Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Pešič

Avtomatska transkripcija zvočnih posnetkov tolkal

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Matija Marolt

Ljubljana, 2016

Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc http://creativecommons.si

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License, različica 3.* Podrobnosti licence so dostopne na spletni strani http://www.gnu.org/licenses/.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomskem delu preučite področje detekcije in transkripcije tolkal. Implementirajte pristop, ki transkripcijo izvede na podlagi faktorizacije spektrograma zvočnega posnetka in ga evalvirajte na ustrezni testni množici. Pri tem evalvirajte tudi različne načine za učenje baznih vektorjev pri faktorizaciji.

Zahvaljujem se mentorju doc. dr. Matiji Maroltu za strokovno vodenje in vsa pojasnila pri pisanju tega dela. Iskreno se zahvaljujem svoji družini, punci in prijateljem za podporo in pomoč skozi vsa leta študija.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
NMF	Non-negative matrix factorization	Nenegativna matrična faktorizacija
STFT	Short-time Fourier transform	Kratkočasovna Fourierjeva transformacija
MIR	Music Information Retrieval	Pridobivanje informacij iz glasbe
P	Precision	Natančnost
R	Recall	Priklic

Povzetek

Naslov: Avtomatska transkripcija zvočnih posnetkov tolkal

Avtor: Miha Pešič

Detekcija in transkripcija udarcev bobnov iz zvočnih datotek je problem, ki trenutno še nima optimalne rešitve. Poznamo več različnih metod, ki dajejo zadovoljive rezultate, vendar je popolno transkripcijo zelo težko doseči zaradi pomanjkanja informacij v zvočnem zapisu. Pristop z nenegativno matrično faktorizacijo predpostavlja, da imamo na voljo izolirane posnetke posameznega bobna, ki ga želimo zaznati. S kratkočasovno Fourierjevo transformacijo dobimo ločene transformacije za vsak časovni trenutek. Spektrograme posameznih bobnov uporabimo za nenegativno matrično faktorizacijo, rezultat tega postopka pa nam omogoča zaznavanje začetkov udarcev ter ponovno sintezo signalov. Razvit je bil sistem za transkripcijo treh različnih bobnov. Sistem smo na dva različna načina preizkusili na testni množici in primerjali rezultate.

Ključne besede: bobni, transkripcija, nenegativna matrična faktorizacija.

Abstract

Title: Automatic transcription of drum recordings

Author: Miha Pešič

Detection and transcription of drum hits from audio files is a problem, currently without an optimal solution. Multiple methods give satisfactory results but perfect transcription is hard to achieve because of lack of information in the digital recording. Approach using non-negative matrix factorization assumes that we have access to isolated recordings of every drum sound we wish to detect. Short-term Fourier transform yields separate transformations for each time frame. Isolated drum spectrograms are used for non-negative matrix factorization, the result of which we can then use for onset detection and signal synthesis. A transcription system for three different drum sounds was implemented. We tested the system in two separate scenarios and compared the results.

Keywords: drums, transcription, non-negative matrix factorization.

Poglavje 1

Uvod

1.1 Motivacija

Z napredkom tehnologije sta se produkcija in urejanje glasbe v zadnjih letih skoraj v celoti preselila iz analognega v digitalni svet. Zaradi nižjih cen in enostavnosti novih orodij je sedaj to področje veliko bolj dostopno, odpirajo se tudi nove možnosti za pristop k izobraževanju na glasbenem področju. Dandanes se z obdelavo glasbe lahko ukvarja vsak, ki ima dostop do računalnika.

Pridobivanje informacij iz glasbe (angl. MIR - Music Information Retrieval) je zadnje čase aktivno področje raziskav. Ukvarja se s prepoznavanjem ritma, prepoznavanjem višine tona, razpoznavanjem melodije, transkripcijo inštrumentov, ... V tem diplomskem delu se posvetimo transkripciji bobnov, bolj specifično transkripciji treh posameznih delov iz kompleta bobnov.

Boben je glasbeni inštrument, ki spada med tolkala, bolj natančno med membranofona glasbila. Bobne se navadno povezuje v komplet (angl. drum set), ki je sestavljen iz malega bobna (angl. snare drum), prehodnih bobnov (tom-tom), velikega bobna (angl. bass drum) in činel. Omejili se bomo na transkripcijo malega bobna, velikega bobna ter pedalnih činel (angl. hi-hat). Ti bobni se v popularni glasbi pojavljajo najpogosteje. Prav tako se bomo omejili le na posnetke, ki od inštrumentov vsebujejo samo bobne.

V glasbeni industriji je v porastu nadomeščanje posnetih bobnov s profesionalno obdelanimi posnetki udarcev (angl. sample replacement). Tu igra avtomatska transkripcija posnetkov ključno vlogo, saj dobri algoritmi prispevajo h kvalitetnemu končnemu izdelku. Transkripcija je uporabna tudi za hitro pridobitev notnega zapisa glasbe, ki je pripomoček za učenje inštrumentov. Postopki, opisani v tem delu, so uporabni tudi za segmentacijo posnetkov glasbe na posamezne inštrumente ter odstranjevanje šuma.

1.2 Cilji

Cilj diplomskega dela je preučiti področje avtomatske transkripcije glasbe in se seznaniti z metodami za transkripcijo bobnov. Praktični del obsega implementacijo sistema za transkripcijo, njegovo testiranje na bazi posnetkov ter obravnavo rezultatov.

Razvita metoda z uporabo nenegativne matrične faktorizacije daje dobre rezultate, hkrati pa ni časovno zahtevna. Implementirani sistem smo preizkusili na zbirki 95 posnetkov za dva različna scenarija:

- ko imamo na voljo tonsko vajo in se tako naučimo različnih zvokov, ki jih bomo kasneje poskusili detektirati v posnetku
- ko tega nimamo na voljo; takrat uporabimo bazne vektorje iz drugih posnetkov

Rezultati so bili primerno ovrednoteni, zanimiva pa je tudi primerjava natančnosti transkripcije med obema scenarijema.

1.3 Vsebina in struktura dela

Diplomsko delo lahko glede na vsebino razdelimo na več poglavij. V uvodu je predstavljena tema naloge, zastavljeni cilji ter oblika dokumenta.

V poglavju 2 si podrobno ogledamo predstavitev signala v digitalni obliki, njegove lastnosti in nekatere metode, ki se pogosto uporabljajo za procesiranje signalov.

Poglavje 3 opisuje različne pristope k transkripciji bobnov. Več pozornosti posvetimo metodi, ki smo jo v tej nalogi izbrali za implementacijo.

V poglavju 4 je predstavljen praktični del diplomskega dela, uporabljena orodja in metode ter celoten postopek transkripcije, ki je bil implementiran v okviru naloge.

V poglavju 5 je opisana zbirka podatkov, nad katerimi so bili izvedeni testi implementiranega sistema. Rezultate primerjamo med sabo in ovrednotimo uspešnost transkripcije.

Poglavje 2

Opis problema

Transkripcija glasbe vključuje analizo zvočnega posnetka ter zapis not, ki se pojavijo v tem posnetku, v glasbeni notaciji. Končni rezultat transkripcije je lahko detajlni zapis vseh inštrumentov ali pa zgolj okvirni opis akordov in melodije. V tem diplomskem delu transkripcija predstavlja avtomatsko zaznavanje udarcev treh najpogosteje uporabljenih bobnov iz posnetkov, ki vsebujejo zgolj zvoke teh treh bobnov. Rezultat transkripcije je v tem primeru časovni zapis udarcev posameznega bobna, ki se nato uporabi za preverjanje natančnosti transkripcije.

