Automatska notna transkripcija muzičkog zviždanja

Vidaković, Mia

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1399

AUTOMATSKA NOTNA TRANSKRIPCIJA MUZIČKOG ZVIŽDANJA

Mia Vidaković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1399

AUTOMATSKA NOTNA TRANSKRIPCIJA MUZIČKOG ZVIŽDANJA

Mia Vidaković

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1399

Pristupnica:	**Mia Vidaković (0036541765)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Davor Petrinović

Zadatak: **Automatska notna transkripcija muzičkog zviždanja**

Opis zadatka:

U sklopu završnog rada potrebno je istražiti poznate postupke računalne analize digitaliziranog audio zapisa muzičke izvedbe u svrhu automatske notne transkripcije glazbe. U slučaju glazbene izvedbe samo jednog muzičkog instrumenta, značajke glazbe koje moraju biti određene u postupku automatske transkripcije su frekvencija i visina tona, trenutci početaka pojedinih nota i njihovo trajanje, dinamika i intenzitet izvedbe, odnosno boja zvuka. Istražiti na koji su način ove značajke reprezentirane i kodirane u MIDI formatu zapisa glazbe u računalnom obliku. Istražiti kako se ovi postupci poopćuju za slučaj polifone glazbe, odnosno istovremene izvedbe više različitih muzičkih instrumenata, poput orkestralnih izvedbi. U praktičnom dijelu završnog rada, istražiti i po mogućnosti realizirati jednostavniju zadaću transkripcije audio snimke zviždanja, gdje se nepoznata melodija izvodi ljudskim zviždanjem. Odabrati pogodne postupke estimacije značajki izvedene melodije iz vremenskih uzoraka snimke zviždanja, tj. odrediti niz izviždanih nota, njihova trajanja i dinamiku izvedbe, te kodirati dobivenu transkripciju u MIDI zapisu. Istražiti i postupke usporedbe dobivene transkripcije žive izvedbe sa zapisima melodija koje želimo automatski prepoznati (klasificirati), na temelju sličnosti notnog zapisa uz odabir odgovarajućih metrika sličnosti i uz potrebna vremenska poravnanja referentnog zapisa i stvarne izvedbe. Za više informacija obratiti se mentoru.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

Uvod

Automatska transkripcija glazbe (engl. *Automatic Music Transcription*, AMT) problem je koji je već dugi niz godina mnogima od velikog interesa. Transkripcija glazbe proces je pretvaranja glazbenog djela iz zvučnog oblika u pisani oblik. Ovo je tehnika koju glazbenici usavršavaju tokom svojeg glazbenog školovanja, no neke skladbe bi rijetko koji čovjek mogao vlastitim uhom i rukom transkribirati. Neupitno je da bi automatizacija ovakvog zadatka bila mnogima od velike koristi.

Transkripcija glazbe jest veoma složen i slojevit problem. Može se odnositi na raspoznavanje osnovnih koncepata poput visine tonova i njihovih trajanja, do složenijih poput mjere, strukture djela, izvođača i podjelu dionica po izvođačima. Sama ideja nije skroz nova te već postoji niz aplikacija koje rješavaju neku varijantu opisanog zadatka, poput Melodyne [1], mobilne aplikacije Chordify [2], Anthem Score [3] i drugi.

U ovom radu izložene su postojeće metode i pristupi opisanom problemu. Cilj praktičnog dijela ovog rada jest implementirati i demonstrirati jednostavnije dijelove aplikacije koja bi transkribirala i klasificirala melodiju proizvedenu ljudskim zviždanjem. Krajnja aplikacija bi radila tako da joj korisnik na ulaz preda snimljeni zvuk neke odzviždane melodije koju ona uspješno interpretira i klasificira u jednu od mogućih melodija iz svoje baze (princip sličan popularnoj aplikaciji *Shazam*). Naziv pjesme iz koje je odzviždana melodija vraća se kao izlaz. Muzičko zviždanje je izabrano zbog svoje svojstvene jednostavnosti, a i dostupnosti svakom pojedincu.

1. Glazbena pozadina

1.1. Glazbene značajke

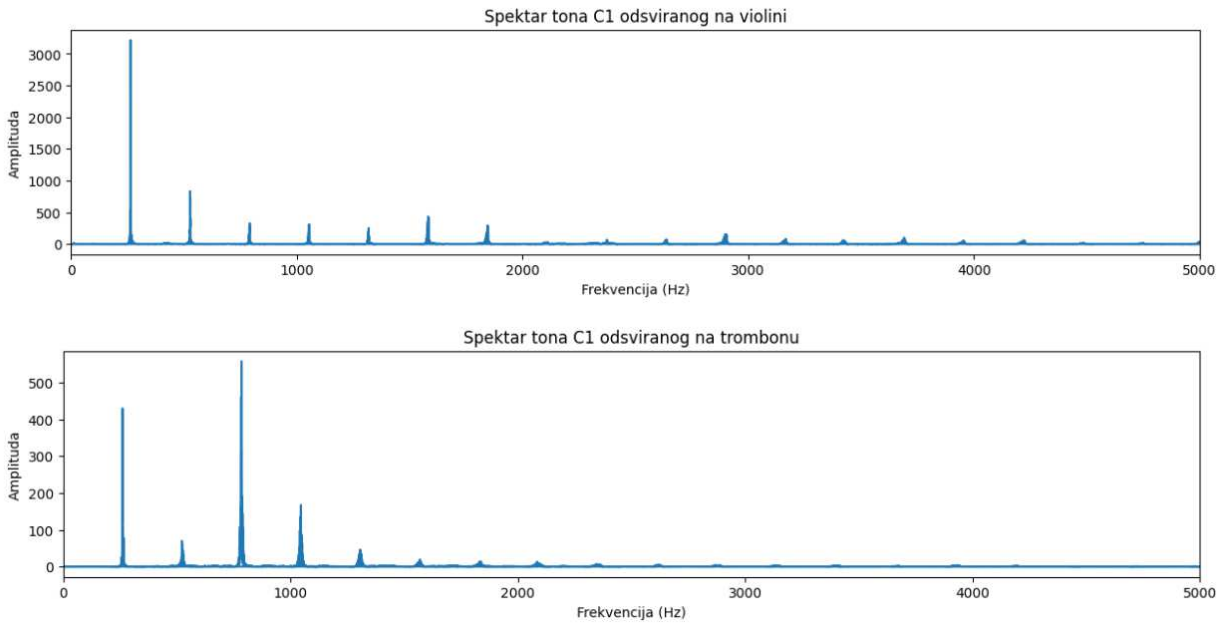
Pri analiziranju nekog glazbenog komada cilj je izdvojiti osnovne značajke koje ga karakteriziraju. Glazba je relativno kompleksan koncept i ima puno razina informacija koje ju opisuju. Osnovne značajke su visina odsviranih tonova, njihov početak i trajanje. Visina, početak i trajanje tonova određuju melodiju promatranog djela odnosno ono po čemu slušatelj pamti skladbu. Osim toga, glazbeno djelo ima i niz drugih značajki koje je potrebno znati pri njegovoj izvedbi. To su značajke poput mjere, tempa, intenziteta i dinamike, harmonije, načina izvođenja te izvođača. Ako je djelo o kojem se radi opera ili nekakva solo pjesma, dodatna značajka mogla bi biti i tekst. Program automatske transkripcije idealno bi prepoznavao sve te značajke u ulazno danom signalu.

1.1.1. Visina tona

Visina tona određena je svojom frekvencijom. Osnovna frekvencija, odnosno ona s kojom slušatelj povezuje ton kojeg čuje, odgovara osnovnom harmoniku i naziva se fundamentalnom frekvencijom (engl. *fundamental frequency*, F0). Osim F0, ton odsviran na većini instrumenata karakterizira i niz drugih frekvencija koje zvuče uz osnovnu. Radi se o alikvotima, odnosno višim harmonicima. To su frekvencije koje odgovaraju višekratnicima F0:

$$f_h = hf_0\,, \quad h > 0 \tag{1}$$

F0 obično je najjačeg intenziteta, a daljnjim harmonicima s visinom frekvencije opada intenzitet pa se oni znatno slabije čuju. Najveći intenzitet F0 nije ključan razlog zašto baš tu frekvenciju najviše čujemo. Nekad je u nekim odsviranim tonovima više zastupljen prvi harmonik od osnovnog, ali zbog komplementarne i pravilne raspodijele ostalih harmonika, svi zajedno zvuče i daju zvuk osnovnog harmonika. Primjer razlike u spektru dva instrumenta dan je na slici (Slika 1.1).

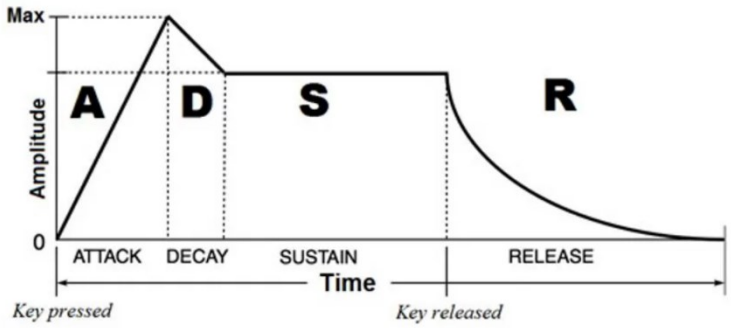


Slika 1.1 Spektar odsviranog tona c1 (261 Hz) na violini i na trombonu

Oktava je interval (razmak između dva tona) od 12 polutonova. Bitno je istaknuti da oktavu karakterizira omjer frekvencija njenih tonova koji je uvijek 2:1. Iz ovog je vidljivo da dok je s jedne strane skala glazbenih tonova linearna, odgovarajuće frekvencije svakog tona su pak u logaritamskom odnosu. Ovu činjenicu je bitno imati na umu pri analizi muzičkih signala.

1.1.2. Nastup tona

Još jedan bitan faktor tona je njegov nastup i trajanje. Razvoj zvučanja tona u vremenskoj domeni karakterizira ADSR ovojnica (engl. Attack Decay Sustain Release) čiji je osnovni oblik prikazan na slici (Slika 1.2) [20]. Prvu fazu, odnosno nastup tona, označava nagli porast energije do njezine maksimalne amplitude. Nakon toga energija se spušta do održive razine na kojoj se stabilizira i ostaje do otpuštanja tona. Nakon otpuštanja tona amplituda s vremenom pada u nulu.



Slika 1.2 osnovni model ADSR ovojnice

Bitno je naglasiti da ima iznimaka i ne proizvodi svaki instrument ovojnicu ovakvog oblika. Na primjer, gudači instrumenti ne proizvode tonove ovakve ovojnice. Zbog načina na koji gudači instrument proizvodi zvuk (potezanje gudala po napetoj žici), moguće je regulirati intenzitet tona tokom njegovog trajanja. Iz tog razloga intenzitet ne mora nužno biti najveći na samom nastupu. To nije slučaj kod instrumenata s tipkama gdje ton nastaje udarom batića u nategnutu žicu. Tako proizvedeni ton maksimalnu amplitudu ima baš na nastupu i nakon pritiska tipke ne može više regulirati njegov intenzitet.

