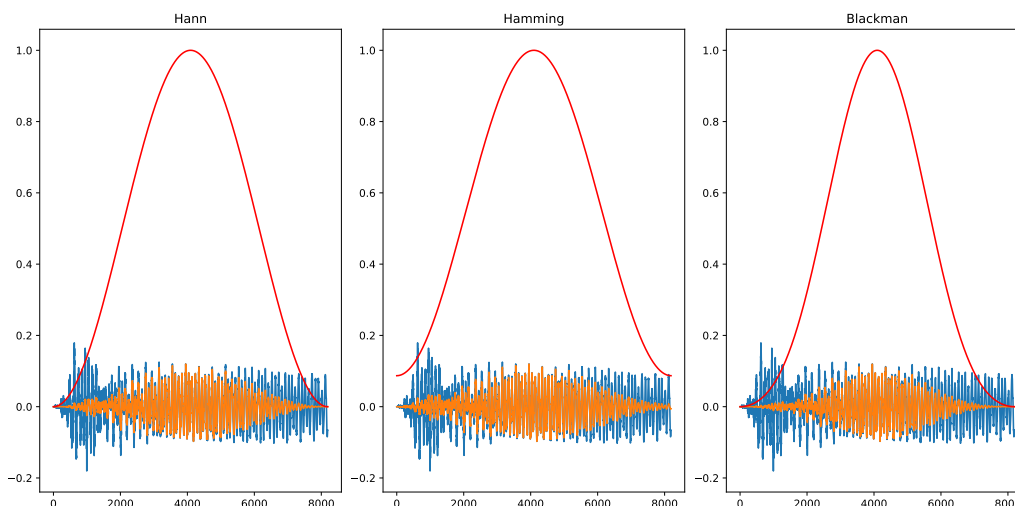
K záznamu zvuku dochází v aplikaci po jeho spuštění kontinuálně. Z tohoto důvodu je dobré vyhodnocovat, zda se jedná o hudbu, hlas, nebo jen o téměř neslyšitelné zvuky, ticho. Na základě střední hodnoty energií a hodnoty příznaku zero crossing je v práci vyhodnocováno, zda je nahrávaný zvuk dále zpracováván nebo ne. U zpracovávaných rámců, ve kterých je zaznamenaná hudba, je vyšší střední hodnota energie a zároveň nižší hodnota příznaku zero crossing než v rámcích, ve kterých je zaznamenané ticho. Hodnoty zero crossing a střední hodnota energií byly určeny experimentálně, viz tab. 4.1. V práci jsou zpracovávány rámce s hodnotou příznaku zero crossing menší než 4096 a zároveň střední hodnotou energií větší než 0,001. Rámce nesplňující tyto podmínky jsou ignorovány.

ticho		hudba	
zero crossing	střední hodnota	zero crossing	střední hodnota
6741	$6 \cdot 10^{-8}$	4070	937
7127	0	3928	131,9
7059	0	3841	1100
6324	$6 \cdot 10^{-8}$	3921	1426
5463	$6 \cdot 10^{-8}$	3841	$3,4 \cdot 10^{-6}$
6794	0	3939	559,5
6879	0	4120	559,5
7152	0	4141	$4,6 \cdot 10^{-6}$
7141	0	4013	$4,3 \cdot 10^{-6}$
6869	$1 \cdot 10^{-7}$	4121	$5,3 \cdot 10^{-6}$

Tabulka 4.1: Porovnání středních hodnot energií a hodnota příznaku zero crossing v případě ticha (nemluví se ani se nehraje) a hudby (hra na kytaru).

4.8 Funkce okna (windowing)

Jak bylo již výše zmíněno, funkce okna je používána k eliminaci zkreslení signálu na krajích rámce. Nejčastěji jsou využívány Hannova, Hammingova a Blackmanova funkce okna, všechny tyto funkce jsou znázorněny na obrázku, viz obr. 4.1. Na krajích rámce mají Hannova i Blackmanova funkce hodnotu 0. Hammingova funkce okna začíná a končí hodnotou 0,8, viz kapitola 2.4. V práci je použita Hammingova funkce okna, která poskytuje více informací o hodnotách signálu na krajích rámce než zbylé zde uvedené funkce.

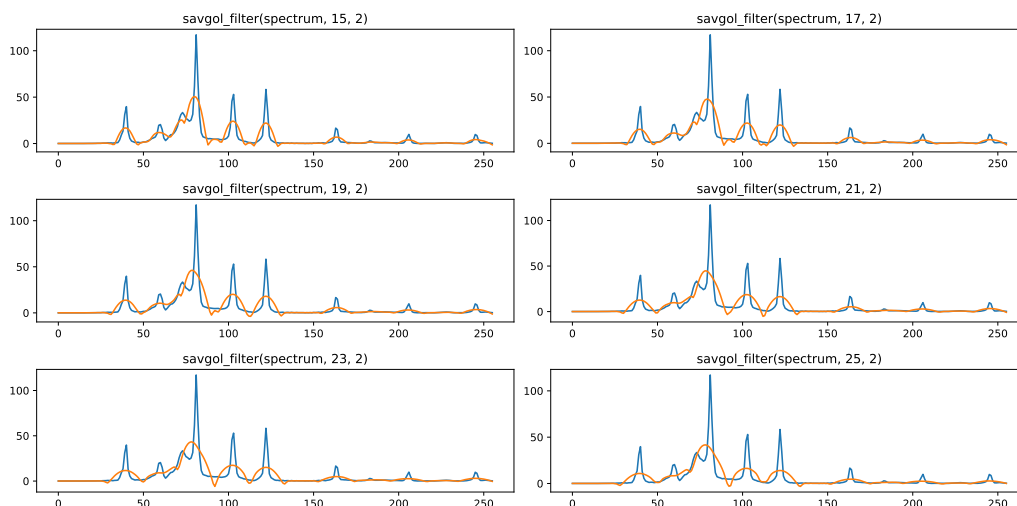


Obrázek 4.1: Porovnání Hannovy, Hammingovy a Blackmanovy funkce okna. Výsledný signál (oranžově) vznikne vynásobením křivky okna (červeně) a původního signálu (modře).

4.9 Savitzky-Golayův filtr

Savitzky-Golayův filtr zajišťuje vyhlazení spektra (výsledek Fourierovy transformace). Vyhlazením spektra jsou eliminovány drobné kmity, ze kterých bychom mohli získat frekvence nenacházející se v akordu. Savitzky-Golayův filtr prokládá v aplikaci spektrum polynomem daného stupně. Výslednou křivku spektra určuje postupně pomocí filtrovacího okna, jehož velikost je dána počtem zadaných bodů, viz kapitola 2.9.

Hodnoty výše uvedených parametrů, tj. stupeň polynomu a velikost filtrovacího okna, je potřeba určit experimentálně. Zvukový signál se skládá ze sinusoid, proto jsem zvolila stupeň polynomu 2, tj. parabola. Velikost filtrovacího okna musí být liché celé číslo, viz kapitola 2.9. V práci jsem vyzkoušela filtrovací okno o velikosti 15, 17, 19, 21, 23 a 25 frekvenčních pásem. Čím větší je velikost filtrovacího okna, tím dochází k většímu vyhlazení spektra, viz obr. 4.2. Při velikostech okna 15, 17 a 19 se ve vyhlazeném spektru projevují některé nerovnosti. Pokud je velikost okna alespoň 21 frekvenčních pásem, je spektrum bez nerovností nebo je jejich výskyt výrazně eliminován. V práci je použita velikost filtrovacího okna 23 frekvenčních pásem, neboť při této hodnotě docházelo k nejpřesnějšímu vyhodnocování akordů.