Da pridemo do končnega rezultata, je najprej potrebno razumevanje nekaterih osnovnih lastnosti zvoka, njegove predstavitve v obliki digitalnega signala ter matematičnega ozadja metod, ki jih uporabljamo pri transkripciji.

2.1 Zvok

Zvok je mehansko valovanje, ki se širi v dani snovi [12]. Nastaja zaradi spreminjanja tlaka v okolju. Njegovi glavni lastnosti sta frekvenca in amplituda zvočnega tlaka. Frekvenca je podatek o višini tona, amplituda pa o glasnosti zvoka. Osnovna sestavina zvoka je sinusno valovanje z določeno frekvenco. Več valovanj se s prištevanjem sestavlja v kompleksnejše zvoke.

Zvok, sestavljen iz enega samega sinusnega nihanja imenujemo ton. Zven je sestavljen iz več sinusnih nihanj z različnimi frekvencami. Navadno je to osnovni ton in višji harmonični toni. To so toni, katerih frekvence nastopajo kot celoštevilski večkratniki frekvence osnovnega tona. Šum je zvok, ki nima izrazitega tona. Vsebuje širok zvezni frekvenčni spekter. Šum nastane ob nepravilnem nihanju prožne snovi. Aperiodični šum imenujemo pok. Pok nastane s kratkotrajnim mehanskim nihanjem z veliko amplitudo.

Boben spada med glasbila brez določljivega tona (angl. unpitched instrument), čeprav to ni popolnoma pravilno, saj je nekatere dele kompleta bobnov mogoče uglasiti na določeno višino tona. Med te spadata tudi veliki boben in mali boben, ker za proizvajanje zvoka uporabljata membrano. Hi-hat po drugi strani za odzvanjanje uporablja vibriranje kovine, ki določa višino tona. Označen komplet bobnov je prikazan na sliki 2.1.

Vseeno pa pri zvoku bobnov višina tona ni tako pomembna, zato nas ta pri transkripciji ne zanima in je tudi ne zaznavamo. Bobni določajo tempo in poudarke v glasbi, zato je kritičnega pomena, da čim natančneje določimo čas udarca določenega bobna.

2.2 Signal

Zvok lahko pride v računalnik na dva različna načina. Prvi je ta, da s pomočjo pretvornika, kot je na primer mikrofon, zajamemo zvok, ki se nato z analogno-digitalnimi pretvorniki pretvori v zapis, ki je primeren za digitalno obdelavo in shrambo na disku, torej digitalni signal. Drugi način je generiranje zvočnih datotek s programsko opremo, bodisi z uporabo predvzorčenih posnetkov glasbil ali pa s sintetizatorji. Avdio signal tipično vsebuje frekvence v razponu od 20 do 20.000 Hz. To je območje, v katerem je tudi človeško uho sposobno zaznavati zvok.

Signal je funkcija ene ali več neodvisnih spremenljivk, ki se spreminja glede na te spremenljivke in običajno vsebuje informacijo. Analogni signal je zvezen po časovni in amplitudni osi. Tak signal denimo naše uho sprejema,

Slika 2.1: Označen komplet bobnov. 1 - hi-hat, 4 - mali boben, 8 - veliki boben [1]

ko poslušamo sogovorca. Preprost primer sinusnega signala lahko opišemo s funkcijo, ki se spreminja po času t:

$$x(t) = A\sin(2\pi Ft + \theta), \tag{2.1}$$

kjer A predstavlja amplitudo, F frekvenco oziroma višino tona, θ pa fazni kot.

V računalniškem svetu je nemogoče predstaviti vse (neskončno) zvezne vrednosti, zato je potrebna neke vrste aproksimacija zveznega signala. Ta se zgodi v dveh korakih. Z vzorčenjem zveznega signala pridobimo diskretni signal, ki je diskreten po času in zvezen po amplitudi. Vzorčenje pomeni, da v določenem vzorčnem intervalu T_s shranjujemo vrednosti signala v teh točkah.

Vzorčeni diskretni signal ima obliko

$$x(nT_s) = A\sin(2\pi \left(\frac{F}{F_s}\right) n + \theta), \tag{2.2}$$

kjer je F_s frekvenca vzorčenja, n pa indeks posameznega vzorca.

Po vzorčenju dobimo zaporedje števil, ki lahko zavzamejo katerokoli vrednost. Naslednji korak je kvantizacija, rezultat katere je diskretnost še po amplitudni osi. Signal, ki je diskreten po času in amplitudi imenujemo digitalni signal. Ta je predstavljen kot zaporedje diskretnih vrednosti in s takšnimi signali imamo opravka pri transkripciji.

2.2.1 Izrek o vzorčenju

S postopkom vzorčenja iz zvezne zaloge vrednosti neodvisnih spremenljivk v signalu dobimo končno zalogo vrednosti. Izrek o vzorčenju postavlja pogoje, ki jih je potrebno izpolniti, če želimo iz vzorcev digitalnega signala natančno razbrati vsebino prvotnega analognega signala.

Izrek 2.1 *Originalni zvezni signal lahko določimo iz vzorčenega signala natanko takrat, ko je najvišja frekvenca F_{max} v signalu vsaj za polovico nižja od frekvence vzorčenja F_s [8]:*

$$F_{max} < \frac{F_s}{2} \tag{2.3}$$

Če se pri vzorčenju ne držimo neenakosti (2.3), pride pri rekonstrukciji analognega signala do pojava, ki mu rečemo prekrivanje ali *aliasing*. Posledica tega je, da se frekvence originalnega signala, ki so višje od $\frac{F_s}{2}$, pretvorijo v frekvence, nižje od $\frac{F_s}{2}$, to pa vodi do neskladja novega signala s prvotnim.

2.3 Frekvenčna analiza

Čeprav pri transkripciji bobnov ne določamo višine tona, si pri veliki večini metod pomagamo s podatki o frekvenčnem spektru posnetkov. Ta nam daje vpogled ne le v časovno predstavitev signala, temveč nam pokaže, kateri toni se v posnetku pojavijo v katerem trenutku s kakšno magnitudo. Preslikava signala iz časovne v frekvenčno domeno je računsko zahtevna operacija, ki je osnovni gradnik sistema, razvitega v okviru tega diplomskega dela.

2.3.1 Fourierjeva transformacija

Vsak signal je sestavljen iz enega ali več sinusnih nihanj, ki se skupaj sestavijo s seštevanjem. Fourierjeva transformacija je dekompozicija signala na več osnovnih sinusnih valovanj z različnimi amplitudami in frekvencami. Ponuja nam drugačen pogled na isti signal, iz katerega lahko izluščimo uporabne informacije. Transformacija diskretiziranega signala se zgodi z diskretno Fourierjevo transformacijo - DFT [9] ali z njeno optimizirano različico, hitro Fourierjevo transformacijo - FFT.