Pri analizi tonova ključan trenutak predstavlja sam nastup kojeg je potrebno detektirati. Pri detekciji koriste se tzv. funkcije detekcije (engl. *detection function*, DF) koje prebacuju signal u domenu u kojoj je lakše promatrati značajke koje bi indicirale pojavu nastupa. Čest pristup jest koristiti DF koja prati amplitudu ovojnice i traži maksimume. Funkcija detekcije može se dobiti i računanjem energije svakog segmenta signala, gdje bi nastupi poprimali lokalne maksimume.

Uočeno je da početak događaja u zvučnim signalima karakterizira i veća zastupljenost visokih frekvencija. Ona se može izmjeriti HFC (engl. *High Frequency Content*) funkcijom koja je jednaka linearnoj kombinaciji frekvencijskih komponenti spektra te je konstruirana tako da višim frekvencijama daje veći značaj [4]. Računa se po izrazu (2) gdje je M broj uzoraka Fourierove transformacije, X(k) odgovara vrijednosti k-tog elementa STFT-a u k-tom trenutku.

$$HFC = \sum_{k=2}^{\frac{M}{2}+1} [|X(k)|^2 * k] \tag{2}$$

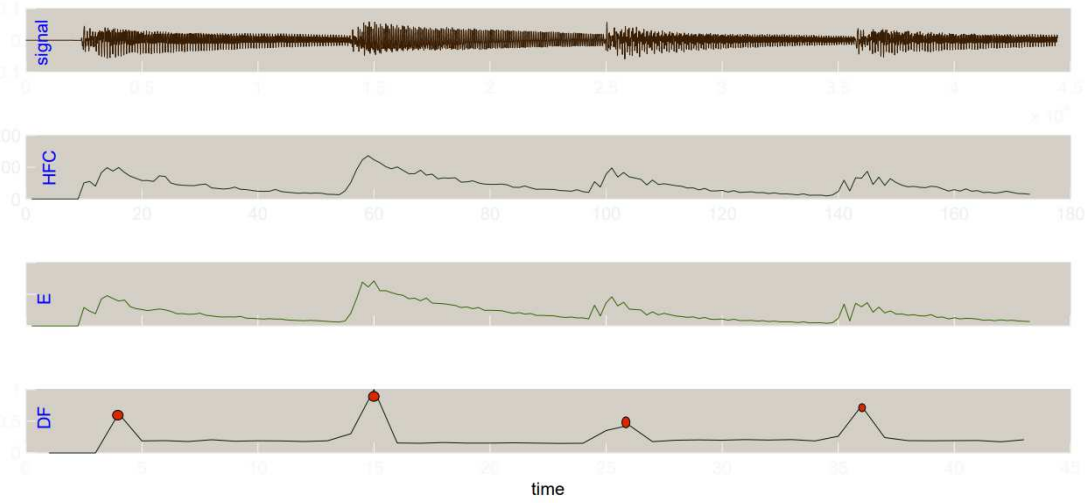
Energija je definirana izrazom (3).

$$E = \sum_{2}^{\frac{M}{2}+1} |X(k)|^2 \tag{3}$$

Kombiniranjem ove dvije funkcije, moguća funkcija detekcije dana je formulom (4) gdje r označava trenutni okvir, a r-1 prethodni okvir.

$$DF_r = \frac{HFC_r}{HFC_{r-1}} * \frac{HFC_r}{E_r} \tag{4}$$

Ovakva funkcija detekcije ima vrhove na mjestima gdje dolazi do naglih promjena u signalu i njegovoj energiji (Slika 1.3). Iz tog razloga je pogodnija za detekciju tonova odsviranih oštrim i naglim nastupom.



Slika 1.3 Funkcija detekcije koja detektira nagle nastupe (DF). Prikazana je i HFC funkcija kao i funkcija energije signala prikazanog na najgornjem grafu.

Nakon postavljanja funkcije detekcije potrebno je u njoj izabrati vrhove koji će reprezentirati nastupe tonova. Za očekivati jest da će funkcija detekcije imati neke lokalne maksimume koji neće odgovarati nastupima tonova pa je potrebno uvesti kriterij koji vrh treba zadovoljiti da bi bio klasificiran nastupom. Najjednostavnije rješenje bilo bi da se postavi prag iznad kojeg se maksimum smatra nastupom. Teško je odrediti prag unaprijed jer on bitno ovisi o kvaliteti snimke, a i o intenzitetu dijela snimke. Kad bi prag bila konstantna vrijednost, lagano bi se moglo dogoditi da se propuste neki nastupi u tišim dijelovima, odnosno da se lažno prepoznaju

nastupi u glasnijim dijelovima snimke. Često korištena metoda jest da se kao prag postavi zaglađena sama funkcija detekcije.

1.1.3. Dinamika i intenzitet

Dinamika određuje glasnoću odsvirane note. Sama glasnoća jest subjektivna mjera i ovisi o osjetljivosti uha no ipak se može objektivno mjeriti pomoću jakosti samog zvuka. Jakost ili intenzitet zvuka jest veličina koja opisuje snagu vala odašiljanu po jedinici površine [6]. Obično se prikazuje u logaritamskoj skali i mjeri se u decibelima (dB). Glasnije note bit će odsvirane snažnije, odnosno većeg intenziteta, dok će tiše note biti slabijeg intenziteta.

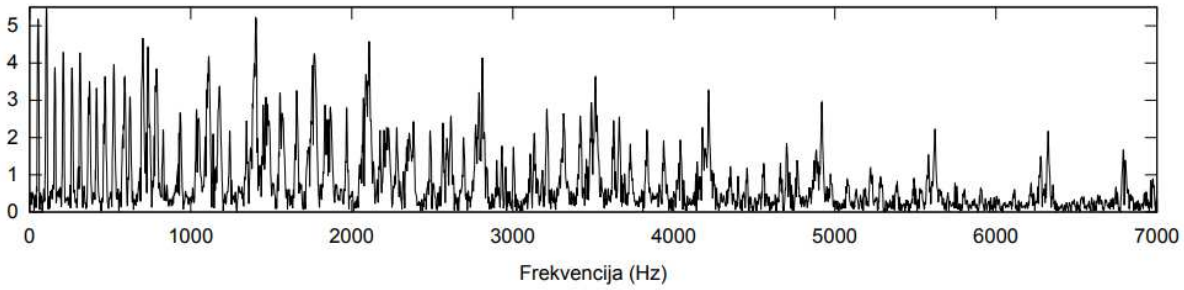
1.1.4. Boja zvuka

Dva tona iste frekvencije i trajanja i dalje mogu zvučati znatno drugačije ako su odsvirani na dva različita instrumenta. To je zbog svojstva boje zvuka (engl. *timbre*) koja je specifična svakom instrumentu. Više faktora utječe na boju zvuka instrumenta no glavni faktori su izgled ovojnice u vremenskoj domeni (opisano u poglavlju 1.1.2) te zastupljenost frekvencijskih komponenata u spektru. Svaki instrument proizvodi zvuk kojeg karakterizira jedinstvena raspodjela intenziteta po odgovarajućim harmonicima. Koliki udio ima koji harmonik u proizvedenom zvuku znatno utječe na njegovu boju. Što su u zvuku bogatiji viši harmonici, to ton zvuči punije i zvučnije. To je jedan od razloga zašto ton odsviran na violončelu zvuči bogatije i harmoničnije od zviždanja koje ima jako siromašan udio viših harmonika.

1.2. Polifona glazba

U glazbi se razlikuju monofonija od polifonije i homofonije. Monofonija jest jednoglasje (ima jednu dionicu), a polifonija i homofonija su višeglasja (imaju više dionica koje istodobno zvuče). Homofonija je specifična po tome što ima jednu glavnu dionicu koju prate ostale dok su u polifoniji sve dionice ravnopravne. Monofone signale je puno lakše analizirati od polifonih. U analizi polifonih signala dolazi do novostvorenih problema uslijed miješanja zvukova svih dionica. Potrebno je na neki način izdvojiti sve dionice i obraditi ih zasebno.

Problemi nastaju kada se spektralne komponente tonova odsviranih u isto vrijeme preklapaju. Sama pojava je veoma česta u glazbi zbog toga što su baš takve kombinacije tonova, zbog svoje harmonijske kompatibilnosti, zvučno ugodne ljudskom uhu. Spektar postaje kompleksna



Slika 1.4 Spektar mješavine tonova

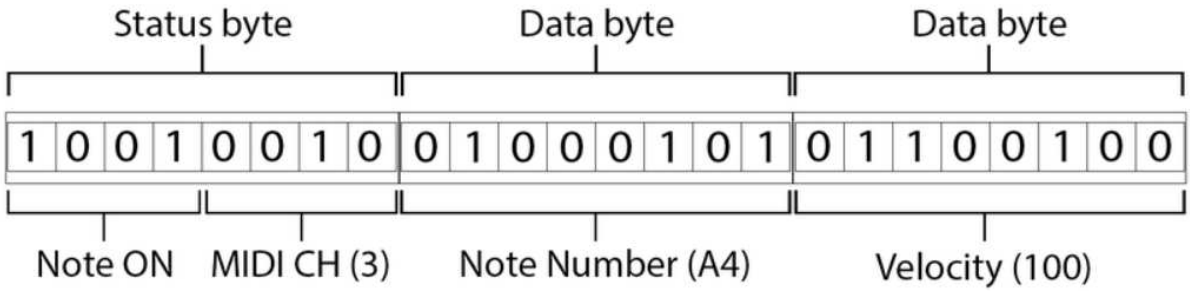
mješavina isprepletenih i stopljenih harmonika svih prisutnih tonova (Slika 1.4). U takvom scenariju više nije trivijalno odrediti koji vrh pripada nečijoj F0, a koji vrh je samo superpozicija viših harmonika.

1.3. MIDI zapis

MIDI (engl. *Musical Instrument Digital Interface*) jest standard koji opisuje protokol kojim se pohranjuje i prenosi glazba. Razlikuje se od audio datoteka ekstenzija ‘.mp3' i '.wav' jer ne sadrži zvuk već samo informacije, odnosno upute o tome kako i u kojem trenutku proizvesti koji zvuk. Razvijen je kako bi bilo lakše prenositi glazbene informacije i kako bi instrumenti mogli međusobno komunicirati. MIDI protokol temelji se na midi porukama. Midi poruke sastoje se od status bajta, koji najavljuje o kakvoj poruci se radi, nakon kojeg slijedi nekoliko podatkovnih bajtova. Bajtovi se razlikuju po tome što status bajt na 7. bitu ima vrijednost 1, dok podatkovni ima vrijednost 0.

