



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**STROJOVÝ PŘEPIS KYTAROVÝCH MELODIÍ DO
TABULATURY**

COMPUTER AIDED TRANSFORMATION OF GUITAR SOLOS FROM RECORDED SONG TO TABS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JURAJ JOŠČÁK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2019

Zadání diplomové práce



19128

Student: **Joščák Juraj, Bc.**
Program: Informační technologie Obor: Inteligentní systémy
Název: **Strojový přepis kytarových melodií do tabulatury**
Computer Aided Transformation of Guitar Solos from Recorded Song to Tabs
Kategorie: Umělá inteligence

Zadání:

1. Seznamte se se současnými metodami analýzy signálů z elektrofonických hudebních nástrojů a jejich dekódování do notového zápisu nebo kytarové tabulatury.
2. Prostudujte metody zpracování signálů, jejich transformace a analýzy, které by umožnily dekódovat sekvence tónů v hudebních nahrávkách, pokud v nich dominuje kytarová melodická linka (sólo).
3. Navrhněte systém, který dekóduje melodie z takových nahrávek a vytvoří pro ně tabulátorový zápis.
4. Implementujte program, ve kterém uplatníte tyto metody a ověřte jejich funkčnost na vhodně zvolených nahrávkách v porovnání s poskytnutými tabulaturami (například ze stránky Ultimate Guitar). Použijte jednak nahrávky, které obsahují jen jednu melodii, tak i nahrávky, ve kterých jsou obsaženy i doprovodné nástroje.
5. Diskutujte přínosy a nedostatky zvolených metod a navrhněte, jak by se dalo postupovat směrem ke kvalitnějšímu řešení tohoto problému.

Literatura:

- Beauchamp, J. W.: Analysis, synthesis, and perception of musical sounds. Springer. 2007.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Zbořil František, doc. Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 22. května 2019

Datum schválení: 1. listopadu 2018

Abstrakt

Táto práca sa zaoberá problémom automatickej detekcie tónu v melodických gitarových linkách a následného prepisu do tablatúry. Výsledný systém používa detekciu tónu pomocou hrebeňového filtra. Jednotlivé noty sú rozdelené pomocou algoritmu na detekciu rytmu. Navrhnutý je algoritmus na prepis nôt do gitarovej tablatúry, založený na minimalizácii pohybu ruky.

Abstract

The aim of this thesis was automatic pitch detection in melodic guitar lines and subsequent transcription to guitar tablature. Final system uses comb filtering to detect pitch. Individual notes are separated by beat detection. An algorithm for transcription of notes to guitar tablature, based on minimalization of hand movement is proposed.

Kľúčové slová

rozpoznávanie tónu, gitara, Fourierova transformácia, FFT, audio, tablatúra

Keywords

pitch detection, guitar, Fourier transform, FFT, audio, tablature

Citácia

JOŠČÁK, Juraj. *Strojový prepis kytarových melodií do tabulatury*. Brno, 2019. Diplomová práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. František Zbořil, Ph.D.

Strojový přepis kytarových melodií do tabulatury

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána Doc. Ing. Františka Zbořila Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Juraj Joščák
21. mája 2019

Podakovanie

Týmto by som rád podakoval Doc. Ing. Františkovi Zbořilovi Ph.D. za vedenie, jeho čas a cenné rady pri tvorbe tejto diplomovej práce. Ďalej by som rád podakoval asociácii LPR za materiálnu pomoc, a taktiež Ing. Pavlovi Richtarikovi a Bc. Jurajovi Medvecovi za pomoc pri testovaní.

Obsah

1	Úvod	2
2	Analýza a doterajší stav	3
2.1	Audio signál a tablatúra	3
2.2	Detekcia sóla	5
2.3	Izolácia sóla	7
2.4	Rozdelenie jednotlivých nôt	9
2.5	Detekcia výšky tónu	12
2.6	Poloha prstov	16
2.7	Kompletné riešenia	17
2.8	Anvil Studio, Finale PrintMusic	19
3	Návrh riešenia	20
3.1	Detekcia tempa	21
3.2	Detekcia sóla	22
3.3	Izolácia sóla	23
3.4	Detektor tónu	23
3.5	Poloha prstov	25
4	Implementácia	27
4.1	Implementačné nástroje a technológie	27
4.2	Štruktúra aplikácie	28
5	Testovanie a vyhodnotenie	31
5.1	Data	31
5.2	Vyhodnotenie	32
5.3	Izolácia sóla	32
5.4	Detekcia tempa	32
6	Záver	35
	Literatúra	36
A	Ovládanie programu	39
B	Obsah priloženého pamäťového média	40

Kapitola 1

Úvod

Transkripčia, teda konverzia akustického hudobného signálu do nejakej formy vizuálneho notového zápisu, je netriviálnou úlohou aj pre skúsených hudobníkov. Je užitočná pri štúdiu hudby, taktiež sa používa pri reinterpretácii hudby, v prípade že originálny notový zápis danej skladby nie je z nejakého dôvodu k dispozícii, prípadne nikdy neexistoval (napríklad pri improvizovanej hudbe). Automatická transkripčia je mnohými považovaná za kľúčovú technológiu v oblasti spracovania akustických hudobných signálov. V posledných rokoch sa tomuto problému venuje značné množstvo úsilia kvôli mnohým možným aplikáciám, ako napríklad automatické vyhľadávanie a anotácia hudobných informácií, interaktívne hudobné systémy (napríklad zapojenie počítača do predstavenia so živými hudobníkmi, určenie aktuálnej pozície v skladbe, či sledovanie rytmu), ako aj muzikologická analýza. Existuje už niekoľko rôznych úspešných riešení, plne funkčný systém schopný bezchybnej transkripcie je však stále otázkou budúcnosti.

V tejto práci sa zameriavam na transkripciu gitarových melodických liniek - sól. Cieľom je preskúmať vhodné metódy a technológie a na ich základe vytvoriť program/nástroj, ktorý z dopredu nespracovanej nahrávky v niektorom štandardnom audio formáte viac-menej automaticky extrahuje gitarové sólo, dekoduje jeho melódiu a vytvorí pre neho tablatúrový zápis. Dôraz je kladený na západnú hudbu v rovnomerne temperovanom ladení, teda hudba v ktorej sa gitara používa najčastejšie.

Takýto program by mohol byť užitočný napríklad pre študentov hry na gitare, ktorí by sa pri pokuse reprodukovať dané sólo nemuseli spoliehať len na vlastné uši alebo tablatúry či noty od iných ľudí, ktoré sa občas ťažko hľadajú. Taktiež ľuďom, ktorí tieto tablatúry vytvárajú a publikujú (napríklad na stránkach ako sú ultimate-guitar.com alebo songsterr.com) by ich práca bola uľahčená, keďže doteraz to robili ručne. V neposlednom rade by určite našiel využitie pri skladaní hudby, najmä medzi hudobníkmi, ktorí sú zvyknutí improvizovať.

Kapitola 2 sa venuje teoretickej analýze, sú určené a pomenované jednotlivé podproblémy. Rozoberajú sa rôzne metódy spracovania signálu a detekcie charakteristík ako je výška tónu či rytmus. Diskutuje sa vhodnosť jednotlivých metód pre tento projekt. Preskúmané sú tiež už existujúce riešenia a skúma sa ich úspešnosť. Kapitola 3 popisuje návrh môjho vlastného riešenia. Udáva, ktoré metódy som sa rozhodol použiť, ako konkrétne ich používam, a ďalej približuje ich princíp. Kapitola 4 uvádza niektoré implementačné detaily. Popisuje štruktúru aplikácie, rozdelenie do modulov, použité technológie a vysvetľuje jej celkové fungovanie. Kapitola 5 sa venuje testovaniu. Aplikácia je dôkladne otestovaná na niekoľkých nahrávkach s rôznymi zdrojov a určená je úspešnosť jednotlivých častí, ako aj celková úspešnosť. Diskutované sú možné vylepšenia.

Kapitola 2

Analýza a doterajší stav

Táto kapitola obsahuje najnutnejšiu teóriu potrebnú pre riešenie. Problém prevodu audio signálu na tablatúru je v nasledujúcej podkapitole jasne definovaný, analyzovaný a rozložený na jednotlivé podproblémy. Ďalšie podkapitoly hlbšie rozoberajú postupne každý z týchto podproblémov a uvádzajú niekoľko existujúcich riešení.

2.1 Audio signál a tablatúra

Problémom, ktorý má táto práca riešiť, je prepis gitarových sól, teda audio signálu do tablatúrového zápisu. Audio signálom rozumieme nahranú hudobnú skladbu obsahujúcu (aspoň na niektorých miestach) dominantnú melodickú gitarovú linku. Ďalšie nástroje môžu byť prítomné, ale gitarové sólo musí byť monofonické, nesmie teda obsahovať akordy. V počítači je takýto signál reprezentovaný vzorkovacou frekvenciou a sledom jednotlivých vzoriek (dvojíc vzoriek v prípade stereo signálu). Táto reprezentácia hudby je dnes štandardom a netreba ju viac približovať.

Tablatúra je jedným zo spôsobov, ako graficky znázorniť hudbu, typický pre gitaru (alebo iné prahčové strunové nástroje). Skladá sa zo série za sebou nasledujúcich nôt alebo akordov. Na rozdiel od štandardnej notácie, tieto noty nie sú definované dĺžkou a výškou tónu, ale polohou na hmatníku nástroja. Nejedná sa teda o jednoznačný grafický záznam skladby, skôr ide o akési inštrukcie, podľa ktorých túto skladbu možno zahrať na konkrétnom nástroji. Štandardne sa tablatúry zapisujú do textových súborov pomocou bežných ASCII znakov (pomlčky, čísla, písmená...). Tým je zaručená kompatibilita s veľkým množstvom zariadení.

Ako je vidieť, medzi týmito dvoma reprezentáciami je priam priepastný rozdiel. Zatiaľ čo prevod tablatúry do audia je vcelku triviálnou záležitosťou, opačný prevod vyžaduje niekoľko medzikrokov.

V prvom rade, zaujímajú nás len monofonické gitarové linky. Ak je vstupom nahrávka kompletnej skladby, je treba identifikovať pasáže, ktoré takúto linku obsahujú. Výstupom tohto kroku sú teda časti vstupného signálu reprezentujúce tieto pasáže, prípadne časové značky, určujúce kde sa v pôvodnej nahrávke nachádzajú. Bolo by tiež vhodné, aby boli tieto pasáže zarovnané na takty skladby.

Následne je potrebné gitarové sólo izolovať od ostatných nástrojov, ktoré môžu byť v nahrávke obsiahnuté a ktoré by mohli komplikovať ďalšie kroky. Týmto krokom končí predspracovanie vstupnej nahrávky. Výstupom je audio signál, ktorý už možno priamo analyzovať a extrahovať z neho informácie potrebné na prevod do tablatúry. Predspracovanie

nie je nutné, ak na vstupe dostaneme osamotené gitarové sólo bez ostatných elementov (na internete je možné nájsť izolované jednotlivé stopy populárnych piesní).

Ďalším krokom je rozdelenie sóla do jednotlivých nôt. Zrejme hlavným rozdielom medzi digitálnym audio signálom a tablatúrou sú ich hlavné stavebné prvky. V prípade audio signálu sú to jednotlivé vzorky, v prípade tablatúry sú to práve jednotlivé noty. Rozdelenie sledu vzoriek do nôt je teda pre túto prácu kritické. Jednou notou rozumieme úsek audio signálu, v ktorom hrá nepretržite jeden nemenný tón. Keďže tablatúry neobsahujú žiadne časové informácie, dĺžkou nôt sa nie je potrebné zaoberať. Ďalej je potrebné pre každú notu určiť výšku tónu. To znamená detekovať fundamentálnu frekvenciu a zistiť, kde sa nachádza v hudobnej stupnici.

Posledným krokom je transformácia na gitarovú tablatúru. V momente, keď už je známa postupnosť jednotlivých nôt, je možné bez ďalších problémov vyprodukovať štandardný notový zápis (v skutočnosti potrebujeme poznať aj dĺžku nôt). Každý tón totiž odpovedá práve jednej pozícii na notovej osnove. V prípade gitarovej tablatúry sa však na takúto jednoznačnosť nemožno spoliehať a okrem výšky tónu je nutné zobrať do úvahy ešte niekoľko faktorov. Ako už bolo spomenuté, tablatúra je v skutočnosti návodom, ako hrať skladbu na konkrétnom nástroji. Jednotlivé nástroje sa však jeden od druhého líšia, najčastejšie ladením, ale aj počtom strún, počtom pražcov a inými charakteristikami. Je teda nutné určiť niekoľko parametrov, ktoré definujú nástroj a ladenie, pre ktoré chceme tablatúru vyprodukovať. Ďalším faktorom je samotný hudobník. Keďže jednotlivé tóny možno zahrať na viacerých miestach na hmatníku, každé sólo možno zahrať niekoľkými spôsobmi (vid. 2.1). V tomto kroku teda treba vygenerovať všetky možné spôsoby a tie menej vhodné eliminovať. Treba však pamätať na to, že objektívne najvhodnejší spôsob často neexistuje. Výstupom posledného kroku je teda malý počet tablatúr (v najlepšom prípade len jedna).

 -----0-----

 ---7-----
 | -0-----6-5-

 ---2-5-----
 -----1-0-
 | -0-----

 | -0-12-14-6-5-

Obr. 2.1: Všetky tri tablatúry opovedajú jedinému notovému zápisu

Nie všetky tieto kroky sú nutne závislé jeden na druhom. Niektoré je možné v určitých prípadoch vynechať, je však potrebné zistiť, kedy sa o takýto prípad jedná. Takisto môže ktorýkoľvek s týchto krokov zlyhať, ak nie je vstupná nahrávka nejakým spôsobom v poriadku. Každý z týchto krokov je podrobne rozobratý v nasledujúcich podkapitolách

2.2 Detekcia sóla

Hoci väčšina poslucháčov dokáže pomerne ľahko identifikovať gitarové sólo, jeho definícia nie je úplne samozrejímavá. Napríklad gitarový riff a gitarový lick sú improvizatívne techniky podobné sólu, ale mierne odlišné z hľadiska použitia a štruktúry. Tieto rozdiely je treba formálne definovať.