DFT signala $x(n)$ je definiran s formulo

$$X_k = \sum_{n=0}^{N-1} x(n) \cdot e^{-2i\pi nk/N}, \qquad k = 0, 1, 2, \ldots, N-1. \tag{2.4}$$

Rezultat transformacije so kompleksna števila, ki nosijo informacijo o amplitudi in fazi. Te dobimo po naslednjih enačbah:

$$|X_k| = \sqrt{Re(X_k)^2 + Im(X_k)^2} \tag{2.5}$$

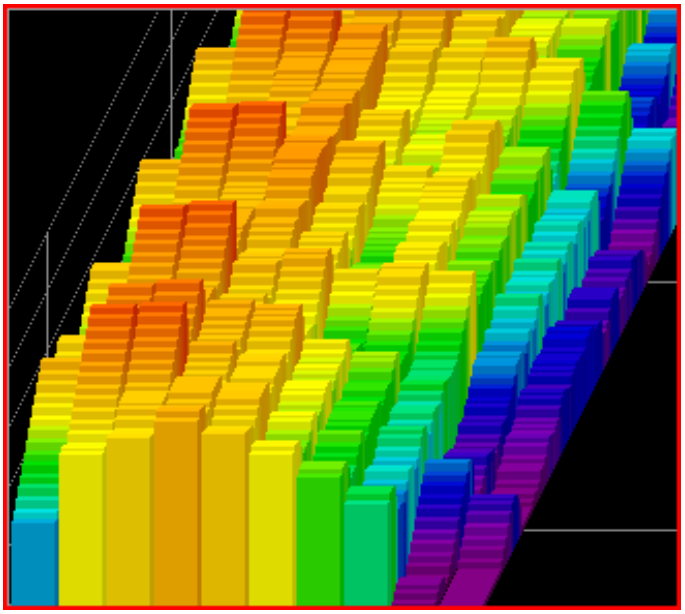
$$arg(X_k) = \arctan 2(Im(X_k), Re(X_k)), \tag{2.6}$$

kjer je $\arctan 2$ krožna funkcija arcus tangens z dvema argumentoma [3].

Algoritem za DFT ima časovno zahtevnost $O(n^2)$, medtem ko ima FFT zahtevnost $O(nlogn)$, zato se v praksi za računanje diskretne Fourierjeve transformacije v večini primerov uporablja FFT.

KRATKOČASOVNA FOURIERJEVA TRANSFORMACIJA

Kratkočasovna Fourierjeva transformacija (angl. short-time Fourier transform - STFT) določa frekvenčno in fazno vsebino dela signala v določenem časovnem odseku. Izračunamo jo tako, da signal razdelimo na več manjših odsekov ali *oken* enake dolžine, nato pa za vsakega od njih s FFT izračunamo Fourierjevo transformiranko. Za delitev signala na okna navadno uporabimo eno od okenskih funkcij. Tako dobimo frekvenčni spekter za vsako okno posebej. Kompleksni rezultat posamezne transformacije dodamo v matriko, ki shranjuje podatke o magnitudi ter fazi za vsako točko po času in frekvenci.



Slika 2.2: Analiza avdio signala skozi čas s pomočjo STFT [2]

Matrika magnitud STFT nam daje *spektrogram*. Spektrogram je vizualna predstavitev frekvenčnega spektra v zvoku ali v poljubnem signalu, ki se spreminja po času ali kakšni drugi spremenljivki. Primer vizualizacije signala po STFT je prikazan na sliki 2.2.

Problem pri kratkočasovni Fourierjevi transformaciji predstavlja omejitev ločljivosti rezultata. Te se ne da izboljšati, temveč je potrebno sprejeti kompromis med boljšo ločljivostjo v frekvenčnem ali časovnem prostoru. Širina okenske funkcije narekuje predstavitev signala. Širše okno daje boljšo frekvenčno ločljivost, a slabšo časovno ločljivost. Tako lahko boljše razpoznavamo frekvence, ki so bliže skupaj, ne moremo pa zelo natančno določiti trenutka, ko se pojavijo. Obratno velja za ožje okno, ki izboljša časovno ločljivost in poslabša frekvenčno.

OKENSKE FUNKCIJE

FFT predpostavlja, da je signal, nad katerim izvajamo transformacijo, periodičen, torej je konec signala direktno povezan z začetkom. Za večino

signalov pri transkripciji to ne velja, rezultat tega pa je, da FFT v končnih točkah signala prikaže prisotnost visokih frekvenc, ki jih v resnici tam ni [13].

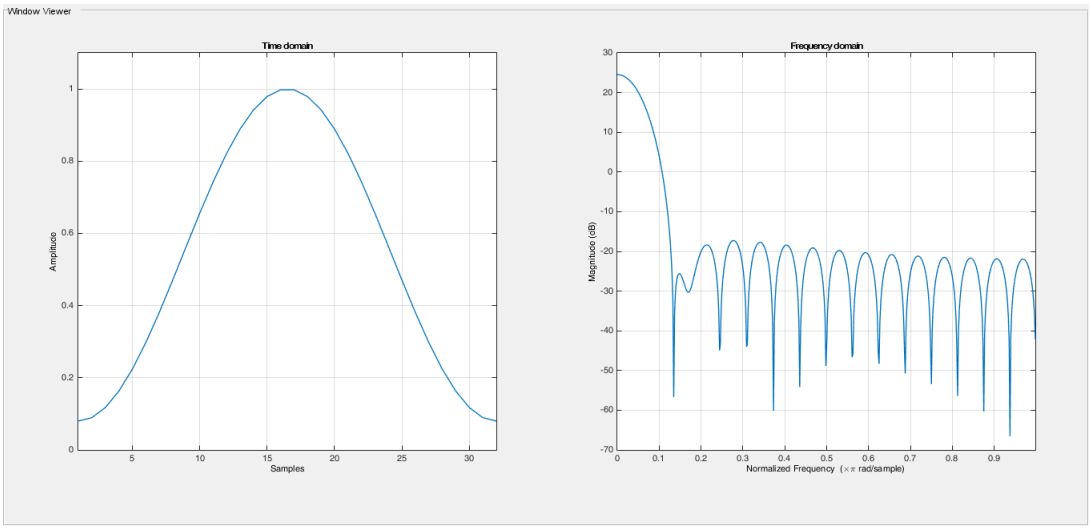
Delitev signala na odseke, iz katerih računamo Fourierjevo transformacijo, izvedemo z množenjem signala z izbrano okensko funkcijo. Ta proces imenujemo oknjenje. Oknjenje minimizira napake FFT na aperiodičnih signalih. Okenska funkcija je matematična funkcija, ki ima ničelno vrednost povsod, razen znotraj določenega intervala. Zelo enostaven primer take funkcije je pravokotno okno, ki ima znotraj intervala konstantno vrednost.

Poznamo več primerov okenskih funkcij, v praksi pa sta pogosto uporabljeni Hannovo in Hammingovo okno. Obe sta si precej podobni in imata enostavno sinusoidno obliko. Sredina okna ima širok vrh, proti robu pa se postopoma spušča proti nič. Hannovo okno ničlo doseže, medtem ko se ji Hammingovo zgolj približa.

Hammingovo okno je podano s formulo

$$w_n = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \tag{2.7}$$

kjer N predstavlja število vzorcev oziroma širino okna, n pa število na intervalu $0 \leq n \leq N-1$. Hammingovo okno je prikazano na sliki 2.3.