Osnovne poruke su NOTE ON i NOTE OFF te one određuju početak i kraj svakog tona. U njihovim podatkovnim bajtovima prenose se informacije o visini i intenzitetu (engl. *velocity*) referenciranog tona [21]. Visina tona kodirana je kao 7-bitni binarni broj te je prikladno reprezentirana cijelim brojem iz raspona od 0 do 127. Srednji ton c1 ima broj 60, a cijeli dostupan raspon tonova proteže se od 4 oktave ispod do 5 oktava iznad njega. Intenzitet tona

opisuje dinamiku kojom je ton odsviran. Isto je reprezentiran 7-bitnim binarnim brojem gdje 0 ne proizvodi zvuk uopće, a 127 proizvodi najglasniji zvuk moguć. Osim informacija o visini i intenzitetu, svaka poruka nosi sa sobom i informaciju o kanalu na koji se odnosi. MIDI ima 16 kanala, odnosno pruža mogućnost istodobnog zvučanja 16 dionica. Osim toga, postoje i *program change* poruke. Te poruke šalju informaciju o instrumentu koji se postavlja na zadani kanal. Instrument je također reprezentiran 7-bitnim binarnim brojem. Sve note poslane nakon te poruke bit će izvedene zadanim instrumentom. Primjer jedne midi poruke dan je na slici (Slika 1.5).



Slika 1.5 NOTE ON midi poruka

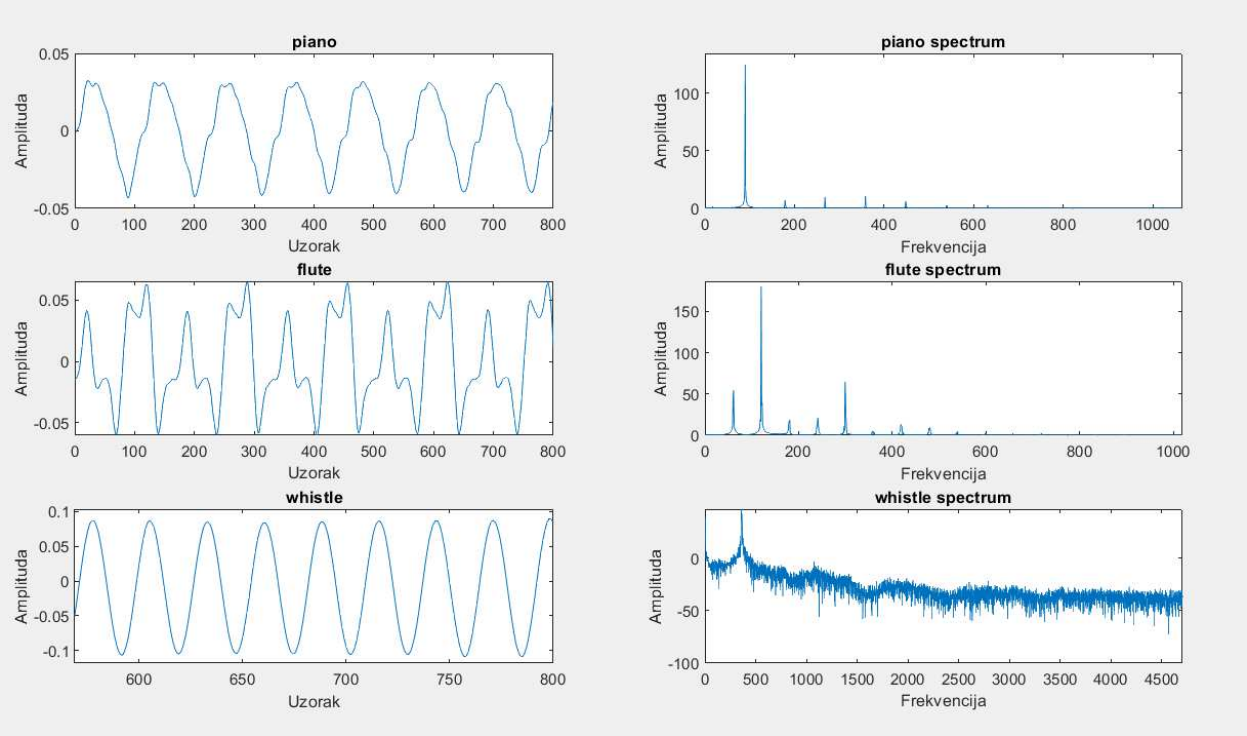
U MIDI zapisu korisnik nema ikakvu kontrolu nad završnom kvalitetom zvuka koji će se proizvesti. Proizvedeni zvuk ovisi o uređaju koji izvodi MIDI. Uređaj pročita upute sadržane u '.mid' datoteci i sam proizvede zvuk u skladu sa svojim mogućnostima.

1.4. Svojstva ljudskog zviždanja

Ljudsko zviždanje jest zvuk proizveden koncentriranim prolaskom zraka kroz uski otvor oblikovan ljudskim usnicama. Frekvencije proizvodive ljudskim zviždanjem tipično se nalaze u rasponu od 500 Hz do 5000 Hz [5] koji se ugrubo proteže kroz 3 oktave počevši s tonom c2 (523.25 Hz) sve do tona c5 (4186.01 Hz). Na raspon zviždanja pojedinca utječe duljina njegovog vokalnog trakta (kog muškaraca 17.5 cm, kod žena 15cm, a kod djece još manja). Pojedinci s dužim vokalnim traktom uglavnom mogu zviždati dublje tonove [7]. Na raspon zviždanja utječe i duljina prednjih zuba; što su kraći to je moguće odzviždati više tonove.

Zviždanje proizvodi zvuk bez gotovo ikakvih alikvota. Zastupljenost i intenzitet alikvota uzrokuje punoću zvuka što je razlog zašto zviždanje zvuči tako jednostavno. U vremenskoj

domeni odgovara idealnoj sinusoidi te očekivano u spektru ima jasan vrh na odgovarajućoj frekvenciji titranja sinusoide (Slika 1.6). Iz tog razloga je zviždanje bitno jednostavnije svojstveno analizirati u usporedbi s drugim instrumentima. Instrumenti poput klavira ili flaute imaju zanimljiviji i puniji sustav harmonika pa se analiza takvih zvukova otežava.



Slika 1.6 Valni oblik i spektar zvuka proizvedenog na klaviru, flauti i zviždanjem

2. Pristupi automatskoj transkripciji

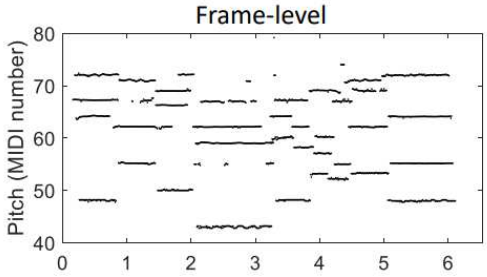
Metode pristupa automatskoj transkripciji ugrubo se dijele u dvije grupe: one koje značajke signala izvlače iz vremenske domene i one koje ih izvlače iz frekvencijske domene. Analiziranje signala u vremenskoj domeni svodi se na praćenje njegovog valnog oblika. Iz njega se mogu dobiti informacije o periodu signala i nastupima tonova. Takva analiza prikladnija je za monofone signale gdje se zna da dani valni oblik pripada samo jednom izvoru. Za polifone signale nekad je informativnije koristiti i metode koje analiziraju signal u frekvencijskoj domeni.

Proces transkripcije može se podijeliti u 4 razine [8]: *Frame-level* (transkripcija razine okvira), *Note-level* (transkripcija razine nota), *Stream-level* (transkripcija razine toka) i *Notation-level* transkripcija (transkripcija razine notacije).

2.1. *Frame-level* transkripcija

Frame-level transkripcija bavi se određivanjem F0 prisutnih u svakom bloku signala. Ako se radi o polifonom djelu, u jednom bloku može biti prisutno više F0 istovremeno, dok je u monofonom prisutna samo jedna. Za monofone signale se uspješnim pokazao algoritam YIN [9]. Njegova modifikacija PYIN (engl. *Probabilistic YIN*) u kombinaciji sa skrivenim Markovljevim modelima (engl. *hidden Markov model*, HMM) je dala nešto uspješnije rezultate [10]. PYIN iz audio signala izvuče note kandidate, odnosno moguće F0 prisutne u trenutku sa svojim vjerojatnostima. Dobiveni podaci se dalje prosljeđuje u HMM koji predviđa najvjerojatniji slijed tonova. Neke od metoda koje rješavaju problem nalaženja F0 u vremenskoj domeni koriste autokorelacijsku funkciju (engl. *Autocorrelation function*, ACF) kojom se može odrediti period signala (detaljnije objašnjeno u poglavlju 3.2.1). Postoji i pristup koji period određuje funkcijom apsolutne razlike između vrijednosti periodičnog signala i njegovih pomaknute verzije (engl. *Average magnitude difference function,* AMDF). Razlika između takva dva signala bit će minimalna kad je pomak upravo jednak višekratniku perioda te se pomoću te informacije može i izračunati sam period. Neki pristupi koriste i kombinaciju ovih dviju metoda [11]. U novije

vrijeme mnogo metoda pristupa ovom zadatku koristeći strojno učenje i neuronske mreže [22]. Neuronske mreže su se pokazale dosta uspješnima no uglavnom samo na podacima na kojima su učene. Veliki nedostatak pristupa temeljenih na neuronskim mrežama jest potreba za velikim i raznolikim skupovima podataka. Ako je mreža trenirana samo na skupu koji sadrži snimke klavira, bit će jako loša u raspoznavanju značajki na snimkama violine. Zadatak izrade takve baze podataka stvara problem sam za sebe. Mogući izlaz iz *frame-level* transkripcije vidi se na slici (Slika 2.1) na kojoj je prikazan niz frekvencija detektiranih u ulaznom signalu.

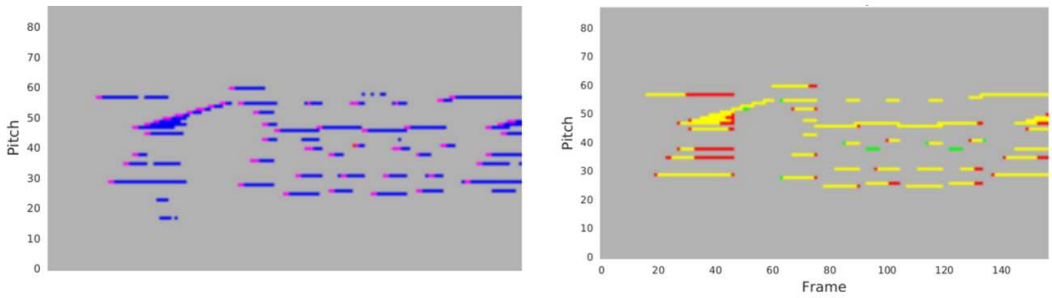


Slika 2.1 Izlaz iz *Frame-level* transkripcije.

