

Porovnání audio vzorků

Comparison of Audio Samples

Bc. Filip Dostál



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Filip Dostál**
Osobní číslo: **A13543**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **kombinovaná**

Téma práce: **Porovnávání audio vzorků**

Téma anglicky: **The Comparison of Audio Samples**

Zásady pro vypracování:

1. Zpracujte stručný přehled dostupných technologií a opensource řešení pro oblast porovnávání audio záznamů.
2. Připravte softwarové moduly pro porovnávání krátkých zvukových stop.
3. Navrhněte a realizujte jednoduchou ukázkovou aplikaci pro operační systém MS Windows.
4. Ověřte funkčnost navržené aplikace a zhodnoťte možnosti dalšího vývoje.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

1. KOSTOLÁNYOVÁ, Kateřina. Úvod do multimédií: (grafika, hudba a zvuk). Vyd. 1. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2003, 54 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-704-2924-0.
2. POLÁK, Josef. Integrální a diskrétní transformace. 3., přeprac. vyd. V Plzni: Západočeská univerzita, Fakulta aplikovaných věd, 2002, 197 s. ISBN 80-708-2924-9.
3. BERG, Richard E. ENCYKLOPAEDIA BRITANNICA. Sound [online]. 2006 [cit. 2014-01-25]. Dostupné z: <http://www.britannica.com/EBchecked/topic/555255/sound>
4. HEROUT, Pavel. Učebnice jazyka Java. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.
5. SCHILDT, Herbert. Java 7: výukový kurz. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.

Vedoucí diplomové práce:

Ing. Viliam Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

7. února 2014

Termín odevzdání diplomové práce:

27. května 2014

Ve Zlíně dne 7. února 2014

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Cílem práce je zpracování softwarového řešení tématu porovnávání krátkých zvukových stop v jazyku Java a realizování GUI desktopové ukázkové aplikace v prostředí MS Windows. Součástí práce je tvorba modulů zajišťující chod programu, grafické rozhraní určené k prezentování výstupů z jádra. Popisuje modifikované části open-source knihovny Musicg API. Současně popisuje experimentální měření vymezující parametry určené pro rozeznání zvuku. Teoretické části je věnována rešarše současných dostupných technologií a „open source“ řešení pro oblast porovnávání audio záznamů. V dalších kapitolách je také více přiblížena Fourierova transformace a problematika akustických otisků. Závěr práce hodnotí funkčnost navržené aplikace a možnost jejího dalšího vývoje.

Klíčová slova: spektrální analýza, Fourierova transformace, Fourier, FFT, Java, zvukový otisk, acousticID, GUI, akustický otisk, Musicg API, porovnávání zvuků.

ABSTRACT

The objective of this thesis is processing of software solution consist of comparing of short audio sample in Java programming language and implementing GUI application of desktop demonstration in operation system MS Windows. Describes modified parts of open-source libraries Musicg API. Part of the thesis describes creation of modules witch ensure measuring sample parameters, also describes graphical interface designed for presentation of the outputs of the comparison's score. At the same time describes experimental measurements which define parameters for the recognition of sound. The theoretical part of the thesis is consecrated to research of currently available technologies and open source solutions for comparison audio records. In next chapters are also more introduced Fourier transform and the issue of acoustic fingerprint. The conclusion of the thesis evaluates the functionality of the demonstrated application and the possibility of its further development.

Keywords: spectral analysis, Fourier transform, FFT, Java, acoustic fingerprint, acousticID, GUI, Musicg API, sound comparison.

Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu diplomové práce panu Ing. Viliamu Dolinayovi Ph.D. za jeho podporu a mnoho cenných připomínek k této diplomové práci. Dále bych rád poděkoval své rodině za morální i finanční podporu při studiu, také mé přítelkyni za její pozitivní přístup a na závěr také mému zaměstnavateli za vytvoření flexibilních pracovních podmínek během realizace této práce.

Motto

„V životě můžete dosáhnout čehokoliv, máte-li odvahu o tom snít, inteligenci vytvořit realistický plán a vůli dotáhnout tento plán do konce.“

Prohlašuji, že

- беру на вѣдомі, же оdevздáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- беру на вѣдомі, же дипломová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- беру на вѣдомі, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- беру на вѣдомі, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- беру на вѣдомі, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- беру на вѣдомі, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST.....	12
1 FYZIKÁLNÍ VLASTNOSTI ZVUKU	13
1.1 ZDROJE ZVUKU	13
1.2 UCHO JAKO SPEKTRÁLNÍ ANALYZÁTOR	14
1.3 ZVUKOVÉ UKAZATELE VHODNÉ PRO ZVUKOVOU IDENTIFIKACI.....	14
1.3.1 Zero-crossing rate (ZRC)	14
1.3.2 Intenzita zvuku	14
1.3.3 Směrodatná odchylky	14
2 ANALÝZA ZVUKU.....	15
2.1 ÚVOD DO FOURIEROVY ANALÝZY	15
2.2 RYCHLÁ FOURIEROVA TRANSFORMACE (FFT).....	17
2.3 AKUSTICKÝ OTISK ZVUKU	17
3 BIOMETRIE LIDSKÉ ŘEČI	19
3.1 OBECNÝ MODEL ROZPOZNÁVÁNÍ ŘEČI.....	19
4 ANALÝZA DOSTUPNÝCH APLIKACÍ POROVNÁVÁNÍ AUDIO ZÁZNAMU	21
4.1 OPEN-SOURCE APLIKACE.....	21
4.1.1 Comparisonics	21
4.1.2 FindSounds.....	22
4.1.3 Picard.....	25
4.1.4 AcoustID	26
4.1.5 Audioscout	26
4.1.6 OpenFP	27
4.1.7 Echoprint	29
4.1.8 libFooID	30
4.2 ČÁSTEČNĚ OPEN-SOURCE APLIKACE.....	30
4.2.1 Last.fm	31
4.3 KOMERČNÍ APLIKACE	31
4.3.1 AudibleMagic.....	31
4.3.2 Content ID	32
4.3.3 Shazam	32
4.3.4 Gracenote	32
4.4 ZÁVĚR ANALÝZY DOSTUPNÝCH SOFTWAREVÝCH NÁSTROJŮ	33
II PRAKTICKÁ ČÁST	34
5 REALIZACE SOFTWAREVÉHO ŘEŠENÍ TŘÍDĚNÍ ZVUKU.....	35
5.1 MUSICG API.....	36
5.2 DEFINICE ÚKOLU	36
5.3 RANNÉ FÁZE NÁVRHU	37
5.4 TESTOVACÍ APLIKACE KLASIFIKACE ZVUKU	37
5.4.1 Zvuková identifikace.....	39
5.4.2 Zvolené Charakteristické parametry	40

5.5	ANALÝZA PARAMETRŮ ZVUKŮ A JEJICH IMPLEMENTACE V APLIKACI.....	41
5.6	ÚPRAVA VÝSLEDKŮ MĚŘENÍ	45
5.7	REDUKCE FALEŠNÝCH ROZPOZNÁNÍ	46
5.8	POSTUP PŘIDÁNÍ NOVÉ „ACTION“ DO SEZNAMU ROZPOZNATELNÝCH ZVUKŮ	48
6	UML CLASS MODEL.....	49
6.1	MODEL GUI.....	49
6.2	MODEL NAHRÁVÁNÍ A PŘEHRÁVÁNÍ ZVUKU	50
6.3	MODEL TŘÍD VYKRESLOVÁNÍ	51
6.4	MODEL „ACTION VLASTNOSTI ZVUKOVÉHO OTISKU“	53
6.5	MODEL TŘÍD ZAJIŠŤUJÍCÍ MANIPULACI SE ZVUKOVÝM SIGNÁLEM	54
7	ZHODNOCENÍ APLIKACE	57
	ZÁVĚR	58
	ZÁVĚR V ANGLIČTINĚ.....	60
	SEZNAM POUŽITÉ LITERATURY.....	61
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	64
	SEZNAM OBRÁZKŮ	66
	SEZNAM TABULEK.....	67

ÚVOD

Zvuk, mechanické vlnění, které nám svou schopností šířit se prostředím podobně jako se šíří i světlo a vůně umožňuje získat autentickou informaci o okolním prostředí, aniž bychom s tímto prostředím museli přijít přímo do kontaktu. Tato fyzikální skutečnost nám umožňuje orientovat se v prostoru kolem nás a závisí na ní elementární principy lidské komunikace. Většina živočichů tyto informace prostřednictvím zvuků nějakým způsobem vnímá a vyhodnocuje. Podle některých vědců se zvukem řídí i některé rostliny, které také působením některých frekvencí mění svůj vegetační cyklus. Schopnost slyšet nebo jiným způsobem vyhodnocovat mechanické vlnění je tedy poměrně běžnou součástí života na naší planetě. My lidé používáme pět základních smyslů, každý z nich nám o našem okolí může prozradit jinou informaci. Je to sluch, který nám umožňuje sofistikovaně vyhodnocovat mluvenou řeč, rozpoznávat jednotlivá slova, věty a celé myšlenky. To je základní dovednost, bez které by nemohla žádná lidská společnost dobře fungovat. Tato práce z těchto poznatků vychází a klade si za cíl vytvořit takové technické řešení, které bude schopné detekovat zvuky podobně jako lidský sluch.

Diplomová práce je členěna do 2 částí. V první části je definována problematika teorie analýzy zvuku. Druhá část diplomové práce dokumentuje výstup realizace ukázkové testovací verze aplikace, která umožňuje rozeznávat definované druhy zvuků.

V první části práce představují teoretický úvod k dané problematice. Zabývám se zde teorií mechanického vlnění a všech aspektů hrajících důležitou roli v rozpoznávání zvuku. Práce si klade za cíl zodpovědět tyto otázky: 1. Na základě, kterých fyzikálních jevů jsme schopni zvuky kategorizovat? 2. Jakou mají důležitost a charakter? 3. Jak tyto poznatky využít v praktické části? Také definuje fyzikální jevy používané při zvukové analýze a rozebírá metodu FFT Rychlé Fourierově transformace a akustických otisků.

Ve druhé teoretické části se věnuji vlastní analýze formou rešarše dostupných softwarových řešení zabývajících se analýzou a rozpoznávání zvuku. Softwarové produkty člením na komerční a open-source řešení. Zde si práce klade za cíl popsat stávající dostupné technologie.

Praktická část popisuje realizaci softwarového řešení ukázkové aplikace, umožňující rozpoznávání definovaných zvuků. Součástí bude dokumentace postupující chronologicky od návrhu k realizaci, dokumentace je i součástí testovací aplikace. V praktické části jsou

popsány použité technologie zejména pak open-source knihovna Musicg. V závěru kapitoly hodnotím funkčnost realizovaného řešení a možnosti dalšího vývoje.

Závěr práce hodnotí význam teoretické části a aplikování rešaršovaných poznatků při návrhu aplikace. Souhrnně rekapituluje splnění bodu zadání, které bylo prezentováno ukázkovou testovací aplikací, která je hlavním výstupem této práce.

I. TEORETICKÁ ČÁST

1 FYZIKÁLNÍ VLASTNOSTI ZVUKU

Ze všeho nejdřív je nutné říct, co to zvuk je a co jej definuje. Zvuk chápeme jako mechanické vlnění a to buďto pevného prostředí vzduchu, kapalného prostředí, nebo plynného. U zvuku můžeme měřit tyto parametry - frekvenci, vlnovou délku, amplitudu, rychlost, intenzitu a směr.

Část tohoto vlnění (zvuku) můžeme vnímat jako slyšitelný zvuk. Pro lidské ucho je to vlnění o frekvenci od 20Hz do 20kHz. Nicméně lidský sluch není dokonalý a slyšitelné pásmo je do značné míry individuální záležitost. Především horní hranice slyšitelnosti dosahuje jen málokdo a s přibývajícím věkem je tento fakt ještě výraznější.

Ze slyšitelné frekvence, vychází i vzorkovací frekvence. Minimální kmitočet, který splňuje vzorkovací teorém je 40kHz. Pro kompaktní disky se volí vzorkování frekvencí 44kHz. Dle Nyquistova teorému lze se vzorkovací frekvencí zaznamenat signál s maximální frekvencí 22050Hz, což je horní limit lidského ucha s malou rezervou. //dp analiza zvuku

Frekvence menší jak 16Hz považujeme za infrazvuk, který používají k dorozumívání např. velryby. Frekvence nad 20kHz má ultrazvuk, jež vnímají například psi. Děje spojené s šířením zvuku, jeho vznikem a vnímáním nazýváme akustika a stejně tak nazýváme i vědu, která se zvukem zabývá [1].

1.1 Zdroje zvuku

Za zdroj zvuku považujeme předmět, nebo nějaký vymezený prostor, z kterého se do okolí šíří akustické vlnění šířící se nějakým hmotným prostředím (např.: vzduchem). Vzduch jakožto médium zvuku zprostředkovává spojení mezi jeho zdrojem a detektorem. Pod pojmem detektor si můžeme představit jak lidské ucho, tak například mikrofón, nebo jiný snímač. Zvuk se může šířit i kapalinami a pevnými látkami. Dokonalou zvukovou izolací je vakuum.

Moderním zdrojem zvuku je reproduktor. Nicméně může jím být jakýkoliv chvějící se předmět. Avšak o šíření zvuku do okolí nerozhodují jen vlastnosti jeho chvění, ale i fakt jestli je dobrým nebo špatným zářičem zvuku. Tato vlastnost je závislá především na jeho geometrickém tvaru [1].

1.2 Ucho jako spektrální analyzátor

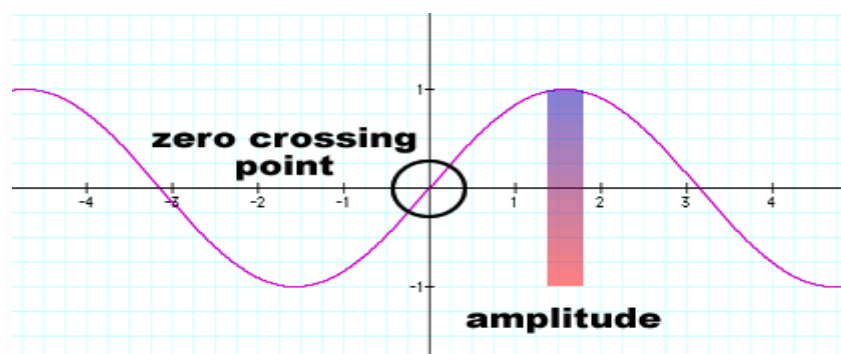
Funkce lidského ucha vlastně fungují na podobném principu jako elektrická zařízení využívající Fourierovi analýzy. Vnitřní ucho převádí mechanické vlny na elektrické impulzy, které popisují intenzitu zvuku v závislosti na kmitočtu. Podle Ohmova zákona náš mozek nezpracovává informace o pomalu se měnících zvucích, ale reaguje na změny v amplitudě a frekvenci [3].

1.3 Zvukové ukazatele vhodné pro zvukovou identifikaci

Kapitola definuje fyzikální jevy, které se vztahují k parametrům, jenž je využito v praktické části.

1.3.1 Zero-crossing rate (ZRC)

Používá se jako jeden z parametrů změny signálu v systémech rozpoznávající zvuky a hlas. Jedná se o frekvenci změny veličiny z kladné do záporné a naopak, tedy frekvence průchodu nulou.