Obrázek 4.2: Zobrazení původního odhadu PSD (modře) a jeho vyhlazení Savitzky-Golayovým filtrem (oranžově) v každém grafu. Parametry filtru jsou odhad PSD (spectrum), velikost filtrovacího okna (15-25) a stupeň polynomu (2). Každý graf znázorňuje použití jiné velikosti filtrovacího okna.

4.10 Konvoluce

Původní myšlenka vyhlazování zahrnovala lineární konvoluční filtr. Konvoluční filtr byl použitý na jednotlivá spektra - výsledky Fourierovy transformace. Spektra byla průměrována a Savitzky-Golayův filtr byl použit až na průměr všech spekter. Aby nedocházelo k dvojímu vyhlazování spektra, byla konvoluce ze spektrální analýzy v aplikaci odstraněna. Konvoluční filtr byl nahrazen Savitzky-Golayovým filtrem, který je v aplikaci použitý na výsledek Fourierovy transformace, tj. odhad výkonové spektrální hustoty. Po průměrování spekter k žádnému dalšímu vyhlazování již nedochází.

4.11 Průměr spekter

Z každého překrytého rámce vznikne jako výsledek Fourierovy transformace odhad spektrální výkonové hustoty. Takto vzniklá spektra jsou následně vyhlazena Savitzky-Golayovým filtrem. Vyhlazená spektra jsou dále průměrována. Hodnoty průměrovaného spektra jsou určena podílem součtu hodnot všech spekter na odpovídajících indexech a počtu průměrovaných spekter.

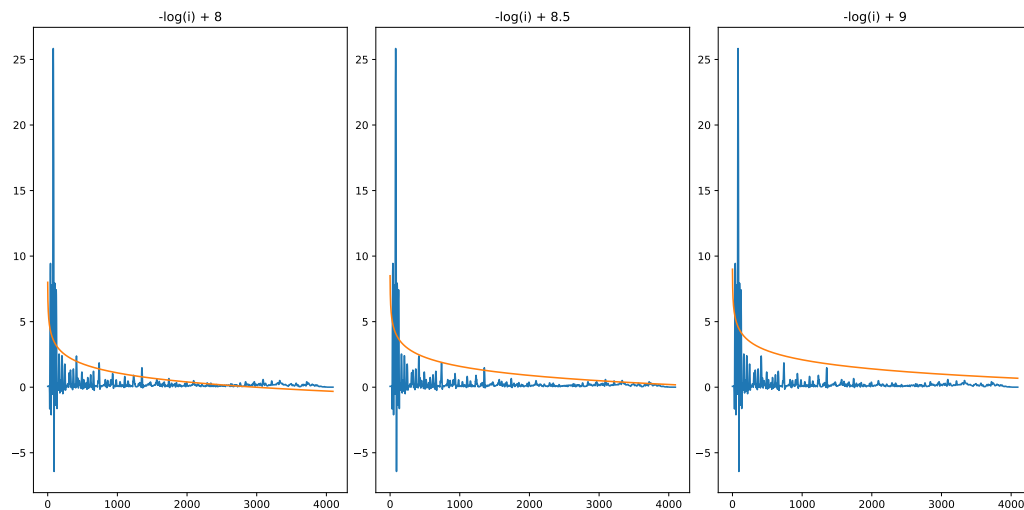
4.12 Váhování spektra logaritmem

Akord lze vyhodnotit již na základě nižších frekvencí spektra, není tedy nutné určovat frekvence na celém spektru. V práci jsou určovány jen frekvence od 53 Hz do 4100 Hz. Vynásobení spektra logaritmem zajistí vyšší energie u nižších frekvencí a zároveň výrazně zmenší energie u vyšších frekvencí, které nebyly při vyhodnocování akordů použity, viz obr. 4.4.

V práci je pro váhování spektra použit záporný logaritmus ve tvaru 4.1, viz obr. 4.3.

$$sp_{log}[i] = (-\log(i) + 8,5) \cdot sp_{avg}[i], \quad (4.1)$$

kde i je index frekvenčního pásma, sp_{log} je vzniklé spektrum, sp_{avg} je průměrované spektrum. Spektrum je tvořeno kladnými hodnotami, proto je logaritmus záporný, hodnota posunutí logaritmu byla určena experimentálně, viz obr. 4.3. Logaritmus je posunutý o hodnotu 8,5, aby se limita v bodě 4096, což odpovídá frekvenci 11025 Hz, na ose x blížila nule.

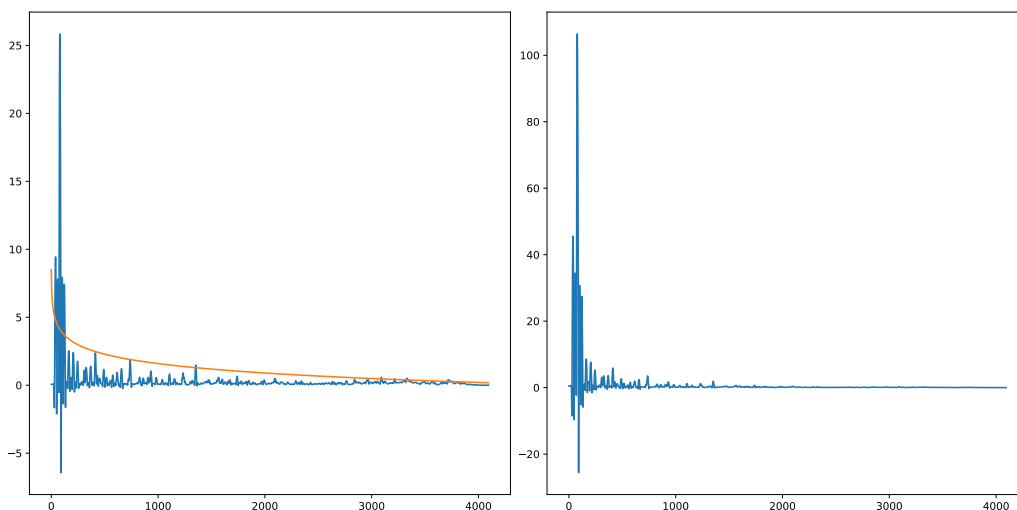


Obrázek 4.3: Zobrazení spektra a logaritmické křivky posunuté o hodnoty 8, 8,5 a 9.

4.13 Střední hodnota spektra

K vyhodnocení akordu není žádoucí určit frekvence všech pásem spektra - získali bychom i frekvence, které nejsou určující pro akord. Střední hodnota spektra vypočtená dle vztahu 4.2 udává hranici energie, při jejímž přesažení dané pásmo zahrneme.

