

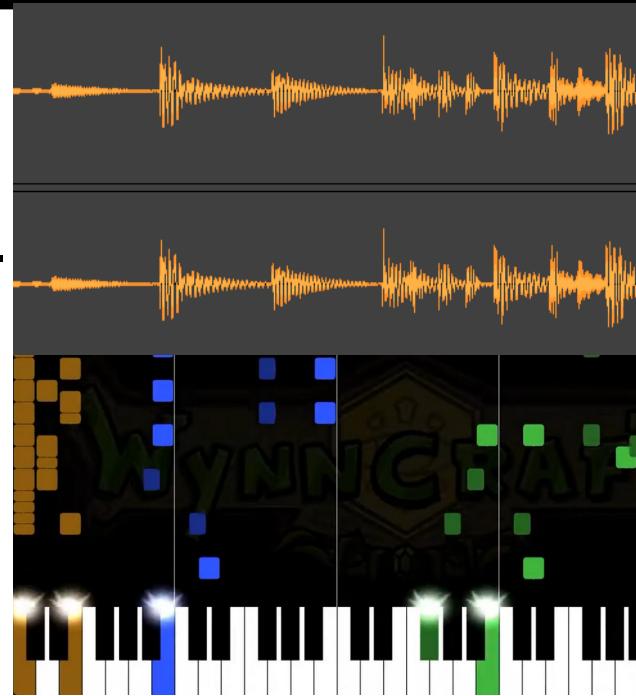
Note Block hangminták felismerése zenefájlban

Vértesaljai Bálint
Konzulens: Grad-Gyenge László György
Mérnökinformatikus BSc, Szakdolgozat



Hangfájlok formátuma

- Wave: wav/mp3/ogg/...
- Számértékek másodpercenként 44100-szor pl.
- Midi: ~kotta
- Zenehang kezdet- és végidőpontok
- Konverzió nem egyértelmű, nem könnyű (főleg wave → midi)



Note Block Studio

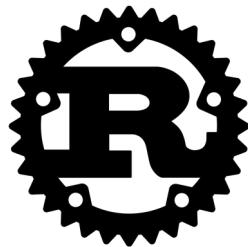


- Minecraft Note Block-ok mint minták
- Note Block Studio: szerkesztőprogram
- NBS fájlformátum
- ~Midi, de speciálisabb: egy időpillanat, konkrét minták



Projekt célja

- Bemenet: Wave fájl
- Hangminták felismerése adott időben, hangmagasságban, hangerővel
- Kimenet: Nbs fájl
- Hasonlít a zene lekottázásra, annak speciális esete
- Megvalósítás Rust-ban



Időbeosztás

- Előző félév (Önálló laboratórium):
 - Működő konverzió, két zenén tesztelve
 - Bonyolult akkordok esetén elég rossz felismerés
- Ez a félév (Szakdolgozat-készítés):
 - Gradiens a költségfüggvényhez
 - Sokkal pontosabb felismerés
 - Nagy adathalmazon tesztelés, elemzés



Video

- Note Block hangminták
- Hullámos fájl (bemenet)
- Spektrogram (konverzióhoz fontos)
- Midi vizualizáció (motiváció része)
- Lejátszás a játékban (motiváció része)
- Eredeti Nbs fájl (elvárt kimenet)
- Felismert Nbs fájl (tényleges kimenet)



