Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Frenyó Péter

BELTÉRI AKUSZTIKUS LOKALIZÁCIÓ ÉS FORRÁSAZONOSÍTÁS

Msc Diplomatervezés

KONZULENS

dr. Orosz György

BUDAPEST, 2014

Tartalomjegyzék

HALLGATÓI NYILATKOZAT

Alulírott **Frenyó Péter**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2014. 12. 19.

...……

Frenyó Péter

Kivonat

Bizonyos alkalmazásokban szükséges lehet egy adott térrészben (például szobában, teremben) különböző objektumok pozíciójának meghatározása és típusának beazonosítása. Ilyen felhasználási terület például az intelligens otthon, amelyben például a „Villany le!” kimondása esetén az adott szoba vagy térrész lámpája kapcsolódik le.

Diplomamunkám során bemutatom, hogyan terveztem meg a jelfeldolgozás folyamatát, majd ahogyan a különböző lokalizációs és forrásazonosító algoritmusokat implementáltam. A lokalizációs algoritmusok idő- vagy frekvenciatartományban késleltetik meg a mikrofonokból beérkező jelet, majd ezeket összeadva teljesítményt számoltam. A teljesítmény maximuma adja meg a feltételezett forrás pozícióját. A forrásazonosítás két részre osztható. Elsőként az időfüggvényekből tulajdonságvektorokat generálunk, amelyek jól jellemzik a hangot. Ezután a tulajdonságvektorokat osztályozzuk, azaz meghatározzuk, hogy a milyen forráshoz tartozhatnak. A mérések elvégzése után kiértékelem az eredményeket.

Abstract

Some applications require the identification of the type and position of various objects in a part of the space (such as outdoors or in a hall). Such functionalities can be used in so called “smart homes”. One concrete example could be a voice controlled environment sensitive light switch.

This thesis proposes a plan for the flow of signal processing and an implementation for various localization and source identification algorithms. The localization algorithms delay the incoming signal from the microphone in time or frequency domain. Summing these signals performance is calculated. The maximum of performance provides the supposed source position. The source identification is divided into two parts. First, the representation of sound is generated from time functions, in the form of feature vectors. Then, the feature vectors are classified, i.e. source is determined. Upon completion of measurements, results are evaluated.

1 Bevezetés

Bizonyos alkalmazásokban szükséges lehet egy adott térrészben (például szobában, teremben) különböző objektumok pozíciójának meghatározása és típusának beazonosítása. Ilyen felhasználási terület például az intelligens otthon, amelyben például a „Villany le!” kimondása esetén az adott szoba vagy térrész lámpája kapcsolódik le.

Az intelligens otthon funkciói közé tartozhat a fűtés hangvezérelten történő ki- és bekapcsolása, a hifi, a televízió kapcsolása, a redőny leengedése. Ezek mind a lokalizáció felhasználásai, amelyeknél a lokalizált térrész eszközét kapcsoljuk be vagy ki. A lokalizáción túl lehetőség nyílik olyan funkciók megvalósítására is, amelyek a forrásazonosításon, azaz a hangforrás típusának meghatározásán alapulnak. Ilyenformán ha a televíziót felnőtt kapcsolja be, az a neki legkedvesebb csatornájára, ha pedig gyerek, akkor egy előre beállított mesecsatornára ugrik, így nem kell csatornát keresgetni, kapcsolgatni, távirányítót keresni. A taps felismerésével lehetőség nyílik arra, hogy egy tapssal fel-, kettővel lekapcsoljuk a lámpát az adott szobában. A lakáson kívül, a kertben is kényelmessé tehetjük a vezérlést, így például a kerti locsolót vagy az öntözőrendszert is kapcsolhatjuk hangvezérelten.

Az otthon kényelmén túl a lokalizáció és a forrásazonosítás az autóban is alkalmazható, például a rádió vagy a klíma hanggal történő beállításával. Többzónás klíma esetén zónánként lehet állítani a hőmérsékletet hely vagy hangszín alapján egyaránt.

Ezek az alkalmazások az energiatakarékosság jegyében szolgálják a praktikusságot és a kényelmet, ugyanakkor elláthatnak az időseket segítő otthoni felügyeleti vagy biztonsági funkciókat is.

További felhasználási lehetőség a multimédiás alkalmazások, amelyben hangvezérelten irányítunk például egy játékot, amelyben a karakterek attól függően mozognak, hogy melyik játékos beszélt. A hangvezérlés virtuális valóság rendszerekbe hatékonyan implementálható, amely gyors információbevitelt tesz lehetővé, anélkül, hogy lefoglalnánk egy-egy testrészünket.

Feladatomban egy olyan kísérleti rendszert hoztam létre, amely különböző, hangforrások pozíciójának akusztikus lokalizációjára alkalmas eljárások tesztelését is

elvégzi. Továbbá képes felismerni olyan jól elkülöníthető hangosztályokból származó hangokat, mint az emberi beszéd, állathang, különféle mesterségesen keltett hangok. A feladat megoldásához szükséges jelfeldolgozási eljárások implementálása PC-n történt, a feldolgozás nem valós idejű.

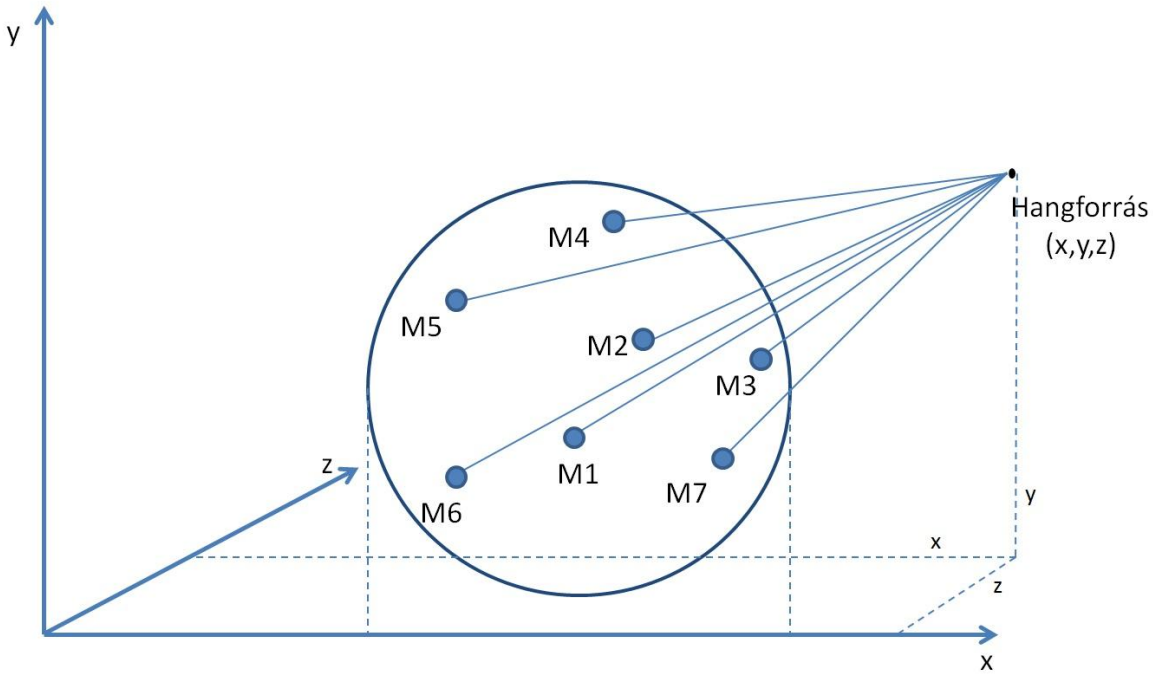
2 Rendszerterv

Diplomafeladatom két fő részre oszlik, az első része a lokalizáció, a másik a hangosztályozás.

A lokalizáció megvalósítható akusztikus jelek felhasználásával. Ha a megfigyelt objektum által kibocsátott hangot több mikrofonnal érzékeljük, a pozíció az akusztikus jelek feldolgozásával becsülhető. Az akusztikus jelek felhasználása mind a pozíciómeghatározásra mind a hangforrás típusának osztályozásra alkalmas.

Feladatomban egy olyan kísérleti rendszert hoztam létre, amelyben különböző, hangforrások pozíciójának akusztikus lokalizációjára alkalmas eljárások tesztelését is elvégzem. Alapvetően hangjeleket dolgozunk fel. A hangjelek változó hosszúságúak, de tipikusan néhány másodperc hosszúak, valamint tőlünk függetlenek (általunk nem befolyásoltak). A feladat megoldásához szükséges jelfeldolgozási eljárások implementálása PC-n történt Matlab környezetben.

Mivel mind a lokalizációhoz mind a hangforrás hangjának osztályozásához akusztikus jeleket használunk fel, a rendszer fizikai kiépítését tekintve azonos elrendezést használtam. Ennek szemléltetése a következő ábrán látható.



2.1 ábra Általános elrendezés

A hangjeleket az ábrán *M*-mel jelölt mikrofonok segítségével rögzítjük. A mikrofonok az ábrán látható módon egy síkban vannak. Ez a sík az *x-y* sík, amely a terem egyik falának feleltethető meg. A *z* irány a terem harmadik tengelyét reprezentálja. A hangforrás *x>0, y>0, z>0* koordinátákkal rendelkezik, tehát a mikrofonok síkjától távolabb helyezkedik el.

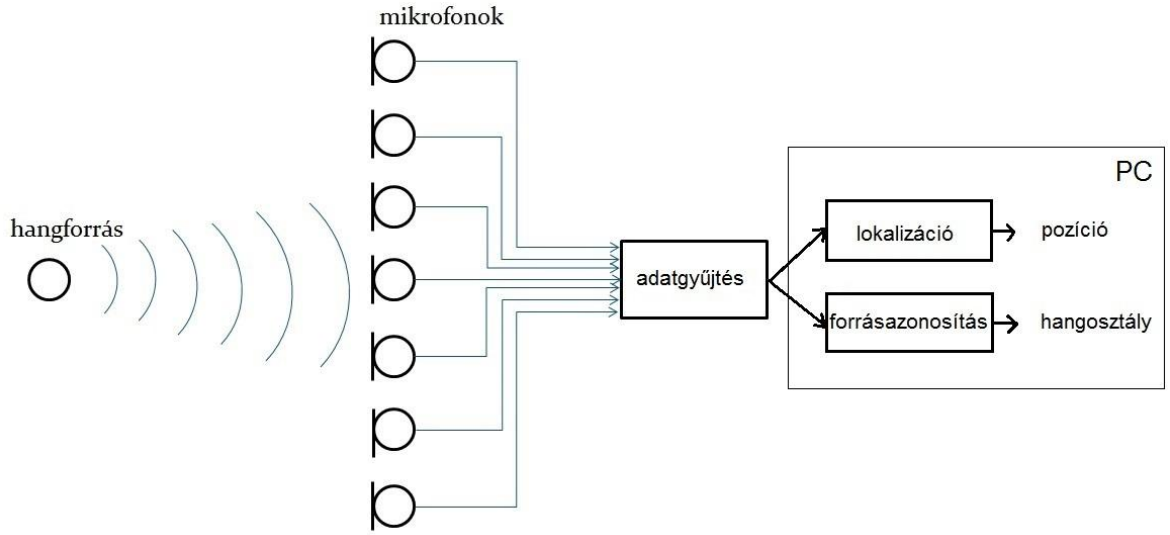
2.1 A rendszerrel szemben támasztott követelmények

A rendszer általános elvárásait tekintve több szempont merült fel, végül az alábbi pontokban összeszedett követelmények fogalmazódtak meg:

- akusztikus jelek felvétele
- több mikrofon jelének együttes (szinkron) rögzítése
- bonyolult jelfeldolgozó algoritmusok hatékony implementálása és futtatása
- lokalizáció 2 dimenzióban történő megvalósítása
- hangosztályok elkülönítése

A valósidejű működésre és a több forrásból egyszerre érkező hang feldolgozására nem volt követelmény.

A következő ábrán a feldolgozás jelfolyama látható. Az adatgyűjtés szinkron módon mintavételező mikrofonok segítségével történik, amelyek a hangforrásból jövő hangot rögzítik. Az adatgyűjtés ezután két részre ágazik, egyrészt a lokalizációra, másrészt a forrásazonosításra. A lokalizáció eredményeként pozíciót, a forrásazonosítás eredményeként pedig egy hangosztályt kapunk.

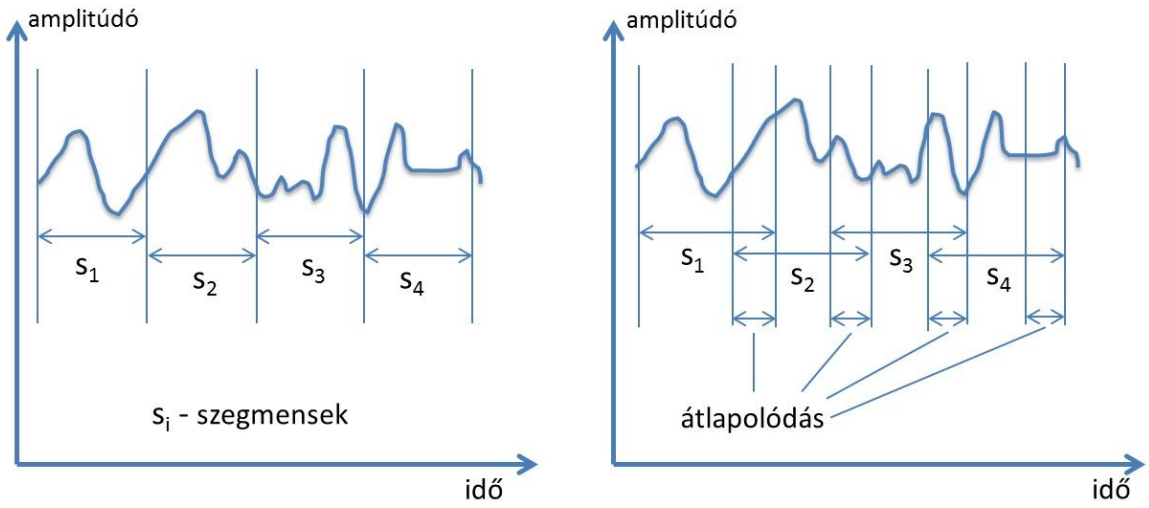


2.2 ábra A feldolgozás jelfolyama

2.2 Szoftverkörnyezet

2.2.1 Szegmentálás

Felvételeink túl hosszúak voltak ahhoz, hogy egyben dolgozzuk fel azokat, ráadásul azok hosszúsága is változó. Ezen túl a blokkméret az algoritmusok főbb paramétere, amelyet nem lehet akármekkorára változtatni. Ezért felvételeinket szegmensenként dolgoztuk fel. A szegmentálás a felvétel lineáris tagolása, vagyis az időfüggvények állandó méretű adatblokkokra bontása. Az állandó méret miatt számításaink jól paraméterezhetők. Erre a legjobb példa az FFT (Fast Fourier Transform), mely esetben a szegmens hossza meghatározza a felbontást. Az alábbi ábrán az átlapolódásmentes és átlapolódásos szegmentálás szemléltetése látható.

