DIPLOMATERV

Bódor Levente

2007

Nyilatkozat

Alulírott, Bódor Levente, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, és a diplomatervben csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

……..
Bódor Levente

Tartalmojegyzék

Abstract

Nowadays musical fountain range weaves magic out of water, light and music. Audiences of up to thousands are entralled by the spectacular show of colours and ever moving water effects synchronised to music. Unlike the traditional style with a limited range of repetitive patterns musical fountains have a tremendous variety of effects and can play beautifully with almost any type and style of music from any culture.

The aim of this thesis is to introduce some methods of the audio-based context recognition by musical signal processing. The musical context information can be used to control spreads of the fountains synchronised to classical music. Special musical parameters can be extracted by content analysis, e.g. melody timing, shape of volume and melody envelope, timing and amplitude of special pitches, tempo and phases of musical beats.

First I developed a melody envelope searcher system, which uses statistic methods to extract special information from the envelope of the wave file.

I dealt with pitch detection algorithms, and I described the calculation of pitch candidates from the Fourier transform of a signal segment where the phase information is used.

I investigated beat tracking systems. The data is processed off-line to detect the salient rhythmic events and the timing of these events is analysed to generate hypotheses of the tempo at various metrical levels. Based on these tempo hypotheses, a multiple hypothesis search finds the sequence of beat times which has the best fit to the rhythmic events.

At last I implemented graphical user interface which can help listeners to control audio-visual relationship between wave file and extracted information.

Tartalmi összefoglaló

Dolgozatomban zenei jelfeldolgozással foglalkoztam. Célom olyan értékes információk kinyerése klasszikus műfajú zenékből, amelyek aktívan segíthetik egy szökőkút zenére történő vezérlését. Manapság a szökőkutak igen változatos látványt nyújtanak, a jól megkomponált vízképnek köszönhetően. A vízsugarakat elektronikusan állítható szelepek segítségével vezérlik, így a vízi koreográfiát és az aláfestő zenét összhangba lehet hozni. Az összhang megteremtésére a zenei tartalom analízisével nyílik lehetőség. A klasszikus zenéből sokfajta paramétert nyerhetünk ki. Munkám során az ívek, motívumok meghatározását, dallamdetektálást és a tempó meghatározást végeztem el. Ezt követően létrehoztam egy ezen paraméterek ellenőrzésére szolgáló felületet.

A zenei ívek meghatározása során, empirikus úton történő fejlesztésbe kezdtem és végül egy statisztikai alapon működő, ugráskereső algoritmust implementáltam. Az algoritmus a változatos hangintenzitású zenékre kielégítő eredménnyel határozza meg az ívek kezdetét és végét.

Dallamívek leírásával is foglalkoztam, a megvalósított rendszer a frekvenciatartománybeli feldolgozást követően a fázisinformációkat felhasználva határozza meg az f_0 alapfrekvenciát az utófeldolgozás során a zöngés és zöngétlen hangokat kiszűri majd interpolálja a rövidebb kimaradó szakaszokat. Vokális előadások esetén száz százalékos teljesítmény ér el a rendszer.

A tempó meghatározása során, egy hipotézisfelállító egységet alkalmaztam a zene alaptempójára vonatkozólag, majd egy tempóillesztő egységet, amely a kialakult tempóhipotéziseket figyelembe véve fázisban illeszti a tempóütéseket a zenére. Alapvető információul a ritmikai események közti idők szolgálnak, ezeket szakértőcsoportok próbálják összeszinkronizálni a zene lüktetésével.

Mindez ellenőrizhető az implementált grafikai felületen az audio-vizuális kapcsolatot bemutató lejátszás segítségével.

A rendszer által eltárolt paraméterek felhasználásával a szökőkútvezérlés programozása egyszerűbbé és gyorsabbá válik.

Bevezető

Manapság nagyon sok helyen találkozhatunk szökőkutakkal. Vannak hagyományos állandó vízsugárral működők, vannak olyanok, amelyek a fúvókák ki-be kapcsolásával játékos előadást nyújtanak, és olyanok melyek a vízsugár erősségét módosítva, az aktívan működő fúvókák számát állítgatva szekvenciális koreográfiát adnak elő. A korszerű technika lehetőséget nyújt még változatosabb látvány előállítására. A vízsugarakat elektronikusan állítható szelepek segítségével vezérlik, így dinamikusan változó vízképek kialakítására is lehetőség nyílik. Mivel a vezérlés gyors, a vízképeket aláfestő zenével is összhangba lehet hozni. A zene és a szökőkút szinkronizálását általában a vezérlés programozásával, manuálisan oldják meg.

Ilyen szökőkutak, gyártásával, telepítésével, felprogramozásával foglalkozik a Szabados & Társai Kft. A vezérlőprogram megírása összetett probléma, a szökőkút geometriáját, a vizuális hatást fokozó beépített elemek számát szem előtt tartva olyan koreográfia megkonstruálása a feladat, amely a zenével összhangban működik. Az előadásnak követnie kell a zenében történő hangulati elemek változását és kerek egész egységet kell alkotnia.

A szelepek, szivattyúk, fények vezérlésének technológiája a kft.-nél már rendelkezésre áll. Ennek alapján a diplomaterv célja olyan automatikus eljárás kidolgozása, amely segítséget nyújt a vezérlés megalkotásában, azzal hogy a vezérlés szempontjából releváns paramétereket kinyeri.

1. Specifikáció

1.1. Motiváció

A tanszéket egy külső cég kereste meg, amely szökőkutakkal, harangjátékokkal, fényjátékokkal foglalkozik. 2003-ban új termékkel jelentek meg a piacon, ez volt a zenélő szökőkút. Ahogy a neve is mutatja, a szökőkút zenére komponált koreográfiát ad elő. Az előadott vízképet eddig kézzel programozták fel a zenei kísérthez. Ez rengeteg időt vett igénybe ezért fordultak hozzánk. A kérésük az volt, hogy egy olyan rendszert hozzunk létre, ami aktívan segíti a vízkép-koreográfia megalkotás. A rendszernek adott klasszikus műfajú zenére létre kell hoznia egy leírófájlt, amelynek segítségével látványosan, a zenei motívumokkal korrelálva lehet vezérelni a szökőkutat.

1.2. Szökőkútvezérlés

A cég a piac igényeinek megfelelő szökőkutakat gyárt. A műtermükben van egy kiállításra, tesztelésre alkalmas darab. Ezt volt alkalmam működés közben is látni. Ezen prototípus, egy hatszög alakú medencerésszel, változtatható színű megvilágítással és 15 darab fúvókával ellátott szökőkút. Minden fúvókához tartozik egy szivattyú, ami változtatható fordulatszámú ezzel lehet a vízoszlop magasságát állítani. A fúvókákban vezérelhető szelepek vannak beépítve, ezekkel a víz kiáramlását lehet szabályozni. A vezérllő egység egy midifájlból olvassa ki a vezérléshez szükséges adatokat. A midi fájlban különböző hangmagasságok különböző kimeneti csatornákhoz vannak hozzárendelve. Így például az egyvonalas ’C’ hang a középső fúvókához tartozó nagyteljesítményű motort vezérli. A midi fájlban minden egyes hangjegyhez hozzá kell rendelni a kezdés és befejeződés idejét, valamint a közben megvalósítandó dinamikát. A megvilágítást is egy-egy hangjegy vezérli és a szelepeket is. Így a midi fájlt egy sokcsatornás bemeneti leíró fájlként alkalmazva, pontosan meghatározza az időbeni működés. A midi fájl szerkesztése eddig kézzel történt, ami azt jelenti, hogy az operátor az audio formátumú zenét hallgatva időket, időtartamokat jegyzett fel, majd ezeket egy midi kottaszerkesztő programba vitte be. A kialakult vezérlés ellenőrzésére csak szökőkútra való feltöltés után nyílt lehetőség.

1.3. A feladat

Tehát a fő feladat a vezérlés ellenőrzésének megkönnyítése, és a látványos vízkép kialakításához hozzájáruló paraméterek megállapítása. A diplomamunkámban egy olyan rendszert kellett megvalósítani, ami aktívan segíti a zene feldolgozását, és ellenőrzési lehetőséget nyújt a paraméterek vizsgálatához, akár vizuálisan akár auditíven.

Létre kellett hozni egy grafikus felhasználói felületet, amely beolvassa az audio fájlt, és lehetőséget nyújt különböző zenei motívumok kinyerésére, megállapításra. A legfőbb elvárás az volt, hogy klasszikus műfajú zenékre működjenek az algoritmusok és hogy a feldolgozás offline történjen. A végső cél olyan leíró fájl létrehozása, amit segít a vizuális elemek és a zene szinkronizálásában.

Ahány szökőkút annyiféle fúvókaszám, motor beállítás, szelep beállítás, fényjáték, létezik. Emberi beavatkozás nélkül képtelenség esztétikailag elfogadható koreográfiát létrehozni. Így a feladat csak a zenei hanganyag feldolgozására irányult. Rögzítettük a fő „paramétereket", amelyek meghatározónak tűntek a zene és a vízkép összehangolását illetően. Ilyenek a zenei motívumok, **ívek, dallamvonulatok** és a **tempó**. Ezen „paramétereket" ellenőrizhető formában érzékelhetővé kellett tenni, így jutottunk el a grafikus felhasználói felület alkalmazásáig. A felület, megmutatja rendszer által kigondolt paraméterek időbeni lefutását, és az audio-vizuális kapcsolat is ellenőrizhető. A kigenerált fájlok meghallgathatóak, miközben egy marker mutatja, hol jár a zene.

2. Zenei és jelfeldolgozási definíciók

Mérő, ütés(beat), taktus: pulzusok, amelyek „nagyjából” egyenlő távolságra vannak egymástól. Megadja a zenei kotta lejátszásának idejét. Befolyásolja az egyezményes jelölés rendszerrel ellátott zenei hangok kitartásának idejét, és a köztük eltelt szünetek hosszát.

Ütési idő(beat time): az előadás kezdetétől eltelt idő, ami a mérő-ütés kezdetét jelöli.

Ütem: zenei hangok egy nagyobb csoportja, ami előírástól függően meghatározott mennyiségű zenei hangot tartalmaz. Minden ütemhez tartozik egy hangsúlyos rész, ami a mérővel esik egybe.

Ütem beosztás: a komponista határozza meg, hogy a zenei hangjegyeket mekkora egységekbe akarja tömöríteni. Így egy ütem lehet négynegyedes (4/4), háromnegyedes(3/4), kétnegyedes(2/4) beosztású, ezek a leggyakoribb szerkezetek.
Hangok időtartamára vonatkozó mértékegységek: egész, fél, negyed, nyolcad, tizenhatod, harmincketted hangok. Ezek mind azt jelölik, hogy egy egész ütem alatt(4/4) hány darab ilyen hangot lehet lejátszani, azaz az egy ütemnek hányad részében szól a kijelölt hang.

Ütemhangsúly: a zenében a hangsúly szorosan összefügg az időmértéki beosztással (ütem, taktus). Minden ütemfajtának megvan a maga szokásos hangsúlya, amelytől azonban számtalan esetben el szokott térni a zeneszerző és az előadó egyaránt. Bevett szokás, és általános szabály, hogy a páros, de egyenértékű hangjegyekre támaszkodó ütemben a hangsúly, mindig a kettős felosztás első kezdő hangjegyére esik. A 4/4-es ütemben tehát az elsőre és a harmadikra, a 2/4-es ütemben az elsőre, a 12/8-os ütemben (amely voltaképpen nem más, mint a négy negyednek 3-3 nyolcadra való osztása négy hangcsoportban) az első és harmadik csoport kapja a hangsúlyt. Ugyanilyen eljárás alá esik a 6/8-os ütem is (amely szintén nem más, mint a két negyednek 3-3 nyolcadra, mint egységre való felosztása). A páratlan egyenértékű hangjegyekre támaszkodó ütem nemeknél a hangsúlyt mindig az első hang kapja. Megjegyzendő, hogy a nyújtott ütem nemeknél, amelyek a három félkottás, három egész kottás stb. írásmódot alkalmazzák,

általában az a hangsúlyszabály az uralkodó, amely a kevesebb értékre redukálható páratlan ütem nemeknél.
A zenében a hangsúly még számos különféle kombinációjú ütemben uralkodik, amelyek külön-külön hangsúlyt igényelnek ugyan, de az alapelvet mindig a fentiekre vonatkozó hangsúlyozás képezi. A zenei hangsúly azonban távolról sem szorítkozik mindig az általánosan elfogadott szabályhoz, sőt, legtöbb esetben eltér tőle a zeneszerző szándéka és gyakran hangulata szerint olyannyira, hogy a szabályostól éppen ellenkező hangsúlyozásra fekteti a súlyt. Minél több hangszer működik együtt, a zenei hangsúlyozás annál kiterjedtebb és művészetileg bonyolultabb.
Dinamikai szempontból a zenei ütem hangsúlya többféle típusra oszlik. Hol csak egyes hangok, hol meg egész hangcsoportok esnek hangsúlyozás alá.

Metrikus előadásmód: Szigorúan, a hangjegyek hosszára vonatkozó előírások betartásával, játszva egy darabot. Minden negyedhang egyenlő hosszú és minden félhang duplája a negyednek.

Expresszív előadásmód: Emberi előadásmód, amely nem olyan szigorú, mint az előző. Érzelmi hatásokat közvetítő kifejezésmód.
Tempó: a kották gyakoriságát határozza meg. Mértékegysége lehet, például adott számú negyedhang szekundumonként vagy a percenkénti ütések száma(beats per minute). Ehhez hasonló egység az ütések közti intervallumot rögzítő egység.
A tempó lehet átlagos tempó, amely a teljes zenei előadásra jellemző. Lehet alaptempó amely körül az expresszív előadás alatt a tempó ingadozik. (Rep, 1994)

Tempó becslés: Az alaptempó becslése, az algoritmus különböző feltevéseket generál a mérő-ütések közti időt illetően és ezekből választ a rendszer.

Tempó illesztés: a mérő ütések ráillesztése a zenei hangsúlyokra.

Sorszerkezet: A zenei dallamokra jellemző, zenei sorok valamilyen szabályosságot követnek. A sorszerkezet ezt hivatott leírni. Például AABA, ABBA

Szubtraktív klaszterezés: Olyan eljárás, ahol a mintapontok csoportosítása önszervező alapon történik. A klaszter középpont körül adott sugarú tartományban a mintapontokat egy csoportba sorolja, a már csoportosított mintákat kivonja az eredeti mintahalmazból és iteratívan csak a megmaradtakkal foglalkozik tovább.

K-means algoritmus: Olyan iteratív klaszterező algoritmus ahol, adott számú klaszter létrehozása a feladat és ezek középpontjai a mintapontok besorolásával folyamatosan módosulhat.

Oktávhiba: A hangmagasság és dallamkereső algoritmusok, legtöbbször előforduló hibája hogy az alapfrekvenciaként megjelölt frekvencia, a valódi frekvenciának (2 pozitív vagy negatív egészszámú kitevőjű) többszöröse

3. Zenei motívumok elkülönítése

3.1. Statisztikán alapuló eljárás, empirikus algoritmusfejlesztés

3.1.1. Bevezetés

A legfontosabb és legkönnyebben érzékelhető zenei motívum az, ami látványosan változik a zene lejátszása alatt. Ilyen a hang intenzitása, hangereje, valamint a dallam lefutása, alakja, íve. Nagyon sok műfaj támaszkodik ezekre a paraméterekre. Vannak bizonyos műfajok, amelyek az intenzitásra és a sorok szerkezetére vonatkozóan szabályokat írnak elő, s ezeket a szerzőnek be kell tartania. A teljesség igénye nélkül a következőkben komolyzenei műfajok rövid meghatározását írtam itt le, az eredeti definíciók a Wikipedia honlapján részletesebben olvashatóak(Wikipedia, 2007).

Canzona: A zenében egy 16-17 századi többszólamú instrumentális kompozíció. Nevét onnan nyerte, hogy eredetileg francia és flamand versek szövegére íródott. Ez és a fantázia vezettek a későbbi szonáta kialakulásához.

Concert grosso: Versenymű egy speciális válfaja ahol több szóló hangszer van és ezek kis együttest képezve versenyeznek a zenekarral. A barokkban igen kedvelt a halk-hangos váltás.

Fantázia: Barokk és klasszikus zenében a fantáziák általában billentyűs hangszerekre íródtak, gyors futamokkal és némi fugatikus elemmel. Gyökerei az improvizációban keresettek.

Francia nyitány: Három forma részből áll, egy lassú bevezető részből, melyre a pontozott ritmusok a jellemzőek (akár a dupla-pontozás is), egy gyors, fugato középrészből és egy lassú visszatérésből. Ha a mű zenekarra íródott mindenképpen játszott benne trombita és üstdob, valamint gyakran előfordulnak a fafúvós hangszerek, illetve kürtök. Ünnepélyes előadásmód.

Fúga: A fúga név a latin *fugere* (kergetni) szóból keletkezett, mert lényege abból áll, hogy több szólam egy rövid zenei frázist vagy hosszabb-rövidebb tételt több száz ütemen keresztül, minden – főleg – rokonhangnemben ismétel. Egyik a másikat pihenés nélkül kergeti, hajszoljam mialatt a többi szólam hozzájuk a kíséreti szólamokat szolgáltatja. Legtökéletesebb formai szerkezete a négy szólam (szoprán, alt, tenor, basszus). Szabvánnyá az vált, hogy a fő dallam szólamot csak egy másik szólam utánozza, különböző de mindig rokonhangnemben, a többi pedig azalatt a főszólamtól eltérő kísérőszólamot csatoljon hozzá oly formán, hogy a 3 vagy 4 szólam mindig szabatos harmónia tételeket képezzen. Szerkezete: vezérszólam kezd egymagában, 2-3 taktus de lehet akár 10-12 taktusból álló periódus is, de a későbbiekben a taktusszámot a többi szólamnak szigorúan be kell tartania. Ha a dallammotívum befejeződik, akkor kezd rá a másik szólam más hangnemben, és az eredeti szólam kíséretül szegődik. Mielőtt a főtémát utánzó szólam felvenné a fonalat az átvezető szakaszon az összes szólam belehúz és hidat képez az új szólam belépéséhez.

Korál: Protestáns egyházi népének, mely legtöbbször egyszólamú. Korálokat verses szövegük ritmikája, sorszerkezetük és rímelése jól érzékelhető korálsorokra, vagy szakaszokra tagolja. Dallamuk sokszor AAB formát ölt. Könnyen énekelhető, kevés a módosított hang ritka a nagy hangköz-ugrás.

Korálprelúdium vagy korálelőjáték: Liturgikus kompozíció orgonára, amit egy korál dallamra komponáltak. Röviden bemutatja az elkövetkező –nép által énekelt- dallamot.

Oktett: 8 szólisztikusan koncsertáló hangszerre írt zenemű.

Oratórium: Az operához hasonlóan, szólóéneket, kórust és zenekart foglalkoztató drámai hatású kompozíció. Tárgya többnyire bibliai eredetű. Bevezette: Handel
Passió olyan sajátos oratórium, amely Krisztus szenvedésének történetét beszéli el.

Partitia: Eredetileg szimpla hangszeres mű, később a szvit szinonimája.

Prelúdium, vagy előjáték: Olyan rövid karakter darab, ami egy nagyobb lélegzetvételű mű előtt felvezetésül szolgál.

Quodlibet: Olyan műfaj, amely kontrapunktban több különböző dallamot kombinál, általában könnyebb hangvételű dallamot műveket.
Ricercar: A fúga korai változatát nevezték így, illetve, olyan fúgákat, ahol a fúgatémák többnyire hosszú hangokból állnak.

Sonata; chiesa és de camera:
A szonáták hangszerekre íródott zenei művek, három vagy több tétellel. A de chiesa inkább visszafogottabb, így jobban alkalmazkodik a templomi körülményekhez. A de camera (kamara szonáta) elevenebb és tánctémákat dolgoz fel.

3.1.2. Módszerek kiválasztása

Az eddigi meghatározásokból látszik, hogy a sorok szerkezetéből ,a dallam ismétlésekből, a halk és hangos részek változásából akár műfaj becslést is végre lehet hajtani. A sorok szerkezetével és a hangerősséggel legjobban és legkézenfekvőbb módón a hangfájlok burkolója korrelál. Ezt a zenei paramétert használja az alábbiakban ismertetett rendszer. A feldolgozás időtartományban történik. A rendszertől nagyfokú önállóságot vártak el a megbízók, ezért arra törekedtem, hogy az algoritmus paraméterek beállítása nélkül találja meg a dallam íveket.
A szakirodalom feldolgozását az intelligens módszerekkel kezdtem, mert a mesterséges intelligenciát használó algoritmusok hatékonyak mintakeresésben, különböző csoportok kialakításában. Abból indultam ki, hogy ha az ember rátekint egy klasszikus mű teljes burkolójára, meg tudja mondani, hol vannak az egységbe tartozó zenei szakaszok. Először a teljes regisztrátum amplitúdóinak számtani átlagát felhasználva próbáltam részekre bontani a burkolót, ez sajnos nem járt kielégítő eredménnyel, mert van nagyon sok olyan mű, ami egy ív alatt húzódik és csak kicsiny modulációk vannak ezen ív lefutása közben. Tehát a halk és hangos részek változása magához az ív globális változásához képest elenyésző. Az ilyen esetekben az átlagoló eljárás során három, négy egység keletkezet. Azt is figyelembe kellett vennem, hogy nem lehetnek túl rövid ívek, és túl hosszúak sem. A rövidek nem használhatóak egy klasszikus zenével vezérelt

szökőkút, tetszetős vízképének kialakításában, mivel a vezérelendő motorok meghatározott tehetetlenséggel rendelkeznek, a túl hosszúak pedig unalmasak.