Slika 2.3: Vizualizacija Hammingovega okna z 32 vzorci

Poglavje 3

Pregled področja

Reševanja problema transkripcije se lahko lotimo na več različnih načinov. Transkripcija bobnov se razlikuje od transkripcije drugih inštrumentov zaradi lastnosti samega inštrumenta. Posamezen boben skozi glasbeno delo ne spreminja višine tona, zato te ni potrebno določati. Namesto tega se posvetimo razločevanju posameznega dela kompleta bobnov iz skupnega posnetka, najpomembnejše pa je čim natančneje določiti čas udarcev. V tem poglavju na kratko predstavimo nekaj različnih pristopov k transkripciji bobnov, metodam, uporabljenim v tem diplomskem delu pa namenimo bolj podroben opis. Naslednjih nekaj razdelkov je povzetih po članku [4].

3.1 Ujemanje vzorcev zvoka

Ta metoda predpostavlja, da poznamo različne zvoke, ki jih moramo v posnetku prepoznati. Najprej si shranimo vzorec vsakega posameznega zvočnega dogodka, ki nas zanima. To lahko storimo tako, da si zapomnimo povprečno vrednost frekvenčnega sestava udarca bobna ali činele. Nato v spektrogramu originalnega posnetka iščemo najbolj podobne vrednosti. Najprej določimo dogodke z največjim ujemanjem, vzorce zvokov pa sproti prilagajamo posnetku.

Ko določimo nekaj najbolj podobnih dogodkov v posnetku, vzorcu zvoka

prištejemo povprečno vrednost frekvenčnega sestava teh dogodkov. Novi vzorec ponovno uporabimo za iskanje podobnih dogodkov v signalu. Skozi iteracije se iskalni vzorec približuje iskanemu zvoku. V zadnjem koraku, ko dobimo že dober približek zvokov v signalu, določimo, ali se je določen dogodek res zgodil. Primer takega pristopa je [14].

3.2 Nadzorovano učenje

Ta kategorija transkripcijskih algoritmov najprej izvede časovno segmentacijo posnetkov, nato pa se posamezen časovni okvir klasificira v skupino dogodkov. V našem primeru bi najprej določili čas vseh udarcev bobnov, ne da bi jih ločili med sabo. Nato bi za vsak udarec določili, za kakšen boben dejansko gre.

Časovna segmentacija se lahko izvede na več načinov. Posnetek lahko razdelimo na dele tako, da nov del začnemo pri vsakem naslednjem zaznanem udarcu bobna. Lahko pa celoten posnetek preprosto razdelimo na časovno mrežo določene ločljivosti. Na ta način dobimo časovne odseke enake dolžine. Ta način je enostavnejši za obdelavo, vendar so rezultati odvisni od ločljivosti mreže. Če so posamezni časovni odseki predolgi, se v enem lahko pojavita dva udarca različnih bobnov. V tem primeru klasifikacija ne bi bila točna.

Ko signal razdelimo na odseke, se iz vsakega naučimo nekaj osnovnih značilnosti, ki jih uporabimo za klasifikacijo. Z izbranim klasifikacijskim algoritmom nato določimo izvor posameznega zvoka. Članek [10] opisuje algoritem z metodo podpornih vektorjev (angl. support vector machine).

3.3 Ločevanje virov zvoka

Signal najprej ločimo glede na različne vire zvoka, nato pa izvedemo detekcijo udarcev nad posameznim delom. Prvi korak je kot ponavadi transformacija signala iz časovne v frekvenčno domeno s kratkočasovno Fourierjevo transformacijo. Sledi dekompozicija magnitudnega spektrograma X na več neod-

visnih spektrogramov vseh virov zvoka, ki so prisotni v originalnem signalu. Ti so predstavljeni kot produkt bazne funkcije B in amplitudne ovojnice G. B pove, kako določen vir zvoka zveni, medtem ko G določa, kdaj in s kakšno močjo se zvok pojavi v času. Znanih je več pristopov k reševanju tega problema, nekateri od teh so:

- Independent component analysis (ICA)
- Independent subspace analysis (ISA) [11]
- Prior subspace analysis (PSA) [6]
- Nenegativna matrična faktorizacija (NMF)

Nenegativno matrično faktorizacijo smo uporabili v praktičnem delu te diplomske naloge, zato je naslednji razdelek namenjen podrobnejšemu opisu te matematične metode.

3.4 Nenegativna matrična faktorizacija

Nenegativna matrična faktorizacija (NMF) spada v skupino algoritmov v statistični analizi in linearni algebri, kjer matriko X razčlenimo v dve matriki, W in H, katerim je skupna lastnost, da ne vsebujejo negativnih elementov. Matriki W in H iščemo tako, da velja:

$$X \approx WH \tag{3.1}$$

Problem v splošnem ni enostavno rešljiv, zato z numeričnimi metodami iščemo približke rešitve. Ker je magnitudni spektrogram, ki je rezultat STFT, prav tako nenegativna matrika, je NMF postopek, ki se ga pri procesiranju signalov pogosto poslužujemo.

Naj bo matrika X dimenzij $n \times m$. Faktoriziramo jo lahko kot produkt W dimenzij $m \times r$ in H dimenzij $r \times n$, kjer r lahko zavzame vrednost, ki je poljubno manjša od n in m. Iz tega sledi, da za aproksimacijo potencialno zelo velike matrike potrebujemo dve matriki, ki sta mnogo manjši.

Iterativni algoritmi za NMF so enostavni za implementacijo in dajejo dobre rezultate. Delujejo tako, da na začetku inicializiramo matriki W in H s poljubnimi vrednostmi, nato pa vsako iteracijo izvedemo posodobitev teh matrik tako, da je njun produkt vsakič bolj podoben originalni matriki. Vrednosti novih matrik se izračunajo z množenjem trenutnih matrik z določenim faktorjem, ki zagotavlja izboljšan rezultat po vsaki operaciji množenja. Iteracije ponavljamo, dokler ne dosežemo maksimalnega števila iteracij ali pa se dovolj približamo originalni matriki. Konvergenca se določa na več različnih načinov, vsak predlaga svojo cenilno funkcijo, s katero se oceni natančnost približka.

3.4.1 Cenilne funkcije

EVKLIDSKA RAZDALJA

Podobnost med A in B lahko enostavno merimo s kvadratom evklidske razdalje:

$$\|A - B\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2 \tag{3.2}$$

Ta ima spodnjo mejo 0, ki jo doseže natanko takrat, ko velja $A = B$. Multiplikativna pravila za posodobitev matrik W in H, ki minimizirajo razdaljo $\|X - WH\|^2$, so definirana kot:

$$H_{ij} \leftarrow H_{ij} \frac{(W^T X)_{ij}}{(W^T W H)_{ij}} \tag{3.3}$$

$$W_{ki} \leftarrow W_{ki} \frac{(X H^T)_{ki}}{(W H H^T)_{ki}} \tag{3.4}$$

KULLBACK-LEIBLER DIVERGENCA

To je še ena informativna mera, ki kaže na podobnost oziroma razliko dveh matrik. Kot evklidska razdalja ima prav tako lastnost, da je njena spodnja meja 0, ki nastopi pri pogoju $A = B$.

$$D(A\|B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right) \tag{3.5}$$

Formula velja, če A in B predstavljata verjetnostni porazdelitvi. Prav tako ni simetrična - če A in B zamenjamo, ne pričakujemo enakega rezultata. Divergenca $D(X\|WH)$ deluje z multiplikativnimi pravili:

$$H_{ij} \leftarrow H_{ij} \frac{\sum_k W_{ki} X_{kj} / (WH)_{kj}}{\sum_l W_{li}} \tag{3.6}$$

$$W_{ki} \leftarrow W_{ki} \frac{\sum_j H_{ij} X_{kj} / (WH)_{kj}}{\sum_n H_{in}} \tag{3.7}$$

Dokazi zgornjih trditev so dobro opisani v članku [7].