Obrázek 1 ZRC [4].

1.3.2 Intenzita zvuku

Je definována jako zvuková energie dopadající na jednotku plochy za jednotku času, tedy akustický výkon na jednotku plochy, jednotkou je $[W \cdot m^{-2}]$.

1.3.3 Směrodatná odchylka

Používá se v matematické statistice a dalších vědních oborech, při zpracování informací. Informuje o tom, jak daleko jsou hodnoty rozptýleny či odchýleny od průměru a vypovídá o změně signálu.

2 ANALÝZA ZVUKU

Většinu kmitajících předmětů můžeme tedy považovat za zdroj zvuku, či hudebního tónu. Frekvencí tohoto tónu určujeme výšku.

Člověk už velice dávno zjistil, že kombinace určitých zvuků zní přirozeně a ladí k sobě. Začal komponovat hudbu a později se pokoušel tuto hudbu matematicky popsat. Vznikla řada teorií pro popis hudby a jedna z nich říká, že frekvence hudby musí být v poměru malých přirozených čísel – 1:2, 3:2, 4:3 atd. Tato teorie se nazývá přirozené ladění. V současnosti se však příliš nepoužívá pro svou nedokonalost – spočívající v množství tónu, které je při skladbě tóniny z větší části nevyužito. Mezi jednotlivými tóny jsou nepravidelné intervaly, takže transpozice do jiných tónin bývá velice komplikovaná stejně tak, jako konstrukce hudebních nástrojů.

Běžné zvukové signály jsou složeninami všemožných signálů o různých frekvencích. Abychom získali co nejpřesnější představu o tom, jak signál vypadá, potřebujeme oddělit jednotlivé frekvence od sebe pomocí spektrální analýzy. Při této metodě převádíme signál z časové domény do frekvenční. Tedy k vzorkům signálu o určitém čase přiřadíme informace o přítomnosti vybrané frekvence signálu.

Algoritmus pro výpočet frekvenčního spektra označujeme jako Fourierovu analýzu. Tato analýza nám umožňuje vytvořit grafickou podobu zvukového signálu [6].

2.1 Úvod do Fourierovy analýzy

Fourierova analýza nese jméno po svém objeviteli Jeanovi Baptistě Josefa Fouriera (1768-1830). Byl to francouzský fyzik a matematik, který se zasloužil o uvedení této teorie do praxe. Přišel s teorií, že libovolný spojitý periodický signál může být reprezentován součtem sinusových, nebo kosinusových průběhů.

Dnes víme, že těchto vln může být nekonečně mnoho, a proto v současnosti o jeho teorii nikdo nepochybuje.

Pojem Fourierova transformace rozdělujeme do čtyř kategorií v závislosti na několika druzích signálu, s kterými je možné se setkat.

1. **Neperiodický spojitý**- Obsahuje signály jako Gaussova křivka, či exponenciály.

Jde o signály, které sahají od záporného do kladného nekonečna. Přitom se však

neopakují žádné periodické vzory. Tato transformace se nazývá Fourierova transformace.

2. **Periodický spojitý** - Sem patří sinusovky, kosinusovky a vlastně jakékoliv vlny, které se stále periodicky opakují od záporného do kladného nekonečna. Tato transformace se nazývá Fourierova řada.
3. **Neperiodický diskrétní** – Tento druh signálu se neopakuje v žádných periodách, takovýto signál je definován jen v určitých diskrétních bodech mezi záporným a kladným nekonečnem. Tato transformace se nazývá Fourierova transformace s diskrétním časem (DTFT).
4. **Periodický diskrétní** – Tento signál se periodicky opakuje a je definován v určitých diskrétních bodech. Tato transformace se nazývá Diskrétní Fourierova transformace, nebo Diskrétní Fourierova řada (DFT).

Vzhledem k tomu, že se jednotlivé druhy transformací nahodile a nezávisle na sobě vyvíjely posledních zhruba 200 let, není zde uvedené rozdělení a pojmenování příliš výstižné. Navíc všechny tyto kategorie počítají se spojitým signálem nekončícím v čase. Mám-li signál digitální – nespojitý v čase ani amplitudě – neexistuje žádná varianta Fourierovi transformace, která by byla pro takovýto signál použitelná.

Nicméně, takovýto digitální signál lze upravit tak, aby vypadal jako nekonečný. Jedním ze způsobů, jak signál konečný učinit signálem nekonečným, je obklopit ho z obou stran nulovým signálem až do nekonečna. Tak ho můžeme považovat za signál Diskrétní a neperiodický a můžeme použít *Fourierovu transformaci s diskrétním časem*.

Signál lze také obklopit duplikáty sebe sama. Pak se z něj stává signál Diskrétní a periodický – použijeme Diskrétní Fourierovu transformaci.

Jedinou metodou použitelnou na digitální signál v počítačovém prostředí je Diskrétní Fourierova Transformace, protože digitální systémy mohou pracovat pouze s informací diskrétní a konečnou v čase.

Z těchto čtyř metod se první tři používají maximálně v řešení teoretických problémů, ale prakticky použitelná je jen poslední DFT.

Dále můžeme tyto varianty rozdělovat podle toho, jestli pracují s reálným, nebo komplexním signálem. Reálný signál je nejjednodušší, využívá reálnou algebru pro syntézu a dekompozici. Komplexní varianta je značně náročnější.

2.2 Rychlá Fourierova transformace (FFT)

FFT je nejefektivnější metodou výpočtu DFT. Předchozí varianty Fourierovy transformace počítají s goniometrickými funkcemi, ale FFT využívá postupného rozdělování vstupní posloupnosti na půlky až k dílkům o délce 1. Proto nejlepší výkon tento algoritmus podává při zpracovávání vstupní posloupnosti o délce 256, 512, 1024 atd. – obecně 2^n . Splňuje-li datové okno tuto podmínku, potom složitost výpočtu pro $\frac{N}{2}$ složek je $N \cdot \log_2 N$, protože spektrum je symetrické kolem poloviny.

Ve srovnání s DFT je toto velice výhodná varianta, nicméně zajímá-li nás jedna spektrální složka (např. základní frekvence), potom je pracnost výpočtu **DFT** rovna **N15**, což bývá výhodnější než při požití **FFT**. Při vypočítávání FFT je odvozování všech spektrálních částí provázáno, a tudíž nelze vypočítat jen jednu vybranou spektrální složku.

Nevýhodou FFT je, že pomocí něj lze detekovat pouze frekvence vyplývající z poměru vzorkovací frekvence a zvolené délky okna (2^n), proto je FFT nepoužitelné pro detekci frekvencí jednotlivých tónů hudební stupnice [7].

2.3 Akustický otisk zvuku

Jeho součástí jsou deterministicky generované informace ze zdroje zvukového signálu. Obsahuje souhrnné údaje, které je možné použít pro zvukovou identifikaci celku nebo k rozeznání shodných částí zvuku v databázi. V praxi se využívají pro identifikaci písní, melodií, reklam, rozeznávání hudebního obsahu digitálních nosičů a kontrole dodržování autorských práv.

Pomocí akustického otisku lze rozeznávat dvě nahrávky podobně jak tomu je v případě lidského sluchu. Dva zvuky mohou být vyhodnoceny jako shodné i v případě, že jejich bitová reprezentace je zcela odlišná. Akustické otisky nejsou bitové otisky, které se používají k porovnávání dokumentů, a proto musí být citlivé na jakékoliv změny v bitové reprezentaci. Akustické otisky jsou spíše analogické k lidským otiskům prstů, kde malé rozdíly u dvou porovnávaných vzorků jsou zanedbatelné.

Mezi často využívané charakteristiky akustických otisků patří zero crossing rate (ZRC), směrodatná odchylka, odhadované tempo, zvuková intenzita, frekvenční spektrum a šířka pásma.

Většina kompresních metod (AAC, MP3, WMA, Vorbis) zásadně pozměňuje bitové uspořádání ve zvukových souborech, aniž by výrazně ovlivnil způsob, jakým bude vnímána lidským uchem. Použití akustických otisků umožňuje, aby nahrávka uložená v kompresním formátu byla rozpoznatelná i v případě významného snížení její kvality. Metoda akustických otisků je vhodná i k rozpoznávání rádiového přenosu zvuku [8] [9].

3 BIOMETRIE LIDSKÉ ŘEČI

Klasifikátory na rozpoznávání izolovaných slov jsou ve středu pozornosti výzkumu od poloviny 20. století. Od začátků práce na této problematice byl učiněn velký pokrok, ale cesta k ideálnímu klasifikátoru, který by byl schopen bezpečně rozpoznat libovolného řečníka v libovolném stavu, s libovolným projevem je stále daleká. Jednou z největších překážek na tomto poli je výpočetní náročnost zpracování a s tím související omezené výpočetní prostředky, obzvláště v případech, kdy je očekávána okamžitá odpověď systému v reálném čase s dobou zpoždění maximálně do půl vteřiny od přijetí řečového signálu. Tato podmínka zásadním způsobem omezuje rozmezí použitelných metod. Faktory ztěžující zpracování a následné rozpoznání plynoucí ze strany přijímaného signálu jsou:

- Různorodost hlasu řečníků (tempo, barva hlasu, přízvuk, vada řeči . . .).
- Stav řečníka a situace, v níž se nachází (hlasitost, rychlost promluvy, nálada . . .).
- Šum, rušení a jiné zvukové jevy přítomné v pozadí řečového signálu.

3.1 Obecný model rozpoznávání řeči

V praxi bývá následující. Řečový signál je většinou nejprve zpracován některou metodou krátkodobé analýzy, poté je vyjádřena posloupností krátkodobých charakteristik (např. LPC, PARCOR, autokorelačních, keprstrálních koeficientů či výstupů pásmového filtru). Jednotlivé číselné údaje jsou nazývány příznaky, a jejich množiny obrazy. Samotné rozpoznávání probíhá rozřazováním obrazů do tříd, které představují slova ve slovníku (referenční). Aplikované metody rozpoznávání izolovaných slov lze shrnout do tří nejpreferovanějších přístupů:

- Slovo je zpracováváno jako celek. Srovnávání probíhá hledáním nejmenší vzdálenosti mezi obrazem rozpoznávaného slova a vzorovým obrazem. Vzdálenost bývá vyhodnocena metodou dynamického programování. Tato metoda spočívá v nalezení takové nelineární transformace časové osy, ve které je zmíněná výsledná vzdálenost nejmenší.
- Slovo je modelováno pomocí tzv. skrytých Markovových modelů. Slova jimi lze modelovat buď jako celky jediným modelem, anebo jsou modelovány subslovní jednotky, z nichž je celistvé slovo složeno. Rozřazení do tříd probíhá způsobem, kdy jsou stanoveny parametry každé třídy a slovo je umístěno do té, jejíž model je generován nejpravděpodobněji. Jde o statistické metody.

- Slovo je zpracováno ve dvou stupních. Akustickou analýzu následuje rozdělení řečového signálu na úseky, které jsou foneticky dekodovány. Druhým stupněm je rozpoznání podle posloupnosti úseků. Tato metoda bývá upřednostňována pro rozpoznávání souvislé řeči. Aplikovány bývají strukturální metody rozpoznávání [10].

4 ANALÝZA DOSTUPNÝCH APLIKACÍ POROVNÁVÁNÍ AUDIO ZÁZNAMU

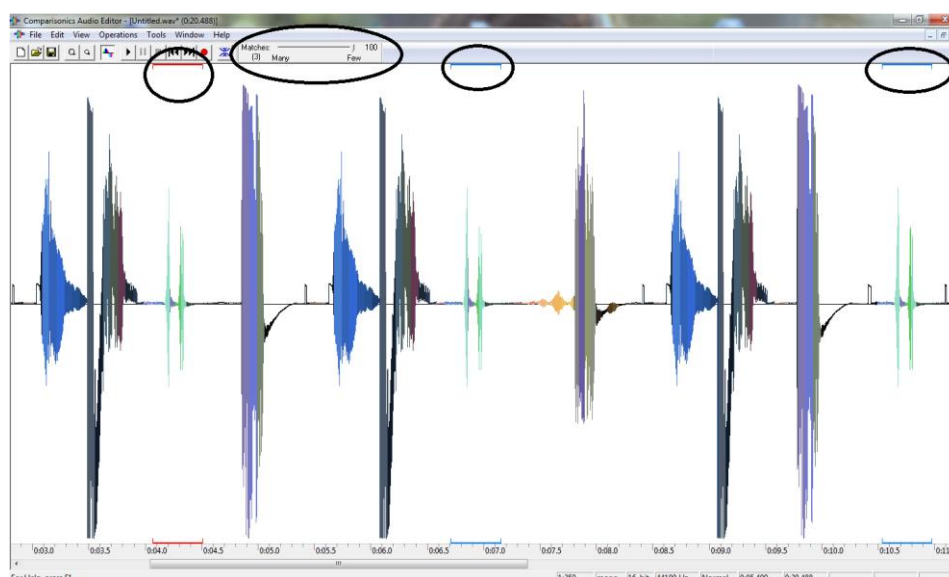
Kapitola obsahuje výčet nejzajímavějších dostupných softwarových řešení porovnávání audio souborů. Je členěna podle licenční dostupnosti a to na zdarma dostupné open-source aplikace a placené komerční aplikace.

4.1 Open-source aplikace

Kapitola věnována volně dostupným aplikacím podporující rozpoznávání zvukových stop.

4.1.1 Comparisonics

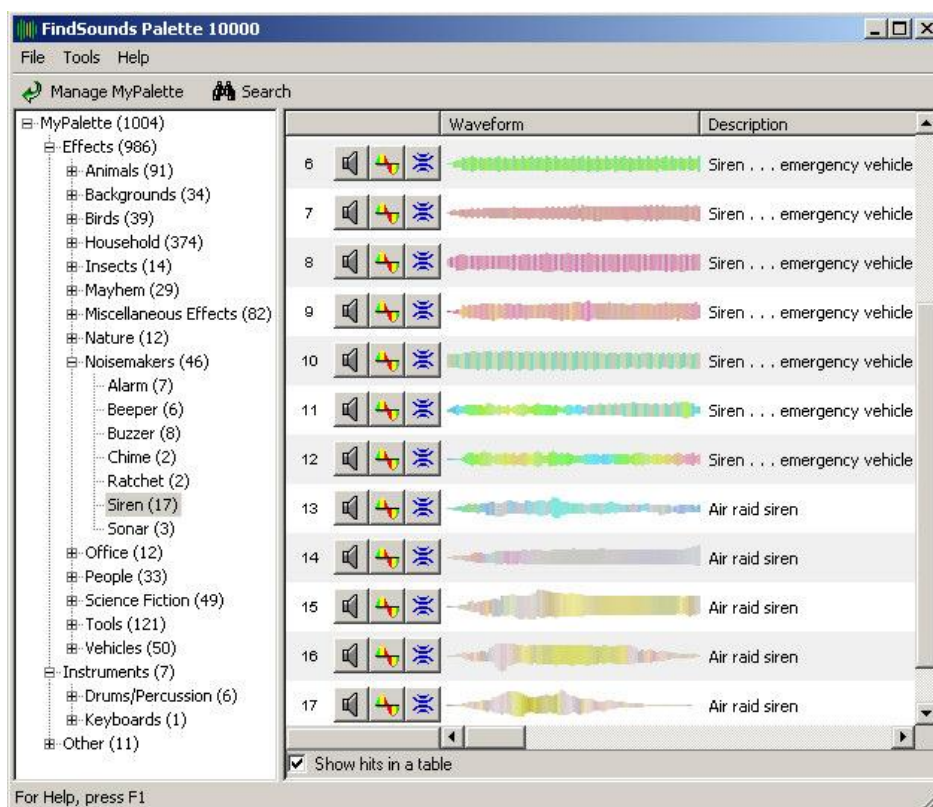
Comparisonics je aplikací s jednoduchým uživatelským prostředím umožňující základní analýzu zvukových souborů ve formátu wave. Comparisonics umožňuje uživateli přehrávat a upravovat zvukové soubory. Mezi editační funkce patří vyjmout, kopírovat, vložit, mazat a ovládat hlasitost. Umožňuje vkládat a upravovat metadata každého zvukového souboru. Uživatel může posouvat a zoomovat zobrazení zvukové vlny. Zvukové vlny se liší podle barvy tónu. Uživatel si může označit libovolnou část průběhu zvuku a tato část se graficky zvýrazní. Tuto část průběhu lze porovnat s celou délkou nahrávky a najít místa, která se shodují. Tuto míru shody můžeme regulovat posuvníkem, ovlivňující míru tolerance porovnávání. Regulace nám v důsledku ovlivní celkový počet výsledků. Program umožňuje také porovnávání zvukových průběhů zprostředkovanou službou FindSounds [11].