Peterschmitt et al. definujú sólo ako “sekcia skladby, v ktorej je daný nástroj v popredí v porovnaní s ostatnými sekciami a ostatnými nástrojmi” [26]. Táto definícia však neodlišuje sólo od riffu alebo licku, ktoré sú takisto často v popredí. Riff je krátka melodická sekvencia, často opakovaná, ktorá slúži ako doprovod základnej harmonickej štruktúry, zatiaľ čo lick je definovaný ako krátke melodický motív alebo fráza používaná pri improvizácii [6]. Na doplnenie týchto definícií je známe, že väčšina gitarových riffov sa spolieha na akordy a opakujú sa po niekoľkých taktach. Lick je oveľa kratší (menej než 1-2 takty) a spravidla nasleduje vokálne frázy, prípadne iné nástroje. Vzhľadom na tieto skutočnosti, sekcia piesne je gitarovým sólom ak:

- gitara je v popredí v porovnaní s ostatnými nástrojmi
- gitara hrá improvizované melodické frázy, ktoré sa zväčša neopakujú
- sekcia je dlhšia než 2 takty [24]

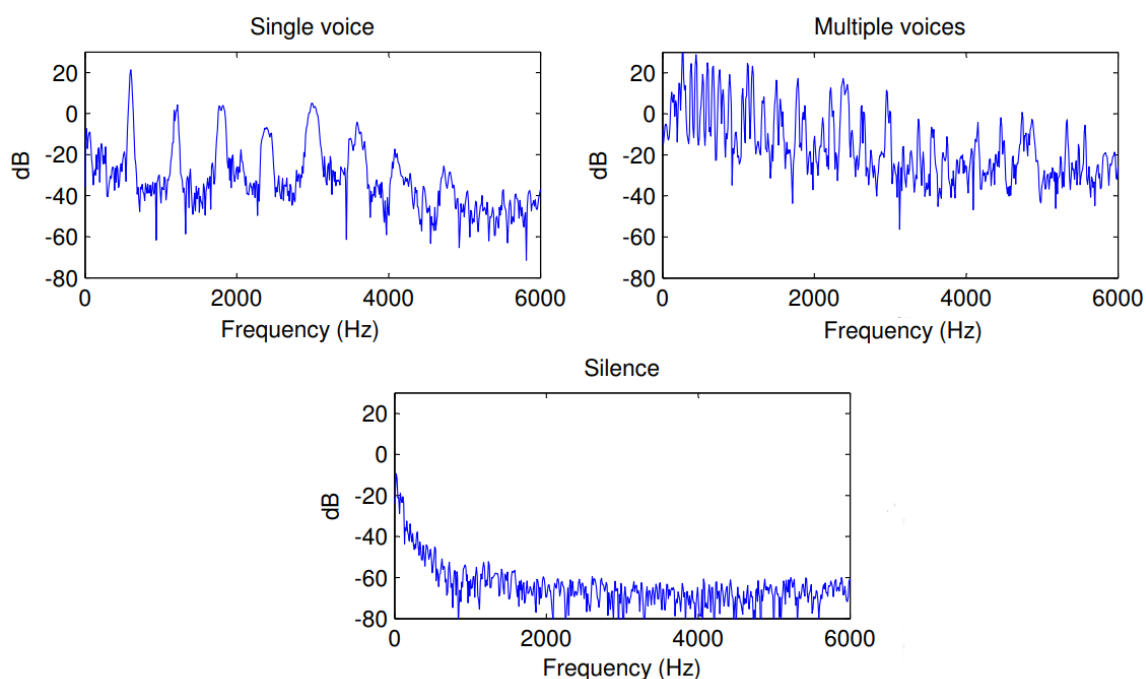
Vo spojení podstate je detekcia sóla klasifikačným problémom, kde sa snažíme úsek piesne zaradiť do jednej z dvoch tried: “obsahuje sólo” a “neobsahuje sólo”. To možno dosiahnuť pomocou niekoľkých metód, zatiaľ sú však všetky do veľkej miery experimentálne a ich využitie je obmedzené.

2.2.1 Metódy založené na spektre

Jednu z prvých štúdií zameraných na detekciu sóla realizovali Peterschmitt et al. pre klasickú hudbu [26]. Hlavnou myšlienkou tejto práce bolo, že spektrá sóla sekcí by mali byť dominované jedným nástrojom. Takéto spektrá by boli menej komplexné (teda mali by menšiu variabilitu lokácií a amplitúd spektrálnych špičiek) než spektrá sekcí bez dominantného nástroja. Miera komplexnosti spektra bola vypočítaná z odchýliek algoritmu Two-Way Mismatch [17]. Tieto hodnoty boli spriemerované pre jednotlivé segmenty skladby získané pomocou Fotevho segmentačného algoritmu [7]. Tento prístup sa ukázal ako problematický pri aplikovaní na nástroje, ktorých spektrá nevykazujú čisté harmonické chovanie. Veľmi dobré výsledky boli dosiahnuté pri práci s veľmi harmonickými nástrojmi, ako je saxofón, trúbka či husle, avšak práve pre gitaru a klavír, algoritmus Two-Way Mismatch sa ukázal ako nespoľahlivý. Ďalší problém tkvel v algoritme pre automatickú segmentáciu, ktorého vstupné parametre sa menili od skladby k skladbe. Vzhľadom na malú veľkosť datasetu neboli poskytnuté žiadne konkrétne kvantitatívne údaje o úspešnosti tohto prístupu.

Smit a Ellis použili iný postup na detekciu sólového spevu v profesionálnej ľudovej a klasickej hudbe [31]. Tento postup bol postavený na myšlienke, že sólová sekcia skladby poväčšine pozostáva z jediného tónu znejúceho v každom časovom okamihu. Spektrálna štruktúra izolovaného tónu je charakteristicky jednoduchá, pozostáva z presne vymedzených, pravidelných rozložených harmonických spektrálnych špičiek. To umožňuje odlíšiť tieto rámce od viacerých nedominantných hlasov znejúcich naraz (ktorých spektrum pozostáva z mnohých prekrývajúcich sa harmonických sérií), alebo od ticha, ktoré obsahuje len určitú mieru šumu (viď obr 1). Jadrom algoritmu je pokus namodelovať krátky časový priebeh

signálu tak, akoby pozostával z jedinej periodicity. Keď sa energia práve tejto frekvencie potlačí pomocou vhodného filtra (napríklad odčítanie signálu oneskoreného o jednu periódu od seba samého), celková energia signálu klesne o určité percento. Ak signál pozostáva prevažne alebo úplne z jedinej periodicity, malo by byť možné potlačiť takmer všetku periodickú (tonálnu) energiu, čo by malo za následok drastický pokles celkovej energie signálu po filtrovaní. Tento algoritmus ponúkol podstatne lepšie výsledky, než systémy ktoré používali iba MFCC (Mel Frequency Cepstral Coefficients) ako príznak, a dosiahol úspešnosť okolo 70%. Vzhľadom na malý dataset však tieto výsledky nemožno považovať za veľmi presvedčivé. Hoci bol tento algoritmus navrhnutý a testovaný len na detekciu sólového spevu, vo svojej podstate je použiteľný na monofonické sóla hrané na akomkoľvek nástroji.



Obr. 2.2: Spektrá jediného dominantného hlasu, viacerých hlasov a ticha (šumu)[31]

Hlavný problém týchto riešení je v samotnom koncepte gitarového sóla. Tieto algoritmy sú jednoduché a používajú nízkoúrovňové deskriptory na extrakciu veľmi abstraktného konceptu. To spôsobuje množstvo inkonzistencií a limituje robustnosť algoritmu. Lokalizácia sóla sa vykonáva za použitia nízkoúrovňových príznakov, čo umožňuje lokalizáciu len vzhľadom na tieto príznaky. To znamená, aby bolo sólo správne detekované musí byť:

- jasné: sólový nástroj musí byť relatívne hlasný v porovnaní s nástrojmi v pozadí.
- nepretržité: ak sa v sóle vyskytujú krátke pauzy, segmentácia nebude pracovať správne.

Sekcie skladby, kde sa sólo nevyskytuje musia byť taktiež veľmi charakteristické. Napríklad ak je jeden motív hraný dvomi nástrojmi v súzvuku, môže byť mylne považovaný za sólo.

2.2.2 Strojové učenie

V posledných rokoch, s nárastom popularity techník strojového učenia, niekoľko štúdií zameraných na detekciu inštrumentálnych sôl bolo vypracovaných za použitia týchto techník.

Fuhrmann et al. použili selekciu príznakov na identifikovanie piatich najlepších príznakov na detekciu sôl v klasickej hudbe. Týmito príznakmi boli miera dominancie tónu (pitch instantaneous confidence), spektrálna disonancia, spektrálny tok, spektrálna plochosť a spektrálna hrebeňovitosť (spectral crest). Z piatich testovaných klasifikátorov bola ako najlepšia vybraná metóda podporných vektorov (SVM). Algoritmus bol otestovaný na datasete pozostávajúcom z 240 audio segmentov (40 segmentov na každý z nástrojov cello, klarinet, hoboj, klavír, saxofón a husle, každý segment pozostávajúci z 15 sekúnd dlhého sóla a 15 sekúnd of ansámbovej hudby). Celková úspešnosť bola 77%. Avšak, pri experimentoch na iných hudobných žánroch, ako je jazz a populárna hudba, úspešnosť algoritmu značne klesla.[9]

Mauch et al. realizovali ďalší výskum za použitia podobných metód na populárnych piesňach. Cieľom bolo detekovať ako inštrumentálne tak aj vokálne sóla, a tiež iné charakteristiky, napríklad pohlavie speváka. V tomto výskume použili štandardné audio príznaky, ako napríklad Mel-frekvenčné cepstrálne koeficienty (MFCC), taktiež ale použili príznaky založené na fluktuácii dominantného tónu, MFC koeficienty resyntetizovanej dominantnej linky a normalizovanú amplitúdu harmonických zložiek. Na klasifikáciu použili dva rôzne modely SVM-HMM (support vector machine hidden Markov model). Algoritmus bol trénovaný a testovaný na datasete 112 populárnych piesní s celkovou úspešnosťou 89.8%. [19]

Foulon et al realizovali štúdiu zameranú čisto na gitary. Cieľom bolo identifikovať rôzne módy hrania na gitare (akordy, sólo, basové linky) v reálnom čase pre potreby systému na automatický hudobný doprovod. Autori použili kombináciu štandardných audio príznakov, a príznakov založených na fundamentálnej frekvencii. Ich dataset obsahoval 84 súborov (takmer jeden a pol hodiny) vytvorených za použitia gitár štyroch rôznych typov (akustických aj elektrických). Ako najlepšia voľba klasifikátoru sa ukázala bayesovská sieť. Hoci priemerná úspešnosť tohto algoritmu bola okolo 95%, bol použitý len zvuk jedinej gitary, a teda výkonnosť tohto riešenia na reálnych skladbách s viacerými nástrojmi nie je známa.[8]

Pati a Lerch sa vo svojej práci zamerali na detekciu elektrických gitarových sôl v rockovej hudbe. Táto práca bola z veľkej časti založená na výskumoch spomenutých vyššie. Okrem spektrálnych a temporálnych príznakov však využíva aj príznaky založené na štruktúrálnej segmentácii skladby (predpokladá sa napríklad, že sólo sa v piesni nebude opakovať). Na účely tejto štúdie bol vytvorený nový anotovaný dataset, ktorý je verejne dostupný. Na klasifikáciu bola znovu použitá metóda SVM. Výsledky sú viac menej podobné, ako výsledky získané v predchádzajúcich výskumoch, ale vzhľadom na rôzne veľkosti datasetov, rozdiel v hudobných žánroch (klasická hudba, populárna hudba, rock) a vzhľadom na to že táto konkrétna práca sa zameriava špecificky na gitarové sóla, priame porovnanie nie je možné.[24]

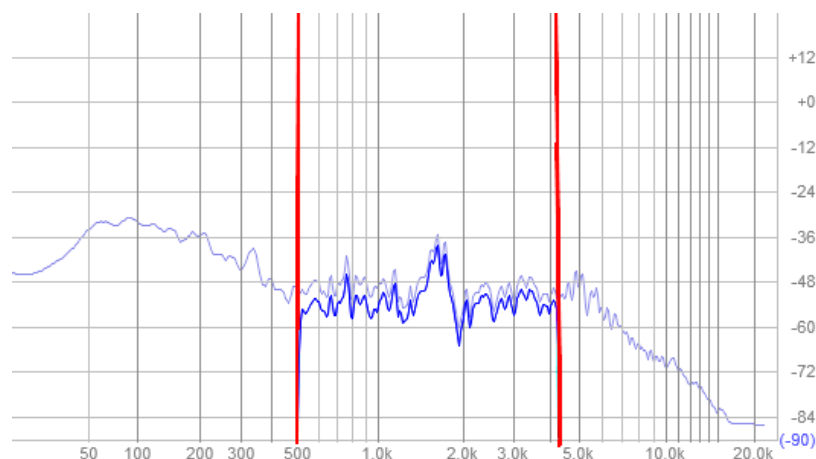
2.3 Izolácia sóla

V nasledujúcich krokoch nás zaujíma iba gitarové sólo. Aby sa predišlo rôznym interferenciám od ostatných nástrojov hrajúcich v rovnakom čase, je vhodné od nich gitarové sólo izolovať. Ide o problém, kde je nutné získať niekoľko nezávislých signálov z jediného signálu vytvoreného ich zmiešaním, známy ako "fenomén koktejlovej párty". Niekoľko metód, kolektívne známych ako BSS (blind source separation), využíva porovnávanie signálov z

niekoľkých mikrofónov rozmiestnených v blízkosti zdrojov zvuku, ktoré sa majú izolovať. Vzhľadom na to, že drvivá väčšina hudobných nahrávok v súčasnosti je v stereo forme, bude sa táto práca zaoberať len metódami, ktoré na svoje fungovanie využívajú maximálne dva kanály.

2.3.1 Ekvalizácia

Najjednoduchšia je založená na fakte, že zvuk gitary sa vyskytuje v určitom frekvenčnom rozsahu. Takisto je veľmi častým javom, že gitarové sólo používa vyššie tóny, než doprovodná gitarová linka. Je nám teda známy frekvenčný rozsah, v ktorom gitarové sóla dominujú a odfiltrovaním frekvencií mimo tento rozsah sa môžeme zčasti zbaviť nechceného vplyvu ostatných nástrojov.



Obr. 2.3: Frekvenčný rozsah v ktorom väčšinou dominujú rockové gitarové sóla

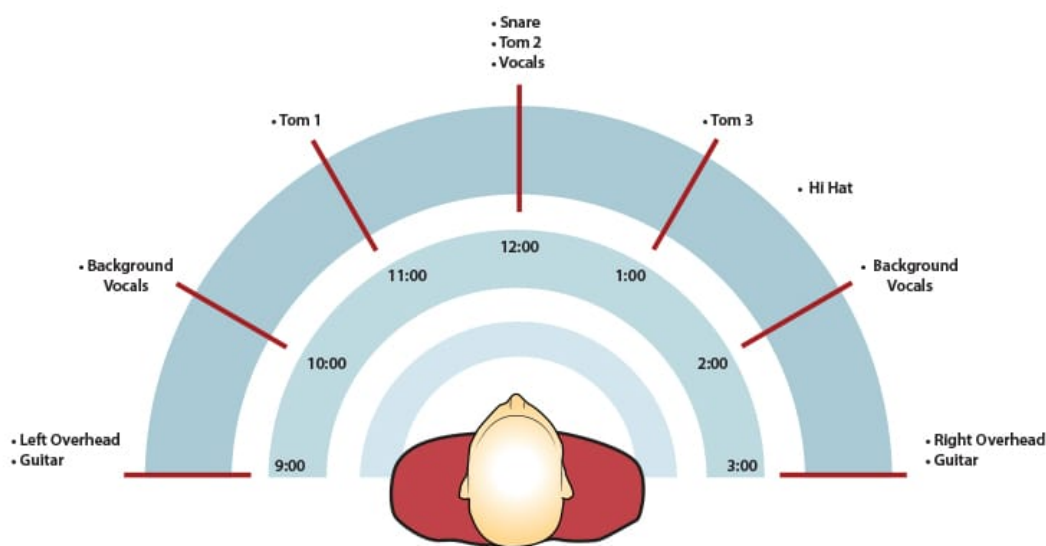
Výhodou tejto metódy je jej jednoduchosť. K implementácii stačí len niekoľko (v najlepšom prípade jeden) lineárnych filtrov. Nevýhodou je fakt, že aj keď je sólo dominantné v istom frekvenčnom rozsahu, fundamentálne frekvencie jednotlivých tónov môžu ležať mimo neho a filtrovaním sú odstránené. Následná detekcia tónu musí byť schopná odhadnúť fundamentálnu frekvenciu z vyšších (alebo nižších) harmonických zložiek.

2.3.2 Stereo separácia

Táto metóda je založená na spôsobe, ako sú moderné skladby mixované. Spev a niektoré nástroje je zvykom umiestniť do stredu (signál prispieva rovnakou mierou do pravého aj ľavého stereo kanálu), iné sa zas umiestňujú viac na jednu stranu, prípadne sa tá istá linka nahrá pre každý kanál zvlášť, čím sa vytvorí hutnejší zvuk. Toto možno využiť k oddeleniu jednotlivých nástrojov, napríklad odčítaním jedného kanála od druhého sa oslabia nástroje hrajúce viac v strede a podobne.

Nevýhodou tejto metódy je, že predpokladá vstupnú nahrávku v určitej forme. Je nutné dopredu vedieť, ako bola zmixovaná a kde sa gitarové sólo nachádza.

