

Optical Music Recognition

Projekat iz predmeta Soft Computing

Profesor: Đorđe Obradović Asistent: Marko Jocić Student: Milan Radeta, RA81/2012 Školska godina: 2015/2016

OMR

- Optical music recognition ili Music OCR predstavlja optičko prepoznavanje muzičkih elemenata sa muzičkih partitura (sheet music)
- Koristi se za pretvaranje muzičkih elemenata u formu koja se može menjati (MusicXML) ili svirati (MIDI)
- OCR teksta prepoznaje reči sekvencijalno s leva na desno, od gore na dole,
- Music OCR uključuje prepoznavanje i paralelnih elemenata, kao što su:
 - note jedne iznad drugih,
 - oznake za dinamiku, tempo,
 - različite deonice predstavljene u jednovremenom toku

– ...

Postojeća rešenja

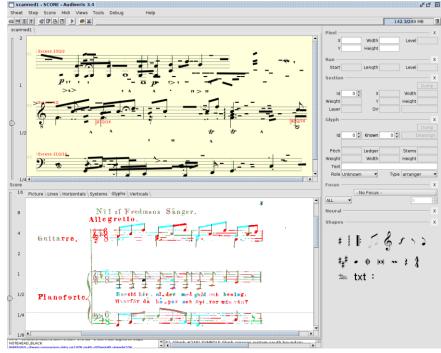
- Rana istraživanja OMR-a su započeta u kasnim šezdesetim na MIT-u i drugim institucijama
- Komercijalan software:
 - SmartScore
 - ranije se zvao MIDISCAN
 - prvo komercijalno rešenje od Musitek-a
 - izašao 1991. za Windows
 - korišćena Lite verzija u Finale-u, programu za pisanje muzičkih partitura
 - Capella Scan (Capella)
 - Vivaldi Scan (Vivaldi Studio)
 - **–** ...
- Besplatna open source rešenja:
 - Audiveris
 - Java
 - GNU General Public License (GPL)
 - import-uje muzičke partiture
 - export-uje MusicXML
 - OpenOMR
 - Java
 - GNU General Public License (GPL)
 - skenira muzičku partituru i odsvira je
- Najteže je napraviti rešenje koje prepoznaje rukom pisanu muzičku partituru.

Postojeća rešenja





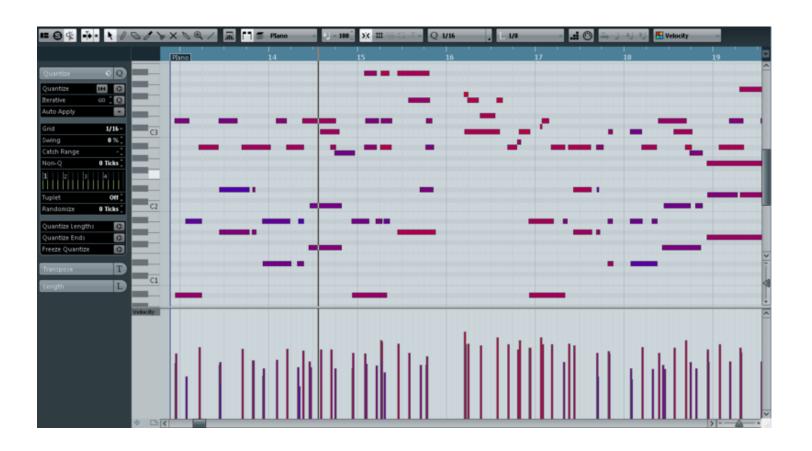




MIDI

- Musical Instrument Digital Interface
- Standard koji opisuje konektore, digitalni interface i protokol između elektronskih muzičkih instrumenata, računara i drugih uređaja
- Prenosi poruke koje imaju specificiraju notaciju, visinu (pitch), glasnoću (velocity), volumen, vibrato, panning, tempo

MIDI



MusicXML

- XML bazirani format koji označava zapadnjačku muzičku notaciju, dokumentovan i slobodan za korišćen pod javnom licencom
- Napravio ga Recordare 2004. preuzeo ga MakeMusic 2011.
- Definiše trajanje i visinu nota, trajanje takta, ključ, tonalitet, itd.
- Koriste ga:
 - programi za pisanje muzičkih partitura
 - Finale, Sibelius...
 - Music OCR programi
 - SmartScore, PhotoScore...
 - music sequencer programi
 - elektronsko pravljenje muzike (MIDI i drugi)
 - Cubase, Logic Pro...
 - HTML5 i JavaScript

MusicXML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <siqn>G</siqn>
          line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```



Representation of middle C on the treble clef created through MusicXML code.

Zadatak projekta

 Napraviti OMR aplikaciju koja će prepoznati note i ostale muzičke elemente sa skenirane, odnosno štampane muzičke partiture i pretočiti ih u MIDI i/ili MusicXML datoteku, korišćenjem Python programskog jezika i OpenCV biblioteke.

Test Dataset

- Korisnički kreirane muzičke partiture u Guitar Pro muzičkom editoru export-ovane kao PNG slike.
- Bez tablatura.
- Samo jednodeonične partiture (za jedan instrument).

Faze projekta

1. Image processing:

- 1. Učitavanje slike
- 2. Grayscaling slike
- 3. Binarizacija slike
- 4. Inverzija slike

2. Prepoznavanje notnih linija

- 1. Horizontalna projekcija slike
- 2. Morfološko otvaranje slike horizontalnim kernelom
- Nalaženje zajedničkih redova pixel-a notnih linija, distance između linija i linijskih sistema

Faze projekta

- 3. Uklanjanje notnih linija
 - 1. Provera *pixel*-a iznad i ispod notne linije
 - Provera debljine linije
- 4. Prepoznavanje objekata u svakom linijskom sistemu (*staff*-u) zasebno
 - 1. Sečenje slike na sliku linijskog sistema
 - 2. Nalaženje objekata:
 - 1. Taktica
 - 2. Ključeva
 - 3. Krajeva (repeticije)
 - 4. Tačke
 - 5. Predznaka
 - 6. Vrste taktova (dužina taktova)
 - 7. Nota sa vratovima (vertikalnih nota)
 - 8. Pauza
 - 9. Celih nota
- 5. Analiza i *export* rezultata

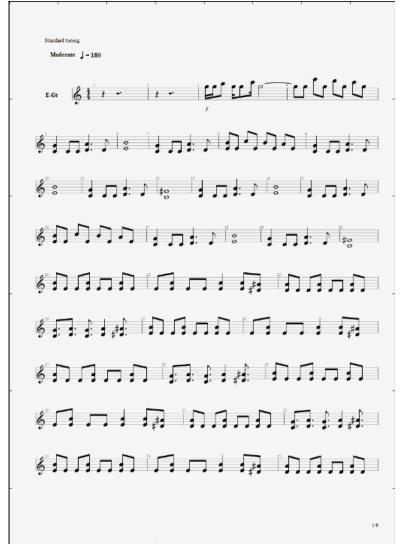
Image processing

- Implementirano putem OpenCV biblioteke.
- Faze:
 - 1. Učitavanje slike
 - 2. Grayscaling slike
 - 3. Binarizacija slike
 - 4. Inverzija slike