Obrázek 2 Comparisonics: nalezení shody ve zvukovém průběhu.

4.1.2 FindSounds

Současná verze placené distribuce pro OS Windows má označení „FindSounds Palette Version 2.3“, umožňuje rozpoznat a reorganizovat zvuky (nikoli hudbu) uložené na disku. Podporuje zvukové soubory WAVE, AU, AIFF nekompresní a mu-law kompresní formáty.



Obrázek 3 Hlavní okno FindSounds Palette [12].

<http://www.findsounds.com/palette.html>

Bezplatná webová služba FindSounds je určena k vyhledávání zvukových stop. Hledané stopy lze kategorizovat podle atributů: název, popis, kategorie, žánr, zdroj, autorská práva, formát, velikost, počet kanálů, rozlišení, vzorkovací frekvence a délky stopy. Databáze obsahuje celkem 250 000 unikátních zvukových nahrávek. Viz obrázek placené aplikace určené pro OS Android určené pro vyhledávání zvukových stop [12].

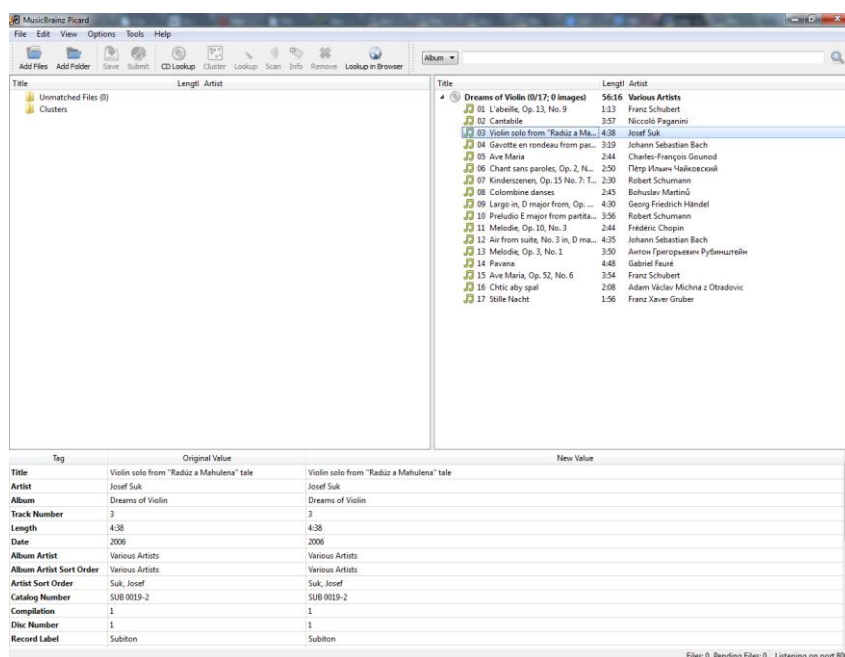


Obrázek 4 Androidová aplikace FindSounds [13].

FindSounds je velice užitečný nástroj pro vyhledávání zvuků prostředí, které jsou uplatnitelné při animaci nebo tvorbě virtuální reality, ale nepodporuje rozpoznávání hudby.

4.1.3 Picard

Picard je oficiální klientskou aplikací internetové hudební encyklopedie MusicsBrainz. Slouží k analýze obsahu hudebních kompaktních disků. Aplikace vytvoří otisky hudebních nahrávek, tyto otisky odešle porovnat s databází otisků na server MusicsBrainz. Používá se zejména pro identifikaci a doplnění metadat neznámých hudebních souborů.



Obrázek 5 Picard: identifikace hudebního kompaktního disku [14].

Picard v minulosti používal několik způsobů identifikace nahrávek. Všechny pracují na principu identifikace hudebních otisků. V prvním kroku se zvolí cesta ke zdroji zvuku. Za druhé, se ze surových zvukových dat vytvoří zvukový otisk, který se zašle na server. Následně server vygeneruje zvukový otisk a porovnává jej s databází zvukových otisků. Rozhodne, zda daný otisk je nebo není unikátní, pokud ano, vytvoří nové ID a přidá jej do databáze. MusicsBrainz dohlíží na unikátnost otisků tak, aby se v databázi netvořily duplicitní otisky a každý otisk je tedy unikátní [14].

TRM ID

Na straně klienta se používala "open source" knihovna zvukových otisků. Systém fungoval velice dobře pro rozpoznávání duplicitních souborů na místním systému, ale měl problémy s porovnáváním otisků umístěných na serveru. Tyto problémy způsobovaly kolize různých zvuků se stejným ID a duplicitu způsobené přidělením stejnému zvuku několika ID. Server navíc nebyl schopný vyhodnocovat velké množství požadavků, MusicBrainz nakonec upustila od využívání tohoto systému v roce 2008 [14].

PUID

PUID je druhým používaným systémem otisků zvuku, který vlastní od roku 2011 firma Gracenote. Po převzetí služby se však stává postupně neveřejnou službou a v současné době je z větší části již nefunkční. Tento systém byl již lepší než TRM, avšak stále měl několik hlavních problémů. Systém předání otisku není „open source“, tak jako takový nemohl být přidán do programu Picard. Proces snímání otisků je pomalý a to jak na straně klienta, tak i na straně serveru. Provozovatelé serveru se stále méně zajímají o chod služby a ta se stala téměř nefunkční [14].

4.1.4 AcoustID

Je třetím a zatím posledním systémem otisků MusicBrainz. Byl vytvořen a zveřejněn Lukášem Latinským v lednu 2011. AcoustID je open source projekt, který si klade za cíl vytvořit bezplatnou databázi zvukových otisků. Všechny součásti softwaru jsou open source, takže každý si může stáhnout zdrojový kód pro klientské knihovny, serverové aplikace a také samotnou databázi otisků [15].

Použití

1. Registrace na portále MusicBrainz.com.
2. Registrace nového API tokenu na Acoustid.org, zde použijete registraci z bodu 1.
3. Instalace aplikace MusicBrainzPicard.
4. V nastavení zadat přihlašovací údaje MusicBrainz a API token Acoustid.

Uživatelské prostředí je přehledné a rychle se v něm dá zorientovat. Velice užitečné jsou statistické funkce, které systém ukládá. V rámci vašeho účtu máte k dispozici informace o všech hledaných zvukových otiscích, které byly vyhodnocovány na serveru.

4.1.5 Audioscout

Aplikace je odolná proti hluku a změnám kódování i jiným druhům zkreslení. Může být použit pro celou řadu aplikací, včetně aplikací detekce duplicitních souborů, identifikaci (rozpoznávání) hudby, i v aplikacích hlídající zákonné používání zvukových nahrávek [16].

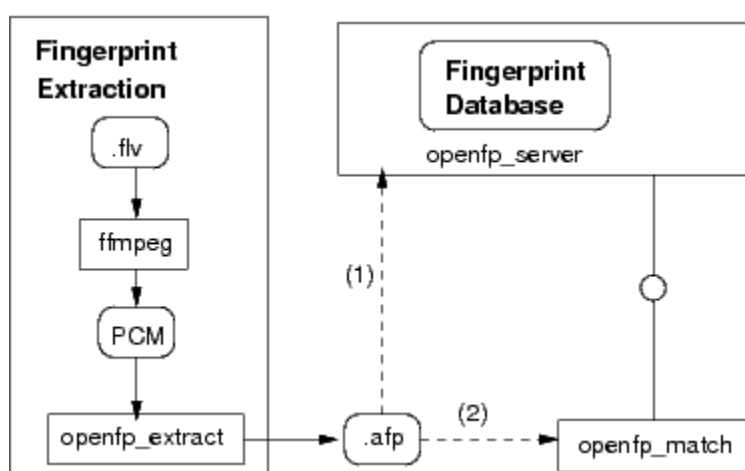
Vlastnosti:

- Přiřazení libovolné části zvukového souboru k originálu.

- Malé nároky na paměť.
- Pracuje s formáty WAV, MP3, OGG, FLAC, AIFC, ARM.
- Klient je dostupný pro všechny platformy, ale server je dostupný pouze pro Linux.

4.1.6 OpenFP

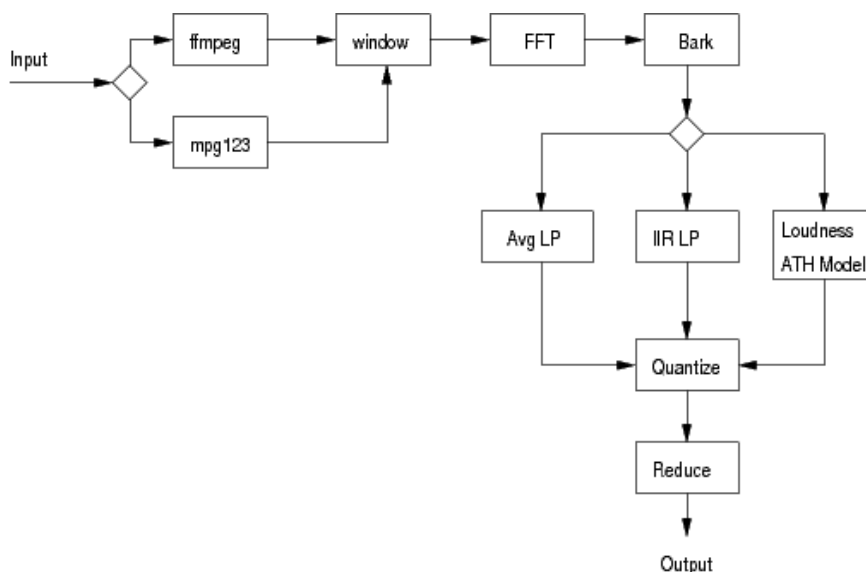
K analyzování hudby využívá akustické otisky vytvořených z hudebních nahrávek, otisky odešle na server OpenFP, kde se provede porovnání s referenční databází zvukových otisků. Následující obrázek znázorňuje vnitřní uspořádání systému OpenFP.



Obrázek 6 Schéma použitých nástrojů [17].

Funkce

Pro zpracování audio vstupu používá funkci libfftw3, která dokáže načíst zvuková data formátu PCM16. Následuje funkce ffmpeg s parametry "-f-u16le -acodec pcm_s16le", která je součástí hlavní funkce openfp_extract. Funkce ffmpeg podporuje audio i video soubory.

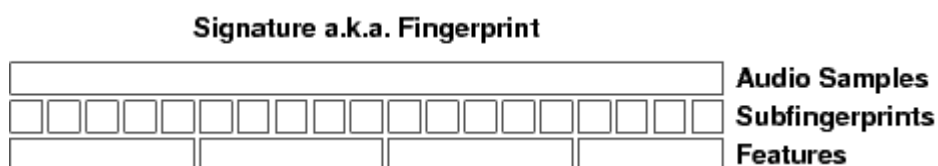
Získání otisku zvuku

Obrázek 7 Schéma znázorňuje extrakci akustických otisků [17].

Tvorba akustického otisku:

1. Na vstupu načítá prostřednictvím kodeku (např. ffmpeg) surová zvuková data.
2. Funkce Windows data převede do datového toku.
3. Aplikace upraví data Fourierovou transformací.
4. Ořeže a potlačí nedůležité části spektra.
5. Sníží hluk použitím dolní propusti, nejlepších výsledků bylo docíleno použitím dolní propusti typu IIR 12Hz bez snížení výkonu.
6. Kvantování dat.
7. Snižuje kvantitu, redukcí výstupních dat.
8. Na výstup posílá akustický otisk.

Výsledkem extrakce zvukového otisku je 32bit hash uložený ve výstupním souboru, reprezentující původní akustický soubor. Každý bit v 32 bitech hodnot popisuje spektrální výkon v určitém pásmu.



Obrázek 8 Postup při vzniku otisku [17].

Porovnání otisku

Shodu vyhodnocuje server (openfp_match), odpovídá na žádosti zaslané z klienta. Porovnávání probíhá ve dvou krocích:

- Porovnají se otisky a vyhledají shodné části
- Vyhodnotí míru shody otisku

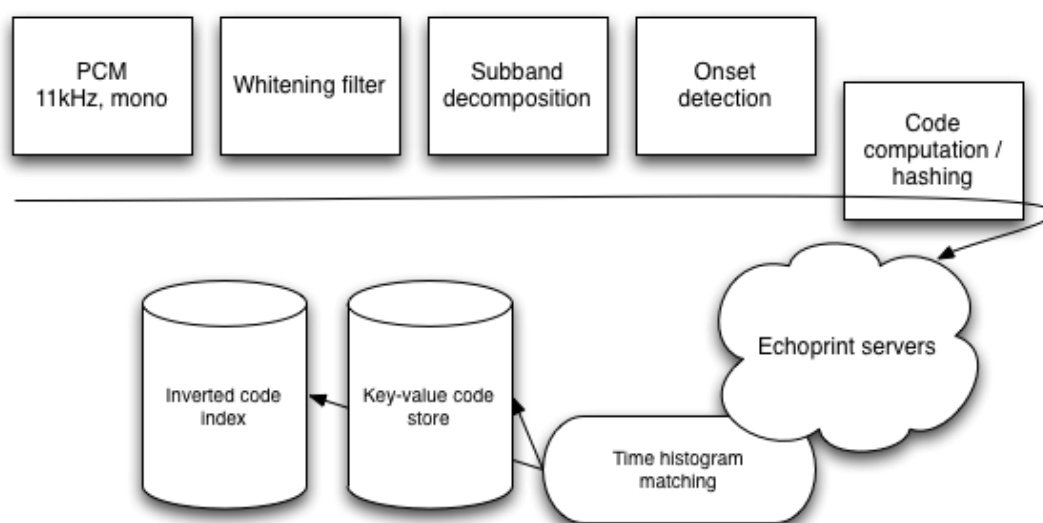
Shodné pozice jsou nalezeny porovnáním jednoho nebo více náhodně zvolených otisků z audio vzorků, které se porovnávají se všemi referenčními otisky. Dva vzorky jsou vyhodnoceny jako shodné podle kritéria Hammingovy vzdálenosti mezi dvěma otisky. Tato Hamingova hranice se pak používá jako minimální u všech dalších porovnávání, dokud není nalezen podobnější otisk [17].