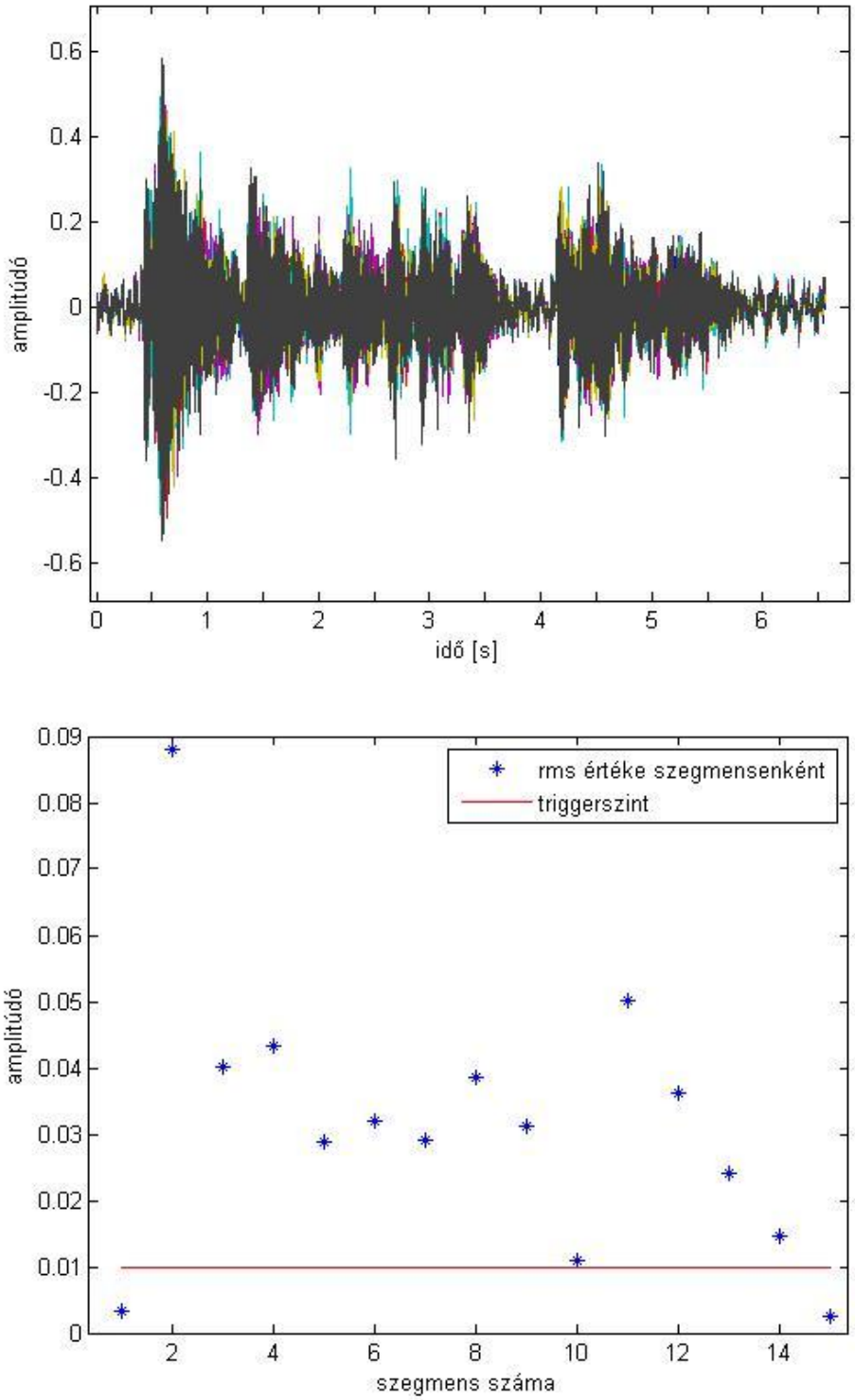


2.3 ábra Átlapolódásmentes és átlapolódásos szegmentálás

2.2.2 Triggerelés

A releváns adatok felhasználása érdekében csak azokat a szegmenseket szerettük volna megfigyelni, eredményeit nagyobb súllyal figyelembe venni, amelyeket a háttérzajból jól kiemelkedő hangerősségű hangforrás eredményezett. Ennek érdekében triggereltünk.

A 2.4 ábrán látható, hogy egy teljes időfüggvényhez milyen RMS érték és milyen triggerszint tartozik. Az ábrán látható, hogy a felvétel eleje és vége a triggerszint alá esik, mivel ott még a csendet vettük fel.



2.4 ábra Az időfüggvény, a hozzá tartozó szegmensenkénti RMS értékek és triggerszintek

3 Lokalizációs algoritmusok

3.1 Lokalizáció különböző jelek alapján

A lokalizáció, azaz a helymeghatározás manapság igen elterjedt kérdéskör. A lokalizáció fogalma jelen van többek között a gazdaságban, az orvostudományban, a tűz-és vízvédelemben, az informatikában. Az alábbiakban azokba a módszerekbe nyerhetünk betekintést a teljesség igénye nélkül, amelyek segítségével megvalósítható a lokalizáció.

3.1.1 Lokalizáció rádiós jelek alapján

Az elektromágneses hullámok modulációjával előállított rádiójelek segítségével történő lokalizáció egyik fontos felhasználása a rádiólokátor, azaz a radar. A radart a második világháborútól a meteorológiától kezdve a haditechnikán, a közlekedésen, térképészeten keresztül a navigációig rengeteg területen alkalmaznak. A radar lényege, hogy 3 MHz – 110 GHz frekvenciájú, azaz 100 m - 2,7 mm hullámhosszúságú rádióhullámokat bocsátanak ki, amelyek visszaverődése alapján meg lehet állapítani a visszaverődés helyét. Rádiós jelek alapján történik a rádió- és tv-adás, a mobiltelefon, a műholdas kommunikáció, a Wi-fi, a GPS és számos orvosi alkalmazás, például az MRI is. [1]

3.1.2 Lokalizáció ultrahang alapján

A 20 000 Hz-nél magasabb frekvenciájú hangokat ultrahangnak nevezzük. Emberek számára nem hallható, de az állatok közül sokan hallják, közismert, hogy a kutyák reagálnak rá. A denevérek és a delfinek maguk is állítanak elő ultrahangot a tájékozódásuk során. [2] Az ultrahangos lokalizáció fogalma továbbá jelen van többek között a kémiában, az ideggyógyászatban, belgyógyászatban, de általában az orvostudományban. Szilárd testekben megbúvó üregeket deríthetünk fel, vagy sérült belső szervek, szövetek helyét határozhatjuk meg. Ez adott esetben életfontosságú is lehet.

3.1.3 Lokalizáció optikai jelek alapján

A lokalizáció történhet optikai úton, azaz kamerákkal történő megfigyeléssel is. Ez megfelelően nagy felbontás mellett elég költséges megoldás, főleg ha elfogadható

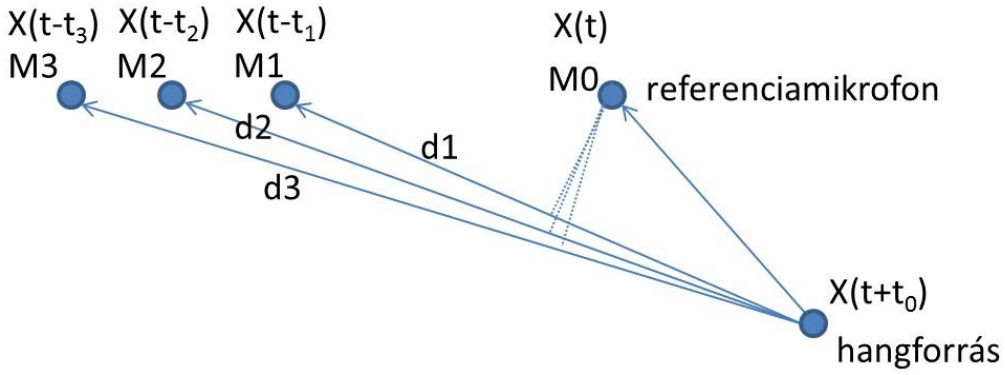
mértékű sebességgel szeretnénk megvalósítani. Ráadásul az optikai lokalizáció nagy hátránya a triggerelés nehézsége, azaz eldönteni, hogy mit kell megfigyelni. Emberi beszéd, állathangok, ajtóbecsapódás, stb. esetében kamerák segítségével nehéz lenne eldönteni, hogy mikor történt az esemény. Lehetne olyan kameraállás, amelyben a hangforrás takarásban van, ez további nehézségeket vet fel.

3.1.4 Lokalizáció hangjelek alapján

Az optikai elven történő lokalizációval szemben a hangjelek alapján történő lokalizáció hatalmas előnye, hogy a megfigyelni kívánt jelet maga az esemény generálja, ilyenformán nem kell külön beállítani a triggerfeltételeket. Az esemény a mikrofonnal felvett időfüggvény maximumához tartozó időpontnak megfelelő helyen történt, hosszabb időtartamú események esetében adott triggerszintet meghaladó hangjelet tekinthetjük az eseménynek. Az akusztikus jelek felhasználása lehetőséget ad továbbá az objektum által kibocsátott hang osztályozására is. Mindezen oknál fogva, mind a technikai megvalósítás tekintetében a hangjelek alapján történő lokalizációt választottam feladatom megvalósításához.

3.2 Az akusztikus lokalizáció alapelve

Feltételeztük, hogy a lokalizálandó hangforrás a tér egy pontjában van, a forrás által keltett hangot gömbhullámmal modellezzük. A forrás által keltett hangot több mikrofonnal is érzékeljük, és mivel a mikrofonok térben elosztva helyezkednek el, így a 3.1 ábrán látható módon az egyes mikrofonok és a hangforrás közötti távolság mikrofononként különböző.



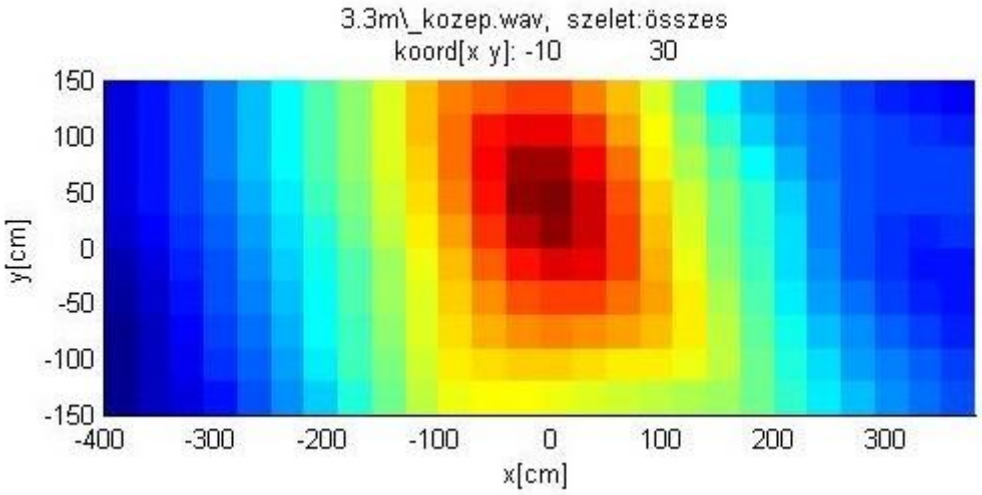
3.1 ábra Mikrofonok és késleltetéseik

Jelölje d_i az i-edik mikrofon és a hangforrás közötti távolságot! Mivel a d_i távolságok általában nem egyeznek meg, így a hangforrás által keltett hanghullámok a különböző mikrofonokba különböző késleltetéssel érnek be. Célszerű kijelölni egy úgynevezett referenciamikrofont, amelynek a jeléhez viszonyítjuk az összes többi mikrofon késleltetését. Mivel a hangforrás és az i-edik mikrofon, valamint a hangforrás és a referenciamikrofon közötti távolságkülönbség (d_i-d_0), így az i-edik mikrofonba a hang a referenciamikrofonhoz képest $t_i=(d_i-d_0)/c$ időkülönbséggel érkezik be, ahol c jelöli a hangsebességet. Ezt szemlélteti a 3.1 ábra. Az ábrán az $x(t-t_i)$ függvények pedig azt szimbolizálják, hogy az egyes mikrofonokban érzékelt időfüggvény milyen időbeli eltolással rendelkezik a referenciamikrofonhoz képest. A számítás megkönnyítése érdekében tehát a késleltetéseket a referenciamikrofonhoz viszonyítottuk, de meg kell jegyezni, hogy annak is van valamilyen t_0 késleltetése a valódi időfüggvényhez képest. Az akusztikus lokalizáció alapgondolata a következő:

1. tételezzük fel, hogy a hangforrás a tér egy adott (x,y,z) pontjában van,
2. számítsuk ki, hogy ebben az esetben az egyes mikrofonokhoz milyen késleltetéssel ér be a hang, (lásd 3.1 és 3.5 ábrák)
3. az egyes mikrofonok által érzékelt hangot ennek a késleltetésnek a mínusz egyszeresével toljuk el (kompenzáljuk a késleltetéseket),
4. számítsunk ki az eltolt időfüggvényekből egyfajta eredő jelintenzitást, például a jelek egyszerű összegzésével (ezen a ponton a különböző módszerek jelentősen eltérhetnek, lásd a későbbi fejezeteket)

A módszert a 3.4 ábra szemlélteti. A fenti módszerrel le kell tapogatni a tér összes lehetséges (x,y,z) pontját megfelelő felbontással, és így gyakorlatilag kiszámoljuk, hogy a tér egyes irányaiból mekkora intenzitású hang érkezik, tehát a tér egy adott pontjára fókuszálva mekkora teljesítményt érzékelünk. Feltételezzük, hogy a 3.2 ábrának megfelelően akkor lesz a legnagyobb a jelteljesítmény, amikor a térnek arra a pontjára fókuszálunk, ahol a hangforrás található, ezt tekinthetjük a forrás pozíciójának (ekkor ugyanis minden időfüggvényt pontosan akkora késleltetéssel kompenzáltunk, amekkora a valódi késleltetés volt, így a jeleket teljes mértékben fázisban adjuk össze). Amennyiben több hangforrás is létezik, akkor több lokális maximumot is érzékelhetünk, de ezzel az esettel most nem foglalkozunk. Ennek megfelelően viszont a lokalizációt jelentős mértékben befolyásolják a környezetből érkező egyéb hangok vagy például beltérben fellépő reflexiók.

Azt a pontot, ahonnan a hangforrást feltételezzük, fókuszpontnak nevezzük. A fókuszpontnak a mikrofonoktól vett távolsága a fókusztávolság, ez az a pont, ahova a mikrofonokkal „fókuszálunk”. Beltérben éppen ezért véges fókusztávolságot feltételezünk.



