



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra radioelektroniky

Identifikace gramofonové desky podle krátké zvukové ukázky

Vinyl record identification based on short sound sample

Bakalářská práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Multimediální technika

Vedoucí práce: Ing. František Rund Ph.D.

Václav Král

Praha 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra radioelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Václav Král**

Studijní program: Komunikace, multimédia a elektronika
Obor: Multimediální technika

Název tématu: **Identifikace gramofonové desky podle krátké zvukové ukázky**

Pokyny pro vypracování:

Seznamte se s problematikou rozpoznávání a identifikace hudebních děl. Implementujte automatický systém pro identifikaci gramofonové desky na základě analýzy krátké zvukové ukázky z této desky.

Testujte úspěšnost identifikace v závislosti na možných artefaktech v testované ukázce, jejím časovém posuvu, apod.

Seznam odborné literatury:

- [1] Wold, E.; Blum, T.; Keislar, D.; Wheaten, J., Content-based classification, search, and retrieval of audio, in MultiMedia, IEEE , vol.3, no.3, pp.27-36, Fall 1996 doi: 10.1109/93.556537
- [2] Neuschmied, H.; Mayer, H.; Batlle, E., Content-based identification of audio titles on the Internet, in First Intern. Conf. on Web Delivering of Music, Proceedings. pp.96-100, 23-24 Nov. 2001 doi: 10.1109/WDM.2001.990163
- [3] Lartillot, O.; Toiviainen P., A Matlab Toolbox for Musical Feature Extraction from Audio, 10th. Int. Conf. on Digital Audio Effects, Bordeaux, 2007.

Vedoucí: Ing. František Rund, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

doc. Mgr. Petr Páta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 2. 2016

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 27. 5. 2016

Václav Král

Poděkování

Chtěl bych poděkovat panu Ing. Františku Rundovi, Ph.D. za odborné vedení práce, za pomoc a rady při zpracování této práce.

Abstrakt

Cílem práce bylo vytvořit systém k identifikaci gramofonové desky podle krátké zvukové ukázky z této desky. Z originálních nahrávek byly vypočteny parametry, které byly spolu s názvem dané desky uloženy do databáze. Následně byl vytvořen program, který vypočte parametry pro neznámou zadanou nahrávku a porovná je s parametry v databázi. Uživateli je nabídnuto 5 desek z databáze, které jsou zadané nahrávce nejvíce podobné. Program je vybaven ochranou proti časovému posunu, kdy je detekován začátek hudby, aby se porovnávaly parametry ze stejných úseků originálních nahrávek i nahrávky neznámé. Úspěšnost určení nahrávky je závislá na poškození neznámé nahrávky. Pro nahrávky s přidáním bílým Gaussovským šumem s odstupem 50 dB bylo správně určeno 82 % neznámých nahrávek mezi 5 nabídnutými a pro nahrávky s přeslechly mezi levým a pravým kanálem s odstupem 20 dB byla úspěšnost určení mezi 5 nejbližšími shodami 92 %.

Abstract

The goal of this project was to develop a system for identification of a gramophone record from a short sound sample of that recording. Parameters were computed from the original sound samples and were saved in a database together with the name of their record. Next a program was developed, which calculates parameters of an unknown recording and compares those parameters with the database. Then names of 5 closest matches are offered to the user. The program is equipped with a protection against time shift, when the beginning of the music is detected, ensuring the comparison of same segments of original and unknown sound. The success ratio of identification is dependent on the corruption of the unknown sample. The unknown sample was identified among the 5 closest matches 82 % of a time with additive Gaussian white noise of ratio of 50 dB and the samples with inter-channel crosstalk of ratio of 20 dB were successfully identified among the 5 closest matches in 92 % of the time.

Klíčová slova

gramofonová deska, rozpoznání a určení zvuku, MIRtoolbox, Matlab, poškození zvuku, databáze

Key words

gramophone recording, recognition and identification of sound, MIRtoolbox, Matlab, sound defects, database

Obsah

1. Úvod.....	7
2. Teoretická část	8
2.1. Gramofonové desky	8
2.1.1. Nahrávání zvuku.....	8
2.1.2. Výroba desek.....	8
2.1.3. Přehrávání desek.....	8
2.2. Strojové rozpoznání zvuku	9
2.3. Parametry popisující nahrávku	10
2.3.1. MIRtoolbox.....	10
2.3.2. Popis signálu v časové oblasti	10
2.3.3. Popis signálu ve spektrální oblasti	10
2.3.4. Popis signálu v dalších oblastech	11
2.4. Databáze	12
3. Praktická část	13
3.1. Schéma výsledného programu.....	13
3.2. Detekce začátku hudby	13
3.3. Volba parametrů	14
3.4. Vytvoření databáze	14
3.5. Výpočet nejbližších shod.....	15
3.6. Určení úspěšnosti porovnávání s databází	16
3.7. Vliv časového posunu na identifikaci	17
3.8. Testování vlivu znehodnocení nahrávek.....	17
3.9. Vliv přidaného šumu a přeslechů.....	18
3.10. Vliv parametru mirmode.....	18
3.11. Skript pro nalezení neznámé nahrávky.....	19
3.12. Výpočetní náročnost	19
4. Závěr.....	20
Seznam literatury.....	21
Seznam příloh	22

1. Úvod

Cílem této práce je vytvořit program, který dokáže porovnat krátkou neznámou zvukovou ukázkou gramofonové desky s databází a nabídne název gramofonové desky, která začíná stejnou zvukovou nahrávkou. Pro ověření funkčnosti bude tento systém testován různě poškozenými původními nahrávkami.

Pro člověka je jednoduché porovnat dvě zvukové ukázky. Díky schopnosti mozku rozpoznávat vzory rozpoznáme jako shodné i velmi poškozené nahrávky. Pro počítač je takový úkol velmi složitý. I mírně odlišný signál od původního by počítač při přímém porovnání určil jako samostatný signál. Pokud chceme, aby počítačový program zvládl porovnat 2 signály, musíme mu určit, jakým způsobem je má porovnávat.

Identifikace neznámých zvuků je s rozvojem internetu zajímavé téma. Existují například aplikace, které ze zvukové ukázky dokáží rozpoznat píseň a nabídnou uživateli její název a další informace. Zřejmě největším přínosem je ovšem identifikace porušování autorských práv. Existují systémy, které prohledávají různá videa nebo internetová rádia a zajišťují, že jejich provozovatel řádně zaplatil majiteli autorských práv za používání nalezených písní.

Identifikace zvuku se dá ovšem využít i u starších systému. Může sloužit například ke kontrole vložení správné vzorové nahrávky před výrobou gramofonových desek.

V teoretické části se budeme zabývat samotnými gramofonovými deskami – jak se vyrábí a přehrávají a co může způsobit poškození přehrávaného zvuk. Dále se podíváme, jak se dnes dají počítačem popsat zvukové nahrávky a jak budou parametry počítány v této práci. Nakonec si představíme základy databází.

Praktická část bude obsahovat jednotlivé kroky k vytvoření programu pro identifikaci. Z původních nahrávek vypočteme parametry a uložíme je do databáze. Poté budeme poškozovat původní signál a počítat z něj nové parametry. Tyto parametry poté porovnáme s parametry v databázi a určíme podobnost dvou nahrávek. Poté otestujeme úspěšnost porovnání pro všechny nahrávky a různá poškození. Spojením dílčích kroků potom získáme samotný program, který po zadání neznámé nahrávky nabídne název podobné nahrávky z databáze.

2. Teoretická část

V této části se budeme zabývat současným stavem problematiky a představíme si informace potřebné k dosažení cíle práce.

2.1. Gramofonové desky

Jelikož se zabýváme identifikací nahrávky na gramofonové desce, je důležité uvědomit si, jak záznam na gramofonové desce funguje a k jakým nedostatkům může docházet.

Vinylové desky jsou jedním z nejtálejších médií k uchování analogového zvuku. U samotného materiálu dochází k degeneraci za více než 100 let. Při nesprávném skladování může ovšem docházet k poškození například střídáním teplot nebo ultrafialovým zářením. [11] V drážkách desky se také může usazovat prach, který způsobí praskot. Při nesprávném zacházení s deskou vzniká i možnost poškrábání, které se projeví jako lupance.