File Edit Settings Help

Fully compatible

00:00:02:556 / 00:00:06:400 5.00 t/s 1.4.1

00:00:02:000 00:00:03:000 00:00:04:000 00:00:05:000 00:00:06:000 00:00:07:000

8 12 16 20 24 28 32 36

Harp Bass Snare Click

C 6 C 6 C 6 C 6

Note Block hangminták
Program: Open Note Block Studio

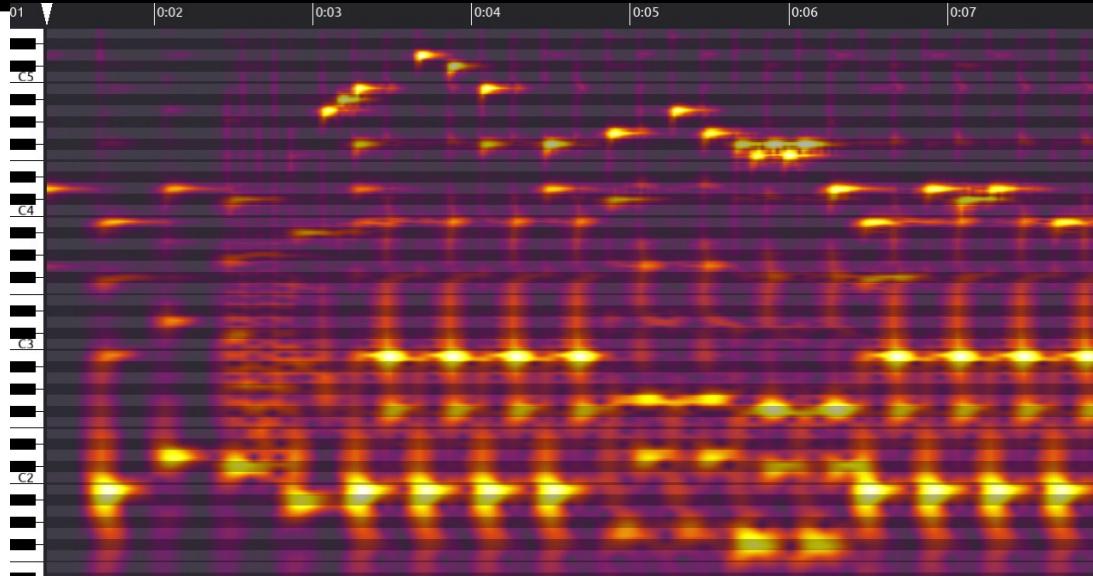
G#2 A#2 C#3 D#3 F#3 0 1 G#3 2 2 A#3 4 3 C#4 7 5 D#4 9 6 F#4 12 8 G#4 14 9 A#4 16 0 C#5 19 S D#5 21 D F#5 24 G G#5 A#5 C#6 D#6 F#6 G6

A2 B2 C3 D3 E3 F3 G3 1 Q A3 3 W B3 5 E C4 6 R D4 8 T E4 10 Y F4 11 U G4 13 I A4 15 O B4 17 P C5 18 Z D5 20 X E5 22 C F5 23 V G5 A5 B5 C6 D6 E6 F6 G6