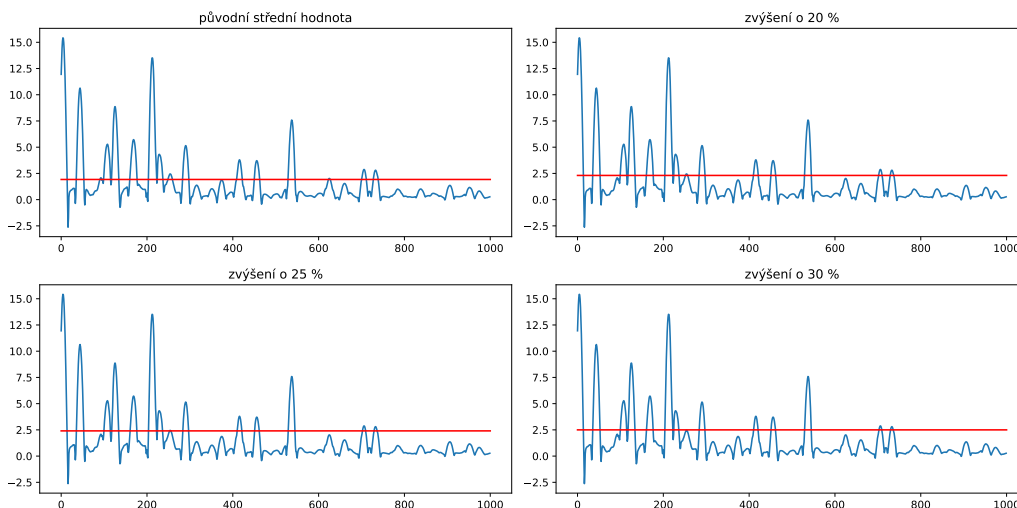
$$E(X) = \frac{\sum_{i=0}^N X[i]}{N}, \quad (4.2)$$



Obrázek 4.4: Zobrazení původního spektra a logaritmické křivky (vlevo) a výsledného spektra po vynásobení spektra logaritmem (vpravo).

kde $E(X)$ označuje střední hodnotu spektra, N počet pásem spektra, $X[i]$ hodnotu (velikost energie) i -tého pásma spektra.

V práci jsem zkusila zvýšit střední hodnotu spektra o 20 %, 25 % a 30 %, aby docházelo jen k vyhodnocování frekvenčních pásem s nejvyššími energiemi, viz obr. 4.5. Nakonec bylo použito zvýšení o 25 %.



Obrázek 4.5: Porovnání původní střední hodnoty (červená konstanta) a jejího zvýšení o 20 %, 25 % a 30 %.

4.14 Určení akordu

K vrcholům spektra s energií vyšší než střední hodnota spektra (zvýšena o 25 %) jsou vypočteny jim odpovídající frekvence. Frekvence vrcholu je vypočtena jako součin šířky pásma (viz vztah 2.7) a indexu spektra, na kterém se daný vrchol nachází.

Jednotlivým frekvencím jsou přidělovány tóny stupnice na základě nejmenšího rozdílu mezi frekvencí tónu a určovanou frekvencí. V práci jsou určovány tóny od A1 (55.0 Hz) do H7 (3951.1 Hz). Pro další analýzu je podstatné o jaký tón se jedná, nikoli z jaké je oktávy.

Každý akord, který je možné určit v této práci, je tvořen třemi nebo čtyřmi tóny. Pokud se akord skládá pouze ze tří tónů, je jako čtvrtý tón přidán první tón akordu, aby všechny akordy byly tvořeny čtyřmi tóny. K určení akordu se využívá podmíněná pravděpodobnost. Ze spektra vyhodnotíme seznam tónů. Předpokládáme, že tóny tvoří akord, některé tóny ale nemusí být určeny správně. U každého akordu proto určíme, jaká je pravděpodobnost, že vyhodnocené tóny tvoří daný akord. Pravděpodobnost akordu je určena na základě počtu shod tónů v akordu a vyhodnocených tónů. Podíl shod a počet zjištěných tónů ze spektra udává pravděpodobnost akordu. Tři akordy s nejvyšší pravděpodobností jsou určeny jako akordy, které mohly být zahrány.

Vyhodnocené akordy jsou porovnávány s očekávanými akordy uvedenými v textu. Každý očekávaný akord je doplněn o akordy, které by mohly být uznány. Pokud je akord durový, je uznáván také jeho septakord a opačně. Dále je mezi uznávané akordy zařazený akord určený dvěma nejhlubšími tóny v 1. obratu (sextakordu), např. k akordu *C* dur je akord vytvořený z 1. obratu *E* moll. Tímto způsobem jsou k durovým akordům přidány mollové a k mollovým durové akordy. Durové akordy jsou vždy doplňovány o jejich septakordy. K dim akordům jsou přidány odpovídající durové akordy a jejich septakordy a mollové akordy vzniklé z 1. obratu, např. akord *C* dim je rozšířený o akordy *C* dur, *C7* dur a *Dis* moll.

Množiny vyhodnocených možných akordů spektrální analýzou a očekávaného akordu doplněný o uznávané akordy jsou porovnávány. Pokud je nalezena shoda, průnik množin je neprázdná množina, aplikace vyhodnotí akord jako zahráný a očekává akord, který je další v pořadí dle textu. Aplikace čeká na akord, dokud nenalezne shodu.

4.15 Posun textu

Text s akordy je posouváný přibližně po stránkách, přesněji po úsecích, které lze v aplikaci zobrazit v textovém poli. Posun textu je závislý na velikosti textového pole a velikosti a fontu písma. Z těchto informací je získáný počet zobrazených řádek. K posunutí textu by mělo docházet, pokud se uživatel dostane na 4. až 3. řádku textového pole (nikoli textu) od spodního okraje textového pole.

Původně aplikace posouvala text po odehrané řádce, docházelo ale k předčasnému posunu textu. Některé akordy např. *A* moll a *C* dur, byly označeny za zahráné dříve, než uživatel stačil zahrát akord *C* dur. To bylo způsobeno tím, že akord *C* dur je vytvořen z 1. obratu akordu *A* moll. Aby se co nejvíce omezilo předčasnému nebo opožděnému posunu textu, byl zvolený posun po více řádcích najednou.

5 Implementace

V této kapitole jsou popsány technologie a knihovny využívané při vytváření aplikace a struktura projektu. Kapitola popisuje jednotlivé kroky implementace, které vychází z předchozí kapitoly návrhu implementace.

5.1 Využité technologie

Vytvořená aplikace je napsaná v programovacím jazyce Python verze 3.8.10. Pomocí instalačního programu **pip**, který je součástí instalace Pythonu, je možné snadno nainstalovat potřebné knihovny (balíky) z Python Package Index (PyPI), což je úložiště softwaru pro programovací jazyk Python.

Systém Git byl použit pro verzování projektu. Git umožňuje spravovat verze nahraných souborů, především se jedná o funkční části zdrojového kódu. Další výhodou systému Git je nahrávání souborů na vzdálený server, čímž dochází k zálohování souborů.

Jako vývojové prostředí byl původně zamýšlen program **PyCharm Edu** od společnosti **JetBrains**, který je pro studenty zdarma a umožňuje snadné psaní a ladění kódu. Z důvodu náročnosti programu na paměť a výkon počítače byl nakonec pro vývoj aplikace používán program **Notepad++**, který umí barevně vyznačit syntax programovacího jazyka Python. Aplikace byla spouštěna z příkazové řádky příkazem `python app.py`, kde `app.py` je hlavní modul aplikace.

K nahrávání jednotlivých akordů byla použita aplikace **Audacity**. Tato aplikace umožňuje pracovat s nahraným zvukem (např. přiblížit grafické zobrazení signálu, přehrát zvolený časový úsek, časový úsek je možné zadat přesně v řádu milisekund) a upravovat jej (např. stříh).

5.2 Využité knihovny a jejich funkce

Při vývoji aplikace byly použité knihovny a moduly **PyQt5** pro vytvoření grafické části aplikace, **PyAudio** k nahrávání zvukového signálu, **NumPy** pro práci s poli, **math** a **statistics** použité v matematických výpočtech a knihovna **SciPy** umožňující některé kroky spektrální analýzy.