2.1.1. Nahrávání zvuku

Mnoho nedostatků může postihnout desky už při nahrávání, například vlivem špatně nastaveného zesilovače mikrofону (přebuzení), použitím staré záznamové desky (šum) nebo rozladěným rycím hrotem (rozdílná vzdálenost drážek). [10]

2.1.2. Výroba desek

Další artefakty vznikají při lisování. Při nečistotě lisu vzniknou na desce bubliny nebo hrboly ovlivňující několik vedlejších drážek. Chyby ve zvuku také záleží na kvalitě materiálu a výroby. Drážky nebudou nikdy dokonale hladké, protože materiál vytváří zrna, která se při přehrávání přenáší jako součást zvuku a zapříčiňují šum. Pokud není při výrobě střed otáčení desky také střed spirály drážek, dochází k periodické změně frekvence přehrávaného zvuku. [10]

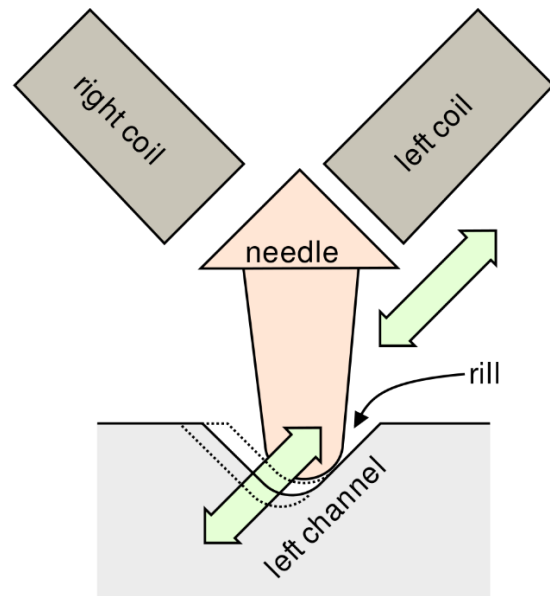
2.1.3. Přehrávání desek

Výše uvedené problémy desek jsou většinou odhalené před prodejem, takže se takové desky nedostanou k posluchačům. Významnější vliv na kvalitu poslechu mají nedostatky, které vznikají při přehrávání.

Nehledě na stav nahrávky dochází k přidání šumu samotným přehrávačem. Tento šum zapříčiní jednak snímání jehlou, ale také elektrický šum obvodů uvnitř přehrávače. Dnes mají běžně prodávané přehrávače udávaný odstup signálu od šumu 50 dB. [3]

Další poškození zvuku může zapříčinit špatně vycentrovaný přehrávač. Deska se poté neotáčí přesně kolem středu spirály drážek, což přináší kolísání hlasitosti a frekvence. Jelikož je tato chyba znatelná už při malém kolísání (nad 0,20 % změny), uživatel by nejspíše takový přehrávač nepoužíval. [3]

Stereofonní zvuk je na gramofonových deskách uchováván v jedné drážce. Pravá stěna drážky obsahuje pravý kanál a levá strana drážky obsahuje levý kanál. Tyto dvě stěny jsou na sebe kolmé. Při průchodu jehly drážkou se poloha jehly snímá cívkami, které jsou na sebe také kolmé, každá cívka tedy snímá jeden kanál. Pokud nebudou stěny drážek desky nebo snímací cívky na sebe přesně kolmé, může docházet k přeslechům kanálů, tedy část signálu levého kanálu se přenesení na pravý kanál a naopak. Dříve byl běžný odstup hlavního kanálu od vedlejšího 20 dB, dnes lze dosáhnout odstupu i 40 dB. [1] Posлуhač přeslechy většinou nepostřehne, protože levý a pravý kanál jsou si značně podobné.



Obrázek 1 - Princip přehrávání stereofonního zvuku [12]

2.2. Strojové rozpoznání zvuku

K rozpoznávání zvuku existuje mnoho přístupů. V článku Jonathana T. Footea [2] je představen systém založený na asistovaném učení z obsahu nahrávek a následném porovnávání neznámých zvuků. Parametry nahrávek jsou získány výpočtem melfrekvenčních cepstrálních koeficientů (MFCC). O povaze nahrávky poté rozhoduje stromová kvantizace. U trénovací množiny nahrávek je vytvořena stromová struktura člověkem, počítačový program poté dokáže využít tuto stromovou strukturu k rozhodnutí, které skupině nahrávek je neznámá ukázka nejvíce podobná. Největší výhodou této metody je, že se soustředí jenom na obsah nahrávek a správným nastavením kvantizace lze zajistit odolnost proti chybám signálu.

V článku Identification of Audio Titles on the Internet [9] je vysloven koncept AudioDNA. Zvuková nahrávka se považuje za sled genů, které představují vypočtené parametry úseků signálu. Jako podobnost dvou nahrávek se vyhodnotí shoda sekvence genů neznámé nahrávky se sekvencí v databázi. Při tvorbě AudioDNA se signál rozdělí na krátké úseky, z nichž se vypočtou parametry týkající se hlavně barvy zvuku. Parametry jsou poté porovnávány s geny a vypočte se nejpravděpodobnější gen, který danému parametru odpovídá. Tato metoda je vhodná k hledání úseků licencované hudby, pokud je tato hudba použita jako hudební podklad pod mluvené slovo nebo v předělaných verzích hudebních skladeb.

Mnoho způsobů klasifikace zvuků bylo představeno ve článku Content-Based Classification, Search, and Retrieval of Audio [12]. Hlavním cílem bylo vytvořit způsob vyhledávání zvuků podobným způsobem, jakým vyhledáváme v textu – zadáním klíčových slov. Jsou použity parametry spojené s akustikou a vnímáním, jako například hlasitost nebo výška tónu. Parametry se počítají vždy pro úsek signálu. Do výsledné databáze se ukládá střední hodnota parametrů, jejich rozptyl a autokorelace.

2.3. Parametry popisující nahrávku

Ačkoliv by bylo možné funkce k výpočtu potřebných parametrů vytvořit ručně, je rychlejší použít už předem připravené funkce, z kterých vybereme vhodné parametry. Za vhodný budeme považovat takový parametr, který jednoznačně popisuje signál v určité oblasti, a to jednou hodnotou.

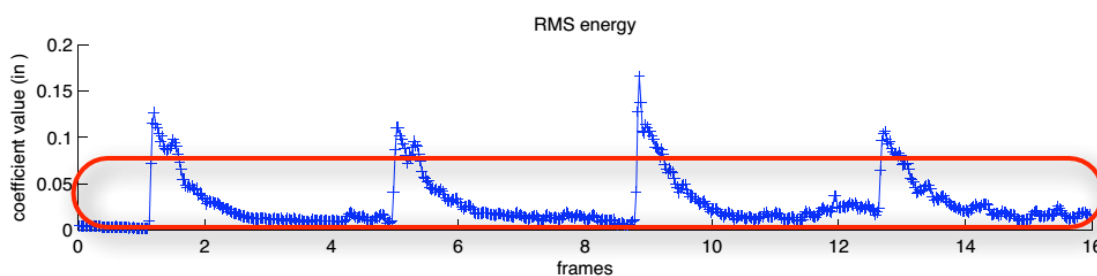
2.3.1. MIRtoolbox

MIRtoolbox nabízí sadu funkcí pro Matlab určenou k výpočtu vlastností hudebních souborů. Analýza je založena na malých dílčích krocích, což umožňuje použití dílčích kroků ve více funkcích. [7] Mezi podporované parametry patří mimo jiné vlastnosti barvy zvuku, tonality nebo rytmu. Nahrávka je automaticky vycentrována a normalizována hodnotou RMS.

Představme si nyní několik funkcí, které se dají pomocí MIRtoolboxu vypočítat.

2.3.2. Popis signálu v časové oblasti

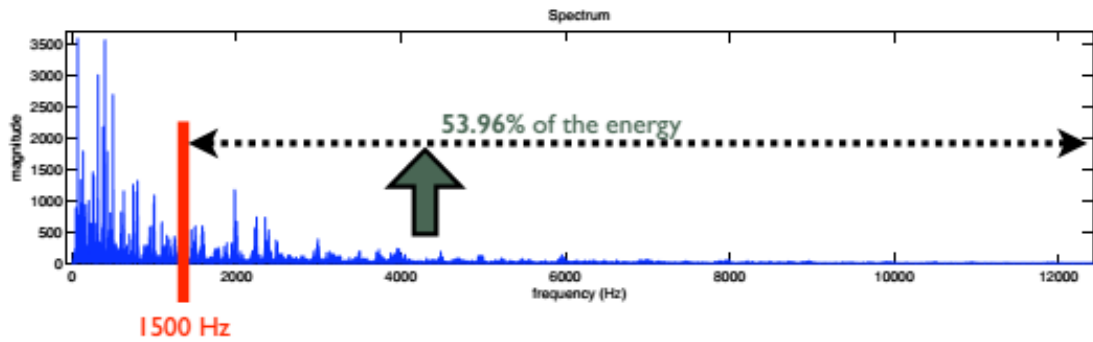
Použitím funkce *mirlowenergy* zjistíme rozložení energie v časové oblasti signálu. Energie každého rámcu je porovnána s energií RMS celého signálu. Výsledkem je procento rámců, jejichž energie je nižší než hodnota RMS signálu. [8] Funkce *mirrms* také popisuje energii signálu v časové oblasti, tentokrát hodnotou RMS signálu.