2.2. *Note-level* transkripcija

Note-level transkripcija jest podzadatak u kojem se detektiraju početci tonova te se na temelju toga frekvencije dobivene u prethodnom koraku grupiraju u cjeline koje će odgovarati notama. Konkretnije, u ovoj fazi transkripcije određuju se početak i završetak svake note. Razni su pristupiti *Note-level* transkripciji od kojih se jedan oslanja na obrađivanje podataka dobivenih iz *Frame-level* faze. U takvim metodama koriste se tehnike filtriranja pomoću medijan vrijednosti, HMM, i neuronske mreže [12] [25]. Neki pristupi iz audio signala izvlače karakteristike visine, početka i trajanja nota zajedno u istom koraku. Na takvom principu napravljen je model *Onsets and Frames, OaF* (*Google Brain Team)* [13] za automatsku transkripciju klavirske glazbe. OaF koristi dvije neuronske mreže od kojih jedna određuje početke tonova, a druga, na temelju izlaza iz prve, određuje fundamentalne frekvencije prisutne u svakom okviru. Ovakav model ima veću uspješnost jer se neće registrirati nova frekvencija ako se ne detektira početak tona u istom segmentu signala. To nekad radi veliku razliku jer se često zbog lošije kvalitete snimke i šuma detektiraju mnoge frekvencije koje zapravo ne odgovaraju ijednom tonu. Na lijevoj slici (Slika

2.2) plava boja predstavlja tonove registrirane u okvirima, crvena predstavlja registriran početak tona, a ljubičasta predstavlja tonove gdje se izlazi slažu, odnosno tamo gdje je detektirana i F0 i početak tona. Ima nekih tonova za koje nije detektiran početak pa se oni izbacuju i dobivamo izlaz koji je prikazan na desnoj slici. Žuta boja prikazuje registrirane tonove koji se preklapaju s očekivanim izlazom, zelena boja prikazuje registrirane tonove kojih nema u očekivanom izlazu, a crvena prikazuje tonove koji nisu registrirani, a nalaze se u očekivanom izlazu.



Slika 2.2 Detekcija nastupa tonova i frekvencija u okvirima, OaF

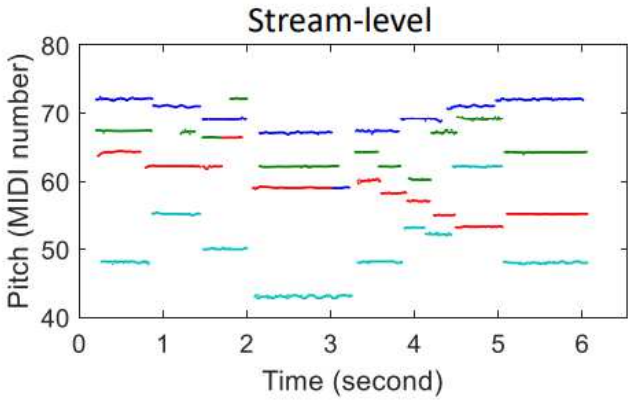
2.3. *Stream-level* transkripcija

Stream-level transkripcija procedura je koja se bavi izdvajanjem glasova, odnosno dionica u polifonim signalima. Može biti slučaj da sve glasove izvodi isti instrument (npr. skladba izvedena na klaviru) ili svaku dionicu izvodi jedan instrument (npr. orkestralna djela). U drugom slučaju to postaje problem separacije izvora zvuka. Ideja je da se praćenjem drugačijih boja instrumenata zvuk podjeli u dionice na kojima se onda odvojeno provodi daljnja analiza. Metode pristupa uglavnom uključuju duboke neuronske mreže ili nenegativnu matričnu faktorizaciju (engl. *Non-negative matrix factorisation*, NMF) [23]. NMF je metoda koja faktorizira početnu matricu podataka V u dvije nenegativne matrice W i H:

$$V \approx WH = V' \tag{5}$$

U ovakvoj obradbi signala matrica V je obično spektrogram, matrica W reprezentira neke značajke signala, a matrica H prikazuje koja značajka je kada aktivna. U ovakvom problemu teži se naći optimalne matrice W i H takve da su razlike između očekivane matrice V i dobivene matrice V' minimalne. Minimiziranjem funkcije pogreške pomoću nagiba gradijenta matrice W i

H se modificiraju tako da bolje opisuju karakteristike signala. Jedan mogući izlaz iz *Stream-level* transkripcije prikazan je na slici (Slika 2.3) gdje je svaka dionica prikazana svojom bojom.



Slika 2.3 Izlaz i *Stream-level* transkripcije

Postoje i neke metode koje iz homofonog djela sastavljenog od više dionica pokušavaju ekstrahirati samo glavnu melodiju [14]. Glavna melodija može se protezati i kroz više dionica tokom izvedbe. Osim glavne melodije zadatak može biti i da se izvuče samo bas, odnosno najdublja dionica.

2.4. Notation-level transkripcija

Notation-level transkripcija zadnji je korak u kojemu se sve prethodno skupljene informacije zapišu u obliku notnog zapisa. Sljedove nota dobivene iz prethodnih koraka transkripcije potrebno je interpretirati u kontekstu glazbenih koncepata. Za ovu fazu transkripcije potrebno je veće znanje same teorije glazbe. Za ispravan i kompletan zapis u notnom crtovlju potrebno je odrediti mjeru, ritam, tonalitet, grupirati note u taktove i sl.

Postoje metode koje ne razdvajaju proces transkripcije u više koraka već koriste *end-to-end* sustav koji na ulaz prima signal i na izlaz vraća kompletni notni zapis [24] [26]. Takvi pristupi uglavnom se temelje na neuronskim mrežama. Druge metode se temelje na analizi MIDI zapisa kojeg pokušavaju pretvoriti u muzičku notaciju [27]. MIDI zapis detaljnije je opisan u poglavlju 1.3, no bitno je napomenuti da on prikazuje samo u kojem trenutku je odsvirana koja nota, a ne i sve ranije opisane glazbene koncepte.

3. Praktični dio

U praktičnom dijelu rada implementirana je jednostavna transkripcija i klasifikacija snimljenog zviždanja. Osnovni zadatak jest odrediti visinu odzviždanih tonova koji se zatim transkribiraju u MIDI zapis. Dobiveni zapis snimljene izvedbe uspoređuje se s nizom melodija prisutnih u bazi podataka i pokušava se točno klasificirati. Melodije u bazi zapisane su u MIDI obliku te su sve sačinjene od jedne dionice. Za konkretnu implementaciju korišten je programski jezik Matlab zbog širokog spektra funkcionalnosti koje pruža u području obrade audio signala.

S obzirom da klasifikacija samog izvora zvuka nije dio kojim se ovaj rad bavi, domena analize je sužena pretpostavkom da su sve ulazne melodije proizvedene ljudskim zviždanjem. Izabrano je zviždanje zbog svojih karakteristično jednostavnih svojstava u usporedbi s drugim instrumentima. Više o svojstvima ljudskog zviždanja opisano je u poglavlju 1.4.

Opisani zadatak može se podijeliti u 4 veća koraka: pred-procesiranje, detekcija F0, post-procesiranje i klasifikacija. Koraci pred-procesiranja i post-procesiranja su prisutni kako bi se dobiveni podaci očistili od mogućih nečistoća i pripremili za sljedeće korake.

3.1. Pred-procesiranje

Prije glavnog zadatka ekstrakcije značajki iz signala, signal je potrebno urediti. Najprije su svi kanali snimke svedeni na jedan. Izlazni kanal Ch odgovara sredini između lijevog i desnog kanala, odnosno izrazu (6) gdje je R desni kanal, a L lijevi kanal.

$$Ch = \frac{R + L}{2} \tag{6}$$

Ovaj korak je suvišan ako snimka ima samo jedan kanal. Pretpostavljeno je da signal koji se dobije na ulazu nije snimljen u idealnim uvjetima stoga on sa sobom osim informacije nosi i niz raznih smetnji i šumova koje je potrebno odstraniti. Filtracija omogućava da se izbace neželjene komponente signala, a ovdje će se pobrinut i za uklanjanje istosmjerne komponente.

3.1.1. Filtracija

Za filtraciju korišten je pojasno propusni IIR filtar 6. reda. Filtar je projektiran korištenjem Chebyshev II aproksimacije koja osigurava jednoliku valovitost u području gušenja i glatku karakteristiku u području propuštanja. Izabran je IIR filtar pored FIR filtra zbog manje složenosti računanja. Iako IIR filtar uvodi fazna izobličenja u signal zbog nelinearnosti faze, to ovdje nije od bitnog značaja jer ljusko uho ne percipira fazu samu za sebe. Interpretativna je samo razlika u fazi što ljudima omogućuje da detektiraju smjer iz kojeg zvuk dolazi. Filtar je implementiran kao kaskada odgovarajućeg visokopropusnog i niskopropusnog filtra. U Matlabu je korištena funkcija `cheby2` kojoj se kao argumenti predaju granične frekvencije područja gušenja. Izabrane su frekvencije od 350Hz za gornju granicu gušenja visokopropusnog filtra i 5000Hz za donju granicu gušenja niskopropusnog filtra. Ovako će područje propuštanja taman zahvatiti raspon frekvencija prikladan pretpostavljenom ljudskom rasponu zviždanja. Gušenje filtra postavljeno je na 40dB. Funkcija `cheby2` vraća koeficijente brojnika i nazivnika prijenosne funkcije filtra. Kaskada filtra odgovara umnošku prijenosnih funkcija filtra, a s obzirom da množenje polinoma odgovara konvoluciji njihovih koeficijenata, do brojnika i nazivnika kaskade može se doći konvolucijom.

```
fd =350;
fg = 5000;
ORD=6;
R=40;
[bh,ah]=cheby2(ORD,R,fd/(fs/2),'high');
[bl,al]=cheby2(ORD,R,fg/(fs/2));
b=conv(bh,bl);
a=conv(ah,al);
y2=filter(b,a,y1);
```

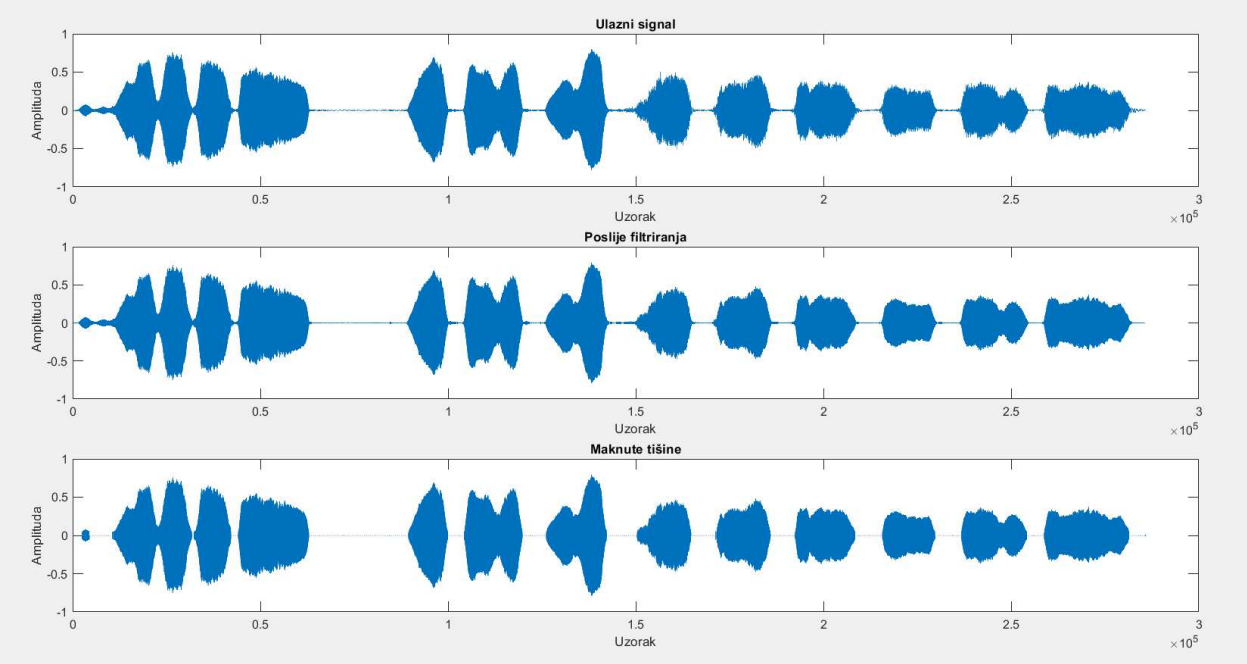
Kod 1.1. – Pojasno propusno filtriranje signala sadržanog u y1

3.1.2. Identificiranje tišina

Kako dijelovi signala u kojima nema bitnih događaja ne bi remetili daljnju analizu, takvi blokovi su prozvani tišinama i njihova vrijednosti je postavljena na 0. Kao prag zvučnih dijelova

postavljena je vrijednost za 50% veća od maksimalne amplitude prvih 50 ms signala. Okvirom je pređeno po cijelom signalu i oni blokovi u kojima maksimalna amplituda nije veća od postavljenog praga, postavljeni su u 0.