5.2.1 PyQt5

Knihovna **PyQt5** vychází z multiplatformního aplikačního frameworku **Qt**. **Qt** umožňuje implementaci uživatelského rozhraní pro desktopové i mobilní aplikace. **PyQt5** je speciálně upravena pro využívání programovacího jazyka Python a **Qt** verze 5, obsahuje 35 rozšiřujících modulů a umožňuje vývoj aplikací na všech podporovaných platformách [19]. Knihovnu **PyQt5** lze nainstalovat příkazem

```
pip install PyQt5
```

Vytvořená aplikace využívá modul **QtWidgets** obsahující vizuální komponenty uživatelského rozhraní, které jsou rozmístěny v okně aplikace. Dalším využívaným modul je **QtGui**, který je v aplikaci použitý k formátování zobrazovaného textu s akordy. Z modulu **QtCore**, který poskytuje základní funkce bez grafického rozhraní, jsou využity třídy pro nastavení velikosti, zarovnání a vlákna.

5.2.2 PyAudio

Knihovna **PyAudio** poskytuje služby multiplatformní knihovny **PortAudio** aplikacím napsaným v programovacím jazyce Python. Knihovna **PyAudio** umožňuje nahrávání a přehrávání zvuku na různých platformách [18]. Knihovnu **PyAudio** je možné nainstalovat příkazem

```
pip install PyAudio
```

Ve vytvořené aplikaci tato knihovna zajišťuje nahrávání zvukového signálu pomocí funkce `open`. Tato funkce nahrává zvukový signál po rámcích zadané velikosti, přičemž signál vzorkuje se zadanou vzorkovací frekvencí. Po nahrání rámce volá zvolenou funkci (callback), která předává informace o nahraném rámci. Knihovna **PyAudio** zajišťuje paralelní nahrávání zvuku, tedy neblokuje grafické rozhraní běžící aplikace. Vytvořená aplikace také využívá funkce knihovny pro spuštění (`start_stream()`) a zastavení nahrávání (`stop_stream()`).

5.2.3 NumPy

Knihovna **NumPy** se v programovacím jazyce Python používá pro práci s poli a vědecké výpočty. **NumPy** obsahuje funkce pro rychlé operace s poli, základní lineární algebru, Fourierovu transformaci, matice a další [17]. V aplikaci je z této knihovny použita funkce `fft.fft`, která provádí diskrétní Fourierovu transformaci. V ostatních případech v aplikaci je knihovna **NumPy**

použita v souvislosti s operacemi týkajícími se polí. Knihovnu **NumPy** lze nainstalovat pomocí programu **pip** příkazem

```
pip install numpy
```

5.2.4 **math, statistics**

Modul **math** poskytuje v programovacím jazyce Python přístup k matematickým funkcím. Modul **statistics** obsahuje funkce z oboru statistiky. Moduly lze nainstalovat pomocí příkazů

```
pip install python-math
```

```
pip install statistics
```

5.2.5 **SciPy**

Knihovna **SciPy** je rozšířením knihovny **NumPy**. Knihovna **SciPy** obsahuje matematické algoritmy a praktické funkce [22]. Ve vytvořené aplikaci je z této knihovny použitý modul **signal**, který poskytuje funkce pro spektrální analýzu. Z tohoto modulu jsou v aplikaci využité funkce pro vyhlazení odhadu výkonové spektrální hustoty Savitzky-Golayovým filtrem (**savgol_filter**) a funkce pro hledání vrcholů (**find_peaks**). Tato funkce **find_peaks** umožňuje nastavení minimální výšky vrcholů, čehož je v aplikaci využito. Knihovnu **SciPy** je možné nainstalovat pomocí programu **pip** příkazem

```
pip install scipy
```

5.2.6 **Matplotlib**

Knihovna **matplotlib** slouží k vizualizaci dat v programovacím jazyce Python. Z knihovny **matplotlib** byl při vývoji aplikace použitý modul **pyplot**, který obsahuje funkce pro práci s grafy. Ve výsledné aplikaci knihovna **matplotlib** použitá není, významně však přispěla k vývoji. Díky této knihovně bylo možné snadno vizualizovat signál v libovolné fázi spektrální analýzy. Knihovnu **matplotlib** lze nainstalovat příkazem

```
pip install matplotlib
```

5.3 Struktura projektu

Projekt je rozdělen do 4 souborů, kde každý soubor zajišťuje jinou funkci aplikace. Hlavními funkcemi aplikace jsou nahrávání zvukového signálu, zpracování signálu a vyhodnocování akordů, tvorba grafické rozhraní a interakce s uživatelem. Aplikace je složena z 3 tříd, `MainWindow`, `Analysis`, `MicrophoneRecorder`.

`const.py`

Soubor poskytuje konstanty, které využívají ostatní soubory projektu. Jsou zde uvedeny hodnoty vzorkovací frekvence, počet vzorků v rámci, maximální a minimální hodnota určených frekvencí, seznam frekvencí jednotlivých tónů a seznam akordů pro transpozici. Soubor `const.py` je importován do všech souborů projektu.

`recorder.py`

Soubor zajišťuje nahrávání zvukového signálu. Soubor je tvořený třídou `MicrophoneRecorder`, která obsahuje funkce pro spuštění a nahrávání zvuku, pro ukládání a předávání nahraných dat jiné třídě. Soubor importuje knihovnu `PyAudio`, která zajišťuje paralelnost nahrávání. Třída pro přístup k proměnné, která uchovává nahraná data, využívá zámku `Lock` z knihovny **`threading`**. Nahraná data jsou ukládána do proměnné `self.frames`. Třída obsahuje funkci, která umožňuje předat uložená data jiné třídě. Po předání těchto dat je proměnná `self.frames` nastavena na počáteční hodnotu. Implementace byla částečně převzata ze zdroje [14]. Soubor `recorder.py` je importován do souboru `app.py`.

`analysis.py`

V tomto souboru jsou definované funkce ke spektrální analýze a vyhodnocení akordu. Soubor obsahuje třídu `Analysis`, která reprezentuje vlákno. Zpracovávání signálu a vyhodnocování akordů probíhá paralelně s nahráváním zvukového signálu a zobrazováním textu. Vlákno je spuštěno při zapnutí nahrávání a běží, dokud uživatel nahrávání nezastaví. Vlákno přebírá informace o nahraném signálu od třídy `MicrophoneRecorder` a ukládá je do proměnné `self.frames`. Uložená data upravuje do požadovaného formátu (počet zpracovávaných oken, překrývání oken atd.) a provádí spektrální analýzu. Ze vzniklého spektra vyhodnotí tři nejpravděpodobnější akordy, které mohly být nahrány. O výsledku informuje třídu `MainWindow` signálem, který slouží k předání informací o progresu vlákna.

app.py

Soubor obsahuje funkce pro spuštění a běh celé aplikace. Třída `MainWindow` zajišťuje vykreslení a rozmístění všech komponent aplikace a interakci s uživatelem. Třída obsahuje funkce pro načtení souboru s textem a akordy ve formátu *ChordPro*. Načtený text rozdělí na akordy a text a zobrazí je v očekávaném formátu, tj. akordy jsou zobrazené nad textem v místě změny akordu. Při spuštění nahrávání uživatelem třída spouští vlákno pro spektrální analýzu a dává pokyn k začátku nahrávání. V průběhu nahrávání dostává výsledky vyhodnocených akordů od třídy `Analysis`, které porovnává s očekávanými akordy z textu. Na základě shodných výsledků vyhodnocených a očekávaných akordů posouvá zobrazovaný text s akordy.