Obrázek 2 – Několik rámců s vysokou energií způsobí, že většina energie rámců je pod hodnotou RMS [7]

2.3.3. Popis signálu ve spektrální oblasti

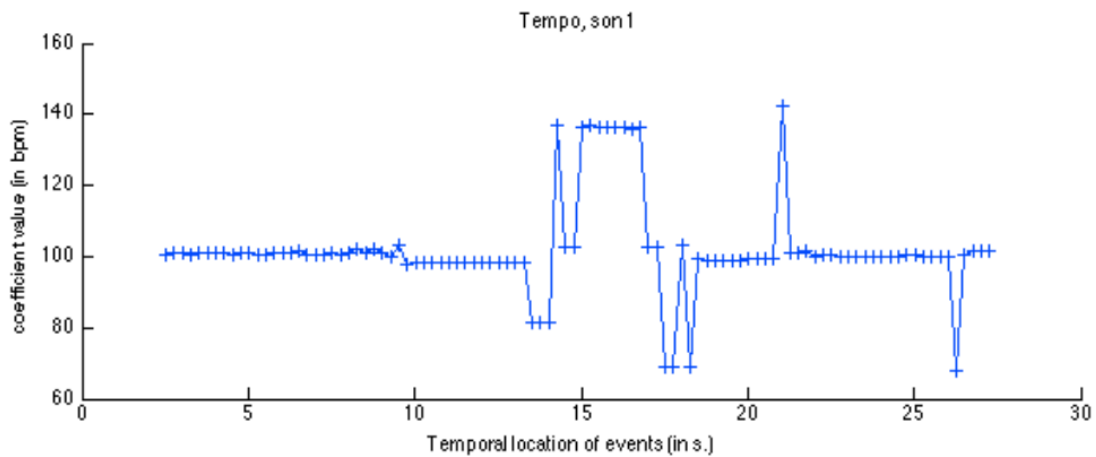
Pro popis rozložení energie ve spektrální oblasti lze použít funkci *mirbrightness*. Výstupem funkce je frekvence, nad kterou se nachází 53,96 % energie signálu. [8] Alternativou je funkce *mirrolloff*, jejímž výstupem je frekvence, pod kterou se nachází 85 % energie ve spektru.



Obrázek 3 - Funkcí *mirbrightness* se vypočte frekvence, nad kterou se nachází 53,96 % energie [7]

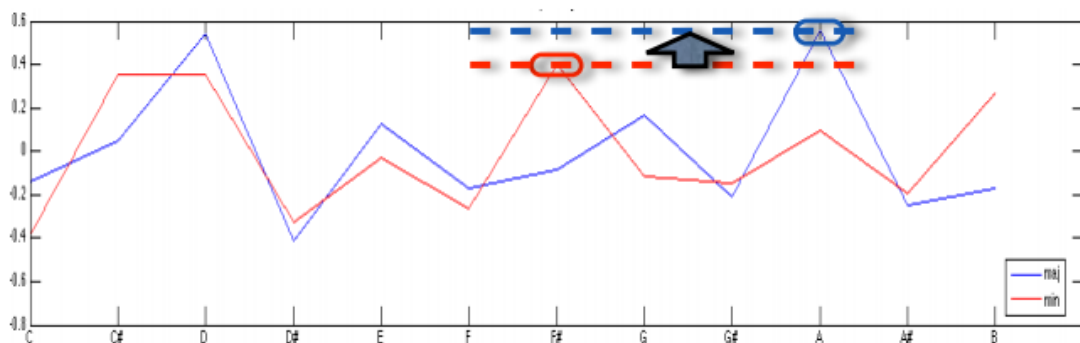
2.3.4. Popis signálu v dalších oblastech

Funkcí *mirtempo* lze odhadnout hudební tempo nahrávky v jednotkách bpm (úderů za minutu). Je vypočtena autokorelační funkce obálky signálu, z které jsou poté vybrány vrcholy a jejich periodičita je přepočtena do jednotek bpm. [8]



Obrázek 4 - Závislost tempa na poloze událostí při výpočtu *mirtempo* [7]

Parametr *mirmode* popisuje tonalitu nahrávky. Výsledkem je rozdíl mezi nejsilnější durovou a nejsilnější mollovou tóninou. [8]



Obrázek 5 - Rozdíl mezi nejvýznamnější durovou (modře) a mollovou (červeně) tóninou [7]

2.4. Databáze

Jelikož budeme potřebovat vytvořit databázi hodnot, představme si krátce různé přístupy k tvorbě databází.

Pojmem databáze můžeme myslet samotnou množinu dat uložených na médiu, případně i softwarové prostředky sloužící k manipulaci s těmito daty.

V hierarchických databázích jsou data uspořádány do stromové struktury, kde podřazený parametr odkazuje na svůj rodičovský parametr. Každý parametr obsahuje vždy pouze jednu hodnotu.

V relačních databázích jsou data uloženy v tabulce (relaci), kde jeden řádek (záznam) popisuje několika parametry (atributy) objekt určený primárním klíčem (jednoznačné jméno). Jednotlivé tabulky mohou vzájemně odkazovat na své objekty.

S rozvojem objektového programování se začaly vyvíjet i objektové databáze a speciální databázové objektové jazyky. Objektové databáze měly nahradit relační databáze, dnes se ovšem nejčastěji používá kombinace obojího, tedy objektově-relační databáze. [5]

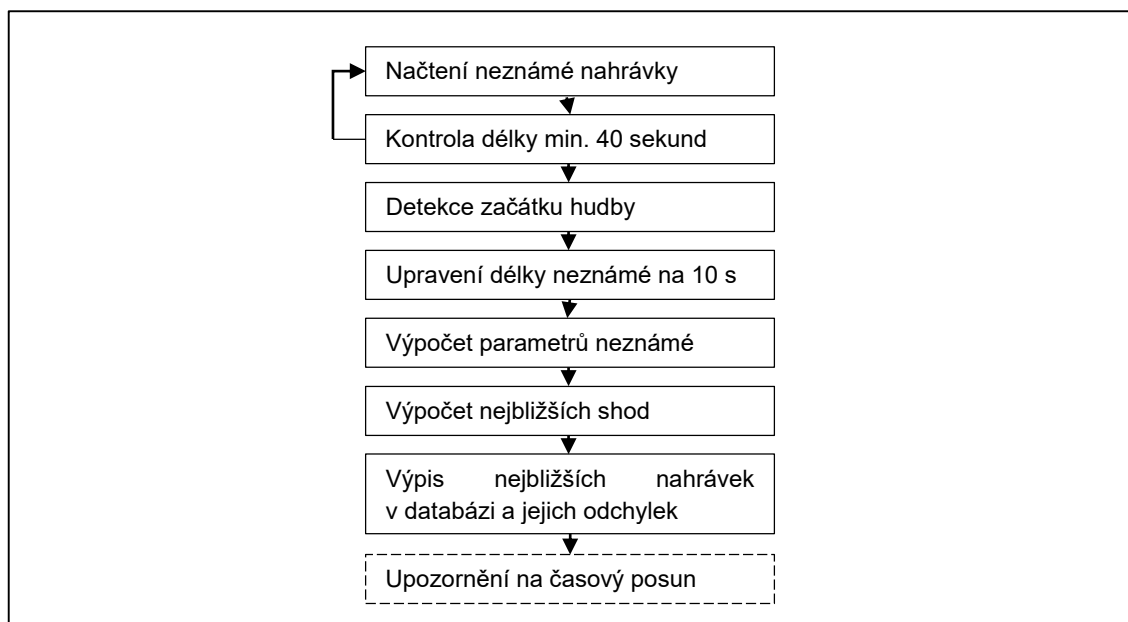
Vzhledem k malému vzorku nahrávek bude nejjednodušší použít relační databázi, kde každá nahrávka bude jednoznačně určena svým názvem a budou jí přiřazeny její parametry. Tento systém se osvědčil i v minulém projektu.

3. Praktická část

V následujícím textu se budeme zabývat vytvořením databáze a programu k vyhledání neznámé nahrávky. Pro zadané nahrávky vypočteme jejich parametry a uložíme je do souboru. Pro účel vyhodnocení úspěšnosti nalezení neznámé nahrávky nahrávky původní poškodíme, vypočteme jejich parametry a poté porovnáme úspěšnost vyhodnocení.

