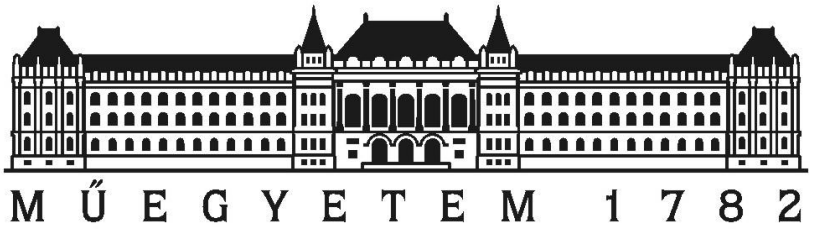


AUTOMATIKUS KOTTÁZÁS NMF-ALGORITMUSSAL

TDK DOLGOZAT

2020

Készítette:

Szemerey Helén

Konzulens:

Dr. Fiala Péter

1 Tartalom

2 Kivonat 3

3 Hasonló alkotások 4

4 Bevezető 5

5 Az NMF módszer 5

1 Folytonossági feltétel 8

2 Ritkasági feltétel 11

6 W és H mátrixok feldolgozása 13

1 Az alapharmonikusok meghatározása 13

2 HPS algoritmus 14

7 W és H mátrixok méretének helyes megválasztása 16

1 Spektrogram vizsgálata csúszóablakkal 18

8 H mátrix meghatározása 20

1 H előállítása időablakok segítségével 20

2 H előállítása csúszóablakok nélkül 22

9 Eredmények, továbbfejlesztési lehetőségek 24

10 Irodalomjegyzék 29

2 Kivonat

A dallamok lejegyzésének igénye feltehetően egyidős a dallamok létezésével. A kotta egy zenei jelrendszer, melyet a zeneszerzők zeneművek leírására használnak.

A számítógépek és digitális jelfeldolgozás megjelenésével lehetőség nyílt a hangjelek automatikus kottázására. Jelenleg számos e célra specializált szoftver érhető el a piacon, de pontosságuk koránt sem tökéletes, és az alkalmazott módszerek aktív kutatás tárgyai.

A feladat kidolgozásával a célom egy újszerű algoritmus kidolgozása volt, mely a bemeneti hangfelvételből egy olyan köztes leírást hoz létre, melyből a kottázás könnyen elvégezhető.

A kidolgozott módszer a nemnegatív mátrixfaktorizáláson alapul, mely a hangjel spektrogramjának alacsony rangú diadikus felbontását adja meg. A diadikus felbontás oszlopai a megszólaló hangjel-bázis spektrumai, sorai pedig az egyes bázishangokhoz tartozó intenzitás-időfüggvények. Az NMF- módszer pontosságát alapvetően befolyásolja a diadikus felbontás rangja. Dolgozatomban az optimális rang meghatározásával foglalkozom, és megmutatom, hogy milyen módszerekkel lehet megközelíteni az optimális értéket. Az egyes módszerek pontosságát generált hangmintákon, illetve valós hangfelvételeken alkalmazott tesztekkel demonstrálom.

3 Hasonló alkotások

Manapság rengeteg különböző alkalmazást találni, melyek zenei hangok feldolgozására specializáltak. A legegyszerűbb programok az alapfrekvencia felismerésére képesek. Ilyen például a *Fundamental Frequency and Harmonics of a Violin Note* névre hallgató program, mely amplitúdóspektrum információinak felhasználásával keresi meg egy-egy hegedűn játszott hang alapfrekvenciáját [ViolinNote].

Vannak egyszerűbb mobilapplikációk, melyek megfelelő pontossággal képesek meghatározni külön-külön megszólaló hangokat, de ha egyszerre két vagy több hangot szólaltattak meg, azt már nem képesek kezelni. Ezek a mobil alkalmazások nem kottázásra lettek kitalálva, magukat hangszerhangoló alkalmazásként hirdetik. Néhány példa erre: Cleartune (Android és iOS), TonalEnergy (iOS), iStroboSoft (Android és iOS) [TunerApp].

Az automatikus kottázásra specializálódott szoftverek is találhatóak a piacon, bár nyílt forráskódút alig találni. Egy úgynevezett *AnthemScore* (melyet vezető szoftverként tartanak számon az automatikus kottázás terén) névre hallgató szoftvert demó verzióját volt szerencsém kipróbálni. Képes kottát felírni digitális hangminták alapján. Magáról a program működéséről, jelfeldolgozásáról nem árulnak el semmit részletesen, csupán annyit említenek, hogy az automatikus kottázást milliónyi adatmintára kiképzett neurális háló segítségével végzik [AnthemScore].

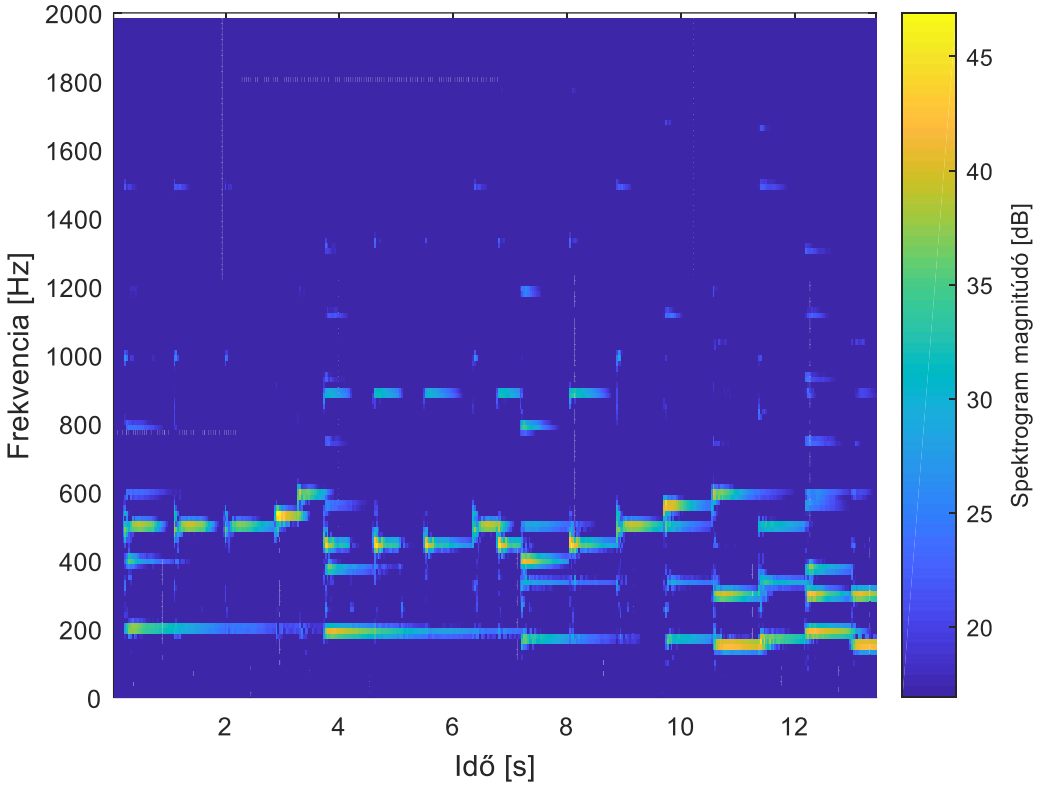
4 Bevezető

Többszólamú zene esetén az automatikus kottázás a mai napig sem teljesen egyértelmű. A megoldására a javasolt megközelítések nagy része előzetes ismereteken alapszik (például jelmodellekre) [Constantini2009]. Az efféle megoldásoknak fő gyengeségük, hogy nem képesek megfelelően alkalmazkodni olyan jelekhez, melyek nem felelnek meg a modellnek. Ennek elkerülése érdekében a lehető legkevesebb hipotézist használjuk fel az egyes hangjegyek meghatározásakor. A nemnegatív mátrixfaktorizálás (NMF) egy ilyen, kevés hipotézist felhasználó módszer, amely nem mellesleg ígéretes eredményeket mutat a polifonikus zene feldolgozása terén. Más algoritmusok valószínűleg hatékonyabbak a számítási időt tekintve, de nehezebb őket megvalósítani, és nem általánosíthatják a különböző költségfüggvényeket [Bertin2007].

E számítási módszer azon alapszik, hogy ismerjük a zenében előforduló különböző hangjelek számát, mert ettől függ a mátrixok méretezése. Egy zenei hang jól jellemezhető egy spektrummal. A különböző spektrumok egy mátrixba rendezése fogja alkotni az úgynevezett bázist, melynek ismeretében már könnyedén elvégezhető a kottázás. Az NMF algoritmus használatához helyesen meg kell becsülnünk e bázis méretét. A TDK dolgozatom során elsősorban erre a problémára kerestem a megoldást. Egyszólamú zene feldolgozása során megfelelően jó közelítést tudtam adni a bázist illetően.