A következő módszer amire gondoltam, a neurális hálók alkalmazása. Ezzel az a probléma, hogy a kielégítő eredmény előállításához nagyon sok processzáló elemet, nagy számítási kapacitást igényel, mert rengeteg bementtel kell rendelkeznie a hálónak és a kimenetnek is nagyon sokféle variációs lehetőséget kell tudnia megjeleníteni. A hagyományos klaszterező eljárások sem megfelelőek, mert csak külön paraméterek megadása estén szolgáltatnak jó eredményt. Tehát ha nem tudjuk előre hány ív lesz egy zenében, hány halk-hangos blokk, vagyis a k-means algoritmus 'k' paraméterét nem ismerjük, akkor az algoritmus használhatatlan.

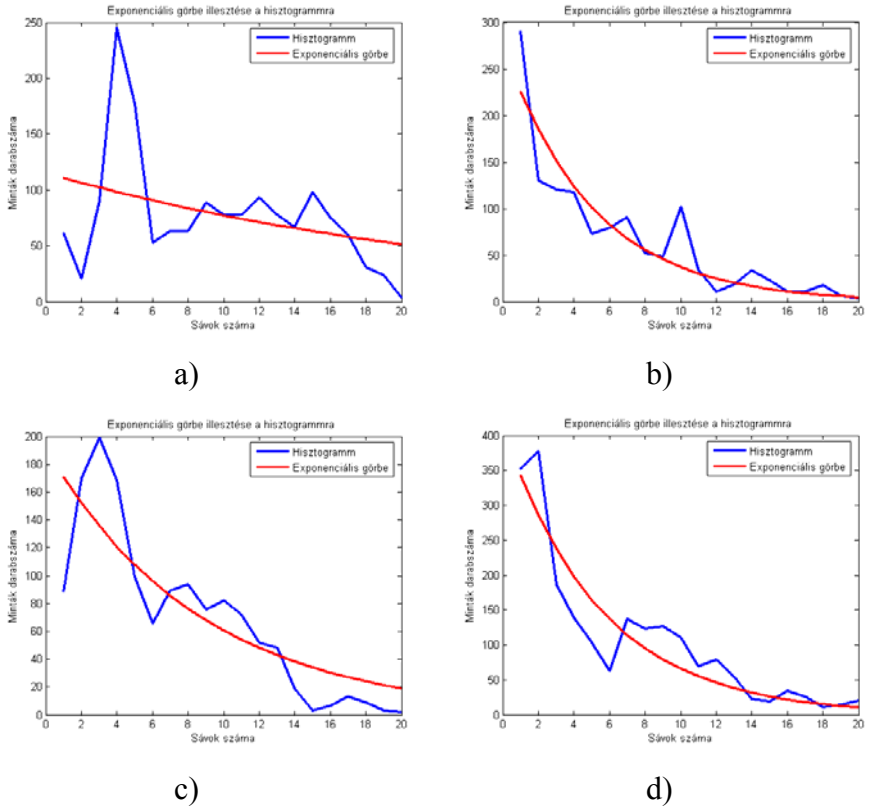
A próbálkozások során az derült ki, hogy a rendszer bemenetén egyszerre meg kell jelenni a teljes burkolónak. Addig nincs értelme elkezdeni a magasabb szintű feldolgozást, míg globális paramétereket nem kerültek megállapításra. Az ember, amikor ránéz egy képre a teljes tartalmat megvizsgálja, automatikusan hasonló mintákat, íveket keres. Egy függvény esetén, annak maximumát és minimumát emeli ki. A zene hullámformájának burkolója is tekinthető függvénynek. Mintákat ismerhetünk fel a sorok ismétlődésében, a sorszerkezet kialakításában, a visszatérő dallamívekben, valamint a hangerő váltakozásában. Ez adta az ötletet arra, hogy mivel a bemenetet (burkoló görbe) egyszerre, egészben kell feldolgozni, valamint szükség van a hangszintek elkülönítésére, statisztikai módszereket alkalmazzunk a mintapontok (a burkoló pontjai) csoportba sorolásához. Ezek után kézenfekvő volt, a burkolót bizonyos számú szintre osztani és megnézni hány elem tartozik egy sávba. A hisztogram lefutása, a dinamikus, hangos halk részeket váltogató zenéknél, (mint amilyen az 5. Magyar tánc, vagy a Suppe- Könnyűlovasság) általában exponenciális görbét követ. A hisztogrammot alapul véve a program meghatározza a két legvalószínűbb hangerő szintet. A rendszer ezt a bemenő paramétert használja, majd ez alapján, a burkolóban íveket keres. Az ívek hosszától függő utófeldolgozás során, ki kell válogatni az egy zenei motívumhoz tartozókat. Ezt követően tárolni kell az ívek paramétereit (elejét és végét). Ezek az ívparaméterek, segítik az operátort a vízkép koreográfia váltásának időzítésében.

3.1.3. Megvalósítás

Az ívkereső algoritmus lelke egy olyan függvény, ami egy előre megadott küszöb átlépését vizsgálja egy adott méretű tartományban. A jelalakban az algoritmus ablakonként hajt végre vizsgálatot. Az ablak hossza bemenő paraméter, amelyet empirikus úton állapítottam meg. A legjobbnak az egynegyed másodperces ablakhossz bizonyult. Így minden másodpercben 4 szer vizsgálja meg azt, hogy történt e a küszöbnél nagyobb ugrás vagy sem. Feldolgozás előtt a kiugró értékek eltávolítása érdekében egy 3 szekundumos ablakkal rendelkező medián szűrést hajtunk végre. Erre azért van szükség, hogy a minimum és maximum értékekre figyelő algoritmus minden körülmény között jól döntsön az ív le vagy felfutását illetően, valamint, hogy a nagy frekvenciás változások ne okozzanak zavart. Miután az egység megvizsgálta, hogy az adott ablakban lévő minimum és maximum értékek különbsége meghalad-e egy küszöböt, már csak azt kell ellenőrizni, hogy a minimum vagy a maximum érték volt-e előbb, ebből következik, hogy az ív felfutó vagy lefutó.

```
function [vag_kezd,vag_vegz]=ugraskereses(ugrás_ablak,küszöb,t2);
vágás_kezdete: az ívek kezdete
vágás_vége: az ívek vége
t2: bementi adat vektor
h0/j0: maximum értéke/indexe
h1/j1: minimum értéke/indexe
Ugrás_keresés (ugrás_ablak, küszöb, t2);
CIKLUS amíg t2 végére nem érünk
        [h0,j0]=max(t2(a:(a+ugrás_ablak)));
        [h1,j1]=min(t2(a:(a+ugrás_ablak)));
        HA h0-h1>küszöb AKKOR
                HA j0>j1 AKKOR
                        vágás_kezdete(a)=t2(a);
                KÖLÖNBEN
                vágás_vége(a)=t2(a);
                HAVÉGE
        a=a+ugrás_ablak;
        HAVÉGE
a=a+1;
CIKLUSVÉGE
```

A legfontosabb kérdés a küszöbparaméter meghatározása. Erre egy teljesen egyéni módszert alkalmaztam, a megfigyeléseimet követve. A küszöb paraméterről tudjuk azt, hogy ez határozza meg, hogy a jelben mely ugrásokat tekintjük ívkezdetnek vagy végnek. Ez a burkoló pontjaiból számolt hisztogram lefutása alapján határozható meg. Jellemzően, a több hangerősség szinttel rendelkező műveknél, ha a burkolót 20 sávba osztjuk, az eloszlás görbe exponenciálishoz közelít, mégpedig úgy, hogy a kis intenzitású pontokból sok van a nagyokból pedig kevés.



1. ábra. a) 5. Magyar tánc.wav; b)12catacomb.wav; c) 13.wav; d) Bartók - Allegro Barbaro.wav hisztogramjai

A hisztogramokban a két legnagyobb kiemelkedés, a két legtöbbet hallható hangerő szintjét mutatja. Ezen kiemelkedések megtalálására a következő függvényt implementáltam, mely a szubtraktív kalszterezés alapgondolatát, csúcskeresést és heurisztikát alkalmaz. A heurisztika az volt, hogy a csúcsok közötti fontossági sorrendet az határozta meg, hogy melyik milyen messze van az illesztett exponenciális görbétől.

Először a hisztogramra egy legkisebb négyzetes hibával számolt exponenciális függvényt illeszt az algoritmus, majd ezt felhasználva, kiszámolja a két legkiemelkedőbb csúcsot az exponenciális görbéhez képest.

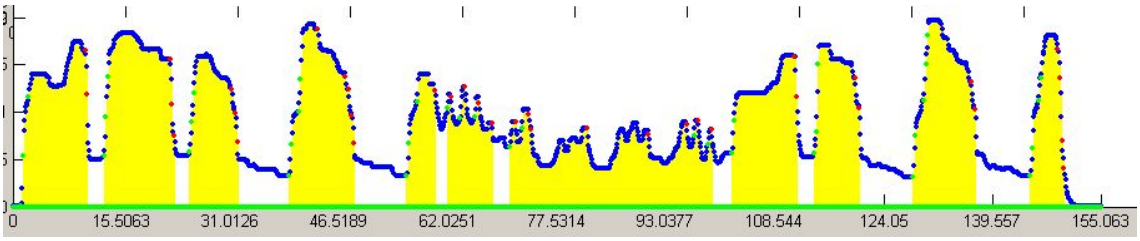
```
hangszintrendezo(data,Illeszett_görbe1,hang_szint)
  CIKLUS u = 1 : hang_szint
      [mennyi,hol]=Legnagyobb eltérés a görbétől
      data(hol)=0;
      hol_p=hol;
      CIKLUS kilépés ha újabb csúcs jön
            data(hol_p)-tól jobbra keressük a kisebb értékeket
            data(hol_p)=0;
            hol_p=hol_p+1
      CIKLUSVÉGE

    hol_m=hol;
      CIKLUS kilépés ha újabb csúcs jön
            data(hol_m)-től balra keressük a kisebb értékeket
            data(hol_m)=0;
            hol_m=hol_m-1
      CIKLUSVÉGE
  Kimenet(u,1)=hol
  CILUSVÉGE
```

Az algoritmus sorba rendezi az hisztogram értékeit aszerint, hogy mennyire távol vannak az illesztett görbétől. Aztán egy csúcskeresést hajt végre. A csúcs környékén lévő, a csúcs lefutásához/felfutásához tartozó értékeket és a csúcsot kivonja az adathalmazból majd iterratívan újra lefut, felállítva ezzel a csúcsok nagyság szerinti sorrendjét. A főprogram, az első két helyre rangsorolt csúcs relatív távolságának bizonyos tört részével számol tovább. Empirikus úton ez egy-tized lett, persze ez függ a küszöböt alkalmazó ablak hosszától is.

Ezek után az ugráskereső algoritmus hívódik meg, a kiszámolt küszöb paraméterrel és a burkoló mediánszűrőn átengedett jelalakjával.

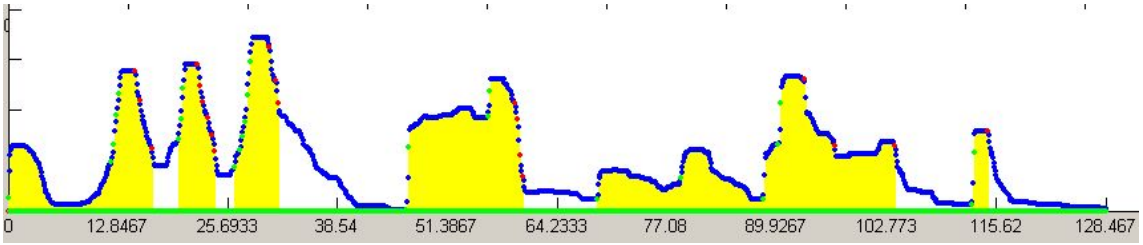
Végül utófeldolgozásra van szükség, mivel gyakori, hogy túl sok kis ívet talál az algoritmus. Ezért egy szegmentációra van szükség, nem létezhet 5 másodpercen belül két ív vég vagy kezdet.

A következő ábrákon a rendszer tippjei tekinthetőek meg az ívek kezdetére és végére vonatkozóan, ahol egy sárga blokk egy jósolt ívet jelöl. Az ábrák a már elkészített felhasználói felület segítségével lettek generálva, ahol a vízszintes tengelyen az idő másodpercben a függőleges tengelyen a burkoló intenzitása látható.



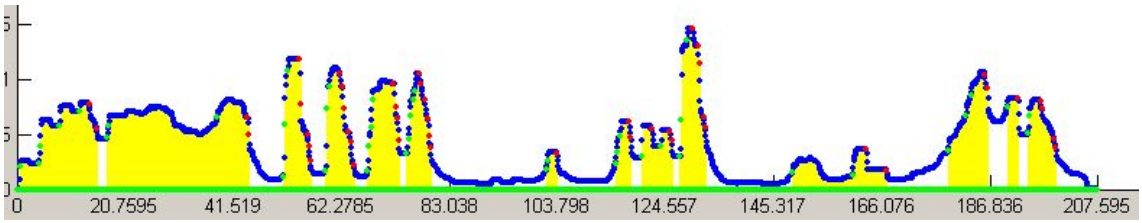
2. ábra. 5. Magyar tánc motívumai.

Ennél a műnél a rendszer teljesen pontosan tippel, az elkülöníthető íveket egytől egyig megtalálja.



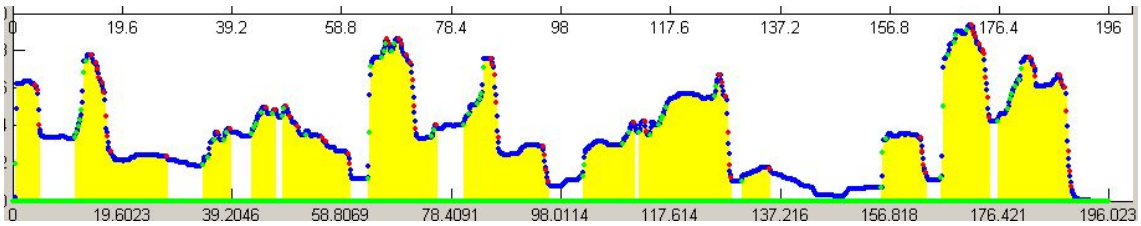
3. ábra. 12Catacomb.wav motívumai.

Itt is a főbb egységeket elkülöníti, persze ezen eredmények a megfelelő vezérléshez további pontosítást igényelnek. Az egy ív alá tartozó eseményeknél, nem pontos az ív vég meghatározása. Például a 4. csúcs esetében látható egy fokozatos lecsengés egészen a 0 intenzitás szintig. Ezt a rendszer jóval rövidebbnek becsüli, mint amilyen valójában.



4. ábra. Bartók-Allegro Barbaro.wav motívumai.

Ebben az esetben több ívet különít el a rendszer, mint amit feltétlenül szükséges. Ez persze a feldolgozás szempontjából jó, mert a későbbiekben megadhatjuk majd a leírófájl feldolgozásakor, hogy mely motívumkezdetekre van szükség.



5. ábra. Copland Rodeo.wav Motívumai.

Ebben az algoritmusban többféle heurisztikát lehet alkalmazni, s így egyre elfogadhatóbb eredményt lehet elérni. Továbbfejlesztési lehetőség hogy minden egyes ív burkolójának alakját további vizsgálatnak vetjük alá. Vizsgálhatjuk, hogy milyen görbékkel lehet lefedni az íveket, valamint a nem ívnek jelölt burkoló részek, nem folytatása-e a szomszédos ívnek. Ezzel ki lehetne szűrni azt a gyakori hibát, - ahol jelenleg gyakran téveszt a program- hogy, ahol folyamatos hosszú távú emelkedés vagy esés van, ott csak a legmeredekebb részeket definiálja ívként a program, így a lecsengő kevésbé meredek részeket már nem veszi hozzátartozónak.

A motívum kereső algoritmus a kimenti fáj egyik csatornáját generálja, amelyben pontosan jelzi az ívek kezdetének és végének idejét. A felbontása állítható, az alapértelmezett érték 100ms.

3.2. Frekvenciatartománybeli analízis, hangmagasság detektálása

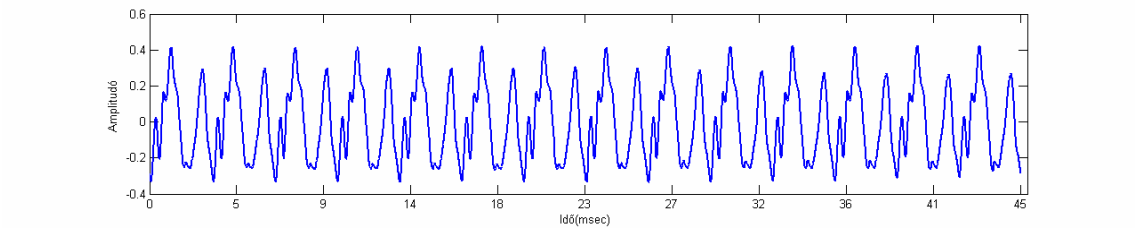
3.2.1. Harmonikus hangok, alapfrekvencia, hangmagasság

A hangmagasságot úgy tekinthetjük, mint egy hanghoz rendelt skálát, amin a beosztások alacsonytól a magasig helyezkednek el. Pontosan ezért egy kevés felharmonikust tartalmazó (egyszerű) 500 Hz-es hangot magasabbnak hallunk, mint például egy 400 Hz-es egyszerű hangot. Ebből látható, hogy valamilyen kapcsolat van a frekvencia és az érzékelhető hangmagasság között. A hangmagasság korrelál a zenei hang alapfrekvenciájával, amit f_0-al jelölünk.

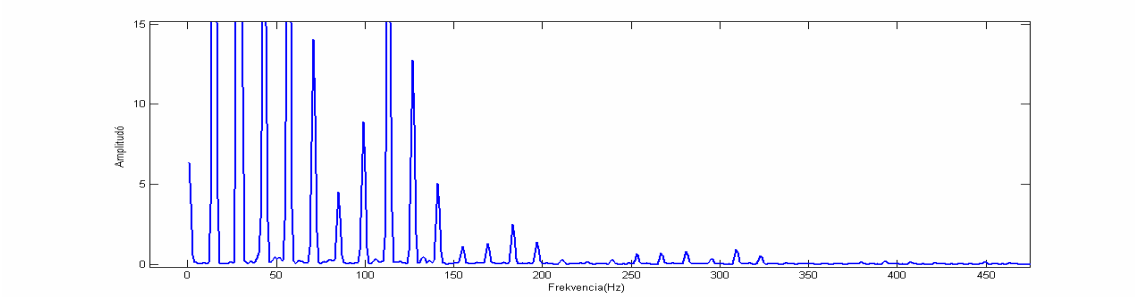
Fontos különbséget tenni a zenei hangok fizikai és az érzékelhető tulajdonságai között. A fizikai tulajdonságok műszerekkel mérhetőek, az érzékelhető tulajdonságok függenek a hallgató asszociációs képességeitől. Néhány - a hallgatók számára - érzékelhető tulajdonság erősen összefügg a fizikai paraméterekkel, mint például a hangmagasság és az alapfrekvencia.

A harmonikus hang az a hang, amit le tudunk bontani olyan összetevőkre, amelyek az f_0 frekvencia egész számú többszörösei. Ezek a többszörös összetevők a felharmonikusok. A harmonikus hangok periodikusak.

A harmonikus hangok spektrális szerkezetére jellemző, hogy az f_0 frekvencia és a felharmonikusok szabályos távolságra helyezkednek el egymástól.(saxofon.m)



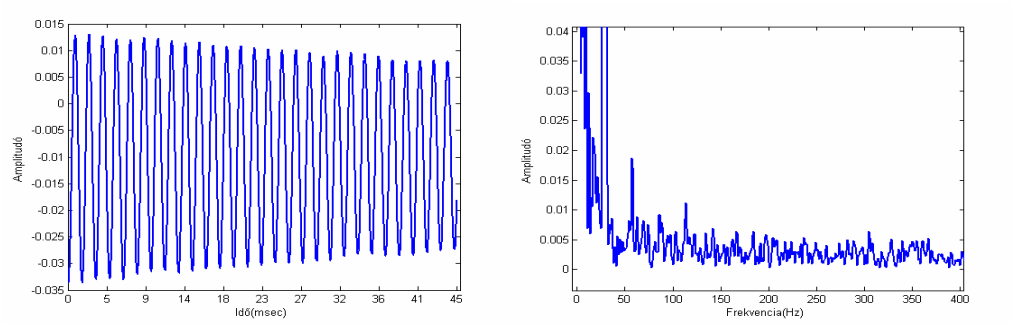
6. ábra. Szaxofon hangjának jelalakja



7. ábra. Szaxofon hangjának spektruma

Az időtartománybeli jelalak nem szigorúan periodikus, ugyanis a valódi hangszerekből kicsalt harmonikus hangok periódusai kis mértékben eltérhetnek egymástól. Ez főleg a fafúvós, húros, nádsípos, rézfúvós hangszerek hangjaira jellemző. Nevezzük ezt a periodicitást ezentúl pszeudo-periódicitásnak.

Ezzel szemben a nem harmonikus hangok esetében, mint például a membránós hangszerek által keltett hangok, nincs periodikusság a jelalakban, itt nincs kitüntetett f_0 frekvencia sem. Ezeknél a hangszereknél zenei hangmagasságról nem beszélünk.
Vannak olyan hangszerek, amelyeknél egy kalapácsszerű eszköz ütése szolgáltatja a hangot, mint például a xilofon és a marimba, ezeknél a jelalak közel periodikus de a spektrális szerkezet teljesen szabálytalan.



8. ábra. Xilofon hangjának jelalakja és spektruma

Általában itt a felharmonikusok száma nagyon kicsi és az f_0 frekvencia egész számú többszöröseinek közelében helyezkednek el.