Primjer očišćenog signala navedenim metodama prikazan je na slici (Slika 3.1).



Slika 3.1 Signal koji sadrži zvuk zviždanja melodije iz pjesme „Here comes the sun“. Na drugom i trećem grafu prikazan je signal nakon filtracije, odnosno nakon čišćenja tišina.

3.1.3. Postavljanje parametara

Potrebno je postaviti parametre na koje će se oslanjati daljnja obrada. Pretpostavljeno je da se ljudski raspon zviždanja proteže od tona c2 (523.25 Hz) do tona c5 (4186.01 Hz) (poglavlje 1.4). Time su određene maksimalna i minimalna moguća frekvencija f_{max}, odnosno f_{min}. Iz toga se direktno može odrediti minimalan očekivani period formulom (7) gdje je f_s označava frekvenciju otipkavanja.

$$T_{min} = \frac{f_s}{f_{max}} \tag{7}$$

Na jednak način izračuna se i maksimalan očekivani period. Pri obradi promjenjivih signala praktičnije je analizu provoditi blok po blok. Cilj je u svakom bloku uhvatiti što frekvencijski stabilniji djelić signala kako bi se jednoznačno mogao odrediti period tog trenutka. Za određivanje samog perioda, potrebno je imati barem 2-3 njegova ponavljanja u bloku. Iz tog razloga je potrebno pametno odrediti duljinu okvira koji određuje blok. Problem s duljinom okvira jest taj što ako je premala i zahvati se dio signala s dužim periodom, zbog nedovoljnog ponavljanja perioda on se neće moći odrediti. S druge strane, ako je duljina okvira prevelika, riskira se hvatanje djelića signala u kojem dolazi do promjene frekvencije i niti tada se neće jednoznačno moći odrediti period. Iz tog razloga je napravljen kompromis i duljina okvira postavljena je na $2 * T_{max}$. Tako će sigurno uhvatiti barem dva perioda signala u svakom bloku. U svakoj iteraciji okvir se pomiče na sljedeći blok signala na kojem nastavlja analizu. Korak pomaka okvira k postavljen je na 50% od originalne duljine okvira čime se osigurava da svaki uzorak bude obrađen dva puta. Postavljeni parametri prikazani su u tablici (Tablica 1).

Tablica 1 Parametri obrade

f_{max}	f_{min}	T_{max}	T_{min}	$duljina\_okvira$	k
4186.01Hz	523.25Hz	$\frac{f_s}{f_{min}}$	$\frac{f_s}{f_{max}}$	$2 * T_{max}$	$\frac{duljina\_okvira}{2}$

3.2. Detekcija F0

Korištena su dva pristupa detekciji F0. U prvom pristupu korištena je metoda upotrebe autokorelacijske funkcije u vremenskoj domeni, a u drugom je F0 određena jednostavnim nalaženjem maksimuma u spektru svakog bloka.

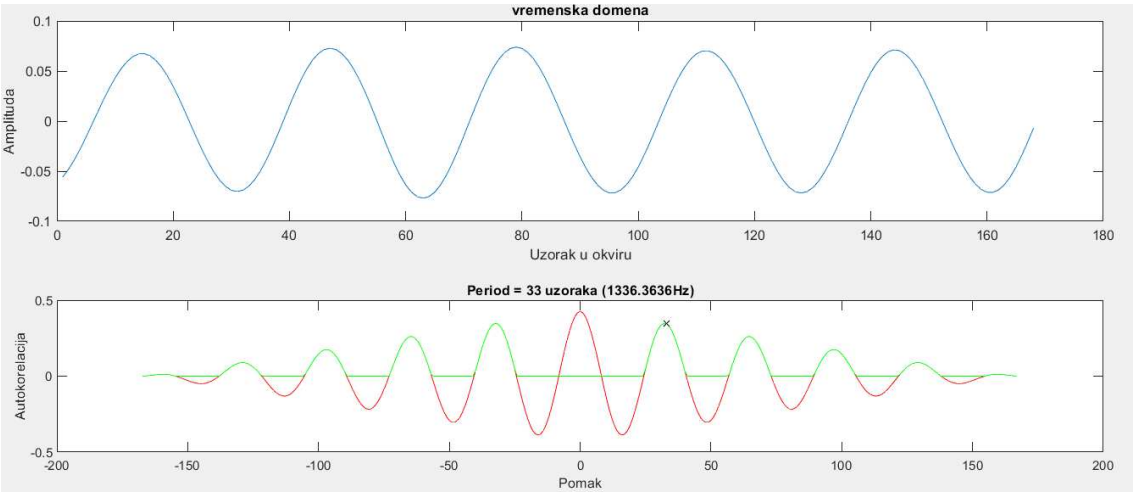
3.2.1. Autokorelacijska funkcija u vremenskoj domeni (ACF)

Korelacija jest operacija koja mjeri sličnost dva signala. Pri računanju korelacije signala x i y, svaki indeks rezultata dobije se kao umnožak signala y i pomaknutog signala x:

$$r_{xy}[n] = x[n] \star y[n] = \sum_{m \in \mathbb{Z}} x[m+n] y^*[m] \tag{8}$$

Autokorelacija jest ista operacija samo što se signal množi s pomaknutom verzijom samoga sebe. Konačan periodičan signal perioda P najviše će korelirati sam sa sobom kad je pomaknut za višekratnik perioda P. Autokorelacijska funkcija stoga ima lokalni maksimum na pomacima jednakim $k * P,\ k \in \mathbb{N}$. Globalni maksimum nalazi se na poziciji gdje je pomak jednak nuli jer je signal najsličniji nepomaknutoj verziji sebe. Računanjem autokorelacije svakog bloka signala može se doći do njegovog odgovarajućeg perioda. Recipročna vrijednost nađenog perioda odgovara traženoj F0.

Dio autokorelacijske funkcije koji zahvaća globalni maksimum postavi se na minimalnu vrijednosti kako bi se našao prvi sljedeći globalni maksimum. Period se zatim računa kao razlika udaljenosti između središnjeg uzorka autokorelacije i novonađenog maksimuma (Kod 1.2). Primjer autokorelacije za jedan segment signala dan je na slici (Slika 3.2).



Slika 3.2 Autokorelacija jednog bloka signala. Na donjem grafu x prikazuje nađeni maksimum.

```
slijed_frekvencija = zeros(1, broj_okvira);
for k = 1 : broj_okvira-1
    poc = (k-1)*korak + 1;
    kraj = poc -1 + frame_len;
    range = poc:kraj;
    frame = ip(range);

    [rxx, lag] = xcorr(frame, frame);
    rxx(rxx < 0) = 0;
    center_peak_width = find(rxx(frame_len:end) == 0 ,1);
    if isempty(center_peak_width) || center_peak_width >= frame_len
        continue;
    end

    rxx(frame_len-center_peak_width : frame_len+center_peak_width) = min(rxx);

    [max_val, loc] = max(rxx);
    period = abs(loc - frame_len+1);

    f0 = fs/period;
    if(f0<f_min)
        continue
    end
    slijed_frekvencija(k)=f0;
end
```

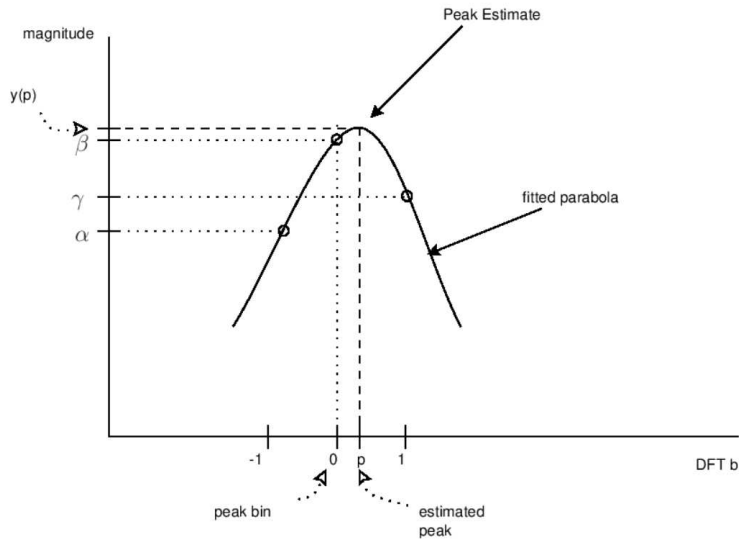
Kod 1.2 – Programski kod koji svakom bloku signala pridjeljuje frekvenciju nađenu autokorelacijskom funkcijom. Osnovni predložak koda preuzet s [18].

3.2.2. Frekvencijska domena

Ekstrahiranje F0 u frekvencijskoj domeni svodi se na nalaženje frekvencije s najvećom energijom, odnosno one za koji spektar poprima globalni maksimum. Osnovna ideja jest dobiti spektar svakog okvira Fourierovom transformacijom i pronaći frekvenciju za koji poprima maksimum. Kako bi se što preciznije mogla odrediti tražena frekvencija koristi se spektralna interpolacija implementirana dodavanjem nula na kraj signala u vremenskoj domeni [15]. Produljivanjem signala nulama u vremenskoj domeni, zgušćuju se uzorci u frekvencijskoj domeni. Kad bi se signal u vremenskoj domeni proširio s beskonačno nula i sam postao beskonačan i aperiodičan, u frekvencijskoj domeni bi razmaci između uzoraka postali beskonačno mali i sam spektar bi postao kontinuiran. Na računalu se ne može dobiti kontinuirani spektar ali može se barem povećati sama rezolucija spektra dodavanjem konačnog broja nula. Ako je signal duljine N, te faktor interpolacije F, na kraj signala pomnoženog Hammingovim

otvorom dodaje se $(F-1)*N$ nula. Dobiva se signal duljine $F*N$ od kojeg se računa FFT u isti broj točaka.