5 Az NMF módszer

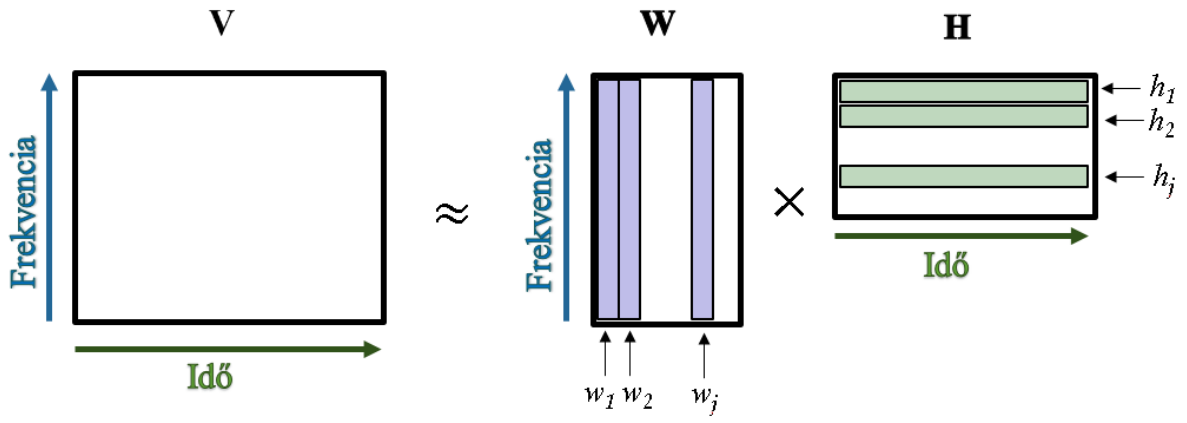
Az NMF algoritmus az adatok spektrogramját *r* elemi spektrumok lineáris kombinációjának tekinti az egyes időpontokban (5-2. ábra). Maga a spektrogram alatt a spektrum időbeli változásának ábrázolását értjük A spektrogram számítása során a felvett jel mintáiból fix méretű ablakokat készítünk –jó frekvenciaátadási képesség érdekében–, és tipikusan átlapolással, hogy ne romoljon az időfelbontás. Ezekre az ablakokra számítunk spektrumot, majd a spektrogramon az egymást követő ablakokon számított spektrumokat (tipikusan az abszolút értéküket, vagy az ebből számított teljesítménysűrűséget) ábrázoljuk (5-1. ábra).



5-1. ábra – Egy zenedarab spektrogramja

Vegyünk egy nemnegatív V mátrixot, mely egy zenei darab idő-frekvencia reprezentációja – a zenedarab spektrogramjának abszolút értéke-, ahol $V \in \mathbb{R}_{+}^{m \times n}$. Két mátrixot keresünk: $W \in \mathbb{R}_{+}^{m \times r}$ és $H \in \mathbb{R}_{+}^{r \times n}$, melyekre igaz:

$$V \approx W\,H \tag{5-1}$$



5-2. ábra – NMF algoritmus

A közelítést úgy kell értenünk, hogy az eredeti V és annak rekonstrukciója WH közötti „távolságot” minimalizálni kell. Érdemes figyelembe vennünk ezeknek a mátrixoknak a nemnegativitását, tehát hogy csak nulla vagy pozitív együtthatóik lehetnek. A W mátrix oszlopai spektrumokat tartalmaznak, melyek a paraméterek helyes megválasztása esetén megfeleltethetőek a spektrogramban megjelenő zenei hangoknak, és H mátrix sorai jelentik az említett spektrumok mindegyikének időbeli aktivitását a megfigyelt jelben:

$$V = \sum_{j=1}^{r} w_j h_j \tag{5-2}$$

Ahol w_j a W mátrix egy oszloa, h_j pedig a H mátrix egy sora (5-2. ábra). A nemnegatív mátrixfaktorizálás során az érintett mátrixok nemnegatív tulajdonsága az egyetlen, amelyet kihasználunk. Ha egy elég hosszú és elegendő számú zenei eseményt tartalmazó zeneművet veszünk figyelembe, akkor képesek leszünk reprezentálni a jelet hangjegyeknek megfelelő spektrumokkal [Bertin2007].

Az NMF algoritmusa a W és a H iteratív frissítésein alapulnak. Algoritmusaink minden egyes iterációjában a W vagy H új értékét úgy kapjuk meg, hogy az aktuális értéket megszorozzuk egy nemnegatív, a gradienssel analóg értelmezésű tényezővel, amely az ($V \sim WH$) egyenletben szereplő közelítés minőségétől függ. A közelítés minősége monoton módon javul ezen szorzó frissítési szabályok alkalmazásával. A gyakorlatban ez azt jelenti, hogy a frissítési szabályok ismételt iterációja garantáltan az optimális mátrixfaktorizációhoz konvergál [Lee2001].

Első lépésként meg kell határoznunk azokat a költségfüggvényeket, amelyek a közelítés minőségét számszerűsítik. Egy ilyen költségfüggvény két nemnegatív mátrix közötti távolság bizonyos mértékének felhasználásával állítható elő $\left(\underline{\underline{\varepsilon}} = \underline{\underline{V}} - \underline{\underline{W}}\,\underline{\underline{H}}\right)$. Az egyik hasznos mérték egyszerűen a két mátrix közötti euklideszi távolság négyzete.

$$C\left(\underline{\underline{\varepsilon}}\right) = \sum\sum \varepsilon_{ij}^2 \tag{5-3}$$

E célfüggvényt szeretnénk minimalizálni az iteratív algoritmusunkkal. Ennek a klasszikus megoldása a gradiens módszer, vagyis mindig vesszük a függvény gradiensét, és mindig negatív gradiens irányába mozdulunk el [Gisbert2005]. Viszont figyelembe kell venni emellett a nemnegatívitást, tehát biztosítani kell, hogy az algoritmus csak nemnegatív eredményt tudjon adni, ezért szorzunk minden iterációs lépésben elemenként egy nemnegatív mátrixszal. Ezt a nemnegatív mátrixot úgy állítjuk össze, hogy a gradienst $\left(\frac{\partial C}{\partial W_{\alpha\beta}}\ \frac{\partial C}{\partial H_{\alpha\beta}}\right)$ két pozitív tag különbségére választjuk szét, majd vesszük azok arányát.

E felbontásból kapott két tag hányadosával fogom megszorozni a *W* és a *H* mátrixokat az egyes iterációs lépésekben [Lee2001]:

$$W_{k+1} = W_k \left(\frac{\frac{\partial C^-}{\partial W}}{\frac{\partial C^+}{\partial W}}\right) = \frac{VH^T}{W(HH^T)} \tag{5-4}$$

$$H_{k+1} = H_k \left(\frac{\frac{\partial C^-}{\partial H}}{\frac{\partial C^+}{\partial H}}\right) = \frac{W^T V}{W^T W H} \tag{5-5}$$

1 Folytonossági feltétel

A bemutatott mátrixfrissítési módszerrel elérhető, hogy W és H szorzata kellően megközelítse az eredeti V mátrixot, vagyis az euklideszi távolságuk minimális lesz. Azonban egy zenei darab esetén ez nem elegendő a valós megoldáshoz. A zenei hangok akusztikus jellemzőik az idő függvényében lassan változnak [Chen2006].

Az időbeli folytonosságot úgy mérjük, hogy az egy-egy spektrumhoz tartozó időbeli aktivitások közötti változásokhoz költségeket rendelünk. Így tehát W és H becslése úgy történik, hogy

minimalizáljuk a $c(W,H)$ költségfüggvényt, amely több kifejezés súlyozott összege: egy rekonstrukciós hiba kifejezés $c_r(W,H)$, egy folytonossági kifejezés $c_t(W,H)$ stb :

$$\nabla c = \nabla c_r + \alpha \nabla c_t + \cdots \tag{5-6}$$

ahol α a folytonossági kifejezés súlyát jelenti. A Hjt és a $Hj(t-1)$ –ahol *j* jelöli *H*-nak a *j*. sorát és *t* a *t.* oszlopát– erősítés közötti nagy változásokhoz költségeket rendelek. A nyereségeket normalizáljuk a σ_j szórással, így az időbeli folytonosság c_t költségfüggvénye így írható:

$$C_t(H) = \sum_{j=1}^{R} \frac{\Delta_j}{\sigma_j^2} \tag{5-7}$$

ahol $\sigma_j^2 = \frac{1}{T}\sum_{t=1}^{T} H_{jt}^2$ és $\Delta_j = \sum_{t=2}^{T}\left(H_{jt} - H_{j(t-1)}\right)^2$.