Az ideális harmonikus hangok esetében a felharmonikusok frekvenciája pontosan az f_0 frekvencia egészszámú többszöröseinél helyezkedik el. A valós harmonikus hangoknál a komponensek frekvenciája eltér az elméleti értéktől (nem ideális harmónia). Például a feszített húros hangszereknél (zongora) a magasabb rendű felharmonikusok eltolódnak egy fix értékkel felfelé (Klapuri A. P., 2004). Az ideális harmóniától való eltérés, tesz minden hangot más hangszínűvé. Tökéletes harmonikus hang a gyakorlatban nem is létezik.

Az érzékelhető hangmagasság legjobban az alapfrekvenciától függ, de ezen kívül függ még a hang intenzitásától, a környezeti adottságoktól és a hallgatótól.

Az észlelt hangmagasság egy oktávnyi növekedéséhez, az alapfrekvencia duplázódása tartozik. Így az összefüggés logaritmikus. Azonban 1000 Hz alapfrekvencia felett, az f_0 duplázódás kicsit kevesebb, mint egy oktávnyinak hallatszik.(Gerhard D., 2000)

A hangmagasság és f_0 közötti összefüggés változhat az intenzitás és a felharmonikus tartalommal is. Például más hangamagasságúnak tűnik egy hang, ami tökéletesen ideális felharmonikusokat tartalmaz, mint aminek a felharmonikusai nem pont az ideális helyen vannak.(Gerhard D., 2003)

Az alapfrekvencia és a hangmagasság közti különbség legérdekesebb példája, a hiányzó alapharmonikus eset. Amikor előállítunk egy hangot úgy, hogy csak a harmadik és ötödik felharmonikusát tartalmazza az f_0-nak, akkor a szintetizált hang megszólalásakor hallható az f_0 frekvencia. Ez azt mutatja, hogy legalább annyira fontos egy hangmagasság érzékelésnél a spektrum szerkezete, mint az alapharmonikus.

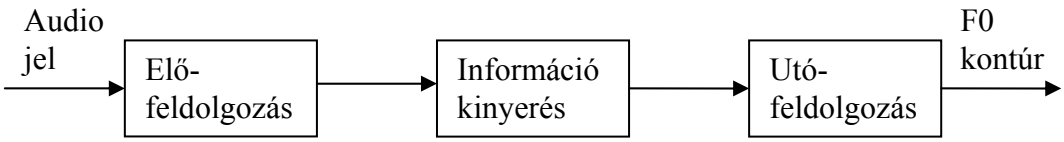
Az emberi hallórendszer megpróbálja összerendelni az akusztikus jeleket bizonyos hangmagasságokkal, akár harmonikusak akár nem. Pszeudo-periódikus jeleket és a

zajokat is képes a hallórendszer speciális frekvenciájú szinuszos jelnek érzékelni. Például ha veszünk egy zajt, és amplitúdó moduláljuk, akkor érzékelhető lesz benne valamilyen hangmagasság, ami korrelál a moduláló jel frekvenciájával. A hallórendszer képes olyan hangokhoz is hangmagasságot rendelni, amik se nem periodikusak, se nem szabályos spektrum struktúrájúak. Ilyen hangokat ad ki a harang, és a membrános hangszerek (Klapuri A. P., 2004).
A módszerek kiválogatásában és tanulmányozásában a következő kivonatok voltak nagy segítségemre: (Klapuri A. P., 2004; Rui Pedro Pinto de Carvalho e Paiva, 2006; Emilia Gómez Gutiérrez, 2001, Udo Zolzer. DAFX: Digital Audio Effects)

A továbbiakban a dolgozat a tiszta alap-harmonikust tartalmazó, hangmagassággal rendelkező hangokkal fog főleg foglalkozni.

3.2.2. Hangmagasság detektálás

Zenei tartalom analizálásával, visszafejtésével, dallamkereséssel, hangmagasság megállapításával számtalan tanulmány foglalkozik. Nagyon bő irodalma van az egyszólamú környezetben történő feldolgozásnak, mivel beszéd analízisben és zenei tartalom kinyerésében is az egyik legjelentősebb eszköz. Az egyszólamú környezetben történő hangmagasság detektálást részben megoldott problémának tekinti a szakirodalom, sok algortimus és valós idejű eljárás létezik. A problémás területek közé tartozik, az énekhangban az oktáv hiba felismerése.
Manapság inkább a többszólamú környezetből történő hangmagasság kinyerés az aktuális probléma (Klapuri A. P., 2004; Sterian A. D., 1999; Kashino K., Nakadai K., Kinoshita T. and Tanaka H., 1995).
Az egyszólamú és többszólamú környezetből történő hangmagasság kinyerésére alkalmas eljárások 3 fő részre bonthatóak. Elő-feldolgozás, információ kinyerés, utó-feldolgozás.



9. ábra. Az audio jel hangmagasság detektálásának menete

Azt elő-feldolgozó egység feladata hogy csökkentse a bementi adat mennyiségét ezzel megkönnyítse a középső egység munkáját. Az elő-feldolgozás elsősorban a zajok elnyomására és azon tulajdonságok kiemelésére szolgál, amik pontosabbá teszik a detektálást.

Az információ kinyerésére szolgáló egység az egész hangmagasság detektálás lelke, tulajdonképpen az időszeletekből kinyert fázis információk alapján határozza meg az alapfrekvenciát.

Az utó-feldolgozó egység feladata a hiba észlelés és korrekció. Használatával az f_0 kontúr kevesebb zajt tartalmaz. Ezt úgy éri el, hogy elkülöníti a zöngés hangokat, kiszedi a jelből a zöngétleneket, és a kimaradó időkben interpolál.

3.2.2.1. Hangmagasság detektálása egyszólamú környezetben

Az első kísérleteket a zenei hangmagasság megállapítására a beszéd felismeréssel foglalkozók tették. A mai algoritmusok már speciálisan a zenei hangfelismerésre vannak kifejlesztve. A zenei jeleknek van némi specialitásuk a beszédjelhez képest, például a hangterjedelmük sokkal szélesebb, és a spektrális tartalmuk is sokkal változatosabb. A diszharmonikus jelek jelentősége is igen nagy.

Ilyen algoritmusokat már az 1960-as években is fejlesztettek. (Noll, 1967; Gold and Rabiner, 1969; Lahat et al., 1987; Talkin, 1995; Clarisse et al., 2002). A beszéd feldolgozásban nem létezik univerzális megoldás, ezért született sokféle speciális esettel foglalkozó munka. Vannak, akik az énekelt hangok detektálásával foglalkoztak. (Ryynänen, 2004; Viitaniemi et al., 2003; Clarisse et al., 2002). Vannak munkák,

amelyek robusztus megoldást keresnek az oktáv hiba kiküszöbölésére, és vannak valós idejű felhasználásra fejlesztett algoritmusok.
A beszédhang feldolgozása esetén a fő probléma a zöngés és zöngétlen hangok megkülönböztetése. A zöngés hangok általában magánhangzók, ezek periodikus hullámformával rendelkeznek, ezért ezek igen közel állnak a zenei hangokhoz és könnyebb őket analizálni. A mássalhangzók jellemzően zöngétlen hangok (kivételt az 'm', 'n', 'l' hangok képeznek, amelyeket zöngések), ezeket nehéz feldolgozni, mivel jellegükben a zajhoz hasonlítanak. Az éneklés alatt a zöngés hangok a dominánsak de jelen vannak a zöngétlenek is, amelyeknek főleg ritmikai szerepük van.

Különböző kategóriákba sorolhatjuk az algoritmusokat aszerint, hogy hogyan kezelik a spektrális információkat. Ez alapján megkülönböztetjük a **spektrum elhelyezkedését** szem előtt tartó, a **spektrum intervallumaira összpontosító**, és az **egységes algoritmusokat**. Az első a harmonikus komponensek elhelyezkedését vizsgálja az alapharmonikushoz képest, a második a résztvevő frekvencia komponensek távolságát, a harmadik kompromisszum a kettő között.

3.2.2.1.1. Spektrális elhelyezkedést vizsgáló algoritmusok

Ezen módszerek harmonikus összetevőkre vonatkozó mintakeresésen vagy hullámforma periodicitáson alapulnak.

Autokorrelációs függvény(ACF)
Az egyik leggyakrabban használt időtartománybeli eljárás, azt mondja meg, hogy a jel mennyire hasonlít önmagára minden egyes pontjában.

$$r[i] = \frac{1}{N} \sum_{k=0}^{N-i-1} x[k] * x[k+i] \qquad (1)$$

Ahol N a jel hossza mintaszámban kifejezve. r[i] az autokorrelációs függvény értéke a i-től függően. Az alapfrekvenciához tartozó periódusidő egyenlő az r[i] függvény maximummának helyével az i=0 indextől nagyobb értékekre.

Az ACF még hatékonyabban számolható a frekvencia tartományban, a Fast Fourier Transform segítségével(FFT). Először frekvencia tartományba transzformáljuk a jelet, majd az amplitúdó-spektrumot megszorozzuk komplex konjugáltjával, végül visszatranszformáljuk újra időtartományba.

$$r[i] = FFT^{-1}\left|X[k]\right|^2 \qquad (2)$$

Általában ezt a megoldást szokták alkalmazni az időtartományban történő konvolúció helyett, mivel ez gyorsabb.
Maga az autokorreláció arra jó, hogy kiemelje a harmonikus frekvenciákon elhelyezkedő amplitúdókat.

$$r[i] = \frac{1}{N}\sum_{k=0}^{N-1}\cos(\frac{2\pi ik}{N})\left|X[k]\right|^2 \qquad (3)$$

Alapvetően ez az egyenlet fejezi ki a pontos működést, amikor az i egyenlő a jelalak periódusával akkor az amplitúdó-spektrum négyzete maximálisan súlyozódik.
Az ACF alapú hangmagasság számító eljárások, érzéketlenek a zajokra, viszont érzékenyen reagálnak a spektrumban megjelenő kiugró értékekre. Az amplitúdó-spektrum négyzetre emelésével nő a zaj, de a kiugró értékek szerepe is megnő.
Ezen módszer hátrányai közé tartozik az oktáv hiba probléma, mivel az ACF után akár a felharmonikusok akár az f_0 felénél levő frekvencia komponens szerepe is megnőhet, túlsúlyozódhat, ezzel hamis eredményt adva.

Kepsztrum analízis

Ezt az eljárást főleg beszédfelismerésnél használják. A kepsztrum a spektrum spektrumaként értelmezhető, ahol az x tengelyen az időegység (a kefrencia) helyezkedik el, a hozzájuk tartozó értékek a spektrum változását jelentik.
A Fourier analízis értelmében a hangot tekinthetjük végtelen sok szinusz hullám szuperpozíciójának. A kepsztrum beszédmodellben azzal a feltételezéssel élnek, hogy a tiszta beszédhangot az átviteli csatorna (száj, környezet) mint egy lineáris szűrő teszi zajossá (Gerőfi Balázs, 2005).

Az ilyen lineáris szűrők az időtartományban a konvolúció műveletével írhatóak le. Az időtartományban a jelet ezen két elválasztható összetevő konvolúciója szerint értelmezve:

$$X(t) = \int_0^t G(t) * H(t-\tau) d\tau \qquad (4)$$

Az időtartománybeli konvolúció szorzás a frekvencia tartományban:

$$X(\omega) = H(\omega)G(\omega) \qquad (5)$$

Majd az abszolút érték logaritmusát véve, aminek hatására a szorzat összegre bomlik:

$$\log|X(\omega)| = \log|H(\omega)| + \log|G(\omega)| \qquad (6)$$

A kapott jel alakját az alacsonyfrekvenciás, míg a finom részleteket a magas frekvenciás összetevők tartalmazzák. Ezek szétválasztására alkalmazzák az inverz Fourier transzformációt:

$$F^{-1}(\log|X(\omega)|) = F^{-1}(\log|H(\omega)|) + F^{-1}(\log|G(\omega)|) \qquad (7)$$

Ezen módszer érzékenyebb a zajra, viszont a spektrális szerkezet kiugrásai nem befolyásolják annyira, így az ACF-hez képest éppen ellentétes előnyökkel és hátrányokkal rendelkezik.

Harmonikus egyezés módszere

Az amplitúdó-spektrum minden egyes csúcsát f_0 frekvenciának tételezik fel, majd az egész számú többszörösnél megnézik a spektrumformát, majd az illeszkedés mértékétől függően pontoznak. Így jön ki a legvalószínűbb alapfrekvencia.

Valószínűségekre támaszkodó módszer

Egy újabb módszer a valószínűségeket figyelembe vevő számítás a spektrumminta illesztésekor [Dovel and Rodet, 1991]. Az alapötlet, hogy meg kell keresni azt az f_0 frekvenciát, ami legjobban indokolt ismerve az amplitúdó-spektrumot. Minden feltételezett f_0 frekvencia egészszámú többszöröséhez elhelyeznek egy gauss függvényt, középre igazítva. Majd az aktuális f_0 feltevéshez tartozó gaussok mentén megvizsgálják, hol helyezkednek el a tényleges spektrum komponensek, ez mutatja a valószínűségét minden egyes összetevőnek. Ezen módszernél a variálható paraméterek száma megnövekszik, mivel a görbék paraméterei állíthatóak így a körülményekhez alakítható a pontozás menete.

3.2.2.1.2. Spektrális távolsággal számoló algoritmusok

A valódi hangszerek által keltett hangok általában nem szabályos harmonikus hangok. Ez előző algoritmusok legfőbb hibája, hogy képtelenek megbirkózni a nem harmonikus hangokkal. A távolságon alapuló algoritmusok ezen problémát kompenzálják.

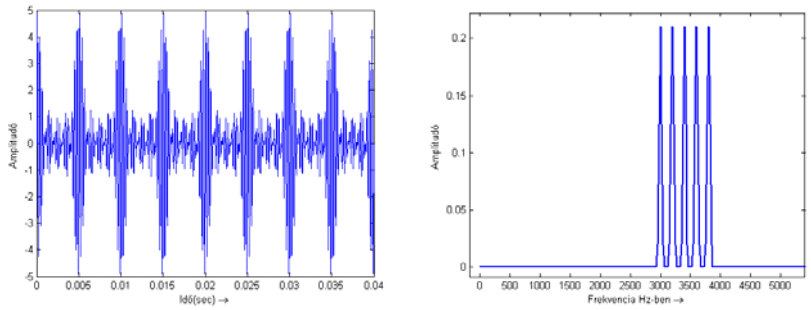
Ahogy a név is sugallja a módszer a spektrum komponensek távolságával számol. Így különösen előnyös nem ideális harmonikus hangoknál alkalmazni, mint például a zongora hangjainál (a zongora hangjának jelalakjában a felharmonikusok nem az egészszámú többszörösöknél vannak, hanem valamilyen konstans értékkel eltolódnak ezekhez képest).

Az amplitúdó spektrum is felfogható periodikusnak, mivel a harmonikusok egyenlő távolságra helyezkednek el egymástól. Itt is lehet autokorrelációt számolni. Ahogy az időtartománybeli ACF-nél az oktáv hibát az f_0 frekvencia felénél lévő összetevő okozta, a frekvencia tartománybeli jelek korrelációjánál inkább duplázódási hibák lépnek fel.

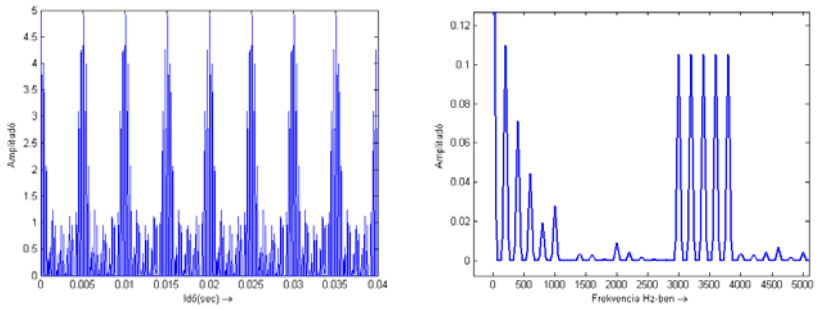
3.2.2.1.3. Kompromisszum

Ezek az algoritmusok, mindkét módszer előnyeit alkalmazzák. Az időtartománybeli burkoló periodicitását figyelik.

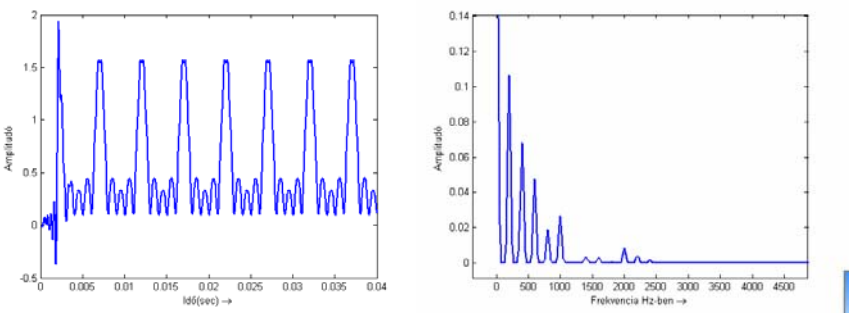
A több frekvencia komponenst tartalmazó jelek burkolója, időtartományban periodikus hullámzást mutat. Az egymást követő frekvencia komponensek különbsége, azaz az alapharmonikus jól megfigyelhető lüktetést eredményez a jelben. Harmonikus hangok esetében (zenei hangok), a komponensek távolsága összefügg az f_0-lal, és ez észrevehető a burkolóban.



10. ábra. 200 Hz-es jel felharmonikusai



11. ábra. A jel a negatív amplitúdók kinullázása után



12. ábra. A szűrt jel

Az ábrákon egy 200 Hz-es jel 15-től 19. felharmonikusai segítségével előállított jelalakot láthatunk. A spektrumban jól láthatóak az összetevő frekvencia komponensek, az időtartománybeli jelben pedig jól kivehető a lüktetés, ami az alapfrekvenciának köszönhető pedig ez a komponens nem is szerepel a jel spektrumában. Először „egyenirányítjuk" a jelet, tehát a hullámforma negatív részét levágjuk, ekkor a frekvencia komponensek között megjelenik az alapharmonikus is. Majd egy 1 KHz vágási frekvenciával rendelkező aluláteresztő szűrővel szűrjük, így megkapjuk a jel burkolóját. A spektrumban fennmaradó komponensek között a második csúcs az alapharmonikus. [Kompromi.m,lowpassfilt.mat]

A burkoló minőségét, felbontását a vágási frekvenciával állíthatjuk.

Ezzel a megoldással Fourier transzformáció és autokorreláció nélkül meg lehet határoznia a burkolót, aminek lüktetéséből már periódusidőt lehet számolni, ezzel megkapva az alapfrekvenciát.

3.2.2.2. Hangmagasság detektálása többszólamú környezetben

Itt a cél az, hogy az egyszerre szóló hangok összes f_0 frekvenciáját megállapítsuk. Jóval bonyolultabb a helyzet, mint előző esetben. Egyszerre több hangszer szólalhat meg, lehetnek zenei hangmagassággal rendelkező, és nem rendelkező hangok (szaxofon, dob páros). Problémát okozhat a helyi csúcsok időbeni találkozása, spektrumok ütközése, ez mind-mind nehezíti a feladatot. Nagy nehézséget okoz azon hangszerek hangjának elemzése, amelyek egymástól kis többszörösre levő f_0 frekvenciával szólalnak meg. Mivel rengeteg kereszthármonikus keletkezik. A spektrális mintakeresésre építő algoritmusok is megbuknak, mivel az egyszerre szóló hangszerek spektrum szerkezete módosul az eredetihez képest.
Általában az egyszólamú hangdetektálásra kifejlesztett algoritmusok nem működnek kielégítően többszólamú környezetben, ettől függetlenül a dallamkeresésben nagy segítséget nyújthatnak. A hagyományos egyszólamú algoritmusok, korreláció alapú fajtái, egy idő szelet alatt több csúcsot is megtalálnak, így az utófeldolgozó egység nagyon komoly feladatot végez, kiválogatja a megfelelő komponenseket.
Az ismertetésre kerülő többszólamú környezetben működő algoritmusok, automatikus zeneleíró rendszerekhez lettek kifejlesztve. A működési mechanizmusuk sokkal szerteágazóbb, mint egyszólamú társaiké. Különböző nézőpontokból, különböző modelleket felállítva fogják meg a problémát. Léteznek modellek, amelyek a hallás fiziológiáját veszik alapul, valamint olyanok, amelyek szem előtt tartják a hallgatás folyamata alatt előforduló speciális jellemzőket. Vannak hangszer modellekre épülő eljárások és olyanok, amik a zenei törvényszerűségekre figyelnek.

Az első próbálkozások 1970 –re nyúlnak vissza, James Moorer munkájában kemény megkötések mellett két hangszer felismerése vált lehetővé (Moorer, 1977). Mind a két

hangszernek változtatható hangmagasságúnak kellett lennie, a dinamikában nem volt megengedett a hangcsúszás (glissando), és a vibráció (vibrato). A két szólam nem keresztezhette egymást. És nem lehetett az egyszerre játszott hangok között kis többszörösű egymás alapfrekvenciájának. Pont ezért a gyakorlati haszna ennek a módszernek nagyon kicsi volt.

Robert Maher úgy oldotta meg a kétszólamú problémát (Maher, 1990), hogy időszeletekben frekvencia analízist hajtott végre majd kiválasztotta azt a két alpharmonikust, amihez előre meghatározva a felharmonikusokat, a legkisebb hibát találta a tényleges komponensekhez képest.