S druge strane, izabrani maksimumi uglavnom neće odgovarati maksimumima stvarnog kontinuiranog spektra. Budući da je na raspolaganju samo diskretni spektar, indeks nađenog maksimuma bit će onaj čija se vrijednost našla najbliže pravom maksimumu. U logaritamskoj domeni vrh spektra ima konturu parabole. Ta činjenica omogućava nalaženje pravog maksimuma provlačenjem parabole kroz 3 uzorka najbliža diskretnom maksimumu (Slika 3.3).



Slika 3.3 Spektralna interpolacija

Pomak p od indeksa koji odgovara diskretnom maksimumu može se izračunati izrazom (9) gdje je β vrijednosti uzorka za koji spektar poprima maksimum, a α i γ su vrijednosti prvog lijevog i desnog uzorka od njega.

$$p = \frac{\alpha - \gamma}{2 * (\alpha - 2\beta + \gamma)} \tag{9}$$

Sada indeks maksimuma odgovara vrijednosti $x + p$, a odgovarajuća frekvencija u Hz izračuna se formulom (10) gdje je i nađeni indeks maksimuma, F faktor interpolacije, N duljina bloka signala kojeg analiziramo i f_s frekvencija uzorkovanja.

$$f = \frac{i-1}{F*N} * f_s \tag{10}$$

S obzirom da je ulazni signal realan, njegov spektar biti će simetričan pa se pri analizi gleda samo njegova prva polovica (Kod 1.3).

```
frekv = zeros(1, broj_okvira);
for k = 1:broj_okvira
    poc = (k-1)*korak + 1;
    kraj = poc -1 + duljina_okvira;
    range = poc:kraj;
    x = ip(range);

    fi = 8;
    xh = x.*hamming(duljina_okvira);
    xph = [xh' zeros(1, (fi-1)*duljina_okvira)];
    sp = fft(xph, fi*duljina_okvira);
    sp_db = 20*log10(abs(sp(1:duljina_okvira/2)));

    [max_y, max_x] = max(sp_db);

    %PARABOLA
    if (max_x == 1)
        indeks_vrha = max_x;
    else
        alfa = sp_db(max_x-1);
        beta = max_y;
        gama = sp_db(max_x+1);

        p = (1/2) * ((alfa - gama)/(alfa - 2*beta + gama));
        indeks_vrha = max_x + p;
    end
    f0 = (indeks_vrha-1) * fs/(fi*duljina_okvira);
    frekv(k) = f0;
end
```

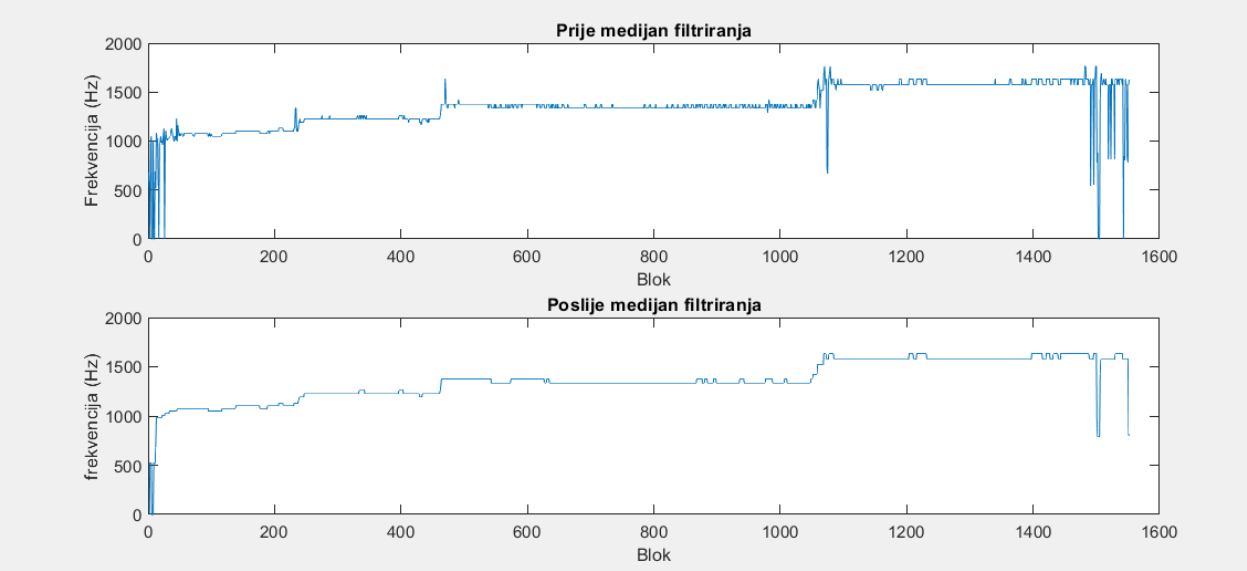
Kod 1.3 – Programski kod koji spektralnom interpolacijom pronalazi frekvenciju svakog bloka signala

3.3. Post-procesiranje

Kao izlaz iz prethodnog koraka dobiven je vektor frekvencija čiji svaki indeks predstavlja F0 odgovarajućeg bloka u signalu. Očekivano je da će dobiveni vektor imati neke stršeće vrijednosti u sebi koje bi trebalo odstraniti. Primjenjuje se medijan filter kao zaglađivačka funkcija. Medijan filter prolazi po manjim segmentima predanog signala sačinjenih od neparnog broja uzoraka te srednjem uzorku svakog segmenta pridijeli medijan cijele grupe. Red filtra određen je brojem

uzoraka grupe koju zahvaća. Za potrebnu operaciju korištena je Matlabova gotova funkcija `medfilt1`. Korištena su dva medijan filtera, prvi 5. reda te nakon njega još jedan 7. reda. Jedan primjer učinka medijan filtra dan je na slici (Slika 3.4).

```
% u frekv su spremljene frekvencije svakog bloka
frekv = medfilt1(frekv, 5);
frekv = medfilt1(frekv, 7);
```



Slika 3.4 Frekvencija svakog bloka signala prije i poslije medijan filtriranja

3.3.1. Pretvorba u MIDI

Dobivene informacije dalje se pretvaraju u MIDI zapis. Najprije je potrebno kvantizirati frekvencije prema skali MIDI tonova.

Jedna oktava se po standardnoj zapadnjačkoj raspodijeli danas dijeli na 12 tonova jednakih razmaka. Frekvencije intervala oktave u odnosu su 2:1 iz čega slijedi da je odnos frekvencija svaka dva tona jednak $1:\sqrt[12]{2}$. Uzevši ton a1 kao referencu, sve više frekvencije mogu se izračunati formulom (11) gdje je f_{a1} frekvencija tona a1 (440Hz), a i odgovara tonski kvantiziranoj udaljenosti traženog tona od referentnog tona a1.

$$f = f_{a1} * \left(\sqrt[12]{2}\right)^{i} \tag{11}$$

Preokretanjem ovog izraza dobiva se izraz (12).

$$i = 12 \log_2 \frac{f}{f_{a1}} \tag{12}$$

Ako tražena frekvencija odgovara tonu a1, dobiveni izraz će za i vratiti 0. Ako je tražena frekvencija jedan ton iznad a1, izraz će vratiti 1. Potrebno je pomaknuti izlaz kako bi se preslikavao u prave vrijednosti visina u MIDI-u. U MIDI zapisu noti a1 pridijeljen je broj 69 pa se u navedeni izraz dodaje još i taj pribrojnik [17]. Završna formula dana je izrazom (13) gdje m predstavlja visinu tona u MIDI kontekstu za željenu frekvenciju f.

$$m = \left\lfloor 12 \log_2 \frac{f}{440} + 69 \right\rfloor \tag{13}$$

Implementirana je jednostavna funkcija koja niz predanih frekvencija pretvara u slijed odgovarajućih MIDI tonova (Kod 1.4).

```
function midi_frekv = to_midi_notes(frekv)
    midi_frekv = zeros(1, size(frekv,2));
    for i=1:size(frekv, 2)
        if(frekv(i) == 0)
            continue
        end
        midi_nota = 12*log2(frekv(i)/440) + 69;
        midi_frekv(i) = round(midi_nota);
    end
end
```

Kod 1.4. – Funkcija koja svaku frekvenciju iz predanog vektora frekvencija preslikava u odgovarajući MIDI ton

3.4. Klasifikacija

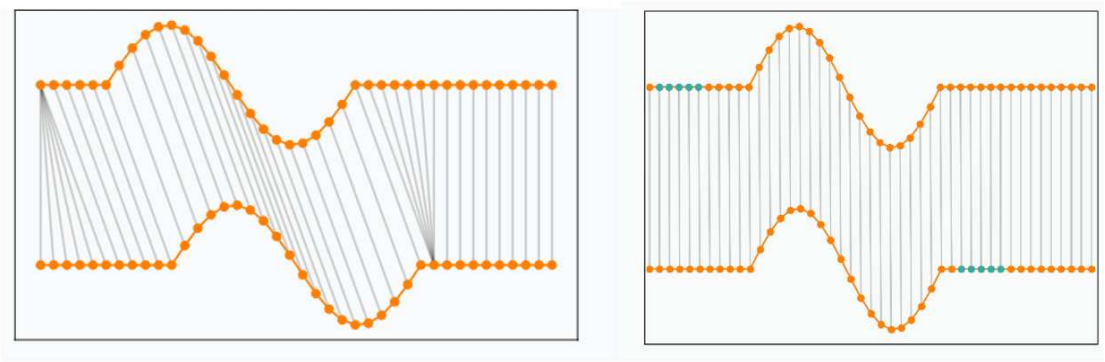
Cilj ove obrade jest prepoznati koju melodiju je korisnik odzviždao na ulazu. U bazi podataka nalazi se niz melodija kandidata s kojima se korisnikova melodija uspoređuje.

Nadobudno je za očekivati da će svaki korisnik melodiju odzviždati jednake brzine kao što je ona u bazi zapisana. Bilo bi pogodnije kad faktor prebrzog, presporog ili promjenjivog izvođenja ne bi utjecao na samu klasifikaciju. Također bi se trebala uzeti u obzir i mogućnost da korisnik nije pogodio apsolutno točnu intonaciju pjesme. Osim ako se ne radi o osobi s apsolutnim sluhom, vjerojatno je da će korisnik melodiju pjevati malo više ili niže od originalne.

Pri uspoređivanju melodija korišteno je dinamičko vremensko poravnavanje (engl. *dynamic time warping*, DTW). DTW osigurava pronalazak najboljeg načina na koji se dva signala mogu poravnati te kao izlaz vraća cijenu poravnanja koja se računa kao suma razlika između vrijednosti poravnatih uzoraka. Kad bi melodije koje uspoređujemo bile zapisane preko frekvencija izraženih u Hz, zbog logaritamskog odnosa glazbenih tonova bi udaljenost vrijednosti uzoraka ovisila o visini samih tonova. U višoj lagi bi udaljenost frekvencija nekog intervala bila puno veća od udaljenosti istog intervala izvedenog u nižoj lagi. Kako to ne bi utjecalo na proces usporedbe, sve su melodije reprezentirane linearnom MIDI skalom. Za lakše rukovanje MIDI datotekama u Matlabu, korišten je skup implementiranih funkcija [16].