A multiplikatív frissítéshez szükséges szorzótényező meghatározásához képezni kell e költségfüggvény *H* szerinti parciális deriváltját, ahogy korábban is tettük.

$$Q = \frac{\partial C_t(H)}{\partial H_{\alpha\beta}} = \sum_{j=1}^{R} \frac{(\Delta_j)'\sigma_j^2 - \Delta_j(\sigma_j^2)'}{(\sigma_j^2)^2} \tag{5-8}$$

A szórásnégyzet $H_{\alpha\beta}$ szerinti parciális deriváltja:

$$(\sigma_j^2)' = \frac{1}{T}\sum_{t=1}^{T} 2H_{jt}\frac{\partial H_{jt}}{\partial H_{\alpha\beta}} \tag{5-9}$$

$$\frac{\partial H_{jt}}{\partial H_{\alpha\beta}} = \begin{cases} 1, & ha \;\; j = \alpha \text{ és } t = \beta \\ 0, & egyébként \end{cases} \tag{5-10}$$

Tehát:

$$\frac{\partial H_{jt}}{\partial H_{\alpha\beta}} = \delta_{j\alpha}\delta_{t\beta} \tag{5-11}$$

ahol $\delta_{j\alpha} = \begin{cases} 1, & ha \;\; j = \alpha \\ 0, & egyébként \end{cases}$,és $\delta_{t\beta} = \begin{cases} 1, & ha \;\; t = \beta \\ 0, & egyébként \end{cases}$

És ebből következik, hogy a szórásnégyzet parciális deriváltja:

$$(\sigma_j^2)' = \frac{1}{T} 2H_{j\beta}\delta_{j\alpha} \tag{5-12}$$

Δ_j parciális deriváltjának számítása:

$$(\Delta_j)' = \sum_{t=2}^{T} 2\left(H_{jt} - H_{j(t-1)}\right)\left[\delta_{j\alpha}\delta_{t\beta} - \delta_{j\alpha}\delta_{(t-1)\beta}\right]\delta_{j\alpha} \tag{5-13}$$

$$(\Delta_j)' = \delta_{j\alpha} 2\left(2H_{j\beta} - H_{j(\beta-1)} - H_{j(\beta+1)}\right) \tag{5-14}$$

Tehát a teljes költségfüggvény parciális deriváltja:

$$Q = \frac{(\Delta_\alpha)'\sigma_\alpha^2 - \Delta_\alpha(\sigma_\alpha^2)'}{(\sigma_\alpha^2)^2} \tag{5-15}$$

$$Q = \frac{2\left(2H_{\alpha\beta} - H_{\alpha(\beta-1)} - H_{\alpha(\beta+1)}\right)\sigma_\alpha{}^2 - \Delta_\alpha \frac{1}{T} 2H_{\alpha\beta}}{(\sigma_\alpha^2)^2} \tag{5-16}$$

$$Q = \frac{4H_{\alpha\beta}}{\sigma_\alpha^2} - \left[\frac{2(H_{\alpha(\beta-1)} - H_{\alpha(\beta+1)})}{\sigma_\alpha^2} + \frac{\Delta_\alpha \frac{1}{T} 2H_{\alpha\beta}}{(\sigma_\alpha^2)^2}\right] \tag{5-17}$$

ahol:

$$\nabla^+ c_t = \frac{4H_{\alpha\beta}}{\sigma_\alpha^2} \tag{5-18}$$

$$\nabla^- c_t = \frac{2(H_{\alpha(\beta-1)} - H_{\alpha(\beta+1)})}{\sigma_\alpha^2} + \frac{\Delta_\alpha \frac{1}{T} 2H_{\alpha\beta}}{(\sigma_\alpha^2)^2} \tag{5-19}$$

Végül ennek megfelelően kell bővíteni a multiplikatív frissítési szabályt a H mátrix esetében [Virtanen2007]:

$$H \leftarrow H\frac{\nabla c^{-}}{\nabla c^{+}} = H\frac{\nabla {c_r}^{-} + \alpha \nabla {c_t}^{-} + \cdots}{\nabla {c_r}^{+} + \alpha \nabla {c_t}^{+} + \cdots} \tag{5-20}$$

2 Ritkasági feltétel

Az úgynevezett ritkasági kritérium bizonyos esetekben javít a zenei hangok felismerésének minőségén. Ha megoldható úgy egy rekonstrukció, hogy nem báziselemek szuperpozícióját, hanem minél kisebb bázist használ fel, akkor törekedjen arra [Hoyer2004].

Ahogy korábban is tettük, a ritkasági kritériumhoz is rendelünk egy $c_S(H) = \sum_{j=1}^{J}\sum_{t=1}^{T} f(\frac{h_{j,t}}{\sigma_j})$ költségfüggvényt, ahol $f(x) = |x|$.

Így a ritkasági költségfüggvény parciális deriváltja a következőképpen fog kinézni:

$$[\nabla c_s(H)]_{j,t} = \frac{1}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} {h^2}_{j,i}}} - \sqrt{T}\frac{h_{j,t}\sum_{i=1}^{T} h_{j,i}}{\left(\sum_{i=1}^{T} {h^2}_{j,i}\right)^{3/2}} \tag{5-21}$$

Ebből következik, hogy:

$$[\nabla {c_s}^{+}(H)]_{j,t} = \frac{1}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} {h^2}_{j,i}}} \tag{5-22}$$

és

$$[\nabla {c_s}^{-}(H)]_{j,t} = \sqrt{T}\frac{h_{j,t}\sum_{i=1}^{T} h_{j,i}}{\left(\sum_{i=1}^{T} {h^2}_{j,i}\right)^{3/2}} \tag{5-23}$$

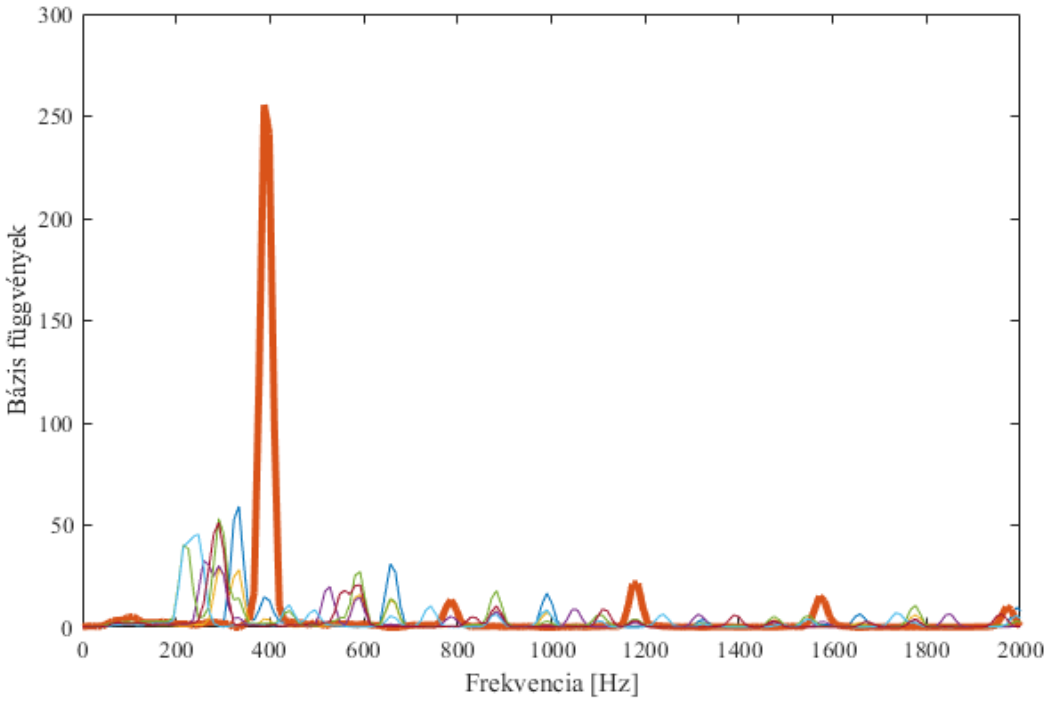
Az összköltség gradiense tehát a rekonstrukciós hiba, a folytonossági és a ritkasági kritérium gradienseinek súlyozott összege lesz [Virtanen2007].

$$\nabla c = \nabla c_r + \alpha \nabla c_t + \beta \nabla c_s \tag{5-24}$$

6 W és H mátrixok feldolgozása

Az utófeldolgozás a *W* és *H* mátrixok értelmezéséből áll. *W* minden oszlopát egy-egy különálló hang spektrumának tekintjük. Minden egyes ilyen spektrumban megkeresve az alapharmonikust, egyértelműen meghatározható a hozzá tartozó zenei hang. A *H* mátrix írja e spektrumok időbeli aktivitását, melyben a felfutások fogják megadni az egyes hangok megszólaltatásának időpontját.

1 Az alapharmonikusok meghatározása



6-1. ábra – A W mátrix oszlopai

Harmonikus hangok esetében tudjuk, hogy azok felharmonikusai egy alapharmonikus frekvenciának megfelelő frekvencia egész számú többszöröseinél jelennek meg (6-1. ábra):

$$\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_8 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ 8 \end{pmatrix} * \widehat{f_1} \tag{6-1}$$

ahol f_1 az alapharmonikus, $f_2, f_3, \dots, f_8$ a felharmonikusok, és $\widehat{f_1}$-vel szeretnénk becsülni az alapharmonikust.

