Zeneszám felismerés

Mérési feladat

Bevezetés

Ebben a mérési feladatban különböző, zeneszám felismerésre is használható hangminta illesztő módszereket fogunk megvizsgálni. A vizsgálat szempontjai: az egyes módszerek sebessége, hatékonysága, különböző akusztikai viszonyok, statisztikai mutatók.

Az alap feladat, aminek különféle variációit le kell futtatni, egy lehetséges zeneszám felismerő rendszer elemi lépése: Adott a referencia zeneszám-adatbázisban egy vagy több zeneszám (valójában most azoknak csak egy-egy részlete, annak érdekében, hogy gyorsan lefussanak a mérések), és egy rövid, többé-kevésbé zajos zeneszám részlet, amiről meg kell állapítani, hogy előfordul-e a referencia zeneszámban, és ha igen, időben hol kezdődik.

A módszerek, amelyekkel az illesztést el kell végezni, a következők:

- időtartartománybeli illesztés csúszó ablakkal (mint a legegyszerűbb, feldolgozást nem igénylő módszer)
- frekvenciatartománybeli (amplitúdóspektrum) illesztés csúszó ablakkal (mint az MFCC lényegkiemelés köztes állomása)
- MFCC jellemzővektor illesztés csúszó ablakkal (ez a módszer a gyakorlatban is használt)
- ujjlenyomat alapú illesztés (ez is egy gyakorlatban használatos módszer)

A különböző akusztikai viszonyok az azonosítani kívánt zeneszám részletre vonatkoznak, a referencia hanganyag mindig a tiszta, eredeti, album-változata a zeneszámnak. Különféle - többé-kevésbé mesterséges, digitális és analóg - torzításokat eszközölhetünk a referencia-zeneszám egy rövid részletén, különféle módszerekkel.

A mérési könyvtárban található könyvtárak és fájlok:

fingerprint: Az ujjlenyomatos illesztő rendszer megvalósítása Matlab/Octave nyelven.

rastamat: Az MFCC lényegkiemelés megvalósítása Matlab/Octave nyelven.

octave_scriptek: A mérés scriptjeihez felhasznált egyéb eszközök Matlab/Octave nyelven.

f1.m, f2*.m, f3.m, f4.m: A mérési feladatokat többnyire ezen scriptek kiegészítésével, módosításával kell megoldani.

sox_scriptek: Az Octave-on kívül, shellben használható, SOX-ot futtató hangfeldolgozó scriptek találhatók itt.

zeneszamok: A méréshez felhasználható zeneszám részletek. 10 db különböző stílusú, kb. 16 mp-es zeneszám részletet tartalmaz, 16 kHz-es mintavételezéssel, 16 bites PCM kódolással, WAV formátumban.

mikrofon: Az egyik referencia zeneszámrészlet analóg torzítást is tartalmazó, hang formában is realizálódott változata. Egy mobiltelefon hangszórójával lett lejátszva, és egy laptop mikrofonjával felvéve.

zajok: Néhány rögzített környezeti zaj található itt. A http://www.steeneken.nl/7-noise-data-base/ helyről származnak. Hadászati gépek, járművek zaját tartalmazzák, a 014-es és 019-es számú fájlban beszéd is hallatszik. Ezeknek a tiszta zeneszámrészlethez való adásával lehet mesterségesen zajosítani az illesztendő hangmintát.

csv: Itt található egy CSV táblázat fájl, ami egy nagyobb szabású landmarkos keresztbe tesztelés eredményét tartalmazza.

1. feladat

Az ***f1.m*** Octave scriptben az időtartománybeli illesztés, amplitúdóspektrum illesztés, MFCC jellemzővektor illesztés és ujjlenyomat alapon illesztés módszerek vannak megvalósítva 1 referencia fájlra és 1 keresendő zeneszámra. Hasonlítsa össze a 4 módszert sebesség szempontjából! Használja referenciának a ***zeneszamok/lvnp.wav*** hangfájlt!

a) Készítse el a keresendő mintát: legyen a referencia hangfájlnak az 5-10 mp-ig terjedő szakasza, mp3 kódolással. (Használja a sox_scriptek könyvtárba lévő .sh scripteket.)

b) Írja be az f1.m script elejére a referencia és keresendő fájlok neveit!