3.4.2 NMF v sistemu za transkripcijo

Digitalni signal, definiran z zaporedjem diskretnih vrednosti, lahko s kratkočasovno Fourierjevo transformacijo pretvorimo v matriko, ki nosi podatke o frekvenčni vsebini signala. Vsaka vrstica te matrike predstavlja frekvenčni pas, vsak stolpec pa trenutek v času. Ker se v tem diplomskem delu ukvarjamo s posnetki, ki vsebujejo le tri različne vire zvoka, lahko za transkripcijo uporabimo nenegativno matrično faktorizacijo, kjer sta matriki W in H dimenzij $m \times 3$ in $3 \times n$. Matriko W inicializiramo z baznimi vektorji vsakega od bobnov, ki ga želimo zaznati v posnetku. Nato skozi iteracije pridemo do matrike H, ki nam pove, kdaj se pojavi udarec katerega bobna.

Poglavje 4

Razvoj sistema za transkripcijo

V tem poglavju je opisan praktični del naloge in uporabljena orodja. V okviru diplomskega dela je bil implementiran sistem za transkripcijo zvočnih posnetkov, ki vsebujejo zvoke treh različnih bobnov. Sistem temelji na postopkih, opisanih v članku [4], glavna uporabljena metoda pa je nenegativna matrična faktorizacija. Ta za inicializacijo uporablja bazne vektorje, naučene na posnetkih posameznih bobnov, zato smo predpostavili dva možna scenarija uporabe sistema.

Prva možnost je, da imamo na voljo izolirane posnetke vsakega bobna, ki je prisoten v posnetku, torej neke vrste tonsko vajo. V tem primeru se bazne vektorje lahko naučimo na zvokih, ki jih tudi zaznavamo.

Druga možnost pa je, da nimamo dostopa do teh posnetkov, zato moramo za bazne vektorje uporabiti zvoke, ki niso nujno podobni tistim v posnetku, nad katerim izvajamo transkripcijo. Uporabimo povprečno vrednost več različnih zvokov posameznega tipa bobna, da dosežemo čim bolj splošne vrednosti. Pri drugem scenariju pričakujemo tudi nekoliko slabšo natančnost transkripcije.

4.1 Razvojno okolje

MATLAB® (MATrix LABoratory) je visokonivojski jezik, ki je optimiziran za operacije z vektorji in matrikami. Pogosto se uporablja na področju analize podatkov in matematike, ponuja pa tudi orodja za vizualizacijo, načrtovanje uporabniških vmesnikov ter komunikacijo z drugimi programskimi jeziki, kot so C, Python in Java. Vsebuje mnogo matematičnih orodij, ki jih potrebujemo za procesiranje signalov, zato je primerna izbira za implementacijo te naloge. Privlačen je tudi zato, ker je sintaksa zelo podobna pisanju enačb na papir.

Razširitev *SignalProcessingToolbox* ponuja funkcije za sintezo, merjenje, transformiranje, filtriranje in vizualizacijo signalov.

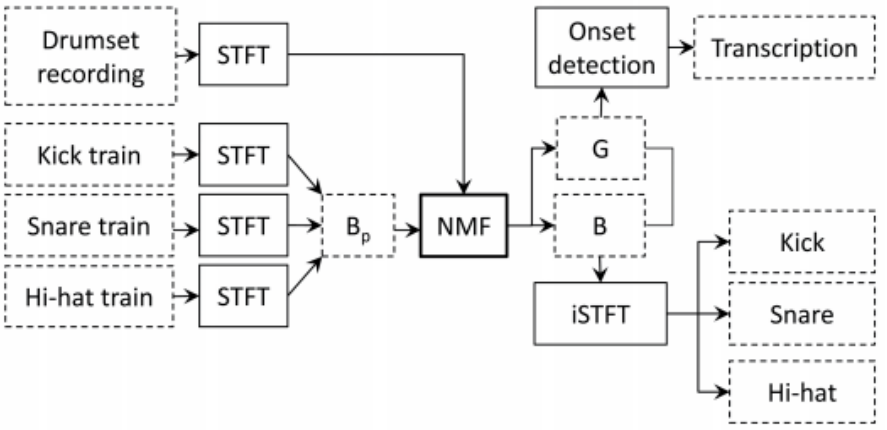
4.2 Implementacija

Slika 4.1 dobro prikazuje sestavne dele sistema, zato bomo v tem poglavju uporabljali enake oznake kot na diagramu. Metoda predpostavlja, da imamo poleg originalnega posnetka za transkripcijo na voljo izolirane posnetke bobnov, ki jih želimo v posnetku razpoznati. Začnemo torej z nalaganjem štirih posnetkov, ki se v programu predstavijo kot vektorji:

- **Drumset recording:** vsebuje zvoke vseh treh bobnov; nad tem posnetkom želimo izvesti postopek transkripcije
- **Kick train:** vsebuje samo udarce velikega bobna
- **Snare train:** vsebuje samo udarce malega bobna
- **Hi-hat train:** vsebuje samo udarce činele hi-hat

Sledi izračun STFT vsakega od teh posnetkov, saj si bomo v nadaljevanju pomagali s podatki o frekvenčni vsebini vsakega posnetka.

Nato nad spektrogramom originalnega posnetka izvedemo nenegativno matrično faktorizacijo, ki nas privede do aktivacijske matrike G, ki vsebuje informacijo o tem, kdaj se na časovni osi pojavi udarec katerega bobna.



Slika 4.1: Diagram postopka v članku [4]

Ta matrika je vhod za naslednji del sistema, v katerem se odločimo, katere zaznane udarce bomo šteli kot veljavne, katerih pa ne bomo upoštevali. Na ta način se izognemo prevelikemu številu udarcev, kar bi slabo vplivalo na natančnost transkripcije. V tem koraku tudi določimo natančen časovni trenutek, ko se je udarec bobna pojavil v posnetku.

Tu se naš postopek transkripcije zaključi, implementiran pa je ocenjevalni sistem, ki na podlagi več posnetkov poroča o natančnosti za vsak boben. Več o rezultatih v poglavju 5. V nadaljevanju sledi nekaj razdelkov, kjer natančneje opišemo zanimive in ključne dele sistema.

4.2.1 Učenje baznih vektorjev

V tem koraku se v programu naučimo lastnosti zvokov, ki jih želimo izluščiti iz posnetka. Bazni vektor vsebuje povprečno informacijo o frekvenčni vsebini zvokov izoliranega bobna. Ker boben z igranjem ne spreminja višine tona, ampak se zvok spreminja glede na to, kako je bil boben zaigran, lahko za učenje uporabimo daljši posnetek, ki vsebuje udarce istega bobna.