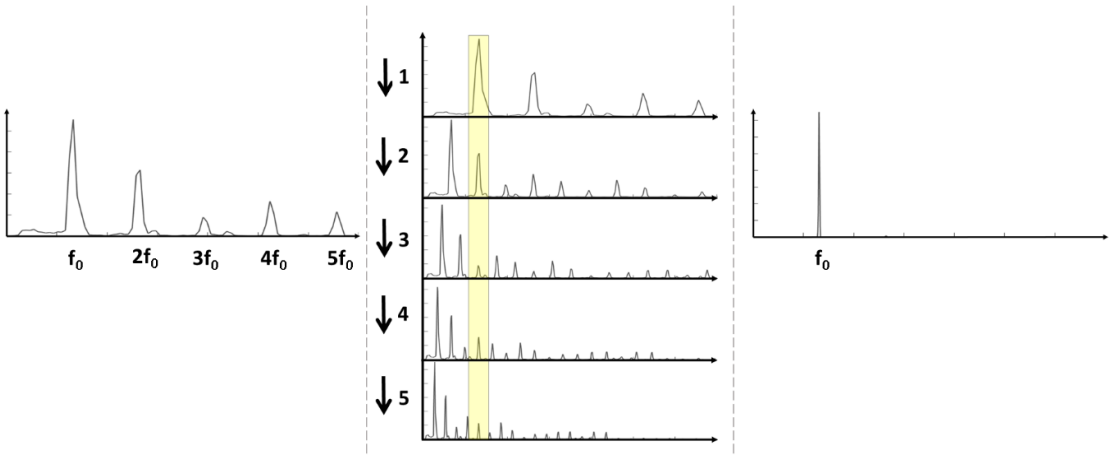
Tehát azt a $\widehat{f_1}$ frekvenciát keressük úgy, hogy $\sum_{i=1}^{8}\left(f_i - i * \widehat{f_1}\right)^2$ négyzetes eltérést minimalizáljuk, és ennek eredményeképp eljuthatunk a következő formulához:

$$\widehat{f_1} = \frac{\sum_{i=1}^{8} i * f_i}{\sum_{i=1}^{8} i^2} \tag{6-2}$$

2 HPS algoritmus

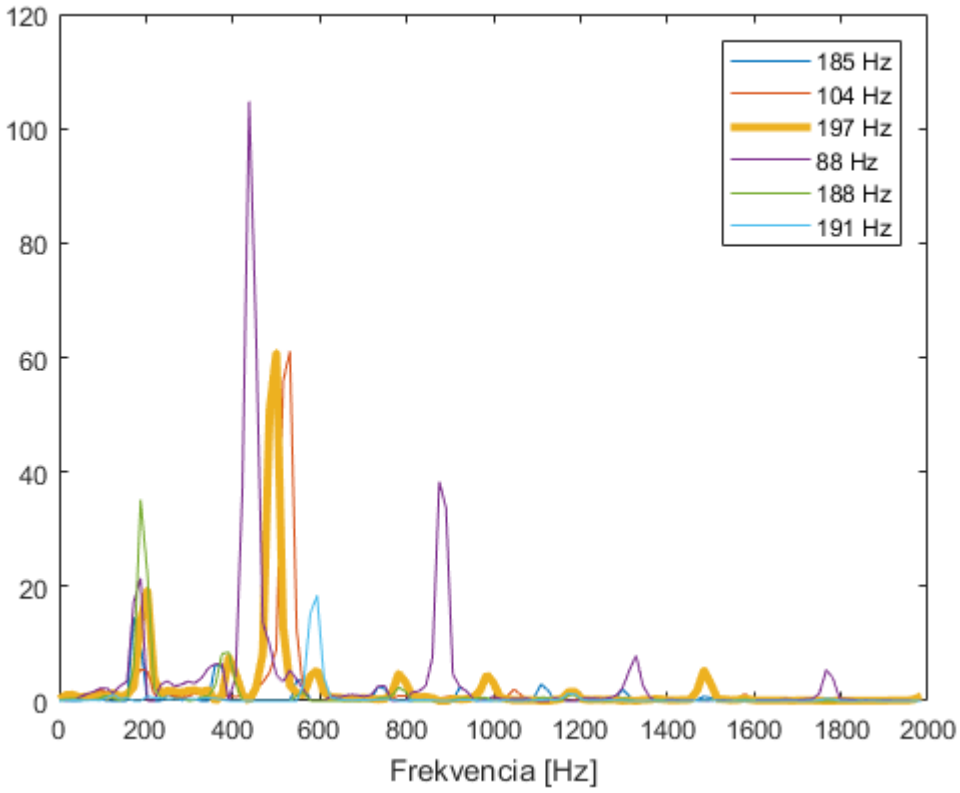
Az alapharmonikus keresésének alternatívája a Harmonic Spectrum Product (HPS) algoritmus, mely hatékonyabb a zajos jelek esetén. Alapötlete a következő: legalább N felharmonikust tartalmazó hang spektrumát N-ed részére összenyomva (ami újramintavételezéssel egyszerűen megvalósítható) az N. felharmonikus épp az alapharmonikus eredeti helyére kerül. Ezután az eredeti és az összenyomott spektrumokat összeszorozva ideális esetben az alapharmonikuson kívüli összetevők elhanyagolhatóvá válnak, és az alapharmonikus egyszerű maximumhely-kereséssel meghatározható [Patricio2001].

Ennek a módszernek az előnye, hogy kis számításigénnyel bír, gyors és érzéketlen az additív és a multiplikatív zajra is.



6-2. ábra – A HPS algoritmus működése. Bal oldalon: Az eredeti spektrum; középen: az újramintavételezett spektrumok; jobb oldalon: az újramintavételezett spektrumok szorzata

A nemnegatív mátrixfaktorizáció során előfordulhat, hogy egy hang alaphangja egybeesik egy másik hang felharmonikásval, ezért a dekompozíció nem tudja tökéletesen szétválasztani a két hangot, a HPS pedig ilyen formában az alaphangot keresi (6-3. ábra). Hogy megoldjam ezt a problémát, a következőt csináltam: amikor megtalálja a HPS az alaphangot, akkor



6-3. ábra – HPS algoritmus nem a megfelelő frekvenciát találja meg alaphangnak

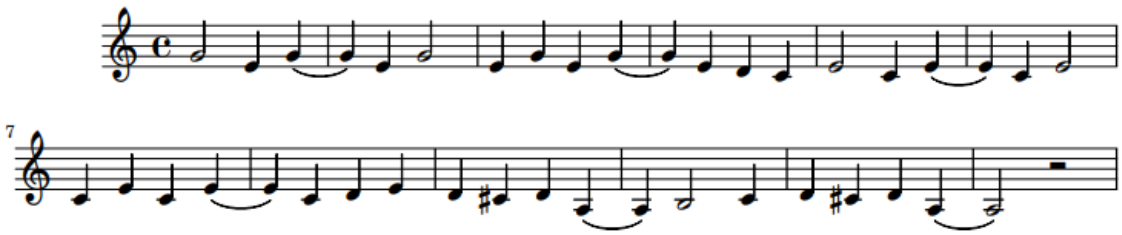
visszakeresem, hogy melyik felhangnak mekkora az amplitúdója, és amennyiben van egy erősen kiugró érték, akkor az ahhoz tartozó frekvenciát választom az alapharmonikusnak.

7 W és H mátrixok méretének helyes megválasztása

Ahhoz, hogy a legjobb közelítést adjam a bázisra, valamilyen módszerrel meg kell határozni a báziselemek számát, azaz hogy hány különböző hang szerepel az adott szegmensben.

Ha a báziselemek számát kisebb értékre állítjuk be, mint a zenedarabban előforduló hangok száma, akkor az algoritmus kihagy bizonyos hangmagasságokat. Ha viszont nagyobbra választjuk a bázist, akkor a *W* mátrixban fellelhetők lesznek redundás oszlopok. Tehát az automatikus kottázás szempontjából várhatóan akkor érhető el a legjobb eredmény, ha a bázisvektorok száma megegyezik a különböző hangmagasságok számával az aktuális zenében.

Ennek vizsgálatára különböző méretű *W* és *H* mátrixokkal végeztem el a dekompozíciót ugyanarra a zenére. Jelen esetben a Mézga család főcímdalából használtam fel egy részletet, melyről tudjuk, hogy 7 különböző hangmagasság váltakozik, vagyis várhatóan 7 lesz a bázis optimális mérete (7-1. ábra).

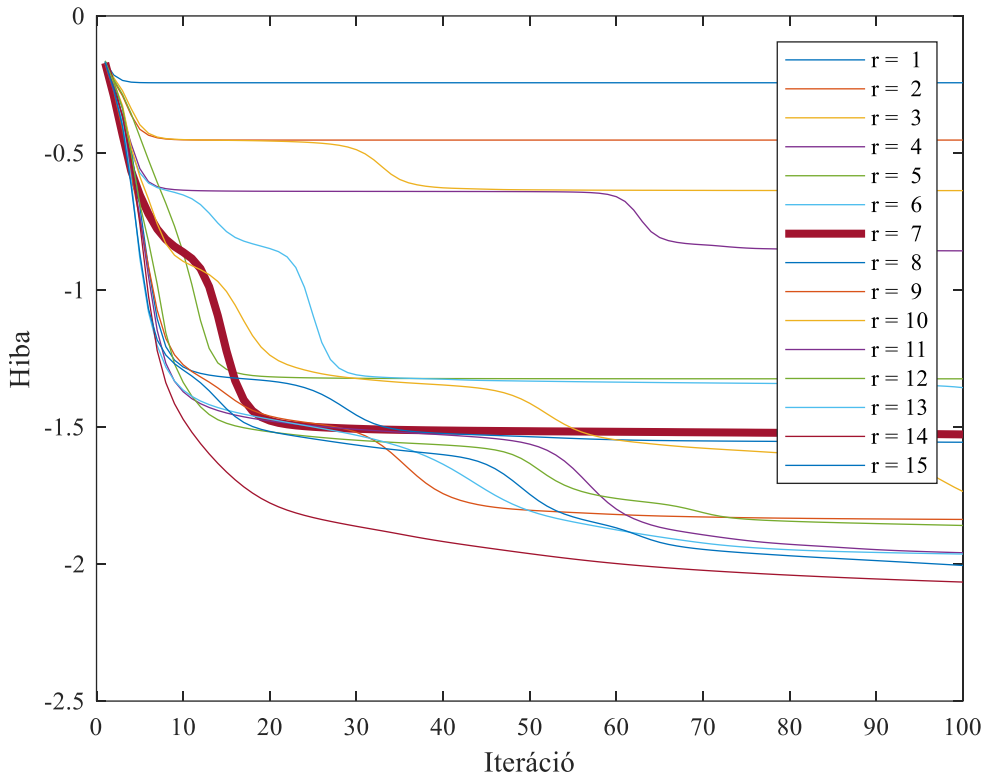


7-1. ábra – Részlet a Mézga család főcímdalából

Hogy össze tudjam hasonlítani a kapott eredményeket, mindegyik bázisszámra kiszámolom a rekonstrukció hibáját. Hiszen magának az NMF algoritmusnak az a fő célja, hogy minimalizálja a kiindulási mátrix és a rekonstrukció között távolságot.