3.1. Schéma výsledného programu

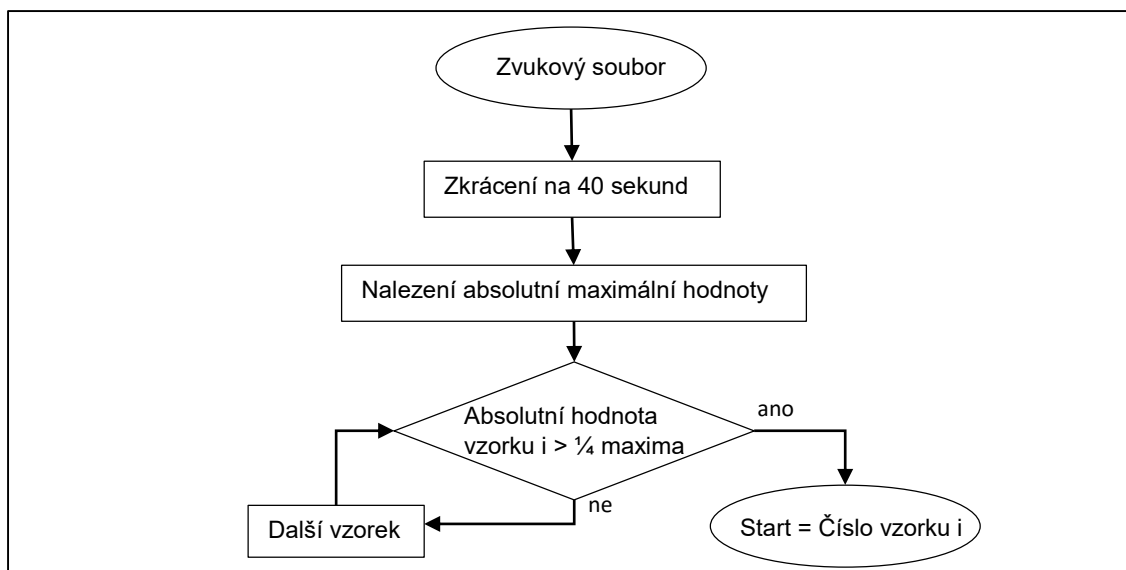
Pro přehlednost je na obrázku 6 uvedeno blokové schéma výsledného programu. Popis a funkci jednotlivých bloků rozebereme v následujícím textu. Uváděné skripty se používají k výpočtu databáze, jejich algoritmy jsou poté použity v samotném skriptu pro vyhledávání.



Obrázek 6 - Blokové schéma skriptu *hledatNeznamou.m*

3.2. Detekce začátku hudby

Pro zvýšení šance správného nalezení nahrávky v databázi je nutné, aby se porovnával stejný časový úsek nahrávek. Originální nahrávky se zkrátí na 40 sekund, kdy by se měla hudba rozvinout do plné hlasitosti. Z takto zkrácených nahrávek je vypočítána absolutní maximální amplituda. Poté se postupně porovnávají absolutní hodnoty amplitudy vzorků od začátku signálu, dokud nedojde k překročení 25 % absolutní maximální amplitudy. Číslo prvního vzorku, který tuto hodnotu překročí, je považováno za začátek hudby a je dále předáno jako proměnná *start*. Vývojové schéma si můžete prohlédnout na obrázku 7.



Obrázek 7 - Vývojové schéma detekce začátku hudby

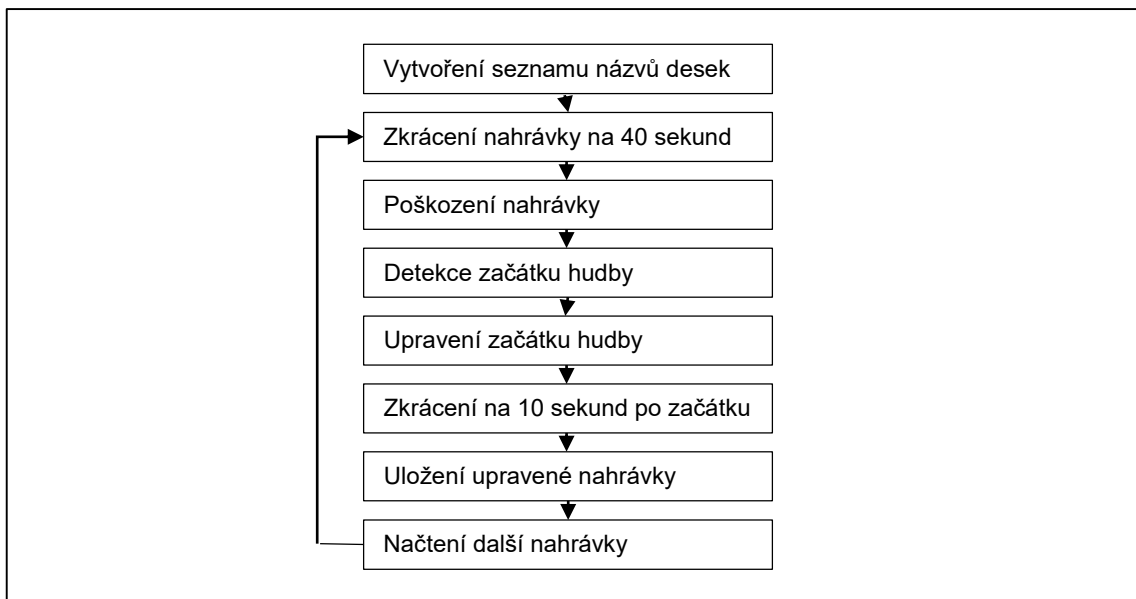
3.3. Volba parametrů

Nahrávky je třeba popsat jednoznačným jednočíselným parametrem. Tyto parametry se volily ze souboru funkcí MIRtoolbox pro Matlab. Cílem bylo popsat nahrávku v co nejvíce oblastech. V předchozím projektu na stejné téma bylo zjištěno, že pro popis v časové oblasti se více hodí parametr mirlowenergy než mirrms. Jelikož jsou signály normovány právě přes hodnotu RMS, parametr mirrms byl pro všechny nahrávky stejný. Při popisu signálu ve frekvenční oblasti přinesl parametr mirbrightness lepší výsledky, než parametr mirrolloff. Dále byl použit parametr mirtempo, který dále zpřesnil určení neznámé nahrávky při použití spolu s mirlowenergy a mirbrightness. [6] Pro další možné zlepšení určení nahrávky bude přidán parametr mirmode, který popisuje nahrávku v další oblasti, tentokrát v tonalitě. Vliv tohoto parametru bude otestován.

3.4. Vytvoření databáze

Pro účely testování systému byl firmou GZ Media poskytnut soubor 264 nahrávek z gramofonových desek. V dalším textu se tento vzorek nahrávek považuje za originální nahrávky, ze kterých se počítají parametry.

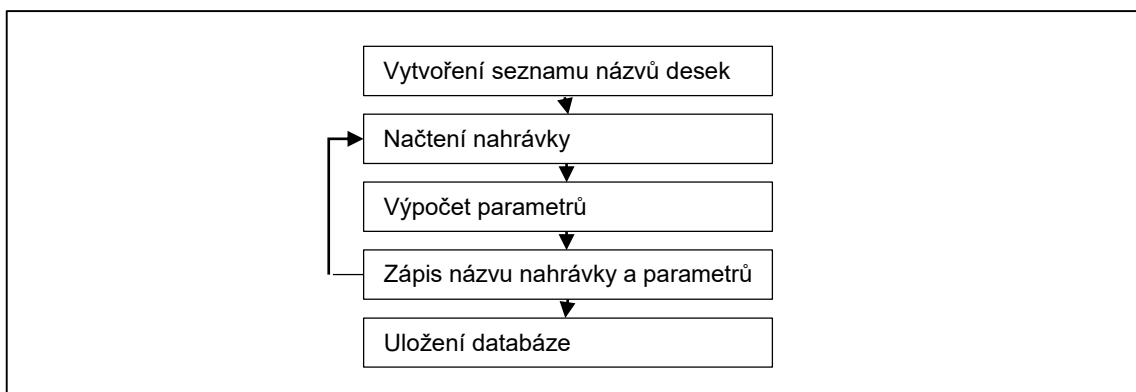
Abychom mohli vytvořit databázi, potřebujeme zkrátit původní nahrávky. V minulém projektu se osvědčila délka nahrávky 10 sekund, která byla kompromisem mezi dostatečným souborem dat pro výpočet parametrů a výpočetní dobou. Úprava nahrávek proběhne použitím skriptu *zkraceni.m* (viz Příloha B). Blokované schéma úpravy nahrávek je na obrázku 8. V tomto případě bloky *poškození nahrávky* a *upravení začátku hudby* neprovádí žádné změny signálu.



Obrázek 8 - Blokové schéma skriptu *zkraceni.m* (Příloha B)

Samotná databáze bude mít formu tabulky, kdy každý řádek odpovídá jedné nahrávce. Řádek bude obsahovat název nahrávky a parametry *mirloenergy*, *mirtempo*, *mirbrightness* a *mirmode* dané nahrávky.

Pro vytvoření databáze jsem použil skript *tvorba.m* (viz Příloha A a obrázek 9). [6] Skript načte složku zvukových souborů předem zkrácených na 10 s. Pro každou nahrávku jsou vypočteny jednotlivé parametry a uloženy po řádcích spolu s názvem nahrávky do tabulky v souboru *delsi_25max.mat* (viz Příloha 2).