4.1.7 Echoprint

Echoprint je open-source hudební identifikační systém, který umožňuje vytvořit otisky akustických nahrávek. Využívá databáze „The Echo Nest“ a je vyvíjen ve spolupráci s MusicBrainz.

Popis zpracování signálu

Echoprint načítá audio vstup na telefonu nebo v počítači, výsledky analýzy se snaží určit, o jakou nahrávku se jedná. Identifikuje i velmi hlučné verze původních nahrávek, poradí si i s nahrávkami mobilního telefonu špatné kvality a vysokým šumem. Výhodou je jeho open-source dostupnost [18].



Obrázek 9 Echoprint: schéma vnitřních funkcí aplikace [18].

Technická specifikace

Echoprint funguje na základě tří elementů. Generátor kódu převádí zvuk na akustický otisk, server je ukládá a indexuje všechny tyto akustické otisky. Třetí částí jsou referenční otisky akustických nahrávek, ty získává od svých partnerů a ostatních uživatelů Echoprint a databáze se tak neustále rozšiřuje.

Generátor kódu generuje otisk vždy s časovým údajem {čas, hash} ze zvukového signálu. Využívá pokročilých metod zpracování signálů, které eliminují okolní šum a hluk. Vstupní modul zpracovává 11kHz mono signál, signál se dále upravuje bělící transformací a na závěr je signál rozložen na 8 subpásem. Tím se informace o hudebním signálu rozloží do 20bitových úseků, které obsahují časové údaje. Posloupnost akustických úseků otisku je důležitá pro porovnávání s ostatními referenčními otisky [18].

4.1.8 libFooID

libFooID je open source knihovna umožňující tvorbu otisků akustických nahrávek. Gian-Carlo Pascutto vyvinul v roce 2006 aplikaci libFooID jako výzkumný projekt na univerzitě HogeschoolGent v Belgii.

4.2 Částečně open-source aplikace

Aplikace částečně zpoplatněné a nezapadající pod licenci open-source.

4.2.1 Last.fm

Celá knihovna Last.fm, včetně modulu, který generuje zvukové otisky je k dispozici na GitHub pod licencí LGPL. Uživatelé mohou vyzkoušet ukázkového klienta, který je k dispozici pro všechny platformy. Pracuje na principu Klient – Server.

4.3 Komerční aplikace

Kapitola věnována komerčně dostupným aplikacím podporující rozpoznávání zvukových stop.

4.3.1 AudibleMagic

Je komerční produkt, který poskytuje řešení pro správu elektronických identifikačních médií a autorských práv. AudibleMagic byla založena v roce 1999 za účelem rozvoje softwaru, určeného k identifikaci zvuku. Specializuje se na použití automatického rozpoznávání obsahu “AutomaticContentRecognition“ a synchronizaci medií v širokém spektru aplikací.

AudioId je komerční aplikace pro automatickou identifikaci audio materiálu pomocí akustických otisků. Audio data jsou automaticky rozpoznána a doplněna o související informace (název skladby, interpret, atp.) v reálném čase. Technologie byla vyvinuta v Fraunhofer Institute, kterou řídí Prof. Karheinz. Ten se proslavil jako šéf vývojář formátu MP3. AudioId je součástí mezinárodního formátu ISO/IEC MPEG-7 (Moving Picture Experts Group) [19].

Postup analýzy

Nejprve je vytvořen digitální podpis (otisk) ze vstupního zvukového souboru, ten je uložen v databázi. K podpisu jsou připojena další metadata s dodatečnými informacemi o skladbě jako jsou název či album. Jakmile je audio podpis vytvořen, AudioId může identifikovat shodu s originálním zvukem v databázi i když je vzorek pouze pár sekund dlouhý. Změna zvukového záznamu jako je míra šumu nebo kódování MP3 nemá vliv na kvalitu rozpoznávacího procesu. Míra úspěšného rozpoznání za normálních podmínek činí 99% [19].

Použití

Audio technologie rozpoznávání jako je AudioID se používají všude tam, kde je nutné, aby byl záznam rozpoznán a identifikován automaticky, např. v radiovém vysílání nebo v případě digitálních knihoven medií.

4.3.2 Content ID

V červnu roku 2007 začal YouTube zkoušet systém pro automatickou detekci nahraných videí, která porušují autorská práva. Tento systém se zdál být nezbytný z pohledu řešení řady soudních sporů. YouTube se tímto způsobem snažil bránit tvrzení, že profituje z pirátského obsahu. Systém, který se stal známý jako Content ID, vytvoří soubor ID pro audio soubor vložený vlastníkem autorského práva, a ukládá je do databáze. Když je video nahrané běžným uživatelem, je porovnáno s databází. V případě shody části videa s databází je takovéto video označeno pro podezření z porušování autorských práv. Pokud k tomu dojde, vlastník autorských práv může video blokovat, sledovat statistiky zobrazení a přidávat komentáře pod video [20].

4.3.3 Shazam

Funguje na základě analýzy zachyceného zvuku a hledá shodu založenou na akustickém otisku v databázi více než 11 miliónů skladeb. Aplikace zaznamená 10 sekund dlouhý úsek skladby a vytvoří akustický otisk. Proběhne vyhledání shody v databázi. Pokud je nalezena shoda, informace je vrácena k uživateli, v opačném případě vrátí chybu. Podporuje rozpoznávání z libovolného zdroje jako je rádio, kino nebo živá hudba. Za předpokladu udržení nutného odstupu signálu a šumu [21].

Vlastnosti:

- Aplikace je dostupná v Android verzi a verzi pro Windows 8.
- Ve verzi Lite umožňuje zdarma 5 porovnání měsíčně.
- Kromě hudby rozpozná i UK TV pořady a reklamy.

4.3.4 Gracenote

Gracenote provozuje internetovou databázi obsahující informace o obsahu kompaktních disků a gramofonových desek. Tato databáze byla původně vytvořena s finanční pomocí uživatelů. K nelibosti uživatelů se ale nakonec Gracenote stal komerčním nástrojem.

Gracenote umí identifikovat skladby na kompaktním disku i skladby uložené v digitální podobě na pevném disku ve formátu MP3. Podporuje nástroje pro správu medií, vytváření playlistů a doporučení hudby na základě vyhodnocení oblíbené hudby [22].

Nástroje Gracenote nachází své uplatnění v produktech velkého spektra společností:

- Online služby: Yahoo! Music Jukebox, AOL, AmazonMP3, Spotify, Winamp, MetroLyrics, Pandora, Google Music, a Tuneup Media.
- Automobilový průmysl: Alpine, Bose, Panasonic, Philips, Loewe a Sony.
- Mobilní hudební aplikace: Samsungu, Sony Mobile Communication (TrackID), KDDI (Japonsko), KTF (Korea), Musicwave (Evropa).

4.4 Závěr analýzy dostupných softwarových nástrojů

Teoretická část popsala nástroje vhodné pro rozeznávání a porovnávání zvukových stop. Vybrala některé dostupné knihovny jako je libFooID vhodné pro vývojáře, kteří hledají vhodný nástroj pro porovnávání zvuku. Veškeré popisované aplikace byly vybrány tak, aby práce obsáhla celé spektrum softwarových modifikací používaných pro různé účely, ať už ze segmentu open-source nebo komerčních aplikací. Od nejjednodušších jako je nástroj Comparison vhodný pro zpracování základní analýzy vlny, který své uplatnění může nalézt např. při výuce. Po sofistikovanější nástroje jako jsou FindSound, Picard, AcousticID a další aplikace, vhodné pro identifikaci souborů na disku a práci s metadaty zvukových souborů. Současně byla popsána technologie ContentID, kterou používá internetová služba YouTube ke kontrole k dodržování autorských práv nahrávaného obsahu. Některé aplikace může uživatel využívat prostřednictvím svého smartphonu jako jsou FindSounds a Shazam. Tyto aplikace umožňují pořizování a identifikaci nahrávek ať se uživatel nachází kdekoliv.

Je patrné, že trendy vývoje tohoto softwarového odvětví ovlivňuje již delší dobu nástup moderních smartphone technologií, které jsou schopné používat sofistikované softwarové nástroje pro záznam a vyhodnocení zvuku pořízeného prostřednictvím zabudovaného mikrofону. Používání je uživatelsky velice přívětivé a nic nebrání dalšímu růstu tohoto segmentu. V budoucnu se předpokládá masivnější využívání podobných aplikací, hlavně v komerčním segmentu. Aplikace se stanou hlasově ovladatelné a hlas bude sloužit i jako způsob autentizace uživatele.

II. PRAKTICKÁ ČÁST

5 REALIZACE SOFTWAREVÉHO ŘEŠENÍ TŘÍDĚNÍ ZVUKU

V přípravné fázi projektu se zvažovaly různé možnosti jak realizovat aplikaci, která by měla být schopná rozpoznat různé zvuky. Během prvotního experimentování bylo zjištěno, že je nutné blíže specifikovat druhy zvuků, které si práce klade za cíl detekovat. Byly objeveny dvě metody, které upřesnily, jakým způsobem je softwarově možné zvukové stopy rozeznávat a to podle jejich délky a komplexnosti. Tyto metody mohou být určeny pro:

- Krátké prosté zvuky
- Komplexní zvukové nahrávky

Krátké prosté zvuky, které lze přirovnat k jednotlivým tónům. Je možné detekovat metodou postupného analyzování klíčových zvukových parametrů, od začátku do konce stopy. Jednotlivé naměřené hodnoty se porovnávají se zvolenými rozsahy klíčových parametrů a v případě, že se v určitém místě stopy najde takové místo, které splňuje všechny kritéria. (tedy naměřené hodnoty jsou v intervalu těchto parametrů) Potom je možné tuto zvukovou stopu prohlásit za typ zvuku, s jehož parametry se stopa porovnávala.

K rozpoznávání komplexních zvukových nahrávek jako jsou hudební soubory se používá metoda akustického otisku zvuku. Z důvodu použití první zmíněné metody se jí již práce nebude více věnovat.

Při návrhu aplikace určené k rozeznávání krátkých zvukových nahrávek se zpočátku experimentovalo s metodou analýzy akustického otisku. Tyto akustické otisky úspěšně rozeznaly dva soubory různé kvality, stejné hudební skladby. Avšak během testování se došlo k závěru, že je tato metoda nevhodná pro porovnávání krátkých zvuků z důvodu časté chybné detekce. Proto byla zvolena metoda analyzování čtyř základních parametrů, tato metoda je sice primitivnější, ale pro účely této aplikace je naprosto dostatečná. Parametry zvolené pro tento druh analýzy jsou tyto:

- Intenzita [W/m^2]
- Frekvence [Hz]
- Směrodatná odchylka
- Parametry pásmové propusti
- Zero Crossing Rate (ZRC)

5.1 Musicg API

Musicg API je open-source knihovna napsaná v jazyku Java pro účely výzkumu a vývoje. Rozhraní umožňuje vývojářům extrahovat zvukové charakteristiky. Metody knihovny zpracovávají zvuková data, podporují funkce čtení, ukládání, ořezávání a posouvání. Také obsahuje nástroje pro zpracování výstupu jako je generování spektrogramu. Má velký potenciál v možných uplatněních i díky své kompatibilitě s OS Android. Podporuje velké množství zvukových analýz vhodných pro porovnávání krátkých zvukových stop, jako jsou analýzy směrodatné odchylky, ZRC, frekvence a intenzity. To jsou některé z analýz, které byly použity při vývoji aplikace.

Zpracovává zvukové analýzy na nejnižší úrovni a poskytuje výstupy pro porovnávání. Má potenciál uplatnění v celé řadě dalších aplikací zpracovávajících zvukové soubory nebo zabývajících se tvorbou grafických výstupů, jako je třeba vizualizace spektrogramu a časových průběhů veličin. Mezi další funkce patří extrakce (záhlaví, spektra, robustnosti), která umožňuje generování a porovnávání akustických otisků zvukových souborů.

Rozhodlo se použít API Musicg z důvodů její dostupnosti a také z hlediska dobré funkčnosti a přehlednosti kódu a přehledné dokumentaci. Přínosné mi byly i zkušenosti uživatelské komunity.

5.2 Definice úkolu

V praktické části má práce za úkol navrhnout a vytvořit testovací Java aplikaci, určenou k identifikaci krátkých a prostých zvukových signálů. Aplikace umožní vyhodnocení zvukových analýz, které budou vytvářeny jádrem, které bude tvořit modifikovaná open-source knihovna Musicg API.

Navrhnout a zhodnotit aplikaci následující konstrukce: aplikace nahraje zvuk a analyzuje jej s přednastavenými zvukovými charakteristikami (parametry) a určí shodu s těmi, které se svými parametry nacházejí v rozsahu všech definovaných parametrů určitého typu zvuku.

Při vývoji aplikace je kladen důraz na fakt, že se jedná o výzkumný projekt, který by měl umožnit flexibilní modifikovatelnost aplikace k rozpoznávání libovolně zvoleného prostého zvuku. Pro implementaci otestování byly zvoleny tři základní prosté zvuky, které jsou (pískání, tleskání, ťukání). Při konstrukci bylo dbáno na dosažení udržení modifikovatelnosti. V GUI se implementovaly funkce ulehčující identifikaci klíčových

akustických parametrů, sloužících pro potřebu sběru a identifikace parametrů, které se používají jako referenční hodnoty u definovaných zvuků tzv. referenční vzorky.

5.3 Ranné fáze návrhu

První přístup

Postup, který byl stanoven, spočíval v implementaci akustických otisků pro účely porovnávání. Během testování bylo umožněno porovnávat velké množství souborů a identifikace fungovala velice dobře. Avšak tento postup porovnávání se neosvědčil při porovnávání krátkých a proměnných zvuků, jako je písknutí nebo tlesknutí. Během tohoto pokusu bylo použito, za prvé referenčního otisku ze souboru a otisku z nasnímaného zvuku (tlesknutí), avšak aplikace nebyla schopna zvuky identifikovat. Dokonce ani pískání nebo tleskání vykonávané po delší dobu úspěšnost nezlepšilo. Dílčího úspěchu bylo dosaženo při identifikaci ticha.