Túto metódu používa niekoľko komerčne dostupných aplikácií, napríklad Riffstation[33] alebo Anytune ReFrame[1].



Obr. 2.4: Štandardné umiestnenie nástrojov v stereo poli[11]

2.3.3 Pokročilé metódy

Dôležitým problémom pri separácii zvukov hudobných nástrojov je odhadnutie časovo-frekvenčných oblastí, v ktorých sa harmonické zložky signálov prekrývajú. Li et al. vo svojej práci navrhli monoaurálny systém na separáciu hudobných zvukov, ktorý rieši prekrývanie harmonických zložiek explicitne. Je založený na pozorovaní, že zatiaľ čo všeobecne, absolútna hodnota amplitúdy harmonickej zložky vzhľadom na susedné zložky sa modeluje ťažko, amplitúdové obálky rôznych harmonických zložiek z toho istého zdroja bývajú podobné. Tomuto javu sa hovorí spoločná amplitúdová modulácia (common amplitude modulation - CAM). [14]

Lal a Wang predložili dve metódy pre separáciu hudobných zdrojov zvuku z monoaurálnej zmesi pomocou informácií z izolovaných nôt a spektrálnej šablóny, pričom tieto informácie boli odhadnuté zo zvukovej zmesi. Hlavnou myšlienkou je odhadnutie parametrov každého zdroja zvuku z krátkych izolovaných úsekov vyskytujúcich sa v signále, a následné využitie týchto parametrov na presnenie metódy CAM. [13]

2.4 Rozdelenie jednotlivých nôt

Vstupný signál možno rozdeliť do nôt niekoľkými spôsobmi. Ak môžeme predpokladať, že skladba udržiava viac-menej stabilné tempo, stačí použiť vhodný algoritmus na detekovanie tempa (beat detection) a následne signál rozdeliť na rovnako dlhé úseky, pričom každý predstavuje jednu štvrtovú notu. Pokročilejšie metódy sú schopné priamo detekovať nábeh každej noty (onset detection).

2.4.1 Detekcia tempa

Ľudský sluchový systém určuje rytmus hudby detekovaním pseudo-periodického sledu úderov (beats). Signál zachytený ušom má určitú energiu. Prirodzene čím viac energie zvuk prenáša, tým sa tento zvuk zdá hlasnejším. Úder bude rozpoznávaný, ak je energia zvuku v

krátkom čase výrazne vyššia než bola do tohto momentu, teda ak mozog detekuje ostrú zmenu v energii zvuku. Keďže tempo skladby nasleduje jednotlivé údery (výrazné variácie v energii), môžeme energiu zvuku využiť na detekciu tempa. V rámci tohto algoritmu sa detekujú variácie v energii zvuku vypočítaním priemernej energie signálu a jej porovnaním s okamžitou energiou. Okamžitá energia je energia v niekoľkých málo vzorkoch, teda za krátky časový okamih (niekoľko milisekúnd). V čase i_0 sa vypočíta podľa vzorca 2.1, kde e je okamžitá energia, n je počet vzorkov a $a[k]$ je vzorka v čase k .

$$e[i_0] = \sum_{k=i_0}^{i_0+n} a[k]^2 \quad (2.1)$$

Priemernú energiu nie je vhodné vypočítať na celej piesni, keďže mnoho piesní obsahuje intenzívnejšie aj tichšie pasáže. Okamžitá energia by sa mala porovnávať s priemernou energiou v tesnej blízkosti. Ak má napríklad pieseň intenzívny, hlasný koniec, energia v ňom obsiahnutá by nemala mať vplyv na detekciu úderov na začiatku. Úder detekujeme len keď je okamžitá energia vyššia než lokálna priemerná energia. Priemernú energiu teda vypočítame na za istý časový okamih (v sekundách). Tento časový okamih nesmie byť príliš dlhý, aby sa nebrali do úvahy príliš vzdialené energie, ani príliš krátky, aby sa priemerná energia dala rozumne porovnať s okamžitou energiou. Priemerná energia sa vypočíta podľa vzorca 2.2, kde E je priemerná energia za N predchádzajúcich vzoriek.

$$E[i_0] = \frac{n}{N} * \sum_{k=i_0-N}^{i_0} a[k]^2 \quad (2.2)$$

Následne, ak:

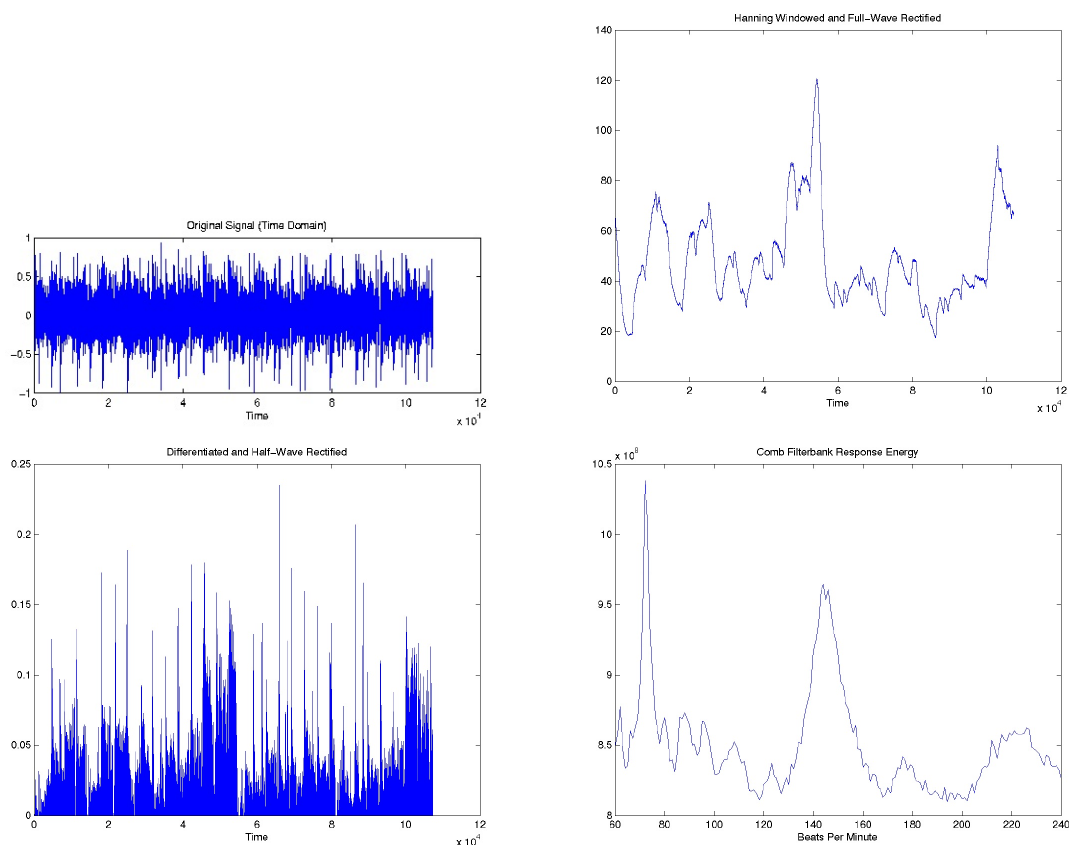
$$e[i] > E[i] * C \quad (2.3)$$

môžeme usúdiť, že v časovom okamihu i sa vyskytuje úder. Problémom je výber konštanty citlivosti C . Pre rôzne nahrávky sú vhodné rôzne hodnoty C podľa toho, ako výrazné sú jednotlivé údery. Algoritmus môže vhodné C určiť automaticky. Stačí vypočítať varianciu energií za N predchádzajúcich vzoriek. Táto variancia je priemerom $e - E$, a pomocou nej možno kvantifikovať, ako výrazné sú jednotlivé údery a teda určiť C . Keď poznáme pozície jednotlivých úderov, môžeme určiť rozostupy medzi nimi, a z najčastejšie sa vyskytujúceho rozostupu určiť tempo skladby. [25]

Podobnú, avšak pokročilejšiu metódu používajú Cheng et al. [3] za účelom synchronizácie rytmu dvoch rôznych skladieb. Ich algoritmus je taktiež založený na zdôraznení prudkých zvukových impulzov v piesni a hľadani základnej frekvencie, s ktorou sa tieto impulzy objavujú. Signál je najprv rozdelený na niekoľko samostatných signálov, pričom každý obsahuje istý frekvenčný rozsah z pôvodného signálu. Tým sa od seba oddelia zvuky produkované rôznymi skupinami nástrojov a môžu sa analyzovať oddelene, aby sa zabránilo prípadným konfliktom úderov. To sa dosiahne fourierovou transformáciou signálu, rozdelením spektra na niekoľko úsekov a následnou inverznou fourierovou transformáciou týchto úsekov.

Následne je sú tieto signály prevedené do podoby, v ktorej možno vidieť prudké zmeny v amplitúde. Toho sa dosiahne redukovaním signálu na jeho obálku. Tá ukazuje prevažný trend v amplitúde, nezávislý na frekvenciách. Najprv sa každý signál dvojcestne usmerní, aby sa ďalej pracovalo len s kladnými číslami. Ďalej sa vykoná konvolúcia každého signálu s pravou polovicou Hanningovho okna. V ďalšom kroku sa vypočíta diferencia signálov. Tá sa jednocestne usmerní, zaujímajú nás totiž len prudké stúpnutia amplitúdy, nie klesnutia.

Nakonies vykoná konvolúcia týchto signálov s rôznymi hrebeňovými filtermi a hľadá sa najvyššia energia. Hrebeňový filter je vlastne séria impulzov, ktoré sa vyskytujú periodicky s určenou frekvenciou. Filtre je vygeneruje pre každé tempo s určeného rozsahu a konvolvujú sa so signálmi. Energie jednotlivých signálov pre každý filter sa sčítajú a tak získame vektor energií pre každé tempo. Maximálna z týchto energií je fundamentálnym tempom, ktoré hľadáme.



Obr. 2.5: (zľava) Pôvodný signál, po konvolúcii Hanningovým oknom, po diferenciacii a energie jednotlivých hrebeňových filtrov[3]

2.4.2 Detekcia nábehu noty

Detekcia nábehov nôt nie je triviálna úloha. Hranice medzi notami a rôznymi typmi udalostí sú často zle definované. Fyzika hudobných nástrojov a nahrávacích prostredí vytvára artefakty, ktoré môžu byť ľahko zamenené za nábehy. Neprekvapujúco, pri práci s polyfonickými signálmi, detekcia nábehov nôt sa ešte viac komplikuje.

Klapuri vo svojej práci[12] navrhol algoritmus na detekciu nábehu nôt založený na energii signálu a poznatkoch z psychoakustiky. Tento algoritmus je do veľkej miery podobný algoritmu z [3] na detekciu tempa, ktorý je popísaný v predchádzajúcej sekcii, až po diferenciaciu signálu. Namiesto obvyčajnej diferencie používa relatívnu diferenciu, čo je diferenciu vydelená amplitúdou v danom bode. To preto, že hlavne tichým zvukom môže chvíľu trvať kým dosiahnú bod v ktorom ich amplitúda maximálne stúpa. Tento bod by tak bol oneskorený oproti skutočnému nábehu a spôsoboval by chyby pri medzipásmovej asociácii

s vyššími frekvenciami. Taktiež, nábeh zvuku väčšinou nie je monotónne stúpajúci, a teda by sme získali niekoľko lokálnych maxím v diferencnej funkcii blízko skutočného nábehu. Špičky v relatívnej diferencnej funkcii nad určitým prahom sú označené ako komponenty nábehu. Následne sa výsledky zo všetkých frekvenčných pásem skombinujú.

Komponenty nábehu z rôznych pásem sa zoradia podľa času a následne ich nazývame, kandidátnymi nábehmi. Každému kandidátnemu nábehu sa priradí hodnota hlasitosti a to tak, že sa zozbierajú komponenty nábehu v krátkom časovom okne okolo kandidátneho nábehu a intenzity sa použijú ako vstup do modelu hlasitosti podľa Moorea et al.[20].

Ako sa ukázalo, robustná detekcia nábehov rôznych druhov sa dá dosiahnuť jednoduchým výberom špičiek, kedy sa hľadajú kandidátne nábehy nad určitým prahom hlasitosti. Kandidátne nábehy, ktorých hlasitosť je pod týmto prahom vyradíme. Takkisto vyradíme kandidátne nábehy, ktoré sú príliš blízko hlasnejšiemu kandidátnemu nábehu. Ak sú rovnako hlasné kandidátne nábehy príliš blízko seba, vyberieme stredný z nich a ostatné vyradíme. Ostávajúce kandidátne nábehy sú akceptované ako skutočné nábehy.

Algoritmy založené na energii signálu sú jednoduché a rýchle. Ich efektivita však klesá, keď prechodné javy v signále nie sú dost' výrazné (napríklad pri neperkusívnych zvukoch), a keď sa nábehy v polyfonických nahrávkach prekrývajú. Bello a Sandler navrhli alternatívny prístup, ktorý používa informácie o fáze[2]. Fáza nesie všetky časové informácie signálu, ale pri spektrálnej analýze je často ignorovaná. Prechodové deje v signále sú presne časovo lokalizované, fázová analýza teda ponúka zmysluplnejšie výsledky než pri spoliehaní sa čisto na energiu signálu. Takisto, fázová analýza naznačuje istú formu tónovej analýzy. Zmeny, ktoré nie sú tak badateľné ako výbuchy energie možno stále detekovať ako výbuchy tónu.

Navrhnutý algoritmus stavia na metóde pre fázovú separáciu prechodových a stabilných stavov(transient/stable-state separation - TSS)[30]. Dáta vyprodukované separačným algoritmom sú analyzované štatistickými metódami.

Algoritmus ponúka dobré výsledky pre sólové nástroje (dokonca aj neperkusné, ako sú husle), s minimom falošných pozitívov. Avšak, s nárastom zložitosti signálov (napríklad jazz a populárna hudba so spevom), vysoká miera detekcie je doprevádzaná väčším množstvom falošných pozitívov. Najhorším prípadom je práve rocková hudba so spevom.

2.5 Detekcia výšky tónu

Robustná detekcia výšky tónu je pre túto prácu kritická. Pod detekciou tónu rozumieme odhadnutie fundamentálnej frekvencie pseudo-periodického signálu, akým je zvuk gitary. Existuje množstvo štandardných metód, ktoré sa používajú na extrakciu fundamentálnej frekvencie f_0 , založené na rôznych matematických princípoch.

2.5.1 Metódy v časovej oblasti

Existuje trieda metód, ktoré sa usilujú zistiť, ako často sa signál plne zopakuje. Hlavná myšlienka za týmito metódami je, že ak je signál periodický, potom existujú extrahovateľné opakujúce sa udalosti, ktoré možno spočítať, a ich počet za sekundu nepriamo súvisí s frekvenciou signálu. Každá z týchto metód je použiteľná pre určité typy signálov. Zero-crossing rate (ZCR)[10] je hodnota, ktorá udáva ako často signál prejde cez nulu za jednotku času. Na prvý pohľad sa zdá, že táto hodnota by mala priamo súvisieť s tým, ako často sa signál zopakuje za jednotku času. Avšak, ak signál obsahuje spektrálne komponenty na vyšších frekvenciách, môže prejsť cez nulu viac než dvakrát za cyklus. ZCR detektor f_0 by mohol fungovať na signále prefiltrovanom dolnou priepustou, aby sa odstránili vysoké

frekvencie, ktoré by mohli meranie kontaminovať. Medznú frekvenciu však treba vyberať opatrne, aby sa odstránilo čo najviac vysokých frekvencií, ale f_0 ostala neporušená. Taktiež by detektor mohol zisťovať vzory v prechodoch cez nulu a ich opakovanie. Podobne, Peak rate počíta kladné špičky za sekundu v signále. Teoreticky, signál má maximálnu a minimálnu hodnotu za každý cyklus, a tie stačí spočítať. V praxi treba použiť detektor lokálnych špičiek. Znovu je problém, ak každá perióda signálu obsahuje viac než jedno lokálne maximum. Možno ešte využiť fakt, že derivácia periodického signálu je sama o sebe periodická s rovnakou frekvenciou. V niektorých prípadoch, prechody nulou a špičky v derivácii signálu môžu byť informatívnejšie, než v originálnom signáli.