5.4 Popis implementace

Třída `MainWindow` zajišťuje vykreslení aplikace, posun textu a interakci s uživatelem. Soubory s textem a akordickým doprovodem, které uživatel do aplikace vkládá, jsou čteny s kódováním UTF-8, aby bylo možné zobrazit i české písně. Předpokládáme, že text je ve formátu *ChordPro*, kde akordy jsou uvedeny v textu písně v hranatých závorkách. Název písně je ve složených závorkách a je označen `title:`. Údaje ve složených závorkách mohou vymezovat sloky a refrény (`start/end_of_verse`, `start/end_of_chorus`), ostatní části písně jsou uvedeny komentářem ve složených závorkách začínajícím `comment:`, podrobněji viz [3]. Soubor s textem a akordickým doprovodem je čten a zpracováván po řádkách. Z řádků začínající `{` jsou zpracovávány pouze ty, které obsahují název písně nebo komentáře `comment: Instrumental`, `comment: Solo`. Po těchto komentářích je předpokládána řádka obsahující pouze akordy (jedná se např. o mezihry). V tomto případě akordy nejsou uvedeny v hranatých závorkách, ale mohou být odděleny např. znakem `|`.

Název písně je ukládán do proměnné `self.title`. Akordy a text jsou ukládány do vnořených seznamů `self.chords` a `self.text` po jednotlivých řádkách. V seznamu akordů jsou uloženy mezery mezi akordy tak, aby akordy odpovídaly textu. Text písně s akordy zobrazuje funkce `setLyrics()`, která zajišťuje zobrazení písně ve formě řádky akordů a řádky textu a zobrazení písma v nastaveném formátu (tučné akordy, font a velikost písma, atd.).

Aplikace obsahuje volby pro nastavení formátu zobrazovaného textu písně. Uživatel může měnit font, velikost, barvu písma a barvu pozadí. Aplikace umožňuje uživateli transponovat akordy do jiné tóniny. Pro ovládání záznamu zvuku slouží tlačítka `Play`, `Pause` a `Stop`. Aplikace po stisknutí tlačítka `Play` spustí záznam zvuku, který zajišťuje třída `MicrophoneRecorder`

a vlákno třídy `Analysis` pro zpracování zvuku a vyhodnocení akordu. Tlačítko `Pause` pozastaví záznam zvuku a posun textu. Po následném stisknutí tlačítka `Play` vyhodnocování akordů a posun textu pokračuje. V případě ukončení záznamu zvuku tlačítkem `Stop` je předpokládáno, že uživatel hru na nástroj úmyslně ukončil a začne od začátku. Pokud aplikace vyhodnotí, že došlo k zahrání celé písně, záznam zvuku automaticky ukončí (funkce tlačítka `Stop`).

Třída `Analysis` zpracovává zaznamenaný zvuk, dokud probíhá záznam zvuku. Zaznamenané rámce získává od třídy `MicrophoneRecorder` voláním funkce `get_frames()`. Získané rámce dělí na rámce poloviční délky a ukládá je do proměnné `self.frames`. Do následné analýzy zvuku vstupují 3 rámce poloviční délky, což odpovídá 557 ms. Rámce poloviční délky jsou spojovány do rámců původní délky tak, aby docházelo k překryvu rámce původní délky v jeho polovině. Z 3 rámců poloviční délky, vzniknou 2 překryté rámce původní délky. 1 rámeček obsahuje 8192 vzorků. U každého rámce je rozhodnuto na základě střední hodnoty energií a hodnoty příznaku `zero crossing`, zda se jedná o hudbu (hrané akordy, ale i hlas) nebo ticho (šum). Pokud je rámeček vyhodnocen jako hudba, provede se `windowing`, Fourierova transformace, vyhlazení odhadu výkonové spektrální hustoty, dále jen spektra, Savitzky-Golayovým filtrem a vzniklé spektrum je uloženo do seznamu spekter `spect`. Spektra ze seznamu `spect` jsou průměrována. Zprůměrované spektrum je vynásobené logaritmem a je určena jeho střední hodnota. Následně jsou funkcí `find_peaks` z knihovny `SciPy` nalezeny vrcholy spektra a jsou vypočteny jejich frekvence. Jednotlivým frekvencím jsou přiřazeny odpovídající tóny podle nejmenšího rozdílu frekvence tónu a určované frekvence. Na základě počtu výskytů vyhodnocených tónů v každém akordu jsou určeny pravděpodobnosti všech akordů. Tři akordy s nejvyšší pravděpodobností jsou vyhodnoceny jako možné akordy, které byly zahrány.

Třída `Analysis` předává vyhodnocené akordy ze zaznamenaného zvuku třídě `MainWindow` pomocí signálu zaznamenávající progres vlákna. Ve třídě `MainWindow` jsou očekávané akordy z textu rozšířeny o akordy, které jsou v rámci očekávaného akordu akceptovatelné. Z vyhodnocených akordů a akceptovatelných akordů jsou vytvořeny množiny, u kterých je následně určen jejich průnik. Pokud je průnik neprázdná množina, aplikace vyhodnotí očekávaný akord jako zahráný a očekává následující akord v textu.

Třída `MainWindow` obsahuje proměnné `self.lines`, která udává počet zobrazených řádek v textovém poli, a `self.currentLine`, která určuje aktuální řádku, na které se v zobrazeném textovém poli nachází očekávaný akord. Pokud se v seznamu akordů (`self.chords`) vyskytuje prázdná řádka (na odpovídajícím indexu je v seznamu prázdný seznam), je aktuální řádka

zvýšena o 2 (prázdná řádka v akordech i v textu). V případě, že uživatel zahraje řádku s akordy, je aktuální řádka zvýšena o 3 (řádka akordů, textu a prázdná řádka zpřehledňující text). Pokud je aktuální řádka větší nebo rovna počtu zobrazených řádek sníženému o 4 (minimálně poslední řádka textu bude po posunutí zobrazena), dojde k posunu textu s akordy. Posunutí textu je zajištěno zvýšením hodnoty posuvníku o počet zobrazených řádek zmenšený o 4 vynásobený velikostí fontu písma.

6 Testování aplikace

Vytvořená aplikace byla testovaná na nepoučených uživatelích, kteří hrají alespoň amatérsky na nějaký hudební nástroj. Testerů (nepoučení uživatelé) se řídily testovacími scénáři, viz příloha B. Testování aplikace bylo rozděleno na dvě části. První částí byl testovací scénář, který ověřoval funkčnost aplikace. Druhá část obsahovala otázky týkající se především subjektivního hodnocení testerů.

Otázka	Tester 1	Tester 2
Nahrál se soubor správně, včetně diakritických znamének?	ano	ano
Je soubor ve správném formátu?	ano	ano
Jsou akordy zobrazeny tučným písmem?	ano	ano
Změní se font zobrazovaného textu při výběru jiného fontu?	ano	ano
Reaguje zobrazený text na změnu velikosti písma?	ano	ne vždy
Došlo ke změně barvy textu?	ano	ano
Projevila se změna barvy pozadí?	ano	ano
Transponují se akordy do zvolené tóniny?	ano	ano
Po zvolení původní tóniny jsou akordy stejné jako před transpozicí?	ano	ano
Kdy došlo k posunutí textu?	vůbec	pozdě
Zobrazuje se text v místě, ve kterém jste přestal?	ano	ano
Dochází k dalšímu posouvání textu?	ne	ano
Po ukončení a spuštění nahrávání se posunul text na začátek písně?	ano	ano

Tabulka 6.1: Odpovědi testerů na otázky z testovacího scénáře, který testuje funkčnost aplikace.