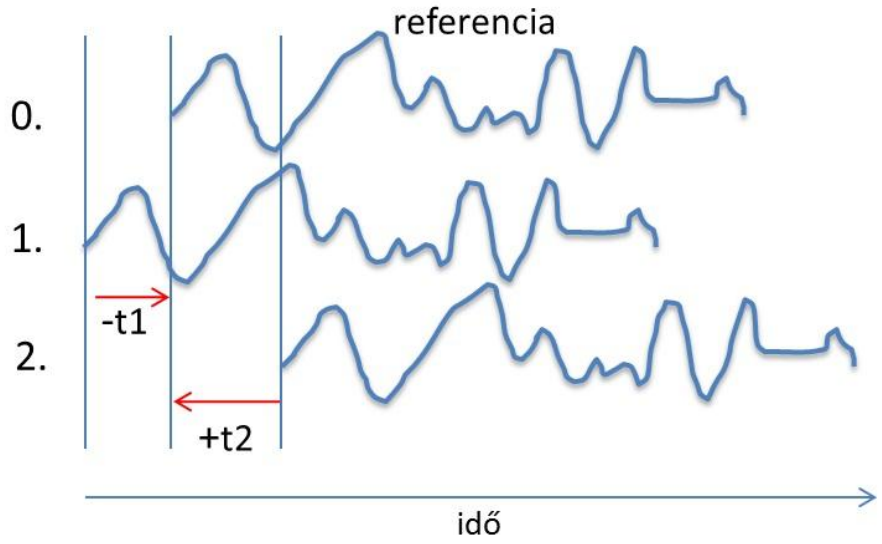
3.2 ábra A tér letapogatása

A tér letapogatását x és y koordináta mentén két dimenzióban végeztük el. A z koordinátát fixre, egy átlagos értékre állítottuk be.

Ahogy a 3.2 ábrán is látszik, a tér letapogatását 30 cm-es osztásokban végeztem el. Az ábrán a piros szín jelöli a legnagyobb teljesítményt, a kék a legkisebbet. Jelen esetben középről érkezett a legnagyobb teljesítmény, így valószínűleg ott található a forrás. A kép címében x és y koordinátaként azért szerepel a -10, 30, mert azok a pontok estek a 30 cm-es rácsosztás pontjaira.

3.3 A delay and sum módszer

A delay and sum (késleltet és összead) módszer esetében a késleltetést, tehát a minták csúsztatását időtartományban végeztük el. A jeleket csak egész mintákkal tudjuk megkésleltetni, ami nagyban leegyszerűsíti számításainkat. Ez a módszer nagy előnye, de egyben a korlátja is, ugyanis törtrész-késleltetést nehézkes megvalósítani. Szerencsére esetünkben megfelelő volt az egész mintával való késleltetés.



3.3 ábra A delay and sum módszer késleltetése

A megkésleltetett jelek összegéből teljesítményt számolunk. A tér x és y koordinátáinak letapogatásával minden pozícióra kiszámoljuk az adott irányból érkező teljesítményt, majd ennek keressük a maximumát, amely megadja a hangforrás számolt irányát. A módszer alapvetően a visszakésleltetett időfüggvények összeadására vonatkozik, ám mi kipróbáltuk azt is, hogy a jeleket összeadás helyett összeszorozzuk. Ez tulajdonképpen a korrelációs számítás általánosítása, két jel esetén a korrelációt kapnánk vissza. A későbbiekben ezért szerepel mindig a delay and sum módszeren belül két eredmény, összeadásra és szorzásra.

$$X_1(t) = x_1(t) \qquad X_1(t) = x_1(t)$$

$$T(x,y) = \frac{d_2(x,y) - d_1(x,y)}{c} \qquad T(x,y) = \frac{d_2(x,y) - d_1(x,y)}{c}$$

$$X_2(t,x,y) = x_2(t + T(x,y)) \qquad X_2(t,x,y) = x_2(t + T(x,y))$$

$$P(t,x,y) = X_1(t) + X_2(t,x,y) \qquad P(t,x,y) = X_1(t) * X_2(t,x,y)$$

$$P_{dir}(x,y) = \sqrt{\sum_t P^2(t,x,y)} \qquad P_{dir}(x,y) = \sqrt{\sum_t P^2(t,x,y)}$$

$x_1(t)$ - az első jel időfüggvénye

$x_2(t)$ - a második jel időfüggvénye

$d_1(x,y)$ – az 1. mikrofon és a hangforrás közötti távolság

$d_2(x,y)$ – az 2. mikrofon és a hangforrás közötti távolság

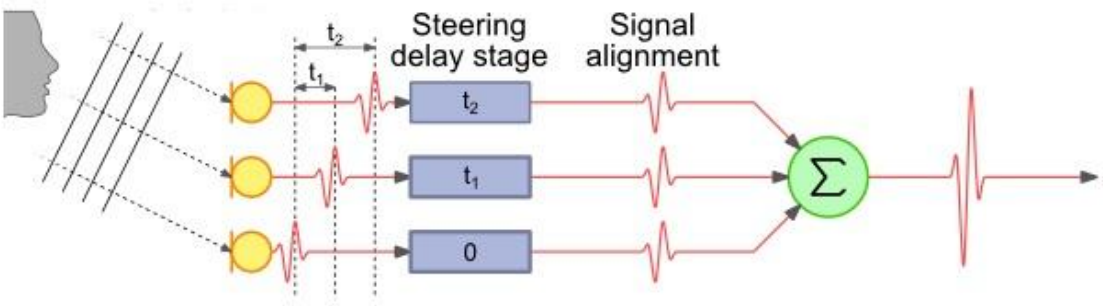
$T(x,y)$ - késleltetés

c - hangsebesség

$X_1(t)$ - az első jel időfüggvénye

$X_2(t, x, y)$ - a második jel időfüggvénye időtartományban visszakésleltetve

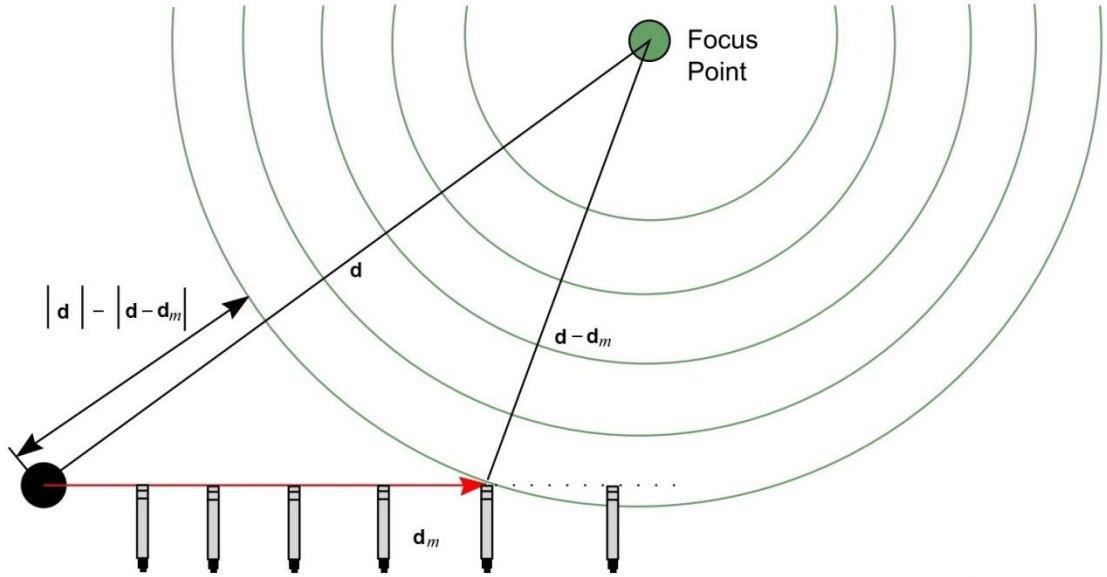
$P_{dir}(x, y)$ – az (x,y) koordináta irányából érkező eredő teljesítmény



3.4 ábra A késleltetések kompenzálása és a jelek összeadása [11]

3.4 A beamforming módszer

A beamforming (nyalábformálás) módszerének két fajtája van, a véges és a végtelen fókusztávolságú beamforming. A végtelen fókusztávolságú akkor alkalmazható, ha a mikrofontömb a hangforrás távolterében helyezkedik el. Ekkor a hanghullámok már síkhullámoknak tekinthetők, míg közeltérben (véges fókusztávolságú nyalábformálás esetén) gömbhullámoknak. Esetünkben – beltérről lévén szó– a véges fókusztávolságú nyalábformálást alkalmaztuk. A véges fókusztávolságú nyalábformálást a 3.5 ábra szemlélteti.



3.5 ábra A gömbhullámok a véges fókusztávolságú beamforming esetében [8]

3.4.1 Normál beamforming

A beamforming módszer lényege, hogy a jelek késleltetését frekvenciatartományban végezzük el, majd a megkésleltetett jelek összegéből a delay and sum módszerhez hasonló módon teljesítményt számolunk. Az alábbi képletek 2 jelre mutatják be sematikusan a normál beamforming módszer lényegét.

$$X_1[k] = fft(x_1(t))$$
$$X_2[k] = fft(x_2(t)) * e^{-i2\pi f_k T(x,y)}$$
$$P(f,x,y) = X_1[k] + X_2[k](x,y)$$

$$P_{dir}(x,y) = \sqrt{\sum_f P^2(f,x,y)}$$

$x_1(t)$ - az első jel időfüggvénye

$x_2(t)$ - a második jel időfüggvénye

X_1 - az első jel spektruma

X_2 - a második jel spektruma frekvenciatartományban visszakésleltetve

$P_{dir}(x,y)$ - az (x,y) koordináta irányából érkező eredő teljesítmény

Az algoritmus eredményeképpen egy olyan mátrixot kapunk, amelyben az adott irányból érkező teljesítményeket minden frekvenciára kiszámoltuk. A tér letapogatásával minden pozícióra kiszámoljuk az adott irányból érkező teljesítményt, majd ennek keressük a maximumát, amely megadja a hangforrás számolt irányát.

A beamforming módszer előnye, hogy a késleltetés tetszőleges finomságú lehet. A módszer további előnye, hogy frekvenciatartományban más módszerek is léteznek a módszer további javítására, fejlesztésére (pl MUSIC beamforming, Capon beamforming) [4], [9].

3.4.2 Capon beamforming

A normál beamforming algoritmus hátránya, hogy eredménye erőteljesen függ a jel-zaj viszonytól (SNR – Sign to Noise Ratio). Ezt kívánja kiküszöbölni a Capon beamforming, másik nevén a minimális varianciájú torzításmentes válasz módszere (Minimum Variance Distortionless Response) [15]. A módszer az autokorrelációs mátrixot használja fel. Eredményeképpen olyan a mikrofonok súlyozása, hogy a vizsgált irányban egységnyi az erősítés. A Capon beamforming minimalizálja a zajteljesítményt. A módszernek nagyobb a számításigénye, mint a hagyományos beamforming algoritmusnak.

$$R_{est} = \frac{x_t * {x_t}^T}{l}$$

$$a_v[k] = e^{-i2\pi f_k T(x,y)}$$

$$steer = \frac{{R_{est}}^{-1} * a_v}{{a_v}^T * {R_{est}}^{-1} * a_v}$$

$$X_1[k] = fft(x_1(t))$$
$$X_2[k] = fft(x_2(t)) * steer$$
$$P(f,x,y) = X_1[k] + X_2[k](x,y)$$

$$P_{dir}(x,y) = \sqrt{\sum_f P^2(f,x,y)}$$

x_t – a szegmens időfüggvénye

l – az időfüggvény(x_t) hossza

a_v - késleltetés

$x_1(t)$ - az első jel időfüggvénye

$x_2(t)$ - a második jel időfüggvénye

X_1 - az első jel spektruma

X_2 - a második jel spektruma frekvenciatartományban visszakésleltetve

$P_{dir}(x, y)$ - az (x,y) koordináta irányából érkező eredő teljesítmény

3.4.3 Normalizált beamforming

A normál beamforming algoritmus megvalósítása során a visszakompenzált vektorokat egyszerűen összeadjuk. Ez adott esetben torzításhoz vezethet, ugyanis ha néhány komponens domináns, akkor azok erőteljesen meghatározzák az eredő teljesítményt. Ez a normál beamforming algoritmus hátránya, amit viszont könnyen tudunk orvosolni a komponensek normálásával [16]. A háromdimenziós teljesítménymátrix minden egyes frekvenciakomponensét egységnyire normalizáljuk, ezután vesszük a teljesítmények négyzetösszegeinek gyökét:

$$P_{norm}(f, x, y) = \frac{P(f, x, y)}{\max_{x,y}\left(P(f, x, y)\right)}$$

$$P_{dir\_norm}(x, y) = \sqrt{\sum_{f} {P_{norm}}^2(f, x, y)}$$

$P_{norm}(f, x, y)$ – frekvencia szerint normált teljesítmény

$P(f, x, y)$ – háromdimenziós teljesítménymátrix ((x,y) párok minden frekvenciára)

f – frekvenciatengely

$P_{dir\_norm}(x, y)$ – normalizált teljesítménymátrix

A normalizált beamforming algoritmusban tehát minden egyes frekvenciakomponens azonos súllyal szerepel, amitől pontosabb eredményt várunk.

3.5 Eredő pozíció számítása a teljes hangeseményre

A lokalizációt szegmensenként végeztük el, így eredményeit is szegmensenként kaptuk meg. Ám ahhoz, hogy egy hangeseményre (fájlra) egy eredmény adódjon, a kapott síkbeli pontokból egy eredő koordinátát kell kapnunk. Ezt több módon tehetjük meg, amelyeket a következő alfejezetekben ismertetek. Mindegyik esetben a triggerelt szegmenseket vettem figyelembe, mivel azt tekintettem releváns információnak.