Ďalšou možnosťou je použiť autokorelačnú funkciu[34]. Korelácia medzi dvomi signálmi je miera ich podobnosti. Autokorelačná funkcia je funkcia, ktorá určuje koreláciu medzi signálom a jeho časovo oneskorenou kópiou. Dá sa predpokladať, že podobnosť so zvyšujúcim sa časovým oneskorením klesá, pričom maximum dosahuje pri oneskorení 0. 2.4 je matematická definícia autokorelačnej funkcie.

$$R_x(v) = \sum_{n=0}^{N-1-v} x[n]x[n+v] \quad (2.4)$$

Autokorelačná funkcia periodického signálu je zaujímavá tým, že je sama o sebe periodická. Ako časové oneskorenie stúpa smerom k polovici periódy, korelácia klesá na minimum. Ďalej korelácia zase stúpa, až dosiahne maximum, keď je časové oneskorenie rovné dĺžke jednej periódy a pôvodný signál s oneskorenou kópiou sú vo fáze. Prvá špička v autokorelačnej funkcii teda označuje periódu signálu. Znova sa tu však vyskytujú problémy pri použití harmonicky bohatého pseudo-periodického signálu. Autokorelačná funkcia takéhoto signálu obsahuje množstvo menších pred hlavnou, ktorá označuje periódu. V takom prípade sa musí algoritmus rozhodovať medzi malými a veľkými špičkami, čo znižuje jeho robustnosť.

Podobný prístup používa YIN estimátor[34], avšak namiesto autokorelačnej funkcie používa diferenčnú funkciu (2.5), kde ide o minimalizáciu rozdielu medzi signálmi namiesto maximalizácie ich súčinu.

$$D_x(t) = \sum_{n=1}^N (x[n] - x[n+t])^2 \quad (2.5)$$

Na odstránenie subharmonických chýb používa YIN kumulatívny priemer:

$$D'_x(t) = \begin{cases} 1 & t = 0; \\ \frac{D_x(t)}{\frac{1}{t} \sum_{j=1}^t D_x(j)} & \text{else.} \end{cases}$$

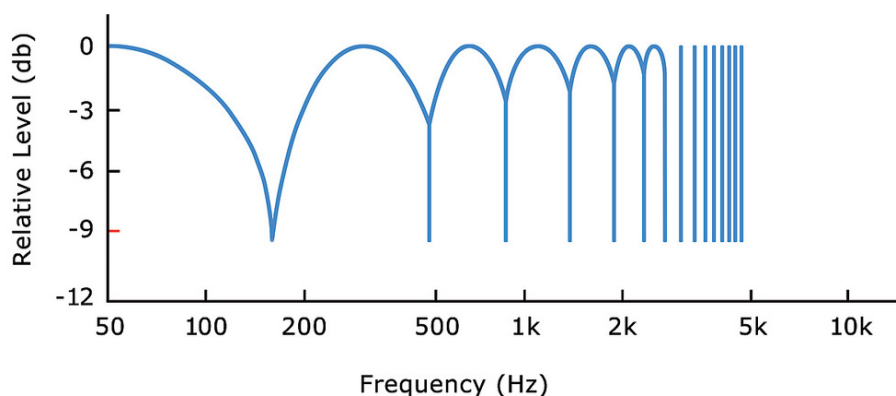
2.5.2 Spektrálne metódy

Vo frekvenčnej oblasti je veľa informácií ktoré môžu súvisieť s fundamentálnou frekvenciou signálu. Hudobné signály sú väčšinou zložené zo série harmonicky súvisiacich zložiek, ktoré možno identifikovať a použiť na odhadnutie f_0

Už v roku 1979, Piszczalski pracoval na kompletnom systéme na transkripciu hudby[27]. Jeho metóda začína spektrálnou transformáciou a identifikáciou jednotlivých zložiek pomocou detekcie špičiek. Pre každé dve z týchto zložiek, algoritmus nájde “najmenšie harmonické čísla”, ktoré odpovedajú harmonickým radám obsahujúcim tieto zložky. Napríklad ak sa zložky vyskytli na 435 Hz a 488 Hz, najmenšie harmonické čísla by boli 6 a 7. Každý z

týchto párov harmonických čísel sa použije ako hypotéza pre fundamentálnu frekvenciu signálu. V tomto prípade, pár zložiek korešponduje s hypotézou, že fundamentálna frekvencia signálu je okolo 70 Hz. Keď sú takto spracované všetky páry zložiek, najsilnejšia hypotéza je zvolená ako fundamentálna frekvencia. "Hlas" niektorých párov harmonických zložiek má väčšiu váhu pri výbere najsilnejšej hypotézy. Táto váha závisí na amplitúde. Páry s vyššou amplitúdou majú väčšiu váhu. Táto metóda dobre funguje aj v prípade prítomnosti neharmonických neprítomnosti niektorých harmonických zložiek.

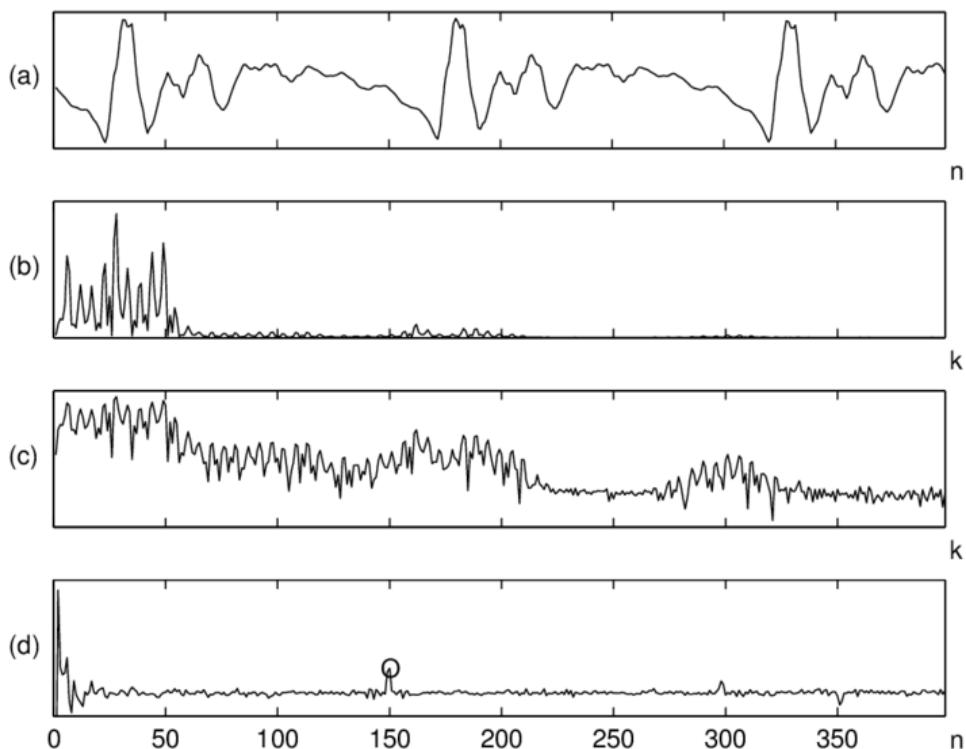
Niekoľko metód je založených na filtrovaní signálu. Optimálny hrebeňový estimátor[21] je robustný, ale výpočetne náročný algoritmus. Hrebeňový filter (viď Obr. 2.6) obsahuje množstvo pravidelne rozmiestnených pásmových priepustí. Napríklad, ak stredová frekvencia prvej priepuste je 10 Hz, potom ďalšie priepuste budú každých 10 Hz až po shannonovu frekvenciu. Algoritmus je založený na filtrovaní vstupného signálu na mnohých frekvenciách. Ak sú v signále prítomné harmonicky súvisiace frekvencie, výstup filtra bude najväčší vtedy, keď sa pásmové priepuste nachádzajú nad týmito frekvenciami. Ak má signál len jednu výraznú frekvenciu, táto metóda zlyhá.



Obr. 2.6: Hrebeňový filter [35]

Ďalšia trieda metód využíva cepstrálnu analýzu[16]. Prirodzene sa vyskytujúce zložky frekvenčného spektra sú často nepatrne neharmonické. Cepstrum sa získa inverznou fourierovou transformáciou logaritmu spektra vstupného signálu, aby sa nelineárny systém stal viac lineárnym. Tieto metódy sa spoliehajú na fakt, že spektrum hudobného signálu obsahuje niekoľko rovnomerne rozložených špičiek reprezentujúcich harmonické spektrum signálu. V prípade logaritmu spektra, tieto špičky sú redukované, ich amplitúda je prevedená do použiteľného rozsahu, a výsledkom je periodické spektrum, ktorého perióda súvisí s fundamentálnou frekvenciou pôvodného signálu. Špička v spektre tohto signálu odpovedá perióde pôvodného signálu. Tak ako mnohé iné metódy estimácie f_0 , aj táto metóda je vhodná len pre špecifické typy signálov. Používa sa hlavne pri rečových signáloch, ktoré sú spektrálne bohaté a majú rovnomerne rozložené zložky.

Každú spektrálnu metódu možno vylepšiť použitím viacerých rozlíšení. Jednoducho, ak je presnosť nejakého algoritmu na určitom rozlíšení pochybná, hypotéza estimátoru sa potvrdí alebo vyvráti použitím rovnakého algoritmu na vyššom alebo nižšom rozlíšení. Každá nové rozlíšenie samozrejme znamená vyššie výpočetné nároky.



Obr. 2.7: (a) pôvodný signál (b) spektrum (c) logaritmus spektra (d) cepstrum [28]

2.5.3 Štatistické metódy

Estimáciu fundamentálnej frekvencie možno v určitom ohľade považovať za štatistický problém. Ide vlastne o klasifikáciu vstupného signálu do triedy podľa f_0 . Niektorí odborníci sa preto domnievajú, že na tento problém možno aplikovať moderné štatistické metódy, ako napríklad neuronové siete. Neuronová sieť na detekciu tónu môže brať na vstupe sadu spektrálnych zložiek, signál v časovej oblasti, alebo fázovú reprezentáciu signálu. Výstupom by bol tón. Sano a Jenkins neuronovú sieť založenú na kochleárnom mechanizme ľudského ucha [29]. V každom prípade, pre neuronové siete musí byť vstupný priestor a testovacie dáta zvolené veľmi starostlivo. Ďalším problémom je, že aj keď sa nájde vhodný model, je len ťažké objasniť jeho fungovanie. Je treba sa uspokojiť s "čiernou skrinkou".

Doval a Rodet používajú na detekciu f_0 metódu maximálnej vierohodnosti. Táto štatistická metóda porovnáva rozne predpovede hodnôt premenných podľa pravdepodobnosti, že sú správne v kontexte s predchádzajúcimi hodnotami týchto premenných. Model, ktorý prezentujú, vyzerá nasledovne: pozorovanie O pozostáva zo série zložiek v krátkodobej fourierovej transformácii signálu. Predpokladá sa, že každé pozorovanie bolo vyprodukované zvukom o určitej fundamentálnej frekvencii, a každé spektrum obsahuje ešte ďalšie informácie vrátane neharmonických zložiek (šum). Pre každú zo sady kandidátnych fundamentálnych frekvencií algoritmus vypočíta pravdepodobnosť, že dané pozorovanie bolo vyprodukované touto frekvenciou, a nájde maximum. Voľba sady fundamentálnych frekvencií je veľmi dôležitá.

2.6 Poloha prstov

V momente, keď už je známa postupnosť jednotlivých tónov, je možné bez ďalších problémov vyprodukovať štandardný notový zápis. Každý tón totiž odpovedá právej jednej pozícii na notovej osnove. V prípade gitarovej tablatúry sa však na takúto jednoznačnosť nemožno spoliehať a okrem výšky tónu je nutné zobrať do úvahy ešte niekoľko faktorov.

2.6.1 Závislosť na nástroji

V prvom rade, na rozdiel od štandardnej notácie, gitarová tablatúra je závislá na použitom nástroji. Zatiaľ čo napríklad klavír má drvivej väčšine prípadov práve 88 klávesov a štandardizované koncertné ladenie, moderné gitary sa môžu podstatne líšiť medzi jednotlivými modelmi (viď Obr. 2.8).



Obr. 2.8: Rôzne typy gitár. Počty strún a pražcov sa zásadne líšia. (prevzaté z [22] [23] [4])

Môžu byť ladené do niekoľkých rôznych ladení, počet pražcov môže byť rozdielny, a v poslednej dobe sa stávajú veľmi populárnymi gitary s rozšíreným rozsahom (sedem, osem, a viac strún), alebo v extrémnych prípadoch zúženým rozsahom (tri alebo dokonca len jedna struna, takzvaná "djent stick").

Všetky tieto charakteristiky majú vplyv na výslednu tablatúru, ale zo samotného audio signálu ich nemožno s dostatočnou istotou rozpoznať. Najjednoduchším riešením by bolo ponechanie voľby na užívateľovi. Vstupom programu by teda okrem samotnej audio nahrávky boli aj informácie jasne definujúce typ gitary a jej ladenie. V prípade že danú melódiu na takomto nástroji nemožno zahrať, vstup by bol považovaný za neplatný.

Alternatívne sa program môže pokúsiť charakteristiky nástroja odhadnúť. V takom prípade možno na začiatku predpokladať 6 strún, 24 pražcov a ladenie E Standard (E A D G B E), ktoré je používané najčastejšie. Ak sa v melódii objaví tón nižší než je v tomto rozsahu možný, ladenie sa zníži, ak sa objaví naopak príliš vysoký tón, ladenie sa zvýši. Ak sa v nahrávke vyskytujú tóny príliš vysoké aj príliš nízke, možno rozšíriť rozsah pridaním ďalšej struny. Takýmto spôsobom možno detekovať podladené a nadladené gitary

a gitary s rozšíreným rozsahom. Alternatívne ladenia (otvorené, modálne, pravidelné atď.) však detekované nebudú.

Pri detekovaní charakteristík nástroja pravdepodobne môže mať vplyv aj miera vhodnosti prstokladu popísaná v nasledujúcej pokapitole. Týmto spôsobom by sa zrejme najjednoduchšie detekovali modálne ladenia.

2.6.2 Závislosť na hudobníkovi

Z faktu, že na gitare možno zahrať rovnaký tón na viacerých miestach vyplýva, že rovnakú melódiu možno zahrať viacerými spôsobmi, a to aj na tej istej gitare v tom istom ladení (viď Obr. 2.1). Väčšinu tónov nad E4 v ladení E Standard možno zahrať na dvoch až troch miestach na hmatníku. Ak teda uvažujeme 2, 5 pozície na jeden tón, pre melódiu obsahujúcu n za sebou nasledujúcich tónov existuje približne P možných prstokladov (viď eq 2.6).

$$P = 2 \cdot 5^n \quad (2.6)$$

Voľba medzi týmito prstokladmi je z veľkej časti na každom individuálnom hudobníkovi, niektoré prstoklady však môžu byť vhodnejšie ako iné, niektoré môžu byť dokonca nemožné. Zrejme teda neexistuje lepšia možnosť, než v mnohých prípadoch poskytnúť užívateľovi viac verzií tablatúry a nechať voľbu na ňom, môžeme však významne zredukovať ich počet odstránením menej vhodných prstokladov.

Je teda nutné vytvoriť funkciu schopnú každému prstokladu priradiť číselnú hodnotu (mieru vhodnosti) a na výstup poslať len tie, ktoré presahujú istý prah vhodnosti. Ak sa nenájde žiadny dostatočne vhodný prstoklad, možno zväziť zmenu ladenia (viď 2.6.1).