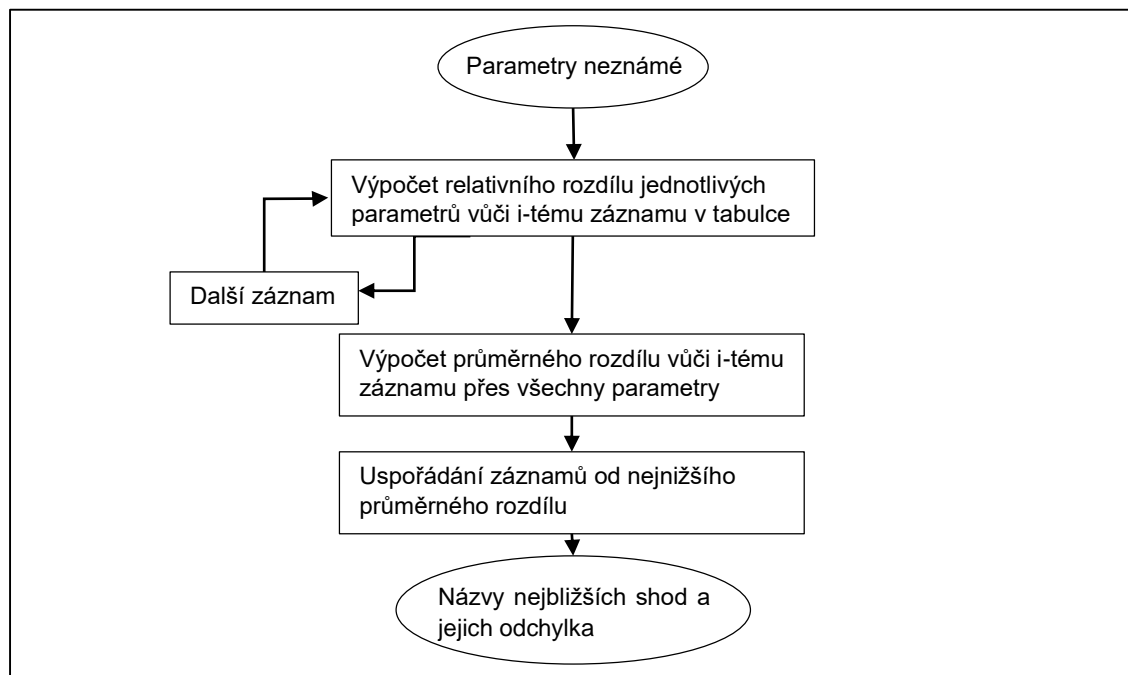
Obrázek 9 - Blokové schéma skriptu *tvorba.m* (Příloha A)

3.5. Výpočet nejbližších shod

Zatímco v předchozím projektu byly vyhledávány meze, do kterých se každý parametr musel vejít, aby byly nahrávky prohlášené za podobné, v této práci bude prozkoumán jiný přístup pro nalezení podobných nahrávek.

Když vypočteme relativní odchylku parametru neznámé nahrávky od parametru originálu ($(\text{parametr originálu} - \text{parametr neznámé}) / \text{parametr originálu}$), získáme, o kolik procent se tyto dvě nahrávky v daném parametru liší. Daná nahrávka by se měla v každém parametru lišit od původní co nejméně, proto by měl i průměr odchylek všech

parametrů být co nejmenší. Když uspořádáme nahrávky od nejmenší odchylky, získáme originální nahrávky nejvíce podobné nahrávce neznámé. Vývojové schéma výpočtu nejbližších shod si můžete prohlédnout na obrázku 10.

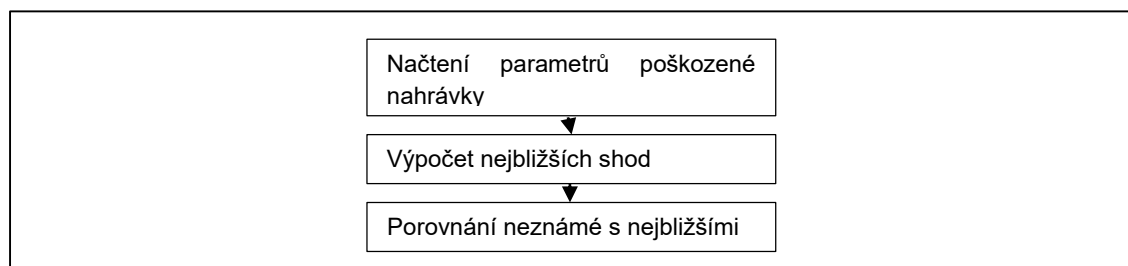


Obrázek 10 - Vývojové schéma výpočtu nejbližších shod

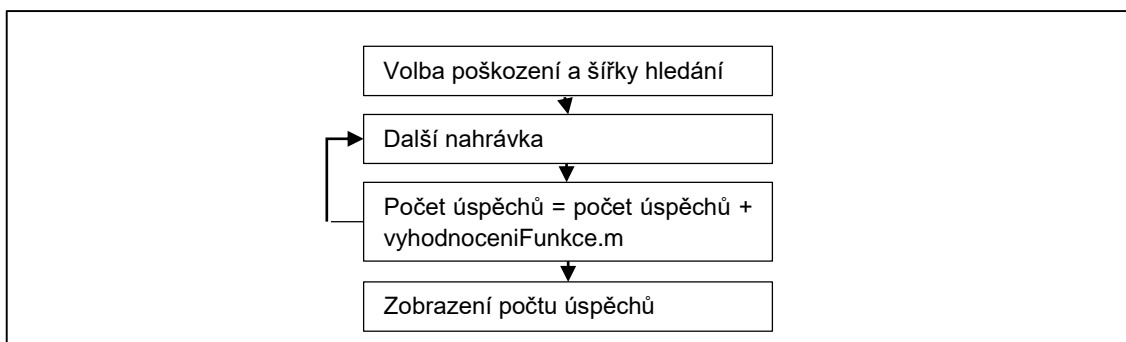
3.6. Určení úspěšnosti porovnávání s databází

Pro vyhodnocení úspěšnosti systému identifikace je potřeba vytvořit systém, který zajistí porovnání většího počtu neznámých nahrávek s databází.

K tomu nám pomůže funkce vyhodnoceniFunkce.m (viz Příloha C), jejíž vstupní parametry jsou pořadí neznámé nahrávky (odpovídající názvu), počet nahrávek s nejnižší odchylkou, v kterých hledáme shodu, a typ poškození nahrávky. Pokud je neznámá nahrávka mezi daným počtem nejbližších shod, vrátí funkce 1, jinak 0. Tuto funkci voláme ze skriptu vyhodnoceni.m (viz Příloha D), kde zadáme poškození nahrávky a počet nejbližších nahrávek, ve kterém vyhledáváme. Parametry poškozených nahrávek jsou předem vypočteny a uloženy do odpovídajících databází. Každá neznámá nahrávka se porovná s databází parametrů originálních nahrávek. Vyhodnocuje se počet neznámých nahrávek, které byly správně určeny mezi daným počtem nejbližších shod.



Obrázek 11 - Blokové schéma funkce vyhodnoceniFunkce.m (Příloha C)



Obrázek 12 - Blokové schéma skriptu vyhodnoceni.m (Příloha D)

3.7. Vliv časového posunu na identifikaci

V uživatelem zadané nahrávce je detekován samotný začátek hudby, čímž je zajištěno, že neznámá nahrávka nebude začínat dříve, než nahrávky použité k výpočtu parametrů v databázi. Toto opatření odstraňuje nechtěný zvuk před začátkem hudby. Je ale také potřeba uvažovat situaci, kdy uživatelova nahrávka začíná pozdě, tedy po uplynutí určité doby od začátku hudby.

K simulaci tohoto problému jsem pomocí skriptu *zkraceni.m* (Obrázek 8) vytvořil z originálních nahrávek nahrávky časově posunuté až o 1 sekundu oproti detekovanému začátku hudby. Upravení začátku zajistí posun prvního vzorku nového signálu o daný časový interval. U těchto nahrávek jsem vypočítal skriptem *tvorba.m* (Obrázek 9) jejich parametry, které byly uloženy do souborů *delší_25max_zpozdeni**.mat*. Samotný vliv časového posunu byl poté vyhodnocen skriptem *vyhodnoceni.m* (Obrázek 12).

Tabulka 1 ukazuje, že i malé časové posunutí způsobí strmý pokles úspěšnosti určení neznámé nahrávky. Toto se ale dá očekávat vzhledem k relativní délce posunu. Při posunu 1 sekunda je totiž desetina signálu odlišná od originálního.