Az ***f1.m*** script ezzel futtatásra kész.

c) Tanulmányozza az f1.m scriptet, és időmérő utasítások elhelyezésével mérje meg, hogy mennyi ideig tart a referencia hangfájlok előfeldolgozása (FFT, MFCC lényegkiemelés, ujjlenyomatozás)! Közölje az eredményeket RTF (real-time factor) mérőszámokként!

d) A c) részfeladathoz hasonlóan mérje meg mind a 4 módszerre külön, hogy mennyi ideig tart a keresendő hangminta teljes feldolgozása, tehát az előfeldolgozás és az illesztés a már előkészített referencia anyagokkal. Közölje az eredményeket a keresendő hangmintára vonatkoztatott RTF mérőszámokként!

Figyelem! Az időtartománybeli illesztés során mintánként (0.0625 ms) kell eltolnunk egymáshoz képest az épp vizsgált és a referencia hanganyagokat (hiszen nincs keretezés). Ennek következtében ez nagyon lassú! Ezért ezzel a módszerrel egy rövidebb teszt van előkészítve a scriptben, olyan módon, hogy a referencia tesztanyag le van rövidítve. Így csak 2 mp-nyi ablak csúsztatás lesz lefuttatva, ami még könnyedén kivárható. Ezt szem előtt kell tartani az RTF kiszámításakor.

e) Mit gondol, milyen gyakorlati alkalmazásoknál lehet kritikus az előbbi, és melyeknél az utóbbi feldolgozási lépés sebessége?

2. feladat

a) Készítsen az 1. feladatban is használt zeneszám kivágott részletéből 8 féle változatot:

1. tisztán, mindenféle torzítás nélkül
2. a mikrofon könyvtárban található változat

3. adja hozzá az eredetihez a **zajok/SIGNAL019-20kHz-2min_16k.wav** beszélgetés-zajt
4. gyorsítsa az eredetit 10%-kal hangmagasság változás nélkül
5. keverjen rá egy másik zeneszámot az eredetire
6. szűrje felüláteresztő szűrővel 4kHz alatt
7. kódolja GSM kodekkel az eredetit
8. kódolja MP3 kodekkel az eredetit

Ellenőrzésképpen hallgassa meg az elkészített hangmintákat!

b) Illessze be a létrehozott fájlok neveit a f2_wav_nevek.m Octave script elején a* test{...}='' *értékadások jobb oldalára! Futtassa le a f2_wav.m, f2_AS.m, f2_MFCC.m, f2_ujj.m scripteket!

A scriptek elvégzik a szükséges előfeldolgozásokat, majd mind a 8 megadott hangmintát illesztik az 1. feladatban is használt zeneszámmal, a hullámforma, amplitúdóspektrum, MFCC, ujjlenyomat alapú módszerekkel. Az egyes scriptek a következőképpen jelenítik meg az eredményeket:

A hullámforma (f2_wav.m), amplitúdóspektrum (f2_AS.m) és MFCC alapú (f2_MFCC.m) illesztés csúsztatott euklidészi távolság számítás módszerével működik, és hasonló ábrákat készítenek. Mind a 3 script két-két ábra-ablakot (figure) hoz létre. Az elsőn az y tengelyek automatikusan vannak skálázva, hogy kitöltsék az adatok a helyet, a másodikon ugyanazok az adatok láthatók, de egységes y tengellyel. (A hullámforma illesztés az 1. feladathoz hasonlóan itt is a referencia hangminta egy rövidebb szakaszára történik, hogy gyorsabban lefusson.) A 8 subplot a 8 tesztfájlnak felel meg.

Az ujjlenyomatos illesztő script (f2_ujj.m) tesztfájlonként egy-egy ábra-ablakot készít. Mindegyiken 2 subplot látható. A felső a teszt hangminta, az alsó a referencia hangminta spektrogramja, az illesztett helyen. Zölddel vannak jelölve az illeszkedő landmarkok, az ábrák feliratairól leolvasható az illeszkedő landmarkok száma, az egyezés ideje (a referencia fájl kezdetétől mérve), és az illesztett hangminta fájl neve.

Hasonlítsa össze, vizsgálja meg az euklidészi távolság alapú illesztésből származó ábrákat, és megfigyeléseit írja le az alábbi kérdésekre válaszolva!