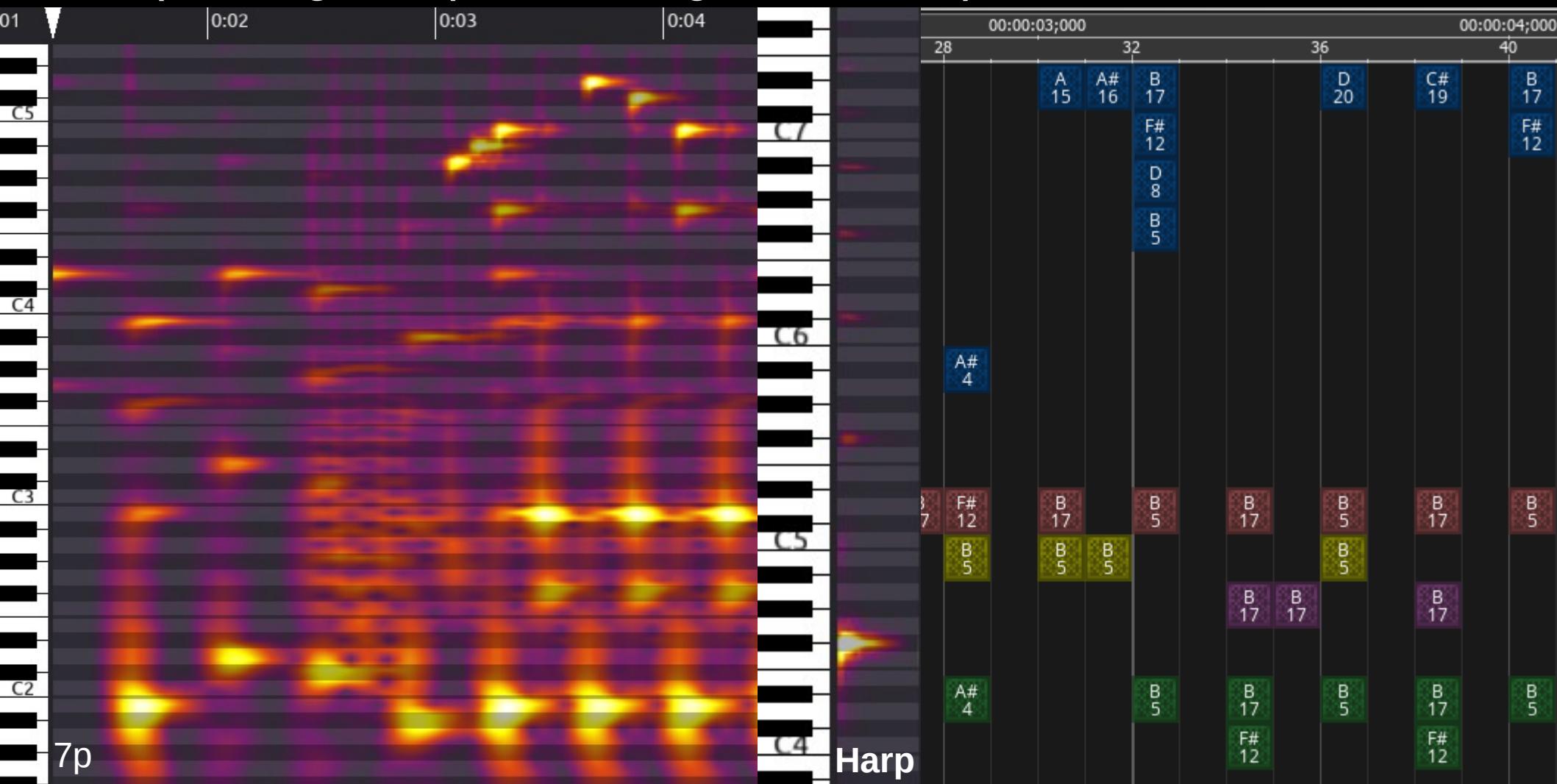
Instrument: Click Key: C4 Tick: None Layer: None Selected: 0 / 4 Sounds: 0 / 256 No connected MIDI devices

Konverzió, keresés módszere

- Spektrogram:
Fourier-transzformációk
fix időközönként
- Vizsgálandó időpillanatok
- Előfelismerés az állapottér
csökkentése érdekében
- Optimalizálás Steepest Descent módszerrel



