Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Anže Mikec

Odkrivanje melodije v zvočnih zapisih slovenskih ljudskih pesmi

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Matija Marolt

Ljubljana 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00073/2013
Datum: 04.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE MIKEC**

Naslov: **ODKRIVANJE MELODIJE V ZVOČNIH ZAPISIH SLOVENSKIH LJUDSKIH PESMI**

MELODY EXTRACTION FROM RECORDINGS OF SLOVENIAN FOLK SONGS

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

V diplomski nalogi raziščite metode za odkrivanje melodije v monofoničnih zvočnih posnetkih. Izberite najustreznejšo metodo za detekcijo višine tona in implementirajte algoritem za segmentacijo posnetka na note. Dodajte ustrezen uporabniški vmesnik in metodo preizkusite na zbirki posnetkov slovenskih ljudskih pesmi.

Mentor:

doc. dr. Matija Marolt

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Anže Mikec, z vpisno številko **63090154**, sem avtor diplomskega dela z naslovom:

Odkrivanje melodije v zvočnih zapisih slovenskih ljudskih pesmi

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matije Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 3 . septembra 2013 Podpis avtorja:

Zahvaljujem se doc. dr. Matiji Maroltu za mentorstvo, pomoč in strokovne nasvete. Iskreno se zahvaljujem tudi svojim staršem in Neži za podporo in potrpežljivost.

Kazalo

Povzetek

Cilj diplomske naloge je analizirati obstoječe algoritme in metode za analizo značilk glasbe, nato pa najprimernejšega implementirati in preizkusiti na testnih podatkih. V primeru dobrih rezultatov bi algoritem lahko bil uporabljen za preprosto transkripcijo posnetkov ljudskih pesmi, zapetih v solo izvedbi. Po pregledu literature je bil za odkrivanje višine tona izbran algoritem YIN, za segmentacijo zvočnega zapisa pa je bila uporabljena programska knjižnica MIRToolbox. Za glajenje podatkov po obdelavi z algoritmom YIN je poskrbel preprost verjetnostni model, ki temelji na podatkovni bazi KernScores. Končni izdelek je implementiran v obliki uporabniškega vmesnika, ki uporabniku omogoča izbiro posnetka za obdelavo, na koncu pa prikaže rezultate. Uporabnik lahko med poslušanjem posnetka istočasno sledi zapisu niza tonov.

Ključne besede

avtomatska transkripcija glasbe, algoritem YIN, MIRToolbox, avtokorelacija, odkrivanje značilk glasbe, odkrivanje višine tona

Abstract

The goal of this bachelor's thesis is to analyse existing algorithms and methods for musical feature extraction, implement the most appropriate algorithm and test it against the given test data. Implemented algorithm could then be used for simple melody transcription of sung Slovenian folk songs, if the results would have shown to be promising. After literature review, YIN algorithm was chosen for pitch detection. Segmentation was performed by a library of implemented algorithms called MIRToolbox and estimated values were post-processed with a simple probabilistic model, based on the KernScores library. The final product was fitted with a user interface that enables users to choose a recording they wish to analyse and displays the results when the algorithm is finished. A user can then listen to the recording and simultaneously track its position on a graph.

Keywords

automatic music transcription, YIN algorithm, MIRToolbox, autocorrelation, musical features extraction, pitch detection

Poglavje 1

Uvod

Odkrivanje in analiza osnovnih značilk zvočnih signalov v računalništvu vsekakor ni nova moda. Področje že več desetletij raziskuje množica raziskovalcev tako v znanstveni sferi kot v gospodarstvu. Splošnost pojma glasbe, veliko število glasbenih žanrov, inštrumentov in osebnih interpretacij drugih konceptov, povezanih z glasbo, kljub obilici različnih algoritmov in teorij onemogočajo zadostno natančno analizo značilk glasbe. Ravno zaradi množice različnih ritmov, inštrumentov, žanrov, tonalitete in kombinacije teh konceptov ideja o splošnem algoritmu, ki bi vračal rezultate z zadovoljivo natančnostjo, meji na nemogoče. Delo, opisano v tej diplomski nalogi, je prav zato specifično omejeno na odkrivanje višine tonov v solističnih slovenskih ljudskih pesmih in njihovo medsebojno povezanost v melodijo. Značilke, kot so ritem, tempo, tonaliteta in harmoničnost, niso del te diplomske naloge. Za reševanje te problematike se najpogosteje uporablja algoritem YIN [3] v kombinaciji z verjetnostnimi modeli, ki temeljijo na skritem Markovem modelu in dogodkovnem modelu, kjer gradnik predstavlja nota.[1] Implementirani sistemi pogosto odkrivajo tudi ritem in tempo, kar omogoča natančnejšo transkripcijo. Bolj napredni uporabljajo muzikološke modele, s katerimi odkrivajo tonaliteto in druge karakteristike, ki jih podrobneje obravnava glasbena teorija. Glasbenonarodopisni inštitut ZRC SAZU poseduje obsežno zbirko ljudskih pesmi. Za testiranje algoritma je bilo izbranih 26 posnetkov pesmi,

zapetih v izvedbi amaterjev solistov iz različnih delov Slovenije. Pesmi so posnete v naravnem okolju, izven studiov. V podatkih se pojavlja obilica šuma, ki je posledica motečih dejavnikov okolja, slabše snemalne opreme, nešolanih solistov in pomanjkanja notnih zapisov. Cilj naloge je analizirati obstoječ nabor algoritmov za odkrivanje značilk glasbe, najprimernejšega implementirati in doseči zadovoljivo natančnost rezultatov. Končni rezultat implementacije je približek zapisa tonov melodije analiziranih pesmi. Algoritem, ki je bil razvit, kot vhod sprejme zvočni zapis v obliki datoteke .wav. Na podlagi zvočnega zapisa izračuna ocene nastopov tonov in ocene frekvenc, ki predstavljajo višino tonov. Te frekvence algoritem s vnaprej izračunanimi verjetnostmi prehodov med toni pretvori v vrednosti MIDI. Niz ocenjenih tonov nato izriše na uporabniški vmesnik, kjer jih lahko uporabnik referencira s predvajanim posnetkom. Implementiran algoritem izriše ocenjene višine tonov in uporabniku omogoča, da med poslušanjem glasbenega izseka na grafu sledi njegovi trenutni poziciji. Analiza rezultatov na prvi pogled ni vzpodbudna, vendar se po podrobnejšem pregledu izkaže za logično – referenčni podatki namreč ne upoštevajo napak pri petju. Uporabljen verjetnostni model se ni izkazal kot uporaben. Vzrok za to lahko leži v dejstvu, da je učna množica premajhna in premalo sorodna testnim podatkom. Tekom raziskovanja in dela se je odprlo nekaj novih vprašanj. Motivacija za nadaljnje delo je zapisana v sklepnem delu.

Poglavje 2

Pregled pojmov ter pogostih algoritmov in metod

V tem poglavju so opisani in razloženi osnovni pojmi ter tehnike, ki se pojavljajo v glasbi in digitalni analizi glasbe.

2.1 Osnovni termini v glasbi

Najbolj temeljna glasbena enota je *nota*, definirata jo višina tona in trajanje. Višine tonov not, ki sovpadajo z belimi tipkami na klavirju, sestavljajo lestvico C-dur. Ta sestoji iz tonov, označenih s črkami C, D, E, F, G, A in H. Več not skupaj sestavlja melodije in akorde. *Melodija* je zaporedje nizov not s prepoznavno obliko. Akord je kombinacija dveh ali več istočasno zaigranih not. Razliki med višinami tonov rečemo interval. Interval z razmerjem razlike v frekvencah tonov 2:1 se imenuje *oktava*. Vsaka oktava je v zahodni glasbi razdeljena na dvanajst not. Ta dvanajstnotni sistem opredeljuje frekvence f višin tonov not z sledečo enačbo:

$$f = 2^{\frac{x}{12}} f_{base} \tag{2.1}$$

kjer je f_{base} frekvenca referenčne note (po navadi se uporabi $f_{base} = 440Hz$), x pa je celoštevilski odmik od referenčne note. Interval med dvema zaporednima tonoma se imenuje *polton*. Višino tona se lahko spremeni tudi z

alteracijo, ki jo označimo z višajem ali nižajem. Višaj (♯) ton zviša za pol tona, medtem ko ga nižaj (♭) za pol tona zniža [1].

2.1.1 Višina tona

Definicija višine tona po standardu ANSI iz leta 1994 [2]:

Definicija 1 *Pitch is that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high. Pitch depends mainly on the frequency content of the sound stimulus, but it also depends on the sound pressure and the waveform of the stimulus.*

Prevod: *Višina tona je lastnost slušne zaznave, ki omogoča razvrstitev zvokov v lestvico od najnižjega do najvišjega. Višina tona je odvisna predvsem od vsebnosti frekvenc zvočnega dražljaja ter tudi zvočnega tlaka in oblike valovnega signala dražljaja.*

Višina tona je močno povezana s frekvenco, vendar izraza ne smemo enačiti. Frekvenca je objektiven, znanstveni pojem, medtem ko je višina tona subjektivne narave. Zvočni valovi sami po sebi nimajo višine tona. Njihovo nihanje je možno izmeriti, da dobimo frekvenco. Za preslikavo notranje kakovosti višine tona so potrebni človeški možgani. Zvok, ustvarjen s katerim koli inštrumentom, proizvede več načinov vibracij, ki se pojavijo istočasno. Vibracija z najnižjo frekvenco se imenuje *osnovna frekvenca*. Preostale frekvence imenujemo *prizvoki* (*overtones*). *Harmoniki* so pomembna skupina prizvokov. Njihove frekvence so večkratniki osnovne frekvence. [4]

2.1.2 Ritem

Večina zvrsti glasbe, plesa in poezije vzpostavijo in vzdržujejo osnovno *metrično stopnjo*. Ta predstavlja osnovno enoto časa, *pulz* ali *takt*, ki je lahko slišna ali implicirana. Sestoji iz ponavljajoče se serije identičnih (pa vendar posebnih) kratkih dražljajev, ki jih zaznamo kot točke v času. Pulz oziroma

udarec ni nujno najhitrejša ali najpočasnejša komponenta ritma ampak tista, ki jo zaznamo kot osnovno.