3.4.1. Dinamičko vremensko poravnavanje (DTW)

Dinamičko vremensko poravnavanje mjeri sličnost dva signala uz dopušteno rastezanje, odnosno skupljanje dijela signala da bi se ostatak uzoraka optimalno poravnao s drugim signalom (Slika 3.5) [19]. Optimalni put poravnanja koji se traži jest onaj koji ima minimalnu euklidsku udaljenost između vrijednosti poravnatih uzoraka.



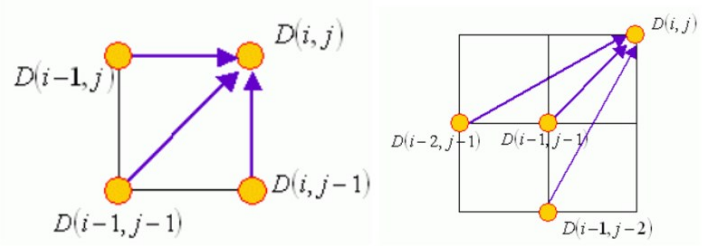
Slika 3.5 Dinamičko vremensko poravnavanje dva signala

Operacija se obavlja uz pomoć matrice cijene (engl. *cost matrix*) dimenzija (n+1) x (m+1), gdje su n i m duljine prvog, odnosno drugog signala. Matrica predstavlja sve moguće puteve kojim se signali mogu poravnati. S dužim signalima broj takvih puteva postaje već jako veliki broj te ispitivanje svih mogućih kombinacija puteva da bi se našao minimum nije opcija. Elegantnije rješenje problema nađeno je upotrebom dinamičkog programiranja. Dinamičko programiranje je način rješavanja problema gdje se pri rješavanju osnovnog problema koriste rješenja više manjih potproblema. U matrici cijene svako polje predstavlja cijenu puta do tog polja pa umjesto da se svaki put računa cijena od samog početka, izračun će se oslanjati na već izračunate cijene mogućih polja prethodnika trenutnog polja. Početno se sva polja matrice postave na beskonačnost, a polje $D_{0,0}$ postavi se na 0. Zatim se prolazi po svim recima i stupcima matrice, počevši od indeksa 1 na dalje, i računa se cijena svakog polja po formuli (14) gdje je $d(x_i, y_j)$ funkcija kojom je definirana udaljenost uzoraka signala (npr. Euklidska udaljenost uzorka x_i i y_j).

$$D_{ij} = d(x_i, y_j) + min \begin{cases} D_{i-1,j-1} \\ D_{i-1,j} \\ D_{i,j-1} \end{cases} \tag{14}$$

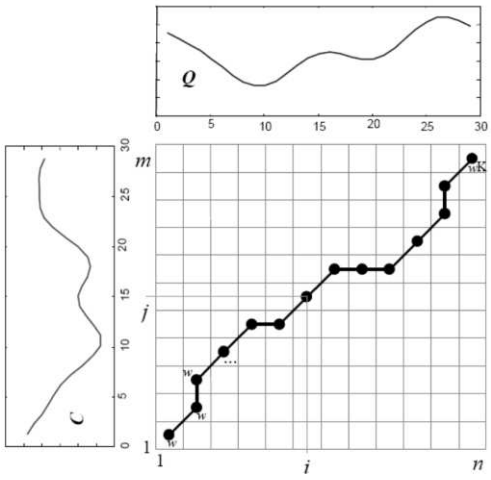
Pri računanju cijene svakog polja matrice nužno je zapamtiti koje od polja mogućih prethodnika $D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}$ je izabran kao minimum u tom koraku kako bi se zapamtio put kojim se išlo. Bitno je napomenuti da kandidati prethodnici između kojih se bira minimum mogu biti i

neka druga polja. Na slici (Slika 3.6) desno prikazan je drugačiji odabir polja za kandidate prethodnike, dok je na lijevoj slici prikazan opisani slučaj.



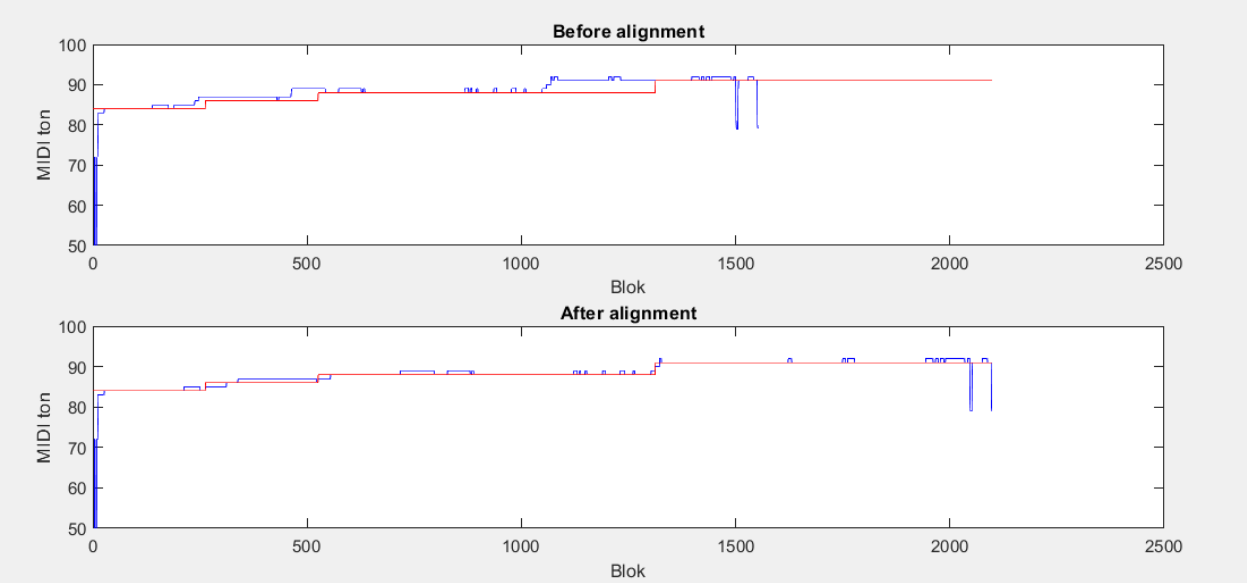
Slika 3.6 Vizualizacija polja prethodnika koji utječu na izračun cijene trenutnog polja $D(i,j)$

Nakon što se ispuni matrica, krenuvši s krajnjim poljem $D_{n,m}$, dobije se optimalni put minimalne cijene prateći prethodnike svakog polja (Slika 3.7). Taj put govori točno koji uzorak prvog signala najbolje odgovara kojem uzorku drugog signala.



Slika 3.7 Optimalni put poravnanja dva signala C i Q nađen pomoću matrice cijene

Nakraju algoritma je u polju $D_{n,m}$ sadržana ukupna cijena cijelog puta. Ta vrijednosti može se koristiti kao kriterij usporedbe kojim se određuje koja melodija iz baze najbolje odgovara korisnikovoj melodiji. Nekad se može dogoditi da je ukupna cijena puta manja samo zbog činjenice da je ulazni signal kraći. Iz tog razloga uspoređivane cijene su normalizirane zbrojem duljina oba signala. Primjer usporedbe i poravnanja tonova dobivenih iz nekog signala s tonovima jedne melodije iz baze prikazan je na slici (Slika 3.8).



Slika 3.8 Primjer dva signala prije (gornji graf) i poslije (donji graf) optimalnog poravnanja.

3.4.2. Transponiranje

Kako bi se uzela u obzir mogućnost da korisnik nije pogodio intonaciju melodije u bazi, uvodi se opcija transponiranja. Za svaku melodiju u bazi, osim direktnog uspoređivanja, odzviždana melodija dodatno se uspoređivala i s transponiranim verzijama svake melodije u bazi za do 5 polutonova iznad i ispod originalnog tonaliteta.

```
function compareData(midi_odzvizdani)
folderPath = './midi_data';
midiFiles = dir(fullfile(folderPath, '*.mid'));

min_ds = Inf;
min_file = midiFiles(1).name;
transp = 0;
for k = 1:length(midiFiles)
    midiFile = fullfile(folderPath, midiFiles(k).name);
    midi_orig = load_and_prepare_midi(midiFile);

    for j = -5:5
        midi_orig_transp = transpose_midi(midi_orig, j);
        ds = dynammic_time_warping(midi_odzvizdani, midi_orig_transp);
        if(ds < min_ds)
            min_ds = ds;
            min_file = midiFiles(k).name;
            transp = j;
        end
    end
end
```

Kod 1.5 – Funkcija koja uspoređuje predani vektor tonova sa svim melodijama iz baze i njihovim transponiranim verzijama

3.5. Usporedba rezultata

Za potrebe testiranja snimljeno je zviždanje dviju osoba. Unaprijed je izabrano 8 melodija od kojih je svaka snimljena 2 ili 3 puta; svaki put u drugačijem tonalitetu. Melodije su bile relativno kratke; u prosjeku su trajale oko 7 sekundi. Najkraća melodija trajala je 3 sekunde, dok je najduža trajala 17 sekundi. Melodije su snimljene na jedan kanal (mono konfiguracija) računalno ugrađenim mikrofonom u relativno tihim uvjetima. Korištena je frekvencija otipkavanja od 44100 Hz. U bazi melodija unaprijed su napravljeni MIDI zapisi svake od melodija. Pri snimanju nekih melodija težilo se promijeniti tempo izvođenja cijele melodije kao i samih dijelova u odnosu na melodiju iz baze. Svrha promjenjivog izvođenja jest vidjeti kako će se DTW algoritam prilagoditi.

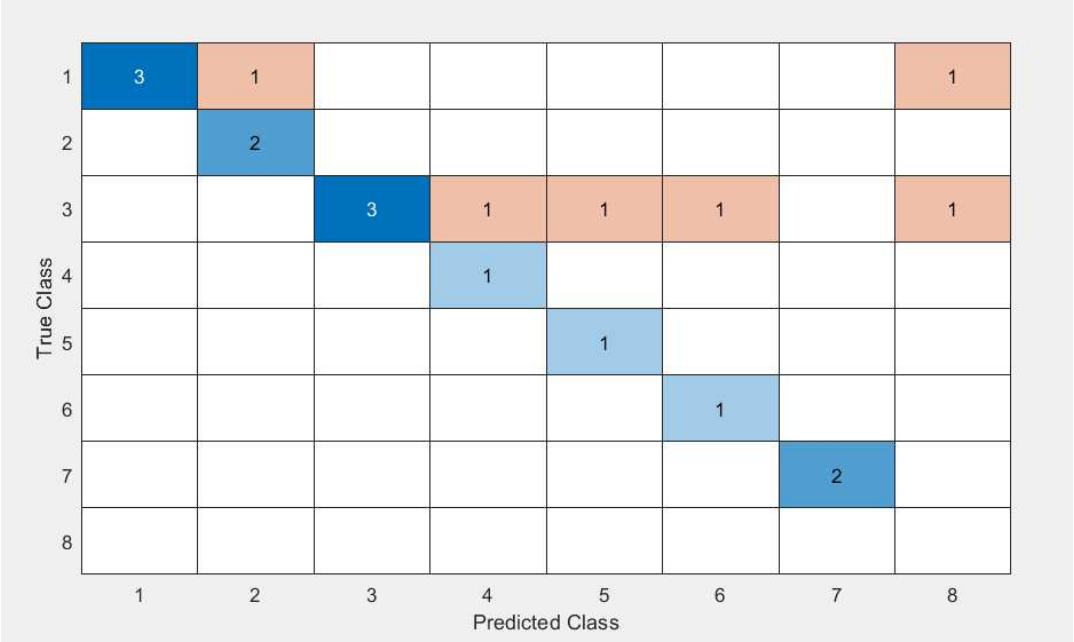
Odzviždana melodija uspoređivana je sa svim melodijama u bazi. Klasificirana je onom melodijom s čijom je usporedbom dobivena najmanja cijena DTW algoritmom.