Hlavným faktorom ovplyvňujúcim vhodnosť je určite pohyb prstov v rámci pozície ruky na hmatníku, keďže prstoklady minimalizujúce tento pohyb sú vo všeobecnosti jednoduchšie. Funkcia vhodnosti teda musí pre daný prstoklad najprv odhadnúť pozíciu ruky a každého prstu a spočítať vzdialenosť, ktorú treba prekonať medzi jednotlivými tónmi. Môžeme teda rozdeliť celú melódiu do niekoľkých sekcií, ktoré obsahujú pozície na hmatníku dosiahnuteľné bez pohybu ruky. K pohybu ruky dochádza len medzi týmito sekciami. V každej sekcií môžeme predpokladať, že tón bližšie ku kobyľke je zahráný prstom ďalej od ukazováka. V momente, keď má každý jednotlivý tón priradený prst, môžeme spočítať celkovú vzdialenosť, akú musia prsty prekonať počas každej sekcie, a následne sčítať všetky sekcie dohromady. Výsledok určuje vhodnosť daného prstokladu, kde nižšie číslo znamená vyššiu vhodnosť.

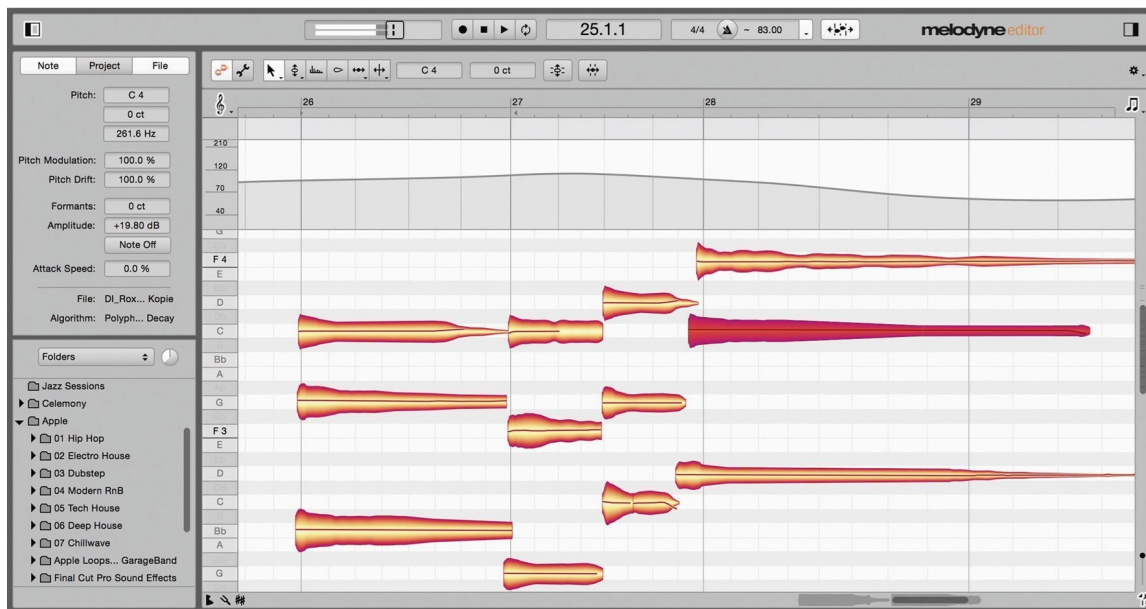
Treba však zobrať do úvahy aj niektoré zložitejšie techniky. Ak sa napríklad objavia dva alebo viac tónov pod sebou klasifikované ako hrané ukazovákom, ide takmer určite o techniku barré a pohyb ukazováku medzi týmito tónmi možno úplne zanedbať. Ak je detekovaný príliš rýchly pohyb ruky tam a späť medzi dvoma pozíciami, ide zrejme o tapping.

Zatiaľ neexistuje dostatočne presná metóda na prepis sledu nôt do tablatúry, bude ju treba pre účely tejto práce vytvoriť.

2.7 Kompletné riešenia

Zatiaľ neexistuje žiadny softvér schopný automaticky transformovať audio signál na gitarovú tablatúru. Hlavne problém prepisu sledu tónov na tablatúru je zatiaľ nepreskúmaný, pravdepodobne vďaka jeho subjektívnemu charakteru. Existuje však niekoľko komerčne dostupných programov, z ktorých niektoré sú schopné riešiť veľkú časť problémov popísaných v tejto kapitole.

Melodyne [32] je veľmi robustný program schopný detekovať rytmus, jednotlivé noty a izolovať jednotlivé nástroje s vysokou mierou presvedčivosti. Hneď po načítaní skladby je možné zobraziť noty od jednotlivých nástrojov, upravovať ich a uložiť vo formáte MIDI. Jeho ovládanie je veľmi intuitívne a nevyžaduje veľa úsilia. Ako nástroj na (takmer) automatickú transkripciu hudby nemá konkurenciu, čo sa odráža aj na jeho cene.

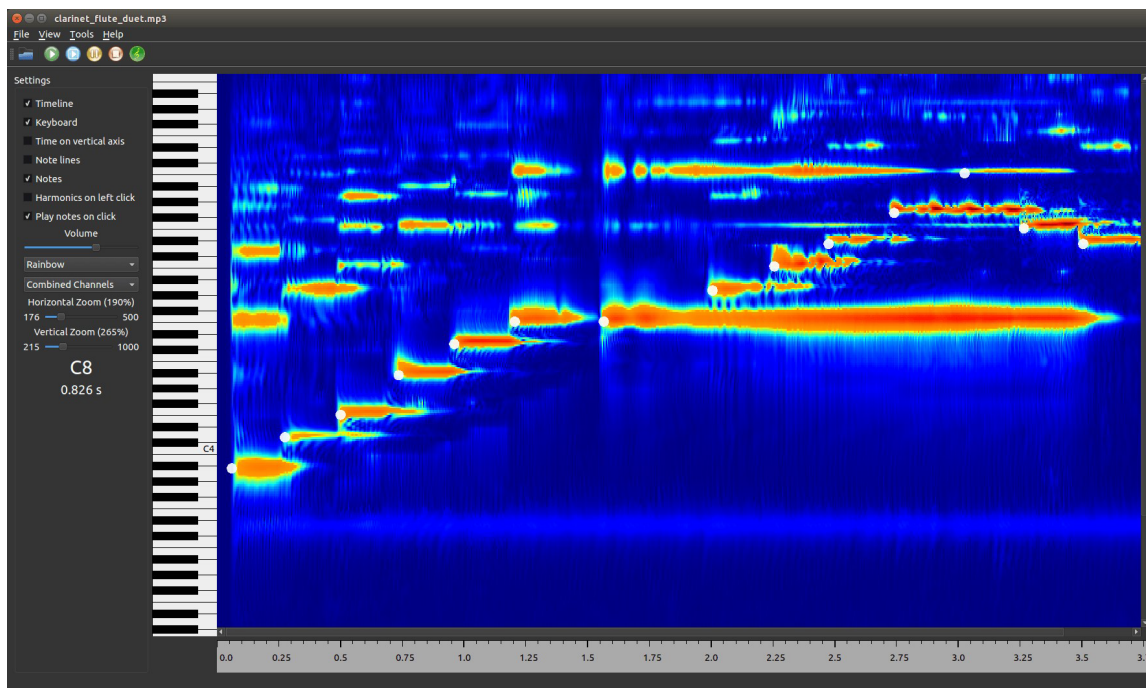


Obr. 2.9: Uživatelské rozhranie programu Melodyne[32]

Riffstation[33] je program určený na samoštúdium hry na gitare. Dokáže analyzovať akýkoľvek audio súbor a manipulovať s ním rôznymi spôsobmi, tak aby používateľovi uľahčil hranie tejto konkrétnej skladby. Medzi jeho silné stránky možno zaradiť veľmi robustnú detekciu rytmu. Taktiež dokáže veľmi presvedčivo izolovať melodickú gitarovú linku od ostatných zvukov, hlavne v kvalitných nahrávkach. Riffstation však nie je zameraný na automatickú transkripciu hudby a nedokáže detekovať melódiu. Obsahuje len veľmi približnú detekciu akordov.

ScoreCloud[5] je nástroj určený hudobným skladateľom, ktorý uľahčuje písanie nôt. Jedným zo spôsobov zápisu je práve analýza audio signálu. Detekcia výšky tónu je veľmi presná, aj keď je jasné, že je optimalizovaná skôr pre klavír, než gitaru. Jeho nevýhodou je, že očakáva na vstupe len jediný nástroj, hoci aj polyfonický. Na extrakciu gitarového sóla s kompletnej skladby sa teda nehodí.

AnthemScore[15] je open-source softvér ktorý automaticky prepisuje audio súbory do štandardného notového zápisu. Používa podobnú konvulčnú neuronovú sieť, aká sa používa pri počítačovom videní, pričom ako vstupný obraz je použitý spektrogram daného audia. Sieť bola natrénovaná na 2,5 miliónoch tréningových vzoriek, každá obsahovala v priemere 3 noty. AnthemScore dokáže celkom dobre detekovať noty aj veľkého počtu súčasne hrajúcich nástrojov. Nedokáže však tieto nástroje od seba odlíšiť, kombinuje ich do jedinej polyfonickej linky. Tú treba následne roztriediť ručne. Taktiež je veľmi výpočetne náročný, na aplikácie v reálnom čase sa nehodí.



Obr. 2.10: Rozhranie programu AnthemScore s detekovanými notami (biele body) v spektrograme. (prevzaté z [15])

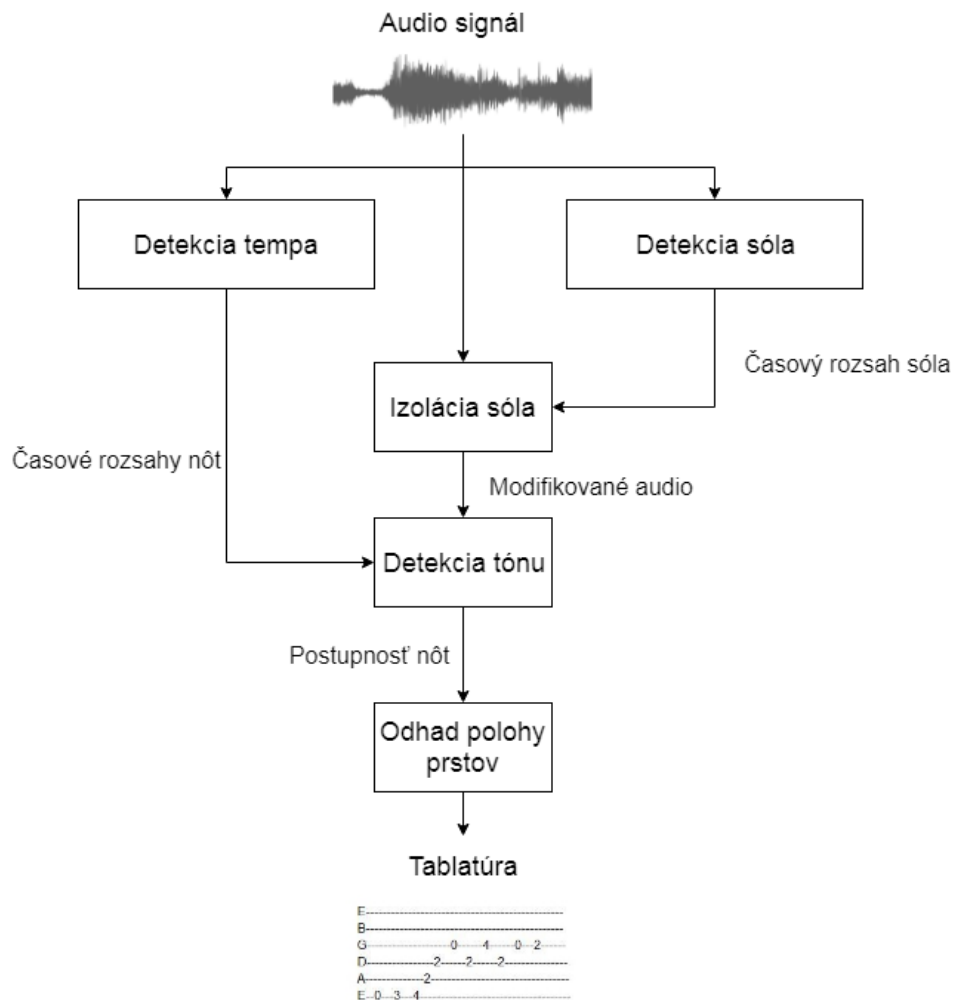
2.8 Anvil Studio, Finale PrintMusic

Obidva tieto programy sú určené pre skladateľov a sú zamerané hlavne na písanie nôt. Nedokážu extrahovať melódiu z audio signálu, dokážu však ako jedny z mála prepísať štandardný notový zápis na tablatúru. Kvalita tohto prepisu však nie je veľmi vysoká a vyžaduje vcelku rozsiahly zásah užívateľa.[36] [18]

Kapitola 3

Návrh riešenia

V tejto kapitole je popísaný návrh riešenia a súhrn použitých techník. Finálny systém pozostáva z niekoľkých samostatných súčastí, z ktorých každá rieši jeden z podproblémov uvedených v podkapitole 2.1. Celý systém je schematicky znázornený na obrázku 3.1.

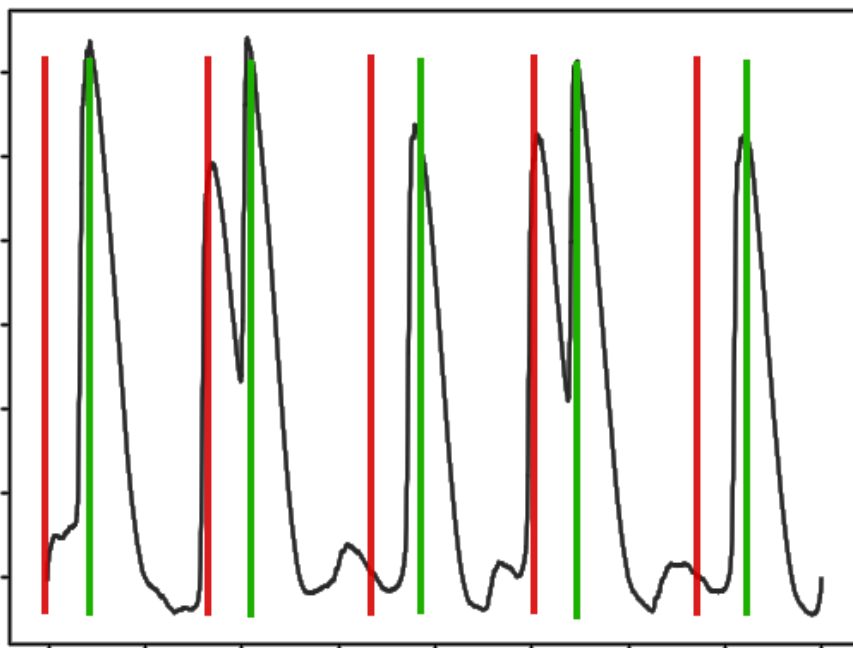


Obr. 3.1: Schéma systému

Detektor tempa vezme na vstupe audio nahrávku a odhadne jej tempo. Výstupom sú časové značky, určujúce rozsahy jednotlivých nôt. Výstupom detekcie sóla sú časové značky určujúce pasáže, v ktorých sa nachádza gitarové sólo. Následne je audio signál spracovaný v module "Ízolácia sóla", ktorý sa pokúsi zo sóla pasáže v nahrávke odstrániť doprovodné nástroje a iné nežiadúce zvuky. Ďalej je vykonaná detekcia tónu na každom z rozsahov určených detektorom tempa. Výsledkom je postupnosť nôt. Nakoniec systém odhadne najvhodnejšie polohy týchto nôt na hmatníku gitary. Výstupom je gitarová tablatúra.