$$\|V - WH\|^2 \rightarrow min$$

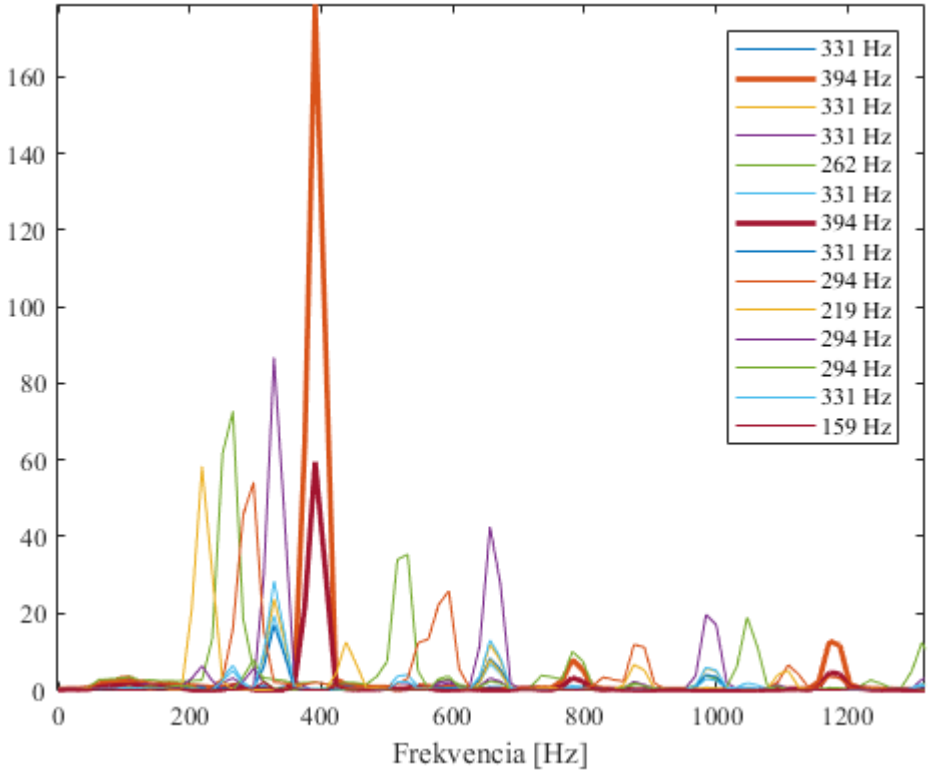
Az így kapott rekonstrukciós hibát különböző nagyságú bázissal, az iterációk számának függvényében vizsgáltam meg (7-2. ábra).



7-2. ábra – A rekonstrukciós relatív hibájának tízes alapú logaritmusa az iterációk számának függvényében különböző méretű bázisokra

Drasztikus csökkenést várunk, amíg el nem érjük a bázis pontos méretét ($r = 7$), és azon túl további moderáltabb csökkenést. A fenti ábra igazolja ezt a felvetést.

A hibán kívül megvizsgáltam, hogy alakul maga a rekonstrukció, ha a szükségesnél nagyobbra választjuk meg a mátrixokat. A fenti ábrán megfigyelhető, amikor $r > 7$, akkor valamennyi hangmagasság többször fel lesz véve a W mátrixba, ami miatt az redundánssá válik. Ezért a hibaértékek mellett célszerű figyelni arra, hogy a báziselemek különböző alapfrekvenciájú hangokat írjanak le.



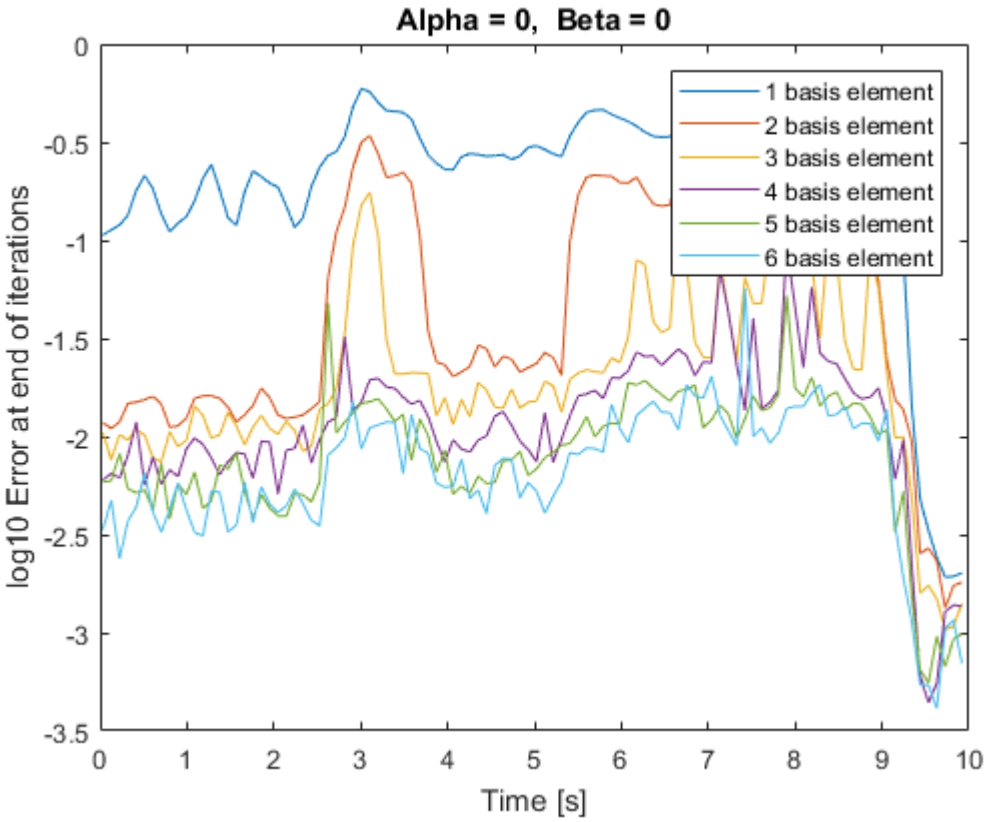
7-3. ábra – Túl nagy bázis esetén az egyes báziselemek ugyanazt az alapfrekvenciájú hangokat írják le

1 Spektrogram vizsgálata csúszóablakkal

Ha a szükségestől nagyobbra választjuk a keresendő mátrixokat, akkor lesznek olyan spektrumok, melyek ugyanazt az alapfrekvenciájú hangot írják le. A cél, hogy ezt elkerülve építsük fel a bázist. Ennek érdekében az idő függvényében létrehoztam úgynevezett csúszóablakokat. Így kapott minden időszeleten külön-külön vizsgálható, hogy mely bázisméret a legmegfelelőbb. Egy részről így képesek vagyunk a lokális hibát vizsgálni a globális hiba helyett, más részről össze tudjuk vetni az aktuális időtartományon belül számolt bázist a korábbiakkal. Ez segíteni fog abban, hogy egy olyan W mátrix álljon össze, mely minden hangmagasságot a zenéből csak egyszer tartalmaz [PSanJuan2016].

A megfelelő méretű bázis megválasztásához készítettem egy egyszerű tesztprogramot. Egy 1 másodperces ablakkal vizsgálom a bemeneti hangjelet, körülbelül 0,1 másodperces lépésközzel. Az így kapott valamennyi időintervallumra végrehajtom az NMF algoritmust 1től

6 elemű bázisig, majd az így kapott rekonstrukciós hibát vizsgáltam meg az idő függvényében (7-4. ábra).



7-4. ábra – A rekonstrukciós hiba különböző méretű bázisok esetén

Az ábrán megfigyelhető, hogy változik a hiba különböző méretű bázisokra. Az ábra alapján elmondható, hogy 1 elemű bázis használata soha nem hoz jó eredményt, ellenben a 2 elemű bázis a dallam elején ígéretes eredményeket mutat, ami magyarázható azzal, hogy a hangmintául szolgált zenerészletről tudjuk, hogy az első 2-3 másodpercben mindössze 2 különböző hang váltakozik. Viszont amint vége ennek az említett szakasznak, a 2 spektrumot tartalmazó W mátrixból számolt. NMF algoritmus rekonstrukciós hibája jelentősen megugrik, emiatt a további szakaszokon célszerű nagyobb bázist választani.