Učitavanje slike

- imread(path) funkcija prima putanju do fajla kao parametar path i vraća BGR sliku predstavljenu kao matricu trojki sa vrednostima za B, G i R (redom) komponentu piksela.
- cvtColor(image, conversion) konverzija slike (matrice, image) koja vraća konvertovanu matricu prema prosleđenoj konverziji (conversion)
- Za inicijalno učitavanje slike korišćena je konverzija učitane BGR slike u RBG sliku – COLOR_BGR2RGB, koja vraća matricu trojki sa vrednostima za R, G i B (redom) komponentu piksela

Učitavanje slike

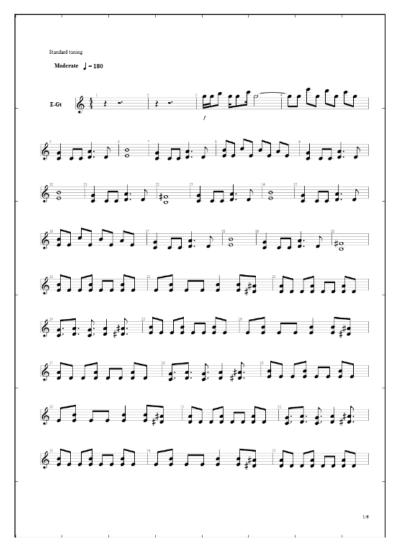


Učitana RGB slika

Grayscaling slike

- cvtColor(image, conversion)
- Za parametar conversion korišćena je konverzija RGB slike u Grayscale (sivu) sliku -COLOR_RGB2GRAY, što vraća matricu brojeva od 0 do 255 (nijansa sive, 0 – crna, 255 – bela)

Grayscaling slike



Grayscale slika

- Binarizacija slike je postupak koji grayscale sliku (matricu brojeva od 0 do 255) pretvara u binarnu, crno-belu sliku (matricu brojeva 0 i 255).
- Testirane su sledeće binarizacije:
 - Globalna binarizacija
 - Otsu binarizacija
 - Adaptivna binarizacija
 - Adaptivna Gausova binarizacija
- Takođe su testirane navedene binarizacije sa zamućenom slikom - GaussianBlur

- threshold(image, threshold, max_value, threshold_type)
- image grayscale slika
- max_value=255
- vraća prag vrednosti i binarizovanu sliku kao matricu brojeva 0 i 255
- Globalna binarizacija
 - threshold_type=THRESH_BINARY
 - za prosleđenu sliku se navede prag vrednosti (threshold) sive (0-255)
 - ukoliko je piksel matrice veći od praga, taj piksel u novoj matrici ima vrednost max_value=255, u suprotnom, ima 0
- Otsu binarizacija
 - threshold_type=THRESH_BINARY+THRESH_OTSU
 - threshold=0, ignoriše se, prag vrednosti se automatski računa
 - ukoliko je piksel matrice veći od praga, taj piksel u novoj matrici ima vrednost max_value=255, u suprotnom, ima 0

- adaptiveThreshold(image, max_value, adaptive_method, threshold_type, block_size, c)
- image grayscale slika
- *max_value=255*
- threshold_type=THRESH_BINARY
- *c* konstanta koja se oduzima od izračunatog praga
- block_size dužina i širina regiona slike
- Adaptivna binarizacija:
 - threshold_type=THRESH_BINARY
 - adaptive_method=ADAPTIVE_THRESH_MEAN_C
 - računa prag vrednosti za regione slike zasebno kao prosečnu vrednost piksela u regionu
- Adaptivna Gausova binarizacija:
 - threshold_type=THRESH_BINARY
 - adaptive method=ADAPTIVE THRESH GAUSSIAN C
 - računa prag vrednosti za regione slike zasebno kao težinsku sumu piksela u regionu, gde su težine iz Gausove raspodele



Nad manjim nezamućenim slikama (samo jednog linijskog sistema) Nema mnogo razlike, osim u jedva primetnim razlikama u debljini notnih linija

Adaptivna binarizacija je rađena sa veličinom bloka od 51 piksela.



ne koristi zamućivanje slika.

prepoznavanje nota. Stoga se u projektu kasnije

Razlike između tipova binarizacije su na maloj ali zamućenoj slici primetnije.

Adaptivne binarizacije

odvajaju note od notnih linije i štete ih više što će otežati

nalaženje notnih linije i

Adaptivna binarizacija je rađena sa veličinom bloka od 51 piksela.

Moderate] - 180

Moderate] - 180 हिं । र प्रस्ता के । र प्रस्ता के स्वाप्त के ו ניטנו ללי בי ביוניטני 흥단 커택 (1) 11 11 11 11 11 11 11 11 11 ו ניטנו וללי ללי לי ניו ניטני לי

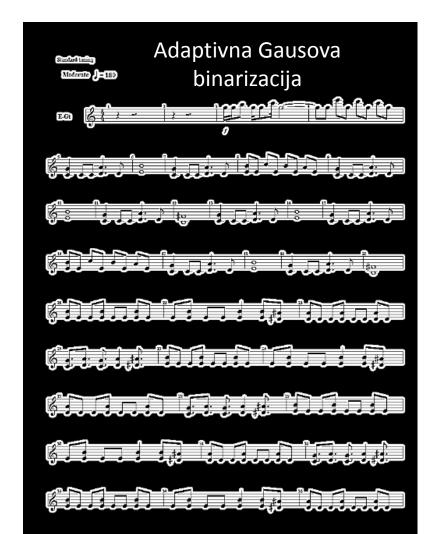
Globalna binarizacija

ול זירון ניני נול דיל לון לוילי לו לו לוילי לו ון בּוֹרָ בוֹ בוֹל בּוֹל רּבּ עוֹל עוֹל עוֹנְיּבּ עוֹן עוֹלְינִינְיּבּ עוֹן עוֹלְינִינִינְיּבּ עוֹנְיִינִינְיִינְיִינְיִינְיִינְיִינְיִ ו ניטונון על ויי נון ניטוניון ו ו בון לות בון תוחות ולון בון בין ו ניטונו בין ויין וויטונו Otsu binarizacija

Izgubljene linije, a možda i drugi pikseli!

Sa istom veličinom bloka (9 piksela), linije nisu izgubljene ali su dobijena dva potpuno različita rezultata. Sa manjim blokom za Gausovu binarizaciju bilo bi poboljšanja.