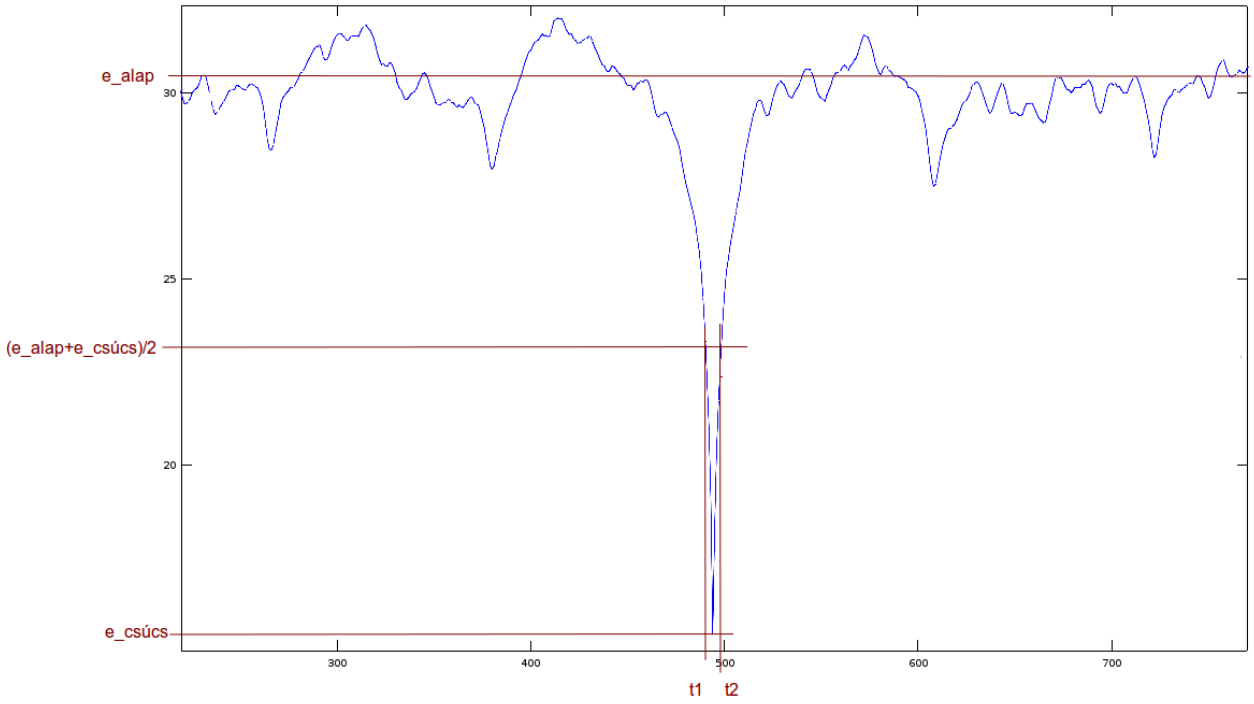
c) Az euklidészi távolság időfüggvények ábráin miből látszik, hogy a zajosabb hangminták azonosítása nehezebb feladat a különböző módszerek számára?

d) Az előző pontban vizsgált ábrák alapján állapítsa meg, hogy az egyes módszerek mely torzításokat viselnek el jobban, és melyeket rosszabbul?

e) Becsülje meg a hullámforma, amplitúdóspektrum és MFCC alapú illesztő módszerek működéséhez minimálisan szükséges időbeli felbontást a következő módon:

A jelenlegi tesztekben minden lehetséges időpontban (mintavételi illetve keret lépésnél) elvégezzük az illesztést. A feldolgozási sebesség növelése érdekében érdemes lenne ritkábban illeszteni. Tegyük fel, hogy a csúcs kereső algoritmus küszöbértékét úgy szeretnénk beállítani, hogy a jelenlegi tesztekben kapott csúcsokhoz képest fele mélységű csúcsokat képes legyen detektálni. A minimálisan szükséges időbeli felbontást megkapjuk úgy, ha megvizsgálunk az euklidészi-távolság-sorozat ábrán egy csúcsot, és leolvassuk róla, hogy időben milyen távol van egymástól a csúcsot

jelentő lokális minimum előtti és utáni fél-mélység átlépés. Lásd az alábbi ábrát.