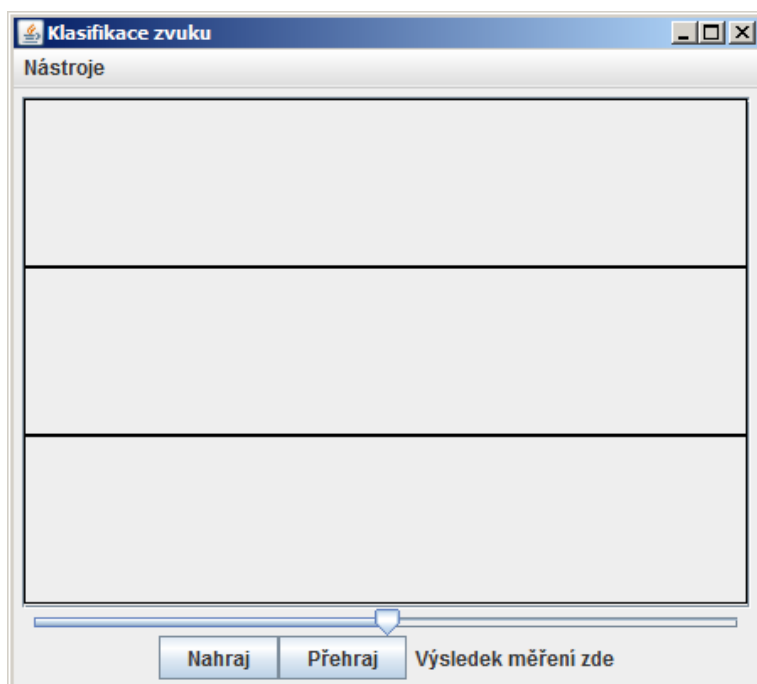
Druhý přístup

Podstatou druhého přístupu je zvolení si explicitních parametrů, které aplikace bude vyhledávat v zaznamenané zvukové stopě. Explicitně byly zvoleny charakteristické vlastnosti zvukové vlny. Jsou jimi frekvenční rozsah, rozsah intenzity, rozsah směrodatné odchylky a ZRC. Na základě těchto definovaných veličin je aplikace schopná určit kriteria podobnosti. Při splnění těchto kritérií jsou zvuky prohlášeny za podobné/shodné. Tento přístup se v praxi osvědčil, ale je nutné najít a zvolit si potřebné rozsahy parametrů experimentálně.

Testovací verze aplikace podporuje detekci krátkých zvuků a také dokáže pomoci při experimentálním měřením hledaných parametrů.

5.4 Testovací aplikace Klasifikace zvuku

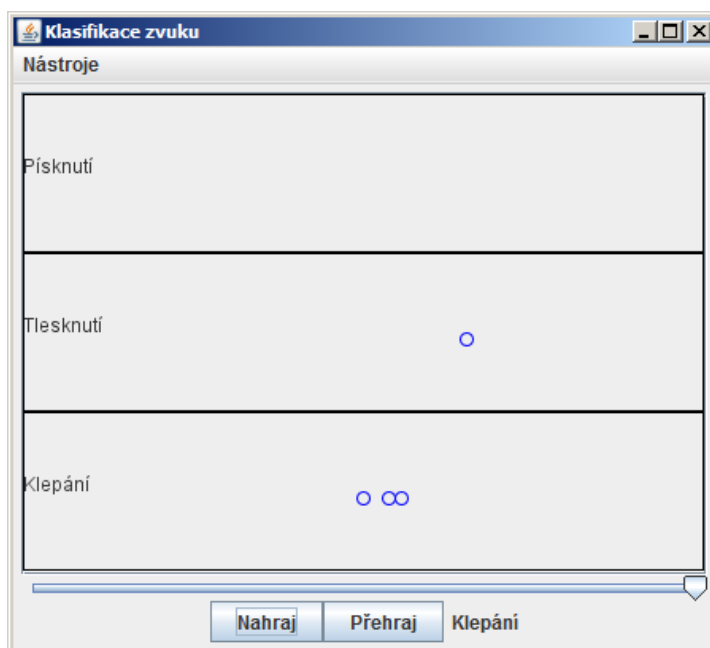
Po spuštění aplikace se zobrazí hlavní okno uživatelského rozhraní. Nacházejí se zde tlačítka "Nahraj" a "Přehraj", panel zobrazující body shody, menu „Nástroje“ v horní liště obsahující záložku „Analýza“. Posuvník zobrazuje průběh přehrávání, není určen k posouvání přehrávaného zvuku. Vpravo dole je pole určené k zobrazování výsledků měření. Viz. Obrázek.



Obrázek 10 Hlavní okno testovací aplikace

Po stisku tlačítka „Nahrávání“ aplikace spustí nahrávání zvuku a také proběhne změna textu tlačítka „Stop“, to nabádá k další akci. Po stisku tlačítka „Stop“ aplikace analyzuje a identifikuje zvuk. Výsledek je zobrazen místo textu „Výsledek měření zde“.

Na panelu nad posuvníkem se zobrazuje na rozdíl od nejpravděpodobnějšího výsledku zobrazeného v dolní části okna, body shody v zobrazené chronologicky na ose x zvukového záznamu, informace o bodě shody se zobrazí v bublině. Pro zobrazení detailní informace stačí umístit kurzor myši nad bodem shody, označený modrým kruhem. Viz. Obrázek níže. Hlavní okno reprezentuje hlavní myšlenku tohoto projektu, je jeho hlavním výstupem.



Obrázek 11 Zobrazení výsledku měření

5.4.1 Zvuková identifikace

Zpočátku bylo nutné identifikovat parametry, které budou pro rozpoznávání klíčové. Každý z těchto parametrů je interval, u kterého je třeba určit dolní mez a horní mez. Hodnoty v rámci těchto intervalů reprezentují charakteristický rozsah pro daný zvuk. Měřené parametry jsou:

- Intenzita
 - minIntenzita
 - maxIntenzita
- Směrodatná odchylka
 - minSmerOdchylka
 - maxSmerOdchylka
- Frekvence
 - minFrekvence
 - maxFrekvence
- Pásmová propust
 - highPass
 - lowPass
- ZRC
 - minPocetPruchoduNulou

- maxPocetPruchoduNulou

Parametry highPass a lowPass aplikace používá při implementaci pásmové propusti na zvukový signál. Pro výpočet pásmové propusti se zde využívá Shannonův teorém, podle kterého je volena vzorkovací frekvence dvakrát větší než je požadovaná přenášená frekvence.

Kombinace výše uvedených faktorů se dohromady podílí na identifikování zvukové stopy. Pokud nahraný zvuk splní parametry těchto intervalů je identifikován a klasifikován jako druh zvuku, kterému náleží referenční parametry a tento výsledek je zobrazen.

5.4.2 Zvolené Charakteristické parametry

Parametry uvedené v tabulce byly zvoleny experimentálně měřením pomocí programu Audacity a funkcemi knihovny Musicg. Dále byly upravovány během testování aplikace, za účelem odstranění anomálií a celkové zlepšení detekčních charakteristik, metodou ad-hoc. Uvedené parametry jsou definovány ve třídách „KlepaniAction.java“ „PiskaniAction.java“ a „TleskaniAction.java“. Parametry těchto tříd jsou uvedeny v tabulce:

Tabulka 1 Charakteristické parametry zvolené pro vyhledávání

Parametr	„Písknutí“	„Klepání“	„Tlesknutí“
minFrekvence	500.0f	0.0f	300.0f
maxFrekvence	5000.0f	43.0f	3000.0f
minIntenzita	1700.0f	8200.0f	10000.0f
maxIntenzita	100000.0f	300000.0f	800000.0f
minSmerOdchylka	0.06f	0.0f	0.0f
maxSmerOdchylka	0.08f	0.05f	0.03f;
highPass	100	100	100
lowPass	10000	10000	10000
minPocetPruchoduNulou	50	0	50
maxPocetPruchoduNulou	200	50	200

Aplikace nepoužívá metody určené k detekci písknutí jak tomu je u knihovny Musicg API. V Musicg API existuje třída „WaveTypeDetector.java“ a její metoda ‘isWhistleProbability()’, navzdory tomu nebyla nevytvořena metoda „isClapProbability()“ uvnitř této třídy a to ze dvou následujících důvodů:

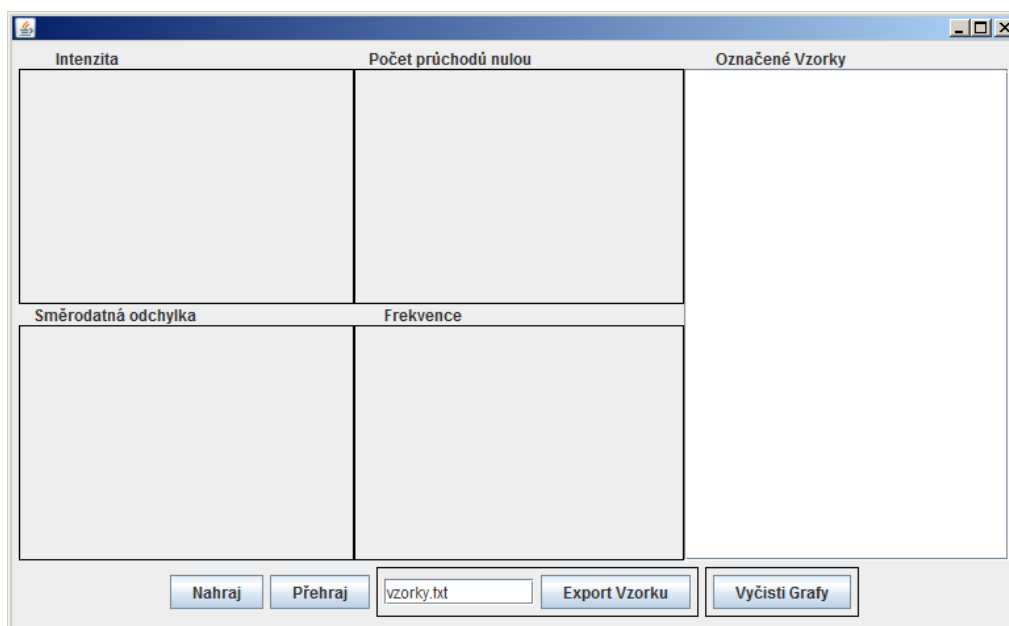
1. Třída by se časem stala po přidání většího počtu metod typu „isSomethingProbability()“ nepřehledná. Každá přidaná metoda nebo úprava parametru by si vyžádala modifikaci této třídy, což může vést ke vzniku chyby.
2. Místo toho aby aplikace rozhodla, zda se jedná o jeden konkrétní zvuk (Pískání, Klepání, Tleskání), umožňuje iteraci hlavní metody třídy 'DetektorAnylyzaZvuku' pro parametry různých typů zvuku (Action). Dle celkového počtu zvuků (Action). Moje řešení snížilo složitost kódu, aplikace místo toho, aby určila pouze jeden nejpravděpodobnější zvuk, je schopná detekovat různé zvuky během jednoho měření a zobrazit je v jedné grafické prezentaci.

5.5 Analýza parametrů zvuků a jejich implementace v aplikaci

Aplikace je konstruována, tak aby její používání umožnilo flexibilně upravovat a vkládat nové druhy zvuků podle současné potřeby uživatele. Zpracováním surových zvukových dat by bylo velice časově složité a zdlouhavé. Pro usnadnění získávání těchto dat bylo do aplikace přidáno další okno zajišťující analýzy zvuku. Získání parametrů se neváže na žádný další software a všechny neznámé hodnoty je možné naměřit v rámci této aplikace. Nástroj pro Analýzu a extrakci klíčových parametrů zvuku se nachází v menu „Nástroje“ položka „Analýza“.

K nalezení všech klíčových parametrů je nezbytné experimentovat. Experimentálním měřením hledáme specifické parametry, které jsou pro každý zvuk unikátní. Při analyzování zvukového průběhu hledáme ty části, které nejlépe reprezentují daný zvuk. Aplikace umožňuje nasbíraná data označit a uložit pro potřeby dalšího zpracování.

Nově získané parametry následně zaneseme do zdrojového kódu aplikace, každá úprava nebo přidání nové funkce detekce zvuku vyžaduje úpravu zdrojového kódu. Tyto změny jsou dále specifikovány v dokumentaci zdrojového kódu.



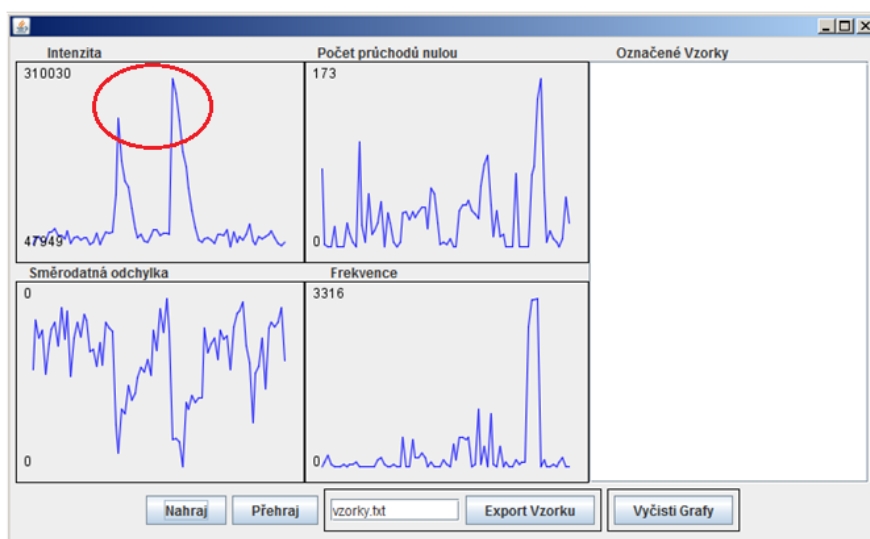
Obrázek 12 GUI Analýza

Okno Analýza zprostředkuje uživateli tři základní funkce. Těmi to funkcemi jsou:

- Záznam a přehrávání zvuku
- Grafickou vizualizaci klíčových parametrů
- Export klíčových parametrů do souboru

Nacházejí se zde dvojice tlačítek sloužící k záznamu a přehrávání (se stejnou funkcionalitou jako v hlavním okně). Čtveřice grafů, seznam vybraných klíčových parametrů, funkce exportu do souboru a funkce mazání grafů.

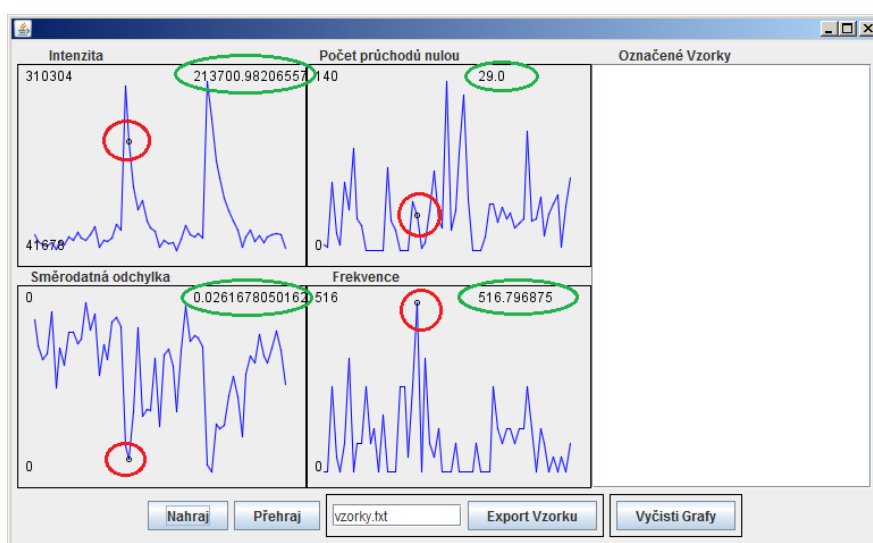
První krok je nahrání zvuku. Začátek snímání se inicializuje stiskem tlačítka „Nahraj“ a ukončí tlačítkem „Stop“. Následně se zvuková stopa zobrazí jako grafický průběh čtyř grafů všech klíčových parametrů. Viz obrázek níže.