Spektrogram (wave-ből generálható), és az eredeti NBS

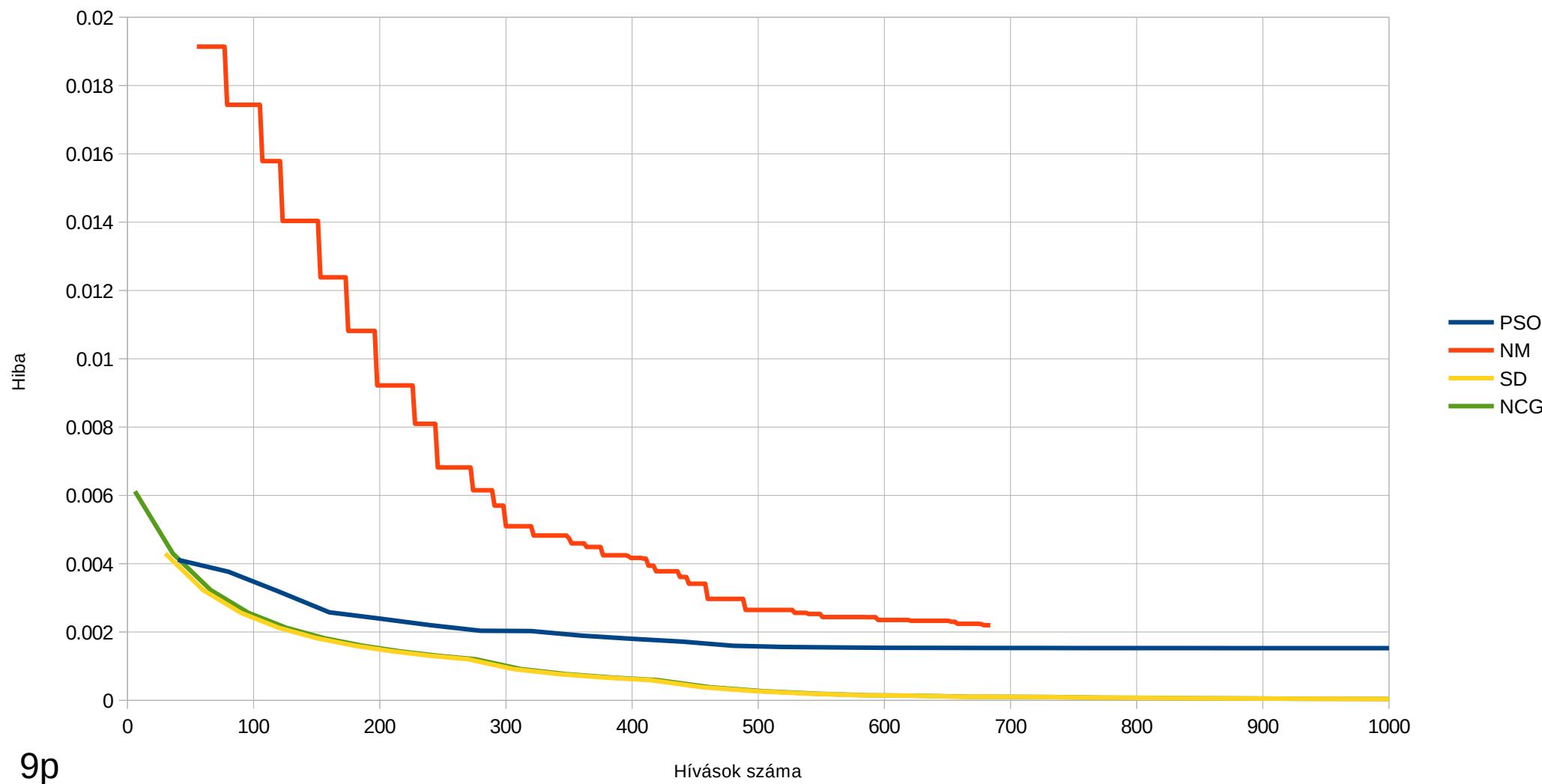


Költségfüggvény az optimalizációhoz

- Költségfüggvény:
bemenet: n db float a tippelt hangerőkkel
kimenet: egy float, a hiba / költség (kisebb = jobb)
- Számítása:
minták spektrogramjainak összeadása a tippelt hangerőn
előző hangok lecsengésének figyelembe vétele
különbség normanégyzete pixelenként
- Gradiens: nem könnyű meghatározni, de sokat pontosít