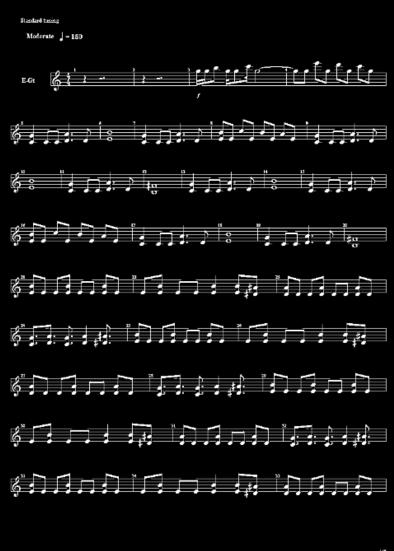


- Adaptivna binarizacija je dala najbolje rezultate.
- Linije nisu izgubljene, iako su prekinute u oblasti nota.
- Manje zavisi od veličine bloka, za razliku od adaptivne Gausove binarizacije.
- Nadalje je korišćena adaptivna binarizacija.

Inverzija slike

- Poslednji korak u procesiranju slike jeste inverzija slike, koja pretvara bele piksele u crne, i obrnuto, odnosno u matrici 0 pretvara u 255, a 255 u 0.
- Ovaj postupak se radi zbog lakšeg prepoznavanja regiona od interesa.

Inverzija slike



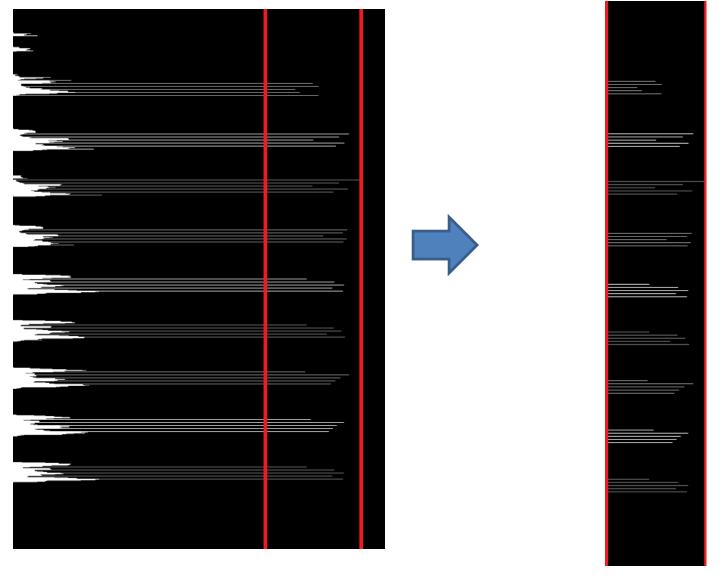
Prepoznavanje notnih linija

- Ova faza je bitna jer se u njoj pamte:
 - pozicije linija
 - koriste se kasnije za prepoznavanje visine tonova
 - distanca između njih (staff spacing)
 - koristi se za prepoznavanje regiona koje su određenih veličina
 - koristi se za proveru regiona red po red, odnosno liniju po liniju, čak i iznad i ispod linijskog sistema
 - distanca između linijskih sistema
 - koristi se za crop-ovanje slike na same linijske sisteme
- Prepoznavanje notnih linija se obavlja kombinovanjem dva pristupa:
 - Horizontalna projekcija slike histogram
 - Morfološko otvaranje slike horizontalnim kernelom

- Horizontal projekcija slike se radi na sledeći način:
 - 1. Za svaki red se računa broj belih piksela ni
 - Napravi se nova matrica koja u svakom redu i ima ni belih piksela s leva na desno (vrednost 255), dok svi ostali pikseli u tom redu su crni (vrednost 0)
- Na ovaj način, dobićemo razdvojene linije, baš tamo u redovima u kojima se nalaze notne linije.



- Pošto se linije na horizontalnoj projekciji razdvajaju tek otprilike trećine slike pa nadalje, najbolje je iseći projekciju tako da obuhvata samo linije.
- Sečenje se obavlja od poslednjeg belog piksela sa desne strane ka levo.
- Takođe se može definisati gde da bude početak (sa leva) isečka.
- Veličina isečka se definiše jedna desetina širine cele slike.



Horizontalno morfološko otvaranje

- Morfološka obrada slike je obilazak slike (svakog njenog piksela) strukturnim elementom (kernel-om) određenog oblika i kreiranje nove izmenjene slike u kojoj su prepoznati oblici.
- Morfološke operacije:
 - Erozija
 - ukoliko kernel pozicioniran na jedan piksel u slici sadrži bar jedan beli piksel, nova slika će imati region u obliku kernela bele boje na tom mestu
 - Dilatacija
 - ukoliko kernel pozicioniran na jedan piksel u slici sadrži samo bele piksele, nova slika će imati region u obliku kernela bele boje na tom mestu

Horizontalno morfološko otvaranje

- Morfološko otvaranje dilatacije erozije slike
- Za prepoznavanje linija, korišćen je horizontalni kernel, visine 1 piksel, širine 20 piksela.

Horizontalno morfološko otvaranje



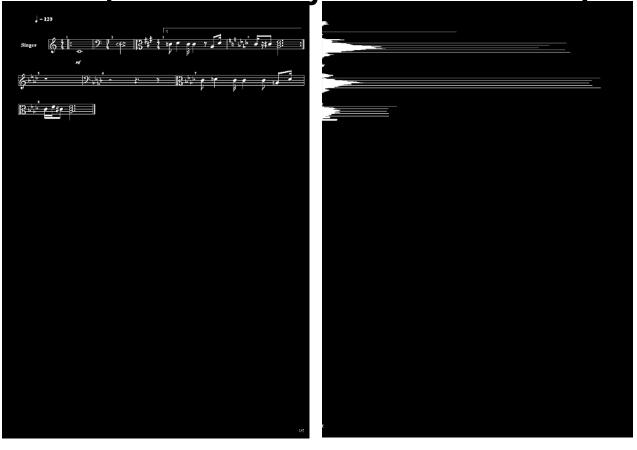
Vide notne linije u linijskom sistemu, kao i pomoćne linije (pomoćnice) izvan linijskog sistema.

Linije su isprekidane i različitih debljina.

Prepoznavanje notnih linija

- Nakon što se dobije isečak horizontalne projekcije i otvorena slika treba proći kroz sliku i zapamtiti redove belih piksela zasebno za svaku sliku.
- Izvršiti presek oba skupa zapamćenih redova.
- Grupisati redove po linijama.
- Izračunati distance između linija.
- Linije grupisati po sistemima (5 linija po sistemu).
 - Linije u sistemu moraju imati distancu između sebe koja ne odstupa od prosečne distance u tom sistemu
 - Ukoliko odstupaju ne pripadaju tom, a možda ni jednom sistemu
 - Može se desiti da je presek zapamćenih linija izbacio neku liniju GREŠKA
 - Može se desiti da se prepozna veoma dugačak kraj repeticije
- Izračunati distance između sistema.

<u>Prepoznavanje notnih linija</u> - problemi



Dugački krajevi repeticija mogu biti prepoznate kao notne linije.