Z odpovědí testerů, viz tab. 6.1, je vidět, že funkce pro přizpůsobení aplikace uživateli fungují správně. Tester 2 upozorňuje, že velikost textu se změní pouze při přechodu některých hodnot, např. 10-11, 13-14, 16-17. Velikost textu je závislá na fontu písma. Některé fonty mají podrobnější škálu velikosti písma než jiné, proto se změna nemusí vždy projevit. Pokud změníme font z *Courier* např. na *Consolas* změna velikosti písma se projeví vždy.

K posouvání textu, jak uvedli testerů, dochází pozdě nebo dokonce vůbec.

Uživatel tedy odehraje část textu výrazně rychleji, než aplikace posune text. To může být způsobeno tím, že aplikace vyžaduje všechny akordy uvedené v textu. Pokud se například v písni akordy rychle mění, aplikace je nedokáže rozeznat, tudíž čeká na akord, který uživatel již zahrál.

Oba testeři testovali aplikaci s akustickou kytarou a oba k testování použili operační systém Windows. Níže jsou uvedeny odpovědi testerů na otázky týkající se hodnocení aplikace.

1. Byly funkce pro přizpůsobení aplikace uživateli dostačující, nebo Vám některé funkce chyběly?
 - Tester 1: návod přímo v aplikaci (příp. nápověda)
 - Tester 2: Dobrá by byla nápověda pro uživatele. Řádka s očekávanými akordy je přehlednější, dal bych ji přes spodní třetinu obrazovky. Možná bych přidal zvýraznění akordů přímo v textu, které mám hrát (styl karaoke). Při delším načítání bych otevřel okno s loadingem, než se aplikace načte celá.
2. Bylo ovládání aplikace intuitivní, nebo pro Vás některé funkce nebyly srozumitelné. Případně uveďte jaké a proč.
 - Tester 1: Řádek s očekávanými akordy je podle mého důležitý, ale všimla jsem si ho až po značné době.
 - Tester 2: Nepostřehl jsem řádek s instrukcemi dole (s akordy).
3. Zhodnoťte posouvání textu. Docházelo k posouvání textu ve vhodnou dobu? Uvítal byste raději posun např. po jednotlivých řádkách?
 - Tester 1: Nedokážu vyhodnotit, zda by to bylo lepší, ale při posunu „skokem“ jsem se občas ztratila v textu.
 - Tester 2: Posun po řádkách se zvýrazněním konkrétních akordů by byl nejlepší. Udělal bych to stejně jako karaoke.
4. Máte nějaké připomínky k aplikaci?
 - Tester 1: Jen bych zdůraznila důležitost řádku s očekávanými akordy.
 - Tester 2: Load button by mohl mít ascii symbol diskety U+1F4BE. Zobrazit loading při spouštění aplikace. Větší řádku pro akordy a text zarovnat na střed. Přiložit k aplikaci testovací soubor.

Testerům v aplikaci chyběla nápověda pro uživatele, která by vysvětlovala jednotlivé funkce zobrazených komponent. Tyto funkce jsou popsány v uživatelské dokumentaci, viz příloha C. Řádek s očekávanými akordy byl původně vytvořen pouze k testovacím účelům. Z důvodu nepřesného vyhodnocování akordů byl nakonec v aplikaci ponechán pro lepší orientaci v textu. Uznávám, že by mohl být více zdůrazněn. Posun „skokem“ může být pro některé uživatele matoucí. Pokud však uživatel přijde na princip, bude se v textu dobře orientovat. Zaujal mě nápad, zvýrazňování akordů a textu ve formě karaoke. Někteří uživatelé by jistě formu karaoke uvítali, jiní by naopak dali přednost statickému textu, který by se posouval po „stránkách“. Možnost karaoke by mohla být rozšiřující funkcí aplikace, kterou by uživatel mohl využít.

7 Správnost vyhodnocení akordů

Ověření správnosti vyhodnocování akordů bylo provedeno se zvukovými záznamy jednotlivých akordů. Záznamy akordů byly nahrány na akustickou kytaru v anechoické místnosti na FAV ZČU v Plzni. Ke každému akordu, který by aplikace měla rozpoznávat, jsou vytvořeny dva záznamy. První záznam zaznamenává hru rytmu, druhý vydrnkávání akordu. Dohromady záznamy u každého akordu zaznamenávají přibližně 1,5 minuty zvukového signálu.

Jednotlivé záznamy byly rozděleny na rámce o velikosti 8192 vzorků se vzorkovací frekvencí 22050 Hz. K určení akordu bylo použito 1,5 rámce, což odpovídá přibližně 557 ms zvukového záznamu. Rámce byly následně překryty, tudíž vznikly 2 rámce o velikost 8192 vzorků. Ze všech takto vzniklých rámců byly vyhodnoceny 3 nejpravděpodobnější akordy, které mohly být zahrány.

U vyhodnocených akordů bylo možné ověřit jejich správnost, neboť jsme věděli, jaký akord byl zpracováván. Výsledky byly rozděleny do 3 kategorií - přesně určené akordy, akceptovatelné akordy, nesprávně určené akordy. Pokud seznam vyhodnocených akordů obsahovala akord, který byl zpracováván, byl označen jako přesně určený akord. Pokud seznam vyhodnocených akordů neobsahoval zpracováváný akord, ale některý z akceptovatelných akordů (podrobněji viz 4.14), byl zařazen do akceptovaných akordů. V případě, že vyhodnocený seznam akordů neobsahoval zpracováváný ani akceptovatelný akord, byl označen za nesprávně vyhodnocený akord. Výsledky vyhodnocování akordů v jednotlivých úsecích jsou uvedeny v tab. 7.1, kde úsekem je označeno 1,5 rámce, tedy 557 ms zvukového záznamu.

Přesné	Akceptovatelné	Nesprávné	Celkem úseků
2554	1012	2166	5732

Tabulka 7.1: Výsledky vyhodnocování akordů v jednotlivých úsecích zaznamenaných akordů rozdělené do 3 kategorií - přesně určené akordy (přesné), akceptovatelné akordy (akceptovatelné), nesprávně určené akordy (nesprávné). Sloupec celkem úseků vyjadřuje celkový počet zpracováváných úseků (1,5 rámce, tj. 557 ms zvukového záznamu).

Na základě získaných výsledků, viz tab. 7.1, lze určit úspěšnost vyhod-

nocování akordů. Akord byl přesně určen v 2554 případech, což odpovídá 45 %. Pokud bychom za správně vyhodnocené akordy uvažovali přesně určené akordy a akceptovatelné akordy, získáme 3566 správně určených případů, což odpovídá 62 %. V 2166 případech, tj. 38 %, byl akord vyhodnocen jako nesprávně určený. Vzhledem k tomu, že aplikace vyhodnotí akord za zahráný v případě přesně určeného i akceptovatelného akordu, lze říct, že úspěšnost vyhodnocování akordů je 62 %.

8 Závěr

Ve své bakalářské práci jsem se snažila vytvořit zpěvník s automatickým posunem textu podle aktuálně hraných akordů. Vznikla aplikace s jednoduchým uživatelským rozhraním, která zaznamenává zvukový signál, vyhodnocuje z něj konkrétní akordy a podle nich posouvá zobrazovaný text písně.