Metrična struktura glasbe vsebuje meter, tempo in vse druge ritmične aspekte, ki proizvedejo časovno pravilnost, na katero so projecirani osprednji detajli vzorcev trajanja glasbe. V zahodni glasbi je terminologija na tem področju slavno netočna. Ritem lahko na splošno razdelimo na metrični ritem, merjeni ritem in prosti ritem. Metrični ritem je daleč najpogostejši v zahodni glasbi.[13]

2.1.3 Barva tona

Barva tona opisuje lastnosti glasbenih zvokov ali tonov, ki razlikujejo različne tipe proizvajanja zvokov (na primer človeški glasovi, brenkala, tolkala, pihala, trobila in podobno). Dve izmed fizičnih lastnosti zvoka, ki določata zaznavanje barve zvoka, sta spekter in ovojnica. Preprosto rečeno je barva zvoka tisto, kar naredi določen zvok drugačen od nekega drugega zvoka, tudi če sta enako glasna in visoka.[5] Izmed vseh značilk je barva tona najbolj subjektivne narave.

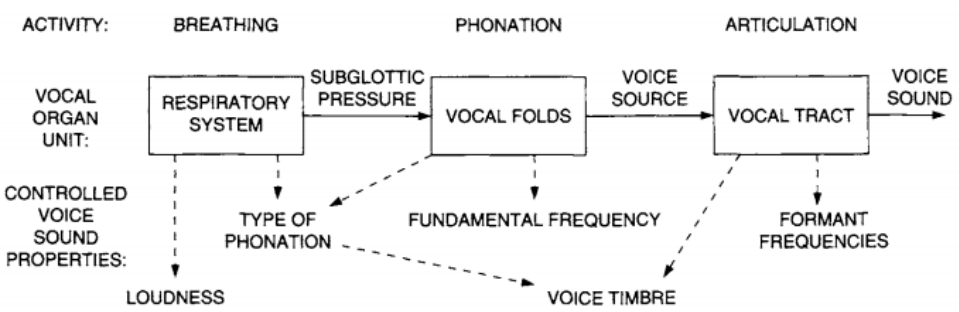
2.1.4 Petje

Problemska domena te diplomske naloge je omejena na solistično petje, zato je smiselno povedati več o tem, kako petje nastane. Glasove pri petju proizvaja človeški vokalni organ, ki je sestavljen iz treh osnovnih enot:

1. dihalni sistem,
2. glasilke,
3. vokalni trakt.

Proizvajanje zvoka poteka na sledeč način. Dihalni sistem ustvari nadtlak zraka v pljučih, kar posledično povzroči pretok zraka skozi glasilke. Glasilke

začnejo vibrirati in s tem razsekajo tok zraka v sekvence kvaziperiodičnih zračnih pulzov. To proizvede zvok, ki mu lahko izmerimo osnovno frekvenco. Zaporedje zračnih pulzov se imenuje *vir glasu*, proces proizvajanja zvoka s pomočjo vibracije glasilk pa se imenuje *fonacija*. V zadnjem koraku gre zvok skozi vokalni trakt, ki spremeni obliko spektra in določi barvo zvoka. Ta korak, ki je zaslužen za različne zvoke govora, se imenuje *artikulacija*. Diagram 2.1 prikazuje korake opisanega postopka.[8]



Slika 2.1: Koraki postopka proizvajanja glasu

2.1.5 Osnovna frekvenca pri petju

Pri petju osnovno frekvenco uravnavamo z glasilkami. Izraz frekvenca fonacije (*phonation frequency*) opisuje frekvenco vibriranja glasilk. Pri zvokih petja je to osnovna frekvenca proizvedenega tona. Vibracijo v glavnem nadzorujejo mišice glasilk.[8]

2.1.6 Neuglašeno petje

Neuglašeno petje se nanaša na situacijo, v kateri se višina tona zapete note nadležno razlikuje od uglašenosti drugih tonov. Glede na raziskave s poslušalci lahko frekvenca fonacije od nominalnega tona odstopa za ±0,07 poltona, preden poslušalec zazna neuglašenost. Odstopanja, ki so v povprečju večja od ±0,2 tona, so sprejemljiva na nepoudarjenih metričnih mestih, med deli s tragičnim razpoloženjem ali kadar je osnovna frekvenca previsoka.

Težave z uglašenostjo se pogosto pojavijo tudi pri nastopih brez spremljave. Pevci na splošno niso sposobni popolnoma uglašenega petja. Uglašenost se lahko med petjem sčasoma tudi postopoma spremeni.[8]

2.2 Standard MIDI

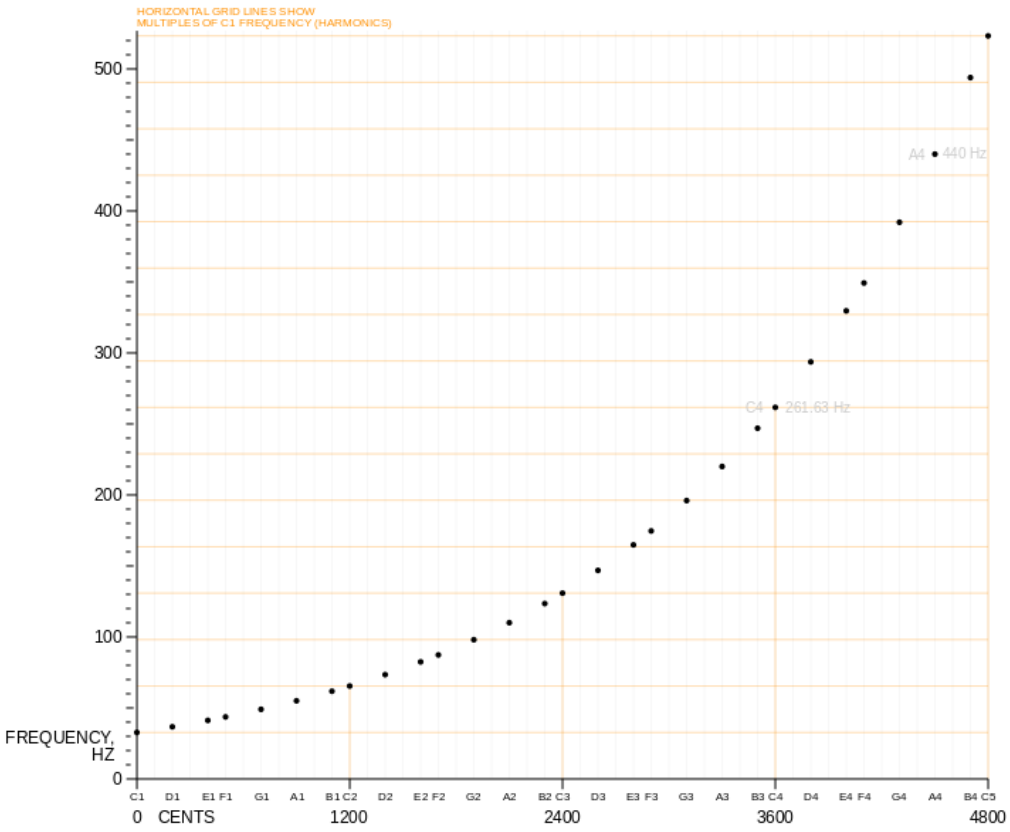
MIDI je tehnični standard, ki opisuje protokol, digitalni vmesnik in priključke ter omogoča komunikacijo med velikim spektrom glasbenih inštrumentov, računalnikov in sorodnih naprav. Zasnovan je bil za uporabo s klaviaturami, zato strukturo sporočil narekujejo glasbeni dogodki. Primer tega je izbira note, ki jo lahko zaigramo, lahko pa ji tudi nastavimo parametre, ki jih drugače najdemo na elektronskih klaviaturah. MIDI omogoča pošiljanje sporočil, ki prek parametrov določajo notacijo, višino tona, hitrost, glasnost in druge kontrolne signale.[7, 6]

$$p = 96 + 12 \times log_2(\frac{f}{440Hz}) \tag{2.2}$$

Digitalni jezik, ki ga ta protokol opisuje, uporablja notacijo s številkami, ki so preslikave višin tonov. Te lahko preslikamo v frekvence (glej enačbo 2.2). S temi lastnostmi je ta notacija ravno pravšnja za uporabo v tej diplomski nalogi.

2.3 Osnovna frekvenca

Osnovna frekvenca je definirana kot najnižja frekvenca periodičnega signala. V smislu superpozicije sinusoid (Fourierovih serij) je osnovna frekvenca najnižja sinusoida frekvenca v vsoti. V nekaterih kontekstih je osnovna frekvenca zapisana kot $f0$ ali FF, kar označuje najnižjo frekvenco, če začnemo šteti z 0. V drugih kontekstih se pogosteje uporablja oznaka $f1$ kot prvi harmonik (drugi harmonik je potem $f2 = 2 \times f1$ in podobno - v tem kontekstu bi bil



Slika 2.2: Preslikava frekvenc v note

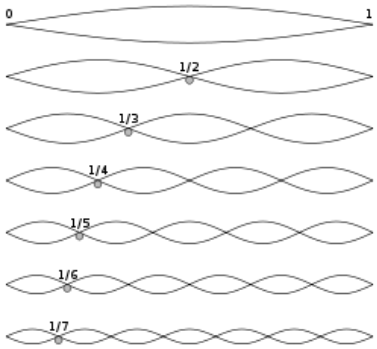
ničelni harmonik 0Hz).

Vsi sinusoidni in veliko nesinusoidnih signalov je periodičnih, kar pomeni, da se ponovijo skozi čas. Ena perioda je torej najmanjša ponavljajoča se enota signala in tako popolnoma opisuje signal. Lahko rečemo, da je signal periodičen, če najdemo periodo T, za katero velja sledeča enačba:

$$x(t) = x(t+T); t \in \mathbb{R}, \tag{2.3}$$

kjer je $x(t)$ funkcija signala. To pomeni, da je za večkratnike neke periode T vrednost signala vedno ista. Najmanj možna vrednost T, za katero velja ta enačba, se imenuje osnovna perioda, osnovna frekvenca (f_0) pa je enaka

$$f_0 = \frac{1}{T} \tag{2.4}$$



Slika 2.3: Osnovna frekvenca in prvih šest prizvokov

2.4 Algoritmi za analizo višine tonov

Proces ocenjevanja osnovne frekvenca je sestavljen iz predprocesiranja, odkrivanja višine tona in postprocesiranja. Predprocesor najprej obdela akustične signale in tako olajša odkrivanje višine tonov. Pogoste metode predprocesiranja so odstranjevanje šuma in poudarjanje tistih akustičnih značilk, ki so pomembne pri ocenjevanju višine tona. Detektor višine tonov ekstrahira obris višine tona iz predprocesiranega signala, na koncu pa postprocesor izboljša rezultat ocene (na primer popravi grobe napake in zgladi obris višine tona).