Drugi problem su kratki linijski sistemi. Prvi i poslednji sistem su kraći od drugog

Prepoznavanje notnih linija – rešenja problema

- Prvi problem je rešen tako što se računa distanca između svih linija u sistemu i porede se sa njihovim prosekom.
- Drugi problem potencijalno rešenje je ponovo odraditi horizontalnu projekciju i morfološku projekciju, ali ovaj put pregledati grupe linije, pošto se već zna distanca između sistema i linija.
 - Nije implementirano u trenutnom projektu

Uklanjanje notnih linija

- Linije se uklanjaju kako bi se ostali oblici izdvojili i prepoznali.
- Potencijalno oštećuje ostale objekte, ukoliko se ne radi pažljivo.
- Alternativa uklanjanju notnih linija je ignorisanje notnih linija.
 - Ne oštećuje objekte prilikom segmentacije
- Postoje 3 načina za uklanjanje notnih linija:
 - provera piksela ispod i iznad linije
 - provera debljine linije po segmentima
 - kombinacija prethodna dva pristupa

Top-Bottom Pixel Removal

- Za svaku kolonu slike, za svaku liniju proveriti njene redove – ukoliko se nalazi bar jedan beli piksel, pretpostaviti da se tu nalazi linija.
- Iznad najgornjeg piksela te linije u toj koloni, i ispod najdonjeg piksela te linije, proveriti po 2 piksela – ukoliko se nalazi bar jedan beli piksel, ne treba brisati liniju (njene piksele) u toj koloni, u suprotnom treba.

Top-Bottom Pixel Removal

ותתתתו בן ות הוותתתתן וּבּוּנוּנוּנוֹתתתונוֹ בּבּוּנת בת בוֹשׁ ותתתתו בנות הותתתתתו

Thickness Based Removal

- Za svaku kolonu slike, proći od najgornjeg piksela najgornje linije do najdonjeg piksela najdonje linije.
- Svaki put kad se naiđe na crni piksel sabrati broj belih piksela do prethodnog crnog piksela – debljina "linije".
- Ukoliko je debljina linije manja ili jednaka prosečnoj debljini linije (+ mala tolerancija) izbrisati je.

Thickness Based Removal

Moderate J-180 المالم المالي ול דְּעדוֹ נְנַעַרְעַנַוֹּן דְּעדוֹן: יוֹלַעַרַעַנַוּן בּיוֹן עוֹלַעַרָּוֹן ול דְּעַדְיּ בֵּוֹ בְּעַדְיּ בְּיִּ בְּעַדְיּ בִּי בְּיִ בְּעַבְיּ בִּי בְּיִ בְּיִי בְּיִ בְּיִ בְּיִ ונתחתו בן ות חותחחת: War I Hay INGLA IN TO 1 ותתחתו בן ותחותתחתו

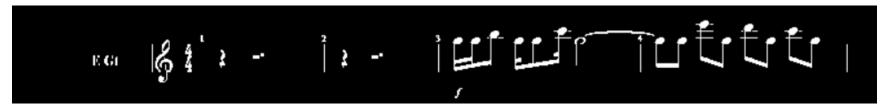
Kombinacija oba pristupa

Moderate _ - 180 ונתתתו בן ות הוותתחת ש FIFE TOTALE TO THE

Prepoznavanje objekata

- Pošto su linije uklonjenje, treba preći na prepoznavanje objekata.
- Najbolje je raditi zasebno za svaki linijski sistem.
 - Može i globalno, ali će biti teže prepoznati objekte između sistema i na koji sistem utiču
- Treba obuhvatiti samo jedan linijski sistem.
 - Za svaki linijski sistem obuhvatiti redove cele slike, počev od reda koji je udaljen od prve linije tog sistema za pola distance između sistema do reda koji je udaljen od poslednje linije tog sistema za isto toliko
 - start_row = first_line staff_distance
 - end_row = last_line + staff_distance
 - Koristiti taj isečak slike za prepoznanje objekata
 - Koristiće se pozicije lokalne u okviru tog sistema

Prepoznavanje objekata



Prvi linijski sistem, isečen iz cele slike

Taktovi (taktice)

- Da bi se prepoznali taktovi trebalo bi prvo izvršiti vertikalno morfološko otvaranje isečka sistema, sa vertikalnim kernelom koji je širok 1 piksel, a visok koliko i prosečna distanca između linija pomnožena sa 1.5.
- Ova operacija će vratiti sliku sa isključivo vertikalnim linijama, neke manje, neke veće.

 Potrebno je vertikalne linije zapamtiti kao regije, odnosno nizove nizova koordinata.

Traženje regija

- find_regions(image, ref_image, pixel_span, eight_way)
- Vraća listu regija.
- Regija je lista koordinata.
- Za svaki beli piksel u referencirajućoj slici (ref_image) traže se susedni pikseli u slici image i dodaju se u regiju u kojoj se nalazi startni piksel.
- Dodati beli pikseli zatim "traže" svoje susedne bele piksele i dodaju ih u istu regiju, ukoliko se već ne nalaze u toj ili nekoj drugoj regiji.
- Postupak se obavlja sve dok nema više belih piksela u okolini.
- Veličina okoline piksela je određena parametrom pixel_span, koja označava koliko maksimalno susedni piksel može biti daleko.
- Parametar eight_way označava da li se susedni pikseli traže samo u 4 smera (False: gore, dole, levo, desno) ili u 8 (True: gore, dole, levo, desno, gore-levo, dole-desno...)
- Ukoliko *ref_image* nije zadat, *image* je sam sebi referencirajuća slika.

Taktovi (taktice)

- Da bi se našli taktovi potrebno je prvo naći sve vertikalne linije iz slike vertikalnih linije i sačuvati ih kao regije.
- Potom je potrebno naći sve vertikalne objekte iz isečka sistema koji sadrže te vertikalne linije.
- Taktovi su one vertikalni objekti koji su same vertikalne linije visoke koliko i sam linijski sistem.
 - U ovom slučaju se debljina ne gleda.
 - Dodatno se može odraditi provera pozicije vertikalne linije da li je od najgornje do najdonje linije linijskog sistema
- Nakon što su taktovi pronađeni, sačuvati ih i izbaciti ih iz isečka
 - Za svaku koordinatu svake regije (svih taktova) obojiti isečak na toj koordinati u crno

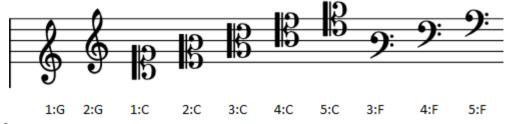
Ključevi

- Postoje tri vrste ključa:
 - G-ključF-ključ

 - C-ključ
- Ključevi određuju visinu tona, zajedno sa predznacima i notnim glavama, u zavisnosti od toga na kojoj liniji stoji ključ i kog je tipa.

Ključevi

 Vertikalne pozicije ključeva:



• Format – notna linija po redu od dole : ton na

toj liniji

• Tri različita ključa sa istim tonom C

 U ovom projektu samo se prepoznaju tipovi ključa, ne gledaju se pozicije ali se mogu izračunati.

Template matching

- Jedna od metoda koja se koristi u OMR-u je template matching (poređenje pomoću obrazaca) koja se u ovom projektu koristi.
- Druge metode koriste i metode projekcije, morfološke, neuronske mreže, geometrijske osobine i momente.
- Faza prepoznavanja (klasifikacije) je generalno najteža faza u OMR-u.