Ebből kifolyólag minden egyes megfigyelt időintervallumon kerestem egy olyan méretű bázist, melynek használatával a rekonstrukciós hiba egy előre meghatározott érték alá esik, és amennyiben olyan spektrumot tartalmazott, mely a korábbi intervallumokon nem szerepelt, felvettem a bázisba. Ezáltal megfelelő pontossággal létrehozható egy olyan W mátrix, melyben valamennyi spektrum különböző hangot jelöl. A későbbi számításokban nem fogom W értékét frissíteni

8 H mátrix meghatározása

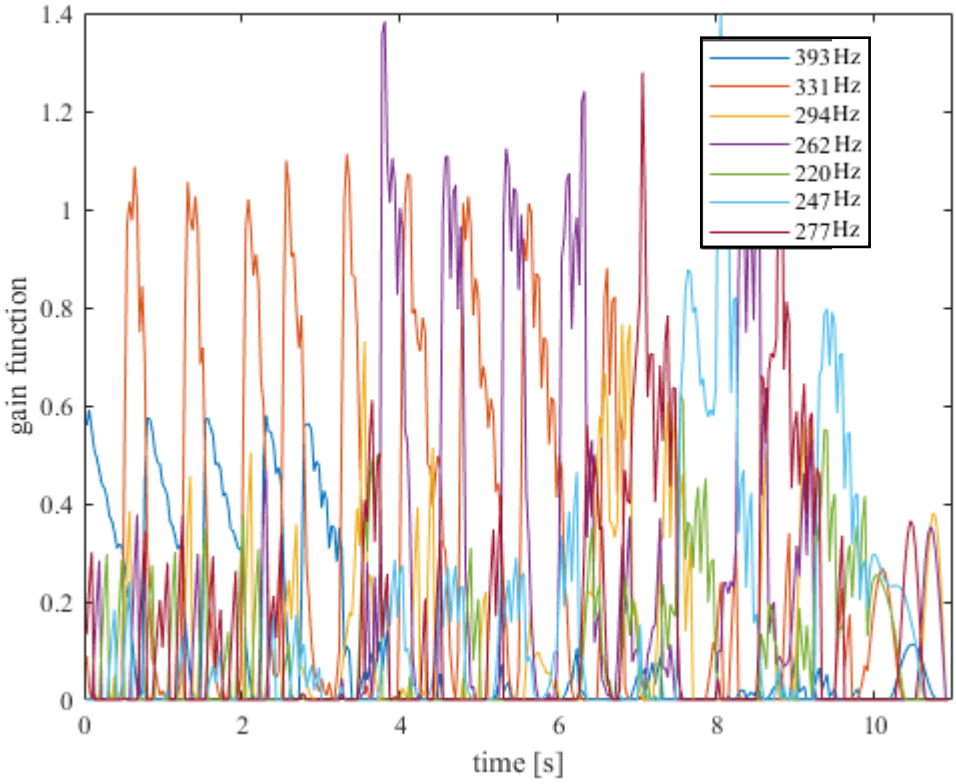
A H mátrix sorai tartalmazzák a *W* mátrixban lévő spektrumokhoz tartozó időfüggvényeit. Amennyiben már előállt egy *H* mátrix, a hangok megszólaltatásának idejét annak első deriváltjában lokális maximumhely kereséssel határozhatjuk meg.

Alapvetően kétféleképpen indulhatunk el, hogy előállítsuk *H*-t: *W*-hez hasonlóan időablakok segítségével vagy a vizsgált dallam teljes tartományára számítom ki az NMF algoritmussal. A dolgozatom során mindkét lehetőséget megvizsgáltam.

1 H előállítása időablakok segítségével

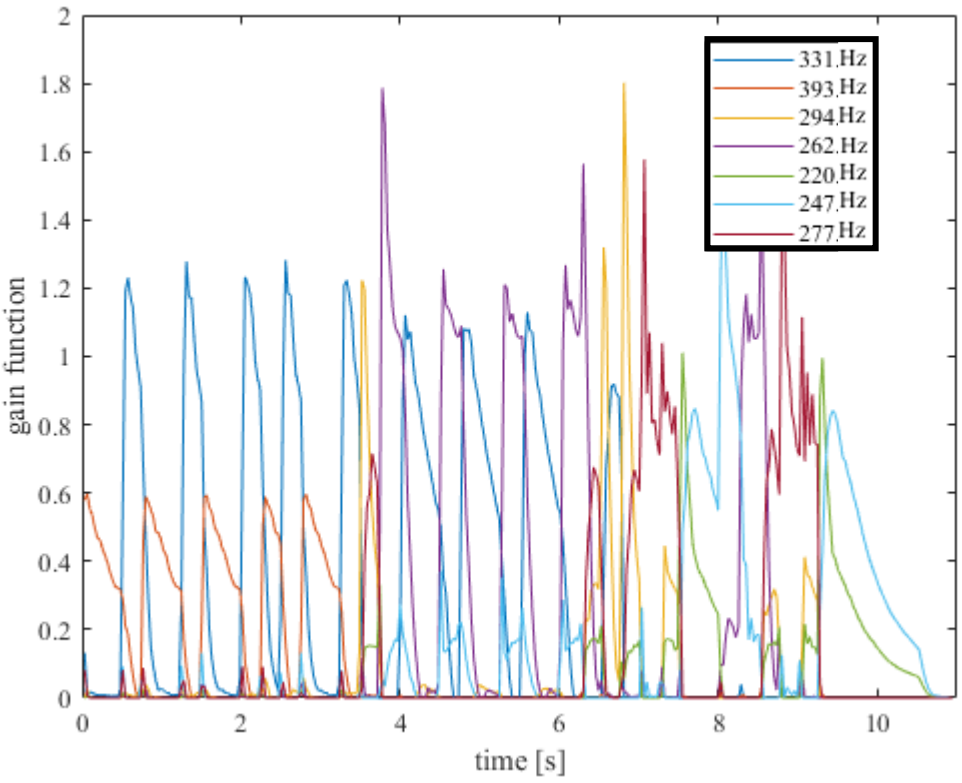
Ebben az esetben a koncepció ugyanaz, mint *W* mátrix előállításakor, tehát veszünk 1 másodperces időablakokat, melyet folyamatosan 0,1 másodperccel léptetünk, és mindegy egyes ilyen szakaszon (*W*-t fixen tartva) NMF segítségével kiszámítom *H*-t. Az alábbi ábrán látható, hogy nem kaptunk optimális eredményt a későbbi feldolgozáshoz, hiszen sok oda nem illő hang megjelenik az adott időben.

Korábbi fejezetben említettem a ritkasági feltételt, mely arra törekszik, hogy minél kevesebb báziselemet használjon fel a rekonstrukció során, de a csúszóablakos módszer esetén nem váltotta a kívánt hatást (ahogy a folytonossági feltétel sem).



8-1. ábra – csúszóablakok segítségével előállított H mátrix, valamennyi ablakon azonos bázisméretet használva

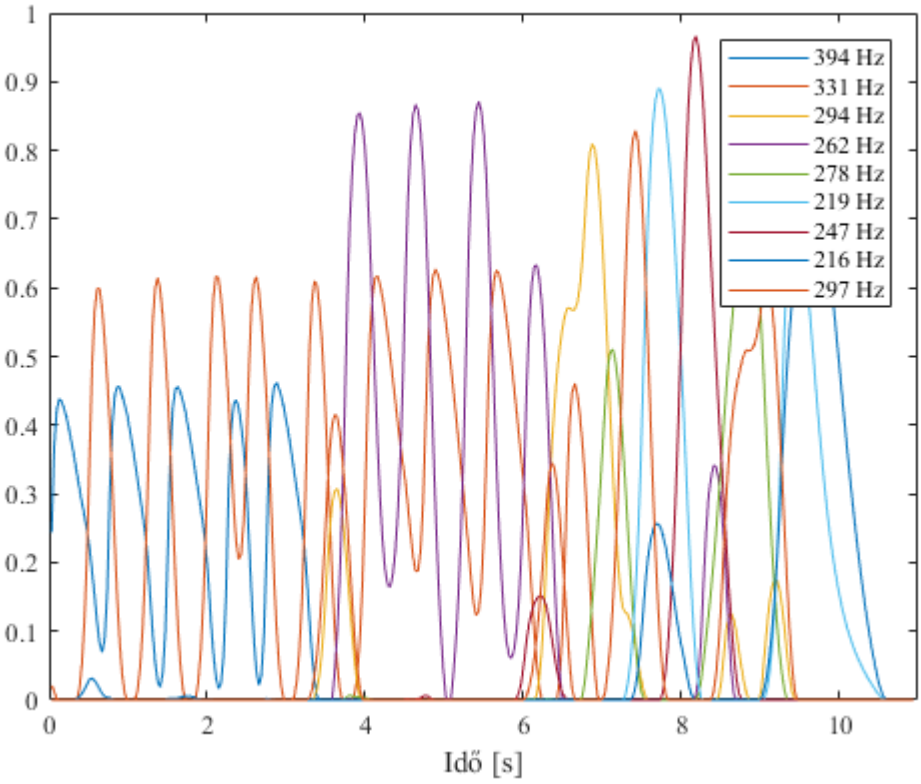
Csúszóablakos megoldásra futtattam egy másik tesztet is, amely annyiban tér el az előzőtől, hogy most nem az egész *W* mátrixot használom fel az időfüggvények számításához minden egyes ablakon, hanem éppen annyi báziselemet, mellyel a lokális rekonstrukciós hiba a lehető legkisebb lesz. Megfigyelhető, hogy ez sokat javít az eredményen, de még továbbra sem tökéletes, illetve ebben a helyzetben sem javít a ritkasági vagy a folytonossági feltétel rajta.