Vizsgáljon meg két hangmintával kapott ábrasorozatot: a tiszta és a GSM kódolt hangminta ábráit! A hullámforma, amplitúdóspektrum és MFCC illesztéssel kapott euklidészi távolság-sorozat diagramok megfelelő nagyításával olvassa le a t2-t1 értékét, és adja meg ms-ban.

Figyelem! A különböző illesztő módszerek ábráin az x tengely lépésköze más és más. A hullámforma illesztés lépésköze 1 minta (1/16000 s), az amplitúdóspektrum illesztésé 256 minta, az MFCC illesztésé 10 ms.

Vizsgálja meg az ujjlenyomatos illesztés spektrogramos ábráit, és megfigyeléseit írja le a következő kérdésekre válaszolva:

f) Milyen jellemző különbségeket figyelt meg a torzított és eredeti spektrogramok között? Írja le a megfigyeléseit 2 jellegzetesebben módosult spektrogramról.

g) Mely torzításokat viseli el jobban az ujjlenyomatos módszer, és melyeket rosszabbul?

3. feladat

Keresse meg az ujjlenyomatos és az MFCC zeneszámfelismerő módszer működőképességének határát! Készítsen olyan torzított, zajosított változatokat az 1. és 2. feladatban is használt azonosítandó hangmintából, amelyekben füllel még felismerhető a zene dallama, de az azonosító módszerek már nem járnak sikerrel!

A feladat megoldásának ellenőrzéséhez segítséget nyújt az f3.m fájl. A létrehozott torzított hangminta fájl nevét bele kell írni, lefuttatva a script elvégzi az illesztést a referencia zeneszámmal mindkét módszerrel, majd kiírja, hogy talált-e egyezést.

a) Készítsen egy torzított változatot az lvnp.wav 5 mp-es részletéből úgy, hogy az f3.m script ne

tudja azonosítani a landmarkos módszerrel!

b) Készítsen egy másik torzított változatot az lvnp.wav 5 mp-es részletéből úgy, hogy az f3.m script ne tudja azonosítani az MFCC módszerrel!

Használhatja a megadott sox-scripteket, octave utasításokat, függvényeket, esetleg külső hangszerkesztő programot, stb.

4. feladat

Ebben a feladatban egy korábban elvégzett tesztsorozat eredményét kell elemezni, amelyek a **csv/kereszt_teszt_2000.csv** fájlban találhatók. Ez a fájl egy 2000x2000 es mátrixot tartalmaz, amely 2000 zeneszámrészlet egymással keresztbe tesztelésének eredménye, az ujjlenyomatos módszerrel. Az i. sor j. oszlopa azt jelenti, hogy az i. zeneszámnak az adatbázis számaihoz való hasonlításkor a j. adatbázisbeli zeneszámmal hány landmark illeszkedett. (A tesztek futtatása több órát vett igénybe, ezért maga a futtatás nem része a mérésnek.)

Az egyes zeneszámok önmagukkal összehasonlítva nyilván nagy fokú egyezést mutatnak, az adatbázis más zeneszámaival összehasonlítva pedig (jól összeállított adatbázis esetén) sokkal kisebb mértékű a hasonlóság. Ezért a mátrix átlójában nagy számok vannak, a többiben kicsik vagy 0-k. Akkor tekintsük az i. zeneszámot illeszkedőnek a j. zeneszámmal, ha a mátrix i. sorának j. oszlopában tárolt landmark illeszkedések száma elér egy K küszöbértéket.

Értékelje ki a fenti mátrix formájában rendelkezésére álló részeredményeket! Használja ehhez az ***f4.m*** scriptet! A script összehasonlítja a mátrix értékeit a K változóban meghatározott küszöbértékkel, és kiszámítja a true positive, false positive, true negative, false negative, accuracy, recall, precision, F-measure mérőszámokat.

a) Futtassa le a scriptet, majd módosítsa K értékét úgy, hogy minden zeneszámot felismerjen! Melyik mérőszám milyen értékét jelenti ez az eset?

b) Próbáljon meg elérni a K érték állításával, hogy a hibásan (is) felismert zeneszámok darabszáma 20 alatt legyen! Hogyan változik eközben a hamis negatívok száma? Mi lehet ennek az oka?

c) Soroljon fel néhány zeneszám-sorszámot, amelyeket érint a b) pontban felfedezett adatbázis-hiba!