Ez idő tájt, japán kutatók heurisztikákat követő hangjegy csoportosítással kezdtek foglalkozni. Az ő rendszerük gitár, zongora és egy hagyományos japán hangszer a shamisen, trió leírására törekedett. Ők is a frekvencia tartományban keresték a hangokat jelölő csúcsokat, aztán különböző zenei heurisztikákat alkalmazva csoportosították ezeket, így kapva meg a hangjegyeket. Sajnos ez a rendszer sem működött megbízhatóan, ha egyszerre kettőnél több hang szólalt meg.

1993–ban a többszólamú zongora előadások feldolgozására, Michael Hawley publikált egy megoldást (Hawley, 1993), melyben egyidőben, több mint két hang is megszólalhatott. Az eljárás az egymás utáni időszeletek spektrumát hasonlította össze. A hangjegyek kezdetének keresése magas frekvenciatartományban történt. Ezzel a módszerrel bebizonyosodott, hogy az egy kitüntetett hangszeren játszott polifonikus darabok visszafejthetőek. Mivel a hangszer karakterisztikája jól ismert nagyobb szabadság adódott a többi korlátozó tényező állíthatóságában. Sajnos ez a módszer sem segítette elő az általánosító képességét a dallam felismerő rendszereknek.

A korábbi automata zeneleíró rendszerek, igencsak behatárolt körülmények között tudtak működni. Csak mostanában fejlesztettek ki olyan rendszereket, amik megbízhatóan működnek többszólamú körülmények között, és nem szűkítik le egy bizonyos hangszer modellre az analízist. Általános célú, robusztus, megbízható algoritmus még nem létezik, mivel az algoritmusok teljesítmény az egyszerre megszólaló hangok számával fordított arányban csökken, és a zaj hatására is jelentősen vissza esik az eredményesség.

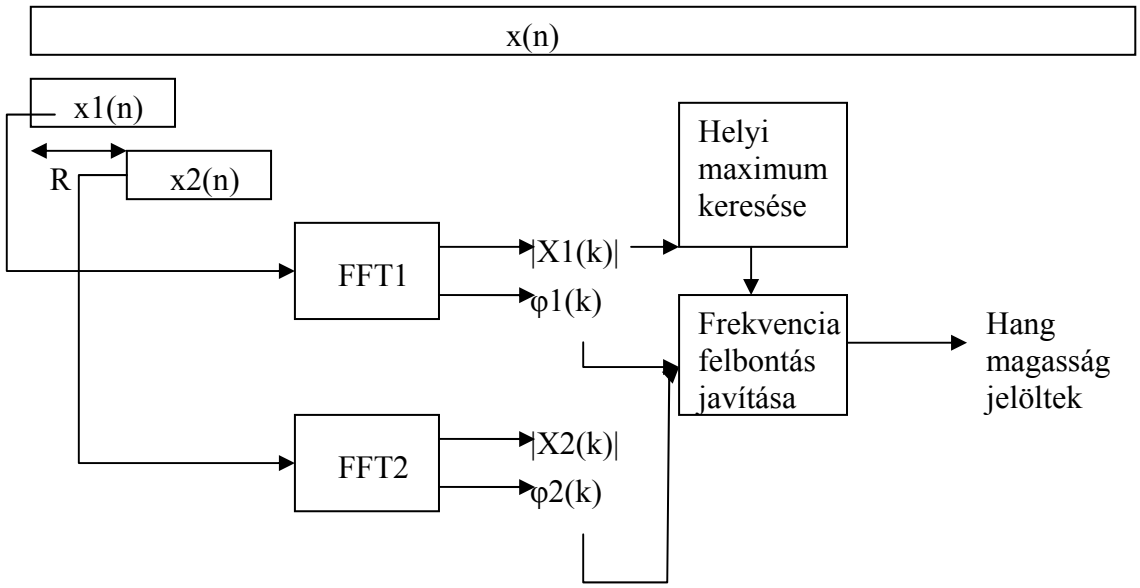
Kunio Kashio és csapata létrehozott egy rendszert, ami szinuszos nyomvonalakat keres majd ezek alapján hangjegy hipotéziseket állít fel, majd megpróbálja a hang forrását megállapítani, mégpedig a hangszínmodell segítségével.

Anssi Klapuri (Klapuri A., 2003) olyan módszerrel próbálkozott, hogy a bementi jelet különböző frekvencia sávokra osztotta.18 darab logaritmikusan felosztott sáv 50 Hz-től 6 KHz-ig, minden sáv 2/3 ad oktávot fed le. Az alapfrekvencia valószínűség vektorát minden sávra kiszámítják, majd az egyes valószínűségek kombinációjából előáll egy globális valószínűség, így majd képes lesz a rendszer kezelni a diszharmonikus hangokat. Aztán egy iteratív becslő és eltávolító eljárás következik. Először is a legvalószínűbb hangmagasságot kiválasztják ez lesz a domináns hang, majd ennek a felharmonikusait kivonják a spektrumból. Aztán a maradékon újra végrehajtják a domináns hang keresést és az eltávolítást. Ezen módszer mellékhatása a jó zaj-elnyomás, és az hogy meg lehet becsülni a konkurens hangok számát. A rendszer egyszerre 6 egyidőben megszólaló hanggal lett tesztelve, és minden esetben megtalált legalább egy hangot helyesen, mégpedig a legfeltűnőbbet. Az akkord felismerés terén pedig kimagasló sikereket ért el, lekörözve 10 képzet zenész átlagteljesítményét.
Klapuri kidolgozott egy halló-modell alapú rendszert is majd tovább fejlesztette Ryynanen-nel közösen (Ryynänen M. P. and Klapuri A., 2005a), ami többszólamú hangdetektálásra alkalmas. Ebben már bonyolult hangjegy modelleket, és zeneelméleti modelleket használnak. A folyamat végén optimális útvonalkeresést alkalmaznak a Token-passing algoritmus segítségével.
1970 óta sokat fejlődött a tudomány ezen ága, az algoritmusok sokkal robusztusabbak és rugalmasabbak lettek.

Végül egy olyan megoldást választottam, ami átmenetet képez a többszólamú és egyszólamú előadásokban hallható művek feldolgozása között. A szóló vokális daraboknál száz százalékos a detektálás hatásfoka. Az egyszerű hangszeres kísérettel ellátott zenéknél azonban a kíséret hangosabb részeinél a rendszer inkább nem ad becslést a hangmagasságra. A megvalósított algoritmus frekvencia tartományban működik. Lényege, hogy először hangmagasság lehetőségeket választ ki a rendszer, majd kiemel egyet és ennek a döntésnek a helyességét méri. A végső kiválasztás az utófeldolgozó részben történik meg, amikor az algoritmus az alapfrekvenciák többszöröseit összehasonlítja a többi valószínűsíthető lehetőséggel és kiválasztja a leginkább megfelelőt. Udo Zolzer - DAFX: Digital Audio Effects könyve nagy segítségemre volt az elméleti működés megértésében és az implementációban.

3.2.3. A megvalósított rendszer működése

Ez a megoldás szegmensenkénti Fourier transzformációval számol, és a fázis információkat használja fel. Minden szegmens után R mintával, egy N hosszúságú szegmens Fourier transzformáltját veszi, mégpedig úgy, hogy egy N pontos FFT (Fast Fourier Transform)-t hajt végre.(Udo Zolzer, 2003)



13. ábra. A rendszer blokkvázlata

Figyelembe véve az FFT számítását, a frekvencia felbontás $\Delta f=Fs/N$ az $Fs=1/Ts$. A bemeneti jelből x(n) N mintaszám hosszú blokkokat vágunk ki $x_1(n)=x(n_0+n)$, n=0,1…,N-1

Az FFT után keletkező amplitúdó spektrum k=0,..,N-1 komponensből áll. A helyi maximumhoz tartozót jelöljük k_0-lal. Az alapharmonikus becslése ekkor:

$$\tilde{f}_0 = k_0 \cdot \Delta f = k_0 \frac{f_s}{N} \qquad \textbf{(8)}$$

Az ehhez tartozó normalizált frekvencia:

$$\tilde{\Omega}_0 = 2\pi \cdot \tilde{f}_0 T_s = k_0 \frac{2\pi}{N} \qquad \textbf{(9)}$$

Hogy a frekvencia felbontást finomítsuk, a fázis információkat kell felhasználni. A harmonikus jelekre fennáll, xh(n)=cos(Ω_0n + φ0) = cos(Φ(n)) ebből az alapfrekvenciát deriválással számolhatjuk:

$$\Omega_0 = \frac{d\Phi(n)}{dn} \qquad \textbf{(10)}$$

A deriváltat úgy is számíthatjuk, ahogy az előző ábrán is látszik. Két egymás utáni blokk FFT-jét számoljuk ki, amelyek egymáshoz képest R mintával vannak elcsúsztatva.

$$\hat{\Omega}_0 = \frac{\Delta\Phi}{R} \qquad \textbf{(11)}$$

$\Delta\Phi$, az egyes blokkokhoz tartozó FFT, k_0-adik komponenséhez tartozó fázisok különbsége.

A második mintaszegmens, amin az FFT végrehajtódik:

$x_2(n)=x(n_0 + R + n)$, n=0,….,N-1

így a két fázis:

$\varphi_1 = \angle\{X_1(k_0)\}$

$\varphi_2 = \angle\{X_2(k_0)\}$

Mind a két fázist [-π ; π] tartományban értelmezzük. Majd kiszámoljuk a 'kibontott' φ_2 fázist [dafx handbook, 262,o] ezt jelöljük φ_{2u}. Feltéve azt, hogy a jel harmonikus komponenseket tartalmaz $\tilde{f}_0 = k_0 \cdot \Delta f$, az előre kiszámítható fázis, R mintával odébb ugorva:

$$\varphi_{2t} = \varphi_1 + \tilde{\Omega}_0 R = \varphi_1 + k_0 \frac{2\pi}{N} R \qquad \textbf{(12)}$$

A fázis hiba a kiterjesztett és az elméleti érték között:

$$\varphi_{2err} = princ\arg(\varphi_2 - \varphi_{2t}) \qquad \textbf{(13)}$$

A princarg függvény kiszámítja fő-fázist azaz [-π ; π] tartományban számol. Így a 'kibontott' fázis az elméleti értéktől, maximum π-vel különbözhet.

$$\varphi_{2u} = \varphi_{2t} + \varphi_{2err} \qquad \textbf{(14)}$$

A végső becslést adva az alapfrekvenciára:

$$\hat{f}_0 = \frac{1}{2\pi}\hat{\Omega}_0 \cdot f_s = \frac{1}{2\pi} \cdot \frac{\varphi_{2u} - \varphi_1}{R} \cdot f_s \qquad \textbf{(15)}$$

Azt feltételezve, hogy az első hangmagasságra adott becslés $\hat{f}_0$ maximum Δf/2-vel tér el az alapfrekvenciától, a fázis hiba maximális értéke:

$$\varphi_{2err,\max} = \frac{1}{2}\frac{2\pi}{N}R = \frac{R}{N}\pi \qquad \textbf{(16)}$$

3.2.4. Az algoritmus megvalósítása

Az előző ábrának megfelelően a rendszer szegmensekre bontja bementi jelet. A szegmensek NFFT darab mintát tartalmaznak. A pontos alapfrekvencia számításhoz egy R mintával elcsúsztatott blokkra is ki kell számolni a spektrumot.

3.2.4.1. A hangmagasság kereső algoritmus működése

Miután az algoritmus kijelölte az x1, x2 blokkokba tartozó mintákat, végrehajt egy NFFT pontos Fourier transzformációt. Ezek után kiválogatja azokat az összetevőket, amik számításba jöhetnek, tehát egy bizonyos megadott fmax frekvencia alattiakat.
A következő lépés a csonkolt X1-ben megkeresni a lokális maximumokat. Ezek indexeit egy idx nevű változó tárolja. Majd kiválasztani a globális maximumot. Ettől az értéktől függően, egy adott küszöbön belüliekkel foglalkozik tovább a rendszer. Ezek a potenciális jelöltek. Ezután az X1-be tartozó komponensek fázisaiból és a harmonikus jel feltételezésből kiindulva ad egy becslést az X2 fázisaira. Ha a becslés és a valódi értékek közti különbség nem túl nagy, egy bizonyos toleranción belül van akkor, harmonikusként megjelöli az aktuális komponenst. Ellenőrzi, hogy a megadott fmin frekvencia felett van e a komponens és kiszámolj a 'kibontott' fázist, majd ebből a becsült alapfrekvenciát.(melléklet\hangmagassag\spektrumszeletek.m)

3.2.4.2. A főprogram

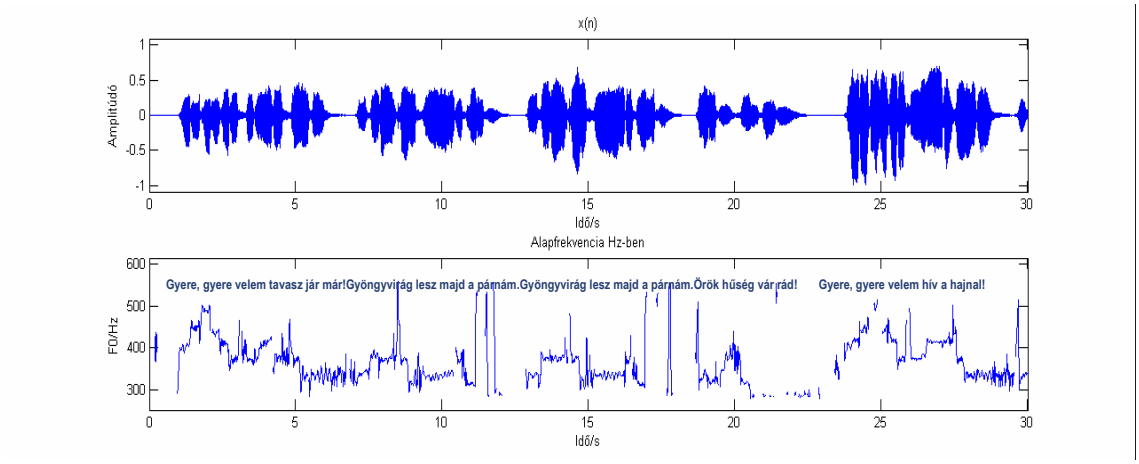
A beadandó paraméterek beállítása után, kialakítja a szegmenseket a bemeneti hangfájlból.
Aztán egy ciklus segítségével minden egyes szegmensre lefuttatja a hangmagasság kereső algoritmust. Végül a visszaadott értékekből a legmélyebb frekvenciájú harmonikus komponenst kiválasztja és ezt rendeli a szegmenshez.
Az utófeldolgozás is itt történik. A környező blokkok középértékét összehasonlítja az aktuális blokk hangmagasságával, ha túl nagy a különbség akkor a blokkot megjelöli zöngétlennek.(melléklet\hangmagassag\ main.m)
Ezek után egy szegmentáció történik, ami arra ügyel hogy ne legyenek túl rövid zöngés, és zöngétlen szakaszok. Pótolja – interpolálja - a hangmagasságokat a rövidebb üres szakaszokon.(melléklet\hangmagassag\korrekcio.m)

3.2.4.3. Eredmények

Elég sok hangfájllal kipróbáltam a rendszert. A leglátványosabb eredményeket, a szóló vokális alkotások esetén produkálja. Sajnos többszólamú esetben, domináns kíséret mellett elvész a dallam információ, mivel a közben megszólaló diszharmónikus hangokat zöngétlen kategóriába csoportosítja a rendszer (például dob, cintányér).

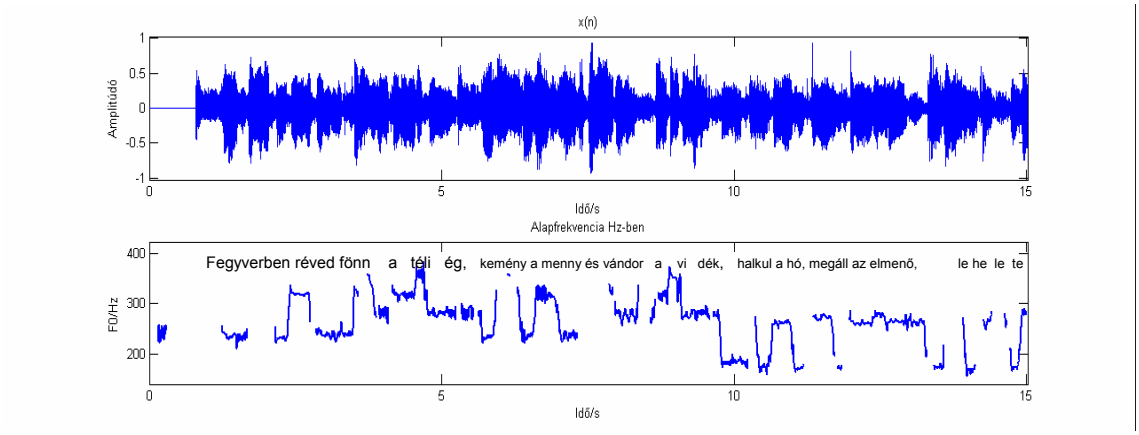
Tesztek

Az első teszt fájl, 30 másodperc Rúzsa Magdolna Ederlezi című számából (melléklet\hangmagassag\ederlezi.wav). Szembetűnő a 0-5 és 25-30 intervallumoknál a dallamismétlődés. Ez a mű rengeteg hajlítást tartalmaz, ezért van sokszor kiugró érték, és szakadás. A paraméter beállításoknál a legfontosabbak a spektrumszeletek alsó és felső határfrekvenciái, amiből az algoritmus válogathat. Itt a minimum frekvencia 250Hz, a maximum 1000Hz:



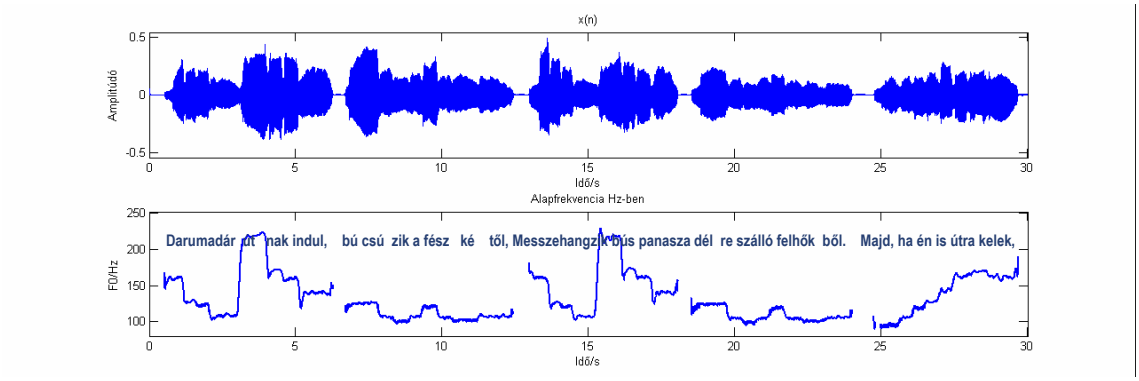
14. ábra. Hangfájl és dallam

Ágnes Vanilla - Óh, szív nyugodj (melléklet\zenék\agnesvanilla.wav) című száma. Erősen ugráló a dallam, ez a grafikonon is jól látszik. A minimum frekvencia 150Hz, a maximum 1000Hz:



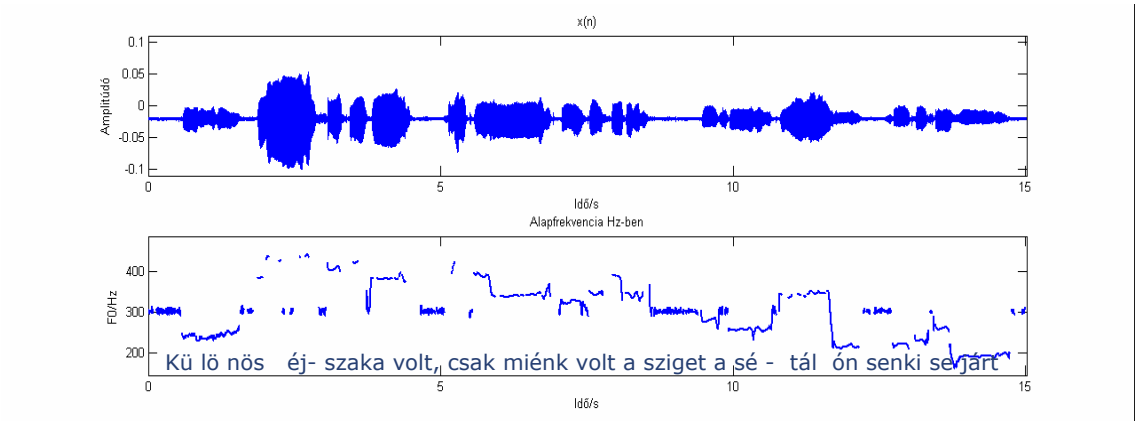
15. ábra. Hangfájl és dallam

A következő, mikrofonnal felvett dúdolás, a darumadár útnak indul című nóta (melléklet\zenék\levdarulala.wav) dallama. A minimum frekvencia 50Hz, a maximum 1000Hz:



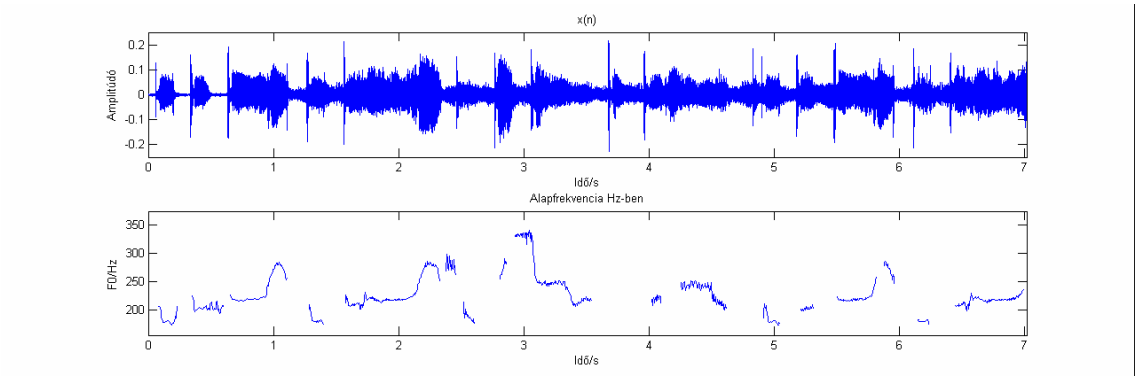
16. ábra. Hangfájl és dallam

Különös éjszaka volt című sláger (melléklet\zenék\különöséj.wav) első sora. A minimum frekvencia 150Hz, a maximum 1000Hz:



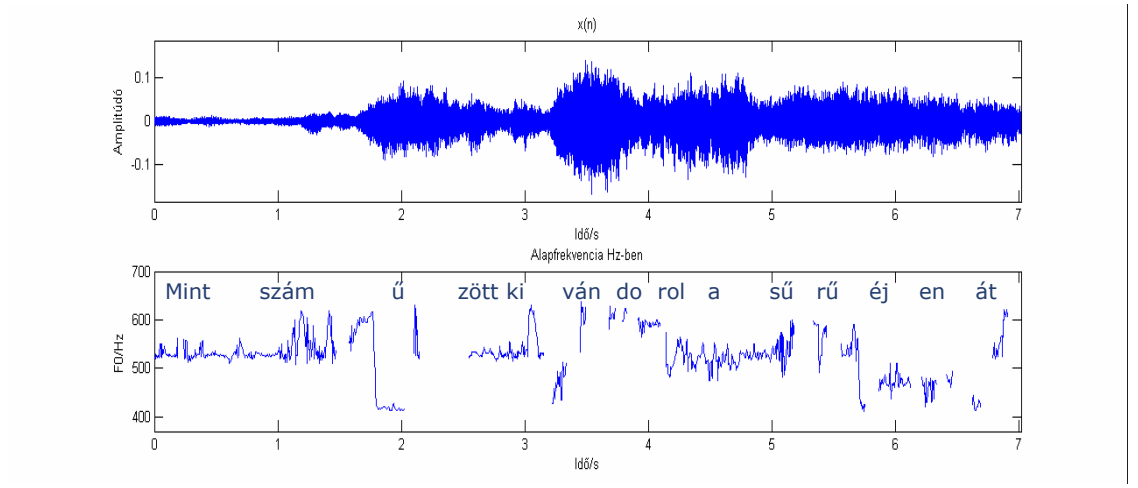
17. ábra. Hangfájl és dallam

Suzanne Vegh – Tom vacsorája című világsláger (melléklet\zenék\tomsdinner.wav) első 6 másodperce, csak dúdolás. A minimum frekvencia 50Hz, a maximum 1000Hz:



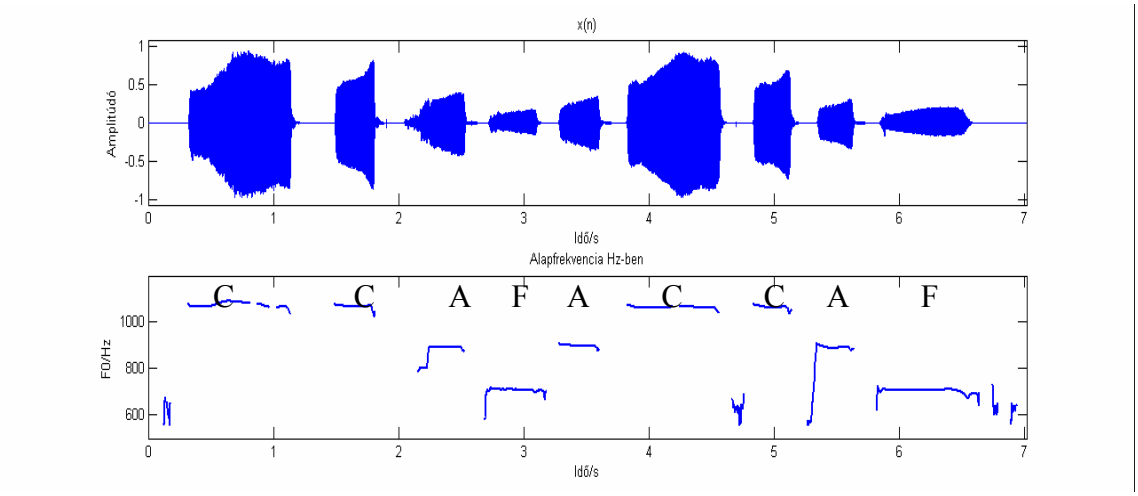
18. ábra, Hangfájl és dallam

BánkBán (melléklet\zenék\bankban.wav): Az eredményen jól látszik a dallam vonulata, a hangok pontos alapfrekvenciája nem kivehető, a minimum frekvencia 380Hz a maximum 1000Hz



19. ábra. Hangfájl és dallam

Furulya dallam (melléklet\zenék\furulyadallam.wav) a minimum frekvencia 500Hz a maximum 2000Hz:



20. ábra. Hangfájl és dallam

Konklúzió

Az algoritmus egy kezdetleges módszerrel dolgozik, szóló vokál és szóló hangszerek dallamát tudja leírni. Sajnos a háttér hangszerek sokszor teljesen összezavarják a működést. Így csak speciális dallamok esetén használható a szökőkút vezérlésére. A grafikus felhasználói felületen lehetőség van rá, hogy csak egy kijelölt szakasz dallam detektálását hajtsuk végre. Így a zenében az olyan részeken, ahol szóló vokális előadásra vagy húros, fúvós hangszer szólójára kerül sor, ott használható eredményesen. Továbbfejlesztési lehetőség, hogy a dallam ívek ismétlődését keressük meg és ezzel határozzuk meg a zenei sorokat és a sorszerkezetet.

4. Ritmikai információk kinyerése, tempó felismerés

4.1. Bevezetés

Az egész rendszer arra az alapgondolatra épül, hogy a ritmusfelismerés alacsonyszintű képesség, tehát nem szükséges zenei előképzettség hozzá. (Drake, C., Penel, A., and Bigand, E. (2000). A felismerés gyorsasága fokozható, tanulható, ez azzal igazolható, hogy képzett zenészek hamarabb fel tudják venni a tempót.

Ritmuskeresésnél az elsődleges információhordozó a zenei események kezdetének időpontja. Komplexebb esetekben a zenei események kiemelkedése, íve látványosan javítja a rendszer ritmus-fázis meghatározó képességét.

Miért is van szükség az ilyen és ehhez hasonló rendszerekre? Rengeteg technikai alkalmazása lehet egy tempóillesztő eljárásnak. Például az előadás analizálása során, a tempó, a tempóváltás segít a strukturális szerkezet és az érzelmi információk kinyerésében (Clarke, E., (1999).

Az ütemérzékelésnek nagyon bő irodalma van. Sokan foglalkoznak zenei paraméterek vizsgálatával és azok összefüggéseivel. A legnépszerűbb témák közé sorolható a ritmika feszessége, a dinamikai törvények betartása és a hangjegyek hangsúlyának vizsgálata. (Steedman, M., (1977); Desain, P., (1992).

Tempófelismerést jelenleg is használ a szórakoztató ipar, egyik ilyen alkalmazás a fénytechnikai eszközök automatikus szinkronizációja a zenéhez. Elektromos és digitális hangszerek összehangolása, irányítása. Felvevő eszközök kontrollja, számítógépes animációk, videóklippek zenei kísérete, hangelemek és grafikai elemek szinkronizálása.

A hangrögzítés módszerei közül kettőt emelnék ki, ami digitális szempontból fontos. A zenét és az egyéb hangokat valamilyen formában rögzíteni kell és erre adott a hagyományos analóg rögzítést alapul vevő mintavételezett, digitalizált jeltárolás (audio) vagy a szintetizált hangokból felépülő szimbolikus jelölésrendszert alkalmazó formátum, a MIDI.

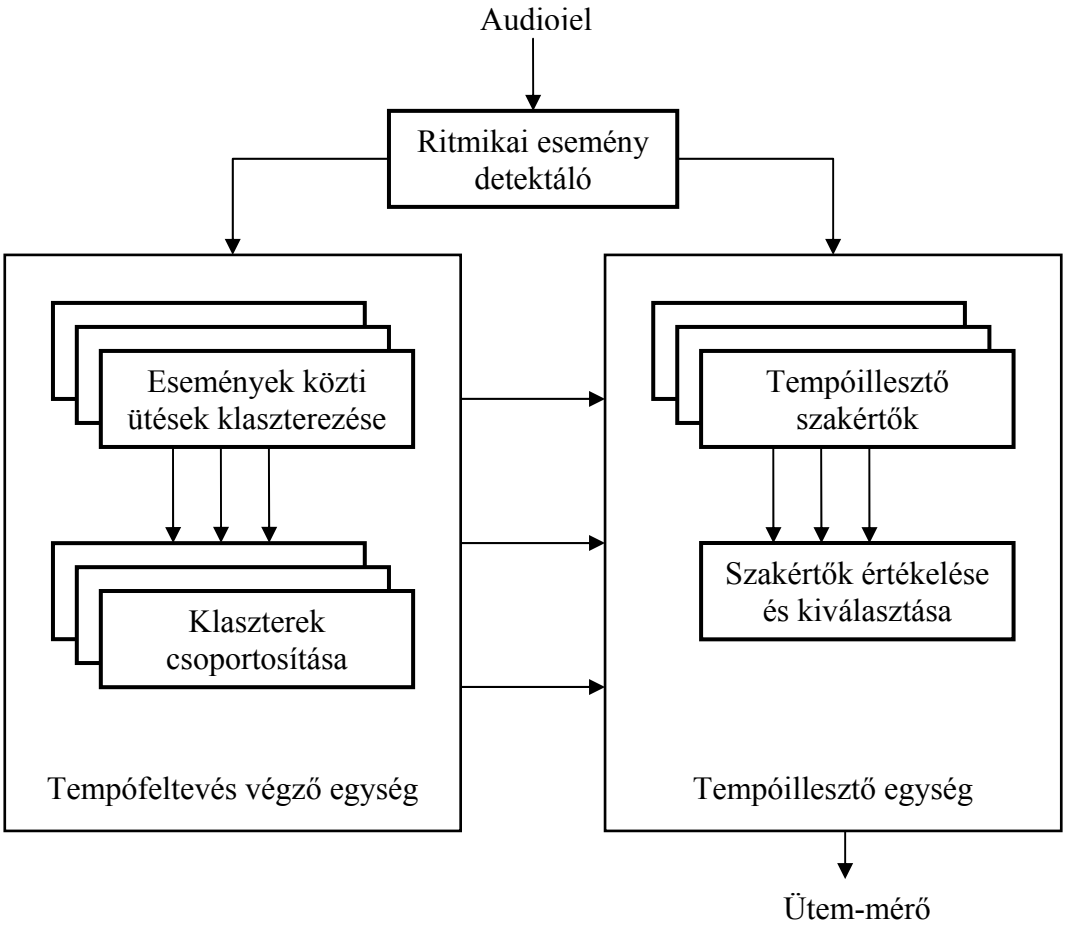
A zenei ritmus szabályos lüktetés, azaz időtől függő. Alapegysége lehet bármely zenei hang, sőt zaj, zörej, de keveredhet elkülönülten zenei dallammal. Tulajdonképpen

egyetlen igazi követelmény van: legyen többé-kevésbé szabályos. („Többé-kevésbé” alatt azt értem, hogy vannak szabadon, azaz rubato előadható zenedarabok is. Ennek a ritmikai kilengésnek, azaz gyorsításnak-lassításnak azonban általában komoly előadásbeli korlátai vannak.) Mivel a ritmus szabályos, így az előre kiszámítható, modellezhető. (Gitáriskola internetes forrás)

4.2. A rendszer rövid bemutatása

A tempó becslése audio adatok alapján úgy történik, hogy a ritmikai események kezdetét kell megtalálni. Erre az a módszer, hogy az amplitúdó burkolóra illesztett érintők meredekségének maximumait kell meghatározni.
A szimbolikus adatok arra jók, hogy a további feldolgozásban segítsenek. Például egy midi kottából néhány változós függvény segítségével könnyedén lehet becslést adni az aktuális hangjegy hangsúlyára. A megvalósított rendszer nem midi fájlokat dolgoz fel, ezért az adatokat az audio jelből kell kinyernie, és mint látni fogjuk ezek közül a legfontosabbak a hang hossza, intenzitása és felfutási meredeksége.
A hangok ritmikai eseményekbe csoportosulnak, s ezen események íveit, hangsúlyát kell megbecsülni. Ezt követően az egymás utáni események közti időket kell megmérni, majd klasztereket kialakítani a nagyjából egyenlő hosszúságú időintervallumoknak. Azután a klasztereket rangsorolni kell attól függően, hogy hány tagjuk van. Ez a rangsor fogja megadni a tempófeltevések (tempóhipotézisek) sorrendjét is.
Majd az alaptempóra vonatkozó rangsor, azaz a feltevések, egy újabb rendszer (beat tracking) bemeneteit képzik, ami egy sok szakértős egységet használ, hogy tesztelje a különböző fázis feltevéseket és megtalálja a legjobban illeszkedő szakértőt (21. ábra). A szakértőt egy egyszerű adattárolóként kell elképzelni, ami minden ritmikai esemény után módosítja belső adatait és pontozza önmagát aszerint, hogy addig milyen teljesítményt nyújtott. A szakértők azt a feltevést használják, hogy a legkiemelkedőbb ívű, leghangsúlyosabb hang általában a legmeghatározóbb a ritmikában. A hangsúly kiszámítása a hangok kitartásának ideje, intenzitása, hangmagassága alapján történik.

Ezek után a tesztelési fázis következik, amiből azt lehet megállapítani, hogy a modern zenében a ritmikai események amplitúdója, a klasszikus, expresszív zenében a hangok kitartásának időtartama a legmeghatározóbb paraméter.



21. ábra. A tempókövetkeztető rendszer

4.3. Irodalmi példák felsorolása, eddigi módszerek ismertetése

4.3.1. Szigorú metrikus előadásmód

Expresszív, érzelmi előadásmód nélkül a mérő-ütések közti intervallum a legrövidebb ritmikai események közti idő sokszorosa. Az összes többi időtartam egész számú többszöröse ennek a legrövidebb időköznek. Expresszív esetben ez jelentősen módosulhat, így erre a törvényszerűségre nem számíthatunk.

Steedman 1977-ben olyan rendszert publikált, ami hanghossz és dallamismétlődés alapján következtetett a ritmus szerkezetére. Azt vette észre, hogy az ütemben a szinkópa előtti ritmika egyszerűen megállapítható, minél kevesebb szinkópa található egy bizonyos szakaszon, annál nagyobb súlyt lehet adni az ezen szakasz alapján kalkuláló szakértők véleményének (Steedman, M., 1977).

Longuet-Higgins és Lee 1982-ben szintén ritmus meghatározással foglalkoztak, az elvük az volt, hogy két egymást követő hang alapján adtak becslést a következő hang megszólalásának idejére. A bináris ütemmutatós zenéket jól kezelte, de például a ¾-eseket nem. Egy továbbfejlesztett változat a szinkópa ritmusára is kitért (Longuet-Higgins, H. and Lee, C., 1982).

Lerdhal és Jackendoff 1983-ban egy olyan módszerrel álltak elő, mely periodikusságot keresett a jelalakban, valamint szabálybázisokat alkotott:

A hangok kezdete általában egybeesik a mérő-ütéssel, a mérő többségében hosszú hangokkal esik egybe, a bináris ütem csoportoknál elől van általában a mérő (Lerdahl, F. and Jackendoff, R., 1983).

Powel és Essens 1985-ben egy modell alapú rendszert fejlesztettek ki, ami ideiglenes mintákat keres és belső órát illeszt a megjelenő csúcsokra. 1991-ben Lee ezt a modellt fejlesztette tovább kibővítve azzal, hogy vannak erős és gyenge mérő ütések és ezek korrelálnak a jelalak kiemelkedéseivel. Ez a modell a gyengén megszólaló hangokkal és a szinkópákkal nem foglalkozott (Povel, D. and Essens, P., 1985).

4.3.2. Szimbolikus adatok

A legtöbb munka a tempófelismerés terén MIDI fájlokat használ bemenetként. A bemenetet úgy tekinthetjük, mint egy eseménysorozat leírást, amely minden egyes eseményről tartalmazza a pontos kezdeti időt, a tartamot, a hangmagasságot, az amplitúdót, és a szintetizálandó hang minőségét. Ezek a paraméterek nem hordoznak alapritmikára épülő információkat, sőt nincs is olyan paraméter, ami a tempóra, az ütemre vagy a mérőre vonatkozó adatokkal szolgálna. Viszont ezen paraméterek alapján számítható a zenei hang hangsúlya, ami erősen korrelál a mérő-ütésekkel.

Rosenthall 1992-ben az emberi ritmus felismerést vette alapul, a ritmikai szerkezetre többszörös hipotézist adott, majd pontozta ezeket annak függvényében, hogy mi a valószínűsége annak, hogy egy emberi hallgató ezt a feltevést választaná. Ez a rendszer rugalmasan követte a tempóváltozásokat (Rosenthal, D., 1992).

Desian 1993-ban többféle modellt összehasonlított, ő ért el előrelépéseket a real-time (valós idejű) feldolgozás terén (Desain, P., 1993).

Large és Kolen 1994-ben nemlineáris oszcillátort alkalmaztak, hogy rendszeres lüktetést keressenek a jelben. Itt nem volt tempófeltevés, az alaptempó és a fázis is szükséges alapinformáció volt, az expresszív modulációból eredő változás követése már könnyedén ment (Large, E. and Kolen, J., 1994).

Rowe 1992-ben olyan algoritmust fejlesztetett, ami rengeteg hipotézist használt, diszkrét csoportokba osztotta a lehetséges tempókat a metronóm skálázását véve alapul. Mégpedig a mérő-ütések közti idő intervallumokra 123 lépcsőt használt 10ms-es lépésekkel 280ms-től 1500ms-ig. (Rowe, R., 1992).

Egy újabb ötlet volt a keretrendszer bevezetés (Cemgil, A., et al, 2001). Dinamikus rendszerrel modellezte a mérőt, gyakoriság és fázis paramétereket használt. Egy Kalman szűrős rendszer becsülte meg a változókat. A rendszer hasonlított egy tökéletes metronómhoz, amihez Gauss zajt kevertek. A paraméterek becslését egy adathalmaz megtanításával állították elő. Ez a rendszer nagy sikereket ért el.

4.3.3. Audio adat

Az első leírások az ütős hangszerek detektálásával foglalkoztak (Schloss,W., (1985). A módszer úgy működik, hogy csúcsokat keres az amplitúdó burkoló meredekségében, ahol az amplitúdót úgy definiálták, hogy minden periódusban a felül-áteresztő szűrővel szűrt jel amplitúdó maximuma és ahol a periódus hossza a legkisebb várható frekvencia komponens reciproka. Sajnos a bemenő paramétereket kézzel kel hangolni.

A legfőbb eredményeket, Goto és Muraoka érték el az audio jelek feldolgozása során. (Goto, M. and Muraoka, Y., 1995, 1997a,b, 1998, 1999). Ők olyan rendszert dolgoztak ki, amely népszerű zenék feldolgozásával foglalkozott. Kettéválasztották a detektálási módszereket aszerint, hogy a feldolgozandó zene tartalmaz-e dobot vagy sem. Az első rendszerük (BTS) a ritmikai események kezdőpontjában vizsgálta a frekvencia összetevőket és ezt hasonlította egy előre eltárolt pergődob és basszusdob mintához. A rendszer csak speciális zenékre működött de azokban majdnem 100 százalékos sikert ért el.

A második rendszerük magas szintű feldolgozást használva, vezérelte az alacsonyszinten működő ütemkereső algoritmust. A frekvencia tartományban figyelték az akkordváltásokat, amelyek nagy valószínűséggel ritmikailag fontos helyeken álltak. Ez is csak a zenék egy speciális fajtájára működött, ráadásul csak 4/4-es ütembeosztásúakra, amelyeknél a tempó 61 és 120 negyedhang között van percenként. Mindkét rendszer valósidejű feldolgozást hajtott végre és párhuzamos szakértőket használt különböző paraméter beállításokkal. A rendszer multiprocesszoros környezetet igényelt.

Scheirer 1998-ban publikált egy olyan módszert, amiben hangolható rezonátorokat használt. Először a jelet 6 frekvenciasávra osztotta, majd mindegyiket átengedte 150 darabos szűrőbankon, amiben fésűs szűrők voltak megvalósítva. Minden szűrő egy diszkrét skálán megjelölt tempóértéket reprezentált. A szűrők kimeneti eredményeit összeadta az összes frekvenciasávra, majd a maximális eredményt elérőt választotta. Különböző stílusú zenékből összeválogatva 60-ból 41-nél helyes eredmény kapott. A probléma mindössze annyi volt, hogy a változó tempójú darabokban a rendszernek váltogatnia kellett a filterek között. Így a fokozatos tempóváltást nem tudta követni a rendszer (Scheirer, E., 1998).