3.5.1 Eredő pozíció számolása átlagolással

Az átlagolás esetében az eredő pont koordinátáit a szegmensek koordinátáinak egyszerű átlagolásával kaptam meg.

$$P(x,y) = \left(\frac{1}{k} * \sum_{k} koordTrig[k]_x , \frac{1}{k} * \sum_{k} koordTrig[k]_y\right)$$

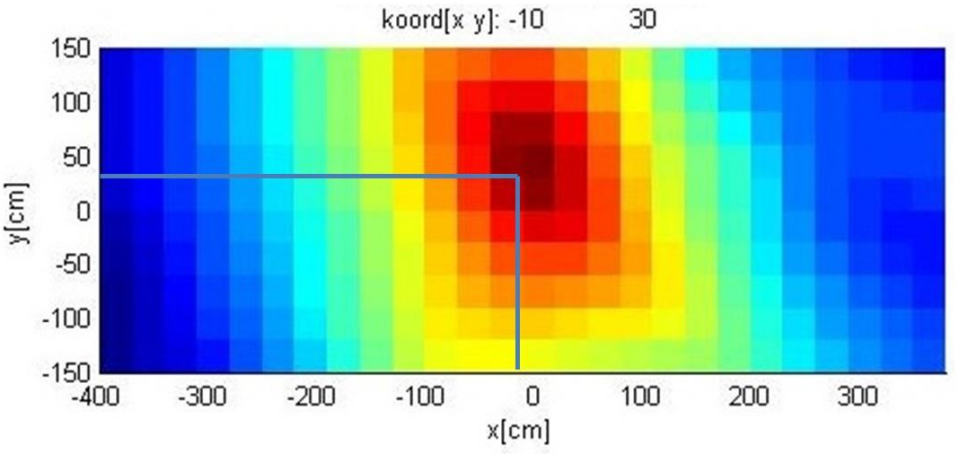
$koordTrig[k]_x$- a *k*-adik triggerelt x koordináta

$koordTrig[k]_y$- a *k*-adik triggerelt y koordináta

$P(x,y)$ – a számolt eredő pont

3.5.2 Eredő pozíció számolása átlagos teljesítményből

Az átlagos teljesítményből számolt pozíció esetében összeadtam a szegmensek síkbeli teljesítményeloszlását, majd megkerestem ennek a kétdimenziós felületnek a maximumát. A 6.2 ábrán az átlagos teljesítmény síkbeli eloszlása látható, amelyen a maximumhely x és y koordinátája le van vetítve a tengelyekre.



3.6 ábra A tér letapogatása

3.5.3 Eredő pozíció számolása a legtöbb szomszéddal rendelkező koordinátából

A legtöbb szomszéddal rendelkező pont kiválasztása esetében végigmentem a triggerelt koordinátapárokon, és kiszámoltam, hogy az egyes pontoknak hány szomszédja van 1.5 m-es sugarú körön belül. [5] Ezeknek a maximumát vettem, és megnéztem, hogy a maximum melyik ponthoz tartozik, azaz kiválasztottam azt a pontot, amelyiknek a legtöbb szomszédja van.

3.5.4 Az eredő pozíció hibája

Kiszámoltam a módszerek hibáit, azaz, hogy az eredő koordinátapár mennyivel tér el a hangforrás valós pozíciójától. A hibaszámítást a következő módon végeztem:

$$h = \sqrt{(realKoord_x - koord_x)^2 + (realKoord_y - koord_y)^2}$$

$koord_x$- az eredő pont x koordinátája

$koord_y$- az eredő pont y koordinátája

$realKoord_x$- a várt x koordináta, tehát a valós pozíció x koordinátája

$realKoord_y$- a várt x koordináta, tehát a valós pozíció y koordinátája

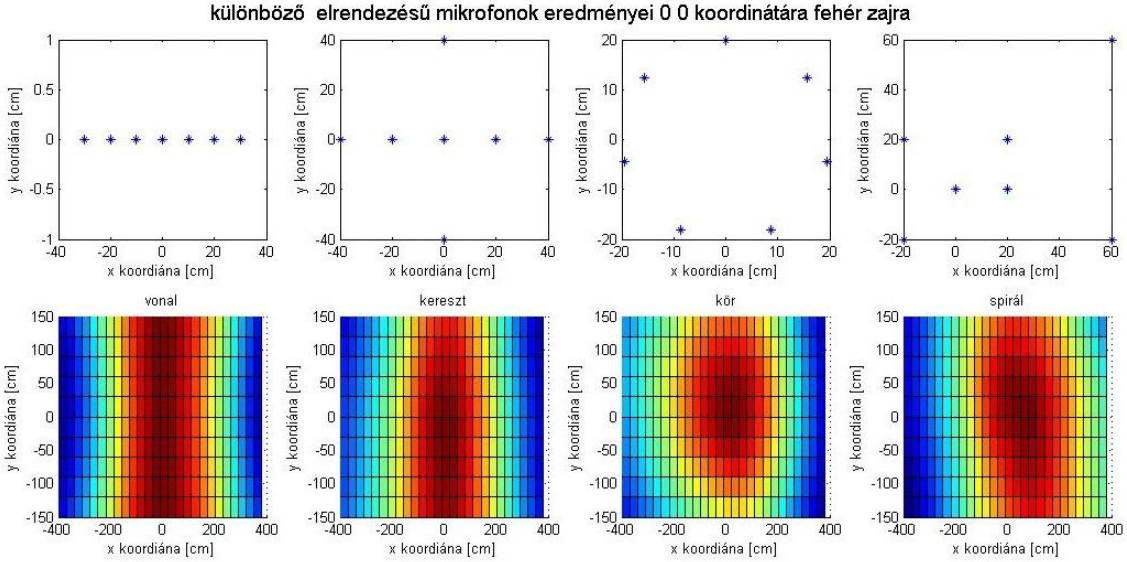
h - hiba

3.6 A mikrofonok elrendezése

A rendszer fizikai kialakítása során elsősorban a mikrofonok elrendezését kellett megterveznem. A rendelkezésre álló eszközök hét csatorna felvételét tették lehetővé, ezt a pontosság érdekében maximálisan kihasználtam.

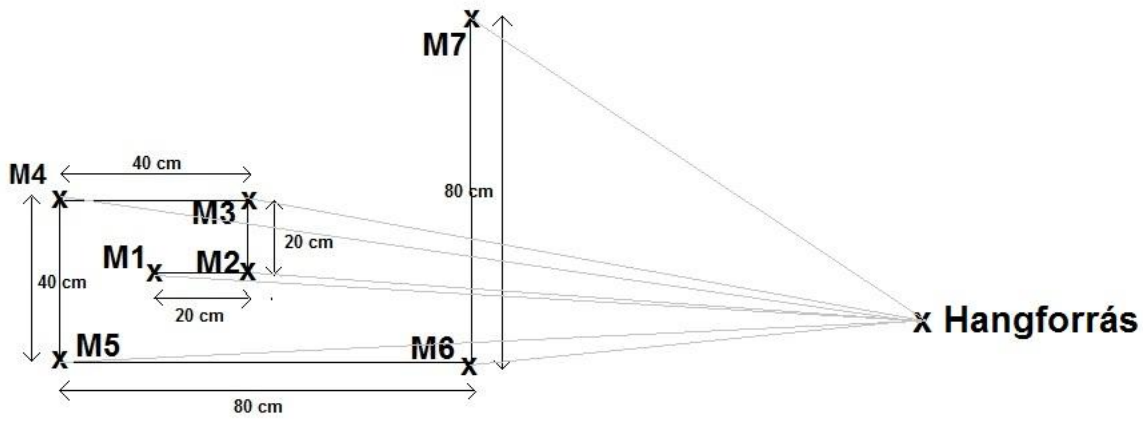
A hét mikrofon elhelyezését részben a szakirodalom által nyújtott információk alapján, részben saját szimulációs eredmények alapján spirális alakban rendeztük el [10]. A szimulációt annak kiderítése végett végeztem, hogy melyik elrendezés a legjobb. A szimuláció során a hét mikrofont négy különböző alakzatba rendeztem (vonalba, keresztbe, körbe és spirálba), majd szimulált körülmények között fehér zajra néztem meg, hogy milyen pontosan közelítik meg a kívánt (0,0) koordinátát. A

szimulációt úgy végeztem el, hogy a hangforrást a (0,0) koordinátába helyeztem el, és széles sávú gerjesztőjelre kiszámoltam a mikrofonpozícióknak megfelelő késleltetéseket. Ezekből a tér minden egyes koordinátájára kiszámoltam a teljesítményeket, amelyeket mérési eredménynek tekintettem. Az eredmények a következő ábrán láthatók:



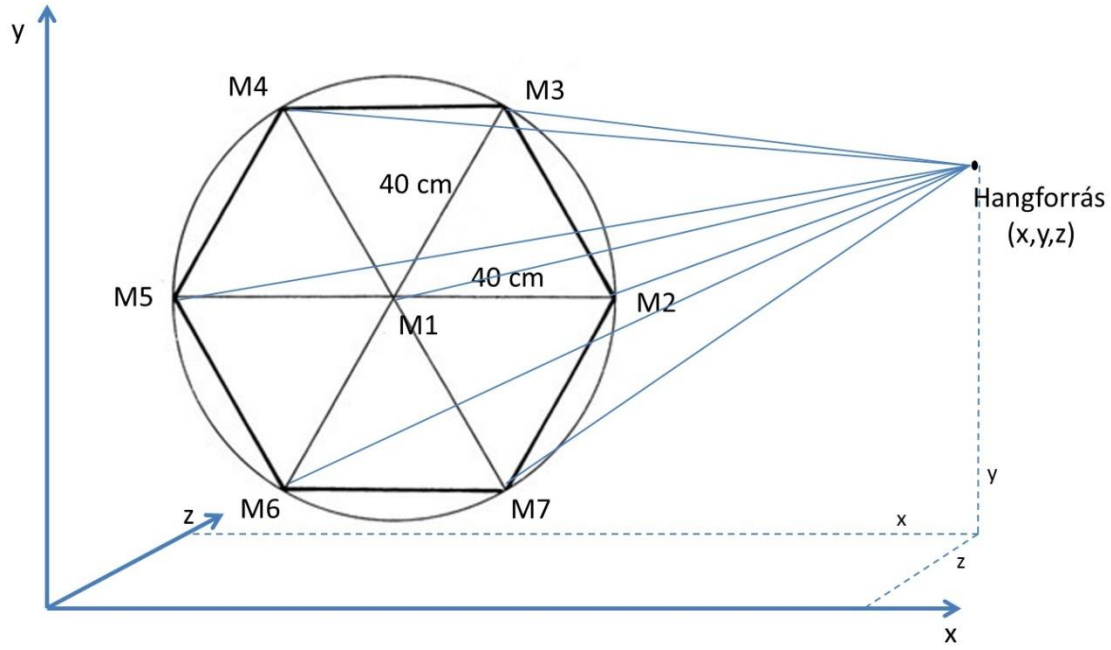
3.7 ábra Mikrofonelrendezések szimulációi

Az ábrán négy mikrofonelrendezés és mindegyik alatt a hozzá tartozó szimulációs eredménye látható. A tér minden egyes koordinátájára kiszámolt teljesítmények láthatók a megfelelő színnel, a vörös szín felel meg a nagyobb teljesítménynek, míg a kék a kisebbnek. Megfigyelhető, hogy a vonal és kereszt elrendezésben a második (magasság szerinti) koordinátát nagyon rosszul becsli. Ezért ezeket elvetettük. A másik két elrendezés közül első ránézésre a kör struktúra tűnik jobbnak, ám a spirál talán szűkebb tartományt fed le az x koordináta mentén, amely azért relevánsabb, mert egy terem vagy szoba általában szélesebb mint magasabb. A fentiek valamint a szakirodalmi javaslatokból kifolyólag a spirál mikrofonelrendezést választottam elsődlegesnek, de a kör alakú elrendezésben is mértem. A 3.8 ábrán a spirál alakú elrendezés látható sematikusan.



3.8 ábra Spirális mikrofonelrendezés

A mikrofonok kör elrendezését mutatja térben a következő ábra. A mikrofonok mindegyik esetben egy síkban voltak, ez az ábrán az x-y síknak felel meg.



3.9 ábra A kör mikrofonelrendezés

4 Hangfelismerő algoritmusok

4.1 Az osztályozás alkalmazásai

A beszélő hangjának azonosítása már nem csak a filmekben létezik, az igazságszolgáltatás és a kriminalisztika területén a mindennapokban is használjuk. Sok cég beszélőazonosító beléptetőrendszert használ. Ebben az esetben a rendszer egy meglévő mintával hasonlítja össze a belépő személy hangját. Ez a meglévő hangminta egy előre felvett jelszó. A két hang bizonyos fokú egyezősége esetén a rendszer elfogadja és engedélyezi a belépést.

A kriminalisztika területén felmerülő hangazonosítási eljárás sokkal összetettebb és bonyolultabb feladat. Egy rendelkezésre álló hangfelvétel, amin az azonosítandó személy hangja hallható, a legtöbb esetben amatőr minőségben és rossz körülmények között készült. Így a hangszakértőnek az elsődleges feladata a zajtalanítás és a beszédérthetőség javítása. A hang azonosításához szükség van egy összehasonlító felvételre, hangmintára, amivel az eredeti hangfelvételt egybe lehet vetni. A felvételek hosszúsága meghatározza vizsgálat eredményességét. Minél több hanganyag áll rendelkezésre, annál több ponton vizsgálható az egyezőség, vagy különbség és annál hatékonyabb a beszélőazonosítás [17].

Diplomamunkámban nem szándékoztam sem kriminalisztikai bonyolultságokba sem igazságszolgáltatásbeli mélységekbe merülni, inkább valóságos példákkal szerettem volna illusztrálni, hogy a témakör mennyire népszerű manapság és hogy alkalmazhatósága milyen sokrétű lehet. Feladatomban jól elkülöníthető hangosztályokból származó hangokat osztályoztam. Ehhez szintúgy szükség volt felismerendő mintákra, ezek voltak az úgynevezett tanító minták, és arra a mintasorozatra, amelyen elvégeztük az osztályozást. A minták számát tekintve általános szabályként fogalmazhatjuk meg azt, hogy minél több mintánk van, annál jobb eredményre számíthatunk az osztályozás során.

4.2 Az osztályozás alapelve

A hangfelismerés során felvételeinket tanító mintákra és felismerendő mintákra válogatjuk szét. A módszer célja, hogy az felismerendő mintákat a megfelelő tanítóknak ismerje fel.

A hangfelismerő algoritmus két fő részre oszlik:

1. Az idősorokból olyan tulajdonságok kinyerése, amelyek jól jellemzik a hangot –tulajdonságvektorok generálása (feature vektorok)
2. Meghatározni, hogy a tulajdonságvektorok milyen forráshoz tartozhatnak (osztályozás). A tulajdonságvektorokat a szegmensekből számoljuk ki. Ezekhez a tulajdonságvektorokhoz szeretnénk társítani egy osztályt. Az osztályozáshoz előzetesen össze kell állítani egy adatbázist, amely ismert forrásokhoz tartozó feature vektorokat tartalmaz. Az új vektort az adatbázisban szereplő hangosztályhoz hasonlítjuk. Amelyik osztályhoz valamilyen hasonlósági norma alapján a legközelebb áll, ahhoz soroljuk.

4.3 Feature vektorok generálása

A feature vektorok generálása során a frekvenciatartománybeli feature vektorokra koncentráltunk. Ez viszonylag egyszerű, ugyanakkor jól működő módszer. Az alábbiakban a feature vektorok generálásának két módját mutatom be.