Obrázek 13 Zobrazení grafů

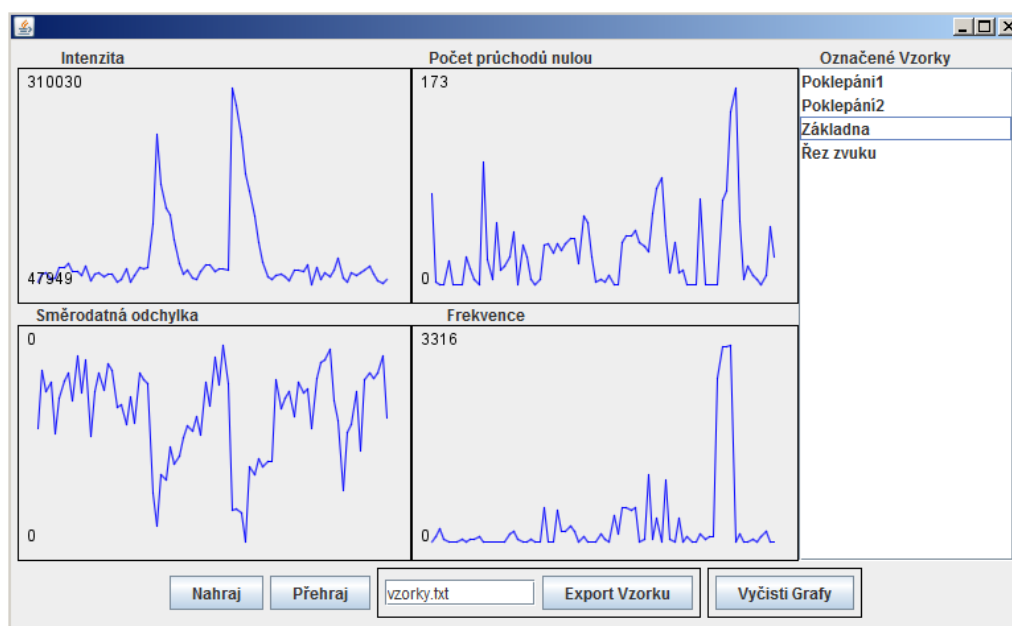
Dva vrcholy v prvním grafu tj. graf Intenzity zvuku reprezentují dvě ťuknutí do stolu. Čísla zobrazená na levé straně každého grafu pak popisují rozsah osy y. Posunutím myši do kterékoli části grafu způsobí zvýraznění bodu se stejnou x souřadnicí u všech grafů zároveň, navíc v pravo nahoře každého grafu dojde k zobrazení číselné hodnoty konkrétního parametru. Místo s výskytem typických parametrů pro daný zvuk nalezeneme po přehrání, kdy se na časovém průběhu pohybuje bod, lze tedy odpozorovat přibližné místo výskytu měřeného zvuku.

Pomocí posunování myši po průběhu můžeme snadno určit rozsahy pro všechny čtyři klíčové parametry.



Obrázek 14 Hromadné zobrazení označených hodnot

Užitečnou funkci, kterou poskytuje modul „Analýzy“ a pomáhá urychlit a zefektivnit proces identifikace parametrů, je možno označit vybrané místa jako řez. Ten lze exportovat do souboru pro účely dalšího zpracování. Stačí kliknout na požadované místo na jednom z grafů. Sejmутý řez je přidán do seznamu a zobrazen v pravé části okna. Řezy je možné přejmenovat dvojitém kliknutím v seznamu. Po sesbírání požadovaného počtu řezů, mohou být exportovány do souboru kliknutím na tlačítko „Export Vzorku“. V kořenovém adresáři aplikace se vytvoří soubor „Vzorky.txt“, ten obsahuje záznamy o všech exportovaných řezech.



Obrázek 15 Seznam řezů

The screenshot shows a text file named 'vzorky - Poznámkový blok' with the following content:

```

Základna1
-----
Intenzita      = 73276.64785750791
Směrodatná odchylka = 0.142510097373445
Počet Průchodů Nulou= 2
Frekvence     = 0.0

Základna2
-----
Intenzita      = 57602.023140030484
Směrodatná odchylka = 0.07460543970880802
Počet Průchodů Nulou= 84
Frekvence     = 387.59765625

Poklepání1
-----
Intenzita      = 247183.11067670665
Směrodatná odchylka = 0.01852274696256753
Počet Průchodů Nulou= 35
Frekvence     = 0.0

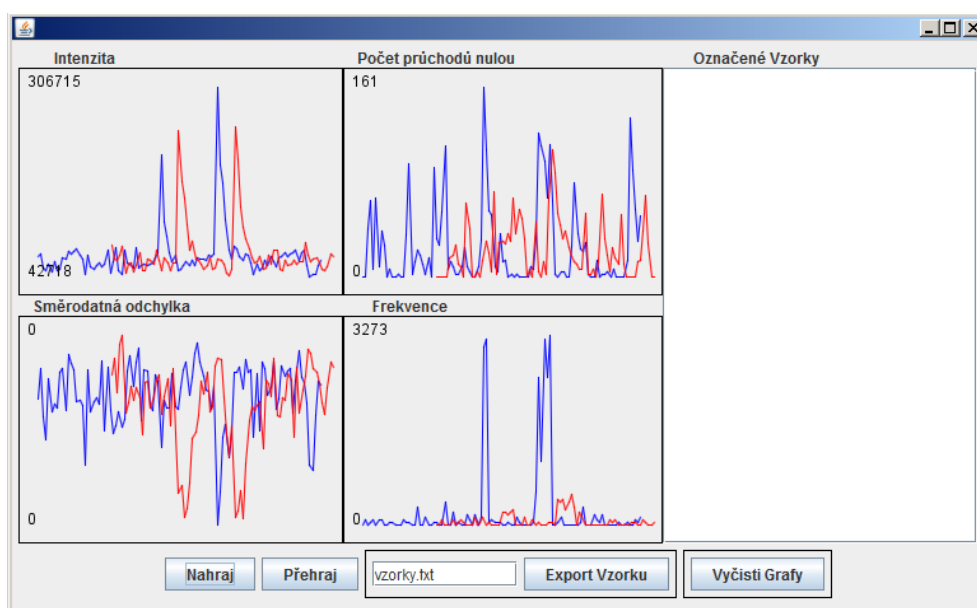
Poklepání2
-----
Intenzita      = 310030.0222319172
Směrodatná odchylka = 0.03228084773812124
Počet Průchodů Nulou= 36
Frekvence     = 559.86328125
  
```

Obrázek 16 Exportovaný soubor

5.6 Úprava výsledků měření

Pro určení klíčových rozsahů pomocí modulu „Analýzy“ je vhodné proces měření několikrát zopakovat, z těchto měření spočítat průměrnou hodnotu, která minimalizuje chybu měření. Další metoda, která byla použita pro vylepšení detekčních vlastností aplikace, je metoda posunutí mezních parametrů různých druhů testovaných zvuků (Action) od sebe tak, aby se minimálně překrývaly. Tímto způsobem lze upravit hodnoty parametrů tak, aby se později při použití dat pro rozpoznávání dařilo předcházet nesprávnému rozpoznání zvuku.

Například parametry „třukání“ mohou být vylepšeny porovnáním se zvukovým průběhem „tleskání“. Korekce parametrů lze provést za podmínky zachování detekční schopnosti zvuku, je nutné schopnost detekce ověřit zkouškou.



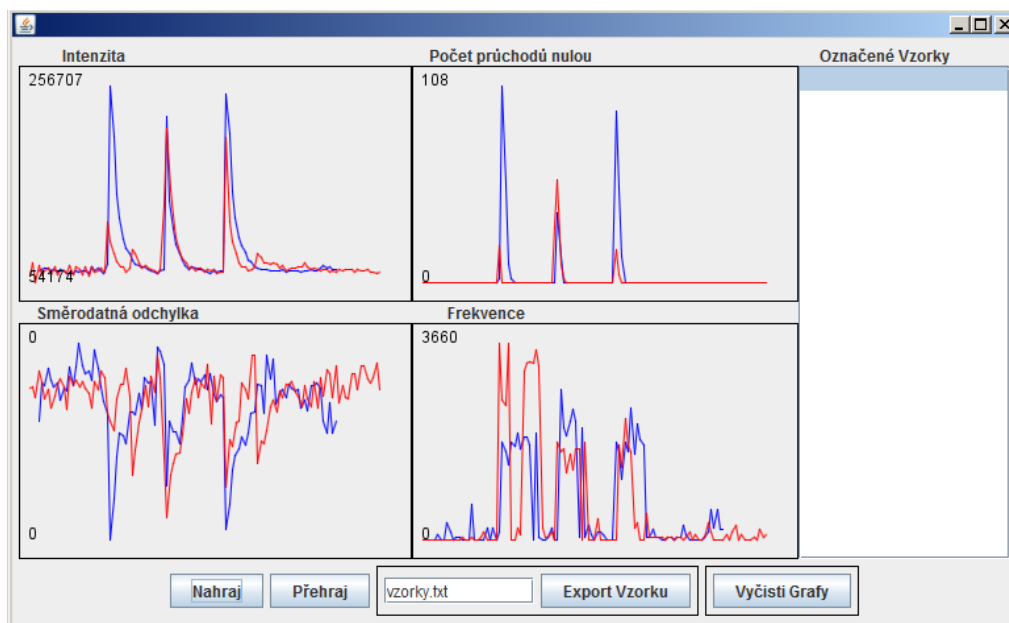
Obrázek 17 Dva průběhy v jednom grafu

Každý analyzovaný průběh se zobrazí v grafech pomocí odlišné barvy.

Grafy na obrázku obsahují dva průběhy „klepání“ (červený) a tleskání (modrý). Zřetelně je vidět, že frekvence klepání je velmi nízká. Směrodatná odchylka je vždy menší než 1 a ZRC nikdy nepřekročí 100.

Důležité jsou pouze hodnoty naměřené od základny po vrchol vlny, v této fázi je zvuk zaznamenán čistěji než ve fázi od vrcholu, kde se mohou objevit různé dozvuky a ozvěny, které zkreslují charakteristiku. Pro lepší porovnání kterékoli části vlny je možné je

libovolně posunovat po ose x (tažením po směru osy) tak, aby došlo k jejich ideálnímu překrytí vrcholů a tak graficky porovnat všechny parametry. Viz obrázek níže.



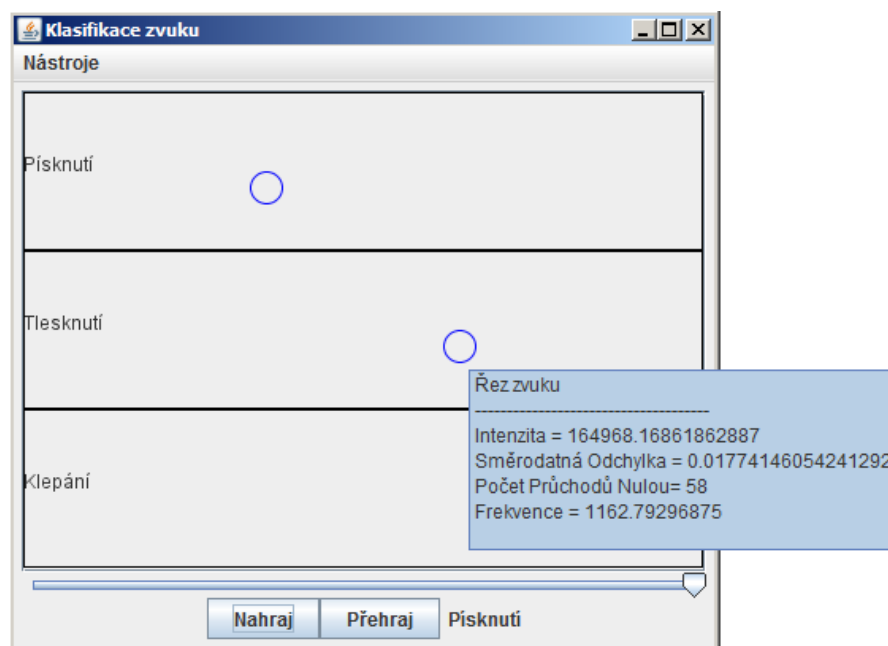
Obrázek 18 Proložení obou změřených průběhů

Jakýkoliv počet vln může být porovnán tímto způsobem, i když se barvy průběhů začnou opakovat.

5.7 Redukce falešných rozpoznání

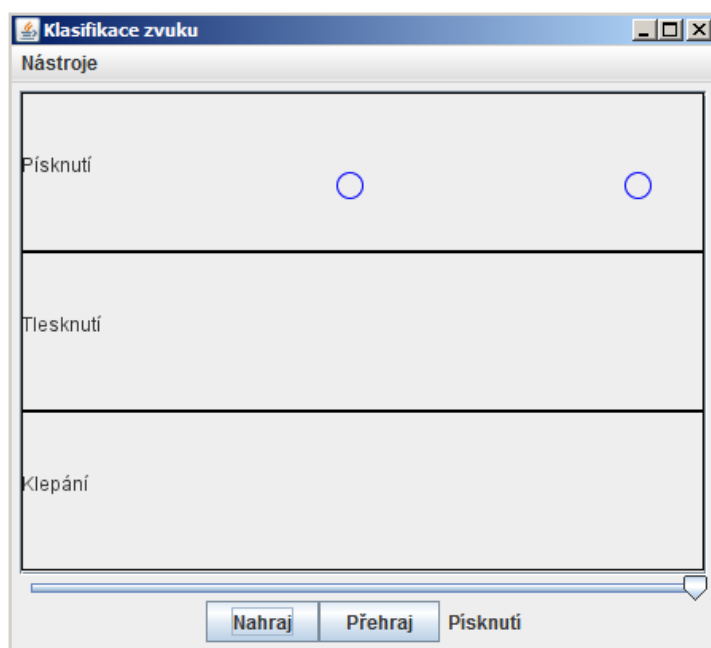
Při vyhodnocování vlny může dojít k chybě. Předpokládejme, že písknutí by mohlo být identifikováno jako ťukání. K odladění chybných identifikací je užitečné mít k dispozici parametry, které program identifikoval jako určující v daném případě. A proto byla do programu zahrnuta funkce zobrazení informace o parametrech v místě, kde byla shoda detekována. Informace se zobrazí při posunutí myši nad bod shody (označený modrým kruhem). Viz obrázek níže.

Při pokusu identifikovat písknutí se může zobrazit tento výsledek:



Obrázek 19 Informace o řezu

Za všimnutí stojí, že i když je výsledek opravdu „písknutí“, bylo také navíc detekováno „tleskání“. Při kontrole parametrů v tomto řezu zjistíme, že dané místo vlny se shoduje s kriteriem detekce „tlesknutí“ i když dříve bylo v signálu detekováno také „písknutí“. Naším úkolem je najít parametr, který se vyskytuje u své definované hranice. Jeho posunutím se pravděpodobně vyřeší problém dvojité detekce za cenu nepatrného zhoršení detekce samotného „třukání“. V tomto případě je vhodné změnit rozsah parametru „ZRC“ z nastavených 50 až 200 na 60 až 200 a vše znovu odzkoušet.



Obrázek 20 Nový výsledek

Obdobný postup může být použit pro zúžení i ostatních hodnot, do doby než je zachována vysoká schopnost detekce a zároveň je redukována míra chybných výsledků.

5.8 Postup přidání nové „Action“ do seznamu rozpoznatelných zvuků

Aplikaci lze rozšířit o nové detekční schopnosti následujícím způsobem.