Az algoritmusokat a Simon Dixon által írt automata tempó és ütem-mérő detektáló tanulmány alapján valósítottam meg és fejlesztettem tovább (Simon Dixon, 2001). Valamint nagy segítségemre voltak még a következő írások:
(Masataka Goto and Yoichi Muraoka, (1994). ; Masataka Goto (2001); Goto, M. and Muraoka, Y., (1997a).; Jarno Seppänen, (2001).; Nick Collins (2004). ; Nick Collins(a); Dafx Handbook; Stephen W. Hainsworth (2003); Masataka Goto, Yoichi Muraoka (1999b))

4.4. A bementi adatok formátuma

Ha a szakirodalmat tekintjük, kétféle bementi adatlehetőség közül lehet választani: a digitális audio formátum és a szimbolikus adatformátum. A szimbolikus adatok hordozására a MIDI fájlformátum szolgál. Az általam megvalósított rendszernek audio formátumot kellett használnia. Ez jelen esetben nem más, mint a tömörítetlen lineáris pulzus modulációs kód (PCM). Ez található az audio CD-ken és a microsoft wav fájlformátum is ilyen rendszerű. Az implementáció Matlab-ban történt, ezért érdemes volt wav fájlokkal dolgoznom, mivel a Matlab támogatja ezt a fájlformátumot, lehetőséget ad egyénileg kiválasztani a mintavételi frekvenciát, a szóhosszt és a csatornák számát.

4.4.1. Tempókövetkeztetés

Az általam implementált rendszer két fő egységből áll, az első a tempókövetkeztetés, amit ebben a részben fogok kifejteni, a második a tempó fázis-illesztés, amit a következő részben részletezek.
A tempókövetkeztetés alapja, hogy páronként a ritmikai események közti időintervallumokat feljegyzik, majd klaszterezik őket. Minden klaszter egy hipotézis a tempóra vonatkozólag. Így a tempóra a két ütés közti idő a jellemző, ami fordított arányban van a tényleges tempómértékegységgel (ütések száma percenként). Ezek után a klaszterek rangsorolása történik meg aszerint, hogy hány elemük van. Ez az egység a fázisról semmit nem mond, arról a következő egység gondoskodik. A tempókövetkeztető egység a ritmikai eseményeken hajt végre műveleteket, mely

súlyozott időpillanatok soraként fogható fel. Ritmikai esemény lehet egy szóló hang kezdete, de akár több hang egyszerre megszólalásából kialakuló komplex jelalak is. Az események leírásakor két fontos információt kell eltárolni, az események időbeni helyét és a hozzá tartozó hangsúlyt, jelalak ívét. A hangsúlyt vagy ívet a megjelenő hangok paraméteriből számíthatjuk: hangmagasság, hangkitartásának ideje, és a jelenlevő hangok száma.

4.4.1.1. Az algoritmus működése

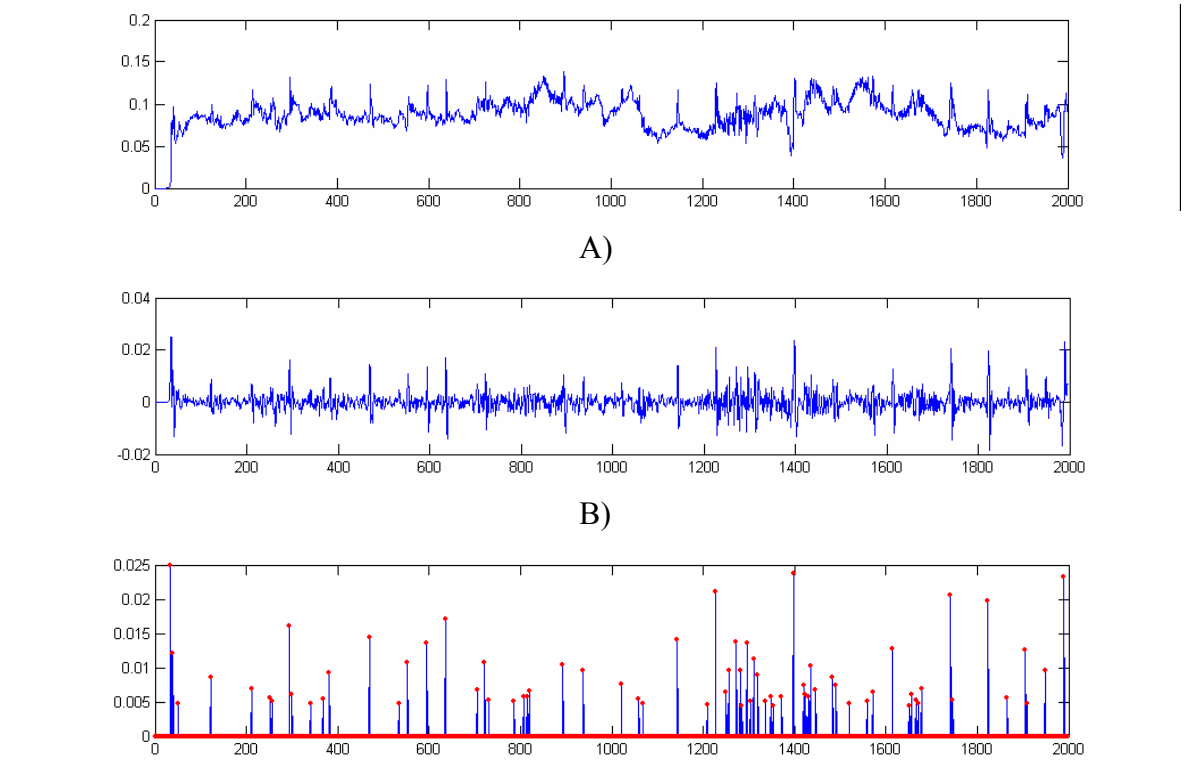
Ritmikai eseményeket a zenei komponensek kezdetének tekinthetjük. Amikor ezen események nagyon közel vannak egymáshoz, összevonhatjuk őket. A fül sem képes elkülöníteni bizonyos időn belül megszólaló hangokat. Audioadatnál nagyon nehéz kiszűrni ezen események kezdeti idejét. Szimbolikus adatok esetén rengeteg hasznos információ kódolva van az adatsorban, például a hangok kezdete, azonban ezek a hullámforma tényleges kezdetét jelölik, nem pedig a hangsúlyos kimagasló részt, ami persze függ az aktuális hangszintetizátor választásától. Minden egyes hangszernek más a megszólalási ideje és ez természetesen nincs kódolva a MIDI adatok között így a leghangsúlyosabb rész időpontkinyerése ebből az adatformátumból szinte lehetetlen. Így midi adatok esetén a legpontosabb ritmikai információt az ütős hangszerek szolgáltatják, mivel ezek rövid felfutási idővel rendelkeznek.

Az első feladat, csoportosítani az együtt megszólaló hangokat egy ritmikai eseménnyé és ennek a hangsúlyát, ívét kiszámolni. Tanulmányok az aszinkronitásról: (Sundberg, J., 1991; Goebl, W., 2001). Két szóló hang esetén 30-40 ms az a határ, amikor még a két hang egynek hallatszik. Több hang „egyszerre” megszólalása esetén ez a határ kitolódhat 70 ms-ra is. A második fő feladat súlyozni ezen ritmikai eseményeket. A zeneelméletben rengeteg faktor található egy hang hangsúlyának megállapítására, én főleg a felfutási meredekséget, az intenzitást és az időtartamot fogom felhasználni. A Dixon-féle tanulmány a hangsúlyt MIDI adatokból állapítja meg, így felhasználja még a hangmagasságot és a dinamikát is.

Audio adatok esetén jelentős feldolgozásra van szükség, a zenei tartalom kinyeréséhez. Manapság nincs tökéletes algoritmus, ami például az összes hang kezdeti idejét, az összes ritmikai eseményt kinyerné. Az ütem-mérő illesztéshez elegendő információt lehet kinyerni majdnem minden esetben. Az ütem-mérő meghatározása úgyis a

hangsúlyos hangokon alapszik, az audio adatformátum ezt megkönnyíti, mivel a kis ritmikai események detektálására általában nincs mód. Az eseménykereső algoritmus a Schloss, 1985-ben publikált módszerén alapszik. Először a jelet felül-áteresztő szűrővel kell szűrni, majd átlagolni, hogy burkoló jellege legyen. Egy ablakba tartozó átlagos abszolút értékkel számolunk, mely 20 ms hosszú és 50 százalékos átfedés van. Így a burkoló felbontása 10 ms-os. Ezután 4 pontos lineáris regresszióval meg kell határozni a burkoló meredekségét, majd egy csúcskereső algoritmus kijelöli a lokális maximumokat, a meredekség függvényében. Majd a csúcsokat meg kell ritkítani, kiválogatva a számunkra fontosakat különböző heurisztikák szerint. Ebben az esetben a szabály az, hogy ha 50 ms-en belül van magasabb csúcs, akkor a többit el kell hagyni vagy ha egy bizonyos küszöb alatt van ez a csúcs akkor is el kell hagyni. Ezt a küszöböt empirikus úton határoztam meg és a burkoló bizonyos számú mintájának átlagára vonatkoztattam. A függvényeket úgy írtam meg, hogy az összes paraméter állítható az algoritmus lefuttatása során.

Az első ábra a burkológörbe, a fent említet módszer használatával (A), a második a regresszió után a meredekségértékek (B), a harmadik a kiválogatott eseményeket jelöli (C).



22. ábra. Az algoritmus eredményei a 08-Air.wav fájl első 20 másodpercére

Amennyiben a ritmikai események meghatározása nem elég pontos, akkor magasabb szintű feldolgozásra van szükség. Az alacsony szinten kiszámolt ritmikai események körüli frekvenciatartománybeli analízis segíthet ebben, ezzel a módszerrel lehet pontosan kinyerni az egyes események súlyát és idejét.

Következő lépés a megtalált események közti időintervallumok csoportosítása. A ritmikai események idejének meghatározása megadja a ritmikai szerkezetet. A klaszterező algoritmus rangsort állít fel a tempó feltevésekre. Ezt a rangsort a következő részben bemutatott ütem-mérő illesztőegység fogja felhasználni.

Itt fontos definiálni az IOI-t, (inter onset interval) két ritmikai esemény között eltelt időt.

A tempó információt az IOI-ből lehet kikövetkeztetni. Ez 50 ms és 2 s között kell, hogy legyen. A hasonló hosszúságúak egy klaszterbe kerülnek, a kirívó intervallumoknak új klasztert hoz létre. A klaszter fontos jellemzője az intervallum, ami a benne lévő IOI-k átlaga. Akkor tartozik egy IOI egy klaszterbe, ha a benne levők átlaga és az adott IOI különbsége nem haladja meg a 25 ms-ot. (IOI konfidencia = klaszter konfidencia) A folyamatos klaszterezés azzal jár, hogy a jellemző intervallumok valamilyen irányba elcsúszhatnak.

Ha a klaszterezés kész, akkor kialakul a rangsor a bent lévő IOI-k számának függvényében. Majd elkezdődnek a különböző klaszter-szám redukáló algoritmusok, Két klaszter rokon, ha az egyik intervalluma a másik egészszámú többszöröse (d-szerese), itt is 25 ms a konfidencia (klaszter konfidencia). A rangsorolás értékkorrekciója úgy történik, hogy a rokon pontjai súlyozva hozzáadódnak a másikéhoz. Ezt egy rokon faktor adja meg f(d).

f(d)=6-d,	ha 1<=d<=4
f(d)=1,	ha 5<=d<=8
f(d)=0,	a többi esetben

23. ábra, Rokon algoritmus súlyozó függvénye

A legmagasabbnak rangsorolt klaszter lesz valószínűleg az alaptempó.

Az algoritmus pszeudokódos leírása

C_1,C_2…: Klaszterek:
C_i.méret: ahány darab IOI van benne
C_i.intervallum=sum(IOI)/C_i.méret
f(d) : rokonsági faktor
E_1, E_2…: eseményekhez tartozó idők
$IOI_{ij}=E_i-E_j$

```
CIKLUS minden Ei-re
        CIKLUS minden Ej-re
                IOI=Ei-Ej
                KERES k ahol abs(Ck.intervallum-IOI) < IOI konfidencia és minimum
                HA k létezik
                        IOI-t tegyük bele Ck-ba
                EGYÉBKÉNT
                        új klaszter nyitása
                VÉGE
        CIKLUSVÉGE
CIKLUSVÉGE

CIKLUS minden Ci-re
        CIKLUS minden Cj-re (i != j)
                HA abs(Ci.intervallum- Cj.intervallum) < klaszter konfidencia AKKOR
                        összevonni Ci és Cj
                        törölni   Cj
                HAVÉGE
        CIKLUSVÉGE
CIKLUSVÉGE

CIKLUS minden Ci-re
        CIKLUS minden Cj-re (i != j)
                HA (Minden d-re) abs(Ci.intervallum- d*Cj.intervallum) < klaszter…
                                                                …konfidencia AKKOR
                        Ci .pontja= Ci .pontja + f(d) * és Cj .pontja
                HAVÉGE
        CIKLUSVÉGE
CIKLUSVÉGE
```

4.4.2. A tempóillesztő egység

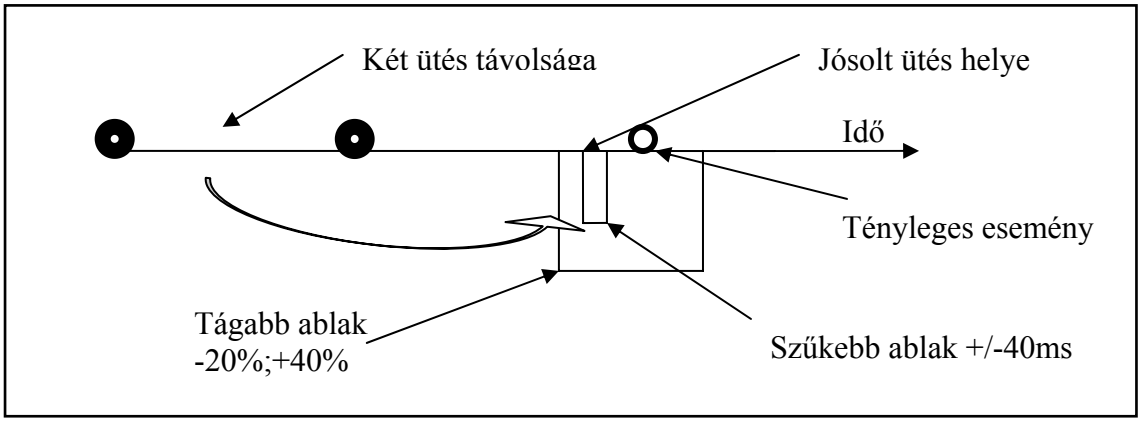
Ebben a részben az ütem-mérő illesztő algoritmust fogom bemutatni, ez a második egység, ami a tempó feltevés eredményét használja saját számításaiban. Mivel a tempó hipotézisek semmit nem mondanak az ütések fázisáról, ezért többszörös szakértőrendszert kell alkalmazni, hogy megtudjuk a pontos idejét egy-egy ütem-mérőnek. Kell egy olyan függvény, ami pontozza a kialakuló eredményeket és ami alapján kiválaszthatjuk a legjobban illeszkedő hipotézist. Minden hipotézishez hozzá van rendelve egy szakértő, ami képes a hipotézis (azaz az a becsült alaptempó) kis mértékű módosítására azért, hogy a folyamatos tempóváltozásokat követni tudja. Állandóan figyeli a ritmikai eseményeket, azokkal szinkronizál, módosíthatja saját feltevését a tempóról és a fázisról egyaránt. A rendszer új szakértőt hozhat létre menetközben, ha úgy dönt, hogy elágazás szükséges és törölheti a duplikálódott szakértőket. Minden szakértő jelenállapottal és előélettel jellemezhető. A jelenállapot a pillanatnyi feltevés az ütemfrekvenciára és fázisra vonatkozólag. Az előélet pedig egy ütemmutató szekvencia, ami időbeli markerekkel van ellátva, tehát egyfajta napló. A szakértő képes értékelni a teljesítményét úgy, hogy összeveti a tényleges eseményeket a jósoltakkal. Ezek egy egységes pontrendszert alkotva a végén minden szakértőre kiadnak egy értéket, majd ki kell választani a legmagasabb pontszámút. A rendszer képes követni a finom tempóváltásokat és a szakadásoknál pótolja a hiányzó ütéseket.

4.4.2.1. Az algoritmus

Minden tempóhipotézisre egy-egy csoport figyel, több szakértő van különböző kezdési időkkel, mivel nem tudjuk, melyik a helyes fázis. Az inicializáló periódus alatt van legalább egy szakértő, aki helyesen tippel, mivel ez idő alatt minden egyes eseményhez hozzárendelünk egy szakértőt egy tempóhipotézissel. Így az inicializálás alatt összesen (tempóhipotézis szorozva az események számával) darab szakértő keletkezik. Az inicializáló szakasz általában 5 másodperc, némely esetben ezt le kellett csökkenteni, mert a kezdeti nagy szakértő szám miatt nagyon sok számítást végez a rendszer. Az expresszív előadások esetében, az inicializáló periódus alatt még mindig előfordulhat, hogy nincs olyan szakértő, ami helyesen kezdene, viszont a korrekció alatt úgyis

módosul a fázisfeltevés, tehát idővel valószínűleg „rááll” a rendszer a helyes megoldásra.

A tempóillesztő algoritmus főblokkjában minden eseményt sorra veszünk, minden szakértőnek megadjuk a lehetőséget, hogy az aktuális eseményt ütem-mérőként kezelje. Minden szakértő az aktuális fázisállapota és tempóhipotézise alapján megjósolja a következő ütést és ezt egy kettős konfidencia ablakkal veszi körül. Az ablakon belül lévőket a szakértő helyes tippként rögzíti. A szűkebb ablak ±40 ms szórást engedélyez, a tágabb ablak pedig az aktuális intervallumnak a 20%-át a jósolt időpont előtt, és 40%-át utána.



24. ábra. A predikciós ablakok

A szűkebb ablak az emberi előadás miatti ritmikai pontatlanságokat reprezentálja, a tágabb ablakra azért van szükség, hogy a zene dinamikai változásait követhesse a rendszer. Az aszimmetria abból fakad, hogy a tempólassítás sokkal gyakoribb és jelentősebb mértékű, mint a gyorsítás (Repp, B., 1994).

Az algoritmus pszeudokódja

A: szakértők halmaza

A_j.intervallum: a j-edik szakértő aktuális tempóhipotézise

Tol_utó= A_j.intervallum*0.4

Tol_elő= A_j.intervallum*-0.2

Tol_belső=40ms=4egység

Hibakorrekció=10%

Relatívhiba=Hiba/(Tágabb_ablak/2)

Inicializáció

CIKLUS minden tempó feltevésre T_i

CIKLUS minden eseményre E_j Ahol E_j < inicializációs periódus

Új szakértő létrehozása A_k

A_k.intervallum:=T_i

A_k.jóslat:=E_j+T_i

A_k.napló:=E_j

A_k.pontozás=E_j.hangsúly

CIKLUSVÉGE

CIKLUSVÉGE

Főprogram

CIKLUS minden eseményre E_i

CIKLUS minden szakértőre A_j

AMÍG A_j.jóslat+Tol_utó<E_i

A_j.jóslat := A_j.jóslat + A_j.intervallum

AMÍGVÉGE

HA A_j.jóslat+Tol_elő <= Ei <= A_j.jóslat+Tol_utó AKKOR

HA abs(A_j.jóslat- E_i) > Tol_belső

Új szakértő létrehozása $A_{k:=}$ A_j

HAVÉGE

Hiba=Ei-A_j.jóslat

A_j.intervallum:= A_j.intervallum+Hiba/Hibakorrekció

A_j.jóslat:=E_i+ A_j.intervallum

A_j.napló:= [A_j.napló+E_i]

A_j.pontozás= A_j.pontozás + (1-Relatívhiba/2)*E_j.hangsúly

HAVÉGE

CIKLUSVÉGE

Hozzáadni a rendszerhez az új szakértőket

Törölni a duplikált szakértőket

CIKLUSVÉGE

Visszaadni a legmagasabb pontú szakértőt

Az események bekövetkezése szempontjából 3 eset lehetséges.

Az első, hogy a tolerancia ablakon kívül esik az esemény, ekkor nem kell foglalkozni a környező eseményekkel és interpolálni kell egy eseményt egyenlő közökre osztva a kimaradó időintervallumot.

A második, hogy a szűkebb ablakba esik az esemény, ekkor ütem-mérőként kell értelmezni és fel kell jegyezni. Közben a szakértő tempóhipotézise is frissül, úgy hogy hozzáadódik a jósolt esemény és a valódi esemény közötti időkülönbség tört része. A pontozás is módosul, a felvett események hangsúlyának mérőszáma hozzáadódik az aktuális pontértékhez.

A harmadik eset, amikor az esemény a tágabb ablakba esik, ekkor a szakértő elfogadja ütem-mérőként de biztosítékként indít egy új szakértőt, ami nem fogadja el. Így mindkét lehetőség követve lesz és a végén a pontok alapján úgyis a jobb győz.
A számítási erőforrások pazarlása ellen minden ciklusban ellenőrizni kell a duplikálódott szakértők lehetőségét. A szakértők pontosan jellemezhetőek tempófeltevésük és fázisuk segítségével, így ha ebben a két paraméterben nincs túl nagy eltérés akkor azonosnak tekinthetőek. A tempó egyezik, ha 10 ms-nál kisebb a különbség, a fázisnál ez 20 ms. Azt a szakértőt dobjuk el, amelyiknek az összpontszáma a kisebb, így a végén biztosan a legmagasabb pontszámú győz.
A pontozás a ritmikai események jelentőségén múlik. Minél kiemelkedőbb, ívesebb, meredekebb felfutású egy hang, annál biztosabb, hogy ritmikailag fontos helyen áll. Ezeket a paramétereket az audioadatokból nehézkes kinyerni.
A továbbiakban bemutatom a teszteredményeket és az ezeken alapuló fejlesztéseket. Legtöbb problémát a helyes pontozás okozta, mivel lehetetlen volt olyan hangsúlyszámoló függvényt találni, amely minden zenére jól működött.