1. **Scenarij s tonsko vajo:** v program naložimo posnetke vseh treh izoliranih bobnov. Za vsakega izračunamo magnitudni spektrogram in shranimo matriko. V vsaki matriki nato seštejemo vse stolpce, ki

predstavljajo informacijo o vsebini celotnega frekvenčnega spektra v določenem časovnem trenutku. Dobljeni vektor še delimo s številom vseh stolpcev, da dobimo povprečno vrednost frekvenčnega spektra bobna. Tako pridobljene tri bazne vektorje nato sestavimo v matriko na način, da vsak od vektorjev zavzame svoj stolpec.

2. **Scenarij brez tonske vaje:** izvedemo 5-kratno prečno preverjanje. Bazo posnetkov razdelimo na 5 delov enake velikosti. Štiri petine posnetkov uporabimo za učenje baznih vektorjev, kar se zgodi na enak način kot prej, le da namesto enega posnetka analiziramo več posnetkov. Bazne vektorje bobnov istega tipa seštejemo in izračunamo povprečno vrednost, nato pa jih sestavimo v matriko. Preostalo petino posnetkov bomo uporabili za testiranje. Ko dobimo rezultate, podatke zamenjamo, tako da testiramo nad naslednjo petino posnetkov. V petih iteracijah pride vsak posnetek na vrsto za transkripcijo. Tako opravimo testiranje, ne da bi bazne vektorje dobili iz zvokov, ki se pojavijo v istem posnetku. Na tak način preprečimo preveliko prileganje podatkom, kar bi sicer pozitivno vplivalo na rezultate, a ti ne bi bili reprezentativni.

4.2.2 NMF

Z nenegativno matrično faktorizacijo poskušamo magnitudni spektrogram originalnega posnetka razdeliti na dve matriki tako, da nam produkt teh dveh daje čim bolj točen približek prave matrike. Za matriko B določimo matriko baznih vektorjev, ki smo jo izračunali v prejšnjem koraku. Matriko G dimenzij $3 \times n$ smo inicializirali z naključnimi celimi števili od 1 do 5. Tej matriki pravimo tudi aktivacijska matrika, saj bo po koncu postopka NMF vsaka njena vrstica vsebovala aktivacije bobna, katerega bazni vektor je v pripadajočem stolpcu matrike B.

Sam postopek NMF je prilagojen hitrosti računanja, saj je bilo potrebno obdelati veliko bazo posnetkov. Zato smo število iteracij algoritma nastavili na 25. To se je izkazalo za najučinkovitejšo izbiro, saj je program dovolj

hiter, rezultat pa je v primerjavi z več iteracijami zelo primerljiv, oziroma je natančnost, ki jo pridobimo z na primer 100 iteracijami, zanemarljiva.

Matriko B smo torej inicializirali z baznimi vektorji. S preizkušanjem različnih metod smo prišli do zaključka, da najboljše rezultate daje NMF, kjer te matrike sploh ne posodabljamo. Tako vsako iteracijo izvedemo eno posodobitev manj, s čimer še nekoliko pridobimo na časovni zahtevnosti.

NMF izvajamo nad vsakim stolpcem matrike originalnega posnetka posebej, zato tudi ne posodobimo celotne matrike G hkrati, ampak se tega lotimo postopoma. Končna zanka NMF izgleda tako:

```
Gn = [];
for i = 1:c
    Xi = X(:, i);
    Gi = G(:, i);
    for j = 1:nIterations
        Gi = Gi.*(B'*(Xi./(B*Gi)))./(B'*ones(r, 1));
    end
    Gn = [Gn, Gi];
end
```

Tu je Gn končna matrika aktivacij, Xi in Gi sta stolpca originalne matrike X in naključno inicializirane matrike G, B je matrika baznih vektorjev, c pa je število stolpcev. Spremenljivka $nIterations$ je nastavljena na 25.

4.2.3 Detekcija začetkov udarcev

Po nenegativni matrični faktorizaciji vsaka vrstica matrike aktivacij vsebuje podatke o prisotnosti udarcev bobna v signalu. K detekciji začetkov udarcev (angl. onset detection) pristopimo tako, da najprej vsako vrstico - torej vektor - aktivacij normaliziramo. S tem dosežemo, da je najvišja vrednost v vektorju enaka 1.

Nato se v zanki sprehodimo čez celotno dolžino vektorja in spremljamo vrednosti. Če trenutna vrednost predstavlja lokalni maksimum in presega

določeno mejo, vrednost vektorja v tej točki ohranimo. V nasprotnem primeru vrednost nastavimo na 0. Prag, ki določa, ali se vrednost kvalificira za udarec bobna, smo določili eksperimentalno. S poskusi smo prišli do ugotovitve, da je optimalna vrednost praga 0.5 za mali boben in veliki boben, 0.3 pa za činelo hi-hat.

Poleg do sedaj omenjenih pogojev smo upoštevali še dejstvo, da se več udarcev istega bobna ne more zgoditi v zelo kratkem časovnem obdobju. Ker imajo bobni, obravnavani v tem diplomskem delu, zelo kratek čas odzvanjanja, smo predpostavili, da se mora energija med dvema zaporednima udarcema znižati pod mejo 0.2. Če se to ne zgodi in naletimo na nov lokalni maksimum, ga ne označimo za udarec bobna.

Tako nam v vektorju ostanejo neničelne vrednosti le tam, kjer se pojavi udarec bobna. Iz teh indeksov lahko izračunamo čas v posnetku, vektor s časovnimi vrednostmi udarcev pa je končni rezultat tega koraka. Ta postopek izvedemo ločeno za vsak boben, podatke pa pošljemo naprej k evalvaciji, ki je zadnji del našega sistema.

Poglavje 5

Testiranje in rezultati

V tem poglavju se posvetimo testiranju razvitega sistema. Predstavljena je uporabljena zbirka posnetkov in nekaj informativnih mer, ki jih uporabimo pri ovrednotenju natančnosti transkripcije. Za vsak način testiranja predstavimo rezultate in jih primerjamo med sabo. Sledi nekaj sklepnih ugotovitev in možnosti izboljšanja sistema.

5.1 Podatki

Za testiranje smo uporabili bazo podatkov IDMT-SMT-Drums. To je baza posnetkov srednje velikosti, namenjena pa je za testiranje algoritmov za avtomatsko transkripcijo bobnov in segmentacijo zvoka. Pod licenco *Creative Commons Attribution-ShareAlike 4.0 International License* so podatki na voljo na [5].

Bazo sestavlja 560 zvočnih datotek v formatu WAV (Mono, 16-bit), vzorčenih pri 44.1 kHz. Skupna dolžina posnetkov je približno 2 uri. Vsebuje 95 polifoničnih posnetkov kompletov bobnov, ki vsebujejo udarce malega (snare) bobna, velikega (bas) bobna ter pedalnih činel (hi-hat). Za vsakega od teh je na voljo še izoliran posnetek, tako da imamo poleg datoteke z vsemi tremi bobni še posamezne zvoke, ki se v originalnem posnetku pojavljajo. Tako je na voljo 285 datotek, namenjenih učenju frekvenčne vsebine bobnov.

V bazi so vključeni zvoki bobnov iz treh različnih virov:

- Akustični komplet bobnov
- Vzorčen (angl. sampled) komplet bobnov
- Sintetizator bobnov

Za vsak posnetek so bili udarci malega bobna, velikega bobna ter hi-hata ročno transkribirani in so priloženi v obliki XML ter SVL datotek. Prav tako je priložena koda za nalaganje in branje teh datotek.