Tabulka 1 - Úspěšnost nalezení neznámé nahrávky mezi daným počtem nejbližších shod

Posun	Vyhledávání v nejbližších shodách			
	1	3	5	10
0,1 s	80,3030 %	86,6364 %	91,2879 %	94,3182 %
0,2 s	76,5152 %	84,8485 %	87,5000 %	92,0455 %
0,5 s	51,8939 %	69,3182 %	76,5152 %	82,9545 %
1,0 s	34,0909 %	47,7273 %	58,3333 %	65,5303 %

3.8. Testování vlivu znehodnocení nahrávek

Pro účely vyhodnocení úspěšnosti bylo nutné vytvořit poškozené nahrávky. Jako poškození jsem uvažoval chyby vzniklé přehráváním, kterých si uživatel nemusí všimnout.

Hlavním nedostatkem může být šum přidaný ke zvuku. Ten jsem vytvořil přidáním bílého Gaussovského šumu s odstupem 50 dB od signálu, což je běžná hodnota u levnějších přehrávačů [3], a 70 dB. Přidání šumu proběhlo skriptem *zkraceni.m*, kde v bloku

poškození nahrávky (Obrázek 8) byla na signál využita v Matlabu dostupná funkce `awgn`(signál, odstup signálu od šumu).

Dalším nedostatkem můžou být přeslechy pravého a levého kanálu, kdy přeslechy s odstupem až 20 dB jsou nepostřehnutelné díky podobnostem levého a pravého kanálu [1]. K původnímu signálu jsem proto přidal skriptem *zkraceni.m* přeslechy s odstupem 20 dB a 40 dB. Toho bylo docíleno v bloku *poškození nahrávky* přičtením patřičného násobku levého kanálu k pravému a naopak (Obrázek 8).

Pro poškozené signály jsem opět vypočítal parametry a uložil je do souborů *awgn50_25max.mat*, *awgn70_25max.mat*, *crosstalk20_25max.mat* a *crosstalk40_25max.mat*.

3.9. Vliv přidaného šumu a přeslechů

Tabulka 2 ukazuje úspěšnost určení neznámé nahrávky mezi určitým počtem nejbližších shod vypočtenou pomocí skriptu *vyhodnoceni.m* (Obrázek 12). Vidíme, že neznámá nahrávka je přibližně v 85 % případů hned první nejbližší shoda. Pokud rozšíříme hledání na 5 nejbližších shod, získáme úspěšnost nalezení nahrávky přibližně 92 %. Pokud dále zvyšujeme počet podobných nahrávek, v kterých vyhledáváme, úspěšnost se sice zvyšuje, ale také se zvyšuje počet nesprávných nahrávek před správnou. Z hlediska uživatele by bylo nejlepší nabídnout 5 nejbližších shod pro zadanou neznámou nahrávku, kdy ve většině případů neznámá nahrávka bude hned první nabídnutá. Ze souboru 264 testovacích nahrávek bylo průměrně 243 neznámých mezi 5 nejbližšími a 224 byla hned první nejbližší. Při vypsání 5 nejbližších je tedy pravděpodobnost 92 %, že hned první nabídnutá shoda je hledaná nahrávka.

Tabulka 2 - Úspěšnost nalezení neznámé nahrávky mezi daným počtem nejbližších shod

Poškození	Vyhledávání v nejbližších shodách			
	1	3	5	10
AWGN 50 dB	71,5909 %	80,3030 %	82,1970 %	84,0909 %
AWGN 70 dB	92,4242 %	95,0758 %	96,2121 %	96,5909 %
Přeslechy 20 dB	85,6061 %	89,7727 %	92,0455 %	93,1818 %
Přeslechy 40 dB	84,8485 %	89,7727 %	92,0455 %	93,1818 %

3.10. Vliv parametru `mirmode`

Vliv nově přidaného parametru lze ukázat výpočtem nalezených shod, pokud nebudeme daný parametr uvažovat. Toho docílíme nezapočítáním tohoto parametru při výpočtu celkové odchylky ve funkci *vyhodnoceniFunkce.m*. V tabulce 3 vidíme úspěšnost určení vypočtenou skriptem *vyhodnoceni.m* (Obrázek 12), když se neuvažuje parametr `mirmode`.

Tabulka 3 - Úspěšnost nalezení neznámé nahrávky mezi daným počtem nejbližších shod bez uvažování parametru *mirmode*

Poškození	Vyhledávání v nejbližších shodách			
	1	3	5	10
AWGN 50 dB	49,2424 %	59,4697 %	64,3939 %	70,4545 %
Přeslechy 20 dB	87,1212 %	90,1515 %	91,6667 %	92,8030 %
Posun 0,5 s	74,2424 %	82,9545 %	84,8485 %	87,1212 %

Při uvažování časového posunu získáme vyřazením parametru *mirmode* o 10 % vyšší úspěšnost při hledání mezi 5 nejbližšími shodami (Tabulka 1). Vyřazením tohoto parametru u poškození nahrávky přeslechy získáme úspěšnost o 1 % menší a vyřazením *mirmode* u signálu s AWGN 50 dB je úspěšnost nižší o 18 % (Tabulka 2).

Jelikož lze chybu posunem minimalizovat detekcí začátku hudby, bude výhodnější parametr *mirmode* použít, protože přináší značně lepší výsledky při identifikaci signálu s přidáním šumem.

3.11. Skript pro nalezení neznámé nahrávky

Skript *hledatNeznamou.m* (viz Příloha E a obrázek 6) slouží k hledání neznámé nahrávky v databázi. Uživatel vybere neznámý zvukový soubor v nabídce. Pokud je zadaný soubor příliš krátký, po uživateli se požaduje delší nahrávka. Program poté upraví zadaný zvuk na potřebnou délku. Pokud je signál po detekci začátku hudby kratší než 10 sekund, je použito posledních 10 sekund nahrávky a uživatel je na tento fakt upozorněn. Následně jsou vypočteny parametry neznámého zvuku a podobnost tohoto zvuku s nahrávkami v databázi. Uživateli je poté nabídnuto 5 názvů nejpodobnějších nahrávek v databázi spolu s mírou odlišnosti od zadané nahrávky (viz Příloha F).

Pro chod vyhledávání stačí skript *hledatNeznamou.m* a soubor obsahující databázi parametrů nahrávek, s kterými chceme neznámou nahrávku porovnávat. Pro případ poskytnutých nahrávek je to soubor *delsi_25max.mat*. V samotném skriptu lze upravit počet nejbližších shod, které jsou vypsány.

3.12. Výpočetní náročnost

Zhodnoťme nyní výpočetní náročnost dílčích kroků. Detekce začátku, zkrácení a uložení nahrávky trvá 124 sekund pro 264 nahrávek. Výpočet parametrů a tvorba databáze trvá velmi dlouho, pro 264 nahrávek je to 10 minut a 11 sekund. Vyhodnocení úspěšnosti určení, tj. výpočet nejbližších shod pro 264 nahrávek, trvá 0,7 sekundy. Jelikož jsou toto dílčí kroky vyhledávání neznámé nahrávky, sečtením jejich délky a vydělením počtem nahrávek získáme odhadovanou dobu trvání vyhledávání neznámé nahrávky. Po zadání umístění souboru budou nejbližší shody vypsány průměrně za 2,8 sekundy. Výpočty probíhaly na 2-jádrovém procesoru s taktem 2,5 GHz.

4. Závěr

Výsledný program po zadání neznámé nahrávky vypíše 5 nejbližších shod v databázi. Úspěšnost určení je velmi závislá na zkreslení zadané nahrávky vůči nahrávkám v databázi.

Vliv záporného časového posunu (nahrávka začíná moc brzy) je omezen detekcí začátku hudby. U kladného časového posunu (nahrávka začíná pozdě) se úspěšnost určení výrazně snižuje. Program je proto vybaven ochranou. Pokud je signál po detekci začátku kratší než 10 sekund, použije se posledních 10 sekund zadané nahrávky. Uživatel je upozorněn, že výsledky nemusí být přesné, a požádán o delší nahrávku.

Vliv přidaného bílého Gaussovského šumu již není tak výrazný. Pro signál s odstupem signálu od šumu 50 dB, což je typická hodnota SNR pro horší kategorii gramofonových přehrávačů, je úspěšnost určení nahrávky 82 %. Pokud uvažujeme přehrávač s odstupem signálu od šumu 70 dB, získáme úspěšnost 96 %.

Vliv přeslechů mezi levým a pravým kanálem je menší než u předchozích poruch nahrávek, a to hlavně díky podobnosti levého a pravého kanálu ve většině nahrávek. Pro odstup hlavního kanálu od druhého 20 dB i 40 dB je úspěšnost identifikace shodná – 92 % neznámých nahrávek se nachází mezi 5 nabídnutými.