V praksi se višino tonov ocenjuje z okviri diskretnega signala, ki imajo končno dolžino. Če okvir vsebuje tišino ali šum, v njem ne obstaja določljiva višina tona. Iz tega sledi, da je pomembno oceniti tudi periodičnost signala v okviru. Temu se reče analiza izražanja (*voicing analysis*). Po navadi obstaja kriterij, na osnovi katerega lahko določimo stopnjo izražanja okvira. Analizo izražanja po navadi opravljajo algoritmi za ocenjevanje višine tona.

Algoritme za ocenjevanje višine tona lahko klasificiramo na več različnih načinov. Lahko jih združimo v skupino algoritmov frekvenčne ali časovne domene glede na njihovo domeno delovanja. Nekatere algoritme lahko pred-

stavimo v obeh domenah. Obstajajo tudi hibridni algoritmi, ki izkoriščajo obe domeni. Po drugi strani lahko algoritme razdelimo na algoritme spektralnega položaja (*spectral-place type algorithms*) in spektralnega intervala (*spectral-interval type algorithms*) glede na spektralne lastnosti, ki peljejo do zaključka o osnovni frekvenci. Algoritmi tipa spektralnega položaja rokujejo s spektralnimi komponentami glede na njihovo spektralno lokacijo, medtem ko algoritmi tipa spektralni interval raziščejo intervale med vrhovi in spektrom.[1]

2.4.1 Algoritmi časovne domene

Avtokorelacijska funkcija

Avtokorelacijska funkcija (AKF) je eno od najpogosteje uporabljenih orodij na področju ocenjevanja višine tona. S statističnega vidika je avtokorelacija pričakovana vrednost produkta signala s svojo časovno zamaknjeno verzijo. Za diskreten signal $s(k)$ v časovni domeni in okvir signala dolžine N je kratkočasovna avtokorelacija $r(n)$ definirana kot

$$r(n) = \frac{1}{N} \sum_{k=0}^{N-n-1} s(k)s(k+n), \qquad 0 \leq n \leq N-1, \tag{2.5}$$

kjer je n zamik, ki ustreza časovnemu zamiku signala. Kratkočasovna avtokorelacijska funkcija zmanjša število vzorcev za seštevanje in tako zmanjša vrednosti $r(n)$, s tem ko se vrednost zamika zvišuje. Dolžina okvira N bi morala biti karseda majhna, da ohrani časovno resolucijo, vendar dovolj velika, da zajame vsaj dve periodi osnovne frekvence. Maksimumi avtokorelacijske funkcije predstavljajo zamike, ki ustrezajo večkratnikom osnovne periode. Po navadi najdemo nekaj vrhov s približno enakimi maksimumi, kar zaplete izbiro pravega vrha. Kratkočasovna avtokorelacija prepreči detekcijo vrhov s previsokimi vrednostmi zamikov, saj se vrednosti AKF zmanjšujejo kot funkcija spremenljivke n. Poleg tega je signal tipično predprocesiran, da se detekcija z AKF olajša. Algoritmi za ocenjevanje višine tona, ki temeljijo

na avtokorelaciji, so precej robustni za glasne signale, vendar občutljivi na spektralne lastnosti ciljnih zvokov. Kratkočasovno AKF bi lahko klasificirali kot algoritem spektralnega položaja, saj obravnava spektralne komponente glede na njihovo lokacijo v spektrumu.[1]

Funkcija povprečne magnitude razlike

Funkcija povprečne magnitude razlike (*average magnitude-difference function - AMDF*) $D(n)$ za signal $s(k)$ v časovni domeni z dolžino okvira N je definirana kot

$$D(n) = \sum_{k=0}^{N-1} |s(k) - s(k+n)|, \qquad 0 \leq n \leq N-1. \tag{2.6}$$

AMDF in AKF sta si podobni, saj tudi tu primerjamo signal sam s sabo, da odkrijemo periodo. Pri AMDF primerjava temelji na absolutni razliki med okvirom signala in njegovi časovno zamaknjeni različici. Bolj kot je signal periodičen, bližje je minimum AMDF ničli. V praksi se stopnja periodičnosti določi z izbiro vrednosti pragu, pod katero morajo minimumi AMDF zadostiti želeno stopnjo periodičnosti. AMDF je računsko lažja kot AKF in je zato bolj zaželena v realnočasovni uporabi. Sistemi z aritmetiko s fiksno vejico še posebej koristijo zaradi odsotnosti množenja.[1]

Križnokorelacijska funkcija

Križnokorelacijska funkcija KKF (*cross-correlation function*) je tudi podobna AKF, le da se okvir, ki ga analiziramo, primerja s poljubnim okvirom signala. KKF $\chi(n)$ signala $s(k)$ v časovni domeni je definirana kot

$$\chi(n) = \sum_{k=0}^{N-e} s(k)s(k+n), \tag{2.7}$$

kjer je n zamik. Če je $0 \leq n \leq N-1$, postane enačba 2.7 definicija avtokorelacije. Pri KKF to ni potrebno in n je lahko večji od N, kadar se okviri

ne prekrivajo. Ker je dolžina okvira signala pri avtokorelacijskih metodah kar velika, uporaba križne korelacije prinese boljšo časovno resolucijo tudi pri ocenah nizkih tonov.[1]

Normalizirana križnokorelacijska funkcija

Normalizirana križnokorelacijska funkcija NKKF (*normalised cross-correlation function*) normalizira vrednosti KKF z energijami primerjanih okvirov. NKKF je definirana kot

$$\phi(n) = \frac{\chi(n)}{\sqrt{e_0 e_n}}, \tag{2.8}$$

kjer je e_i definiran kot

$$e_i = \sum_{k=0}^{N-1} s(k+l)^2. \tag{2.9}$$

Normalizacija zmanjša vpliv hitrih sprememb v amplitudi signala. Osnovno periodo lahko najdemo pri zamikih z vrednostmi n, kjer je $\phi(n)$ blizu 1. NKKF je robustna za glasne signale.[1]

Periodičnost ovojnice

Model periodičnosti ovojnice (*envelope periodicity*) temelji na opažanjih, da obstaja periodično nihanje v amplitudi ovojnice kompleksnih tonov. Če je ton sestavljen iz več harmonikov s približno enakim medsebojnim frekvenčnim intervalom, bodo ti povzročili utripanje v amplitudi ovojnice. Posledično bo čas med vrhovi ovojnice ustrezal osnovni periodi. Ker se zaznava frekvenčno razdaljo med harmoniki, se EP klasificira kot algoritem spektralnega intervala.[1]

2.4.2 Algoritmi frekvenčne domene

Algoritmi te skupine pri ocenjevanju višine tona procesirajo signal v frekvenčni domeni. To zahteva transformacijo signalov iz časovne domene v

frekvenčno z diskretno Fourierovo transformacijo (DFT).

Cepstrum

Cepstrum $c(n)$ signala $s(k)$ je inverzna Fourierova transformacija (IDFT), vzeta iz logaritma spektra magnitude:

$$c(n) = IDFTlog|DFTs(k)|. \tag{2.10}$$

Ker lahko AKF v frekvenčni domeni izračunamo tudi tako, kot kaže enačba 2.11, lahko rečemo, da sta AKF in analiza cepstrum podobni. Cepstrum uporablja logaritem kvadriranja.

$$r(n) = IDFT|DFTs(k)|^2 \tag{2.11}$$

Čeprav sta si podobni, obstaja opazna razlika v robustnosti v prid AKF pri večji glasnosti. Po drugi strani se cepstrum dobro odreže pri govoru z močnimi strukturami formantov in se zato pogosto uporablja v sistemih za analizo govora. Podobno kot avtokorelacijske metode se cepstrum klasificira kot algoritem spektralnega položaja.

Metoda spektralne avtokorelacije

Metoda za ocenjevanje višine tona s spektralno avtokorelacijo dela z lastnostmi spektra harmoničnih zvokov. Harmonični deli kompleksnega tona so med sabo ločeni z enakim intervalom frekvence. Najbolj očiten interval frekvence, ki ustreza osnovni frekvenci, se odkrije z izračunom AKF nad spektrom magnitude. Če ima kompleksen ton močnejšega harmonika (na primer na vsakem drugem večkratniku osnovne frekvence), je večja možnost, da dobimo dvakrat prenizko oceno višine tona, medtem ko so dvakrat previsoke ocene zelo malo verjetne. Metoda s spektralno avtokorelacijo se šteje k metodam spektralnega intervala.

Ujemanje harmonikov

Metode ujemanja harmonikov iščejo vrhove v spektru magnitude, merijo relacije harmonikov in vrhov ter jih nabirajo, da odkrijejo osnovno frekvenco.

Poglavje 3

Samodejna transkripcija melodije

Osnovni cilj je implementirati algoritem, ki bi omogočal samodejni zapis melodije glasbenega sestavka. V tem primeru je glasbeni sestavek posnetek slovenske ljudske pesmi, zapete v solo izvedbi. Domena je s tem omejena na monofonične glasbene zapise, kjer se v določenem trenutku pojavi le en ton. S tem se znebimo problematike razločevanja različnih frekvenc v spektru in odkrivanja več tonov, ki se pojavijo istočasno.

Algoritmi, ki vračajo notne zapise predvajanih posnetkov, poleg višine tona odkrivajo tudi ritem in tempo. Ta sta pomembna za boljšo segmentacijo posnetka in natančnejšo definicijo trajanja osnovnega gradnika transkripcije. Pri posnetkih z več glasbili se osredotočijo na tiste, ki so zaslužni za ritem. V veliko pomoč so glasbila, kot sta boben ali bas kitara, ki proizvajata zvok v nižjem frekvenčnem območju. To nam omogoča, da se osredotočimo le na to območje in lažje izluščimo relevantne podatke.

Ker testni podatki, vključeni v to diplomsko delo, sestojijo zgolj iz posnetkov solo petja, poudarek leži na višini tonov in ne na segmentaciji na posamezne note. Lahko bi argumentirali, da je za pravilno transkripcijo melodije

vendarle potrebna tudi pravilna segmentacija. Trajanja posameznih not, ritmičnimi poudarki, tempo in dinamika namreč vsekakor močno vplivajo na to, kako skladba zveni. Kljub tem dejstvom se to delo omejuje na specifičen del melodije. Rezultat, ki ga pričakujemo, je zaporedje tonov z definiranimi višinami. Trajanja posameznih enot v dobljenem nizu ne bodo sledila nekemu definiranemu vzorcu, bodo pa odraz posnetka petja.