Template matching

- Template matching, odnosno poređenje sa obrascima, podrazumeva poređenje slike regije sa template slikama, piksel po piksel.
- Ukoliko je template slika manja od slike regije koristi se sliding window tehnika kroz sliku regije i nalazi se oblast sa najvećim procentom prepoznanosti jedne od template slika.
- U ovom projektu se koriste obrasci za predznake, ključeve, notne glave, krajeve, zastavice, pauze i vrste taktova.
- Obrasci se učitaju, resize-uju ukoliko je potrebno, pretvore u grayscale sliku, binarizuju pomoću Otsu binarizacije i invertuju.

Ključevi

- Za nalaženje ključeva uzimamo vertikalne regije (regije koje sadrže vertikalne linije) koje se nalaze sa desne strane svakog takta, dovoljno blizu (otprilike 2 puta distanca između linija).
- Ukoliko su regije veće od pola visine takta porede se sa obrascima koji se resize-uju na veličinu regije.
- Ukoliko je obrazac koji najviše odgovara regiji 75% ili više sličan njoj, regija je prepoznata kao taj obrazac, pamti se i izbacuje iz slike.

Ključevi - obrasci

188818

C – ključ

9:99:99:9

F – ključ

Postoje obrasci za
C-ključ koje sadrže
vertikalne linije i koje
ne sadrže, u slučaju da
se regija prepozna sa ili
bez njih.

Postoje obrasci za F-ključ koji sadrže i koji ne sadrže tačke, u slučaju da se regija prepozna sa ili bez njih.



G – ključ

Ključevi



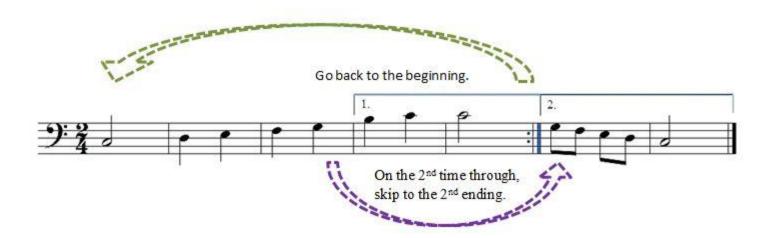
Regija koja se proverava



Prepoznat obrazac

Krajevi

 Krajevi repeticije označavaju koji taktovi se sviraju u kom ponavljanju deonice.



Krajevi

 Za krajeve se traže regije koje se nalaze iznad najgornje notne linije i porede se sa obrascem kraja.

- Ukoliko se prepozna, čuva se i izbacuje iz slike
- U okviru ovog projekta nije implementirano prepoznavanje brojeva (osim vrste takta), pa time ni oznake kraja.

Tačke

- Tačke se nalaze:
 - Sa desne strane note ili pauze da produže njihovo trajanje za pola trenutnog trajanja
 - Sa desne strane bas ključa
 - Sa desne ili leve strane takta da označe ponavljanje (repeticiju)
 - Stakato (staccato, skraćivanje dužine trajanja note)
 - Nije implementirano ali se može izračunati prema njenoj poziciji iznad ili ispod notne glave, slično prvoj vrsti tačaka

Tačke

- Da bi se pronašle tačke potrebno je pronaći regione na slici pomoću pixel_span-a od 1, i to samo u 4 smera.
- Sačuvati regije čija je veličina manja ili jednaka polovini distance između linija (+ tolerancija)
- Opciono: učitati obrasce tačaka i resize-ovati ih na veličinu regije i izvršiti poređenje.

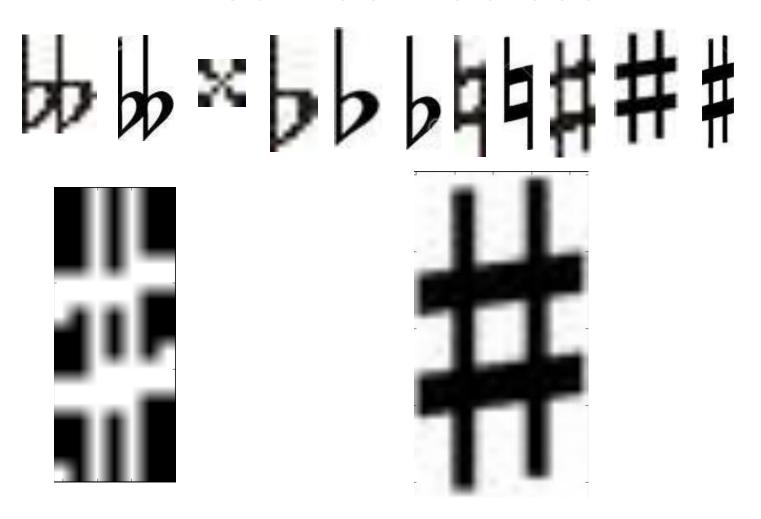
- Predznaci menjaju visinu tona (ne visinu note).
- Predznaci se mogu nalaziti:
 - Na početku takta, posle ključa i pre vrste takta (ukoliko postoje)
 - Označavaju tonalitet.
 - Utiču na sve note koje se nalaze na istim vertikalnim pozicijama.
 - Ispred notnih glava (sa leve strane)
 - Utiču na susednu notu i sve njene pojave u tom taktu.
 - Sve prethodne promene visine tona nad tom notom prestaju da važe.

- Vrste predznaka:
 - Povisilica #
 - Povisuje ton za polustepen (sledeća desna, crna ili bela dirka na klaviru, sledeći prag na gitari)
 - Snizilica
 - Snižava ton za polustepen (sledeća leva, crna ili bela dirka na klaviru, prethodni prag na gitari)
 - Razrešilica
 - Ukoliko je nota bila povišena ili snižena, razrešena je i vraća se na "osnovni" ton
 - Dupla povisilica ×
 - Povisuje ton za 2 polustepena iliti ceo stepen
 - - Snižava ton za 2 polustepena iliti ceo stepen

- Za prepoznavanje predznaka se ponovo otvara isečak sistema, ovaj put vertikalnim kernelom visine 3.5 * distanca između linija.
- Nalaze se regije koje sadrže pronađene vertikalne linije, sa pixel_span-om od 2 piksela, u 8 smerova.
- Potom se ponovo vrši otvaranje isečka sistema, sa vertikalnim kernelom od 1.5 * distanca između linija.
- Nalaze se regije koje sadrži vertikalne linije, ali nisu bile pronađene pre toga, sa pixel_span-om od 1 piksel, u 4 smera.