Před samotným vývojem aplikace jsem musela nastudovat základní informace týkající se procesu zpracování zvukového signálu, spektrální analýzy a hudební teorie v souvislosti s akordy. Také jsem společně se svým vedoucím práce pořídila zvukové záznamy jednotlivých akordů, které by aplikace měla rozpoznávat.

Po prostudování teoretických poznatků jsem navrhla aplikaci implementovanou v programovacím jazyce Python. Aplikace ze zvuku zaznamenaného v reálném čase vyhodnocuje akordy a porovnává je s očekávanými akordy z textu písně. Na základě shody pak posouvá text po „stránkách“.

Aplikaci otestovali nepoučení uživatelé, kteří se řídili testovacími scénáři. Výsledky testů potvrdily funkčnost ovládacích prvků aplikace, ale zároveň odhalily pomalé a ne zcela přesné vyhodnocování akordů, v důsledku čehož nedochází k posunu textu, jak by uživatel předpokládal. Podle provedeného ověření správnosti vyhodnocování akordů, ke kterému byly použity jednotlivé záznamy nahraných akordů, určí aplikace shodu hraného akordu s akordem rozpoznaným v 62 % případů.

Pro vyhodnocování akordů by mohla aplikace využívat umělou neuronovou síť, čímž by mohlo dojít ke zvýšení přesnosti určení akordu. Aplikace by mohla být rozšířena o funkci umožňující zvýrazňovat a posouvat text s akordy formou karaoke, případně možnost uchování nahraných písní v aplikaci.

Literatura

- [1] BÄCKSTRÖM, T. *Pre-emphasis* [online]. Aalto University Wiki, 2019. [cit. 2022/04/04]. Dostupné z: <https://wiki.aalto.fi/display/ITSP/Pre-emphasis>.
- [2] CHAUDHARI, Q. *Understanding sample rate conversion and scaling factors* [online]. EDN, 2016. [cit. 2021/01/04]. Dostupné z: <https://www.edn.com/understanding-sample-rate-conversion-and-scaling-factors/>.
- [3] *ChordPro directives* [online]. [cit. 2022/05/3]. Dostupné z: <https://www.chordpro.org/chordpro/chordpro-directives/>.
- [4] COX, G. *Peak Detection in a Measured Signal* [online]. Baeldung, 2020. [cit. 2022/04/30]. Dostupné z: <https://www.baeldung.com/cs/signal-peak-detection>.
- [5] DUBEY, A. *Fourier Transformation and Its Mathematics* [online]. Toward Data Science, 2018. [cit. 2021/04/28]. Dostupné z: <https://towardsdatascience.com/fourier-transformation-and-its-mathematics-fff54a6f6659>.
- [6] GALLAGHER, N. B. *Savitzky-Golay Smoothing and Differentiation Filter* [online]. Eigenvector Research Incorporated, 2020. [cit. 2022/04/07]. Dostupné z: <https://eigenvector.com/wp-content/uploads/2020/01/SavitzkyGolay.pdf>.
- [7] HLAVÁČ, V. *Fourierova transformace v 1D a 2D* [online]. České vysoké učení technické v Praze, 2020. [cit. 2021/01/02]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/12FourierTxCz.pdf>.
- [8] HOLČÍK, J. – KOMENDA, M. *1.2 Vzorkovací teorém* [online]. Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity. [cit. 2021/01/04]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-modelovani-dynamickych-biologickych-dat--signaly-a-linearni-systemy--casove-rady-i--1-vzorkovani--1-2-vzorkovaci-teorem>.
- [9] KLUSAL, R. *Obraty akordů* [online]. 2015. [cit. 2022/05/1]. Dostupné z: <http://klavir.klusik.cz/obraty-akordu/>.

- [10] KRÁLOVÁ, M. *Vznik a druhy zvuku* [online]. Techmania Science Center o.p.s., 2007. [cit. 2022/05/1]. Dostupné z: <http://edu.techmania.cz/cs/encyklopedie/fyzika/akustika/vznik-druhy-zvuku>.
- [11] KRÁLOVÁ, M. *Výška zvuku* [online]. Techmania Science Center o.p.s., 2007. [cit. 2022/05/1]. Dostupné z: <http://edu.techmania.cz/cs/encyklopedie/fyzika/akustika/vyska-zvuku>.
- [12] KRÁLOVÁ, M. *Barva zvuku* [online]. Techmania Science Center o.p.s., 2007. [cit. 2022/05/1]. Dostupné z: <http://edu.techmania.cz/cs/encyklopedie/fyzika/akustika/barva-zvuku>.
- [13] KRÁLOVÁ, M. *Základy hudební akustiky* [online]. Techmania Science Center o.p.s., 2007. [cit. 2022/05/1]. Dostupné z: <http://edu.techmania.cz/cs/encyklopedie/fyzika/akustika/zaklady-hudebni-akustiky>.
- [14] LEBOURDAIS, F. *Building a PyQt application to record your microphone and plot its spectrum* [online]. 2015. [cit. 2022/05/2]. Dostupné z: <https://flothosof.github.io/pyqt-microphone-fft-application.html>.
- [15] LEPIL, O. *Fyzika pro gymnázia Mechanické kmitání a vlnění*. Prometheus, spol. s.r.o, 2001. ISBN 80-7196-216-3.
- [16] MUSÍLEK, J. *Pianovka* [online]. 2020. [cit. 2022/05/3]. Dostupné z: <http://www.pianovka.cz/index.php>.
- [17] *NumPy Documentation* [online]. [cit. 2022/04/26]. Dostupné z: <https://numpy.org/doc/stable/>.
- [18] *PyAudio* [online]. [cit. 2022/04/26]. Dostupné z: <http://people.csail.mit.edu/hubert/pyaudio/>.
- [19] *PyQt5* [online]. 2021. [cit. 2022/04/26]. Dostupné z: <https://pypi.org/project/PyQt5/>.
- [20] ROSE, W. C. *Windowing* [online]. Mathematics and Signal Processing for Biomechanics, 2011. [cit. 2021/01/03]. Dostupné z: <https://www1.udel.edu/biology/rosewc/kaap686/notes/windowing.html>.
- [21] SCHALDENBRAND, P. *Digital Signal Processing: Sampling Rates, Bandwidth, Spectral Lines, and more...* [online]. Siemens Digital Industries Software, 2020. [cit. 2022/04/09]. Dostupné z: <https://community.sw.siemens.com/s/article/digital-signal-processing-sampling-rates-bandwidth-spectral-lines-and-more>.

- [22] *SciPy - Introduction* [online]. [cit. 2022/04/26]. Dostupné z: <https://scipy.github.io/devdocs/tutorial/general.html>.
- [23] TOLEDO PÉREZ, D. – RODRIGUEZ, J. – GÓMEZ LOENZO, R. A Study of Computing Zero Crossing Methods and an Improved Proposal for EMG Signals. *IEEE Access*. 01 2020, PP, s. 1–1. doi: 10.1109/ACCESS.2020.2964678. Dostupné z: https://www.researchgate.net/publication/338444312_A_Study_of_Computing_Zero_Crossing_Methods_and_an_Improved_Proposal_for_EMG_Signals.
- [24] ZÁHORA, J. *Učebnice statistiky* [online]. Univerzita Karlova v Praze, Lékařská fakulta v Hradci Králové, 2015. [cit. 2022/04/09]. Dostupné z: <https://publi.cz/books/201/13.html>.

A Seznam zkratek

A/D	Analog to Digital
FAV	Fakulta aplikovaných věd
FIR	Finite Impulse Response
Hz	hertz
ms	milisekunda
PSD	Power Spectral Density
ZČU	Západočeská univerzita

B Testovací scénář

B.1 Ověření funkčnosti aplikace

Pro lepší orientaci a pochopení, kde se aplikace v textu nachází, slouží řádek pod textem, který zobrazuje akord, na který čeká.