3.1 Transkripcija glasbe

Enega izmed možnih pogledov na problematiko transkripcije lahko dobimo s proučevanjem procesa zavestne transkripcije, ki jo opravljajo glasbeniki, in strategij, ki jih uporabljajo. Cilj je določiti zaporedje akcij oziroma korakov, ki vodijo k rezultatu transkripcije. Branje in pisanje glasbe je pridobljena zmožnost in je zato močno odvisna od ustanov, ki to učijo. V tem kontekstu se za vajo, pri kateri je treba na podlagi predvajanega oziroma zaigranega glasbenega izseka zapisati note, uporablja izraz *glasbeni narek*.
Značilno za usposabljanje posluha je, da poudarek ni na tem, da slišimo več, ampak da prepoznamo, kar slišimo. Cilj postane natančno prepoznati odnose med zvoki. Učencem se predstavi različne intervale tonov, ritme in akorde, nato pa se jih nauči, da jih prepoznajo. Sprva se posamično obravnava preproste primere, na kompleksnejše pa se preide, ko so učenci dovolj seznanjeni s preprostimi primeri. Melodije se tipično obravnava kot sintezo višine tonov in ritma. Primer tega je, da si učenci najprej zapomnijo glasbeni izsek, ki ga bo treba zapisati, potem zapišejo višine not, na koncu pa dodajo ritem. Vaje iz posluha so očitno predvidene za posameznike z vsaj povprečnim posluhom, ki lahko zaznajo različne zvoke, njihove višine in časove v glasbenih izsekih. To so aspekti, ki jih je v računalništvu težko predstaviti.[14]

Proces odkrivanja in zapisovanja melodije je sestavljen iz več korakov:

- segmentacija zvočnega zapisa,
- priprava podatkov - čiščenje,

- ocena osnovne frekvence posameznih segmentov,
- pretvorba ocene osnovne frekvence v zapis MIDI,
- glajenje podatkov z verjetnostnim modelom.

3.2 Orodja

Uporabljena so bila orodja, ki se na področju analize glasbe in signalov uporabljajo največ. Obilica implementiranih in optimiziranih programskih knjižnic uporabniku omogoča, da ne izgublja časa s pisanjem svojih implementacij matematičnih funkcij in metod, ampak se osredotoči na raziskovanje in preizkušanje algoritmov in tehnik.

3.2.1 Matlab

MATLAB® (matrix laboratory), ki ga je razvilo podjetje MathWorks, je okolje za numerične izračune in programski jezik četrte generacije. Omogoča manipulacijo z matrikami, izris krivulj in podatkov, implementacijo algoritmov, izdelovanje uporabniških vmesnikov in povezovanje s programi, napisanimi v drugih programskih jezikih.[10] Zaradi obsežnih knjižnic matematičnih funkcij ter metod za analizo in procesiranje zvočnih signalov in glasbenih datotek popolnoma ustreza potrebam te diplomske naloge.

3.2.2 MIRToolbox

MIRToolbox je odprtokodna programska knjižnica, ki je namenjena uporabi v okolju Matlab. Razvita je bila v kontekstu evropskega projekta "Tuning the Brain for Music"(Uglaševanje možganov z glasbo). Sestoji iz obširne zbirke metod in funkcij za odkrivanje značilk glasbe, ki se nanašajo na ritem, višino in barvo tona.
Uporaba metod te programske knjižnice je na prvi pogled zelo privlačna ideja, vendar se je pozneje izkazalo, da otežujejo procesiranje podatkov, saj

nekatere funkcije delujejo kot "črna škatla", kjer uporabnik nima dostopa do podatkov v vmesnih korakih. Zato je v tej diplomski nalogi iz te knjižnice uporabljena le funkcionalnost iskanja nastopov tonov. Algoritem in njegova implementacija sta se za to specifično domeno namreč izkazala za bolj uporabna od drugih preizkušenih.

3.3 Pregled uporabljenih algoritmov in metod

V sklopu priprav in proučevanja literature je bila opravljena tudi primerjava obstoječih algoritmov in implementacij. Izbira ni bila trivialna, saj raziskave in delo na tem področju potekajo že vrsto let. Seznam obstoječih algoritmov ni kratek, treba pa je omeniti, da je večina namenjenih specifičnim problemskim domenam, kar je po poglobljeni primerjavi zmanjšalo množico uporabnih algoritmov. Algoritem mora biti dovolj preprost, da ustreza ravni zahtevnosti diplomske naloge, in dovolj prilagodljiv, da lahko obdela dane podatke in vrne uporabne rezultate.

3.3.1 Algoritem YIN

Algoritem je bil zasnovan za ocenjevanje osnovne frekvence v govoru in glasbi. Temelji na znani metodi avtokorelacije s popravki, ki odpravljajo napake. Testiranje s podatkovno bazo glasovnih posnetkov je pokazalo, da ima v primerjavi s konkurenčnimi metodami približno trikrat nižjo stopnjo napak. Ker pri iskanju nima zgornje meje frekvenčnega območja, je algoritem primeren za obdelavo glasov in glasbe z visokimi zvoki. Algoritem se lahko implementira učinkovito in z nizkimi latencami, saj je relativno preprost. Vsebuje majhno število parametrov, ki jih je treba nastaviti. Temelji na modelu signala (periodičnega signala), ki se lahko razširi na več načinov, da omogoča obdelavo različnih oblik aperiodičnosti.[3]

Algoritem je sestavljen iz sledečih korakov:

1. Izračunaj funkcijo kvadratne razlike $d(\tau)$ za željen razpon vrednosti zamikov τ:

$$d(\tau) = \sum_{n=0}^{N-1} (s(n) - s(n+\tau))^2 \tag{3.1}$$

2. Oceni funkcijo kumulativne povprečne normalizirane razlike $d'(\tau)$:

$$d'(\tau) = \begin{cases} 1, & \tau = 0 \\ \frac{d(\tau)}{[(1/\tau)\sum_{j=1}^{\tau} d(j)}, & \text{drugače} \end{cases} \tag{3.2}$$

3. Poišči τ z najmanjšo vrednostjo, za katero obstaja lokalni minimum $d'(\tau)$, ki je manjši od danega absolutnega praga κ. Če take vrednosti ni, poišči lokalni minimum $d'(\tau)$. Naj bo $\hat{\tau}$ iskana vrednost zamika.

4. Interpoliraj vrednosti funkcije $d'(\tau)$ na abscisah $\hat{\tau} - 1, \hat{\tau}, \hat{\tau} + 1$ s polinomom druge stopnje. Poišči minimum polinoma v razponu $(\hat{\tau} - 1, \hat{\tau} + 1)$, da dobiš oceno osnovne periode.

Algoritem YIN vsebuje nekaj lastnosti, ki izboljšajo njegovo natančnost in robustnost. Drugi korak normalizira vrednosti funkcije kvadratne razlike s povprečenjem krajših vrednosti zamikov. S tem ohrani vrednosti funkcije $d'(\tau)$ za zamike z manjšimi vrednostmi visoke. To prepreči algoritmu, da za oceno osnovne periode izbere zamik s premajhno vrednostjo. Poleg tega drugi korak normalizira funkcijo, tako da lahko za iskanje prvega minimuma funkcije uporabimo absolutni prag κ, kar pomeni, da stopnja vhodnega signala ne vpliva na zmogljivost algoritma. Tretji korak z vrednostjo praga določi niz sprejemljivih vrednosti osnovne periode. Zadnji (četrti) korak interpolira funkcijo $d'(\tau)$, da izboljša resolucijo vrednosti zamikov; v nasprotnem primeru bi bile vrednosti osnovne periode omejene na celoštevilčne vrednosti.[1]

3.3.2 Ocenjevanje nastopa tona

Nastop tona se nanaša na začetek note ali drugega zvoka, kjer amplituda zraste z ničle do začetnega vrha. Povezan je s konceptom prehoda: vse note

imajo nastop, vendar ni nujno, da vsebujejo začetni prehod.
Različni pristopi k odkrivanju nastopa tona operirajo s časovno domeno, frekvenčno domeno, fazno domeno ali kompleksno domeno in vsebujejo iskanje

- povečanja energije spektra,
- sprememb v distribuciji energije spektra,
- sprememb v zaznani višini tona,
- spektralnih vzorcev, prepoznavnih z uporabo metod strojnega učenja.

Preprostejše tehnike, kot je zaznavanje povečanja amplitude v časovni domeni, lahko pripeljejo do manj natančnih rezultatov.[11]

V implementaciji, uporabljeni v tej diplomski nalogi, je bila za odkrivanje nastopov tonov uporabljena funkcija *mironsets* iz programske knjižnice MIRToolbox. Funkcija omogoča uporabo več parametrov, naštetih je nekaj uporabljenih:

- *Detect* - določimo, ali naj funkcija išče lokalne maksimume ali minimume,
- *Attacks* - funkcija naj išče nastope tonov,
- *Releases* - funkcija naj išče konce tonov,
- *Contrast* - kontrast, ki ga metoda upošteva pri iskanju,
- *Threshold* - prag, ki ga funkcija upošteva pri iskanju.

S parametrom *'Envelope'* funkciji določimo, naj najprej izračuna ovojnico na podlagi filtra (parameter *'Filter'*) oziroma spektra (parameter *'Spectro'*).[9]

3.3.3 Verjetnostni model

Stohastična narava signalov je motivacija za uporabo verjetnostnih metod. Vsak nastop kljub formalnosti predstavitve glasbe vsebuje napake, kot so nepravilno zapete note in melodije, nekontrolirane vibracije glasu in netočnosti v času. Če želimo pravilno transkripcijo melodije, moramo te napake odpraviti.[1] Korak v pravo smer je uporaba verjetnostnih modelov. Z dovolj veliko učno množico lahko implementiramo model, ki oceni verjetnost stanja na podlagi prejšnjih stanj. Pogosto se v sistemih za avtomatsko transkripcijo glasbe uporablja skriti Markov model, ki je v osnovi Markov model s skritimi stanji.

Odpravljanje šuma in popravljanje napak glasbenikov ni edina možna uporaba verjetnostnih modelov in statistike. Skupaj z metodami strojnega učenja omogočajo nove možnosti uporabe z novimi funkcionalnostmi. Vsekakor pa to ni novost. Obstoječi sistemi za prepoznavanje melodije, žanra ali drugih značilk so sposobni z metodami strojnega učenja priporočati podobne skladbe, na podlagi zaporedja tonov najti podatke o skladbi in še marsikaj. Čeprav delo na analizi glasbe poteka že vrsto let, novih možnosti še zdaleč ni zmanjkalo.