3.1 Detekcia tempa

Na rozdelenie signálu do jednotlivých nôt som sa rozhodol použiť metódu detekcie tempa a následného rozdelenia signálu podľa tohto tempa. Detekcia nábehov nôt je totiž problematická, hlavne v polyfonických nahrávkach, napríklad s výraznými bicími, a je náchylná na mierne odchýlky v signále. Naopak, detekcii tempa výrazné bicie prospievajú a mierne odchýlky majú len minimálny vplyv. Dá sa tiež predpokadať, že gitarová hudba nasleduje viac-menej stabilný rytmus.



Obr. 3.2: Skutočné downbeaty (zelené) sú posunuté oproti zistenému tempu (červené).

Detektor tempa som umiestnil na úplný začiatok, pred akékoľvek iné spracovanie signálu, aby bola detekcia tempa čo najpresnejšia. Na samotnú detekciu tempa som použil algoritmus navrhnutý pre projekt Beat This[3], ktorý som spomenul v podkapitole 2.4.1. Algoritmus vezme krátky úsek nahrávky a rozdelí ho do 6 samostatných signálov, rozložených rovnomerne do osobitných frekvenčných pásem. Každý z týchto signálov je usmernený plnou vlnou a prefiltrovaný Hannovým oknom, čím sa získa amplitúdová obálka. Dĺžka okna je nastavená na 0.2 sekundy, čo je približná dĺžka dočasného zvýšenia amplitúdy vy-

volanej výrazným downbeatom¹. Ďalej sa vypočíta derivácia amplitúdovej obálky a usmerní sa polovičnou vlnou, zaujímajú nás totiž len prudké nárasty amplitúdy. Výsledný signál sa prefiltruje sériou hrebeňových filtrov, pre každé skúmané tempo. Každý filter obsahuje niekoľko impulzov vzdialených od seba tak, ako sú od seba vzdialené takty v danom tempe. Nakoniec sa zistí, ktorý filter po prefiltrovaní vyprodukoval signál s najväčšou energiou a dané tempo je priradené tomuto časovému úseku v skladbe.

Dôležitou časťou je ešte zistenie offsetu tempa, nemôžeme totiž predpokladať že prvý downbeat bude vždy okamžite na začiatku (viď obr. 3.2). K tomu využijeme znovu signál, ktorý vznikol bezprostredne po prefiltrovaní hrebeňovým filtrom vypočítaného tempa. Jeho najvyššia hodnota označuje offset, s ktorým je skutočný rytmus vo fáze s vypočítaným tempom.

Ak si skladba udržiava rovnaké tempo v celej svojej dĺžke, celú túto procedúru stačí vykonať len raz. Na to sa však nemožno spoliehať. Väčšina skladieb má lokálne tempo stabilné, v rámci celej skladby môže dôjsť k prudkým jednorazovým zmenám (rýchla skladba má pomalú predohru), alebo sa môže tempo meniť postupne (sliding tempo). Tempo teda musí byť detekované pre celú skladbu, po krátkych úsekoch. Tieto tempá sa následne uložia za sebou do takzvanej tempomapy, kde každý prvok určuje tempo jedného krátkeho úseku skladby. Tieto úseky musia byť dostatočne krátke na to, aby mala tempomapa dostatočné rozlíšenie pri zmenách, ale dostatočne dlhé, aby bol algoritmus schopný detekovať pomalé tempá. V tejto práci som zvolil dĺžku úseku 5 sekúnd, čo teoreticky stačí na detekovanie temp od 40bpm vyššie.

Tempomapa je ešte prefiltrovaná mediánovým filtrom, aby sa vyhladili prípadné špičky, ktoré sú takmer vždy produktom chyby pri detekcii tempa.

Nakoniec podľa tempomapy delením jednotlivých taktov vygenerujem indexy začiatkov jednotlivých nôt. Noty sú osminové, pri rýchlejších tempách len štvrtové, pretože nasledujúca detekcia tóny potrebuje noty určitej dĺžky. Tieto indexy sú výstupom tohto modulu.

3.2 Detekcia sóla

Je si treba uvedomiť, že moderná technológia ešte nie je v stave, keď by bolo možné s dostatočnou istotou detekovať gitarové sólo v audiu bez akýchkoľvek ďalších informácií. Doterajšie výskumy aj moje vlastné experimenty s rekurentnou neurónovou sieťou a SVM klasifikátorom sa svojou úspešnosťou pohybovali okolo 70%. To môže stačiť pre iné prípady použitia, ako je napríklad indexovanie, automatická anotácia a vyhľadávanie. V tejto práci však prípadné chyby v detekcii sóla ovplyvnia ďalšie moduly, ktoré následne vyprodukujú priam neakceptovateľné výsledky.

Z tohto dôvodu som sa rozhodol sústrediť na samotný prepis sóla do tablatúry, ktorý považujem v tejto práci za kľúčový a ktorého vnímaná kvalita by použitím nepresnej detekcie utrpela, a detekciu nechať na užívateľovi. K tomu prispel aj fakt, že takáto detekcia je pre užívateľa triviálnou záležitosťou. Je omnoho jednoduchšie zadať programu čas začiatku a konca sóla, než hľadať správne noty v chaose, ktorý vznikne keď sa systém rozhodne prepísať ako sólo pasáže, ktoré nim nie sú.

¹**Downbeat** - pomenovanie ktoré sa vzťahuje na označenie prvého zvýrazneného rávania v takte, ktoré dirigent alebo kapelník naznačuje vertikálnym pohybom dole (down)

3.3 Izolácia sóla

Na izoláciu sóla som sa rozhodol skombinovať metódu ekvalizácie s metódou Out of Phase Stereo (OOPS). Pokročilejšie metódy separácie nástrojov sa bohužiaľ spoliehajú na to, že sa v nahrávke nachádzajú krátke úseky, kde hrá daný nástroj samostatne. V prípade gitarových sól však takéto úseky spravidla neexistujú. Nie je teda možné odhadnúť spektrálnu šablónu ani amplitúdovú obálku.

V prípade ekvalizácie som sa spoliehal na fakt, že melodické gitarové sóla sa zvyčajne hrajú vo vyšších tónoch, než doprovod. Toho sú si vedomí aj štúdiovní audio inžinieri, a preto sa vo väčšine štandardných nahrávok gitarové sólo nachádza v mierne vyšších frekvenciách, než ostatné gitarové linky. Experimentálne som zistil, že odfiltrovaním frekvencií pod 500Hz hlasitosť sóla oproti doprovodu významne stúpne. Takisto je vhodné odfiltrovať frekvencie nad približne 5KHz, ktoré do zvuku sóla nijak významne neprispievajú, zato sú však plné sýčavého zvuku produkovaného hlavne činelmi a inými nástrojmi.

Metóda OOPS využíva rozloženie nástrojov v stereo poli. Gitarový doprovod, a väčšina bicích sa zvyčajne mixuje viac na strany. Gitarové sóla sa zvyknú mixovať do stredu, prispieva tak ako do pravého, tak aj do ľavého kanálu rovnakou amplitúdou. Odčítaním pravého kanála od ľavého (alebo naopak) môžeme frekvencie v strede stereo poľa takmer úplne potlačiť. Naopak, sčítaním oboch kanálov sa amplitúda "stredného kanálu" zosilní o približne 3dB, zatiaľ čo na stranách ostane približne rovnaká. Okrem gitarových sól sa ešte do stredu zvykne mixovať basová gitara, spev, veľký a malý bubon. Veľká časť nízkych frekvencií basovej gitary a veľkého bubna bola však už odfiltrovaná. Čo sa týka spevu, ten je sám o sebe považovaný za sólový nástroj a takmer nikdy neznie v rovnakom čase ako gitarové sólo.

Tieto metódy samozrejme nedokážu izolovať sólo uveriteľne pre ľudského poslucháča. Po vykonaní ekvalizácie a sčítaní oboch stereo kanálov by sme však mali dostať mono signál, v ktorom je gitarové sólo výrazne dominantné svojou amplitúdou.

3.4 Detektor tónu

Detektor tónu dostane na vstupe časové úseky audio signálu predstavujúce jednotlivé noty. Jeho úlohou je pre každú notu určiť správny tón. K tomu som sa rozhodol použiť metódu optimálneho hrebeňového estimátoru[21]. Táto metóda ponúka pri aplikovaní na zvuk gitary veľmi presné výsledky, používa sa dokonca v niektorých digitálnych gitarových ladičkách.

Najprv sa vstupný signál vynásobí Hammingovým oknom. Tým sa zvýraznia jednotlivé harmonické zložky v spektre. Následne sa toto spektrum vypočíta pomocou rýchlej fourierovej transformácie. Frekvencie pod 70Hz sa odfiltrujú, aby do výpočtu nevnesli nepresnosti, keďže gitary tieto frekvencie neprodukuje. Ďalej sa spektrum normalizuje tak, aby mala najväčšia špička hodnotu 1. To pomôže zotrieť rozdiely medzi hlasnými a tichými notami, a tiež pri počítaní energie signálu nižšie.

Nasleduje hlavná časť algoritmu, a to násobenie spektra hrebeňovými filtermi. Podobné filtre boli použité pri detekcii tempa v podkapitole 3.1, tam však išlo o násobenie, ale konvolúciu. Hrebeňové filtre sa generujú len pre frekvencie odpovedajúce notám nachádzajúcim sa na hmatníku gitary. Tým sa celý výpočet podstatne zrýchli. Filter, ktorý vyprodukuje spektrum s najvyššou energiou odpovedá detekovanej note, avšak len vtedy, ak je táto energia vyššia než 0.5. Nižšie energie zvyčajne znamenajú nehudobný signál, v takom prípade predpokladáme, že zvuk gitary nie je prítomný, výsledkom je prázdna nota.

Veľkým problémom tejto metódy je takzvaná chyba oktávy (octave error). Tá je zapríčinená tým, že každý tón obsahuje tie isté harmonické zložky ako tón o oktávu vyššie (dvojnásobná frekvencia). Optimálny hrebeňový estimátor teda vždy vypočíta pre spektrum nižšieho tónu energiu vyššiu (aj keď len nepatrne) než tónu o oktávu vyššie (viď obr. 3.3). Výsledkom je vždy tón na najnižšej oktáve. Toto by sa dalo riešiť lokalizovaním prvej špičky v spektre, ktorá predstavuje fundamentálnu frekvenciu, a v monofonických signáloch je to najväčšia špička v spektre. V polyfonických signáloch však táto špička nemusí byť najväčšia, môže dokonca úplne chýbať (pravdepodobne sme ju odstránili pri filtrácii v snahe izolovať sólo, viď 3.3). Lepším riešením je porovnanie energií medzi oktávami. Keď už bol detekovaný tón, predpokladáme že je to tón na najnižšej oktáve. Energiu spektra vynásobeného hrebeňovým filtrom vygenerovaným pre tento tón porovnáme s energiou pre tón o oktávu vyššie. Ak zistíme rapídny pokles (približne dvojnásobný), znamená to, že vyšší filter nevzal do úvahy niektoré harmonické zložky, a teda správnym riešením je nižší tón. Naopak, ak zistíme len nepatrnú zmenu, posunieme sa o oktávu vyššie a proces opakujeme.



Obr. 3.3: Zelený filter predstavuje správne riešenie, červený však vygeneruje spektrum s nepatrne vyššou energiou.

V posledom kroku treba odlíšiť dlhé noty od krátkych. Vstupom detektora tónu sú približne rovnako krátke segmenty, z ktorých každý predstavuje krátku notu (osminovú alebo štvrtovú), vo vstupnom sóle sa však môžu nachádzať dlhšie noty, ktoré sa rozložia do viacerých segmentov. Preto sa ešte vykoná v postupnosti nôt agregácia. Ak za sebou nasleduje viac inštancií tej istej noty, považujú sa za jednu dlhú notu a duplikáty sa jednoducho vymažú. Tým sa dĺžka noty zanedbá, čo však nepredstavuje problém, pretože informácie o dĺžkach nôt sa v tablatúre nenachádzajú.

Problémom však je, ak sa vo vstupnom sóle naozaj nachádza dva a viackrát tá istá nota za sebou. V ideálnom prípade sa medzi nimi detekuje prázdna nota, na to sa však nemožno spoliehať. Je však známe že pri hre na gitare vznikajú vo vyšších frekvenciách

prechodové deje tesne po údere na strunu. Keď dôjde k zvýšeniu energie v týchto frekvenciách v danom segmente oproti predchádzajúcemu, môžeme usúdiť, že sa jedná o novú notu, nie doznievajúcu. V takom prípade vložíme medzi tieto noty ešte jednu prázdnu.

Týmto spracovanie audia končí. Ďalej program pracuje už len s postupnosťou čísel, z ktorých každé predstavuje jednu notu.

3.5 Poloha prstov

Posledný modul preloží postupnosť nôt do postupnosti jednotlivých pozícií na hmatníku gitary. Ako už bolo spomenuté vyššie, väčšinu nôt možno zahrať na viacerých miestach na hmatníku (priemerne 2,9 pozícií na každú notu). S každou ďalšou notou sa počet možností ako zahrať dané sólo exponenciálne zvyšuje (viď rovnicu 3.1). Výber z týchto možností je z veľkej časti na samotnom muzikantovi. Objektívne najlepšie riešenie neexistuje, ale veľa muzikantov sa zhodne na niekoľkých jednoduchých pravidlách.

$$P = 2.9^n \tag{3.1}$$

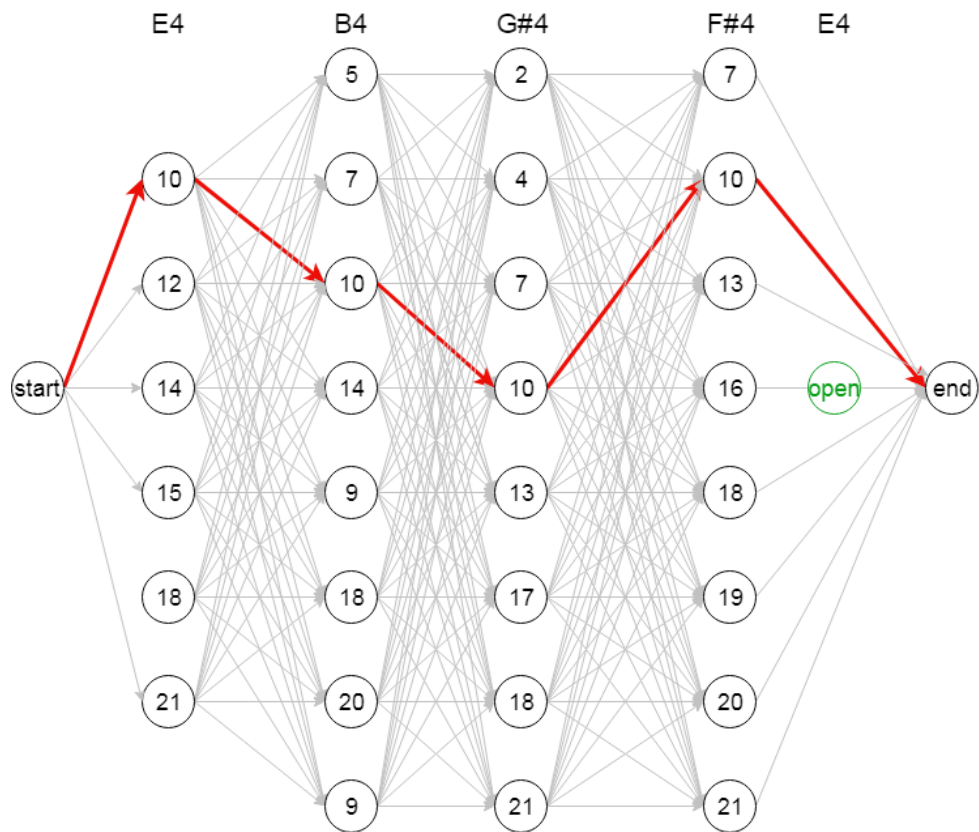
Môj algoritmus vyberá najvhodnejšie riešenie podľa dvoch takýchto pravidiel. Po prvé, najvhodnejšie riešenie by malo čo najviac minimalizovať pohyb celej ruky po hmatníku. Napríklad riešenie, keď sa všetky noty hrajú na jednej strune a pre každú z nich sa mení pozícia ruky je očividne nesprávne (resp. nevhodné). Po druhé, najvhodnejšie riešenie je také, kde sa sólo hrá na čo najnižších pražcoch, keďže hra na vysokých pražcoch je náročnejšia a menej pohodlná.