Nově přidaný parametr *mirmode* zapříčiní horší úspěšnost určení u nahrávek s časovým posuvem. Vliv časového posunu je ovšem omezen detekcí začátku hudby. Parametr byl ve výpočtech podobnosti nahrávek ponechán, protože přináší značně lepší výsledky úspěšnosti určení pro signály poškozené přidaným bílým Gaussovským šumem.

Při testech porovnávání poškozených signálů s databází má systém vysokou úspěšnost. Jelikož byl zvolen vysoký vliv artefaktů, úspěšnost určení neznámé nahrávky by měla být vyšší. Pro správné otestování systému by bylo potřeba použít nově nahrané ukázky, jelikož všechny zkoušky probíhaly s upravenými originálními nahrávkami, ze kterých se také počítali parametry databáze.

Seznam literatury

- [1] Badenhausen, R., The Genesis Of Vinyl Record, 2011
http://www.badenhausen.com/VSR_History.htm
- [2] Foote, J. T., Content-Based Retrieval of Music and Audio, National University of Singapore, 1997
<http://www.music.mcgill.ca/~ich/classes/mumt614/Query%20Retrieval/footespie97.pdf>
- [3] Gettin' Serious About Vinyl: Wow, Flutter, and S/N Ratio, 120studio.com, 2015
<http://120studio.com/audio/wowflutter.htm>
- [4] Gramophone record, Wikipedia.org, 2016
https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Plattenschrift_en.svg/1000px-Plattenschrift_en.svg.png
- [5] Hellerstein, J. M., Stonebraker, M., Readings in Database Systems, The MIT Press, 2005
https://books.google.cz/books?id=7a48qSMuVcUC&printsec=frontcover&dq=database&hl=cs&sa=X&redir_esc=y#v=onepage&q=database&f=false
- [6] Král, V., Rozpoznání obsahu gramofonové desky podle krátké zvukové ukázky, ČVUT v Praze, 2016
- [7] Lartillot, O.; Toiviainen P., A Matlab Toolbox for Musical Feature Extraction from Audio, 10th. Int. Conf. on Digital Audio Effects, Bordeaux, 2007
<http://dafx.labri.fr/main/papers/p237.pdf>
- [8] MIRtoolbox User's Guide 1.6.1
<https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/MIRtoolbox1.6.1guide>
- [9] Neuschmied, H.; Mayer, H.; Batlle, E., Content-based identification of audio titles on the Internet, in First Intern. Conf. on Web Delivering of Music, Proceedings. pp.96-100, 23-24 Nov. 2001 doi: 10.1109/WDM.2001.990163
https://www.researchgate.net/profile/Eloi_Batlle/publication/3939812_Content-based_identification_of_audio_titles_on_the_Internet/links/00463520de02a674fe000000.pdf
- [10] Owen, W. D., Bryson, H. C., Defects in Gramophone Records, The Gramophone, 1931 and 1932
http://www.charm.rhul.ac.uk/history/p20_4_2.html
- [11] St-Laurent, G., The Care and Handling of Recorded Sound Materials, National Library Of Canada, 1996
<http://cool.conservation-us.org/byauth/st-laurent/care.html#fn9>
- [12] Wold, E.; Blum, T.; Keislar, D.; Wheaten, J., Content-based classification, search, and retrieval of audio, in MultiMedia, IEEE , vol.3, no.3, pp.27-36, Fall 1996 doi: 10.1109/93.556537
http://knight.cis.temple.edu/~vasilis/Courses/CIS750/Papers/muscle_fish.pdf

Seznam příloh

Příloha A – tvorba.m

Příloha B – zkraceni.m

Příloha C – vyhodnoceniFunkce.m

Příloha D – vyhodnoceni.m

Příloha E – hledatNeznamou.m

Příloha F – Příklad výpisu skriptu hledatNeznamou.m

Příloha G – CD se soubory – skripty z Matlabu a databáze pro rozdílná poškození

Příloha A – tvorba.m

```
clear;
clc;
%% vytvoreni seznamu skladeb
zvuk = dir('delsi_25max_zpozdeni\');
zvuk(1:2,:) = [];

%% vytvoreni databaze
name = [];
name = char(name);
le = zeros(length(zvuk),1);
tempo = zeros(length(zvuk),1);
br = zeros(length(zvuk),1);
mode = zeros(length(zvuk),1);
for i=1:length(zvuk)
    name = [name; zvuk(i).name]; % nazev skladby
    name1 = ['delsi_25max_zpozdeni\' name(i,:)]; % cesta ke skladbe
    x = miraudio(name1, 'Center', 'Normal'); % nahrani dat,
    centrovani a normalizace
    le(i) = mirgetdata(mirlowenergy(x)); % procento casu, kdy je
    energie signalu pod RMS
    tempo(i) = mirgetdata(mirtempo(x), 'Frame'); % tempo bpm
    br(i) = mirgetdata(mirbrightness(x)); % frekvence, nad kterou je
    53,96% energie
    mode(i) = mirgetdata(mirmode(x)); % tonalita - mol/dur (cislo
    mezi -1 a 1)
end

%% vytvoreni databaze
name2 = name(:,1:7);
data = struct('name', name2, 'le', le, 'tempo', tempo, 'br', br,
    'mode', mode);

%% ulozeni databaze
save('delsi_25max_zpozdeni.mat', 'data');
```

Příloha B – zkraceni.m

```
clear;
clc;

%% nahrani
load zvuk.mat;
name = data.name;
cesta = zeros(264,17);
for i=1:length(name)
    cesta(i,:) = ['delsi\' name(i,:) '.wav'];
end
cesta = char(cesta);

%% zkraceni
for j=1:length(name)
    [temp_zvuk, Fs] = audioread(cesta(j,:));
    temp_zvuk = temp_zvuk(1:Fs*40,:); % zkraceni na 40 sekund
    max_zvuk = max(abs(temp_zvuk)); % nalezeni max amplitudy
    for i=1:length(temp_zvuk)
        if (temp_zvuk(i,:)>0.25*max_zvuk) % nalezeni zacatku
nahravky
            start = i;
            break;
        end
    end
    temp_zvuk = temp_zvuk(start:start+Fs*10-1,:);
    cesta2 = ['delsi_25max\' name(j,:) '.wav'];
    audiowrite(cesta2, temp_zvuk, Fs);
end

%% zkraceni + awgn
for j=1:length(name)
    [temp_zvuk, Fs] = audioread(cesta(j,:));
    temp_zvuk = awgn(temp_zvuk, 70);
    temp_zvuk = temp_zvuk(1:Fs*40,:); % zkraceni na 40 sekund
    temp_zvuk = awgn(temp_zvuk, 50); % pridani sumu
    max_zvuk = max(abs(temp_zvuk)); % nalezeni max amplitudy
    for i=1:length(temp_zvuk)
        if (temp_zvuk(i,:)>0.25*max_zvuk) % nalezeni zacatku
nahravky
            start = i;
            break;
        end
    end
    temp_zvuk = temp_zvuk(start:start+Fs*10-1,:);
    cesta2 = ['awgn50\' name(j,:) '.wav'];
    audiowrite(cesta2, temp_zvuk, Fs);
end
```



```

%% zkraceni + crosstalk
for j=1:length(name)
    [temp_zvuk, Fs] = audioread(cesta(j,:));
    temp_zvuk = temp_zvuk(1:Fs*40,:); % zkraceni na 40 sekund
    levy = temp_zvuk(:,1);
    pravy = temp_zvuk(:,2);
    temp_zvuk = [levy+pravy*(0.01) pravy+levy*(0.01)]; % pricteni
    opacneho kanalu s odstupem 20 dB
    temp_zvuk = temp_zvuk/max(abs(temp_zvuk));
    max_zvuk = max(abs(temp_zvuk)); % nalezeni max amplitudy
    for i=1:length(temp_zvuk)
        if (temp_zvuk(i,:)>0.25*max_zvuk) % nalezeni zacatku
nahravky
            start = i;
            break;
        end
    end
    temp_zvuk = temp_zvuk(start:start+Fs*10-1,:);
    cesta2 = ['crosstalk40\' name(j,:) '.wav'];
    audiowrite(cesta2, temp_zvuk, Fs);
end

%% zkraceni + posuv
for j=1:length(name)
    [temp_zvuk, Fs] = audioread(cesta(j,:));
    temp_zvuk = temp_zvuk(1:Fs*40,:); % zkraceni na 40 sekund
    max_zvuk = max(abs(temp_zvuk)); % nalezeni max amplitudy
    for i=1:length(temp_zvuk)
        if (temp_zvuk(i,:)>0.25*max_zvuk) % nalezeni zacatku
nahravky
            start = i+Fs/10; % zacatek nahravky o sekundu zpozdeny
            break;
        end
    end
    temp_zvuk = temp_zvuk(start:start+Fs*10-1,:);
    cesta2 = ['delsi_25max_zpozdeni\' name(j,:) '.wav'];
    audiowrite(cesta2, temp_zvuk, Fs);
end