Optimalizációs algoritmusok összehasonlítása



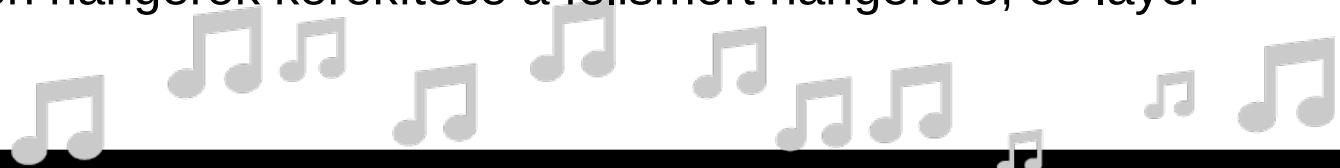
Értékelés

- nbswave kimenetéből 4 hangszert pontosan felismert
- 223 zenéből (631469 hangból) álló adathalmazon teszteltem
- többféle pontosság-mérték:
 - hangok hangerőkülönbségeinek négyzetösszege
a csendéhez viszonyítva: 98.4%
 - Jól felismert hang / összes eredeti hang (szenzitivitás): 96.8%
 - Jól felismert hang / összes tippelt hang (precizitás): 83.61%
 - Az utóbbi kettő a hangerőt nem veszi figyelembe



Fejlesztési lehetőségek

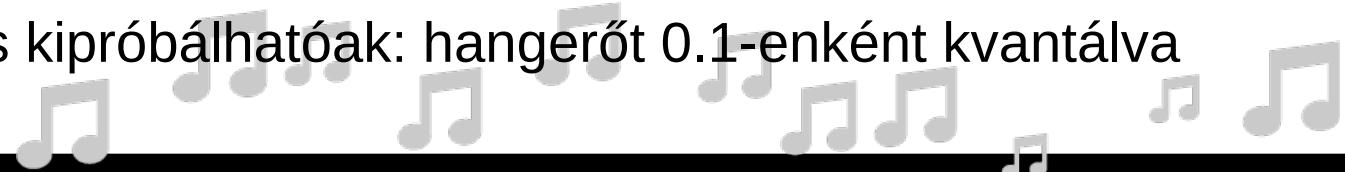
- több hangszer felismerése
- tesztelés más adathalmazon
- tps felismerésének javítása, túlvezérlés-kompenzáció felismerése
- robusztusabb felismerés másfajta nbs to wave exporterek / módszerek által generált hangfájlokra
- tesztelés más resource pack-kel / soundfonttal
- futásidő optimalizálása: gyorsabb minimumkereső algoritmus, iterációszám okos megválasztása
- megpróbálni a layereket felismerni, amiken általában ugyanolyan hangerővel szerepelnek a hangok, ezen hangerők kerekítése a felismert hangerőre, és layer volume-ok használata



Bírálói kérdések

**Milyen egyéb optimalizáló algoritmust ismer?
Esetleg mely algoritmust lehetne bevonni a keresésbe?**

- Nagyon sok módszer elérhető
- Steepest Descent variánsait mindenkor érdemes lehet használni
 - Stochastic Gradient Descent, illetve továbbfejlesztései, pl: Adam
 - potenciálisan sokat gyorsíthat
- Broyden–Fletcher–Goldfarb–Shanno algoritmus:
hasonlóan $f: \mathbb{R}^n \rightarrow \mathbb{R}$ függvény minimumát keresi, a második deriváltak (Hesse-mátrix) becslésével
- Diszkrét algoritmusok is kipróbálhatóak: hangerőt 0.1-enként kvantálva



Bírálói kérdések

Lenne-e lehetőség az irodalomkutatásban is bemutatott mátrix faktorizációs eljárások (pl.: NMF) alkalmazására?