5.2 Rezultati

Rezultate transkripcije smo primerjali z vrednostmi, ki so bile priložene bazi podatkov. Za vsakega od dveh scenarijev je bilo izvedeno testiranje na celotni bazi posnetkov. Transkribiranih je bilo torej 95 posnetkov, skupno pa poročamo o povprečnih vrednostih rezultatov. Ti se namreč razlikujejo od posnetka do posnetka, ker na učinkovitost transkripcije vpliva tudi zvok bobnov. Če sta si na primer zvoka dveh različnih bobnov zelo podobna, ju lahko algoritem zameša med sabo. Menimo, da je uporabljena baza dovolj obširna, da daje dober vpogled v natančnost našega sistema in se pri transkripciji kateregakoli posnetka lahko pričakuje podobne rezultate.

Poročamo o povprečni vrednosti treh različnih mer uspešnosti transkripcije:

1. **Natančnost** (angl. Precision): je mera relevantnosti rezultatov. Pove nam, koliko od vseh transkribiranih udarcev bobnov smo pravilno zabeležili. Definirana je z enačbo

$$P = \frac{T_p}{T_p + F_p} \tag{5.1}$$

2. **Priklic** (angl. Recall): pove nam, koliko od vseh dejanskih udarcev bobnov smo uspeli prepoznati. Definicija:

$$R = \frac{T_p}{T_p + F_n} \tag{5.2}$$

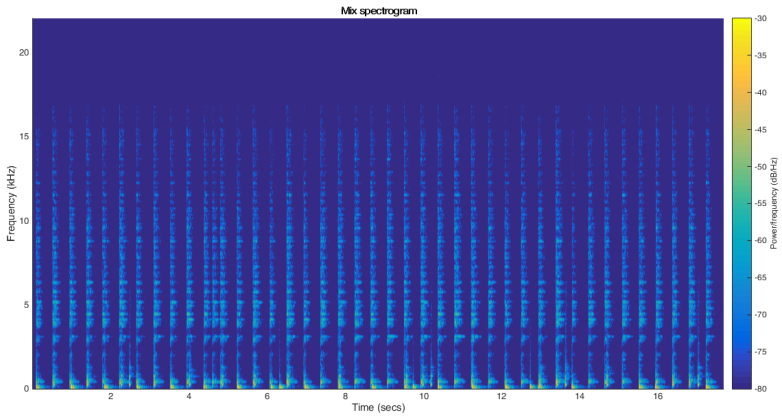
3. **Mera F** (angl. F-mesure): je harmonična sredina natančnosti in priklica. Dobimo predstavo o uspešnosti obeh:

$$F - measure = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.3}$$

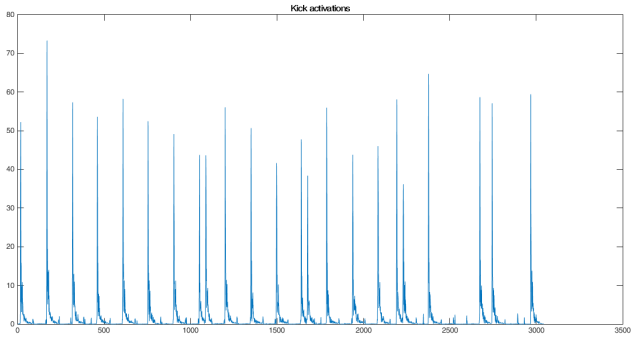
Vse tri mere imajo zalogo vrednosti od 0 do 1, kjer vrednost 1 pomeni popolnoma pravilen rezultat.

Za pravilno transkribiran udarec smo šteli vsako vrednost, ki se od dejanske ne razlikuje za več kot 50 milisekund. Vsi ostali so označeni kot napake.

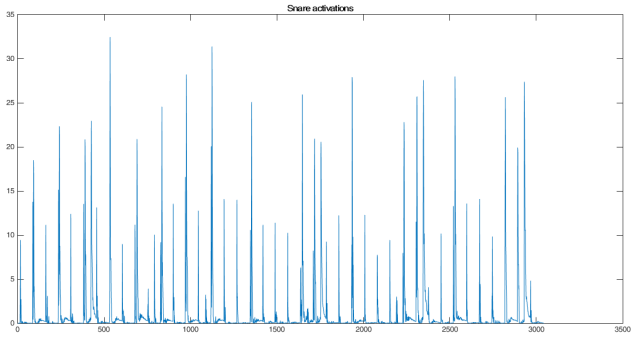
Slika 5.1 prikazuje spektrogram posnetka, nad katerim bomo izvedli transkripcijo. Vidimo, da vsebuje udarce treh različnih virov, ki so si različni po frekvenčni vsebini. Na slikah 5.2, 5.3 in 5.4 so prikazani udarci posameznih bobnov, kot jih dobimo po postopku nenegativne matrične faktorizacije. Nad takšnimi podatki izvedemo še detekcijo začetkov udarcev, rezultat tega pa preverimo s testnimi podatki.



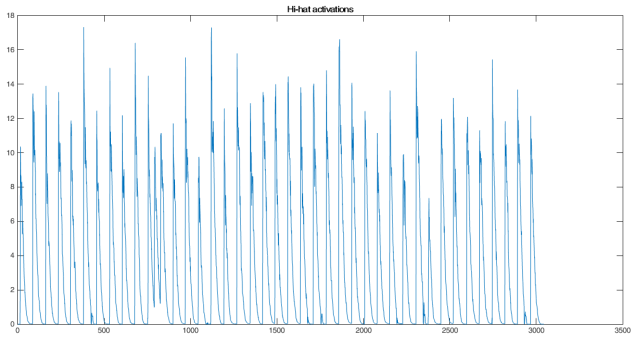
Slika 5.1: Spektrogram originalnega posnetka s tremi bobni



Slika 5.2: Aktivacije velikega bobna po postopku NMF



Slika 5.3: Aktivacije malega bobna po postopku NMF



Slika 5.4: Aktivacije hi-hata po postopku NMF

5.2.1 Scenarij s tonsko vajo

V tem primeru smo imeli za učenje baznih vektorjev na voljo zvoke, ki se pojavijo v posnetku, ki ga transkribiramo. Za vsak posnetek se najprej izvede učenje zvoka, nato pa še transkripcija. Rezultati so podani v tabeli 5.2.1.

Boben	Natančnost	Priklic	Mera F
Mali boben	0.91692	0.90788	0.89963
Veliki boben	0.92368	0.94642	0.92558
Hi-hat	0.95004	0.8702	0.89637

Tabela 5.1: Rezultati transkripcije s tonsko vajo.

Opazimo, da najboljšo mero F dosežemo pri transkripciji udarcev velikega bobna, sledi mali boben in nato činela hi-hat. Natančnost in priklic sta si relativno blizu, razen v primeru hi-hata, kjer je razlika večja. Tu imamo dobro vrednost natančnosti in nekoliko nižji priklic. Iz tega lahko sklepamo, da so zaznani udarci hi-hata večinoma pravilno transkribirani, vendar pa algoritem nekaterih udarcev ne zazna.