- 1, Vytvořte novou podtřídu (Třídy Action jsou umístěny v com.Felipe.Actions) rozšířením třídy „ActionVlastnostiZvukovehoOtisku“ (com.Felipe.Wave).
- 2, Vytvořte konstruktor třídy, deklarujte proměnné a vložte experimentálně naměřené parametry podle vzoru nadtřídy.
- 3, Přidejte novou „Action“ ve třídě UiForm (com.Felipe), konstruktory všech „Action“ se nacházejí na řádce 201.

```
//-----V případě rozšíření programu Doplnit ZDE-----//  
wd.addAction(new PiskaniAction());  
wd.addAction(new TleskaniAction());  
wd.addAction(new KlepaniAction());
```

Obrázek 21 Přidání konstruktoru detekce UiForm.

6 UML CLASS MODEL

Kapitola je věnována grafickému popisu konstruovaného programu, pro tuto prezentaci byl zvolen UML model tříd. Tento model umožňuje zachycení obsahu tříd jejich metod a poměných, grafickým způsobem znázorňuje provázanost třídy s jinými třídami.



















6.1 Model GUI

Tabulka 2 Model GUI: Grafického uživatelského prostředí



6.2 Model nahrávání a přehrávání zvuku

Tabulka 3 Model tříd nahrávání a přehrávání

<p><<Java Class>>  PrehravacZvuku com.Felipe</p> <p> ▲ line: SourceDataLine ▲ vlakno: Thread ▲ audioInputStream: AudioInputStream ▲ bufSize: int ▲ progress: double ▲ onShutdown: Runnable ▲ playbackTimer: Runnable </p> <p>  PrehravacZvuku()  setOnShutdown(Runnable): void  setPlaybackTimer(Runnable): void  start(AudioInputStream): void  stop(): void  shutDown(String): void  run(): void </p>	<p><<Java Class>>  ZaznamZvuku com.Felipe</p> <p> ▲ line: TargetDataLine ▲ thread: Thread ▲ is: ByteArrayInputStream ▲ format: AudioFormat ▲ ais: AudioInputStream ▲ recordingCompleted: Runnable ▲ debug: boolean </p> <p>  ZaznamZvuku(boolean)  start(): void  stop(): void  shutDown(String): void  run(): void  getFormat(): AudioFormat  setOnRecordingCompleted(Runnable): void  getRecordedStream(): InputStream  getAudioStream(): AudioInputStream </p>
---	---

Tabulka 4 Události tříd nahrávání a přehrávání

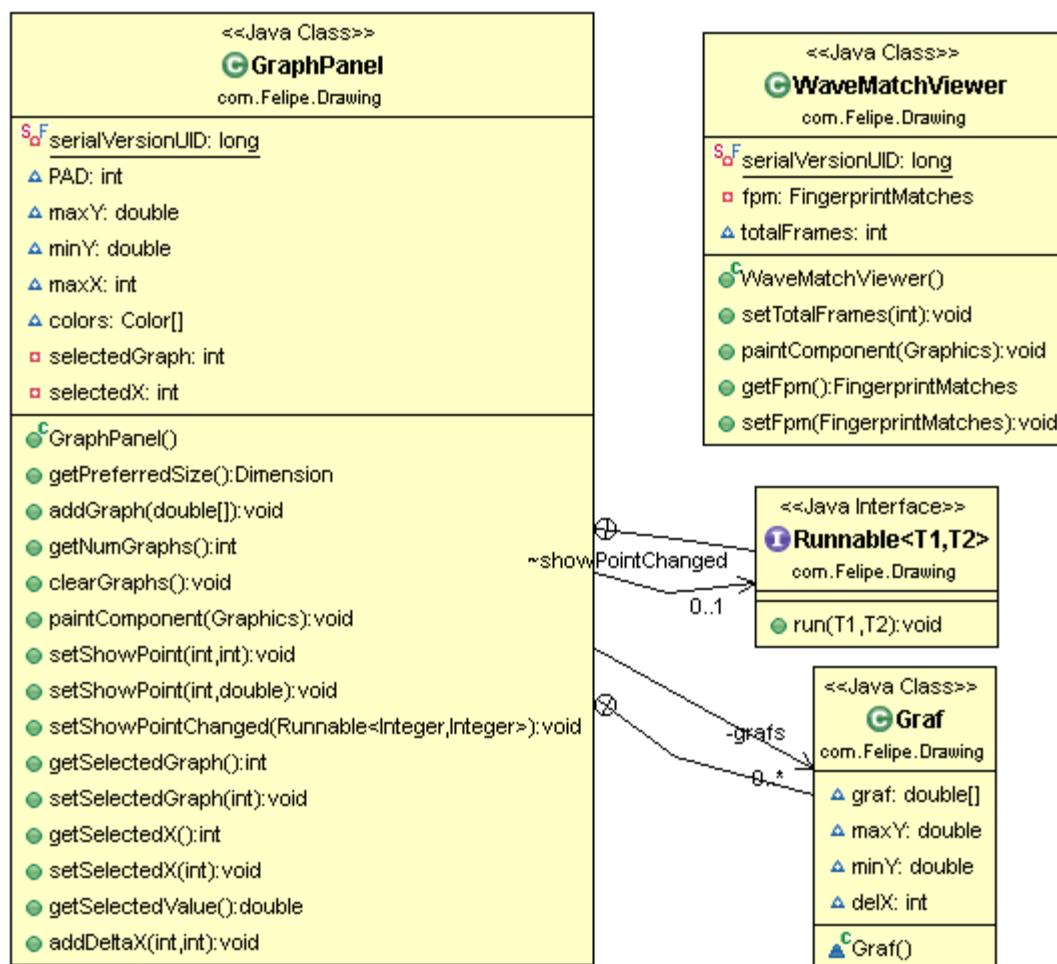
Události Events	
recordingCompleted	Indikace konce nahrávání

Tabulka 5 Metody Public tříd nahrávání a přehrávání

Metody Public	
start	Začátek nahrávání
stop	Konec nahrávání
getFormat	Vrací formát zaznamenané vlny
setOnRecordingCompleted(Runnable)	Po dokončení záznamu proved'
getRecordedStream()	Předává uložený zvukový datový tok
getAudioStream()	Předává zvukovou vlnu „AudioInputStream“

6.3 Model tříd vykreslování

Tabulka 6 Model vykreslování



Graph Panel

- Slouží k zobrazení diskrétních hodnot typu double v grafické podobě.
- Grafy mohou být přidány pomocí metody "addGraph".
- Událost "showPointChanged" slouží k indikaci změny vybraného bodu v grafu. Vybraný bod zde odkazuje na body, nad kterými je umístěn kurzor myši. Indikuje posun po ose x vzhledem k zobrazenému průběhu.

WaveMatchViewer

- Třída zobrazuje výsledky shody v 1D grafické podobě, je používán v hlavním okně aplikace.
- Body shody jsou reprezentovány modrým kruhem.
- Pokud nad tyto kruhy umístíte myš, zobrazí se info obsahující parametry řezu.

Tabulka 7 Metody Public třídy GraphPanel

Metody Public	
addGraph(double[])	Přidání grafu do graf panelu.
getNumGraphs()	Vrací současný počet grafů.
clearGraphs()	Maže všechny grafy.
setShowPoint(int, int)	Nastaví zvolený graf a index zvoleného bodu na grafu.
setShowPoint(int, double)	Nastaví zvolený graf a index zvoleného bodu na grafu. Druhý parametr je typu double v rozsahu hodnot 0 až 1. Hodnota 0 reprezentuje pozici krajního levého bodu a 1 pozici krajního pravého bodu.
setShowPointChanged(Runnable<Integer, Integer>)	Nastaví spouštěč události pro vybraný bod. Sets the trigger for selected point changed. Parametry jsou index grafu a zvolený index.
getSelectedGraph()	Navrací vybraný graf.
setSelectedGraph(int)	Nastaví vybraný graf.
getSelectedX()	Navrací index vybraného bodu.
setSelectedX(int)	Nastaví index vybraného bodu.
getSelectedValue()	Navrací vybrané hodnoty.
addDeltaX(int, int)	Počítá rozdíl při posunu na ose x.

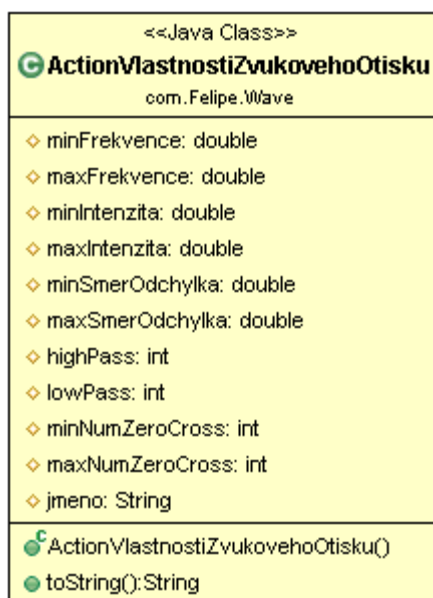
Tabulka 8 Metody Public třídy WaveMatchViewer

Metody Public	
setTotalFrames(int)	Nastavuje celkový počet řezů
getFpm()	Vrací výsledek řezu Fpm
setFpm(FingerprintMatches)	Předání výsledku k zobrazení

6.4 Model „ActionVlastnostiZvukovehoOtisku“

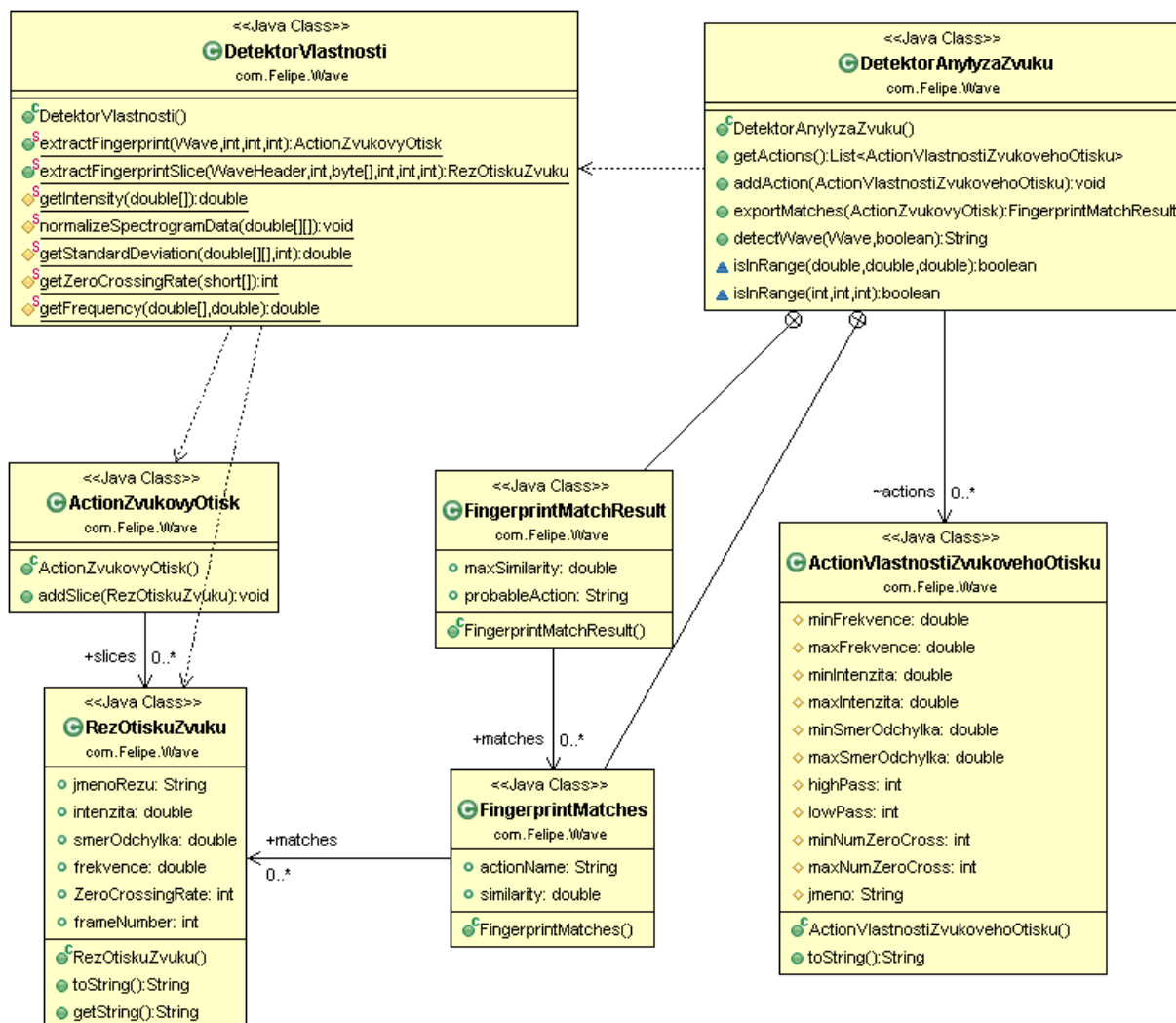
Definuje rámec všech klíčových parametrů, které jsou získány experimentálním měřením. Potomci této třídy jsou referenční vzorky, které jsou k dispozici detektoru hodnotícího shodu s nahraným zvukem.

Tabulka 9 Diagram nadtřídy „ActionVlastnostiZvukovehoOtisku“



6.5 Model tříd zajišťující manipulaci se zvukovým signálem

Tabulka 10 Model manipulace se zvukovým signálem



DetektorVlastnosti jeho součástí jsou modifikovaná třídy Musicg:

- „com.musicg.wave.WaveTypeDetector.getWhistleProbability()“
- „com.musicg.api.DetectionApi“

Poskytuje metody pro extrakci následujících parametrů:

1. Intenzita
2. ZRC
3. Frekvence
4. Směrodatná odchylka

Tabulka 11 Metody Public DetektorVlastnosti

Metody Public	
extractFingerprint(Wave wave, int lowPass, int highPass, int robustness)	Vrací otisk krátké zvukové nahrávky.
extractFingerprintSlice(WaveHeader waveHeader, int frameNumber, byte []audioBytes, int lowPass, int highPass, int robustness)	Vrací akustický řez.

Tabulka 12 Metody Private DetektorVlastnosti

Metody Private	
getIntensity(double[])	Vrací intenzitu.
normalizeSpectrogramData(double[][])	Navrací normalizované absoultspectrogram.
getStandardDeviation(double[][], int)	Vrací směrodatnou odchylku v bodě.
getZeroCrossingRate(short[])	Navrací ZRC zvukové vlny.
getFrequency(double[], double)	Navrací frekvenční spektrum zvukové vlny.

Tabulka 13 Metody Public DetektorAnalyzaZvuku

Metody Public	
getActions()	Vrací dostupné vzorky Action .
addAction(ActionVlastnostiZvukovehoOtisku)	Předává parametry Action detektoru.
exportMatches(ActionZvukovyOtisk)	Vrací detailní výsledek porovnání dostupné Action a změřeného akustického otisku.

detectWave(Wave, boolean)	Navrací nejpravděpodobnější výsledek typu string.
----------------------------------	---

Tabulka 14 Metody Private DetektorAnalyzaZvuku

Metody Private	
isInRange(double, double, double)	Vrací true, pokud první parametr je větší než druhý a zároveň menší než třetí.