Algoritmus funguje nasledovne. Najprv sa vytvorí vhodná reprezentácia gitarového hmatníku. Zo vstupných nôt sa vyberie najnižšia a najvyššia. Štandardne sa predpokladá šesť strún, 24 pražcov a ladenie E standard, teda rozsah štyri oktávy od E2 po E6. Ak sa najnižšia nota nachádza pod týmto rozsahom, je treba ladenie patrične znížiť. Ďalej ak je najvyššia nota nad aktuálnym rozsahom, treba pridať ďalšiu strunu (ladenia vyššie než E standard neuvažujeme). Na tieto zmeny treba samozrejme upozorniť užívateľa.

Keď je reprezentácia hmatníka hotová, môžeme pre každú notu vo vstupe zistiť, v ktorých polohách ruky ju možno hrať. Polohou ruky rozumieme číslo pražca nad ktorým je ukazovák. Každú notu možno hrať na tomto pražci, alebo až o tri pražce vyššie ostatnými prstami (na vysokých pražcoch ešte viac, to však zanedbáme). Napríklad notu A2, ktorá sa nachádza na najhrubšej strune na pražci 5 možno zahrať s ukazovákcom nad pražcami 2, 3, 4, alebo 5. Zaujímavou výnimkou sú noty na otvorených strunách, tie možno hrať kedykoľvek, nezávisle od polohy ruky.

Keď už poznáme pre každú notu všetky polohy, v ktorých ju možno hrať, môžeme z nich zostrojiť orientovaný graf (viď obr. 3.4). Každá cesta zo štartovacieho uzlu do koncového predstavuje jednu možnosť, ako hrať danú postupnosť nôt. Váha každej hrany predstavuje pohyb rukou medzi polohami v počte pražcov. Noty na otvorených strunách hrany nemajú, keďže pohyb rukou nie je nutný, v takom prípade spojíme noty pred a po takejto note. Takisto hrany vychádzajúce zo štartovacieho uzla a vchádzajúce do koncového uzla majú váhu nula.

V tomto grafe najkratšia cesta predstavuje spôsob hrania s minimálnym pohybom ruky. Nájdeme ju pomocou Dijkstrovho algoritmu. Ten je mierne modifikovaný tak, aby pri výbere minimálneho uzla bral do úvahy okrem cesty aj to, ako nízko sa táto poloha na hmatníku nachádza. To zaručí, že ak existuje viac minimálnych ciest, vyberie sa práve tá na najnižších pražcoch.



Obr. 3.4: Každý stĺpec obsahuje všetky polohy ruky, v ktorých možno zahrať danú notu, prepojenia medzi nimi predstavujú pohyb ruky. Notu E4 možno hrať na otvorenej strune, pohyb ruky nie je nutný. Minimálna cesta označuje spôsob hrania, ktorý celkový pohyb ruky minimalizuje.

V tejto chvíli už môžeme spárovať jednotlivé noty z polohami ruky na hmatníku. Ostáva už len z reprezentácie hmatníku zistiť, kde presne sa táto nota nachádza a môžeme vygenerovať finálnu tablatúru.

Kapitola 4

Implementácia

Táto kapitola sa venuje samotnej implementácii. Popisuje štruktúru aplikácie, použité nástroje a technológie, fungovanie jednotlivých častí a uvádza niektoré zaujímavé implementačné detaily.

4.1 Implementačné nástroje a technológie

Aplikáciu som začal programovať v jazyku C++, neskôr však bola preportovaná do jazyka Python, a to kvôli väčšej dostupnosti rôznych knižníc a jednoduchosti ich použitia. Použil som Python vo verzii 3.6.2 a nasledujúce knižnice:

numpy - obsahuje dátové štruktúry na rýchlu prácu s veľkými objemami dát, ako sú napríklad audio signály. Taktiež implementuje potrebné matematické funkcie nad týmito štruktúrami a hlavne rýchlu fourierovu transformáciu.

scipy - knižnica pre vedecké výpočty nad štruktúrami z numpy. V mojej práci bola použitá hlavne na načítanie a ukladanie audio súborov.

time - knižnica obsahujúca funkcie na meranie času. Použitá na meranie časovej náročnosti jednotlivých častí kódu vo fáze testovania.

tqdm - knižnica na zobrazovanie ukazateľa priebehu (progress bar) pri pomalších výpočtoch.

matplotlib - obsahuje funkcie na vykresľovanie a zobrazovanie grafových reprezentácií dát. Použitá pri testovaní.

argparse - knižnica na spracovanie vstupných argumentov programu.

Na analýzu dát pri návrhu riešenia som použil GNU Octave, open-source klonu Matlabu, a takisto audio editor Reaper. Pri testovaní poslúžil tabletový program Guitar pro 6.

Z hardvéru som na vytváranie vlastných nahrávok použil gitaru Jackson JS32 King V, mierne upravenú, zesilovač Jyo Zombie, reprobox Harley Benton G112 Vintage a mikrofón Electro-Voice Cobalt Co4. Ako AD/DA prevodník som použil audio rozhranie Lexicon Alpha.

4.2 Štruktúra aplikácie

System bol naprogramovaný ako konzolová aplikácia, ktorej vstupom je audio súbor vo formáte Wave (mono alebo stereo, ľubovoľná vzorkovacia frekvencia) a výstupom je text reprezentujúci tablatúru. Text je možné vypísať na štandardný výstup alebo do zadaného súboru.

Aplikácia sa skladá z piatich modulov:

audio_to_tabs.py - hlavný modul ktorý sa stará o načítanie vstupu a vypisovanie výstupu. Volá funkcie ostatných modulov.

isolation.py - vykonáva izoláciu

beat_detector.py - detekuje tempo a rozdeľuje signál na noty.

pitch_detector.py - detekuje, či sa jedná o skutočný tón alebo ide o prázdnu notu a zisťuje výšku tónu.

notes_to_tab.py - prepisuje postupnosť tónov na tablatúru

V nasledujúcom texte sú tieto moduly bližšie popísané.

4.2.1 audio_to_tabs.py

Tento modul v prvom rade spracuje vstupné argumenty štandardným spôsobom (pre zoznam argumentov viď prílohu) pomocou knižnice argparse. Ďalej načíta vstupný súbor a zavolá funkcie z modulov isolate.py a beat_detector.py. Potom vyberie časť nahrávky, ktorú užívateľ označil ako sólo a zvyšok vynuluje. Ako vidieť, detekcia tempa sa vykonáva nad celou nahrávkou, nie len nad sólom, kvôli presnosti. Ďalej sa v cykle predávajú jednotlivé segmenty nahrávky modulu pitch_detector.py a zistené noty sa ukladajú do poľa, kde sa nakoniec zoskupia za sebou nasledujúce rovnaké noty. Nad touto postupnosťou nôt sa nakoniec zavolá modul notes_to_tab.py a jeho výstup sa uloží do výstupného súboru, alebo sa vypíše na štandardný výstup.

4.2.2 isolation.py

Modul pôvodne vykonával izoláciu sóla pomocou mixovania stereo kanálov a odfiltrovanie vysokých a nízkych frekvencií. Táto filtrácia bola v záujme optimalizácie presunutá do modulu pitch_detector.py. Tento modul teda v konečnom dôsledku iba zistí, či sa jedná o stereo signál, a ak áno, oba kanály zmixuje do jedného. Vďaka tomu je možné tento modul umiestniť ešte pred detektor tempa, ktorý tiež pracuje s jednokanálovým signálom

4.2.3 beat_detector.py

Vstupnou funkciou tohto modulu je funkcia generate_tempo_map. Táto funkcia najprv zavolá funkciu interpolate, kde sa vstupný signál podvzorkuje do frekvencie 8000Hz. Detekcia tempa je totiž pomerne náročná na čas a nižšia vzorkovacia frekvencia pomáha tento čas skrátiť. Na presnosť detekcie to nemá veľký vplyv. Pri podvzorkovaní je tiež dôležité prefiltrovať signál dolnou priepustou, aby nedošlo k aliasingu. Na to slúžia funkcie butter_lowpass a butter_lowpass_filter. Nové vzorky sú vypočítané zo starých pomocou

lineárnej interpolácie. Tento podvzorkovaný signál sa používa len v tomto module, ostatné moduly pracujú s pôvodnou verziou signálu.

Ďalej sa na každý päťsekundový úsek signálu zavolá funkcia `get_local_beat`. Tá vykonáva samotnú detekciu rytmu podľa algoritmu popísaného v 3.1. Signál sa delí do šiestich frekvenčných pásem medzi frekvenciami 0, 200, 400, 800, 1600, 3200 a 8000 hertzov. Šírka hannovho okna je 0,2 sekundy. Hrebeňové filtre sa generujú s rozstupmi 1bpm. Pri úsekoch o dĺžke 5 sekúnd by algoritmus mal byť teoreticky schopný detekovať tempá už od 40bpm vyššie, to však len s nulovým offsetom. Preto som ako najnižšie možné tempo zvolil až 60bpm. Najvyššie tempo je 240bpm. Pod tento limit sa zmestia aj tie najrýchlejšie gitarové sóla.

Po získaní tempa pre každý päťsekundový interval sa na tieto tempá aplikuje mediánový filter so šírkou jadra 3, len aby sa vyhladili prípadné chyby. Nakoniec sa tieto tempá použijú na vygenerovanie indexov do pôvodného signálu. Tieto indexy označujú začiatky úsekov, v ktorých bude prebiehať detekcia tónu. Tie v tejto chvíli ukazujú do podvzorkovanej verzie signálu, treba ich teda ešte upraviť tak, aby ukazovali do pôvodnej verzie.

4.2.4 `pitch_detector.py`

Hlavnou funkciou tohto modulu je `get_local_pitch`. Vstup sa najprv vynásobí hanningovým oknom a vykoná sa rýchla fourierova transformácia. V spektre sa vynulujú frekvencie pod 500hz a nad 5000hz a následne sa spektrum normalizuje. Ďalej sa vykonáva detekcia tónu podľa 3.4. Testujú sa tóny od C2 po A6. Tóny mimo tento rozsah sa v gitarových sólach nepoužívajú takmer nikdy.

Po zistení najpravdepodobnejšieho tónu sa ešte testuje oktáva, taktiež podľa 3.4. V štandardnom ladení sa na hmatníku gitary môže jeden tón vyskytovať v 5 oktávách. V mojom programe sa ich testuje 6 pre prípad že ide o gitaru z rozšíreným rozsahom.

Nakoniec sa ešte overí, či je zistený tón naozaj tónom, alebo išlo o šum pozadia. Skutočný tón má v zvyčajne v normalizovanom spektre energiu väčšiu než 0.5. Ak toto platí, vráti funkcia tento tón, v opačnom prípade vráti `None`.

4.2.5 `notes_to_tab.py`

Vstupnou funkciou tohto modulu je `generate_tab`. Vo vstupnej postupnosti nôt sa najprv nájde najnižšia a najvyššia nota. Podľa týchto informácií sa následne vygeneruje reprezentácia hmatníka gitary v podobe dvojrozmerného poľa, kde každý prvok určuje notu na danej pozícii (viď obr 4.1). Najnižšou notou na hmatníku je E2 v prípade, že sa vo vstupe nenašla nižšia. Ďalšie noty sú rozložené podľa štandardného ladenia.

$$\begin{bmatrix} 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 \\ 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 \\ 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

Obr. 4.1: Dvojrozmerné pole reprezentujúce noty na hmatníku. 0 je nota E2

Ďalej sa pre každú notu vo vstupe nájdú všetky polohy ruky, v ktorých ju možno hrať. Týmito polohami sú horizontálne súradnice všetkých výskytov tejto noty na hmatníku, plus tieto súradnice znížené o 1, 2 a 3.

Nasleduje funkcia `generate_path`, v ktorej sa vygeneruje graf všetkých možných postupností polôh podľa vstupnej sekvencie nôt (viď obr 3.4). Všetky polohy pre notu sa spoja hranou s všetkými polohami pre nasledujúcu notu. Váha hrany je vzdialenosť medzi týmito polohami na hmatníku. Noty, ktoré možno hrať na otvorených strunách sa v tomto kroku ignorujú, nemali by totiž mať vplyv na najkratšiu cestu v grafe. Z postupnosti nôt sa vymažú a vložia sa naspäť neskôr. Na začiatok a koniec grafu sa ešte pridajú začiatočný a koncový uzol, a spoja sa s polohami prvej a poslednej noty hranou o váhe 0. To preto, aby bolo vždy jasné, ktorým uzlom sa má začať a ktorým skončiť. Nad takto pripraveným grafom sa vykoná dijkstrov algoritmus a výsledkom je najkratšia cesta.

V tejto chvíli už poznáme pre každú notu polohu ruky. Nájde sa teda výskyt danej noty na hmatníku, ktorý je najbližší polohe ruky pre túto notu určenej. Keď už poznáme presné súradnice každej noty, vygeneruje sa textová reprezentácia tablatúry.

Kapitola 5

Testovanie a vyhodnotenie

Táto kapitola sa venuje testovaniu a vyhodnoteniu finálneho systému. Obsahuje Popis dát použitých pri testovaní, dosiahnuté výsledky a ich zhodnotenie.

5.1 Data

Na testovanie aplikácie som zvolil nasledujúcich sedem skladieb. Každá z nich obsahuje monofonické gitarové sólo a každé z nich je niečím špecifické (v zátvorke je udaný čas sóla v piesni):

Whitesnake - Is this love (2:45 - 3:28) - Sólo na silno skreslenej elektrickej gitare, doprevádzané basovou gitarou, bicími a sotva počuteľnou neskreslenou elektrickou gitarou. Patrný je tiež výrazný efekt ozveny (reverb). Veľmi pomalé s častým vibrátom.

Beatles - Something (1:43 - 2:14) - Sólo na takmer nepostrehnuteľne skreslenej elektrickej gitare, doprevádzané bicími, basovou gitarou a Hammondovým organom. Miestami trochu nevýrazné v zmesi ostatných nástrojov. Relatívne pomalé.

Warrant - Heaven (2:15 - 2:38) - Sólo na silno skreslenej elektrickej gitare, doprevádzané bicími, basovou gitarou a skreslenou elektrickou gitarou. Pomalé až stredne rýchle.

Gary Moore - Parisienne walkways (0:00 - 0:33) - Sólo na relatívne skreslenej gitare, doprevádzané bicími, basovou gitarou a syntetizátorom. Veľmi výrazné v porovnaní s ostatnými nástrojmi, veľmi pomalé. Miestami sú počuteľné zvuky publika.

Poison - Every rose has its thorn (2:35 - 3:05) - Sólo na skreslenej gitare, doprevádzané akustickou gitarou, bicími a basovou gitarou. Nevýrazné, zdá sa byť akoby v pozadí, hlavne vďaka silnej ozvene. Relatívne pomalé.

Liquid Boogie Roll - Mechanical tornado (1:12 - 1:28) - Sólo na akustickej gitare doprevádzané ďalšou akustickou gitarou a takmer nepočuteľným klavírom. Stredne rýchle s veľmi rýchlym zakončením. Hrané v nižších tónoch.

Heart - What about love (2:30 - 2:53) - Sólo na relatívne skreslenej gitare doprevádzané basovou gitarou, bicími a syntetizátorom. Pomalé až stredne rýchle.

5.2 Vyhodnotenie

Testované boli ako jednotlivé súčasti, tak aj program ako celok. Všetky súčasti boli testované ručne. Metodika testovania a výsledky pre každý implementovaný modul sú popísané v nasledujúcom texte.