Poglavje 4

Implementacija algoritma

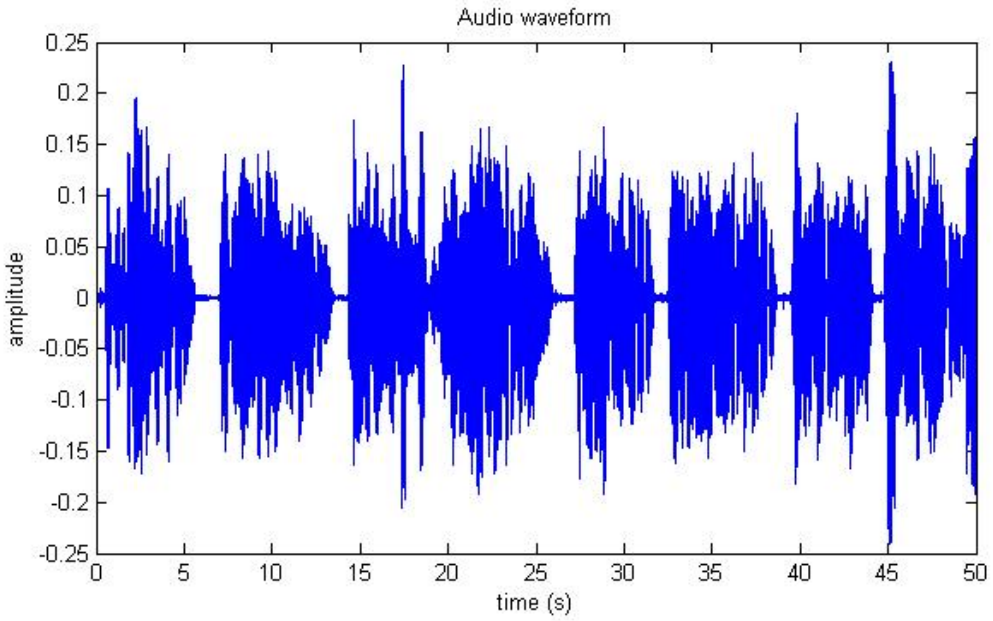
V tem podpoglavju je podrobneje opisana implementacija algoritma. Prva točka govori o uporabljenih testnih podatkih, medtem ko naslednje točke predstavljajo korake algoritma.

4.1 Podatki

Zbirko slovenskih ljudskih pesmi, zapetih v solo izvedbi, je v raziskovalne namene posredoval Glasbenonarodopisni inštitut ZRC SAZU. Izbranih je bilo 26 posnetkov, med njimi najdemo tudi več različic istih pesmi. Stanje te zbirke otežuje njihovo analizo zaradi več dejavnikov.

Besedila in melodije pesmi so del ustnega izročila in so se prenašale iz roda v rod. Posamezniki, ki pesmi izvajajo, niso šolani v petju, zato se v podatkih pojavi veliko šuma. Nemogoče je zagotovo vedeti, ali je zapet ton pravilen ali napačen, saj za pesmi ne obstajajo notni zapisi. Velik delež vrednosti zapetih glasov se pojavi med dvema tonoma, kar zopet oteži klasifikacijo po tonih.

Posnetki so bili posneti v njihovem domačem okolju in ne v studiih, posledično vsebujejo šum iz okolice. V nekaterih delih posnetkov se pojavljajo nekateri moteči dejavniki, ki jih algoritem ne prepozna kot šum (na primer



Slika 4.1: Podatki, ki jih preberemo iz datoteke.

zvoki predmetov, ki so podobni tistim, ki jih proizvaja pevec).
Del podatkov so tudi referenčne transkripcije v formatu MIDI, spremljajo jih datoteke v formatu Sibelius in tekstovne datoteke z začetki in konci kitic. Za vsak posnetek je po posluhu zapisana transkripcija za eno kitico.

4.2 Branje in segmentacija zvočnega zapisa

Posnetek preberemo z uporabo klica funkcije

```
audio = miraudio(filename, 'Extract', from, to).
```

Ta nam vrne objekt z vektorjem, ki vsebuje vzorce posnetka, in frekvenco vzorčenja. S parametrom *Extract* določimo, kateri del posnetka želimo vzorčiti.

Segmentacija zvočnega zapisa je narejena z uporabo odkrivanja nastopov tonov. Ker je v podatkih ogromno šuma, petje pa ni zanesljivo, se predpostavi, da ton traja do začetka naslednjega tona, čeprav v praksi ni vedno tako. S tem obdržimo tudi dele posnetka, kjer ni petja, ampak le šum. To

bi lahko pri ocenjevanju frekvence predstavljal velik problem, zato bomo za te anomalije morali poskrbeti v naslednjem koraku.

Segmentacija z nastopi tonov je narejena z uporabo knjižnice MIRToolbox, bolj specifično funkcije *mironsets*, ki jo kličemo na sledeč način:

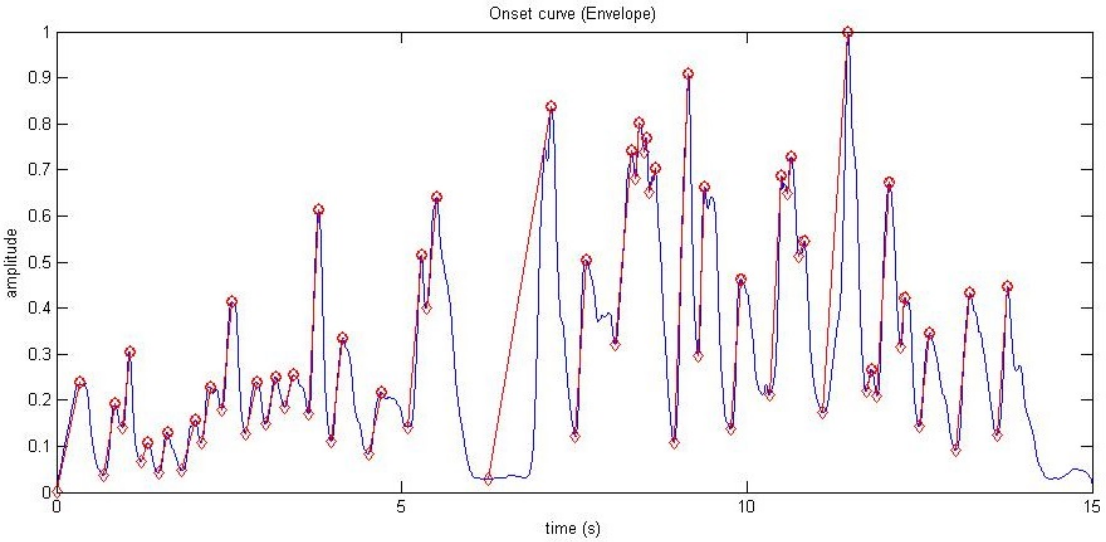
```
o2 = mironsets(audio, 'Filter', 'FilterType', 'IIR', ...
'Tau', .035, 'NbChannels', 10, 'Contrast', .03, ...
'Sum', 1, 'Attacks', 'PostDecim', 0, 'HalfwaveDiff', 1, ...
'Max', 2000, 'Min', 0, 'Detect', 'Peaks',);
```

S parametrom *Filter* funkciji povemo, da želimo nad vhodnimi podatki uporabiti filter, ki ga nato podamo takoj za parametrom *FilterType*. V tem primeru je uporabljen filter *IIR* (polovično Hanningovo okno). Nato z *NbChannels* povemo, na koliko različnih kanalov želimo vhod filtrirati. Parameter *Contrast* predstavlja kontrast, uporabljen pri iskanju vrhov, medtem ko *Tau* predstavlja časovno konstanto nizkovalovnega filtra pri računanju ovojnice signala v sekundah. Z *Max* in *Min* določimo frekvenčni razpon, v katerem sploh želimo iskati nastope. *Sum* funkciji pove, da naj po končani obdelavi kanale zopet sešteje v en kanal. Želimo, da nam vrne le nastope tonov, zato postavimo parameter *Attacks* (za konce tonov nastavimo še *Releases*).

4.3 Ocena osnovne frekvence segmenta in preslikava v predstavitev MIDI

Ocenjevanje osnovne frekvence se vrši z uporabo algoritma YIN (glej 3.3.1). Obdelava z algoritmom poteka na celotnem posnetku in ne po posameznih segmentih. Rezultati segmentacije se uporabijo na koncu pri izrisu oziroma izpisu zaporedja tonov.

Funkcija *yinRecording* sprejme prvotni vektor vzorcev in ga prevzorči z manjšo frekvenco vzorčenja. Vektor poda algoritmu YIN, ki ga obdela. Relevantne podatke nato prečisti s filtrom mediane, neuporabne podatke pa odstrani.



Slika 4.2: Ovojnica z označenimi nastopi tonov in njihovimi konci

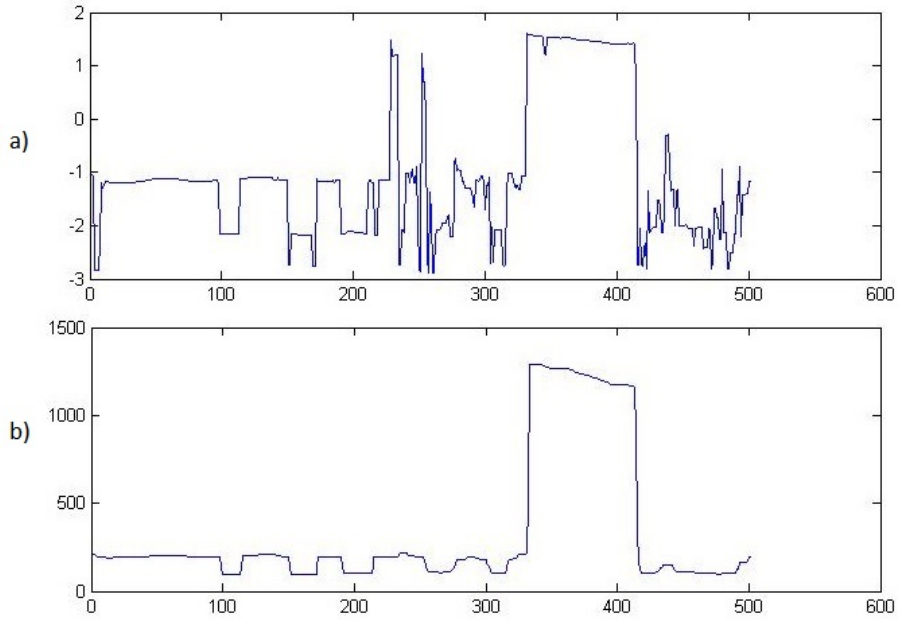
Rezultat zopet poda algoritmu YIN in še enkrat izvede čiščenje.