4.3.1 Feature vektorok számítása FFT -vel

A feature vektorok generálásának legegyszerűbb módja az FFT, azaz a gyors Fourier-transzformáció (Fast Fourier Transform). Ez esetben a feature vektor az időfüggvény egyszerű Fourier-transzformáltja.

$$Feature_i = fft(vekt_i)$$

$vekt_i$ – i-edik időfüggvény

$Feature_i$ – a Fourier-transzformált tulajdonságvektor (spektrum)

Az algoritmus egyetlen paramétere a blokkméret, azaz hogy az időfüggvényt mekkora szegmensekre daraboljuk fel. A blokkméret egyértelműen meghatározza az FFT felbontását., amely a következőképpen számolható:

$$felbontás = \frac{Fs}{blokkméret}$$

Az összefüggésben Fs a mintavételi frekvencia. Akusztikus jelek esetében a megfelelő felbontás általában néhányszor 10 Hz köré tehető. Ehhez a fenti képlet alapján $Fs = 48000$ Hz mintavételi frekvencia mellett néhány ezer pontos FFT-t kell végezni.

4.3.2 Feature vektorok számítása Mel spektrummal

A feature vektorok FFT-vel történő generálásának legnagyobb hátránya az, hogy nincs arányban a természetben előforduló frekvenciaskálákkal. Ugyanis az FFT egyenletes leképezést ad a frekvenciatengely mentén, ám a természetben előforduló hangok spektruma nem lineáris, nem egyenletesen tartalmaz információt. A hangot alacsonyabb frekvencián nagyobb felbontással érdemes feldolgozni, mivel ott több információt hordoz. Ehhez az idők során az emberi fül is idomult. A fülünkben lévő csigában szőrsejtek ismerik fel az adott frekvenciát, a szőrsejtek helye pedig logaritmikus mértékű [12]. Gondoljuk el, hogy az 50 Hz és 100 Hz közötti különbség mennyivel számottevőbb, mint például a 10.000 Hz és 10.050 Hz közötti, amelynél talán meg sem halljuk a különbséget. A feature vektorok FFT-vel történő generálása esetén tehát feleslegesen sok mintát számolunk ki nagy frekvencián, ami zajként viselkedhet, és elnyomhatja az alacsonyabb frekvenciás komponenseket, amelyek a legtöbb esetben hasznos információt hordoznak. Ezért az FFT alternatívájaként implementáltam azt a módszert, amelyben a feature vektorokat az úgynevezett Mel spektrum segítségével állítottam elő.

A Mel spektrum létrehozásához elsőként exponenciálisan növekvő sávközépi frekvenciákat generáltunk, ezek lettek az úgynevezett Mel szűrőbank sávközépi frekvenciái. A sávközépi frekvenciákat az alábbi módon számítottuk ki [3].

$$fM_{low} = 1125 * log(1 + f_{low}/700)$$

$$fM_{up} = 1125 * log\left(1 + f_{up}/700\right)$$

$$fM[i] = fM_{low} + i * \frac{fM_{up} - fM_{low}}{n}$$

$$f_{savkoz}[i] = 700 * (e^{\frac{fM[i]}{1125}} - 1)$$

f_{low} – a szűrőbank alsó frekvenciája

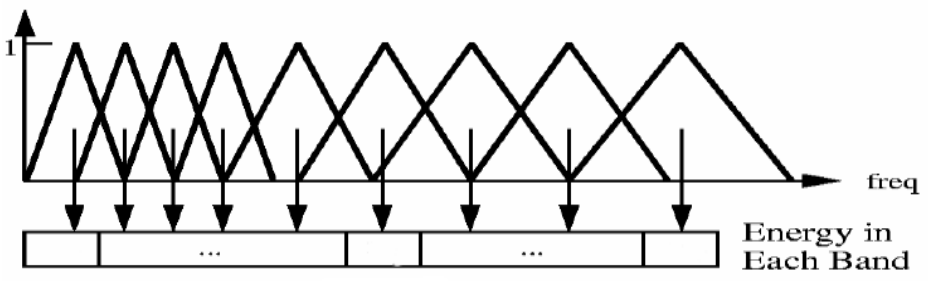
f_{up} – a szűrőbank felső frekvenciája

n – a sávközépi frekvenciák száma

$f_{savkoz}[i]$ – sávközépi frekvenciavektor i-edik eleme

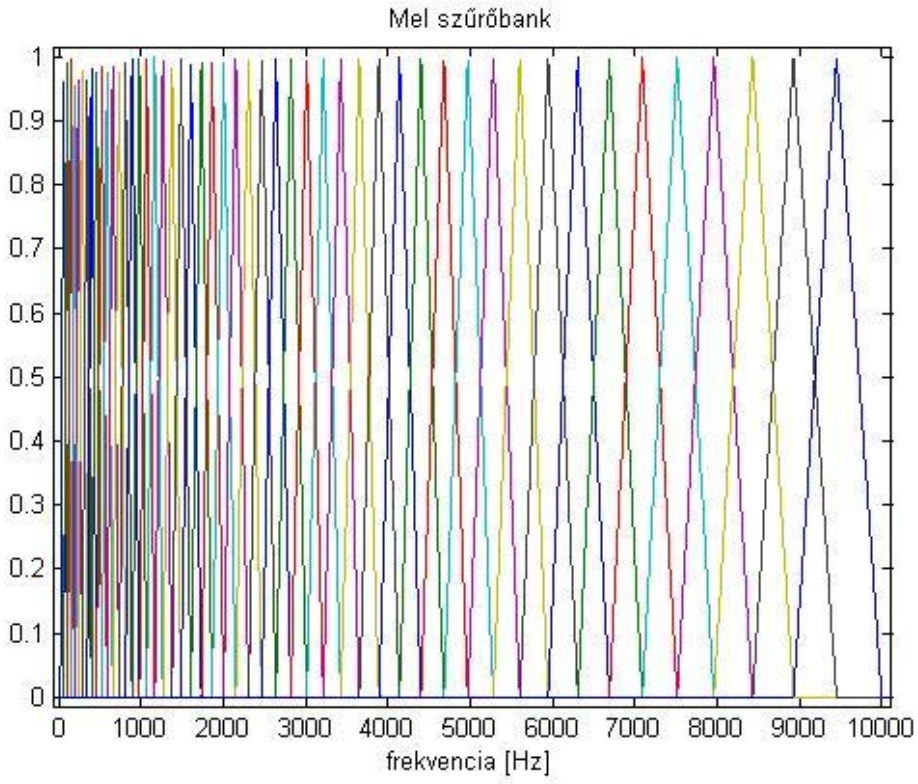
A Mel spektrum a frekvenciaskálát logaritmikusan transzformálja. Azt azután egyenletes részekre osztja fel, így végeredményben logaritmikus felbontást alakít ki.

A Mel szűrőbankunk háromszögablakokból áll, ezekkel az ablakokkal szorozzuk meg az FFT-vel számolt spektrumokat. Az összeszorzás ilyenformán 0 és 1 között mindig arányosan súlyozza a spektrumot. Ez a szorzat lesz a feature vektor. Az alábbi képen a Mel szűrőbank sematikus ábrája látható.



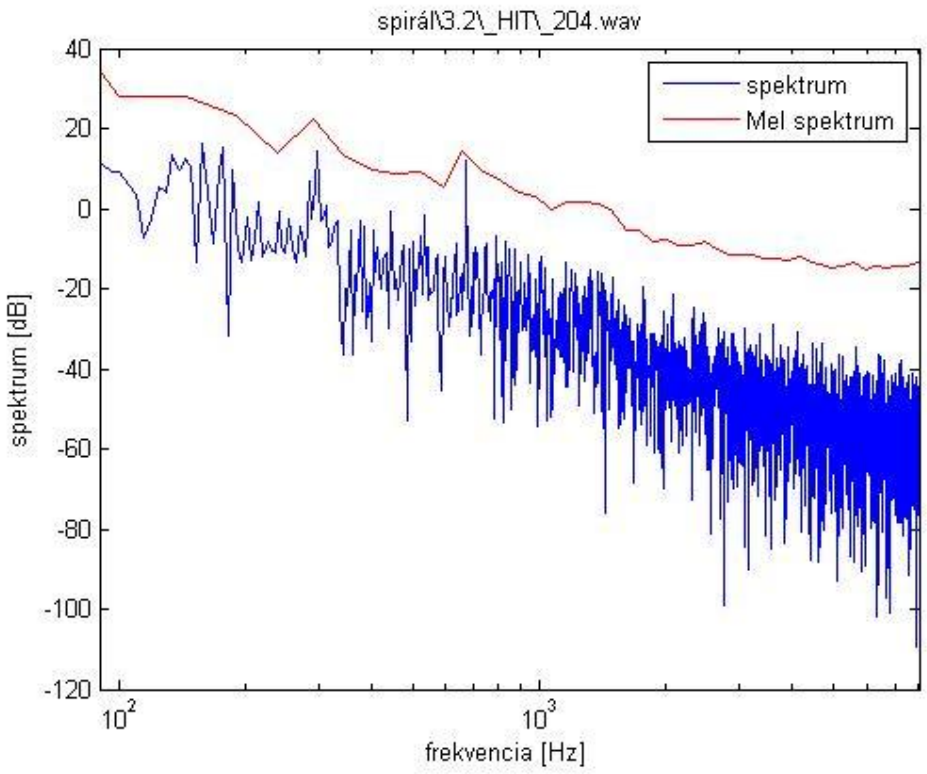
4.1 ábra Mel szűrőbank sematikusan [13]

A szűrőbank Matlabban történő megvalósítását a 4.2 ábra illusztrálja. A sávközépi frekvenciák számát, illetve alsó és felső frekvenciáit mi választottuk meg. A sávközépi frekvenciák száma 2-vel nagyobb a szűrőbank háromszögablakainak számánál, mivel az alsó és felső frekvenciához nem tartozik ablak. A következő ábrán a Matlabban generált Mel szűrőbank látható.



4.2 ábra A Matlabban generált Mel szűrőbank

A 4.3 ábrán a két fajta spektrum összehasonlítása látható. Az ábrán jól látszik, hogy a Mel spetrum az FFT-vel számolt spektrum burkolójának tekinthető. Továbbá feltűnő, hogy az FFT-vel számolt spektrum esetén mennyivel több számítást végzünk feleslegesen a nagyobb frekvenciákra, a spektrum ezért látszik ilyen sűrűnek. Ezzel szemben a Mel spektrum illeszkedik a logaritmikus skálához.



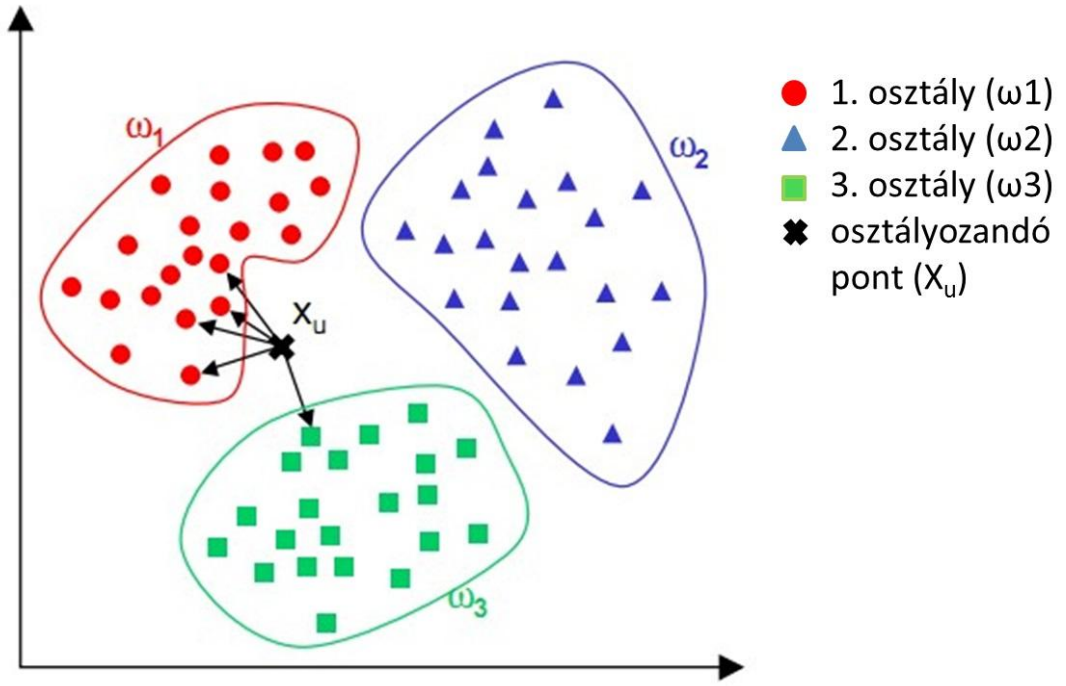
4.3 ábra FFT és Mel spektrum

4.4 Osztályozás

4.4.1 Az osztályozó algoritmus ismertetése

A generált feature vektorok lesznek a hangazonosító algoritmus tanító mintái, amelyek alapján az algoritmus osztályozza a felismerendő mintákat. Az osztályozás célja meghatározni, hogy a tulajdonságvektorok milyen forráshoz tartozhatnak. Ez az osztályozás. Ha az egyik hang valamilyen hasonlóság alapján a legközelebb áll egy hangosztályhoz, ahhoz soroljuk. A módszert a jelfeldolgozás szempontjából közelítettük meg, és az egyszerűnek számító *k* legközelebbi szomszéd analízisének vetettük alá (*k* nearest neighbour). Az osztályozást többek között neurális hálózattal, vagy support vector maschine módszerrel is meg lehetett volna valósítani [14].

A *k* legközelebbi szomszéd modell alkalmazásával feltételezzük, hogy egy adott térrész tulajdonságai közel azonosak lesznek a szomszédainak tulajdonságaival, majd a *k* legközelebbi elem megvizsgálásával és többségi döntéssel döntünk az ismeretlen elem tulajdonságáról (ahol *k* természetes szám) [6]. Az analízis eredményeképpen egy osztályt és a hozzá tartozó valószínűséget kaptunk. A döntéshozás mechanizmusát a 4.4 ábra szemlélteti.



4.4 ábra A *k* legközelebbi szomszéd algoritmus [7]

Az ábrán három különböző hangosztályt láthatunk. X_u egy olyan új minta, amelyre meg szeretnénk határozni, hogy az melyik ismert osztályba tartozik. Feltételezzük, hogy a minta a hangosztályok valamelyikébe tartozik, hiszen pont a mérések eredményeképpen generáltuk le a tulajdonságvektorainkat, amelyek az osztályokat alkotják. Megvizsgáljuk az X_u-hoz legközelebb eső *k* mintát, jelen esetben az 5 legközelebbit. Láthatjuk, hogy ebből 4 tartozik az 1. osztályhoz, 1 pedig a 3. osztályhoz. Ezek után döntést kell hoznunk, hogy melyik osztályhoz soroljuk X_u-t, amelyet az egyszerű többség módszerével könnyű megtenni. A megoldás a legnagyobb összeghez tartozó osztály, jelen esetben a 4-hez tartozó 1. osztály, és az ahhoz tartozó valószínűség (4/5=80%) lesz.