8-2. ábra – csúszóablakok segítségével előállított H mátrix, valamennyi ablakon különböző bázisméretet használva

2 H előállítása csúszóablakok nélkül

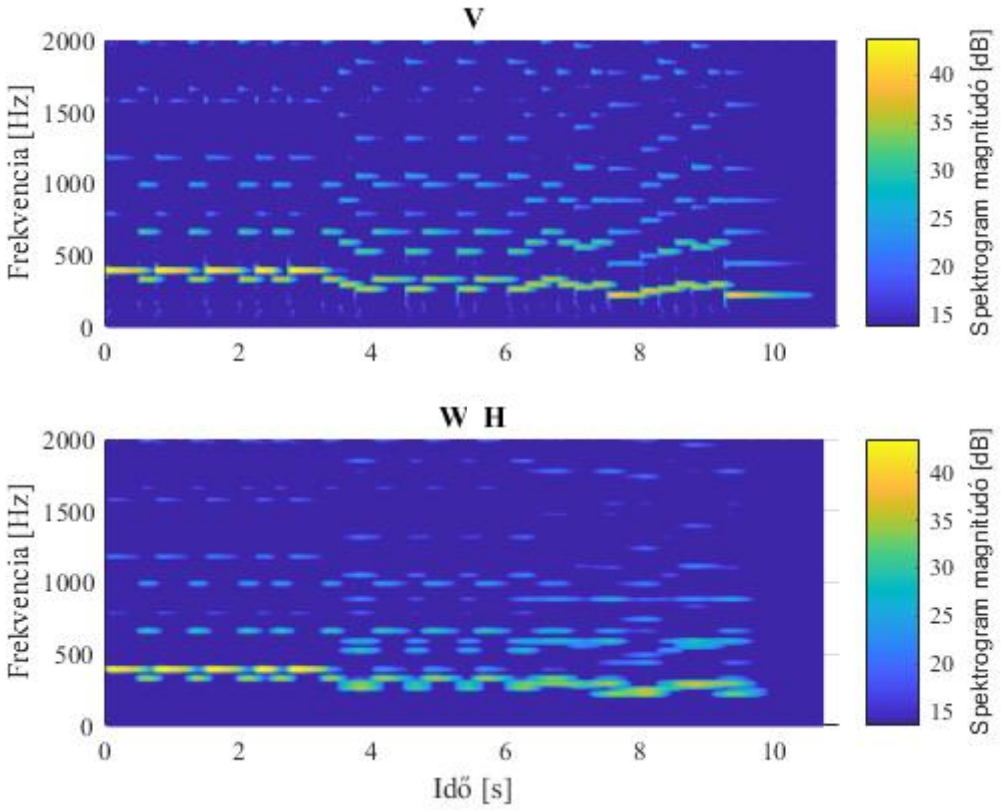
Ekkor *H*-t a zenedarab teljes egészére számítom ki az NMF algoritmussal, a korábban meghatározott bázist változatlanul hagyva. Tisztán a rekonstrukciós hibával számolva hasonló eredményt kaptam, mint az előző esetben. Viszont a folytonossági és ritkasági kritériumot figyelembe véve látványosan javultak az eredmények, és így egy sima, könnyen feldolgozható mátrixot kaptam.



8-3. ábra – H mátrix előállítása csúszóablakok nélkül

9 Eredmények, továbbfejlesztési lehetőségek

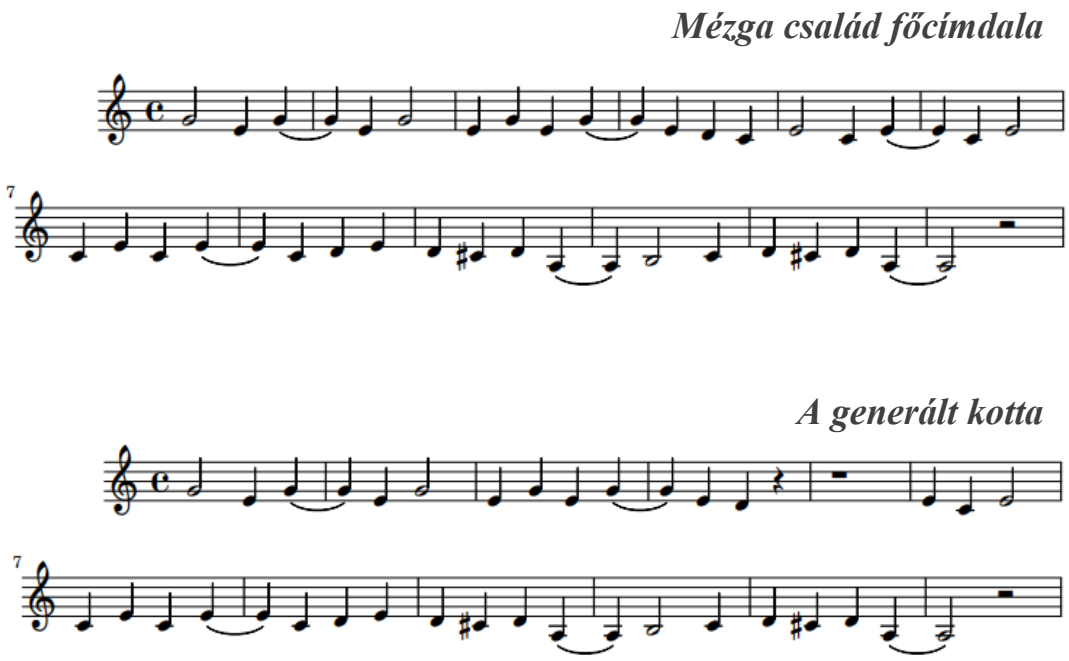
Maga az NMF algoritmus minden esetben egy megfelelő közelítést adott a zenék spektrogramjára. Példaként az alábbi ábrán látható egy zenerészlet eredeti, alatta pedig a rekonstruált spektrogramja.



9-1. ábra – egy zenedarab eredeti és rekonstruált spektogramja
Bázisméret: r = 7; Folytonossági kifejezés súlya: $\alpha = 10000$; Ritkasági kifejezés súlya: $\beta = 500$

Ahhoz, hogy tesztelhessem az algoritmus működését, olyan zenerészleteket érdemes alkalmaznom, melyeknek ismerem a kottaképét, hogy legyen mivel összehasonlítanom az végeredményt. Az eredményeket generált hangmintákat, illetve valós hangfelvételeket használva mutatom be.

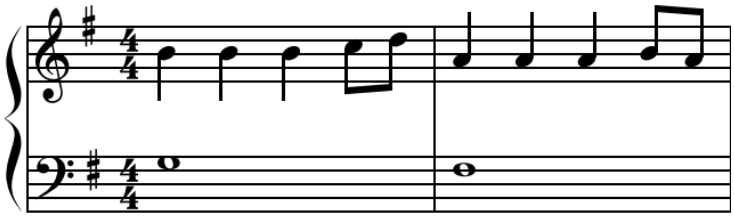
A következő ábrán a „Mézga család” főcímdalából egy részlet látható. Ez egy szintetizált egyszólamú zenerészlet, melyet az egyes módszerek kidolgozásakor is használtam tesztelésre.



9-2. ábra – Mézga család főcímdala

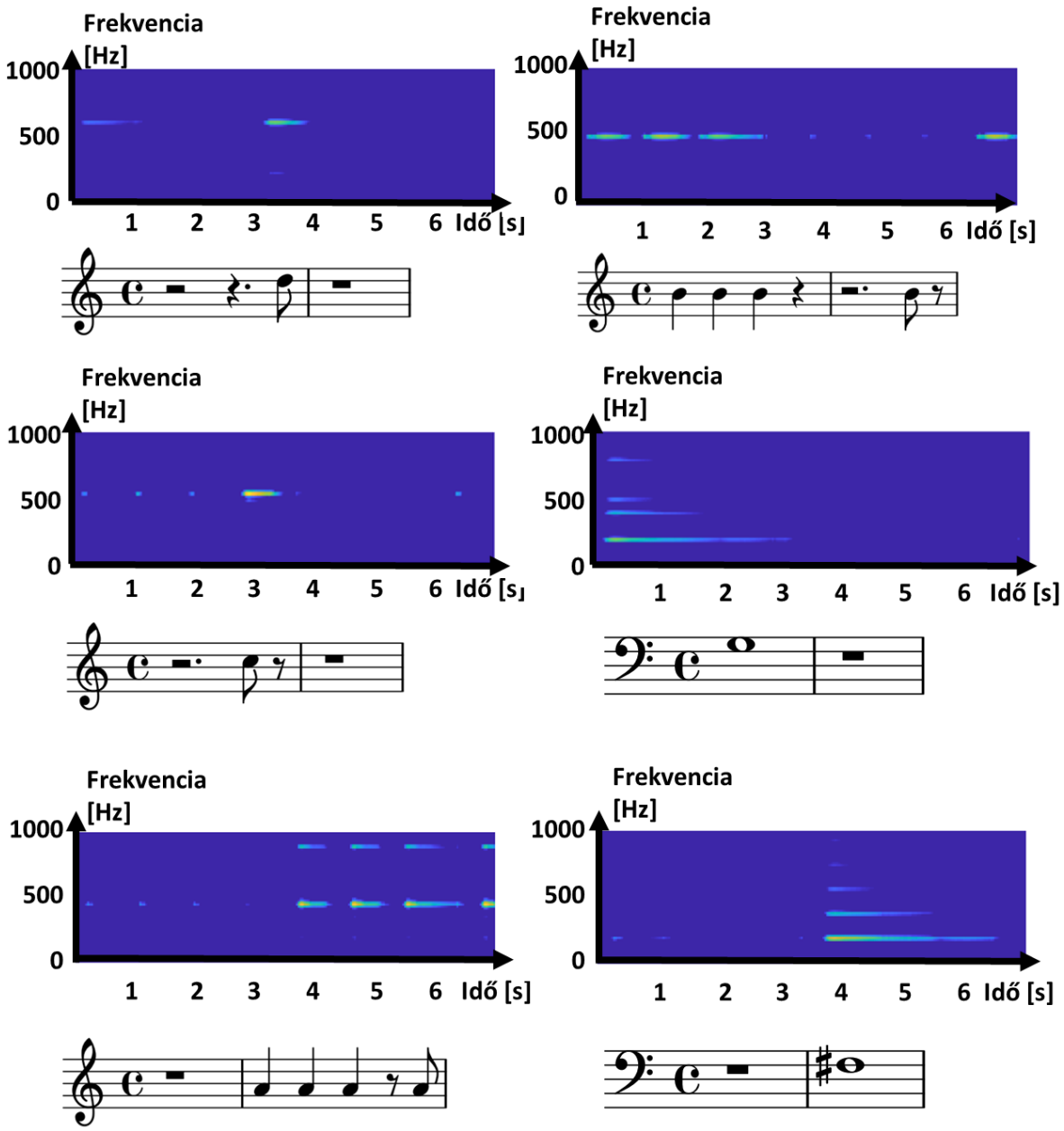
Felül az eredeti kotta, alul pedig az algoritmusom által felismert hangokból összeállított kotta látható. Ebben a tesztesetben egyszer sem lépett fel hangtévesztés, mindössze négy hangnak a kihagyását figyelhetjük meg. Az egyszólamú dallamok analizálásakor hasonló eredmények voltak jellemzők.