5.3 Izolácia sóla

Vzhľadom na charakter tejto súčasti bolo testovanie skôr kvalitatívne a výrazne subjektívne. Každému sólu bola priradená známka, ktorá určuje ako veľmi sa počuteľnosť sóla v zmesi nástrojov zlepšila oproti originálu, pričom 5 = "výrazné zlepšenie", 2 = "žiadne postrehnutelné zlepšenie" a 1 = "zhoršenie". Izolácia bola vykonávaná vrátane odfiltrovanie nízkych a vysokých frekvencií, hoci táto funkcionalita bola implementovaná v inom module. Výsledky sú v tabuľke 5.1.

Skladba	Známka	Poznámky
Is this love	4	-
Heaven	4	-
Something	3	-
Every rose has its thorn	3	-
Mechanical tornado	1	Nízke tóny boli odfiltrované
What about love	5	-
Parisienne walkways	5	-

Tabuľka 5.1: Výsledky testovania izolácie sóla

Tento modul sa výrazne spolieha na kvalitu nahrávky. Hlavne predpokladá sólo v strede stereo poľa a v určitom frekvenčnom rozsahu. Ak sa sólo nachádza mimo tento rozsah, jeho počuteľnosť je výrazne znížená. Taktiež sa zdá, že ak je sólo v pôvodnej nahrávke veľmi výrazné, použitím týchto metód sa zvýrazní ešte viac.

Pokročilejšie metódy izolácie nástrojov by v tejto časti určite pomohli, napríklad automatická detekcia frekvenčného rozsahu sóla. V súčasnej dobe však dostatočne presné metódy nie sú k dispozícii.

5.4 Detekcia tempa

Pri detekcii tempa bola použitá pôvodná nahrávka, ktorá neprešla modulom na izoláciu. Pri detekcii sa sledovala schopnosť rozdeliť sólo na intervaly reprezentujúce jednotlivé noty. Ako chyby boli označené prípady, keď došlo k zmene tónu vo vnútri tohto intervalu (ak jeden tón pretrváva počas viacerých za sebou nasledujúcich intervalov, nie je to chyba a zvyšok systému s tým počíta). Keďže algoritmus počíta s minimálne osminovými notami, je už na začiatku jasné, že v rýchlych pasážach nebude úspešnosť vysoká. Výsledky sú v tabuľke 5.2.

Ako sa predpokladalo, hlavnou slabinou tohto modulu je rýchlosť sóla. Delenie jednotlivých taktov na kratšie, než osminové noty, by pomohlo, avšak mohlo by to negatívne ovplyvniť ďalšie moduly. Ďalej sa tu vyskytuje problém podobný "chybe oktávy", keď je detekované tempo násobkom, prípadne zlomkom skutočného tempa. V neposlednom rade,

Skladba	Celkový počet nôt	Chyby	Percentuálna úspešnosť
Is this love	92	18	80
Heaven	78	28	64
Something	53	7	86
Every rose has its thorn	57	13	77
Mechanical tornado	94	49	47
What about love	51	19	62
Parisienne walkways	39	3	92

Tabuľka 5.2: Výsledky testovania detekcie tempa

tento modul je extrémne pomalý, čo je následkom veľkého množstva dát, s ktorými pracuje, a celkovej časovej náročnosti použitého algoritmu.

5.4.1 Detekcia tónu

Pri testovaní detekcie tónu boli použité intervaly, ktoré boli správne detekované v predchádzajúcom module. Pre každý interval sa detekoval v ňom obsiahnutý dominantný tón. Každá nesprávna detekcia bola označená ako chyba. Výsledky sú v tabuľke 5.3.

Skladba	Celkový počet nôt	Chyby	Percentuálna úspešnosť
Is this love	74	15	80
Heaven	50	12	76
Something	46	6	87
Every rose has its thorn	44	9	80
Mechanical tornado	45	25	44
What about love	32	10	69
Parisienne walkways	36	2	94

Tabuľka 5.3: Výsledky testovania detekcie tónu

Tento modul funguje vcelku presne, má však problém s veľmi nízkymi tónmi. Takisto skreslený zvuk gitary vytvára množstvo harmonických frekvencií, ktoré detekciu zťažujú. Riešením by mohlo byť použitie viacerých metód na detekciu tónu a kombinovanie ich výsledkov.

5.4.2 Prevod do tablatúry

Ako už bolo spomenuté, prevod nôt do tablatúry je veľmi subjektívny problém. Pri testovaní boli vstupom korektné noty získané z portálu ultimate-guitar.com. Výstup bol ohodnotený tromi užívateľmi (vrátane mňa) a tiež porovnaný s oficiálnymi tablatúrami, taktiež z portálu ultimate-guitar.com. Dôraz bol kladený na jednoduchosť a prirodzenosť hrania sóla štýlom určeným aplikáciou. Každému sólu bola priradená známka od 5 = "veľmi prirodzene hrateľné" po 1 = "takmer nehrateľné". Výsledky sú v tabuľke 5.4.

Jedným z výrazných problémov tohto modulu bol fakt, že noty prítomné na otvorených strunách sú vždy interpretované ako hrané na otvorených strunách. Toto nie je podľa skúsených gitaristov vždy najideálnejšie. Taktiež by bolo lepšie, ak by tento modul bral do úvahy aj pohyb jednotlivých prstov, nie len celej ruky. Vylepšenie tohto algoritmu vyžaduje rozsiahlejšiu konzultáciu s väčším množstvom gitaristov.

Skladba	Známka
Is this love	4
Heaven	4
Something	5
Every rose has its thorn	4
Mechanical tornado	2
What about love	3
Parisienne walkways	5

Tabuľka 5.4: Výsledky testovania detekcie tónu

5.4.3 Kompletný systém

Pri testovaní kompletneho systému bol vstupom audio súbor obsahujúci danú skladbu. Výstupom bola známka subjektívne určujúca ako zahrateľnosť výstupnej tablatúry, tak aj jej zhodu z originálnym sólom. Tieto známky boli poskytnuté tromi jednotlivcami. V tabuľke 5.5 známka 5 = "celkom korektná tablatúra" a 1 = "aboslútne nesprávna tablatúra".

Skladba	Známka
Is this love	4
Heaven	4
Something	3
Every rose has its thorn	3
Mechanical tornado	2
What about love	2
Parisienne walkways	4

Tabuľka 5.5: Výsledky testovania celého systému

Je evidentné, že chyby v každom module sa propagujú do nasledujúcich modulov. Bolo by vhodné implementovať nejakú formu spätnej väzby medzi modulmi, ktorá by tieto chyby minimalizovala.

Kapitola 6

Záver

Cieľom tejto práce bolo navrhnúť a vytvoriť systém schopný extrahovať zo zvukovej nahrávky gitarové sólo a prepísať ho do tablatúry. Tento problém bol analyzovaný a rozložený na 5 podproblémov. Boli preskúmané možné metódy vhodné na riešenie jednotlivých podproblémov.

Navrhnutý systém pozostáva zo štyroch modulov. Detekcia gitarového sóla v skladbe bola ponechaná na užívateľovi. Detekcia tempa bola použitá na rozdelenie sóla do intervalov reprezentujúcich jednotlivé noty. Každému intervalu bol priradený tón pomocou hrebeňového spektrálneho filtra. Prevod nôt do tablatúry sa spolieha na myšlienku minimalizácie pohybu ruky.

Systém bol implementovaný v jazyku python a otestovaný na troch užívateľoch, najprv po jednotlivých moduloch a nakoniec ako celok. Výsledný systém dokáže správne interpretovať niektoré typy gitarových sól. Problémom sú hlavne veľmi rýchle sóla, sóla v nízkych tónoch a skreslený zvuk gitary. Napriek tomu by vytvorený systém mohol o niečo urýchliť vytváranie gitarových tablatúr, ktoré doteraz prebiehalo ručne.

Výsledky tejto práce by bolo možné ešte zlepšiť použitím pokročilejších technológií, akými sú napríklad neurónové siete, a rozsiahlejšou spoluprácou s viacerými gitaristami.

Literatúra

- [1] Anytune Inc.: Anytune ReFrame. [Online; navštívené 11.04.2019].
URL <https://anytune.us/blog/2016/10/04/reframe-audio-isolation/>
- [2] Bello, J. P.; Sandler, M.: Phase-based note onset detection for music signals. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*. ročník 5, April 2003, ISSN 1520-6149, s. V–441, doi:10.1109/ICASSP.2003.1200001.
- [3] Cheng, K.; Nazer, B.; Uppuluri, J.; aj.: Beat This: A Beat Synchronization Project. [Online; navštívené 12.03.2019].
URL https://www.clear.rice.edu/elec301/Projects01/beat_sync
- [4] Dines, J.: *Custom Djent Guitar (closer look)*. [Online; navštívené 07.01.2019].
URL https://www.youtube.com/watch?v=7Lw0Ap_Vezw
- [5] Doremir Music Research AB: *ScoreCloud*. [Online; navštívené 13.01.2019].
URL <https://scorecloud.com/>
- [6] Edward, J.: *The New Grove Dictionary of Jazz(3 Volume Set)*. Oxford University Press, feb 2003, ISBN 1561592846.
- [7] Foote, J.: Automatic Audio Segmentation using a Measure of Audio Novelty. In *IEEE International Conference on Multimedia and Expo*, 2000.
- [8] Foulon, R.; Roy, P.; Pachet, F.: Automatic Classification of Guitar Playing Modes. In *Sound, Music, and Motion*, editácia M. Aramaki; O. Derrien; R. Kronland-Martinet; S. Ystad, Cham: Springer International Publishing, 2014, ISBN 978-3-319-12976-1, s. 58–71.
- [9] Fuhrmann, F.; Herrera, P.; Serra, X.: Detecting Solo Phrases in Music using Spectral and Pitch-related Descriptors. *Journal of New Music Research*, ročník 38, 12/2009 2009: s. 343 – 356, doi:<https://doi.org/10.1080/09298210903406632>.
- [10] Kedem, B.: Spectral analysis and discrimination by zero-crossings. *Proceedings of the IEEE*, ročník 74, č. 11, Nov 1986: s. 1477–1493, ISSN 0018-9219, doi:10.1109/PROC.1986.13663.
- [11] Keller, D.: Mixing in stereo: Adding width and depth to your recordings. [Online; navštívené 11.04.2019].
URL <https://www.uaudio.com/blog/studio-basics-mixing-stereo/>

- [12] Klapuri, A.: Sound onset detection by applying psychoacoustic knowledge. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, ročník 6, March 1999, ISSN 1520-6149, s. 3089–3092 vol.6, doi:10.1109/ICASSP.1999.757494.
- [13] Lal, A.; Wang, W.: Monaural Audio Separation Using Spectral Template and Isolated Note Information. In *Independent Component Analysis for Audio and Biosignal Applications*, editácia G. R. Naik, kapitola 4, Rijeka: IntechOpen, 2012, doi:10.5772/51606.
- [14] Li, Y.; Woodruff, J.; Wang, D.: Monaural Musical Sound Separation Based on Pitch and Common Amplitude Modulation. *IEEE Transactions on Audio, Speech, and Language Processing*, ročník 17, č. 7, Sep. 2009: s. 1361–1371, ISSN 1558-7916, doi:10.1109/TASL.2009.2020886.
- [15] Lunaverus: *AnthemScore*. [Online; navštívené 13.01.2019].
URL <https://www.lunaverus.com/>
- [16] M. Noll, A.: Short-Time “Cepstrum” Pitch Detection. *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, ročník 36, 05 1964, doi:10.1121/1.2143271.
- [17] Maher, R. C.; Beauchamp, J. W.: Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Acoustical Society of America Journal*, ročník 95, Apríl 1994: s. 2254–2263, doi:10.1121/1.408685.
- [18] MakeMusic: *Finale PrintMusic*. [Online; navštívené 13.01.2019].
URL <https://www.finalemusic.com/products/printmusic/>
- [19] Mauch, M.; Fujihara, H.; Yoshii, K.; aj.: Timbre and Melody Features for the Recognition of Vocal Activity and Instrumental Solos in Polyphonic Music. 01 2011, s. 233–238.
- [20] Moore, B. C. J.; Glasberg, B. R.; Baer, T.: A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *J. Audio Eng. Soc*, ročník 45, č. 4, 1997: s. 224–240.
- [21] Moorer, J. A.: On the Transcription of Musical Sound by Computer. *Computer Music Journal*, ročník 1, č. 4, 1977: s. 32–38, ISSN 01489267, 15315169.
- [22] Musikhaus Thomann: *Höfner HGL8 Green Line*. [Online; navštívené 07.01.2019].
URL https://www.thomann.de/cz/hoefner_hgl8_green_line.htm
- [23] Musikhaus Thomann: *Ibanez RG8-BK*. [Online; navštívené 07.01.2019].
URL https://www.thomann.de/cz/ibanez_rg8_bk.htm
- [24] Pati, K. A.; Lerch, A.: A Dataset and Method for Guitar Solo Detection in Rock Music. In *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*, Audio Engineering Society, 2017.
- [25] Patin, F.: Beat Detection Algorithms . [Online; navštívené 15.04.2019].
URL
<http://archive.gamedev.net/archive/reference/articles/article1952.html>

- [26] Peterschmitt, G.; Gomez, E.; Herrera, P.: Pitch-based solo location. In *Proceedings of MOSART Workshop on Current Research Directions in Computer Music*, Music Technology Group, 2001.
- [27] Piszczalski, M.; Galler, B. A.: Predicting musical pitch from component frequency ratios. *The Journal of the Acoustical Society of America*, ročník 66, č. 3, 1979: s. 710–720, doi:10.1121/1.383221.
- [28] Raitio, T.: Hidden Markov Model Based Finnish Text-to-Speech System Utilizing Glottal Inverse Filtering. 05 2019.
- [29] Sano, H.; Jenkins, B. K.: A Neural Network Model for Pitch Perception. *Computer Music Journal*, ročník 13, č. 3, 1989: s. 41–48, ISSN 01489267, 15315169.
- [30] Settel, J.; Lippe, C.: Real-time musical applications using the FFT-based resynthesis. 01 1994.
- [31] Smit, C.; P.W. Ellis, D.: Solo Voice Detection Via Optimal Cancellation. 11 2007, ISBN 978-1-4244-1620-2, s. 207 – 210, doi:10.1109/ASPAA.2007.4393045.
- [32] Software, C.: Melodyne. [Online; navštívené 08.04.2019].
URL <https://www.celemony.com/en/melodyne/what-is-melodyne>
- [33] Sonic Ladder Ltd.: *Riffstation*. [Online; navštívené 13.01.2019].
URL <https://pro.riffstation.com/>
- [34] Tan, L.; Karnjanadecha, M.: M.: Pitch Detection Algorithm: Autocorrelation Method and AMDF. In *In: Proceedings of the 3rd International Symposium on Communications and Information Technology*, 2003, s. 551–556.
- [35] White, P.: *What exactly is comb filtering?* [Online; navštívené 10.01.2019].
URL <https://www.soundonsound.com/sound-advice/q-what-exactly-comb-filtering>
- [36] Willow Software: *Anvil Studio*. [Online; navštívené 13.01.2019].
URL <https://www.anvilstudio.com/>

Príloha A

Ovládanie programu

Na spustenie programu je potrebný python vo verzii minimálne 3.6 s knižnicami popísanými v kapitole 4. Program je možné spustiť v nasledujúcom formáte:

```
audio_to_tabs.py [-s <start>] [-e <end>] [-o <output>] <input>
```

Význam jednotlivých parametrov:

<start> - Začiatok gitarového sóla v sekundách

<end> - Koniec gitarového sóla v sekundách

<output> - Výstupný súbor

<input> - Vstupný súbor

Príloha B

Obsah priloženého pamäťového média

Priložené CD obsahuje technickú správu vo formáte PDF a zdrojové súbory vo formáte LATEX, z ktorých bola správa vytvorená. Nachádzajú sa v priečinku DOC.

V priečinku SRC sa nachádzajú zdrojové súbory samotnej aplikácie. Súbor README.txt obsahuje návod na spustenie.