4.4.2 Az osztályozás eredményeinek kiértékelési módszere

Az analízis eredményeképpen kapott információk közül számunkra csak az osztály releváns, azaz hogy a mintát melyik hangosztálynak ismerte fel. Az eredményként kapott hangosztályokat összeszámoltuk egy úgynevezett confusion mátrixban (felosztás mátrixa), amelyben összesítettük, hogy az egyik csoportból származó tanító minták alapján hányszor ismerte fel megfelelően az adott mintát, és hányszor más hangcsoportnak. Az öt felismerendő minta a beszéd, a furulya, a kürt, a kutyaugatás, és a széktolás hangja volt. Mivel 5 hangosztály van, ezért a mátrix 5x5-ös.

Ennek megfelelően a mátrixban a diagonális helyén várjuk a nagyobb értékeket, amely azt jelenti, hogy a beszédet legtöbbször beszédnek, a furulya hangját furulyának, a kürtöt kürtnek, a kutyát kutyának, valamint a széktolás hangját széknek ismeri fel. A második táblázatban ugyanezek az értékek százalékos arányban szerepelnek.

darab		minek ismerte fel				
		beszéd	furulya	kürt	kutya	szék
felismerendő	beszéd	572	16	18	26	26
	furulya	14	240	24	17	6
	kürt	19	12	233	21	5
	kutya	36	15	53	385	8
	szék	24	11	46	22	102

4.5 táblázat Confusion mátrix darabszámra (példa)

%		minek ismerte fel				
		beszéd	furulya	kürt	kutya	szék
felismerendő	beszéd	87	2	3	4	4
	furulya	5	80	8	6	2
	kürt	7	4	80	7	2
	kutya	7	3	11	77	2
	szék	12	5	22	11	50

4.6 táblázat Confusion mátrix százalékosan (példa)

4.4.3 A teljes hangesemény osztályozása

Mind a feature vektorok generálásánál, mind az osztályozásnál a hangokat eddig csak szegmensenként vizsgáltuk, ám a célunk a teljes hangesemény osztályozása. Ehhez az egyazon hangeseményekhez tartozó szegmenseket együtt kell vizsgálnunk, és a szegmensek eredményeiből egy eredőt kell számolnunk. Ezt többségi döntéssel tesszük meg. Az új fájl feldolgozása előtt az előzőre hozunk egy döntést arról, hogy milyen hangosztálynak ismerte fel. Ennél a pontnál hozzuk meg a többségi döntést, és csak azután írjuk be a confusion mátrixba. A program a beolvasott fájlnevek alapján azonosítókat rendel a különböző hangforrásokból származó fájlokhoz, így az azonosító 1, 2, 3, 4 vagy 5 lehet.

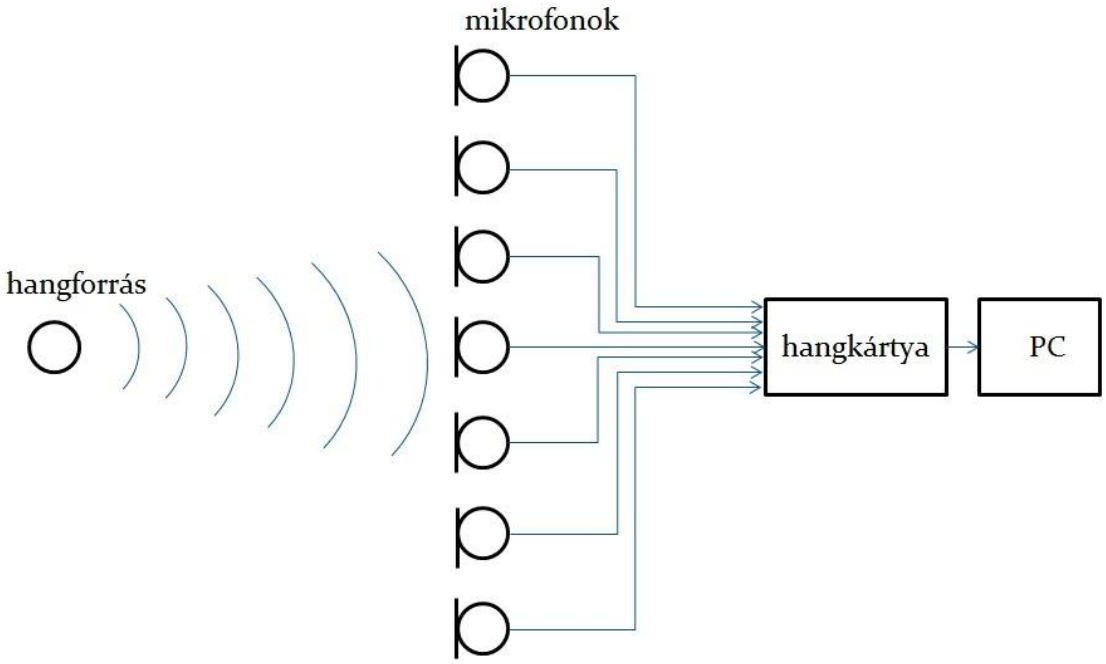
A fenti leírást illusztrálandó, a következőkben egy rövid példát mutatok be egy valós hangfájl kiértékeléséről. Összehasonlítás és ellenőrzés céljából kiírattam a valódi osztályt és a felismert osztályt, értelemszerűen, ha azonos volt, akkor jól ismerte fel. Ez a Matlabban az alábbi módon nézett ki:

```
Valodi: 5;   osztaly: 1;
Valodi: 5;   osztaly: 4;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 5;
Valodi: 5;   osztaly: 3;
Valodi: 5;   osztaly: 4;
Valodi: 5;   osztaly: 4;
```

Látható, hogy az ötös osztályt, azaz a széktolás hangját egyszer ismerte fel beszédnek, egyszer kutyaugatásnak, azután hatszor széktolásnak, egyszer kürtnek, kétszer kutyának. Furulyának pedig egyszer sem. Tehát összességében legtöbbször eltalálta a megfelelő hangosztályt.

5 Mérések

A mérési elrendezés megtervezése elsősorban a mikrofonok elhelyezésére koncentrálódott, majd ezután hanganyagot gyűjtöttünk. Ehhez több mérést végeztünk el. Minden esetben hét szinkron módon működő mikrofonnal mértünk, amelyeket egy laptopon hangkártya segítségével rögzítettünk. A sokcsatornás adatgyűjtő és erősítő közvetítette a jeleket egy számítógép felé, amely valamilyen formátumban eltárolta azokat. A mérési elrendezések minden esetben fixek voltak, ismertek voltak a mikrofonok koordinátái, illetve a hangforrás pozíciója, de ez utóbbit csak az eredmény összevetésére használtuk fel, a program nem kapta meg paraméterként.



5.1 ábra Sematikus mérési elrendezés

A mérés során az alábbi fontosabb eszközöket használtuk fel:

- 7 db Behringer ECM-8000 mérőmikrofon
- Oneway RTO3 8 csatornás mikrofonerősítő
- Cakewalk UA101 8 csatornás külső hangkártya
- PC
- hangforrások (beszéd, furulya, kürt, kutyaugatás, széktolás)

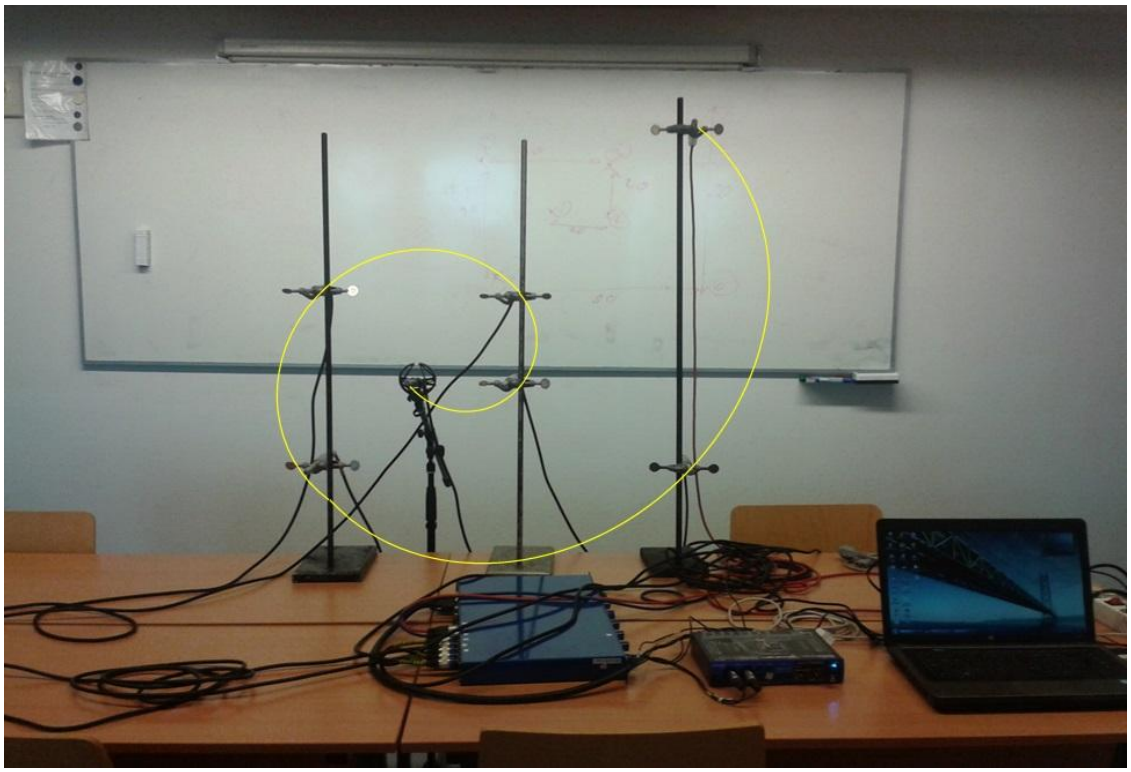
Megjegyzendő, hogy a kutyaugatást egy adott felvétel lejátszásával mesterségesen generáltuk.

A hanganyaggyűjtést beltérben végeztük el. Ez felvet bizonyos nehézségeket, például a reflektált hullámok zavaró hatását, a felvételekbe belejátszhat a teremrezonancia, ugyanakkor leszűkíti mind a meghatározandó távolságokat, mind például azt a fókusztávolságot, amit a program futása során felhasználunk. Ugyanakkor a terem megfelelő kiválasztásával megpróbáltuk minimalizálni a beltéri mérésből fakadó hibalehetőségeket.

A mérés helyszínének a Budapesti Műszaki Egyetem Informatikai épületének IE224-es termét választottam. A teremben nincsenek szekrények vagy olyan nagy tárgyak, amelyek tovább befolyásolhatják a visszaverődéseket. A környezeti zajok minimalizálása érdekében az ablakokat becsuktuk. A duplafalú üvegek meglehetősen jó hangszigetelést biztosítottak a kültéri zajok beszűrődése ellen.

Ahogy az 5.2 ábrán is látható, a terem közepén és két oldalán végeztünk méréseket. Az átláthatóság kedvéért a terem két oldalát az IE 225 felőli illetve a HIT felőli oldalnak fogom hívni a továbbiak során, ahol a HIT rövidítés a Híradástechnikai Tanszékre utal (a tábla felől nézve a terem jobb oldala felé van a Híradástechnikai Tanszék, a bal oldalon pedig az IE 225 terem). A felvételek során három különböző távolságból (3.3 m, 5.8 m, 8.2 m) középen és a terem két szélén a fal mellett állt a hangforrás (jelen esetben a konzulensem) az 5.3 ábrán mutatott mérési pontokban. Így méréseinket összesen kilenc helyen vettük fel, ezeket a pozíciókat a következő ábrán csillagok jelölik.

5.2 ábra Egy mérési elrendezés fényképe, a csillaggal jelölt helyek a hangforrás pozíciói



5.3 ábra Spirális mikrofonelrendezés fényképe

A tér letapogatását 30 cm-es osztásokban végeztük el, ennek fényében a lokalizáció eredményének elvi pontossága is 30*(√2)/2, azaz 21.2 cm, mivel ez a legnagyobb távolság a 30x30 cm-es négyzeten belül, ami még a négyzethez tartozó ponthoz van a legközelebb (a négyzet átlójának a fele).

Hangfelismerés esetén csak a spirál alakú mikrofonelrendezésben készült mintákat használtuk fel, hogy eredményeink egységesen kiértékelhetőek legyenek, ráadásul kör elrendezésben csak beszédről készült felvétel. Annak érdekében, hogy a különböző hangosztályokból egyenlő számú fájl álljon rendelkezésre, a kör mikrofonelrendezéssel készült hangokon túl a terem középvonalában felvett beszédet sem olvastattuk be a programmal. Így mind az öt hangosztályból kilenc fájlt használunk fel, amelyeket a terem 9 pontjában mértünk.

6 Eredmények

6.1 A paraméterek beállításai

Mind a lokalizáció, mind a hangosztályozás esetében az algoritmusok lekódolása után viszonylag sok időt töltöttem a paraméterek megfelelő beállításával. Beállítottam egy paramétert az aktuális többi függvényében, majd így haladtam a többivel. Ez nem éppen az optimális megoldás volt. N db paraméter beállítása esetén ugyanis az optimum egy N dimenziós mátrixban keresendő, ám ennek megkeresése sok időt és energiát vett volna igénybe. Ráadásul, ahogy a tapasztalatom mutatta, ha nem is lett volna felesleges, de sokkal több időbe telt volna, mint amilyen javulást eredményezett volna. Ezt azért feltételezem, mert a paraméterek egyenként módosításával csak kis javulást tudtam elérni az eredményeket tekintve.

A lokalizáció során beállított főbb paraméterek:

- szegmensek hossza
- fókusztávolság
- alul- és felüláteresztő szűrők vágási frekvenciája
- triggerszint

A hangfelismerés során beállított főbb paraméterek:

- a Mel spektrum alsó és felső frekvenciája
- a Mel szűrőbank hány szűrőből álljon
- az osztályozás hány legközelebbi szomszédot vizsgál
- szegmensek hossza
- átlapolódás mérete
- alul- és felüláteresztő szűrők vágási frekvenciája
- triggerszint

6.1.1 A fókusztávolságtól való függés

A programban kritikus pont volt a fókusztávolság beállítása, ugyanis e paramétert kézzel állítjuk be a futás előtt, de nem tudjuk, hogy a hangforrás milyen távolságban helyezkedik el. Ezért vizsgáltam meg, hogy hogyan függ a hiba a fókusztávolságtól. Azt vártam, hogy az az adatok alapján talán meg lehet adni egy fókusztávolságot, amely egyik futásra sem ad nagy hibát. Táblázatba gyűjtöttem a valós távolsághoz adott fókusztávolságra beállított programfutás hibáit a valós pozícióhoz viszonyítva. Előzetesen azt vártuk, hogy a hiba a 3x3-mas mátrixok főátlójában lesz a legkisebb, mivel az az optimális, amikor olyan messzire fókuszálunk, amennyire a hangforrás valójában van. Ehhez képest a 6.1 táblázatban látható módon a pirossal jelölt helyeken kisebb értékeket kaptunk az elvárt főátlóbelieknél. Ahol ezek eltértek, minden esetben a nagyobb fókusztávolság eredményezett kisebb hibát. Ez feltételezhetően azzal lehet összefüggésben, hogy a triggerelt szegmensek pozíciói a valós forráshoz viszonyítva általában az origóhoz voltak közelebb, a távolabbi oldalra ritkán estek koordináták.