4.4.2.2. A tempóhipotézist felállító egység teszt eredményei különböző paraméter beállításokkal és küszöbfüggvényekkel

A rendszert 20 különböző klasszikus darabon teszteltem, azt vettem észre, hogy nagyon fontos paraméter a regressziós illesztésnél használt pontok száma, a klaszter konfidencia, valamint a ritmikai események ritkításánál használt alsó küszöböt meghatározó százalék. Végül az algoritmust kétféle tempóhipotézissel valósítottam meg. A táblázatban a fájlnevekhez két sor tartozik, az első a gyakorlati szempontból jobban működő hipotézis-algoritmus eredménye, a második a fentiekben specifikált algoritmus eredménye. A kettő közt annyi a különbség, hogy az első nem hajtja végre a klaszterezés során a rokon klaszterek összevonását.
Első körben az alsó küszöböt úgy határoztam meg, hogy az egész mérési regisztrátum átlagát vettem és ennek valahány százaléka alá eső csúcsok estek ki a lokális maximumok közül. Így meglepően jó eredményt kaptam, azonban az intenzitás szempontból sűrűn változó daraboknál sok lyukas rész keletkezett. Egyértelmű volt,

hogy finomítani kell ezt az alsó küszöb felbontást és valahogy a környező csúcsok értékeihez kell viszonyítani.

Ezen eredmények a következő paraméter beállításokkal születtek. A burkoló számításához az ablak 20 ms, az átfedés 50%, regressziós pontok száma 4, a vizsgált ablakhossz 5 egységnyi, a klaszter konfidencia 3.5 egység (azaz 35ms). A következő táblázatban minden fájlhoz két sor tartozik. A sorok rangsorolva tartalmazzák a tempóhipotéziseket a burkoló felbontásának mértékegységében, azaz 10 ms-os egységekben. Az első sor az egyszerűsített algoritmusból származik, a második pedig a klaszterek rokonságát is figyelembe vevő algoritmusból.

	Tempó hipotézisek rangsorolva (10ms)							
Fájlnév	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
08-Air.wav	83.2	87.6	171.3	129.6	41.6	46.0	177.8	74.8
	175.0	92.0	151.3	74.8	17.7	12.7	41.6	83.2
09-Sabre Dance.wav	16.4	193.8	127.9	48.6	63.6	145.3	161.4	178.3
	48.6	95.8	99.5	16.4	193.8	81.9	111.2	161.4
Bartók - Allegro Barbaro.wav	45.8	68.2	91.6	24.7	138.3	162.5	21.2	113.0
	138.3	135.2	24.7	147.0	116.7	128.0	58.6	45.8
Bizet - Carmen.wav	44.6	70.2	113.8	137.6	159.6	118.8	80.0	48.5
	97.8	181.1	132.8	188.1	37.9	41.1	70.2	58.2
Brahms - 5. Magyar tánc.wav	23.1	9.0	64.0	175.3	39.4	188.5	59.1	93.7
	18.6	129.8	127.0	78.2	39.4	111.1	27.4	23.1
Brahms - 6. Magyar tánc.wav	18.2	191.5	50.0	181.0	38.5	32.1	60.5	80.6
	38.5	98.5	22.4	121.5	158.8	144.9	107.2	29.0
Chicago.wav	179.3	47.5	167.3	119.3	19.4	61.2	53.4	41.3
	123.1	107.1	27.3	23.7	132.6	139.1	13.0	19.4
Copland , Fanfare For The Common Man.wav	18.4	26.3	13.6	30.1	44.6	10.2	6.9	22.4
	37.6	26.3	50.9	18.4	57.6	60.9	13.6	22.4
Copland , Rodeo, Hoedown.wav	132.2	44.3	176.5	93.4	71.7	11.5	26.5	67.8
	22.5	187.2	176.5	54.0	51.0	136.8	44.3	132.2
Emerson - Odeon rag – 77.wav	25.4	75.5	151.1	177.6	51.0	98.8	133.4	84.5
	198.6	142.8	79.5	21.7	67.8	88.6	25.4	98.8
Erkel - Bánk Bán.wav	8.1	21.0	48.9	40.5	130.3	25.7	65.8	61.7
	16.3	40.5	53.7	140.6	70.5	33.0	122.3	61.7
Erkel - Hunyadi László.wav	20.9	145.7	125.5	16.3	25.5	77.7	93.7	104.6
	44.1	145.7	155.0	93.7	135.8	138.9	61.1	25.5
Grieg - Peer Gynt.wav	20.4	44.1	81.4	40.5	110.2	124.6	77.2	91.3

	87.8	153.3	20.4	60.5	16.6	121.2	31.1	34.2
Händel - Solomon- Arrival Of The Queen Of Sheba.wav	36.8	48.7	80.8	101.6	116.5	52.8	175.8	65.3
	48.7	132.3	52.8	43.1	129.0	21.6	25.3	105.1
Liszt - Magyar rapszódia.wav	76.9	192.7	123.8	173.2	42.5	197.1	138.0	142.8
	84.9	72.5	142.8	23.2	55.2	146.0	123.8	165.0
Liszt - Magyr fantázia.wav	9.8	20.2	124.2	44.8	49.4	81.5	153.4	28.6
	20.2	17.2	89.3	31.8	161.9	142.4	78.1	28.6
Lumbye , Champagne Galop, Opus 14.wav	18.4	36.4	196.0	14.1	132.6	185.4	8.8	73.5
	36.4	18.4	28.1	53.6	185.4	56.3	108.6	132.6
Mozart_a_la_turka.wav	7.1	151.8	69.9	19.7	15.6	77.5	142.5	106.7
	15.6	11.8	19.7	69.9	155.0	23.0	65.6	81.0
Offenbach - Kánkán.wav	66.5	94.1	14.8	62.2	33.9	18.5	29.9	185.7
	62.2	58.5	14.8	185.7	140.8	198.1	119.2	52.5
Webber - Phantom of the Opera.wav	91.4	67.6	129.4	73.5	41.6	48.5	17.2	6.9
	13.4	34.3	27.1	73.5	147.3	55.2	20.4	109.7

1. táblázat. Tempófeltevések táblázata, kiemelve a jó eredményt

A színezett cellák a jó eredményt jelölik, ahol nincs ilyen, ott az algoritmus első nyolc feltevése között nincs a helyes tempó. Ez alól kivétel a **Brahms - 6. Magyar tánc.wav** **Copland, Fanfare For The Common Man.wav** mivel ezek annyira expresszív előadásúak, hogy azt én sem tudtam követni, így nem tudtam mihez viszonyítani az eredményeket. Ezért 18 darab fájl az, amelyen a tényleges teszt továbbfolyhat. Az eredmények ellenőrzése úgy történt, hogy írtam egy programot, ami a zene lejátszása közben figyeli az egér klikkelések között eltelt időt és nyolcasával átlagolja azokat. Így az egéren kellett ütnöm a mérőt. Sokszor magamon is észrevettem, hogy egy előző mérési alkalommal feleannyit, vagy dupla annyit kattintottam ugyanarra a zenére, ezért a program esetében is elfogadtam mindkét értéket. A szökőkút vezérlése, a vízjelek, fényjelenségek hangolása szempontjából a dupla tempó nem okozhat gondot.

A következőkben megpróbáltam finomítani a küszöbfüggvényen, a paramétereken és hosszas kísérletezés után (az eredmények megtekinthetőek a mellékelt excel táblákban klaszterx.xls) megszületett a legjobb eredményt adó beállítás. A burkolóhoz az ablak 20 ms, az átfedés 50%, a regressziós pontok száma 5. A küszöböt 10%-os paraméterrel számoltam, a vizsgált ablakhossz 5 egységnyi, a klaszter konfidencia 4.5 egység (azaz 45 ms). Ezek a paraméterek megtalálhatók minden fájlnév alatt a táblázatban.

						Tempó hipotézisek rangsorolva (10ms)							
Fájlnév						**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
08-Air.wav						86.1	170.7	127.8	42.3	166.0	176.0	9.0	7
20	50	5	10	5	4.5	176.0	127.8	166.0	86.1	42.3	170.7	15.0	9
09-Sabre Dance.wav						61.0	14.9	129.6	109.2	79.8	94.6	144.6	4
20	50	5	10	5	4.5	158.5	185.8	61.0	129.6	109.2	144.6	94.6	1
Bartók - Allegro Barbaro.wav						44.8	87.4	64.9	22.4	180.1	154.0	109.5	1
20	50	5	10	5	4.5	87.4	44.8	64.9	131.8	180.1	175.1	114.7	1
Bizet - Carmen.wav						47.9	70.4	187.5	137.7	159.5	90.1	8.3	2
20	50	5	10	5	4.5	47.9	137.7	181.7	187.5	19.6	35.2	24.2	1
Brahms - 5. Magyar tánc.wav						96.2	21.9	58.8	38.1	136.3	150.0	115.4	7
20	50	5	10	5	4.5	190.1	115.4	121.2	196.3	76.9	21.9	33.0	8
Brahms - 6. Magyar tánc.wav						16.3	8.5	54.5	22.7	62.4	90.1	35.3	7
20	50	5	10	5	4.5	35.3	16.3	29.6	107.8	41.2	47.7	123.9	2
Chicago.wav						173.0	180.6	61.9	120.4	114.2	164.5	187.1	1
20	50	5	10	5	4.5	126.8	106.9	14.6	51.8	67.6	19.9	25.6	3
Copland , Fanfare For The Common Man.wav						16.9	10.2	41.2	26.9	58.6	73.8	134.9	1
20	50	5	10	5	4.5	16.9	33.8	84.0	78.8	50.1	26.9	127.8	1
Copland , Rodeo, Hoedown.wav						48.9	181.9	81.9	132.7	73.1	114.5	40.0	1
20	50	5	10	5	4.5	101.4	81.9	144.3	132.7	175.6	66.8	60.1	1
Emerson - Odeon rag - 77.wav						25.1	50.4	98.5	190.2	127.3	78.2	162.4	3
20	50	5	10	5	4.5	78.2	73.2	155.2	98.5	190.2	114.6	90.9	1
Erkel - Bánk Bán.wav						46.9	126.2	118.5	8.6	30.3	52.3	81.9	9
20	50	5	10	5	4.5	19.8	144.8	15.1	166.5	160.2	126.2	134.8	3
Erkel - Hunyadi László.wav						168.0	53.7	177.1	112.7	119.9	104.5	96.3	1
20	50	5	10	5	4.5	177.1	15.5	21.0	161.2	60.9	26.3	37.5	6
Grieg - Peer Gynt.wav						19.9	98.2	83.0	183.2	140.8	43.4	120.6	5
20	50	5	10	5	4.5	38.3	83.0	164.3	114.2	159.1	134.3	170.6	1
Händel - Solomon- Arrival Of The Queen Of Sheba.wav						22.4	140.7	176.2	60.5	49.1	101.5	13.1	1
20	50	5	10	5	4.5	49.1	22.4	124.0	101.5	95.6	150.1	154.7	1
Liszt - Magyar rapszódia.wav						34.7	42.0	69.9	76.5	144.7	105.9	14.1	4
20	50	5	10	5	4.5	69.9	150.9	157.4	99.0	138.7	166.3	186.2	1
Liszt - Magyr fantázia.wav						103.0	121.1	16.6	82.6	61.6	42.2	76.0	1
20	50	5	10	5	4.5	121.1	82.6	30.0	36.7	61.6	42.2	76.0	1
Lumbye , Champagne Galop, Opus 14.wav						18.0	93.1	150.8	73.9	34.6	132.2	113.4	1
20	50	5	10	5	4.5	150.8	144.3	107.8	132.2	101.0	179.8	171.9	1
Mozart_a_la_turka.wav						74.9	187.5	37.5	149.0	112.7	55.7	170.0	1

20	50	5	10	5	4.5	187.5	37.5	170.0	194.6	163.4	30.4	132.8	1
Offenbach - Kánkán.wav						60.7	14.7	185.5	32.0	137.7	46.4	78.4	9
20	50	5	10	5	4.5	125.4	119.6	178.3	92.8	185.5	155.2	98.5	1
Webber - Phantom of the Opera.wav						9.3	36.5	28.4	50.0	44.3	21.6	60.3	1
20	50	5	10	5	4.5	21.6	15.8	72.5	54.9	28.4	119.6	50.0	3

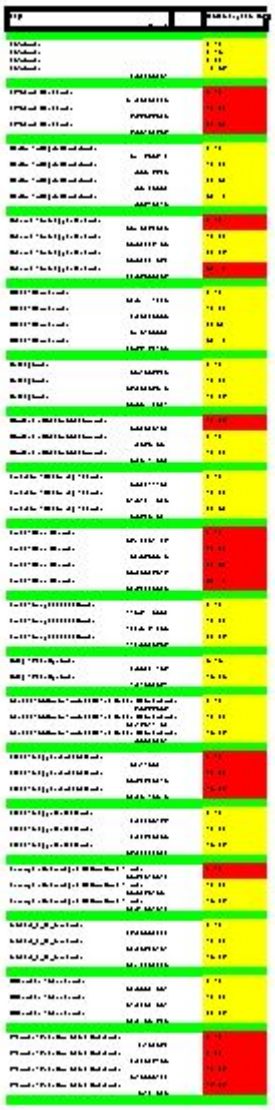
2. táblázat, Tempófeltevések táblázata, kiemelve a jó eredményt

Itt már a specifikációban megadott küszöbszámítást használtam, és 60 másodpercnyi adattal dolgoztam. Látható hogy az eredmények javultak. Sok helyen előrébb kerültek a rangsorban az elfogadható tempó feltevések és a 18 értékelhető fájl mindegyikére a rendszer jó hipotéziseket ad az első 7 helyen. Még egy fejlesztési lehetőség volt, ha módosítottam a klaszterező algoritmuson, úgyhogy nem az 50 ms és 2 s közötti intervallumokat engedélyezi, hanem egy kicsit szűkebb tartományt (200 ms - 1800 ms). Ekkor az első 4 hipotézis között ott volt a jó eredmény, viszont a bankban.wav-ra nem adott jó feltevést a rendszer. (Klaszter1.xls)

Amikor a rendszer első 5 helyre rangsorolt hipotézise között ott volt a jó megoldás, úgy gondoltam, hogy tovább haladhatok és ezt az eredményt már átadhatom a tempóillesztő egységnek, így elkezdtem tesztelni ennek a működését.

4.4.2.3. A tempóillesztő egység, tesztje

A legnagyobb jelentősége a hangsúlyszámítás kifejlesztésének volt. Az első feltevésem az, hogy rendeljük hozzá a felfutási meredekségeket az eseményekhez. Az eredmények kielégítőek lettek, azonban ebben az esetben az események kiválogatására szolgáló algoritmus egy kezdetleges küszöbfüggvényt használt: (melléklet: beatrack1.xls/munka1-es lap, eredeti_beattrack1_munka1.xls)



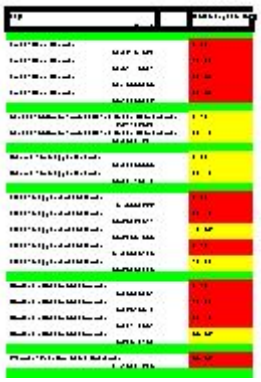
25. ábra. Tempóillesztő egység eredményei

Ezen a képen jól látható színekkel próbáltam illusztrálni a fájlok feldolgozásához tartozó eredményeket. Minden Zöld sorral elválasztott fájlhoz tartozik 3-5 darab eredmény. Ezek 10 szekundumos időintervallumok. A sárga cellák a jó ütem-mérő illesztést mutatják. A pirossal azokat az időintervallumokat jelöltem, ahol a mérő egyáltalán nem stimmelt vagy sokszor elvesztette az ütemet az algoritmus.

A következőkben fejlesztettem a küszöbfüggvényen és a hangsúlyszámító függvényen (salientof.m). Az új eredmények nem lettek számottevően jobbak. A táblázat, csak az előzőekben rosszul detektált fájlokra így néz ki: (melléklet: beatrack1.xls/munka2-es lap, eredeti_beattrack1_munka2.xls):

26. ábra. A tempóillesztő egység eredményi

Próbálkoztam kevesebb szakértő használatával, több pontos lineáris regresszióval a burkoló meredekségének számításánál, mivel úgy gondoltam, hogy a főként fúvósokat, vonósokat tartalmazó daraboknál a hangok hosszabbak. A kottajegyek ívéhez talán ez a modell jobban illeszkedik. (melléklet: beatrack1.xls/munka3-es lap, eredeti_beattrack1_munka3.xls):



27. ábra. A tempóillesztő egység eredményei

A tanulság az, hogy nem jó a 8 pontos regresszió, ez valószínűleg azért van, mert a különálló hangok hossza nem lehet rövidebb 40 ms-nál, mert akkor együtthangzónak érzékeljük. Együtthangzó hangok között nem lehet olyan, hogy a később megszólalón van a hangsúly. A kevesebb szakértő használata arra volt jó, hogy sokkal kisebb lett a számítási idő, mivel ilyenkor az inicializációs periódusban előforduló ritmikai

eseményekre csak két hipotézist illeszt. Így az alaphalmaz, amit fejleszteni és redukálni kezd az algoritmus, sokkal kisebb.

A következőkben kifejlesztettem egy olyan hangsúly megállapító algoritmust, ami figyelembe vette az amplitúdó változását, a hang meredekségét és az időtartamot, amíg a hang felfut.

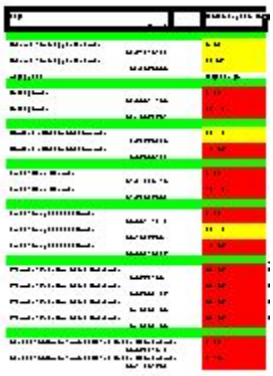
Az eredmények: (melléklet: beatrack1.xls/munka4-es lap, eredeti_beattrack1_munka456.xls/2-es lap):

28. ábra. A tempóillesztő egység eredményei

Itt 20 darab szám van, mert két modern vokális számot is kipróbáltam. Ennél a táblázatnál már kicsit több eredmény született, mert 10 másodperces felbontásban egy

percet vizsgáltam meg a fájlokból. Így egy fájlhoz (két zöld sor közé) 3-6 darab sor társul. Az eredmény sokkal jobb, mint az előző esetben. Nagyon sokszor észrevehető, hogy az algoritmus újra szinkronizálja magát. A hangsúlykereső algoritmussal a rendszer robusztusabban működött. Hirtelen tempóváltáskor elvesztette a ritmust, majd újra rászinkronizálódott. A fennmaradó rossz teszteredmények azt bizonyítják, hogy nem lehet minden zenére jól működő algoritmust létrehozni. A tapasztalatok azt mutatták, hogy a vokális művek, az operák ritmusát igen nehezen veszi fel a rendszer, valamint a vonósok, fúvósok detektálása sem igazán tökéletes. Így az orgona művek, a hegedű, gordonka versenyek és az extrémen expresszív előadású esetekben a rendszer nem működhet. Így a feketelistára a 20 szám közül a következők kerültek*: Brahms – 5. Magyar tánc.wav, Chicago.wav, Erkel - Bánk Bán.wav, Erkel - Hunyadi László.wav, Händel - Solomon- Arrival Of The Queen Of Sheba.wav, Liszt - Magyar rapszódia.wav, Webber - Phantom of the Opera.wav*

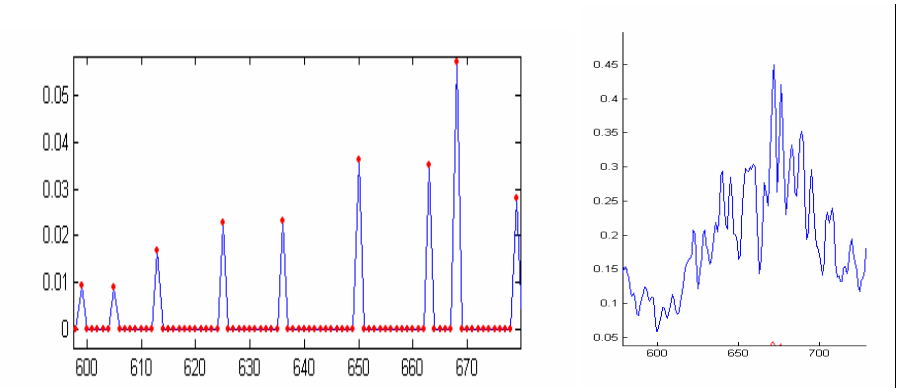
Ha százalékosan akarjuk kifejezni, ez 65%-ára jól illesztett ütem-mérőt. Ezt fokozni szerettem volna, így a rosszul működők esetében egy kis módosításra volt szükség. Először is az volt az előző egység hibája, hogy a két darab tempóhipotézis között az általam mért és jónak tartott tempó nem is szerepelt, így bővítettem a tempóhipotézisek számát 5-re. Ezzel az 5. Magyar táncra már jól működött. A többire viszont nem javult az eredmény: (melléklet: beatrack1.xls/munka5-ös lap, eredeti_beattrack1_munka456.xls/3-es lap):



29. ábra. A tempóillesztő egység eredményei

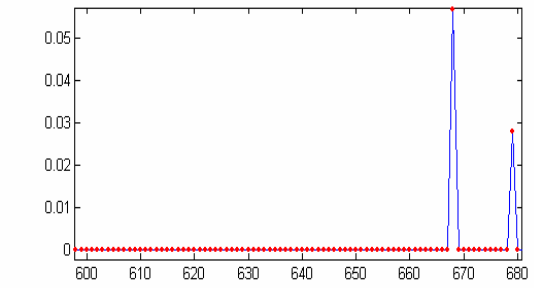
Próbálkoztam a hangsúly függvény változtatásával, az sem segített. Mivel hiába súlyozom a meredekség paramétert, arra a következtetésre jutottam, hogy egy ritmikai esemény körül valószínűleg túl sok a kimagasló csúcs, amit megtalál és ezek közel azonos nagyságúak. Ez akkor lehetséges, ha egy-egy hang felfutás hosszabb ideig tart és

az esemény-redukció során az algoritmus 50 ms-os ablaka (5 beosztás) túl rövid, így egy hanghoz több eseményt is hozzárendel. A következő ábrákon az idő az x tengelyen 10 ms-os egységekben van ábrázolva.