- Svaka pronađena regija se zatim poredi sa obrascima predznaka, koji se resize-uju na veličinu regiju koju porede.
- Ukoliko najbolji pronađen obrazac se 70% ili više podudara sa regijom, regija se pamti i uklanja iz slike isečka.
- S obzirom da dupla povisilica nije vertikalni objekat (ne sadrži vertikalne linije), u ovom projektu njeno pronalaženje nije implementirano
 - Potencijalno rešenje bi bilo da u toku pronalaženja tačaka se izvrši poređenje sa obrascem duple povisilice

Predznaci - obrasci



Regija

Prepoznat obrazac

- Vrste taktova određuju dužinu takta tročetvrtinski, četvoročetvrtinski, šestosminski...
- Predstavljaju se pomoću razlomka 2 broja
- Gornji broj (deljenik) predstavlja koliko udaraca (beat-ova) ima u jednom taktu.
- Donji broj (delilac) predstavlja koliko svaki udarac (beat) traje u jednom taktu.
- U datom primeru 4/4, znači da u jednom taktu mora biti 4 udarca, koji traju po četvrtinu note, odnosno jedan takt traje 4 četvrtine.

- Takt od 4 četvrtine se takođe može predstaviti simbolom C
- Takt od 2 polovine se može predstaviti simbolom •
- Vrsta takta određuje dužinu taktova odatle pa do sledeće pojave vrste takta.

- Potencijalne regije se traže pomoću vertikalnih linija sa pixel_span-om od 2 piksela, u 8 smerova.
- Vrste taktova se traže među regijama koje su najbliže desnim stranama taktova ili ključeva.
- Čak ni predznaci ne predstavljaju problem (ukoliko su svi pronađeni) zato što su otklonjeni sa slike.
- Ukoliko je regija unutar linijskog sistema vrše se dalje provere nad njom.

- Ukoliko je regija visoka koliko i sam linijski sistem ili ukoliko obuhvata sredinu linijskog sistema to znači da je vrsta takta prepoznata kao jedna regija - regija se deli na dve podregije u tom slučaju.
- U suprotnom, pronađena je jedna od dve regije i traži se druga sa sličnim dimenzijama.
- Kad se pronađe i druga regija vrši se poređenje obrazaca obe regije, sa obrascima koji sadrže broje, resize-ovani na veličinu regija.
- Ukoliko obe regije su prepoznati kao brojevi, vrsta takta je predstavljena razlomkom.
- U suprotnom, vrši se ponovo poređenje ali oba regiona kao jedan.
- Ukoliko se prepozna kao broj, a ne kao znak, to je greška.
- Prepoznata vrsta takta se pamti i izbacuje iz slike.

Vrste taktova – problemi

- Moguć problem, osim razdvojenih regija, su i dvocifreni brojevi, čije cifre su razdvojene.
- U ovom projektu druga cifra se neće uzimati u obzir.
- Potencijalno rešenje je provera horizontalnih pozicija prepoznatih regiona.

Vertikalne note

- Vertikalne note su objekti za koje je najviše potrebna analiza regije, kako bi se mogle prepoznati:
 - Same note
 - Spojene note
 - Dužine trajanje nota
 - Pozicija note (visina tona)

Vertikalne note

- Traže se regioni na slici koji sadrže vertikalne linije, sa pixel_span-om od 4 piksela, u 8 smerova.
- Razlog velikog pixel_span-a je to što notne glave polovine mogu biti dosta oštećene nakon odstranjivanja notnih linija.
- Potom je potrebno učitati obrasce punih notnih glava u jednu grupu i praznih notnih glava u drugu grupu.
- Obrasci bi trebalo da budu resize-ovani na veličinu jednakoj distanci između linija.

Vertikalne note

- Potom je potrebno za svaki region učitati ga kao sliku.
- Sliku regiona je potrebno morfološki otvoriti vertikalnim kernelom visine 3 * distanca između linija.
- Zapamtiti regije vertikalnih linije iz morfološki otvorene slike i ukloniti ih iz slike regije.
- Analiza regije koja predstavlja vertikalnu notu obuhvata:
 - Nalaženje notnih rebara
 - Nalaženje notnih zastavica
 - Nalaženje notnih glava

Notna rebra

- Sledeći korak je pronalaženje notnih rebara.
- Cela notna rebra spajaju notne vratove (vertikalne linije).
- Polovična notna rebra se nalaze uz samo jedan vrat.
- Notna rebra, kao i notne zastavice određuju dužinu trajanja note – svako notno rebro polovi četvrtinu note.





2 puna notna rebra



1 puno notno rebro koje povezuje tri vrata i 1 polovično koje je uz desni vrat

Notna rebra

- Potrebno je pronaći podregije unutar slike regije bez vratova, pixel_span=1, u 8 smerova.
- Za svaku podregiju potrebno je naći koje sve vertikalne linije povezuje.
- Ukoliko povezuje više vratova, znači da je podregija puno notno rebro.
- Ukoliko je povezana sa samo jednim vratom znači da je jedno od sledećeg:
 - Pola notnog rebra
 - Zastavica
 - Notna glava
- Ukoliko nije povezana ni sa jednim vratom, regija je verovatno obuhvatila suvišne bele piksele.

Notna rebra

 Među preostalim neprepoznatim podregionima koji su povezani uz jedan vrat traže se oni čiji vrat obuhvata i neko puno rebro – taj podregion predstavlja pola notnog rebra.

Notne zastavice

- Notne zastavice, kao i notna rebra određuju dužinu trajanja note – svaka notna zastavica polovi četvrtinu note.
- Notna zastavica se nalazi isključivo uz jedan vrat.
- Među preostalim podregionima se vrši template matching sa obrascima zastavica resize-ovani na veličinu podregiona.
- Obrasci sadrže jednostruke, dvostruke i višestruke zastavice.

Notne zastavice

 Zastavice mogu biti pronađene kao zasebni regiona ili kao jedan veći region, u zavisnosti od izvršene segmentacija, stoga je dobro imati različite vrsta zastavica i izračunati koliko zastavica određenog tipa se nalazi uz jedan vrat.

Notne glave

- Među poslednjim preostalim neprepoznatim regionima se traže regioni notnih glava.
- No, regioni se ne porede direktno, jer se tako gubi mogućnost nalaženja više notnih glava, koje se nalaze jedna na drugoj.
- Za nalaženje više notnih glava koristi se vertikalna sliding window tehnika, koja se pomera za pola distance između linija.

Notne glave

- Prozor se prvo postavi na poziciju prve notne linije, odnosno tako da obuhvata prvu prazninu.
- Potom se pomera vertikalno do one linije ili praznine koja obuhvata najviši piksel regiona koji se prepoznaje i analizira – odavde počinje prepoznavanje i klizanje prozora.
- Pri pomeranju se pamti broj pomeraja, kako bi se zapamtilo na kojoj visini se nalazi notna glava, jer se po tome prepoznaje visina tona.

Notne glave

- Pregledaju se podpodregioni podregiona, odnosno onaj deo podregiona koje obuhvata klizeći prozor.
- Svaki deo podregiona se poredi sa obrascima punih notnih glava
 - Lakši su za prepoznati od praznih notnih glava, stoga je prag sličnosti 80%
- Ukoliko podregion nije puna notna glava, poredi se sa obrascima praznih notnih glava
 - Prag sličnosti 70%
- Za prepoznatu notnu glavu se pamti i njena vertikalna pozicija, odnosno na kojoj liniji ili u kojoj praznini se nalazi.