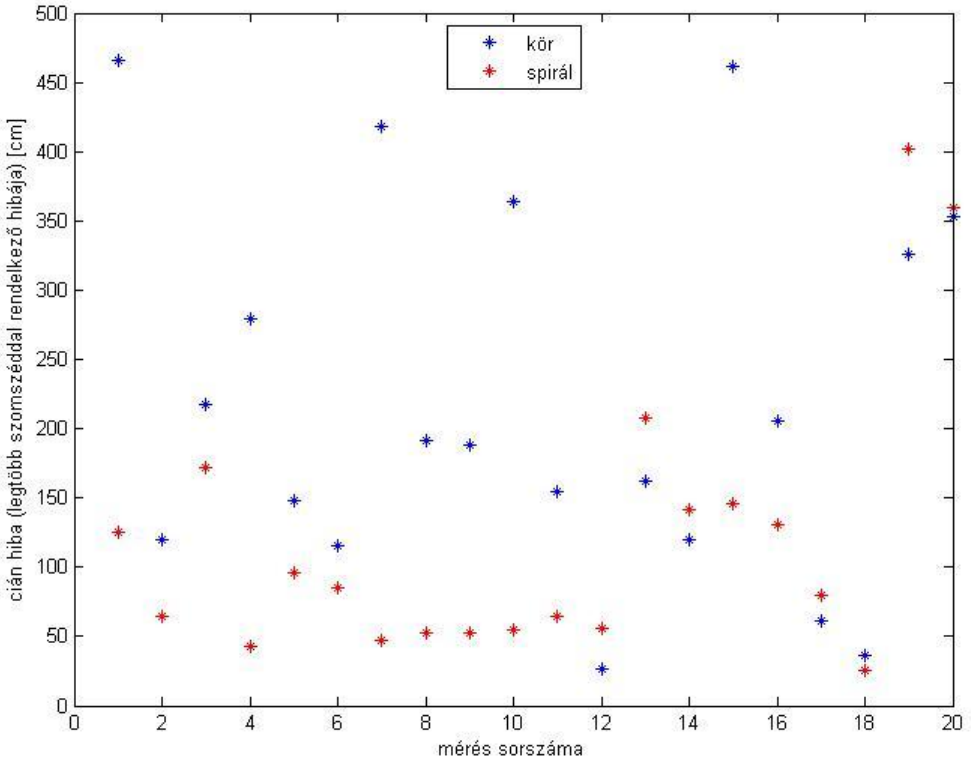
	HIT Valós távolság: 3.3 m / 5.8 m / 8.2 m			Közép Valós távolság: 3.3 m / 5.8 m / 8.2 m			225 Valós távolság: 3.3 m / 5.8 m / 8.2 m		
3.2 m fókusz	**116**	220	216	**16**	43	156	**45**	206	208
5.2 m fókusz	94	**115**	124	27	**43**	153	46	**157**	151
8.2 m fókusz	141	96	**47**	55	65	**117**	75	125	**77**

6.1 táblázat Triggerelt szegmensek átlagos négyzetes eltérése a fókusztávolságtól függően [cm]

Mivel a fókusztávolságot nem lehet kinyerni a felvételekből, azt előre be kell állítanunk. Így a fenti eredmények tudatában 5.2 méterre állítottam, mivel úgy egyik esetben sem ad nagy hibát, ráadásul például a HIT oldalról 3.3 méter távolságból készített felvétel esetében jobb eredményt is ad, mint a valóságnak megfelelő 3.2 méteres fókusztávolság.

6.2 A mikrofonelrendezések eredményei

A mikrofonelrendezések szimulációinak két legjobb eredménye a kör és a spirál alakú elrendezés lett. Mindkét elrendezésben végeztünk méréseket. A méréseket mindkét esetben egyazon helyről készítettük, azaz a hangforrás mindkét elrendezésben azonos pozícióban volt. Az alábbi ábrán ezek összehasonlításai láthatóak.

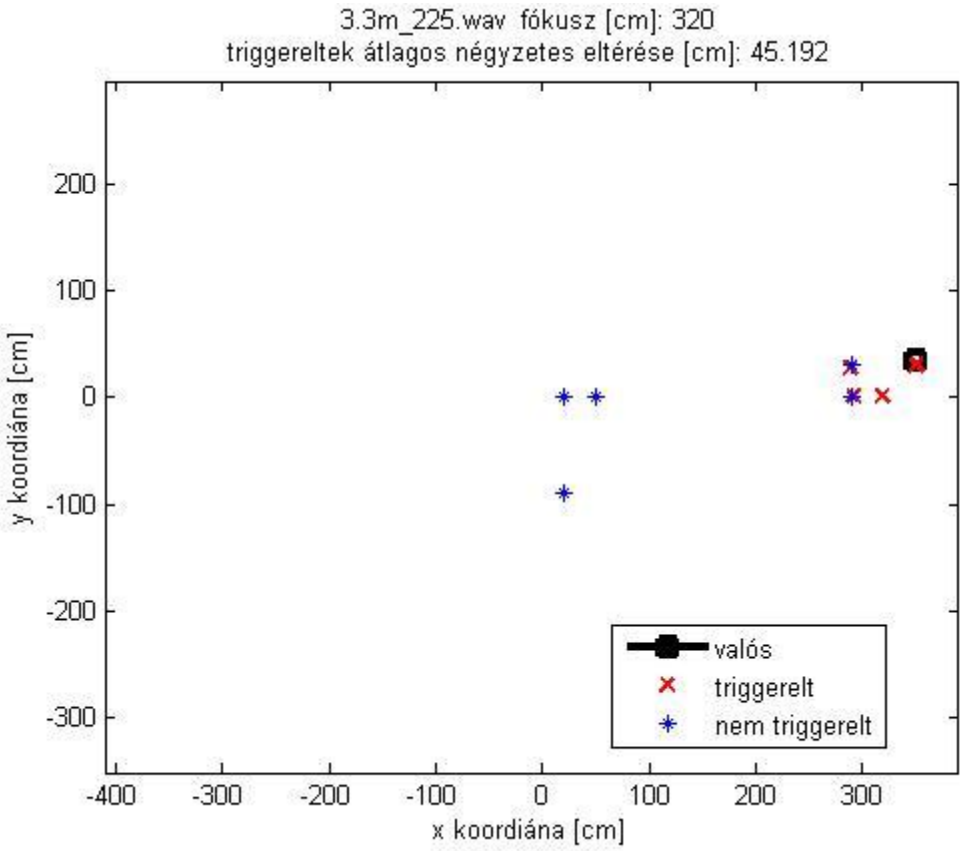


6.2 ábra A kör és spirál mikrofonelrendezés eredményei

Az ábrán jól látszik, hogy a kör alakú mikrofonelrendezések méréseihez legtöbbször nagyobb szórás tartozik. Egy összhangba vág a szimulációs eredményeinkkel, amelyben az x koordináta szerinti pontosabb becslés miatt a spirál elrendezést tekintettük a legjobbnak.

6.3 A lokalizáció eredményei

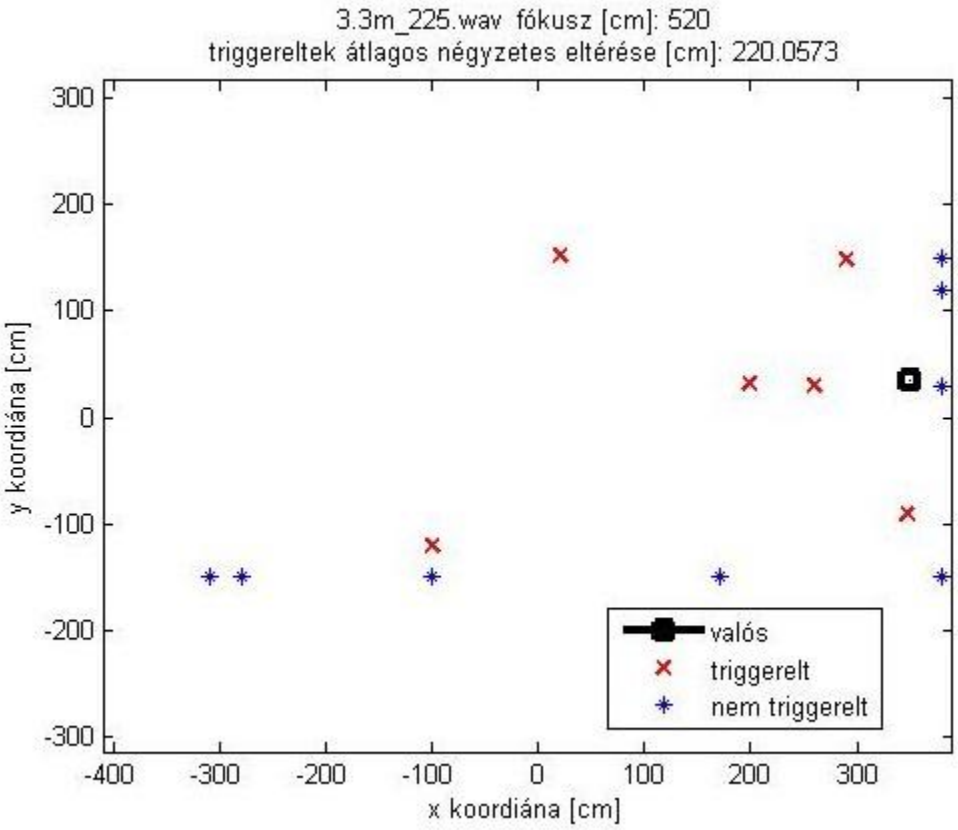
Az alábbi ábrán egy tipikus eredmény látható, amelyben egy piros vagy kék jel egy szegmens számolt koordinátájához tartozik attól függően, hogy a szegmens elérte-e a triggerszintet:



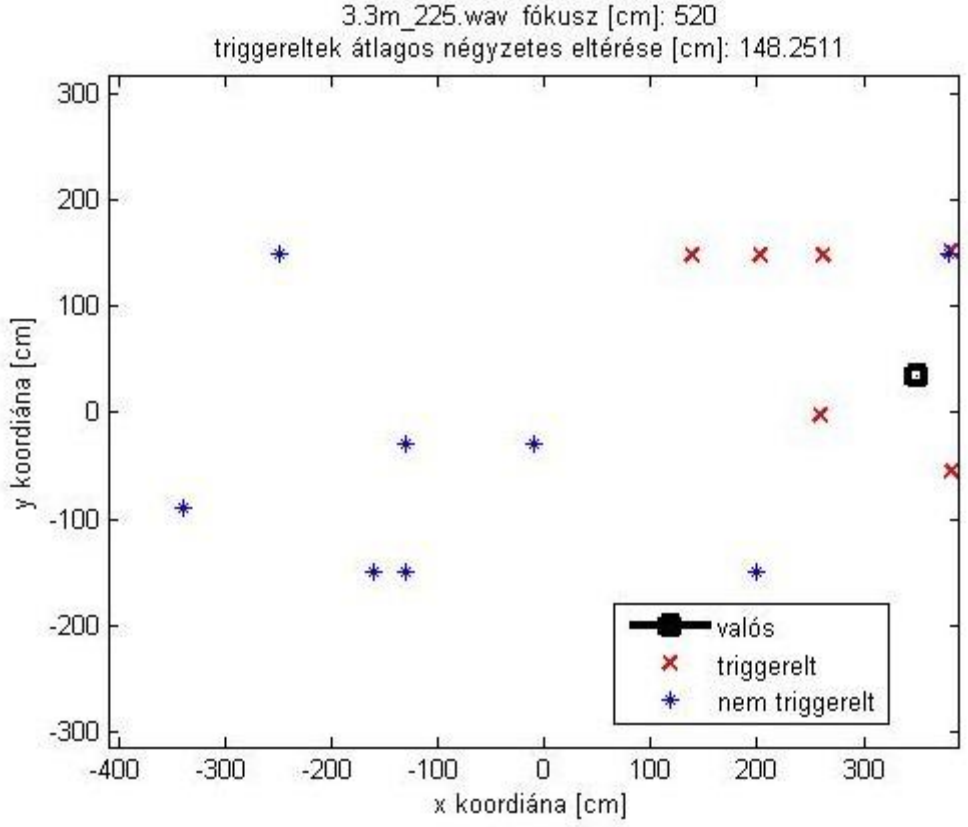
6.3 ábra Tipikus mérési eredmény

A fenti ábrán az átlagos négyzetes eltérés értéke a valós pozícióhoz képest a triggerelt szegmensek átlagos négyzetes eltéréseire értendő. Általános megfigyelésünk volt, hogy a triggerelt szegmensek számolt pozíciói legtöbbször alulról közelítették meg a valós pozíciót, vagyis közelebb voltak az origóhoz.

A 6.4 és 6.5 ábrákon a delay and sum módszer eredményeit hasonlítom össze azon két esetre, amikor a késleltetéssel visszatolt időfüggvényeket összeszoroztuk illetve összeadtuk.



6.4 ábra A delay and sum módszer eredménye összeadásra



6.5 ábra A delay and sum módszer eredménye szorzásra

A 6.4 és a 6.5 ábrákon látható, hogy a delay and sum módszer szorzásra kisebb átlagos négyzetes eltérést eredményezett, de a hiba még így is elég nagy, másfél méter.