- Egy megfelelő optimalizációs algoritmus használatához képest önmagában feltételezhetően jóval pontatlanabb lenne
 - lecsengést nem tudja a pontosság növeléséhez figyelembe venni
- Előfeldolgozáshoz, és közelítő eredményhez viszont a most alkalmazott módszer helyett érdemes lehet kipróbálni
- A felismerendő zene időbeli pontatlanságára is kevésbé volna érzékeny



NMF kottázásra

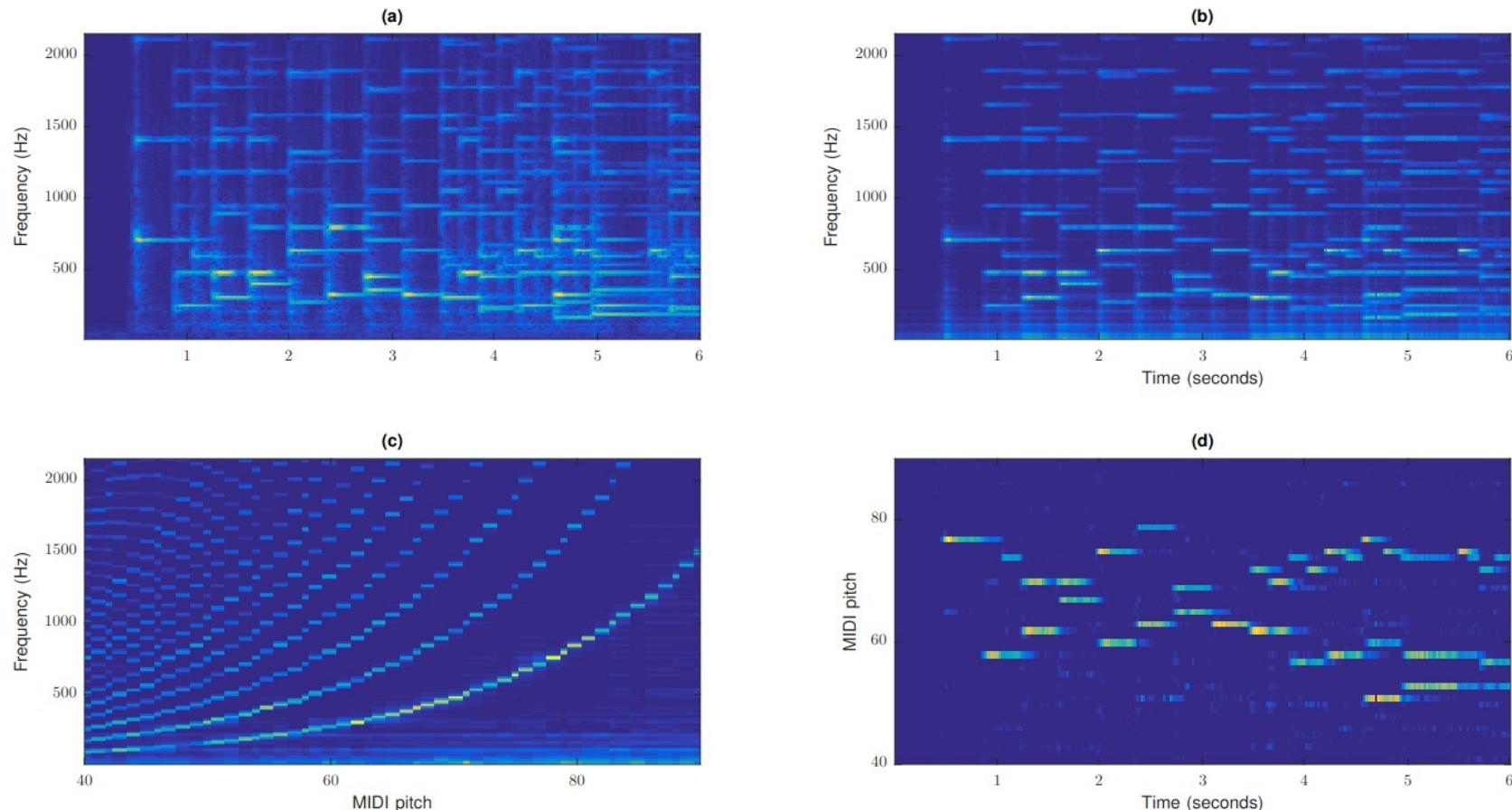


Figure 3. NMF example, using the same audio recording as Fig. 1. (a) Input spectrogram \mathbf{V} , (b) Approximated spectrogram \mathbf{DA} , (c) Dictionary \mathbf{D} (pre-extracted), (d) Activation matrix \mathbf{A} .