30. ábra. Kitartott fuvola hang

Händel - Solomon- Arrival Of The Queen Of Sheba című darabjának hatodik másodpercében megszólaló kitartott fuvola hang, ezen jól látszik, hogy az algoritmus egy fél szekundumos tartományban, a felfutás alatt legalább 4 nagyságrendileg azonos csúcsot detektál. Ezért az ablakot 200 ms hosszúra választottam, az eredmények meglepően jók lettek.



31. ábra. A ritmikai eseményhez tartozó csúcsok

A fúvós, vonós darabokra is tökéletesen illesztett tempót az algoritmus:
(melléklet: beatrack1.xls/munka6-os lap, eredeti_beattrack1_munka456.xls/4-es lap):

32. ábra. A tempóillesztő egység eredményei

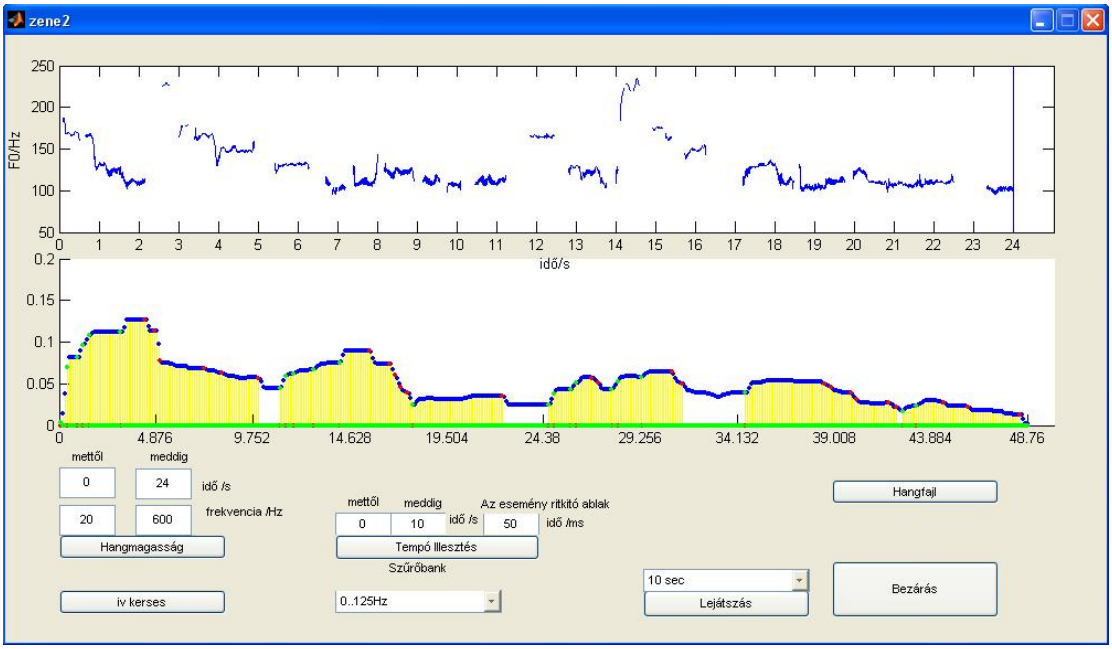
Még mindig nem működött megfelelően az algoritmus: Erkel - Bánk Bán.wav,
Liszt - Magyar rapszódia.wav, Webber - Phantom of the Opera.wav, A Magyar rapszódia egy sokkal nagyobb (1000ms-os) ablakméretre tökéletesen működött, ha a teljes egy perces regisztrátumot analizáltam. A másik két darab tempódetektálása nem volt elég jó.

Továbblépési lehetőségként a ritmikai esemény detektálásán kellene javítani, valamilyen hosszabb ablakkal először le kellene válogatni a nagy csúcsokat és ezek körül újra kellene szűrni a kisebb csúcsokat és újra visszarakni. Ez segítene az elveszített ritmus gyorsabb megtalálásában. A hangsúly függvény további tesztelésére volna szükség, hogy még pontosabban meg tudjuk mondani egy hang erősségét, mértékét a ritmikában.
Emellett megfigyeltem, hogy sok esetben a rendszer 3-at vagy 5-öt ütött addig, amíg én kettőt. Tehát minden harmadik vagy ötödik ütés volt a helyén. Ezt úgy tudjuk kiküszöbölni, hogy minden egyes szakértőhöz tartózó ütésszám értékre normálnánk a végső pontozást. A tempóhipotézis rész felállít egy sorrendet, ezt nem súlyozva használja a tempóillesztő rész, sőt egyenlőre nem vesszük figyelembe a tempók egymáshoz képesti arányát, tehát nem súlyozunk f(d)-vel. Így a fázisillesztő rész végén azt kellene tenni, hogy kiválasztjuk a legtöbb ütést predesztináló jó eredményt elérő szakértőt, majd az összes többit, pontozását normáljuk erre a szintre, tehát mintha a többi szakértő is ennyit ütött volna. Mivel az nem helyes, hogy amelyik kevesebbet üt, de egy kicsit alacsonyabb hangsúlyú ritmikai eseményekkel korrelál, azt a magasabb ütésszám miatt előrébb rangsoroljuk. Erre írtam egy kezdetleges verziót de ez nem hozott egyelőre értékelhető eredményjavulást.

5. Grafikus felhasználói felület

A grafikus felületet is Matlab-ban fejlesztettem. A legújabb verziókban már van egy külön eszközcsoport, ami segíti grafikus, interaktív ablakok létrehozását, így a grafikai részek implementálása, az egérkezelés, a gombok működtetése nem volt túlságosan bonyolult. A legfőbb problémát a hang és animáció összeegyeztetése jelentette.

A felület két fő rajzterülettel és a vezérlő gombokat magába foglaló panellel rendelkezik.



33. ábra. Grafikus Felhasználó felület

A Felületen az egér segítségével navigálhatunk.

Hangfájl gomb:

Ennek a gombnak a segítségével betölthetjük az elemezni kívánt wav fájlt. A gombnyomás hatására előugrik egy fájlkezelő ablak, amelyben a számítógép adathordozóinak tartalmából válogathatunk.

ívkeres gomb:

Ez a gomb az alsó grafikus panelhoz van rendelve, az ábrán látható módon megjeleníti a teljes hangfájl burkolóját és sárgára színezi a motívum kereső algoritmus által kijelölt egy ív alá tartozó részeket. Egy külön ábrán ábrázolja a burkoló alapján készített hisztogramot és a ráillesztett exponenciális görbét.

A megbízók fontosnak találták egy szűrőbank beépítését, amely [0-125Hz; 125-250Hz; 250-500Hz; 500-1KHz; 1-2KHz; 2-4KHz; 4K-8KHz; 8K-22KHz] 8 sávban szűri a bementi jelet. Ezért a **'Szűrőbank'** címszó alatt egy legördülő menü található, ahol kiválaszthatjuk milyen sávszűrőt szeretnénk alkalmazni. Az eredmény a felső panelen jelenik meg.

Lejátszás gomb:

A lejátszás gomb arra szolgál, hogy le tudjuk ellenőrizni mennyire számolt helyesen az algoritmus. A gomb megnyomásakor elindul a hangfájl lejátszása és közben egy kék marker az alsó grafikuspanelen mutatja, hogy hol jár a zene. Így vizuálisan is követni tudjuk az eredményeket. Kiválaszthatjuk mennyi ideig szeretnénk lejátszani a fájlt. Ez azért hasznos, mert a Matlab nem ad lehetőséget megállítani a lejátszást.

Hangmagasság gomb:

Ehhez a gombhoz négy bemenő paraméter tartozik. A beviteli mezők felső sorába azt adhatjuk meg, hogy melyik intervallumot szeretnénk hangmagasság analizálásnak alávetni. Az első cellába kezdés idejét kell beírni másodpercben, alapértelmezésben a 0. szekundomtól indul az analizálás. A második cellába az analízis végének idejét kell beírni szintén másodpercben, az alapérték 10 szekundum.

A második sorba beírhatjuk a minimum és maximum frekvenciát, amivel érdemes foglalkozni a feldolgozás során az algoritmusnak. A hangmagasság detektáló algoritmus leírásakor, ezekre az értékekre fmin és fmax-ként hivatkoztam. Az eredményesség nagyban javítható, ha az alsó frekvenciát a megfelelő értékre állítjuk, általában 0 és 500 Hz között.

Tempóillesztés gomb:

Ez a gomb a beállított intervallumra elindítja a tempófeltevést végző és tempóillesztő algoritmusokat. Az esemény ritkító ablak segítségével, azt az ablakhosszt állíthatjuk be

milliszekundumban, amit a ritmikai-eseménykereső algoritmus használ az értékes csúcsok kinyerésére. A tesztelési ciklus alatt ennek a paraméternek a változtatásával értem el a legjobb eredményt. Ajánlott értéke: 50 ms – 2000 ms-ig. Nem szabad túl hosszú intervallumot analizálni, mivel nagyon számításigényes a tempóillesztő algoritmus működése. Kis ablakhossz használatakor sok ritmikai eseményt talál az algoritmus, ilyenkor maximum 20 szekundumos intervallum elemzése megengedett.

A rendszer az összes kiszámolt eredményt időben markerezve, előre megadott felbontással (alapértelmezett 100ms) egy fájlban rögzíti. A szűrőbankos kimeneti eredmények esetén minden sávot külön fájlban rögzít, ezen fájlok paraméterei (méret, mintavételi frekvencia, szóhossz) az eredeti audio fájlal megegyeznek.

6. Összefoglalás

Munkám során egy speciális feladat megoldásával foglalkoztam, amelyet a Szabados & Társai Kft. határozott meg számomra. Zenélő szökőkutak vezérlését megkönnyítő alkalmazás megvalósítására kellett törekednem. A cél egy olyan leírófájl kinyerése zenei audio fájlból, amely tartalmazza a zenére jellemző motívumokat, paramétereket.

A feladat megoldása során foglalkoztam zenei ívek keresésével, dallamkereséssel, ritmusfelismeréssel, majd létrehoztam egy kezelői felületet. A teljes rendszert Matlab program segítségével implementáltam. Négy jól elkülöníthető egységre osztható a munkám.

Az első a zenei motívumok, ívek keresésére vonatkozó szakasz, melynek végső eredménye egy olyan algoritmus, amely apriori információk beadása nélkül, pusztán a bementi adatok (hangfájl) statisztikája alapján működik. A paraméterek kinyerésére egy nem szokványos megoldás szolgál, amely egy empirikus úton történő fejlesztés eredménye. Az algoritmus a statisztikán alapuló paraméterek meghatározása után több hagyományos jelfeldolgozási eljárást hajt végre, majd a zenére jellemző hangfájl burkolójában kijelöli az íveket, motívumokat. A rendszer a széles dinamika- és hangerőtartománnyal rendelkező, halk-hangos részeket váltogató zenére működik tökéletesen. Így a véletlenszerűen válogatott zenék esetében 60% körüli átlagteljesítményt nyújt az ívek meghatározásában. Egy-egy zenénél nyújtott teljesítmény nagyban függ a zenei minőségtől, műfajtól.

A második szakaszban átfogóan tanulmányoztam a dallamkeresésre, hangmagasság detektálására szolgáló algoritmusokat. Ebben az egységben egy, egyszólamú környezetben kielégítően működő rendszert implementáltam. Frekvenciatartománybeli feldolgozás segítségével, fázisinformációk kinyerésével az algoritmus minden egyes időpillanatban meghatároz egy feltételezhető alapfrekvenciát, amelyet később újra megvizsgál és módosít, végezetül kialakítja a végleges dallamvonulatra vonatkozó feltevését. A kielégítő megoldás elérése érdekében, itt vannak beadandó paraméterek, például a minimális és maximális frekvencia, amik között az alapfrekvenciát a rendszer keresi. A rendszer vokális előadások esetében 100%-os hatásfokkal működik.

A harmadik legmeghatározóbb szakasz a tempófelismeréssel foglalkozik. Az egység működése két részre bontható, a tempófeltevésre (tempóhipotézisre), és a tempóillesztésre. Az előbbiben a rendszer klaszterező algoritmus segítségével rangsort

állít fel a ritmust illetően, teljesítménye 95%-os hatásfokú. Az utóbbiban szakértő csoportok segítségével, a tempóhipotéziseket fázisban illeszti a zenére. Itt a hatásfok 80% körüli.

A feladat negyedik szakaszában egy grafikus ellenőrző felületet implementáltam Matlab-ban. A felületről elérhetőek az előzőekben említett egységek, kimenetük vizuálisan is megjeleníthető és meghallgatható.

A rendszer által kinyert paraméterek nem minden zenei fájlra tökéletesek. A tempóillesztő algoritmusnál a hangsúlyszámítási függvény hatásfokát frekvenciatartománybeli analízissel lehetne fokozni. A motívumkereső algoritmusba heurisztikákat kellene beépíteni az egymást követő ívek alakjára vonatkozóan, a hangmagasság detektáló algoritmust pedig hangszermodellekkel kellene ellátni a többszólamú környezetben történő működés hatékonyságának növelése érdekében.

A továbbfejlesztési lehetőségként egy olyan rendszer kialakítása a cél, amelyben a kinyert paramétereket a vízkép koreográfus szabadon felhasználhatja, hozzárendelheti különböző vízkép szubrutinokhoz. Ezen felhasználó alkalmazásnak az aktuális szökőkút paramétereit szem előtt tartva grafikus szerkesztőfelülettel kell rendelkeznie, platform függetlennek kell lennie és kimeneti eredményül a végleges vezérlőprogramot kell szolgáltatnia.

7. Irodalom jegyzék

(Cemgil, A., et al, 2001) Cemgil, A., Kappen, B., Desain, P., and Honing, H. (2001). On tempo tracking: Tempogram representation and Kalman filtering. Journal of New Music Research.

Clarisse L. P., Martens J. P., Lesaffre M., De Baets B., De Meyer H. and Leman M. (2002). "An Auditory Model Based Transcriber of Singing Sequences", In Proc. International Conference on Music Information Retrieval – ISMIR'2002.

Clarke, E. (1988). Generative principles in music performance. In Sloboda, J., editor, Generative Processes in Music, pages 1–26. Clarendon Press, Oxford.

Clarke, E. (1999). Rhythm and timing in music. In Deutsch, D., editor, The Psychology of Music, pages 473–500. Academic Press, San Diego CA.

Desain, P. (1992). A (de)composable theory of rhythm perception. Music Perception, 9:439–454.

Desain, P. (1993). A connectionist and a traditional AI quantizer: Symbolic versus subsymbolic models of rhythm perception. Contemporary Music Review, 9:239–254.

Drake, C., Penel, A., and Bigand, E. (2000). Tapping in time with mechanically and expressively performed music. Music Perception, 18(1):1–23.

Emilia Gómez Gutiérrez, (2001). Melodic Description of audio signals for music content processing.

Gerhard D. (2000). Audio Signal Classification, PhD Depth Paper, School of Computing Science, Simon Fraser University, Canada.

Gerhard D. (2003). Pitch Extraction and Fundamental Frequency: History and Current Tech-niques, Technical Report, Department of Computer Science, University of Regina, Canada.

Gerőfi Balázs (2005). Hangegérhang és beszédfelismerés Diplomamunka

Gitáriskola internetes forrás: http://www.gitariskola.hu/ritmika.html

Goebl, W. (2001). Melody lead in piano performance: Expressive device or artifact? Journal of the Acoustical Society of America, 110(1):563–572.

Gold B. and Rabiner L. (1969). "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain", Journal of the Acoustical Society of America, Vol. 46, No. 2, pp. 442-448.

Goto, M. and Muraoka, Y. (1995). A real-time beat tracking system for audio signals. In Proceedings of the International Computer Music Conference, pages 171–174, San Francisco CA. International Computer Music Association.

Goto, M. and Muraoka, Y. (1997a). Issues in evaluating beat tracking systems. In Issues in AI and Music – Evaluation and Assessment: Proceedings of the IJCAI'97 Workshop on AI and Music, pages 9–16. International Joint Conference on Artificial Intelligence.

Goto, M. and Muraoka, Y. (1997b). Real-time rhythm tracking for drumless audio signals– chord change detection for musical decisions. In Proceedings of the IJCAI'97 Workshop on Computational Auditory Scene Analysis, pages 135–144. International Joint Conference on Artificial Intelligence.

Goto, M. and Muraoka, Y. (1998). An audio-based real-time beat tracking system and its applications. In Proceedings of the International Computer Music Conference, pages 17–20, San Francisco CA. International Computer Music Association.
34

Goto, M. and Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals. Speech Communication, 27(3–4):331–335.

Hawley M. (1993). Structure Out of Sound, PhD Thesis, Media Laboratory, Massachusetts Institute of Technology, USA.

Jarno Seppänen, (2001). Computational models of musical meter recognition. Master of Science Thesis

Kashino K., Nakadai K., Kinoshita T. and Tanaka H. (1995). "Organization of Hierarchical Perceptual Sounds: Music Scene Analysis with Autonomous Processing Modules and a Quantitative Information Integration Mechanism", In Proc. International Joint Conference on Artificial Intelligence – IJCAI'95, pp. 158-164.

Klapuri A. P. (2003). "Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness", IEEE Transactions on Speech and Audio Processing, Vol. 11, No. 6, pp. 804–816.

Klapuri A. P. (2004). Signal Processing Methods for the Automatic Transcription of Music, PhD Thesis, Tampere University of Technology, Finland.

Large, E. and Kolen, J. (1994). Resonance and the perception of musical meter. Connection
Science, 6:177–208.

Lahat A., Niederjohn R. J. and Krubsack D. A. (1987). "A Spectral Autocorrelation Method for Measurement of the Fundamental Frequency of Noise-Corrupted Speech", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 35, No. 6, pp. 741-750.

Lerdahl, F. and Jackendoff, R. (1983). A Generative Theory of Tonal Music. MIT Press, Cambridge MA.

Maher R. C. (1990). “Evaluation of a Method for Separating Digitized Duet Signals”, Journal of the Audio Engineering Society, Vol. 38, No. 12, pp. 956-979.

Masataka Goto and Yoichi Muraoka, (1994). A Beat Tracking System for Acoustic Signals of Music

Masataka Goto, Yoichi Muraoka, (1999b). Real-time beat tracking for drumless audio signals:Chord change detection for musical decisions

Masataka Goto, (2001). An Audio-based Real-time Beat Tracking System for Music With orWithout Drum-sounds

Moorer J. A. (1977). “On the Transcription of Musical Sound by Computer”, Computer Music Journal, Vol. 1, No. 4, pp. 32-38.

Nick Collins, (2004). Beat Induction and Rhythm Analysis for Live Audio Processing:1st Year PhD Report

Nick Collins, (a).Towards a Style-Specific Basis for Computational Beat Tracking

Noll A. M. (1967). “Cepstrum Pitch Determination”, Journal of the Acoustical Society of America, Vol. 41, No. 2, pp. 293-309.

Povel, D. and Essens, P. (1985). Perception of temporal patterns. Music Perception, 2(4):411–440.

Repp, B. (1994). On determining the basic tempo of an expressive music performance. Psychology of Music, 22:157–167.

Rosenthal, D. (1992). Emulation of human rhythm perception. Computer Music Journal, 16(1):64–76.

Rowe, R. (1992). Machine listening and composing with Cypher. Computer Music Journal, 16(1):43–63.

Rui Pedro Pinto de Carvalho e Paiva (September 2006). Melody Detection in Polyphonic Audio

Ryynänen M. P. (2004). Probabilistic Modelling of Note Events in the Transcription of Mono-phonic Melodies, MSc Thesis, Tampere University of Technology, Finland.

Ryynänen M. P. and Klapuri A. (2005a). "Polyphonic Music Transcription Using Note Event Modeling", In Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics – WASPAA'2005.

Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. Journal of the Acoustical Society of America, 103(1):588–601.

Schloss,W. (1985). On the Automatic Transcription of Percussive Music: From Acoustic
Signal to High Level Analysis. PhD thesis, Stanford University, CCRMA.

Simon Dixon (2001). Automatic Extraction of Tempo and Beat from
Expressive Performances

Steedman, M. (1977). The perception of musical rhythm and metre. Perception, 6:555–569.

Stephen W. Hainsworth, (2003). Techniques for the Automated Analysis of
Musical Audio

Sterian A. D. (1999). Model-based Segmentation of Time-Frequency Images for Music Transcrip-tion, PhD Thesis, Department of Electrical Engineering and Computer Science, University of Michigan, USA.

Sundberg, J. (1991). The Science of Musical Sounds. Academic Press, San Diego CA.

Talkin D. (1995). “A Robust Algorithm for Pitch Tracking”, In Kleijn W. B and Paliwal K. K. (eds.): Speech Coding and Synthesis, pp. 495-518, John Wiley & Sons.

Viitaniemi T., Klapuri A. and Eronen A. (2003). “A Probabilistic Model for the Transcription of Single-Voice Melodies”, In Proc. Finnish Signal Processing Symposium – FINSIG’2003.

Udo Zozler. (2003). DAFX - Digital Audio Effects

Wikipedia. 2007 - http://hu.wikipedia.org/wiki/Kateg%C3%B3ria:Komolyzenei_m%C5%B1fajok