7 ZHODNOCENÍ APLIKACE

Primární cíl práce bylo navrhnout a vytvořit softwarové řešení pro oblast porovnání krátkých zvukových stop. Testovací aplikace prezentovaná v praktické části diplomové práce splňuje všechny body zadání. Navíc je rozšířena o analytický modul, který lze použít pro sběr dat.

Zpočátku při zadání práce byl známý cíl a též byl zvolen jazyk Java jako prostředek realizace, z důvodů jeho relativně snadné přenositelnosti na další systémy. Aplikace je tedy potencionálně rozšiřitelná na systém Android. Před samým návrhem proběhl rozsáhlý průzkum, který cílil na aplikace s podobným zaměřením a směřoval k určení technických prostředků, kterých bude použito.

Konstrukce aplikace stojí na základech výkonné knihovny Musicg API (knihovna je mimo jiné kompatibilní s OS Android), kterou bylo nutné modifikovat tak, aby byla vhodná pro porovnávání krátkých zvukových stop. Dokumentace knihovny a uživatelská komunita byla v tomhle ohledu vítaným zdrojem informací.

Jednotlivé funkční části aplikace lze členit podobně, jako jsou rozložené balíky tříd ve zdrojovém kódu. (Main, Actions, Drawing, Wave)

V hlavním balíku se nachází třídy, starající se o chod celé aplikace, je zde definováno grafické rozhraní a třídy starající se o záznam a přehrávání zvuku.

Balík Actions aplikace používá jako databázi definovaných zvuků je lehce přístupná a umožňuje snadnou editaci pro odladění parametrů.

Balík Drawing zpracovává grafické výstupy vhodné pro prezentaci výsledků a pro potřeby experimentálního měření a používá Java knihovny java.awt a javax.swing.

Balík Wave obsahuje třídu DetektorAnylyzaZvuku.java, ta je jádrem aplikace a provádí porovnávání akustických parametrů.

ZÁVĚR

Diplomová práce byla vypracována podle zadání a splňuje všechny body zadání. Její součástí je i přiložená ukázková aplikace „Klasifikace zvuku“. První kapitola teoretické části je věnována teorií zvuku. Důraz byl kladen na definování zdroje zvuku, jeho šíření a funkci lidského ucha jako zařízení, určeného k zpracování zvukového signálu analogicky, jako jej zpracovává Fourierova transformace. Pozornost byla věnována i bezpečnostním aplikacím biometrických systémů pro rozpoznávání hlasu. Další kapitola teoretické části zpracovává přehled dostupných technologií a open-source řešení pro oblast porovnávání audio záznamu. Popisuje a kategorizuje zvolený software do několika kategorií a snaží se popsat princip funkce jednotlivých aplikací. Tato část je zhodnocena v závěru kapitoly. Celkově teoretická část definovala technologie a fyzikální jevy, které byly aplikovány v praktické části.

Praktická část diplomové práce popisuje hlavní výstup, kterým je ukázková testovací aplikace „Klasifikace Zvuku“. Při návrhu a implementaci vznikly softwarové moduly vhodné pro porovnání krátkých zvukových stop. Těmito moduly jsou DetektorAnylyzaZvuku.java a DetektorVlastnosti.java. Tyto moduly vykonávají funkce vyhodnocování a porovnávání akustických parametrů. Jejich použití není nutně vázané pouze na tuto aplikaci.

Ukázková aplikace splňuje požadavky kladené na porovnávací funkce a je určená pro OS Windows. Při vhodném nastavení vstupního mikrofону má ukázková aplikace vysokou přesnost při rozeznání pravděpodobného zvuku. Práce tímto ověřila funkčnost ukázkové aplikace.

Vývoj aplikace, jak je tomu od začátku, se ubírá směrem k upřesňování porovnávaných parametrů. V budoucnu by bylo dobré věnovat více pozornosti použité pásmové propusti, která je dosavadně nastavená defaultně pro všechny zvuky a její větší zapojení do procesu rozpoznávání by mělo vést k přesnějším výsledkům.

Dalším možným vylepšením je změna procesu načítání referenčních parametrů, v současné době k tomuto účelu vyhrazen balík Actions. V budoucnu by bylo vhodné zavádět tyto parametry do aplikace při spuštění, čímž by byly načteny ze souboru.

Perspektiva dalšího možného vývoje může směřovat k vytvoření aplikace pro zařízení Android, schopné počítat četnost zvoleného zvuku v měřeném čase nebo se schopností vybrat jeden nejpravděpodobnější výsledek detekce.

ZÁVĚR V ANGLIČTINĚ

This thesis was elaborated according to tasks and fulfill all points of the task. It also includes a sample application attached "Classification of Sound". The first chapter of the theoretical part is dedicated to the theory of sound, the emphases was put on defining the source of sound, its dispersion and function of the human ear as a device, which is assigned to processing analogically of audio signal as Fourier transform does. Attention was also paid to the security applications of biometric systems for the voice recognition. Another chapter of the theoretical part process overview of available technologies and open-source solutions for a compare audio recording. It describes and classifies the selected software into several categories and try to describe the principle of functions of individual applications, this part is reviewed at the end of the chapter. Generally the theoretical part defined technology and the physical phenomena that have been applied in the practical part. The practical part of the thesis mainly describes its main output, which is a sample test application "Classification of Sound". When designed and implemented originated software modules proper for the comparison of short audio tracks. These modules are `DetektorAnylyzaZvuku.java` and `DetektorVlastnosti.java`. These modules perform functions of evaluation and comparison of acoustic parameters. Their use is not necessarily fixed only to this application. The sample application meets the requirements for the comparative function, is designed for Windows OS. With appropriate settings, input microphone i sample application has a high precision in recognizing the probable sound , the thesis verified its functionality. The application development, as from the beginning is moving towards the refinement of compared parameters, in the future would be good to pay more attention to the used band-pass filter, which Is existing in all sounds default setting for all sounds, its increasing involvement in the the recognition process of sounds should lead to more precise results. Another possible improvement is a change in loading reference parameter, which is currently reserved the package `Actions`. It would be appropriate to introduce these parameters to the application at startup, which should have been loaded from a file. Perspective of further possible developments may evolving towards the development applications for Android devices, can calculate the frequency of the selected sound in the measured time or the ability to choose the one most likely outcome detection.

SEZNAM POUŽITÉ LITERATURY

- [1] SMETANA, Ctirad, a kol.. *Hluk a vibrace. Měření a hodnocení.*. Praha : Sdělovací technika, 1998. ISBN 80-90 1936-2-5. (CZ).
- [2] KOSTOLÁNYOVÁ, Kateřina. Úvod do multimédií: (grafika, hudba a zvuk). Vyd. 1. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2003, 54 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-704-2924-0.
- [3] ENCYKLOPAEDIA BRITANNICA. *The ear as spectrum analyzer* [online]. 2014 [cit. 2014-05-26]. Dostupné z: <http://www.britannica.com/EBchecked/topic/555255/sound/64017/The-ear-as-spectrum-analyzer>.
- [4] TONY SCARLATOS. *Center for visual Computing* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://xsrv.mm.cs.sunysb.edu/334/nature/amplitude.gif>
- [5] POLÁK, Josef. Integrální a diskrétní transformace. 3., přeprac. vyd. V Plzni: Západočeská univerzita, Fakulta aplikovaných věd, 2002, 197 s. ISBN 80-708-2924-9.
- [6] VUT. *Frekvenční analýza zvuku* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://fyzika.fce.vutbr.cz/file/martinek/frekvanalzvuku.pdf>
- [7] HOLČÍK, Lukáš. *Spektrální analýza hudební skladby*. Brno, 2009. Diplomová. VUT. Vedoucí práce MgA. Rudolf Růžička.
- [8] ISO IEC TR 21000-11 (2004), *Multimedia framework (MPEG-21) -- Part 11: Evaluation Tools for Persistent Association Technologies*.
- [9] MUSIGBRAINZ. *MusicBrainz Blog* [online]. 2006 [cit. 2014-05-26]. Dostupné z: <http://blog.musicbrainz.org/2006/03/12/new-fingerprinting-technology-available-now/>
- [10] RAK, Roman. *Biometrie a identita člověka ve forenzních a komerčních aplikacích*. 1. vyd. Praha: Grada, 2008, 631 s., 32 s. barev. obr. příl. ISBN 978-80-247-2365-5.
- [12] Comparison: Audio Editor. COMPARISONICS CORPORATION. *Comparison* [online]. 2014 [cit. 2014-05-26]. Dostupné z: <http://www.comparisonics.com/CSeditor.html>

- [12] COMPARISONICS CORPORATION. *FindSounds: Search web for the sounds* [online]. 2014 [cit. 2014-05-26]. Dostupné z: <http://www.findsounds.com/palette.html>
- [13] COMPARISON. *Google play* [online]. [cit. 26.5.2014]. Dostupný na WWW: <https://play.google.com/store/apps/details?id=com.comparisonics.fsa>
- [14] *MusicBrainz Picard* [online]. 2008 [cit. 2014-05-26]. Dostupné z: http://musicbrainz.org/doc/MusicBrainz_Picard
- [15] LALINSKÝ, Lukáš. *Fingerprinter* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://acoustid.org/fingerprinter>
- [16] GPLv3. *audioscout* [online]. [cit. 26.5.2014]. Dostupný na WWW: <https://code.google.com/p/audioscout/> [5] <http://open-fp.sourceforge.net/index.html>
- [17] PFALZONLINE.DE. *OpenFP* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://open-fp.sourceforge.net/>
- [18] ECHONEST. *Echoprint* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://echoprint.me/>
- [19] AUDIBLEMAGIC. *AudibleMagic* [online]. [cit. 26.5.2014]. Dostupný na WWW: <https://www.audiblemagic.com/>
- [20] YouTube. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-05-26]. Dostupné z: http://en.wikipedia.org/wiki/Content_ID#Content_ID
- [21] Shazam. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-05-26]. Dostupné z: [http://cs.wikipedia.org/wiki/Shazam_\(software\)](http://cs.wikipedia.org/wiki/Shazam_(software))
- [22] GRACENOTE. *Gracenote* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://www.gracenote.com/>
- [23] BERG, Richard E. ENCYKLOPAEDIA BRITANNICA. Sound [online]. 2006 [cit. 2014-01-25]. Dostupné z: <http://www.britannica.com/EBchecked/topic/555255/sound>
- [24] HEROUT, Pavel. *Java - bohatství knihoven*. 3. vyd. České Budějovice: Kopp, 2008, 251 s. ISBN 978-80-7232-368-5.

-
- [25] HEROUT, Pavel. *Java: grafické uživatelské prostředí a čeština*. 2. vyd. České Budějovice: Kopp, 2007, 316 s. ISBN 978-80-7232-328-9.
- [26] HEROUT, Pavel. *Java a XML*. 1. vyd. České Budějovice: Kopp, 2007, 313 s. ISBN 978-80-7232-307-4. [15] text
- [27] SCHILDT, Herbert. *Java 7: výukový kurz*. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.
- [28] HEROUT, Pavel. *Učebnice jazyka Java*. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.
- [29] *Integrální a diskrétní transformace*. 2. vyd. Plzeň: Západočeská univerzita, 2002, 136 s. ISBN 80-708-2040-3.
- [30] TROJAN, Jan. *Akustická ekologie a soundscape v kontextu multimédií*. 2011. vyd. Praha: Akademii múzických umění v Praze, 2011. ISBN 978-80-904506-4-6.
- [31] CHEUNG, Ka Wai. *Vývojářův kód*. 1. vyd. Překlad Andrea Hodaňová. Brno: Computer Press, 2013, 193 s. ISBN 978-80-251-3786-4.
- [32] OBJECTAID. *objectaid* [online]. [cit. 26.5.2014]. Dostupný na WWW: <http://www.objectaid.com/>
- [33] ECLIPSE. *WindowBuilder* [online]. [cit. 26.5.2014]. Dostupný na WWW: <https://www.eclipse.org/windowbuilder/>
-

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GUI	Grafické uživatelské rozhraní.
API	Rozhraní pro programování aplikace.
FFT	Rychlá Fourierova transformace.
kHz	Kilohertz.
ZRC	Frekvence průchodů nulou.
DTFT	Fourierova transformace s diskrétním časem
DFT	Diskrétní Fourierova transformace
AAC	Standart pro ztrátovou kompresi zvuku.
MP3	Standart pro ztrátovou kompresi zvuku.
WMA	Komprimační zvukový formát.
LPC	Algoritmus ztrátové komprese.
WAVE	Zvukový formát.
AIFF	Zvukový formát.
mu-law	Telekomunikační algoritmus.
ID	Identifikační údaj.
OGG	Formát pro digitální multimedia.
FLAC	Bezztrátový kodek.
AIFC	Zvukový formát.
AMR	Metoda komprese zvuku používaná především pro řeč.
PCM16	16 bitová pulzní kódová modulace.
ISO	Mezinárodní organizace pro normalizaci.
IEC	Mezinárodní elektrotechnická komise.
MPEG-7	Multimediální kompresní formát.
UML	Grafický jazyk pro vizualizaci a navrhování.

SEZNAM OBRÁZKŮ

Obrázek 1 ZRC [4].	14
Obrázek 2 Comparisonics: nalezení shody ve zvukovém průběhu.	22
Obrázek 3 Hlavní okno FindSounds Palette [12].	23
Obrázek 4 Androidová aplikace FindSounds [13].	24
Obrázek 5 Picard: identifikace hudebního kompaktního disku [14].	25
Obrázek 6 Schéma použitých nástrojů [17].	27
Obrázek 7 Schéma znázorňuje extrakci akustických otisků [17].	28
Obrázek 8 Postup při vzniku otisku.	28
Obrázek 9 Echoprint: schéma vnitřních funkcí aplikace [18].	30
Obrázek 11 Hlavní okno testovací aplikace	38
Obrázek 12 Zobrazení výsledku měření	39
Obrázek 13 GUI Analýza	42
Obrázek 14 Zobrazení grafů	43
Obrázek 15 Hromadné zobrazení označených hodnot	43
Obrázek 16 Seznam řezů	44
Obrázek 17 Exportovaný soubor	44
Obrázek 18 Dva průběhy v jednom grafu.....	45
Obrázek 19 Proložení obou změřených průběhů	46
Obrázek 20 Informace o řezu.....	47
Obrázek 21 Nový výsledek.....	48
Obrázek 22 Přidání konstruktora detekce UiForm.	48

SEZNAM TABULEK

Tabulka 1 Charakteristické parametry zvolené pro vyhledávání.....	40
Tabulka 2 Model GUI: Grafického uživatelského prostředí	49
Tabulka 3 Model tříd nahrávání a přehrávání.....	50
Tabulka 4 Události tříd nahrávání a přehrávání.....	50
Tabulka 5 Metody Public tříd nahrávání a přehrávání	50
Tabulka 6 Model vykreslování	51
Tabulka 7 Metody Public třídy GraphPanel	52
Tabulka 8 Metody Public třídy WaveMatchViewer	53
Tabulka 9 Diagram nadtřídy „ActionVlastnostiZvukovehoOtisku“	53
Tabulka 10 Model manipulace se zvukovým signálem.....	54
Tabulka 11 Metody Public DetektorVlastnosti	55
Tabulka 12 Metody Private DetektorVlastnosti	55
Tabulka 13 Metody Public DetektorAnalyzaZvuku	55
Tabulka 14 Metody Private DetektorAnalyzaZvuku.....	56