Ocena osnovne frekvence je na tej stopnji že kar natančna, vseeno pa jo lahko še izboljšamo. Na krivulji se pojavlja veliko "pobočij", kjer vrednosti hitro rastejo ali padajo. Ti deli krivulje predstavljajo dele posnetka, kjer solist začne ton, končuje ton ali pa del, kjer izgubi nadzor nad svojim glasom, zato teh vrednosti v oceni ne smemo upoštevati.

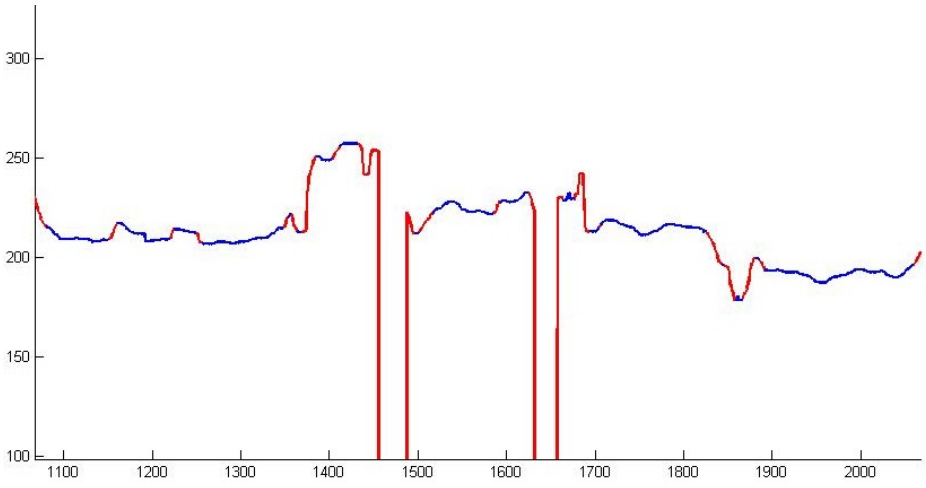
Vektor s končnimi vrednostmi tega koraka se razbije na segmente s pomočjo podatkov iz prvega podatka. Vrednosti v segmentih se nato povprečijo in preslikajo v predstavitve MIDI. Ker se vrednosti skoraj nikoli ne ujemajo, se zaokrožijo k najbližjemu celemu številu. V primerih, kjer je sredinska vrednost v sredini med dvema tonoma, se uporabi verjetnostni model iz naslednje točke.

4.4 Verjetnosti prehodov med stanji

Kot je bilo omenjeno, igrajo verjetnostni modeli eno od glavnih vlog pri samodejni transkripciji, saj omogočajo odpravljanje napak, ki jih povzročijo



Slika 4.3: Ocena osnovne frekvence, ki jo vrne algoritem YIN pred *a)* in po filtriranju *b)*.



Slika 4.4: Z rdečo barvo so označeni deli krivulje, kjer ta prehitro raste ali pada.

nastopajoči. Zavedati pa se moramo, katere tehnike in metode so primerne v določenih situacijah. Popravljanje za solisti je z danimi testnimi podatki tudi z uporabo naprednih tehnik zelo težko. Prav tako množica testnih podatkov ni dovolj velika, da bi omogočala implementacijo sistema, ki bi na podlagi učne množice zgradil verjetnostni model. Kljub temu vsebuje algoritem preprosto tehniko, na podlagi katere se odloča, kateri ton bo izbral, če je ocenjena vrednost med dvema tonoma.
Ocene verjetnosti prehodov med toni so bile izračunane na podlagi podatkov iz knjižnice ljudskih pesmi KernScore [12], ki poleg drugih vsebuje tudi veliko zbirko transkripcij evropskih ljudskih pesmi. Treba je omeniti, da so karakteristike ljudskih pesmi različnih evropskih narodov zelo različne. Ocena verjetnosti zato upošteva le pogostost prehodov iz enega tona v drugega, neodvisno od tonalitete. Transkripcije v podatkovni bazi so v formatu *kern*. Uporabljene so bile transkripcije ljudskih pesmi iz Češke, Poljske, Romunije, Rusije in bivših držav Jugoslavije, saj je bilo ocenjeno, da so te po karakteristikah slovenskim bližje kot druge. Preprosta skripta prebere prenesene datoteke s transkripcijami, prečisti podatke in v podatkovno strukturo slovar zapiše verjetnosti prehodov. Vse nižaje pretvori v predstavitev z višajem, da poenoti množico tonov.

4.5 Uporabniški vmesnik

V okolju Matlab je bil izdelan preprost uporabniški vmesnik, ki omogoča izbiro posnetka, ki ga želimo analizirati, in pregled rezultatov algoritma. Interakcija se začne s preprosto uporabniško kontrolo, ki vsebuje kombinirano izvlečno listo s seznamom posnetkov in gumb *Analiza vseh* (glej sliko 4.6). Algoritem se zažene takoj, ko s klikom na element kombinirane izvlečne liste izberemo željen posnetek. V interaktivnem oknu okolja Matlab lahko sledimo izpisom programa in vidimo, kako algoritem napreduje oziroma na kateri stopnji analize se program trenutno nahaja. Gumb *Analiza vseh* zažene primerjavo dobljenih rezultatov z referenčnimi podatki in na koncu ne izriše

```
>> result.probability.E1

ans =

    count: 321
       D1: 0.3333
       F1: 0.1028
       C1: 0.0841
     Fis1: 0.1028
       E1: 0.1682
       A1: 0.0343
       G1: 0.0748
       C2: 0.0062
     Dis1: 0.0187
       H1: 0.0312
     Gis1: 0.0249
       E2: 0.0062
       D2: 0.0062
       B1: 0.0031
       H2: 0.0031
```

```
>> result.probability.D2

ans =

    count: 542
       E2: 0.0996
       C2: 0.3672
       A1: 0.0627
       G1: 0.0203
       D2: 0.1771
       H1: 0.1531
       H2: 0.0314
     Ais1: 0.0258
     Dis2: 0.0129
     Ais2: 0.0018
       G2: 0.0314
     Fis2: 0.0055
       D1: 0.0074
     Fis1: 0.0037
```

Slika 4.5: Verjetnosti prehodov iz tonov E1 in D2 v druge tone

ocenjene melodije.

Začetno uporabniško kontrolo zaženemo z ukazom

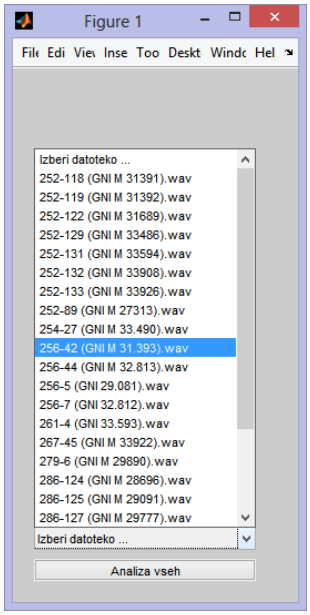
```
analyzeFolder(<direktorij>, <dolžinaPosnetka>),
```

kjer parameter *direktorij* predstavlja pot do direktorija, ki vsebuje posnetke, *dolžinaPosnetka* pa je dolžina, ki jo želimo obdelati. Zadnji parameter je podan zgolj iz performančnih razlogov, saj je lahko obdelava daljših posnetkov performančno zelo zahtevna.

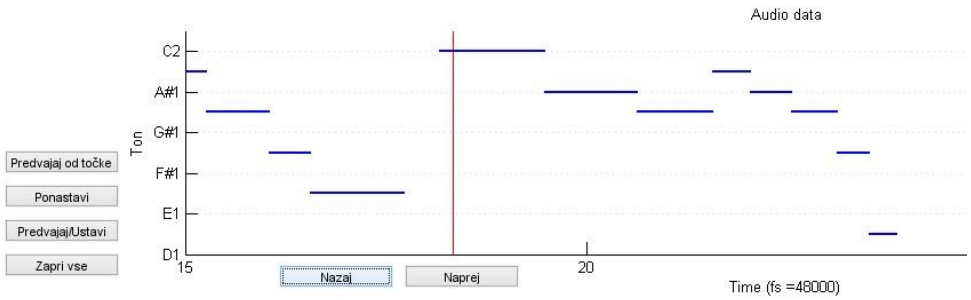
Program v zadnjem koraku prikaže novo uporabniško kontrolo (4.7), na njej pa izriše graf z dobljenim nizom tonov, kjer os x predstavlja časovno domeno, os y pa višino tona, ki je označena z ustrezno črko. Uporabniška kontrola omogoča poslušanje analiziranega posnetka in med predvajanjem s povratnim klicem izrisuje trenutni položaj na grafu.

Uporabnik lahko rokuje z uporabniško kontrolo prek gumbov

- *Predvajaj od točke*, ki začne predvajati posnetek na želenem mestu na grafu,



Slika 4.6: Uporabniški vmesnik za izbiro zvočnega posnetka



Slika 4.7: Uporabniški vmesnik z izrisanim nizom tonov ter gumbi za rokovanje s posnetkom in grafom

- *Ponastavi*, ki ponastavi posnetek na začetno mesto,
- *Predvajaj/Ustavi*, ki predvajanje začne ali ustavi,
- *Zapri vse*, ki zapre vse uporabniške kontrole,
- *Naprej*, ki pomakne graf za en časovni interval naprej,
- *Nazaj*, ki pomakne graf za en časovni interval nazaj.

Poglavje 5

Analiza rezultatov algoritma

5.1 Referenčni podatki

Referenčne podatke je v formatu datotek MIDI (.mid) priskrbel Laboratorij za računalniško grafiko in multimedije s Fakultete za računalništvo in informatiko. Vsaka datoteka MIDI vsebuje transkripcijo ene kitice posnetka, ki mu pripada. Vse transkripcije so zapisane v isti tonaliteti in predstavljajo zapis, ki se drži pravil teorije glasbe. Zapis se drži standarda, opisanega v podpoglavju *Standard MIDI* (2.2). Poudariti je treba, da to zmanjša težo rezultatov analize implementiranega algoritma, saj solisti na posnetkih ne pojejo točno. Transkripcije v obliki datotek MIDI so torej približki. Vsak posnetek spremlja tudi tekstovna datoteka, ki vsebuje podatke o približnih začetkih in koncih posameznih kitic v posnetkih. Začetki in konci so bili ocenjeni s poslušanjem posnetkov.