Vertikalne note

- Nakon što su pronađena rebra, zastavice i notne glave, treba zaključiti sledeće stvari:
 - Broj nota u regiji
 - Dužina nota u regiji
- Za svaku notnu glavu treba proveriti koliko se na njenom vratu nalazi:
 - Punih rebara
 - Polovičnih rebara
 - Zastavica
- U zavisnosti od navedenog se računa dužina trajanja note
 - Svako rebro, polovično rebro, zastavica dodatno polovi četvrtinu
- Ukoliko ne postoji nijedno od navedenog, za dužinu se oslanjamo na tip notne glave
 - Puna notna glava četvrtina
 - Prazna notna glava polovina
- Za svaku notu se pamti njena pozicija (vertikalna i horizontalna) i dužina i uklanja se cela regija iz isečka sistema.

Vertikalne note - problemi

- Ukoliko postoje podregioni prepoznati kao zastavice ali nisu iskorišćeni prilikom prepoznavanja svojstava nota za svaku notnu glavu, najverovatnije su to regioni koji su zapravo prazne notne glave.
- Ovaj problem se rešava tako što se ponovo vrši prepoznavanje notnih glava sa tim podregionima, ali samo praznih notnih glava.
- Za svaku prepoznatu notnu glavu pamti se polovina note, sa njenom pozicijom i dužinom.

Pauze

- Pauze označavaju mesta gde nastupa tišina u muzici određene dužine trajanja.
- Mogu biti produžene pomoću tačke.
- Vrste pauza:



Pauze

- Traže se regioni iz isečka sa *pixel_span=2*, u 8 smerova.
- Za svaki pronađeni region proverava se da li se nalazi u linijskom sistemu.
- Ako jeste, poredi se obrascima četvrtinske pauze
 - Teža za poređenje
 - Prag sličnosti 68%
- Ukoliko nije četvrtinska pauza, porediti je sa ostalim
 - Prag sličnosti 70%
- Svi obrasci su resize-ovani na veličinu regije.
- Pronađene pauze se pamte i brišu iz isečka sistema.

Pauze - problem

- Oštećena četvrtinska pauza koja se teže prepoznaje.
 - Porediti region prvo sa četvrtinskom pauzom sa manjim pragom vrednosti
 - Povećati po potrebi pixel_span pri nalaženju regiona
- Cela i polovinska pauza isto izgledaju
 - Ali se nalaze na različitim linijama
 - Cela se nalazi ispod druge linije (od gore)
 - Polovinska se nalazi na trećoj liniji (od gore)

Cele note

- Na kraju se prepoznaju, odnosno porede se sa celim notama, preostali neprepoznati regioni koji se nalaze između taktica.
- Traže se regioni između taktica sa pixel_span=3, u 8 smerova.
- Učitavaju se obrasci praznih notnih glava za cele note i resize-uju na veličinu distance između linija.
- Koristi se ista sliding window tehnika kao i kod vertikalnih nota.

- Nakon prepoznavanja i klasifikacije pronađenih regija treba izvršiti analizu tih regija i export u MusicXML.
- U ovom projektu nije obuhvaćen rad sa *MusicXML*-om, ali su podaci exportovani u vidu *custom* tekstualne notacije.

- Prvo se ispisuje redni broj taktice.
 - Bar Line #
- Potom se ispisuje tip ključa koji se nalazi odmah pored taktice.
 - G-Clef
 - F-Clef
 - C-Clef
- Vrsta takta koja se nalazi najbliža taktici se zatim ispisuje:
 - Time: #/#

- Kraj repeticije se ispisuje ukoliko počinje od trenutnog takta
 - Ending
- Sortiraju se preostali elementi u taktu (note, tačke, pauze i predznaci) po kolonama.
- Traže se predznaci koji se nalaze na početku takta, tako što se uzimaju oni koji se pojavljuju pre prvih nota i pauza u tom taktu i čija distanca do prve note je veća od distance između linija:
 - Key Accidentals:
 - natural
 - sharp
 - flat
 - Visina u suštini nije bitna jer se predznaci na početku takta ispisuju određenim redosledom:
 - povisilice: fis, cis, gis, dis, ais, eis, his
 - snizilice: be, es, as, des, ges, ces, fes

- Note u taktu se ispisuju nakon predznaka:
 - Notes:
 - stem note ili whole note
 - column: #
 - » prva kolona regiona
 - height: #
 - » visina data kao odstupanje od prve notne linije od gore
 - » 0.0 prva notna linija
 - » 0.5 prva praznina
 - » 1.0 druga notna linija
 - » -1 prva pomoćnica iznad prve notne linije
 - » 5 prva pomoćnica ispod poslednje notne linije
 - duration: #
 - prolonged duration
 - » pronađena je tačka pored notne glave
 - accidental: sharp | flat | natural | double flat | double sharp
 - » pronađen je predznak ispred notne glave

- Detaljnijom analizom položaja (kolone) note se može zaključiti koje note su zapravo u sklopu akorda.
- Sa definisanim ključem, predznacima i visinom notne glave se može odrediti pitch (visina) tona.
- Sa vrstom takta, dužinom note i informacijom o produženju note se može izračunati da li je takt validan (da li je u skladu sa njegovom vrstom).

- Pauze:
 - Rests:
 - rest
 - column: #
 - duration: #
 - prolonged duration
 - Polovinsku pauzu proverava da li je zapravo cela na osnovu pozicije.
- Repeticije se ispisuju, ukoliko su pronađene tačke blizu početne ili krajnje taktice
 - Repeat Begin
 - Repeat End



					P . P .			
Analysis results of staff 1			Key Accidentals:			Key Accidental	s:	
Bar Line 1	.CS OF SCALE I		key Accidenta	sharp			natural	
G-Clef				sharp			natural	
Time: 4/4				sharp			natural	
Bar Line 2			Notes:	Silairp			flat	
Bar Line 2			Notes.	stem note			flat	
Notes:				Stem note	column: 425		flat	
Notes.	whole note						flat	
	whote hote	column: 197			height: 2.0 duration: 0.125	Notes:		
					accidental: natural		stem note	
		height: 5.0			accidental: natural			column: 634
Damast Damin		duration: 1		stem note	column: 443			height: 2.0
Repeat Begin								duration: 0.0625
Bar Line 4					height: 1.5		stem note	
Bar Line 5				-4	duration: 0.25			column: 668
F-Clef				stem note	1 460			height: 2.0
Time: 3/4					column: 468			duration: 0.125
Notes:					height: 2.0			accidental: sharp
	stem note				duration: 0.0625		stem note	•
		column: 303		stem note	1 400			column: 690
		height: 1.5			column: 482			height: 1.0
		duration: 0.5			height: 2.0			duration: 0.5
		prolonged duration			duration: 0.25			prolonged duration
	stem note			stem note			stem note	, g
		column: 303			column: 527			column: 690
		height: 2.5			height: 3.0			height: 2.0
		duration: 0.5			duration: 0.125			duration: 0.5
		prolonged duration		stem note	_			prolonged duration
Rests:					column: 545		stem note	Processor and account
	rest	_			height: 1.5			column: 690
		column: 296			duration: 0.0625			height: 3.0
		duration: 0.5	Rests:					duration: 0.5
		prolonged duration		rest				prolonged duration
Bar Line 6					column: 511	Rests:		p. 310118ca aai ac1011
Bar Line 7					duration: 0.125		rest	
C-Clef			Bar Line 8				. 232	column: 652
Time: 4/4			Bar Line 9					duration: 0.0625
Ending						Repeat End		dc10 0.0023
						Bar Line 10		

Bar Line 11

Test Dataset

- Korisnički kreirane muzičke partiture u Guitar Pro muzičkom editoru export-ovane kao PNG slike.
- Bez tablatura.
- Samo jednodeonične partiture u jednom vremenskom toku.