```

Příloha C – vyhodnoceniFunkce.m

```
function [ ano ] = vyhodnoceniFunkce( pozice, pocet, znehodnoceni )

if(znehodnoceni == 1)
    load awgn50_25max.mat;
end
if(znehodnoceni == 2)
    load awgn70_25max.mat;
end
if (znehodnoceni == 3)
    load crosstalk20_25max.mat;
end
if (znehodnoceni == 4)
    load crosstalk40_25max.mat;
end
if (znehodnoceni == 5)
    load delsi_25max_zpozdeni.mat;
end
nahravka = pozice;
neznam_name = data.name(nahravka,:);
neznam_le = data.le(nahravka);
neznam_tempo = data.temp(nahravka);
neznam_br = data.br(nahravka);
neznam_mode = data.mode(nahravka);

%% vypocet relativniho rozdilu slozek
load delsi_25max.mat;
relat_rozdil = zeros(length(data.name),5);
for i=1:length(data.name)
    relat_rozdil(i,1) = i;
    relat_rozdil(i,2) = abs((data.le(i)-neznam_le)/data.le(i)*100);
    relat_rozdil(i,3) = abs((data.temp(i)-
neznam_tempo)/data.temp(i)*100);
    relat_rozdil(i,4) = abs((data.br(i)-neznam_br)/data.br(i)*100);
    relat_rozdil(i,5) = abs((data.mode(i)-
neznam_mode)/data.mode(i)*100);
end

%% celkovy relativni rozdil - prumer
relat_rozdil_prum = zeros(length(relat_rozdil),2);
for i=1:length(relat_rozdil)
    relat_rozdil_prum(i,1) = i;
end
relat_rozdil_prum(:,2) = mean(relat_rozdil(:,2:5), 2); % prumer
rozdilu v kazdem radku
nejblizsi_prumer = sortrows(relat_rozdil_prum, 2);
nejblizsi_prumer = nejblizsi_prumer(1:pocet); % pocet nejblizsich

%% byla nalezena nahravka v pocet nejmensich vzdalenostech
ano = 0;
for i=1:pocet
    if(nejblizsi_prumer(i) == pozice)
        ano = 1;
        break;
    end
end
end
end
```

Příloha D – vyhodnoceni.m

```
clear;
clc

%% volba parametru
znehodnoceni = 5; % 1 - awgn50, 2 - awgn70, 3 - crosstalk20, 4 -
crosstalk 40, 5 = zpozdeny original
pocet = 3; % pocet nejblizsich shod

%% pocet nalezenych
nalezeno = 0;
for i=1:264
    nalezeno = nalezeno + vyhodnoceniFunkce(i, pocet, znehodnoceni);
end

%% pocet urcenyh nahravek v pocet nejblizsich
disp('Pro zvolene znehodnoceni a pocet nejblizsich shod bylo
nalezeno')
disp(nalezeno)
disp('z 264 nahravek. To je v procentech')
disp(nalezeno/264*100)
```

Příloha E – hledatNeznamou.m

```
clear;
clc;

%% vyber souboru
pocet = 5; % pocet zobrazenych nejblizsich shod
zprava = 0; % zprava pro uzivatele

fprintf('Vyberte neznamou zvukovou nahravku dlouhou alespon 40
sekund.\n\n');
vstup = 1;
while (vstup) % prijme pouze nahravku delsi nez 40 sekund
    [name, cesta, index] = uigetfile('.wav', 'Vyberte neznamou
zvukovou nahravku'); % nacte nazev a cestu k souboru
    cesta = [cesta name];
    [zvuk, Fs] = audioread(cesta);
    if (length(zvuk)>40*Fs)
        vstup = 0;
    else
        fprintf('Nahravka je prilis kratka, prosim vyberte nahravku
dlouhou alespon 40 sekund.\n\n');
    end
end

%% uprava souboru
zvuk = zvuk(1:Fs*40,:); % zkraceni na 40 sekund
max_zvuk = max(abs(zvuk)); % nalezeni max amplitudy
for i=1:length(zvuk)
    if (zvuk(i,:)>0.25*max_zvuk) % nalezeni zacatku nahravky
        start = i;
        break;
    end
end
if (start+Fs*10-1>length(zvuk)) % osetreni nahravky kratsi nez 10
sekund po detekci zacatku
    start = length(zvuk)-10*Fs;
    zprava = 1;
    fprintf('Nahravka je po detekci zacatku kratsi nez 10 sekund,
zacatek byl posunut. Vysledky nemusi byt spravne!\n\n');
end
zvuk = zvuk(start:start+Fs*10-1,:); % zkraceni na 10 sekund
audiowrite('temp.wav', zvuk, Fs); % vytvoreni docasneho souboru
clear zvuk max_zvuk start; % vycistení pameti

%% nacteni upravene nahravky
zvuk = miraudio('temp.wav', 'Center', 'Normal'); % nacteni nahravky
jako objekt miraudio
delete('temp.wav'); % smazani docasneho souboru

%% vypocet parametru
neznam_le = mirgetdata(mirlowenergy(zvuk)); % procento casu, kdy je
energie signalu pod RMS
neznam_tempo = mirgetdata(mirtempo(zvuk), 'Frame'); % tempo bpm
neznam_br = mirgetdata(mirbrightness(zvuk)); % frekvence, nad kterou
je 53,96% energie
neznam_mode = mirgetdata(mirmode(zvuk)); % tonalita - mol/dur (cislo
mezi -1 a 1)
```

```

%% vypocet reativniho rozdilu slozek
load delsi_25max.mat; % nacteni databaze
relat_rozdil = zeros(length(data.name),5);
for i=1:length(data.name) % vypocet odchylek mezi neznamou nahravkou
a kazdou nahravkou v databazi
    relat_rozdil(i,1) = i;
    relat_rozdil(i,2) = abs((data.le(i)-neznam_le)/data.le(i)*100);
    relat_rozdil(i,3) = abs((data.tempo(i)-
neznam_tempo)/data.tempo(i)*100);
    relat_rozdil(i,4) = abs((data.br(i)-neznam_br)/data.br(i)*100);
    relat_rozdil(i,5) = abs((data.mode(i)-
neznam_mode)/data.mode(i)*100);
end

%% celkovy relativni rozdil - prumer
relat_rozdil_prum = zeros(length(relat_rozdil),2);
for i=1:length(relat_rozdil) % tvorba oznaceni
    relat_rozdil_prum(i,1) = i;
end
relat_rozdil_prum(:,2) = mean(relat_rozdil(:,2:5), 2); % prumer
rozdilu v kazdem radku

nejblizsi_prumer = sortrows(relat_rozdil_prum, 2); % poskladani od
nejnizsiho prumeru
nejblizsi_prumer = nejblizsi_prumer(1:pocet); % ulozi vybrany pocet
nejblizsich

%% vypis vysledku
zobraz = zeros(pocet,21);
fprintf('\nHledana nahravka %s je podobna nahravce\n', name)
for i=1:pocet
    fprintf('%s s odlisnosti %.2f %s\n',
[data.name(nejblizsi_prumer(i),:) '.wav'],
relat_rozdil_prum(nejblizsi_prumer(i),2), '%');
end
if (zprava == 1)
    fprintf('Nahravka musela byt oriznuta, pro lepsi vysledky
pouzijte delsi nahravku.\n');
end

```

Příloha F – Příklad výpisu hledatNeznamou.m

Vyberte neznamou zvukovou nahravku dlouhou alespon 40 sekund.

```
Computing miraudio related to temp.wav...
Computing mirsum related to temp.wav...
Elapsed time is 0.604178 seconds.
Computing mirrms related to temp.wav...
Computing mirlowenergy related to temp.wav...
Computing mirfilterbank related to temp.wav...
Computing mirenvelope related to temp.wav...
Computing mirenvelope related to temp.wav...
Computing mironsets related to temp.wav...
Computing mirenvelope related to temp.wav...
Computing mirenvelope related to temp.wav...
Computing mirsum related to temp.wav...
Computing mirenvelope related to temp.wav...
Computing mirautocor related to temp.wav...
Computing mirpeaks related to temp.wav...
Computing mirpeaks related to temp.wav...
Computing mirsum related to temp.wav...
Computing mirpeaks related to temp.wav...
Computing mirtempo related to temp.wav...
Computing mirspectrum related to temp.wav...
Computing mirbrightness related to temp.wav...
Computing mirrms related to temp.wav...
Computing mirspectrum related to temp.wav...
Computing mirchromagram related to temp.wav...
Computing mirkeystrength related to temp.wav...
Computing mirmode related to temp.wav...
```

```
Hledana nahravka 136_545.wav je podobna nahravce
136_545.wav s odlisnosti 0.10 %
142_035.wav s odlisnosti 8.12 %
147_168.wav s odlisnosti 14.22 %
147_568.wav s odlisnosti 14.66 %
```