5.2.2 Scenarij brez tonske vaje

Ker predpostavimo, da nimamo na voljo posnetkov bobnov, ki se v originalnem posnetku pojavijo, za testiranje tega scenarija uporabimo 5-kratno prečno preverjanje. Rezultati so v tabeli 5.2.2

Boben	Natančnost	Priklic	Mera F
Mali boben	0.86828	0.88544	0.8617
Veliki boben	0.91154	0.94658	0.916
Hi-hat	0.92727	0.84736	0.87101

Tabela 5.2: Rezultati transkripcije brez tonske vaje.

Kot pričakovano, so rezultati v tem primeru nekoliko slabši. Pri nekaterih posnetkih smo opazili, da ni bil pravilno transkribiran niti en udarec, medtem

ko smo pri drugih dosegli popoln rezultat. Razlog za to je najverjetneje ta, da so v posnetkih s slabim rezultatom prisotni nestandardni zvoki. Majhna razlika med obema scenarijema se pojavlja pri velikem bobnu, medtem ko se natančnost transkripcije malega bobna in hi-hata občutno poslabša. To pripisujemo dejstvu, da so si različni zvoki velikega bobna med sabo praviloma bolj podobni, medtem ko se zvoki malega bobna in hi-hata bolj razlikujejo. Tej pomanjkljivosti bi se lahko izognili tako, da bi v uporabniškem vmesniku ponudili več predpripravljenih vzorcev bobnov, med katerimi bi uporabnik izbral najbolj podobne tistim v posnetku. Na teh bi nato izvedli učenje baznih vektorjev kot pri scenariju s tonsko vajo.

5.2.3 Ugotovitve

Transkripcijski sistem z nenegativno matrično faktorizacijo se na bazi posnetkov obnese dobro, nekoliko slabši pa so rezultati transkripcije udarcev hi-hata in malega bobna. Razlogi za to se lahko razlikujejo od posnetka do posnetka, v splošnem pa velja, da imata ta dva bobna en drugemu bolj sorodno frekvenčno vsebino in ju je zato težje pravilno razločiti ob hkratnem odzvanjanju več različnih bobnov. V primeru akustičnega kompleta bobnov moramo upoštevati še dejstvo, da so udarci zaradi človeškega faktorja zaigrani nekonsistentno in so zato zvoki istega bobna lahko zelo različni, spreminja pa se tudi glasnost udarcev.

Glede na število transkribiranih posnetkov in raznolikost zvokov v bazi so rezultati zadovoljivi, sistem pa bi se seveda dalo izboljšati. Možnost za napredek vidimo v dodelavi algoritma za detekcijo začetkov udarcev, kjer bi se prag zaznavanja lahko dinamično prilagajal posnetku. Poleg boljše natančnosti transkripcije je možna še razširitev na zaznavanje več vrst bobnov in na posnetke, kjer so poleg bobnov prisotni tudi drugi inštrumenti.

Poglavje 6

Zaključek

V tem diplomskem delu je bil obravnavan problem avtomatske transkripcije bobnov. Predstavili smo pot, ki jo zvok prepotuje od pretvorbe v digitalni signal do končnega rezultata. Opisali smo nekatere matematične metode, ki jih pri tem pogosto uporabljamo in na kratko podali nekaj različnih pristopov k transkripciji bobnov. Implementiran je bil sistem za transkripcijo, ki se problema loti z ločevanjem posnetka na različne vire zvoka. To smo storili z metodo nenegativne matrične faktorizacije. Ta se pogosto uporablja za reševanje transkripcijskih in drugih problemov, njene prednosti pa so enostavnost, uspešnost ter hitrost izračuna. Natančnost tega sistema smo preverili s testiranjem na obširni zbirki posnetkov, rezultati pa so pokazali na primernost te metode za transkripcijske probleme. Pokazali smo, da sistem dobro deluje na posnetkih, za katere vemo, kakšne zvoke vsebujejo. Pri posnetkih, kjer nimamo predznanja o zvokih bobnov so sicer rezultati nekoliko slabši, a še vedno zadovoljivi.

Uporaba takega sistema je možna na več področjih. Je pripomoček za hiter približek transkripcije daljših glasbenih posnetkov, kjer do neke mere olajša postopek ročne anotacije glasbe. Algoritem je z nekaj dodelavami primeren tudi za računanje v realnem času, kar odpira možnosti za uporabo na področju učenja igranja inštrumentov ter glasbenih iger. Primer take uporabe je program, ki v realnem času zajema zvok akustičnega kompleta bob-

nov in ocenjuje pravilnost zaigranih udarcev. Nenazadnje pa vidimo možno uporabo v glasbeni produkciji, kjer se pogosto posnete zvoke bobnov zamenjuje s profesionalno obdelanimi posnetki. Tak sistem bi občutno skrajšal čas programiranja MIDI datotek.

Razviti sistem je dal rezultate, ki so primerljivi z drugimi obstoječimi metodami. Prostor za izboljšanje je predvsem v algoritmu za detekcijo začetkov udarcev. Veliko izboljšanje smo dosegli s predpostavko, da zvok med dvema zaporednima udarcema hitro odzveni, menimo pa, da je potencial še v dinamičnem določanju praga zaznavanja udarcev. Poleg tega možne dodelave obsegajo še transkripcijo bobnov ob prisotnosti drugih inštrumentov, zaznavanje več vrst bobnov in določanje tempa glasbe.

Literatura

[1] Labelled drum kit. http://drumsandpercussion.co.uk/blog/wp-content/uploads/2014/05/drum-kit-labeled.jpg. Accessed: 2016-03-25.

[2] An stft being used to analyze an audio signal across time. https://upload.wikimedia.org/wikipedia/commons/9/9b/Short_time_fourier_transform.PNG. Accessed: 2016-03-25.

[3] atan2. Dosegljivo: https://en.wikipedia.org/wiki/Atan2, 2016. [Dostopano: 1. 8. 2016].

[4] Christian Dittmar and Daniel Gärtner. Real-time transcription and separation of drum recordings based on nmf decomposition. In *DAFx*, pages 187–194, 2014.

[5] Christian Dittmar Daniel Gärtner Feliks Weber, Manuel Winges. Idmt-smt-drums. Dosegljivo: http://http://www.idmt.fraunhofer.de/en/business_units/m2d/smt/drums.html.

[6] Derry FitzGerald, Robert Lawlor, and Eugene Coyle. Prior subspace analysis for drum transcription. In *Audio Engineering Society Convention 114*. Audio Engineering Society, 2003.

[7] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[8] Bruno A Olshausen. Aliasing. *PSC 129–Sensory Processes*, pages 3–4, 2000.

[9] Julius O Smith. *Mathematics of the discrete Fourier transform (DFT): with audio applicaitons*. Julius Smith, 2007.

[10] Koen Tanghe, Sven Degroeve, and Bernard De Baets. An algorithm for detecting and labeling drum events in polyphonic music. *Proceedings of the 1st Annual Music Information Retrieval Evaluation Exchange*, pages 11–15, 2005.

[11] Christian Uhle, Christian Dittmar, and Thomas Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proc. ICA*, pages 843–847. Citeseer, 2003.

[12] Wikipedia. Zvok — Wikipedia, the free encyclopedia, 2004.

[13] Understanding FFTs and Windowing. Dosegljivo: `http://www.ni.com/white-paper/4844/en/`, 2016. [Dostopano: 1. 8. 2016].

[14] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *ISMIR*, pages 184–191, 2004.