Obrasci

- Slike sa Google Images:
 - Ručno sečene
- Predznaci, ključevi, krajevi, zastavice, notne glave, pauze, vrste taktova

- Nakon testiranja sa više partitura zaključeno je sledeće:
 - Uvek prepozna notne linije
 - Osim poslednjeg linijskog sistema ako je kratak
 - Prepoznaju se uglavnom sve vertikalne note
 - Prepoznaju se svi ključevi i vrste taktova
 - Prepoznaju se uglavnom sve taktice
 - Retko se desi problem da se ne prepoznaju usled toga što redosled koordinata kod regiona vertikalnih linija i taktica nije isti ili se veličine razlikuju za mali broj piksela
 - Rešenje:
 - » Provera pozicija regiona
 - » Provera oblika regiona
 - » Template matching regiona sa obrascem taktice
 - » Sortiranje koordinata
 - » Poređenje broja koordinata

- Nakon testiranja sa više partitura zaključeno je sledeće:
 - Polovine su često dupliraju
 - Provera notnih glava nade 2 prazne notne glave (jednu ispod druge) umesto jedne notne glave
 - Pomoćnice kod notne glave se uglavnom pomešaju sa tačkama s toga produže notu.
 - Pomoćnice koje se ne nalaze kod notne glave ponekad se interpretiraju kao polurebra i dodatno skrate vreme trajanja note
 - Razrešilica se u retkim slučajevima prepozna kao povisilica

- Nakon testiranja sa više partitura zaključeno je sledeće:
 - Cele note se često prepoznaju ako nisu veoma oštećene nakon segmentacije
 - Pauze se uglavnom dobro prepoznaju
 - Čak i kad se nalaze između nota koje su povezane
 - Predznak ispred note ukoliko je previše blizu takta se prepozna kao predznak koji važi za ceo takt.
 - Pojava elemenata čije prepoznavanje nije implementirano poremeti prepoznavanje ostalih elemenata
 - Lukovi, dinamičke oznake, tekstovi, brojevi

- Nakon testiranja sa više partitura zaključeno je sledeće:
 - Sa različitim distancama između sistema napravi se loš isečak širokih sistema
 - Rešenje: uzeti isečak koristeći distance od trenutnog sistema do sledećeg i do prethodnog

Problemi

- Problemi navedeni u rezultatima testiranja.
- Problemi navedeni u fazama prepoznavanja linija i objekata.
- Problemi navedeni u fazi binarizacije.
- Testiranje sa drugim dataset-ovima
 - Bezuspešno
 - Razlozi
 - Potrebne drugačije vrednosti parametara
 - Potreban drugačiji redosled nalaženja objekata
 - Potrebna drugačija binarizacija
 - Drugačiji stil pisanja, zahteva drugačiju logiku nalaženja objekata
 - Npr. taktica se ne piše na početku linijskog sistema.

Dalji rad

- Vršenje ispravki pomenutih problema
- Refaktorisanje, optimizacija i poboljšanje korišćenih algoritama.
- Dopunjenje trenutne dokumentacije.
- Export u MusicXML.
- Pravljenje GUI aplikacije pomoću koje će moći da se:
 - Otvaraju slike
 - Prikazuju rezultati podfaza i konačan rezultat
 - Podešavaju parametri određenih procesa
 - Direktno edituje konačan rezultat
 - Export-uje konačan rezultat
 - Pušta MIDI melodiju na osnovu konačnog rezultata.

Dalji rad

- Proširenje algoritma:
 - Da prepozna i razlikuje više deonica u istom vremenskom toku
 - Da prepozna ostale muzičke elemente
 - Lukovi
 - Dinamičke oznake
 - Propratni tekstovi i brojevi
 - Tempo
 - Itd.
 - Rotacija slike
 - Ukoliko je zaokrenuta
 - Ispravljanje slike
 - Ukoliko je savijena

Dalji rad

- Proširenje algoritma:
 - Da radi sa ručno pisanim notama
 - Da radi sa drugim metodama i pristupima
 - Neuronskim mrežama
 - Projekcije
 - Morfološke operacije
 - •

Izvori informacija

- Wikipedia
 - Music OCR
 - https://en.wikipedia.org/wiki/Music OCR
 - Audiveris
 - https://en.wikipedia.org/wiki/Audiveris
 - OpenOMR
 - https://en.wikipedia.org/wiki/OpenOMR
 - MIDI
 - https://en.wikipedia.org/wiki/MIDI
 - MusicXML
 - https://en.wikipedia.org/wiki/MusicXML
- Audiveris
 - https://audiveris.kenai.com/
- OpenOMR (Sourceforge)
 - http://sourceforge.net/projects/openomr/
- Music Theory
 - http://www.musictheory.net/
- Music Reading Savant
 - http://www.musicreadingsavant.com/

Izvori informacija

- Osnovna teorija muzike, Marko Tajčević
- Optical Music Recognition, Linn Saxrud Johansen, University of Oslo
- The Challenge of Optical Music Recognition, David Bainbridge & Tim Bell, University of Waikato, Hamilton, New Zealand
- Optical Music Recognition, Live camera recognition of handwritten musical notes Barak Ben-Dayan, Ilai Giloh, University of Negev, Israel
- Recognizing Musical Notation Using Artificial Neural Networks, Pascal Attwenger, University of Vienna
- Automatic Music Score Recognition System Using Digital Image Processing, Yuan-Hsiang Chang, Zhong-Xian Peng, Li-Der Jang
- Adaptive Optical Music Recognition, Ichiro Fujinaga, Faculty of Music, McGill University, Montreal, Canada
- Optical Music Recognition CS 194-26 Final Project Report, Andy Zeng
- Optical Music Recognition for structural information from high-quality scanned music, Søren Bjerregaard Vrist, University of Copenhagen
- Optical music recognition: state-of-the-art and open issues, Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, Jaime S. Cardoso
- Introduction to Optical Music Recognition: Overview and Practical Challenges, Jir Novotny and Jaroslav Pokorny, Charles University, Prague, Czech Republic
- Interpreting the semantics of music notation using an extensible and object-oriented system,
 Michael Droettboom, Ichiro Fujinaga, Johns Hopkins University, Baltimore, MD