5.2 Postopek analize

Algoritem obdela vsak posnetek posebej in nato niz tonov v predstavitvi MIDI pošlje funkciji za analizo. Funkcija za dan posnetek najprej v pomnilnik zapiše transkripcijo ter čase začetkov in koncev kitic iz pripadajočih datotek. Zapisana trajanja kitic, pridobljena s poslušanjem, se v nekaterih primerih

razlikujejo s trajanjem transkripcij MIDI tudi za nekaj sekund. To neskladje funkcija odpravi in prilagodi niza, tako da se po razporeditvi in časih ujemata (tisti iz datoteke MIDI se prilagodi, da ustreza dolžini kitice iz tekstovne datoteke).

Niz celotnega posnetka, ki ga je vrnil algoritem, se razdeli na manjše nize (kitice), tako da ustrezajo začetkom in koncem kitic, kot so označene v referenčnih podatkih. Vsaka dobljena kitica se nato primerja z referenčno kitico.

Kitica se enakomerno razbije na n vzorcev. Primerjava kitic poteka tako, da se algoritem sprehodi po obeh nizih in za vsak vzorec izračuna razdalje med tonom trenutnega vzorca in toni predhodnih vzorcev. Vsaka razdalja med tonom vzorca in tonom predhodnega vzorca se primerja z istoležno razdaljo v referenčni kitici in prišteje normalizirani vsoti razdalj za trenutni vzorec. Algoritem sešteje in povpreči normalizirane vrednosti razdalj za vsak vzorec. Dobljena vrednost predstavlja stopnjo neujemanja med referenčno in primerjano kitico. Postopek ponovimo za vse kitice v posnetku in zopet povprečimo. Končna vrednost predstavlja povprečno stopnjo neujemanja niza ocenjenih tonov z referenčnim nizom.
Postopek primerjave kitic je prikazan tudi na 5.2.

5.3 Rezultati analize

Analiza rezultatov je osredotočena na tri specifične dele oziroma korake algoritma: segmentacija, odkrivanje višine tona in verjetnostni model. Vrednosti v priloženih tabelah predstavljajo stopnjo ujemanja z referenčnimi podatki. Vrednosti so v razponu med 0 in 1, kjer 1 predstavlja popolno ujemanje. Ker se referenčni podatki že v osnovi razlikujejo od stanja na posnetkih, pričakujemo nižjo stopnjo ujemanja. Ta predpostavka pripelje do dejstva, da rezultati z visoko stopnjo ujemanja temeljijo na anomalijah. Po podrobnejši analizi je opazno, da je pri vzorcih z visoko stopnjo ujemanja rezultat, ki ga vrne algoritem, preveč ali premalo segmentiran. Če je segmentiran preveč,

```
function CompareStanzas(refStanza, estStanza)
    diff = 0;

    for i = 1:length(est)
        innerDiff = 0;

        for j = 1:i
           refDiff = ref(i)-ref(j);
           estDiff = est(i)-est(j);

           if abs(refDiff-estDiff) > 0
               innerDiff = innerDiff + 1;
           end
        end

        if innerDiff > 0
            diff = diff + innerDiff/i;
        end
    end

    diff = 1-diff/length(est);
end
```

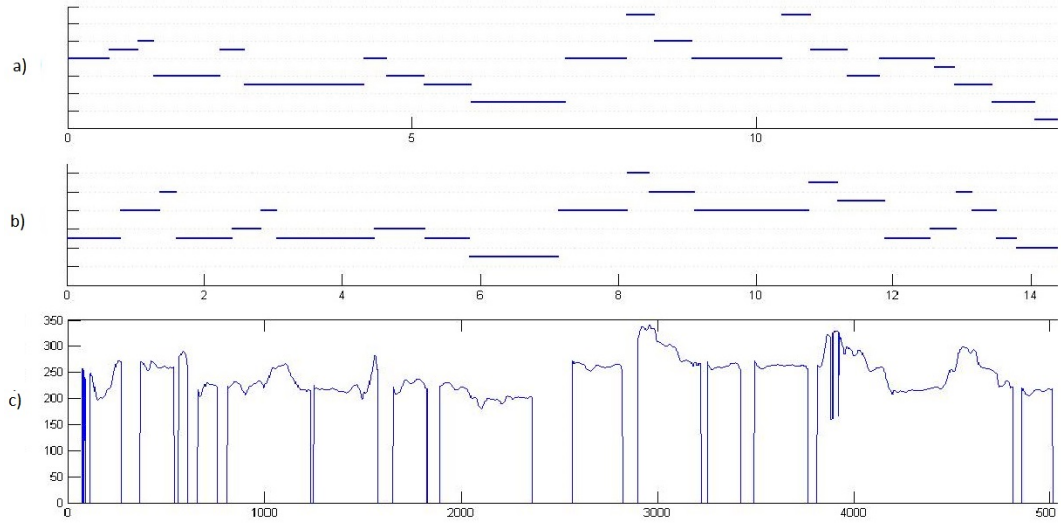
Slika 5.1: Metoda, ki primerja dve kitici oziroma dva niza. Spremenljivka *refStanza* predstavlja referenčni niz, spremenljivka *estStanza* pa niz, ki ga vrne algoritem.

dobimo enakomernejšo distribucijo možnih intervalov, torej je tudi verjetnost ujemanja večja. Pri manjši segmentaciji je število korakov primerjave manjše in ni dovolj vrednosti za primerjavo. Poleg tega se moramo zavedati, da so referenčni podatki zapisani po posluhu, kar ne izključuje možnost napak. Na posnetkih uporabniškega vmesnika je viden primer uspešne (5.2) in manj uspešne (5.3) samodejne transkripcije.

Podatki v tabeli 5.1 kažejo ujemanje rezultatov algoritma s testnimi podatki za različne vrednosti parametrov τ in *Contrast*. Sprva je očitno, da se algoritem odreže boljše pri nižjih vrednostih obeh parametrov, vendar so lahko podatki zavajajoči. S tem, ko nižamo vrednost parametrov, algoritem posnetek segmentira na več manjših segmentov, ki ne predstavljajo realne slike petja. Ocena osnovne frekvence temelji na povprečni vrednosti frekvenc segmenta. Kot že omenjeno, če je segmentov več, dobimo več tonov. Če solist začne segment s tonom A, konča pa pol tona višje na tonu Ais, bomo namesto povprečne vrednosti dobili dva tona. Z večjim številom različnih tonov dobimo višjo stopnjo ujemanja, saj je možnih kombinacij več. Pregled grafa z izrisanimi toni pokaže, da je najbližje realnemu stanju rezultat, kjer je $\tau = 0.035$ in $contrast = 0.03$.

Primerjano je bilo tudi delovanje z verjetnostnim modelom in brez njega. Rezultate te primerjave kaže tabela 5.2. Vidimo, da razlika med rezultati ni velika. Zanimivo je, da se algoritem odreže bolje brez verjetnostnega modela. Verjetnostni model je v tem primeru edino orodje, ki bi lahko rezultate obdelave zapetih posnetkov pripeljala bližje referenčnim podatkom. Težava je v dejstvu, da je implementiran verjetnostni model zelo preprost. Upošteva namreč le verjetnosti prehodov iz tona v ton. Pri tem se ne ozira na tonaliteto, uči pa se na majhni množici podatkov, za katere je sorodnost s testnimi podatki vprašljiva.

Tabela 5.3 kaže, da se algoritem YIN najbolj odreže z vrednostjo praga



Slika 5.2: Primer uspešnejše samodejne transkripcije (252-119 (GNI M 31392)). Prvi del a) predstavlja ocenjen niz tonov, b) predstavlja referenčno kitico, c) pa je ocena osnovne frekvence, ki jo vrne YIN. Prikazana je naključna kitica.

0,01. Višje vrednosti zmanjšajo stopnjo ujemanja. Vidimo, da so rezultati analize enaki za vrednost praga 0,02 in 0,03. Če rezultate primerjamo še z uporabniškim vmesnikom, vidimo, da pri višjih vrednostih dobljen niz tonov izgubi smisel. Vrne namreč zelo dolge tone, ki ne ustrezajo posnetku. Algoritem z vrednostjo pragu nad 0,01 ne zmore uspešno zaznavati sprememb v krivulji.

Medtem ko primerjava z referenčnimi podatki vrača slabe rezultate, je pregled rezultatov z uporabniškim vmesnikom bolj vzpodbuden. Medtem ko poslušamo posnetek, lahko opazimo veliko podobnost pri zaporedju in trajanju tonov ter obliki melodije, ki je izrisana na grafu (glej sliko 4.7).

Ime datoteke	$\tau = 0.02$, $c = 0.02$	$\tau = 0.035$, $c = 0.03$	$\tau = 0.04$, $c = 0.04$
252-118 (GNI M 31391)	0.82	0.56	0.72
252-119 (GNI M 31392)	0.48	0.33	0.26
252-122 (GNI M 31689)	0.68	0.55	0.64
252-129 (GNI M 33486)	0.82	0.61	0.59
252-131 (GNI M 33594)	0.26	0.27	0.27
252-132 (GNI M 33908)	0.23	0.22	0.23
252-133 (GNI M 33926)	0.56	0.52	0.51
252-89 (GNI M 27313)	0.92	0.52	0.52
254-27 (GNI M 33.490)	0.65	0.39	0.37
256-42 (GNI M 31.393)	0.79	0.79	0.79
256-44 (GNI M 32.813)	0.82	0.63	0.55
256-5 (GNI 29.081)	0.79	0.28	0.20
256-7 (GNI 32.812)	0.85	0.64	0.50
261-4 (GNI 33.593)	0.73	0.38	0.38
267-45 (GNI M 33922)	0.33	0.31	0.31
279-6 (GNI M 29890)	0.21	0.19	0.18
286-124 (GNI M 28696)	0.25	0.24	0.24
286-125 (GNI M 29091)	0.28	0.28	0.28
286-127 (GNI M 29777)	0.96	0.82	0.71
286-130 (GNI M 31010)	0.76	0.82	0.69
286-138 (GNI M 33449)	0.28	0.24	0.24
286-142 (GNI M 34570)	0.91	0.27	0.27
287-66 (GNI M 28695)	0.34	0.47	0.32
Skupaj	0.61	0.47	0.44