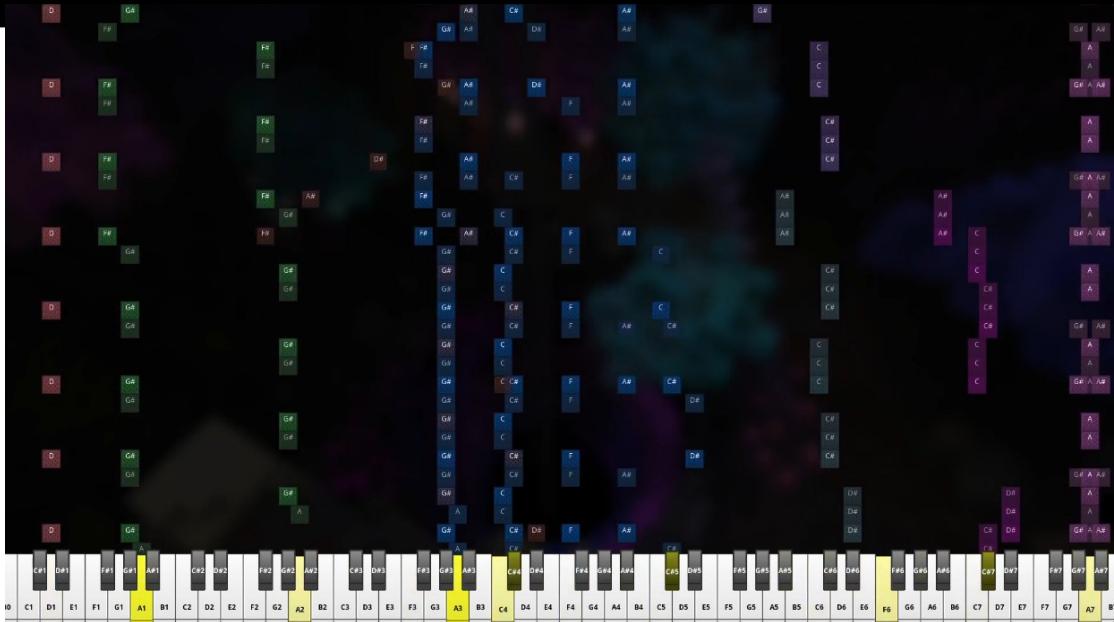
Kérdések

- Köszönöm szépen a figyelmet!
- Bizottság kérdései



Extra dia: Motiváció

- Sok szerző készít ilyen zenét
 - Vizualizáció
 - Remix / átdolgozás
 - Kottakészítés
 - Lejátszani hangszeren
-
- Wave → Midi → Nbs nagyon pontatlan, közbeavatkozás kell



Extra dia: Adat tulajdonságai

- Különböző hangszerek (5..16)
- 2 oktávnyi hangmagasság (25)
- Hangerő 0.0 és 1.0 között, de lehet többszörös is
- Időzítés tps-ben (tick: legkisebb időegység), ONBS-ben oszlop



Források az optimalizációs kérdéshez

- https://en.wikipedia.org/wiki/Mathematical_optimization
- <https://docs.scipy.org/doc/scipy/tutorial/optimize.html>
- https://en.wikipedia.org/wiki/Stochastic_gradient_descent
- https://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno_algorithm
- <https://medium.com/codechef-vit/why-bfgs-over-gradient-descent-3ecc3e7ffd>
- <https://emiliendupont.github.io/2018/01/24/optimization-visualization/>