A generált hangminták mellett valós hangfelvételeken végeztem el teszteket. A 9-3. ábrán látható kétszólamú zongora részlet segítségével mutatom be az eredményeket.



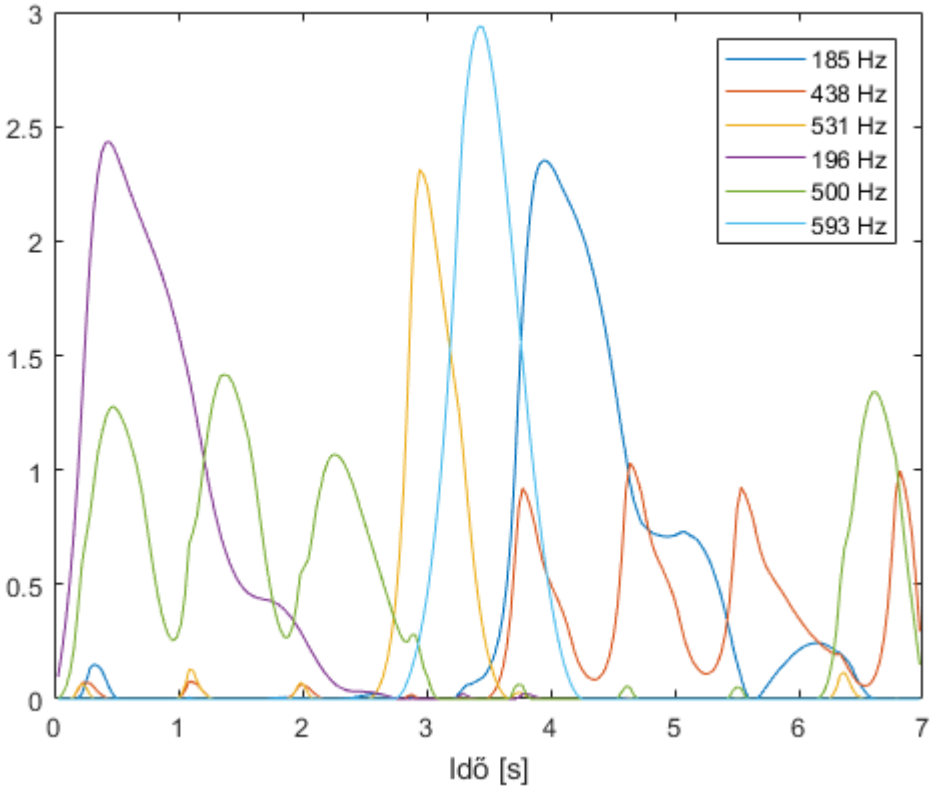
9-3. ábra – Vizsgált kétszólamú dallam

A korábbiakhoz hasonlóan lefuttattam az NMF algoritmust, melyből megkaptam a *W* és *H* mátrixokat. Már a diadikus felbontáson jól kivehetőek az eredmények (9-3. ábra). Ezeket összeadva megkapjuk a rekonstruált spektrogramot, mely megfelelően közelíti az eredetit.



9-4. ábra – Diadikus felbontás spektogramon és kottán ábrázolva

E mátrixokból már könnyedén kinyerhető valamennyi hangmagasság a HPS algoritmus segítségével, illetve azok időbeli információi, tehát az egyes időfüggvények felfutásai fogják megadni az egyes hangok megszólaltatásának időpontját (9-5. ábra). E meredek felfutások helyét legkönnyebben úgy határozhatjuk meg, ha a H mátrix sorainak vesszük az első deriváltját és abban megkeressük a lokális maximumokat.



9-5. ábra – A 9-3. ábrán látható zenerészletből generált H mátrix

A dolgozat elkészítése során létrehoztam egy olyan egyszerű programot, mely bizonyos korlátok között jó pontossággal felismeri a zenei hangokat NMF algoritmus segítségével. E algoritmus pontosságát a felbontás rangja nagyban befolyásolja, ezért a rekonstrukciós hibát figyelve megbecsültem az optimális rangot.
A felállított követelményeket az elkészült program sikeresen teljesítette, tehát képes megfelelő pontossággal hangmagasságokat felismerni és azokhoz időbeli aktivitásokat társítani, egy- vagy kétszólamú valós felvételek alapján.

Továbbfejlesztési lehetőségek:

Az algoritmust több irányba is érdemes továbbfejleszteni. Például a már felismert hangok hangsorokra való illesztésével az egyes dallamok hangnemének meghatározásával, de a hang magasságán kívül van még egy fontos paraméter, mely elengedhetetlen, ha egy kottát szeretnénk felírni. Ez a paraméter pedig a hang időtartama, vagy más néven a hangérték, hogy az adott hangot vagy hangegyüttest el tudjuk helyezni az ütemekben. Az ütemek a zenei lüktetést veszik figyelembe, és a zenét szakaszokra tagolják a könnyebb követhetőség kedvéért. Emellett a továbbiakban a hangnemfelismeréssel is lehetne foglalkozni (a már felismert hangok hangsorokra való illesztésével meghatározható az egyes dallamok hangneme).

10 Irodalomjegyzék

[AnthemScore] Audio to Sheet Music With Machine Learning, *Utolsó letöltés: 2020.10.27. https://www.lunaverus.com/*

[Bertin2007] Nancy Bertin, Roland Badeau, Gaël Richard - Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2007, Honolulu, Hawaii, United States. pp.65–68. hal-00945282*

[Chen2006] Zhe Chen, Andrzej Cichocki and Tomasz M. Rutkowski- Constrained Non-negative Matrix Factorization Method for EEG Analysis in early Detection of Alzheimer Disease. *Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute Wako-shi, Saitama 351-0198, Japan.*

[Constantini2009] Giovanni Costantini, Renzo Perfetti, Massimiliano Todisco - Event based transcription system for polyphonic piano music. *Department of Electronic Engineering, University of Rome 'Tor Vergata', Italy, 2009*

[Gisbert2005] Stoyan Gisbert, Takó Galina - Numerikus módszerek 1. *Typotex Kiadó 2005*

[Hoyer2004] Patrik O. Hoyer - Non-negative Matrix Factorization with Sparseness Constraints. *HIIT Basic Research Unit Department of Computer Science P.O. Box 68, FIN-00014 University of Helsinki Finland 2004*

[Lee2001] Daniel D. Lee, H. Sebastian Seung - Algorithms for Non-negative Matrix Factorization. *In Advances in Neural Information Processing Systems 13 - Proceedings of the 2000 Conference, NIPS 2000 (Advances in Neural Information Processing Systems). Neural information processing systems foundation.*

[Patricio2001] Patricio de la Cuadra, Aaron Master, Craig Sapp - Efficient Pitch Detection Techniques for Interactive Music. *Center for Computer Research in Music and Acoustics, Stanford University*

[PSanJuan2016] P. San Juan∗, A.M. Vidal, V.M. Garcia-Molla - Updating/downdating the NonNegative Matrix Factorization. *Department of Information Systems and Computing, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain, 2016*

[TunerApp] Five Best Tuner Apps. *Utolsó letöltés: 2020.10.27. https://bulletproofmusician.com/five-best-tuner-apps/*

[ViolinNote] Fundamental Frequency and Harmonics of a Violin Note, *Utolsó letöltés: 2020.10.27. https://www.maplesoft.com/applications/view.aspx?SID=154524*

[Virtanen2007] Tuomas Virtanen - Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on audio, speech, and language processing, vol.15, no. 3, march 2007*