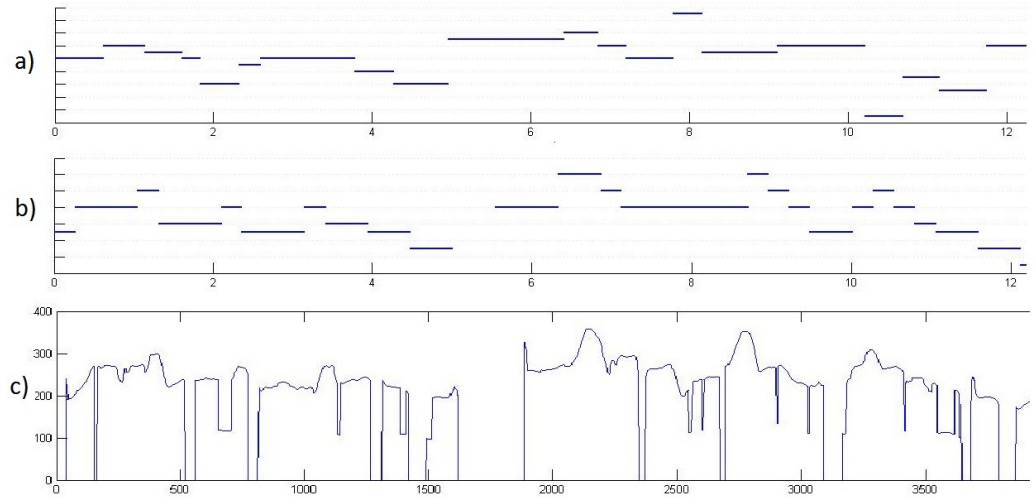
Tabela 5.1: Primerjava rezultatov različnih vrednosti parametrov τ in c (contrast), ki sta uporabljena pri odkrivanju nastopa tona oziroma segmentaciji posnetka. Vrednosti so zaokrožene na dve decimalki.

Ime datoteke	verj. model	brez verj. modela
252-118 (GNI M 31391)	0.56	0.56
252-119 (GNI M 31392)	0.33	0.33
252-122 (GNI M 31689)	0.55	0.56
252-129 (GNI M 33486)	0.61	0.61
252-131 (GNI M 33594)	0.27	0.27
252-132 (GNI M 33908)	0.22	0.23
252-133 (GNI M 33926)	0.52	0.52
252-89 (GNI M 27313)	0.52	0.52
254-27 (GNI M 33.490)	0.39	0.39
256-42 (GNI M 31.393)	0.79	0.79
256-44 (GNI M 32.813)	0.63	0.67
256-5 (GNI 29.081)	0.28	0.27
256-7 (GNI 32.812)	0.64	0.64
261-4 (GNI 33.593)	0.38	0.38
267-45 (GNI M 33922)	0.31	0.31
279-6 (GNI M 29890)	0.19	0.19
286-124 (GNI M 28696)	0.24	0.24
286-125 (GNI M 29091)	0.28	0.28
286-127 (GNI M 29777)	0.82	0.77
286-130 (GNI M 31010)	0.82	0.82
286-138 (GNI M 33449)	0.24	0.24
286-142 (GNI M 34570)	0.27	0.27
287-66 (GNI M 28695)	0.47	0.32
Skupaj	0.45	0.44

Tabela 5.2: Primerjava rezultatov algoritma z uporabo verjetnostnega modela in brez uporabe verjetnostnega modela

Ime datoteke	max f0=2000 min f0=0 prag=0.01	max f0=2000 min f0=0 prag=0.02	max f0=2000 min f0=0 prag=0.03
252-118 (GNI M 31391)	0.56	0.27	0.27
252-119 (GNI M 31392)	0.33	0.27	0.27
252-122 (GNI M 31689)	0.55	0.27	0.27
252-129 (GNI M 33486)	0.61	0.27	0.27
252-131 (GNI M 33594)	0.27	0.28	0.28
252-132 (GNI M 33908)	0.22	0.25	0.25
252-133 (GNI M 33926)	0.52	0.28	0.28
252-89 (GNI M 27313)	0.52	0.27	0.27
254-27 (GNI M 33.490)	0.39	0.24	0.24
256-42 (GNI M 31.393)	0.79	0.32	0.32
256-44 (GNI M 32.813)	0.63	0.29	0.29
256-5 (GNI 29.081)	0.28	0.20	0.20
256-7 (GNI 32.812)	0.64	0.22	0.22
261-4 (GNI 33.593)	0.38	0.29	0.29
267-45 (GNI M 33922)	0.31	0.30	0.30
279-6 (GNI M 29890)	0.19	0.20	0.20
286-124 (GNI M 28696)	0.24	0.23	0.23
286-125 (GNI M 29091)	0.28	0.28	0.28
286-127 (GNI M 29777)	0.82	0.36	0.36
286-130 (GNI M 31010)	0.82	0.24	0.24
286-138 (GNI M 33449)	0.24	0.25	0.25
286-142 (GNI M 34570)	0.27	0.26	0.26
287-66 (GNI M 28695)	0.47	0.23	0.29
Skupaj	0.45	0.23	0.23

Tabela 5.3: Primerjava rezultatov algoritma z uporabo različnih vrednosti parametrov algoritma YIN



Slika 5.3: Primer manj uspešne samodejne transkripcije (252-122 (GNI M 31689)). Prvi del a) predstavlja ocenjen niz tonov, b) predstavlja referenčno kitico, c) pa je ocena osnovne frekvence, ki jo vrne algoritem YIN. Prikazana je naključna kitica.

Poglavje 6

Sklepne ugotovitve

Implementacija algoritma, opisana v tej diplomski nalogi, z zadovoljivo natančnostjo vrne obris melodije predvajanih zvočnih posnetkov zapetih ljudskih pesmi v solo izvedbi. Algoritem prebere verjetnosti prehodov med toni, na podlagi katerih se odloča med dvema sosednima tonoma v primeru, da je višina tona ocenjena na vrednost med obema. Postopek nadaljuje s segmentacijo posnetka in ocenjevanjem osnovne frekvence celotnega posnetka. Višine tonov izlušči iz teh ocen na podlagi ocen začetkov in koncev segmentov. Uporabniški vmesnik podatke izriše, kjer jih lahko uporabnik pregleda. Izpostaviti je treba, da je to prvi, čeprav pomemben korak do delujočega sistema za transkripcijo melodije posnetkov slovenskih ljudskih pesmi. Rezultati primerjave z referenčnimi podatki res niso vzpodbudni, je pa to lahko zavajajoče. Referenčne transkripcije predstavljajo stanje, kakršno bi v veliki meri moralo biti, a v realnosti ni. Solisti niso šolani glasbeniki in ne sledijo pravilom teorije glasbe, ampak pesem zapojejo po svoji najboljših zmožnostih. Popravljanje napak solistov ni tema te diplomske naloge. Motivacija za nadaljnje delo je razviti sistem, ki bi z različnimi modeli in metodami uspešno odkril, v kateri tonaliteti se pesem nahaja, ter bil zmožen popraviti napake v petju. To vsekakor ni preprosta naloga.

Naslednji korak k izboljšavi algoritma je implementacija boljše segmentacije posnetkov. Priporočljivo je uporabiti model z dogodki, ki jih opisujejo note.

Odkrivanje dodatnih značilk, kot sta tempo in ritem, bi pripomoglo k pravilni transkripciji, s tem bi bilo možno definirati osnoven gradnik posnetka. To bi omogočilo glasbenikom, da melodijo ponovijo. Algoritem na tej stopnji ne loči med enim daljšim tonom in več zaporednimi toni z isto višino. Tudi ta opcija bi bila rešljiva v prej omenjeni razširitvi. Tako izboljšana transkripcija bi omogočala iskanje po večji zbirki transkripcij z zaporedjem tonov. Med dolgoročnejše cilje spada razširitev algoritma za ocenjevanje osnovne frekvence. Razširitev bi omogočala obdelavo posnetkov z več inštrumenti in pevci.

Raziskovanje na tem področju poteka že vrsto let, kljub temu pa obstaja še ogromno odprtih vprašanj. Področje je zelo obširno, iskanje rešitev pa poteka na specifičnih problemih. Odkrivanje značilk glasbe in sistemi za avtomatsko transkripcijo glasbe niso preprosti, vendar je zaradi intenzivnega zanimanja za področje vsekakor vredno vlagati v nadaljnje delo.

Literatura

[1] M. Ryynänen, "Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies", 2004. Dostopno na: http://www.cs.tut.fi/sgn/arg/matti/mryynane_thesis.pdf

[2] ANSI, "American national standard acoustical terminology", v zborniku: ANSI S1.1-1994, Acoustical Society Of America, New York, 1994.

[3] A. de Cheveigne, H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", v zbirniku: J. Acoust. Soc. Am., vol. 111, no. 4, strani 1917-1930, April 2002. Dostopno na: http://audition.ens.fr/adc/pdf/2002_JASA_YIN.pdf

[4] Wikipedia - Pitch(music), 2013. Dostopno na: http://en.wikipedia.org/wiki/Pitch_(music)

[5] Wikipedia - Timbre, 2013. Dostopno na: https://en.wikipedia.org/wiki/Timbre

[6] Midi, 2013. Dostopno na: http://www.midi.org/aboutmidi/tut_protocol.php

[7] Wikipedia - Midi, 2013. Dostopno na: http://en.wikipedia.org/wiki/MIDI

[8] M. Ryynänen, "Singing transcription", v knjigi: "Signal processing methods for music transcription" (A. Klapuri and M. Davy, eds.), Springer, 2006.

[9] O. Lartillot, "MIRtoolbox 1.4.1 User's Manual", Finnish Centre of Excellence in Interdisciplinary Music Research. Dostopno na: https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/MIRtoolbox1.4.1Guide

[10] Wikipedia - Matlab. Dostopno na: http://en.wikipedia.org/wiki/MATLAB

[11] Wikipedia - Onset(audio). Dostopno na: http://en.wikipedia.org/wiki/Onset_(audio)

[12] KernScores - A library of virtual musical scores in the Humdrum **kern data format. Dostopno na: http://kern.ccarh.org/

[13] Wikipedia - Ritem. Dostopno na: http://en.wikipedia.org/wiki/Rhythm

[14] A. Klapuri, "Introduction to Music Transcription". Dostopno na: http://www.cs.tut.fi/sgn/arg/klap/amt-intro-old.pdf