Za ekstrakciju F0 korištene su dvije metode: ACF u vremenskoj domeni i metoda nalaženja maksimuma spektra. Dobiveni rezultati prikazani su u tablici (Tablica 2).

Tablica 2 Prikaz rezultata

Metoda	**ACF**	**Maksimum spektra**
Točnost	68.42%	68.42%

Obje metode su od 19 snimaka uspješno klasificirali istih 13. Prikazana je matrica konfuzije (Slika 3.9) dobivenih podataka (dobivena je ista matrica za obje metode) kojoj stupci odgovaraju klasificiranim melodijama, a retci svim očekivanim melodijama. Elementi te matrice prikazuju koliko puta je koja snimka prepoznata kao očekivani zapis. Dijagonala matrice predstavlja brojnost uspješno klasificiranih melodija (plava boja).



Slika 3.9 Matrica konfuzije ispitnih podataka

Iz navedenih rezultata se vidi da na uspješnost klasifikacije nije utjecao odabir metode. Usporedba je rađena na relativno malom skupu podatka koji je vjerojatno premali da bi se vidjele značajnije razlike korištenih metoda. Pri analizi nisu uočene veće razlike u izračunatim cijenama poravnanja signala među metodama. Jedna je imala manju cijenu u jednom primjeru dok je druga imala manju cijenu u drugom te nije uočen uzorak među snimkama koji bi to objasnio.

Najveći problem predstavljala je sama priroda zviždanja. U većini snimljenih primjera slijedno odzviždani tonovi uglavnom nisu bili povezani već je između njih postojala barem mala praznina bez tona. Tonovi melodija u bazi bili su uglavnom slijedno povezani osim ako se nije radilo o prazninama koje se tamo nalaze i u originalnoj pjesmi. To je predstavljalo problem DTW algoritmu jer bi u tim nepostojanih prazninama razlika vrijednosti uzoraka bila jako velika i znatno bi povećavala ukupnu cijenu poravnanja (iako se radilo o usporedbi s točnom melodijom). Iz tog razloga bi se nekad dogodilo da neke druge melodije iz baze, koje imaju u sebi više praznina, imaju manju cijenu poravnanja s tom melodijom jer je bitnije bilo da se praznine poravnaju nego sami sljedovi tonova.

Još jedan problem predstavljale su manire koje bi zviždači uveli u svoje izvedbe; poput klizanja sa tona na ton i dekoriranje tona vibratom ili tremolom. Te pojave uvele bi nepostojeće frekvencije kojih nema u originalnim melodijama. Problematičan je bio i način na koji su se frekvencije kvantizirale u MIDI skalu tonova. Nekad bi izvođač melodije kao referentnu frekvenciju izabrao frekvenciju koja se nalazila baš između neka dva tona skale. Zbog dodatne nepreciznosti i frekvencijske nestabilnosti zviždanja, ta frekvencija bi se uglavnom kvantizirala tako da titra između prvog višeg i prvog nižeg tona.

Zaključak

AMT složen je problem koji je doživio veliki napredak zadnjih desetljeća. Iako se računalna transkripcija monofonih melodija smatra već gotovo riješenim problemom, transkripcija polifonih melodija i dalje ostaje otvoren zadatak. Postoje brojni pristupi koji zadatku pristupaju uporabom strojnog učenja i neuronskih mreža. S današnjim rapidnim razvojem umjetne inteligencije za očekivati je da će u skorijoj budućnosti i problem AMT biti u potpunosti riješen.

Metode korištene pri transkripciji muzičkog zviždanja nisu dale predobre rezultate, ali je uspješnost bila veća od 50% što se može smatrati dovoljnih uspjehom. Najveći problem predstavljala je sama priroda zviždanja i šum u signalu. Implementirani su samo osnovni koncepti transkripcije te ima još puno prostora za nadogradnju.

Literatura

[1] Peter Neubäcker, *Melodyne* (2001.), Celemony. Poveznica: https://www.celemony.com/en/melodyne/what-is-melodyne

[2] Bas de Haas, *Chordify* (2013.). Poveznica: https://chordify.net/

[3] Lunaversus, *Anthem Score* (2015, prosinac). Poveznica: https://www.lunaverus.com/

[4] Masri, P. *Computer modeling of sound for transformation and synthesis of musical signal*. Phd dissertation, University of Bristol, 1996.

[5] Nilsson, M., Bartunek, J. S., Nordberg, J., Claesson, I., *Human Whistle Detection and Frequency Estimation,* 2008 Congress on Image and Signal Processing, Sanya, China, 2008, pp. 737-741

[6] *Jakost zvuka*. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2013. – 2024. Pristupljeno 3.lipnja 2024.

[7] J. Schlitz. *Kunstpfeifen traditions* (2002, veljača). Poveznica: http://www.synthonia.com/artwhistling/, pristupljeno 20. Svibnja 2024.

[8] Hernandez-Olivan, C., Zay Pinilla, I., Hernandez-Lopez, C., Beltran, J.R. *A Comparison of Deep Learning Methods for Timbre Analysis in Polyphonic Automatic Music Transcription*. Electronics 2021, 10, 810

[9] de Cheveigné A, Kawahara H. *YIN, a fundamental frequency estimator for speech and music*. The Journal of the Acoustical Society of America vol. 111,4 (2002): 1917-30.

[10] Mauch, M., Dixon, S., *PYIN: A fundamental frequency estimator using probabilistic threshold distributions*, 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 659-663

[11] Shimamura, T., Kobayashi, H., *Weighted autocorrelation for pitch extraction of noisy speech*, in IEEE Transactions on Speech and Audio Processing, vol. 9, no. 7, pp. 727-730, Oct. 2001

[12] Benetos, E., Dixon, S., Duan, Z., Ewert, S., *Automatic Music Transcription: An Overview*, in IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 20-30, Jan. 2019.

[13] Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S., Eck, D., *Onsets and Frames: Dual-Objective Piano Transcription*, in 19th International Society for Music Information Retrieval Conference, Pariz, 2018

[14] Salamon, J., Gomez, E., *Melody Extraction from Polyphonic Music Signals*, 2013

[15] Smith, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*. 2. izdanje, California: San Diego, 1999.

[16] Ken Schutte, *MIDI file tools for MATLAB,* Github poveznica: https://github.com/kts/matlab-midi, poveznica na dokumentaciju: https://kenschutte.com/midi/, pristupljeno 2. lipnja 2024.

[17] Joe Wolfe, *Note names, MIDI numbers and frequencies*, The university New South Wales, Music acoustics. Poveznica: https://newt.phys.unsw.edu.au/jw/notes.html, pristupljeno 3. Lipnja 2024.

[18] David Dorran, *pitch/period tracking using autocorrelation,* (2014), dadorran. Poveznica: https://dadorran.wordpress.com/2014/09/24/pitchperiod-tracking-using-autocorrelation/, pristupljeno 23. svibnja 2024.

[19] Romain Tavenard, *An introduction to Dynamic Time Warping*, (2021). Poveznica: https://rtavenar.github.io/blog/dtw.html, pristupljeno 25. svibnja 2024.

[20] Allanon, *Attack, Decay, Sustain, Release : ADSR Explained*, (2018). Poveznica: https://www.a-mc.biz/makemusic/2018/05/30/attack-decay-sustain-release-adsr-explained/, pristupljeno 2. lipnja 2024.

[21] Dominique Vandenneucker, MIDI tutorial, Poveznica: https://www.cs.cmu.edu/~music/cmsip/readings/MIDI%20tutorial%20for%20programmers.html, pristupljeno 2. lipnja 2024.

[22] Sigtia, S., Benetos, E., Dixon, S., *An End-to-End Neural Network for Polyphonic Piano Music Transcription*, 2016

[23] Wang, B., Plumbley, M., *Musical audio stream separation by non-negative matrix factorization*, Proceedings of the DMRN Summer Conference, 23-24 July 2005, Glasgow, Scotland, UK

[24] Carvalho, R. G. C., Smaragdis, P., *Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score*, 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA): 151-155.

[25] Sigtia, S., Benetos, E., Boulanger-Lewandowski, N., Weyde, T., Garcez, A., Dixon, S., *A Hybrid Recurrent Neural Network For Music Transcription*, 2014

[26] Román, M., Pertusa, A., Calvo-Zaragoza, J., *A holistic approach to polyphonic music transcription with neural networks*, 2019

[27] Grohganz, H., Clausen, M., Müller, M., *Estimating Musical Time Information from Performed MIDI Files*, Proceedings of the International Conference on Music Information Retrieval (ISMIR), 2014

Sažetak

Naslov: Automatska notna transkripcije muzičkog zviždanja

Sažetak: Automatska notna transkripcija (AMT) postupak je u kojemu se glazbeno djelo iz zvučnog zapisa prebacuje u neki oblik pisane notacije. U ovom radu opisani su osnovni problemi i zadatci notne transkripcije. Predstavljene su neke često korištene metode i pristupi. U praktičnom dijelu radu implementirani su dijelovi jednostavnije transkripcije i klasifikacije muzičkog zviždanja. Pri detekciji visine odzviždanih tonova korištene su dvije metode; autokorelacijska funkcija u vremenskoj domeni (ACF) i nalaženje frekvencije najvećeg intenziteta u spektru signala. Nakon postupka izvlačenja značajki iz odzviždane melodije, dobivene informacije zapisuju se u MIDI formatu te se pokušavaju klasificirati u jednu od dostupnih melodija u bazi podataka. Odzviždana melodija uspoređuje se sa svim melodijama iz baze uz dinamičko vremensko poravnanje (DTW).

Ključne riječi: Automatska notna transkripcija, MIDI standard, detekcija visine tona, autokorelacijska funkcija, dinamičko vremensko poravnavanje

Summary

Title: Automatic Transcription of Musical Whistling

Summary: Automatic music transcription (AMT) is the process of converting a musical piece from an audio recording into a form of written notation. The goal of this work is to give an insight into fundamental problems and tasks of automatic transcription and to present some commonly used methods and approaches. Simpler components of transcribing and classifying the sound of musical whistling are implemented. Two methods were used for detecting the pitch: time domain autocorrelation function (ACF) and identifying the frequency with highest intensity in the spectrum of a given signal. After feature extraction from the whistled melody, the obtained information is recorded in MIDI format and attempts are made to classify it into one of the available melodies in the database. The whistled melody is compared with all the melodies in the database using dynamic time warping (DTW).

Keywords: Automatic music transcription, MIDI standard, pitch detection, autocorrelation function, dynamic time warping