1. Nahrajte do aplikace text s akordy ve formátu ChordPro (akordy jsou uvedeny v textu v hranatých závorkách).
 - Nahrál se soubor správně, včetně diakritických znamének?
 - Je soubor ve správném formátu, tj. akordy jsou zobrazeny nad textem v místě, kde má dojít ke změně akordu?
 - Jsou akordy zobrazeny tučným písmem?
2. Vyzkoušejte více možností fontu písma a zvolte si libovolný z nich.
 - Změní se font zobrazovaného textu při výběru jiného fontu?
3. Nastavte velikost písma zadáním čísla v rozmezí 10 až 30, nebo použijte šipky (nahoru a dolů).
 - Reaguje zobrazený text na změnu velikosti písma?
4. Nastavte libovolnou barvu písma a pozadí.
 - Došlo ke změně barvy textu?
 - Projevila se změna barvy pozadí?
5. Transponujte akordy do jiné tóniny.
 - Jsou akordy transponované do zvolené tóniny?
6. Transponujte akordy zpět do původní tóniny.
 - Jsou akordy stejné jako před transpozicí?
7. Zapněte nahrávání zvuku a začněte hrát akordy podle textu. Po prvním posunutí pozastavte nahrávání.
 - Kdy došlo k posunutí textu?
8. Opět zapněte nahrávání a pokračujte ve hře, kde jste přestal.

- Zobrazuje se text v místě, ve kterém jste přestal?
 - Dochází k dalšímu posouvání textu?
9. Zastavte nahrávání a znovu jej spusťte.
 - Posunul se text na začátek písně?

B.2 Hodnocení aplikace

1. S jakým hudebním nástrojem jste aplikaci testoval?
2. Na jakém operačním systému jste aplikaci testoval?
3. Byly funkce pro přizpůsobení aplikace uživateli dostačující, nebo Vám některé funkce chyběly?
4. Bylo ovládání aplikace intuitivní, nebo pro Vás některé funkce nebyly srozumitelné. Případně uveďte jaké a proč.
5. Zhodnoťte posouvání textu. Docházelo k posouvání textu ve vhodnou dobu? Uvítal byste raději posun např. po jednotlivých řádkách?
6. Připomínky k aplikaci

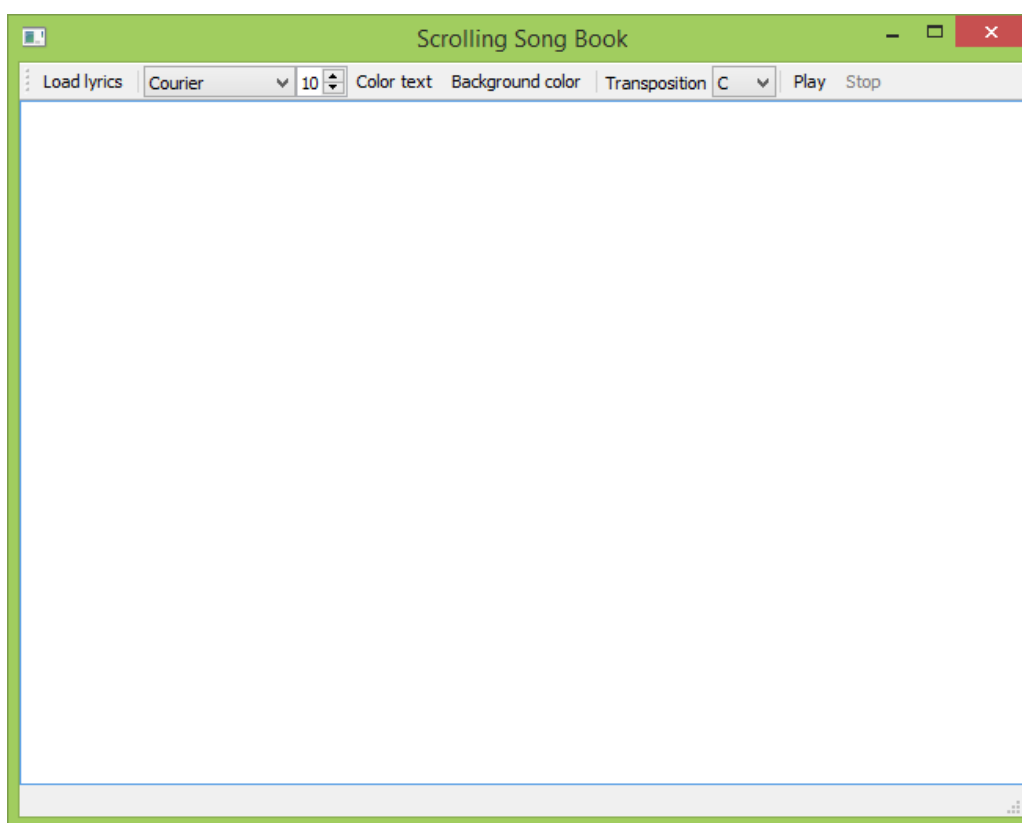
C Uživatelská dokumentace

Aplikace posouvá text písně s akordy na základě zaznamenaného zvuku a vyhodnocených akordů, jedná se o zpěvník s automatickým posunem textu.

Pro spuštění aplikace je potřeba mít nainstalovaný programovací jazyk Python. Aplikaci je pak možné pustit z adresáře, který obsahuje zdrojové kódy aplikace, v příkazové řádce příkazem

```
python app.py
```

Po spuštění je aplikace zobrazena v samostatném okně, viz obr. C.1.



Obrázek C.1: Grafické rozhraní vytvořené aplikace.

V horní části okna je zobrazen panel nástrojů, který lze uchopením na levé straně v místě teček přesunout na libovolnou stranu okna. Bílá plocha představuje textové pole, které slouží k zobrazení načteného textu s akordy. Ve spodní části okna je vymezený prostor, ve kterém jsou zobrazovány akordy z textu písně, které aplikace očekává a vyžaduje.

Panel nástrojů obsahuje tlačítko **Load lyrics** pro vložení textu akordy ze souboru. Text s akordy by měl být ve formátu *ChordPro*, aby aplikace fungovala správně. Vedle tlačítka **Load lyrics** se nacházejí funkce pro změnu fontu a velikosti písma zobrazovaného textu. Následují tlačítka pro změnu barvy textu (**Color text**) a barvy pozadí (**Background color**). Pojmem **Transposition** je uvedena funkce umožňující transponovat zobrazené akordy do jiné tóniny. Tlačítka **Play** a **Stop** slouží k ovládání zaznamenávání zvuku. Tlačítko **Play** se po stisknutí změní na **Pause** a spustí záznam zvuku. Pokud chcete zaznamenávání zvuku z nějakého důvodu pozastavit a po opětovném spuštění pokračovat v místě, kde jste přestali, stiskněte tlačítko **Pause**. V případě stisknutí tlačítka **Stop** a následně tlačítka **Play** začne aplikace vyhodnocovat akordy od začátku písně.

D Obsah přiloženého ZIP souboru

ZIP soubor obsahuje následující složky a soubory:

- *Text_prace* obsahující průvodní dokument bakalářské práce ve formátu PDF,
- *Aplikace_a_knihovny* obsahující zdrojové soubory aplikace,
- soubor *Readme.txt*, ve kterém je popsána adresářová struktura ZIP souboru.