6.3.1 Az egységesség vizsgálata

Megvizsgáltam azt, hogy a szegmensek eredményei mennyire koncentrálódnak egy területre, mennyire egységesek. Ehhez szórást számoltam, amelyet az alábbi módon valósítottam meg:

$$kozep = \frac{1}{k} * \sum_{k} koordTrig_k$$

$$atlelteres_k = |koordTrig_k - kozep|$$

$$szoras = \sqrt{\left(\frac{1}{k} * \sum_{k} (atlelteres[k]_x)\right)^2 + \left(\frac{1}{k} * \sum_{k} (atlelteres[k]_y)\right)^2}$$

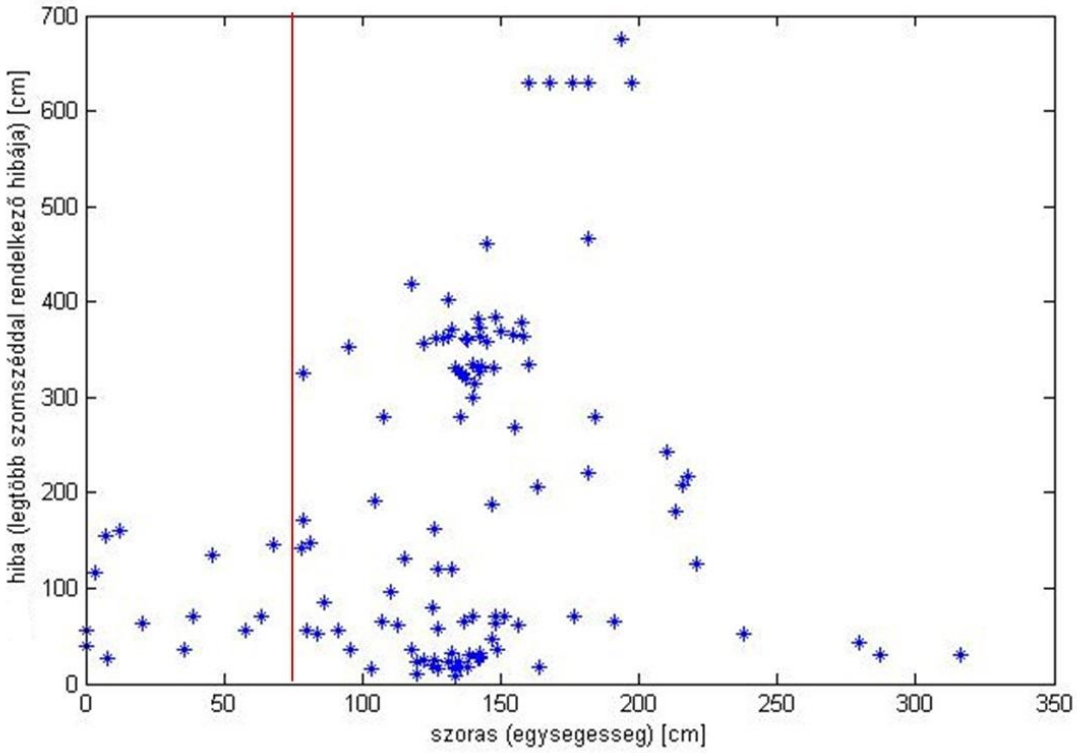
$koordTrig_k$- a *k*-adik triggerelt koordinátapár

$kozep$- a triggerelt koordináták átlaga

$atlelteres_k$- a triggerelt koordináták eltérése az átlagtól

$szoras$- a triggerelt koordináták szórása (egységessége)

Érdekességképpen megvizsgáltam, hogy mi az összefüggés a szegmensek egységessége és a hibája között. Azt reméltem, hogy ha a szegmensek kis területre koncentrálódnak, akkor nagyjából el is találják a forrás pozícióját, azaz kicsi lesz a hibájuk, ha viszont nagy a szórásuk, akkor nehéz elérni, hogy pontos eredményt adjanak, így a hiba nagyobb lesz. Így ideális esetben akár lineáris görbét is lehetne illeszteni a pontokra. Ez persze az elméleti optimum, de reméltem, hogy az eredmények hasonlítani fognak a görbéhez. Az eredmények a következő ábrán láthatók, amelyen a kis szórás jelenti az egységességet.

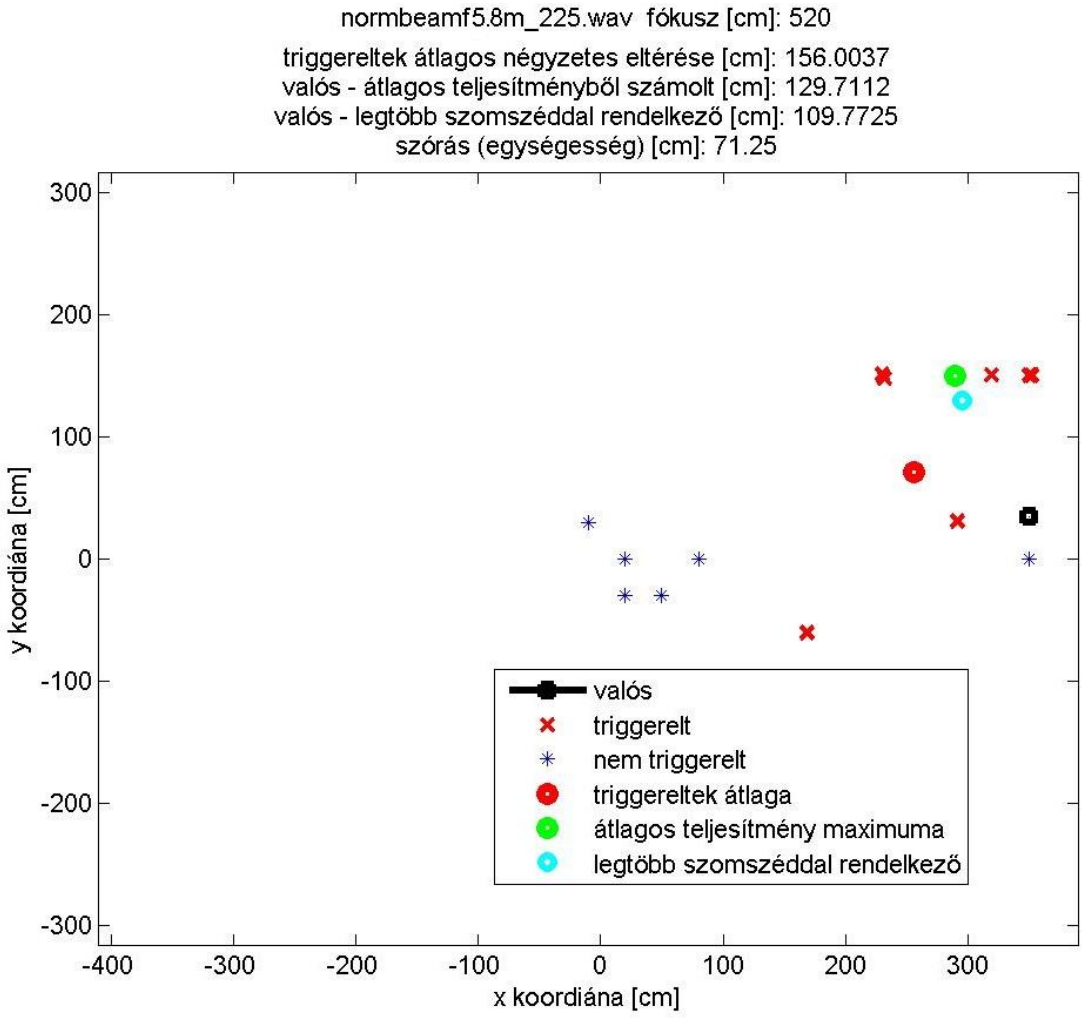


6.6 ábra Egységesség - hiba összefüggése

A vártnál rosszabb eredményt kaptam. Azt tapasztaltam, hogy közepesen nagy szórás esetén egyaránt előfordulhat kis és nagy hiba. Egy dolog viszont örömmel töltött el. Ha gondolatban függőleges vonalat húzunk a 75 cm-es szóráshoz, akkor láthatjuk, hogy az annál kisebb szóráshoz viszonylag kis hiba társul. Az ábrán látható, hogy ez a viszonylag kis hiba is lehet akár másfél méter, ami nem túl jó eredmény, de ezáltal egy kisebb térrészt tudunk mondani, ahol a lokalizálandó hangforrás elhelyezkedhet. Tehát várakozásainknak csak egy része teljesült. Az összefüggés később felhasználható a lokalizáció pontosságának jellemzésére.

6.3.2 A különböző módon számolt eredő pozíciók hibái

A triggerelt szegmensekből a 3.5 fejezetben bemutatott módon háromféleképpen számoltam eredő pozíciót. A 6.7 ábrán ezen eredő pontok hibáinak összehasonlítása látható egy példán. A címbe kiírattam a hibákat. Programom a képeket automatizáltan lementette egy megadott könyvtárba, a hozzá tartozó hibaértékeket pedig időbélyeggel ellátva kiírtam egy szöveges fájlba. Ilyenformán az eredmények és a változások visszakövethetőek voltak.



6.7 ábra Különböző hibaszámítások eredményei

Az ábrán látható, hogy a legnagyobb hibát az átlagos négyzetes eltérés, a legkisebbet pedig a legtöbb szomszéddal rendelkező hibaszámítás eredményezte. Ez utóbbi általános tapasztalat volt.

Elvégeztem a különböző módszerek statisztikai összehasonlításait. A hibákat mind a három hibaszámítási módszer szerint kiszámoltam. A 6.8 táblázatban sok eredmény átlagos hibája szerepel. A szöveges fájlba elmentett eredményeket offline dolgoztam fel, amelyekre az alábbi átlagos hibák adódtak:

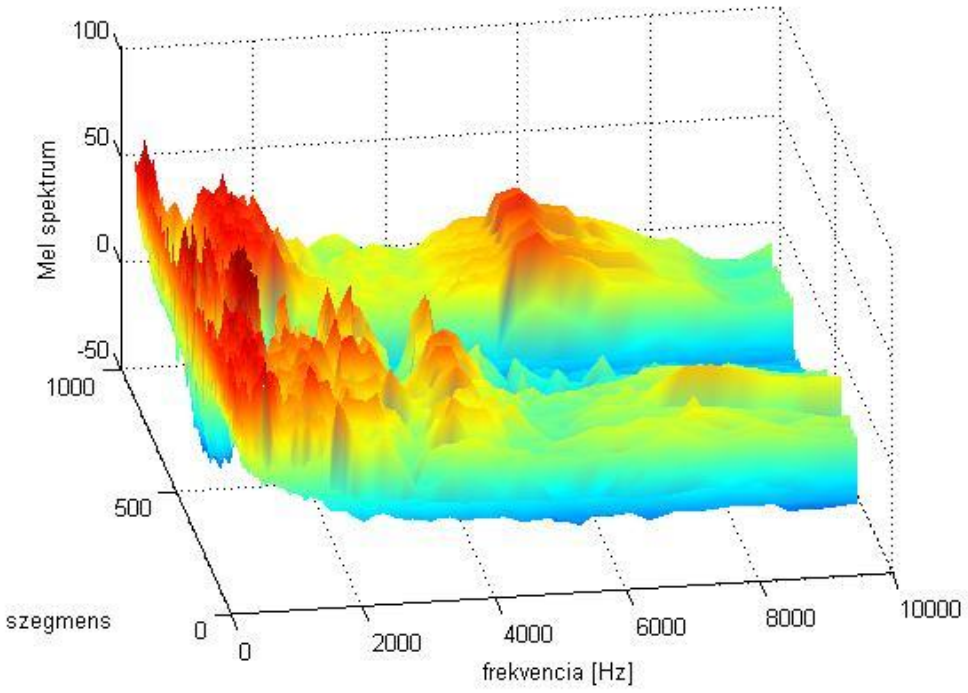
Hibaátlagok [cm]	delay and sum összeadásra	delay and sum szorzásra	normál beamforming nem normalizált	normál beamforming normalizált	capon beamforming normalizált	capon beamforming nem normalizált
triggereltek átlagos négyzetes eltérése	321.8	325.4	293.8	233.9	259.0	297.0
átlagos teljesítményből számolt hiba	269.2	265.3	238.5	235.4	321.5	321.5
a legtöbb szomszéddal rendelkező hibája	276.8	276.3	246.1	186.9	253.5	269.7

6.8 táblázat Módszerek és hibaátlagaik [cm]

A táblázatban látható, hogy a triggereltek átlagos négyzetes eltérése egyik esetben sem adta a legjobb eredményt. Ezt már sejtettük a 6.7 ábra alapján. Megfigyelhető, hogy míg a dely and sum módszerekre még az átlagos teljesítményből számolt hiba a legjobb, a capon beamformingnál már a legtöbb szomszéddal rendelkező hibája adja a legkisebb értéket. A táblázatban szereplő értékek közül kitűnik a normalizált beamformingra kapott 186.9 cm-es átlagos hiba, így végül legjobb módszernek a normalizált beamforming módszert, az eredő pozíció számítási módjának pedig a legtöbb szomszéddal rendelkező pontot választottam. A 6.8 táblázatban szereplő hibák az összes fájlra (120 fájlra) lefuttatott átlagok, így elmondható, hogy megfelelő mintaszám miatt a statisztikák reprezentatívnak tekinthetők.

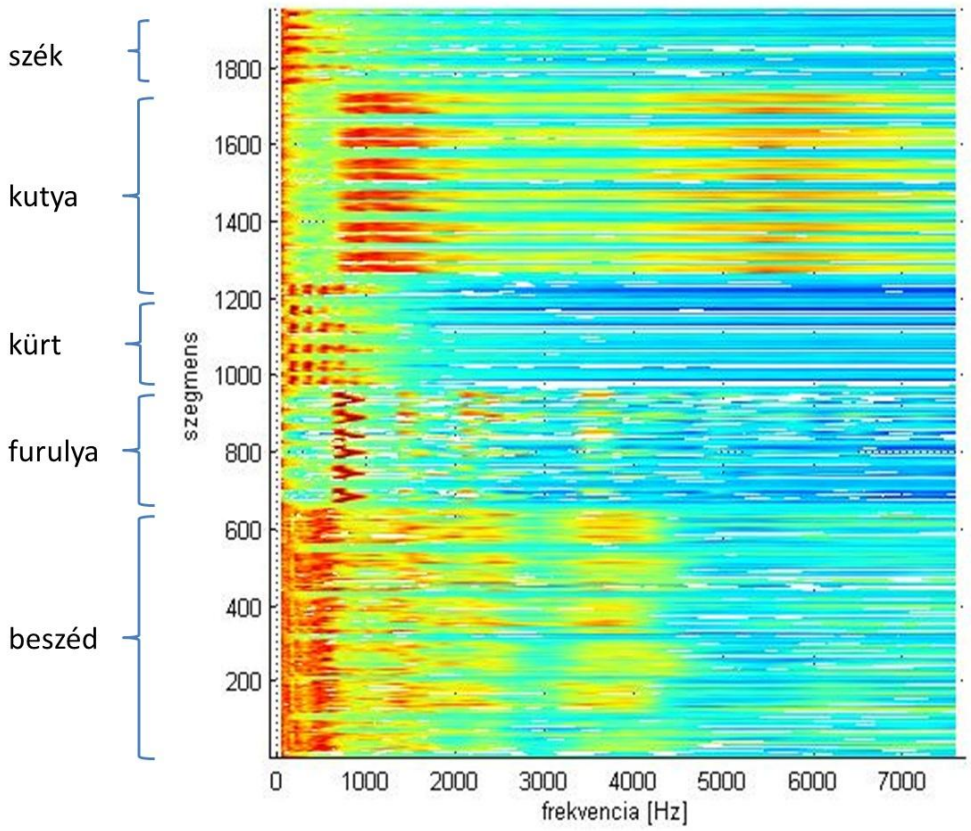
6.4 A forrásazonosítás eredményei

A forrásazonosítás első lépéseként az idősorokból olyan tulajdonságvektorokat generáltam (feature vektorok), amelyek jól jellemzik a hangot. Ennek eredménye egy háromdimenziós mátrix volt, amelyben a tulajdonságvektorok a szegmensek és a frekvencia függvényében jelentek meg. Ezt mutatja be a 6.9 ábra.



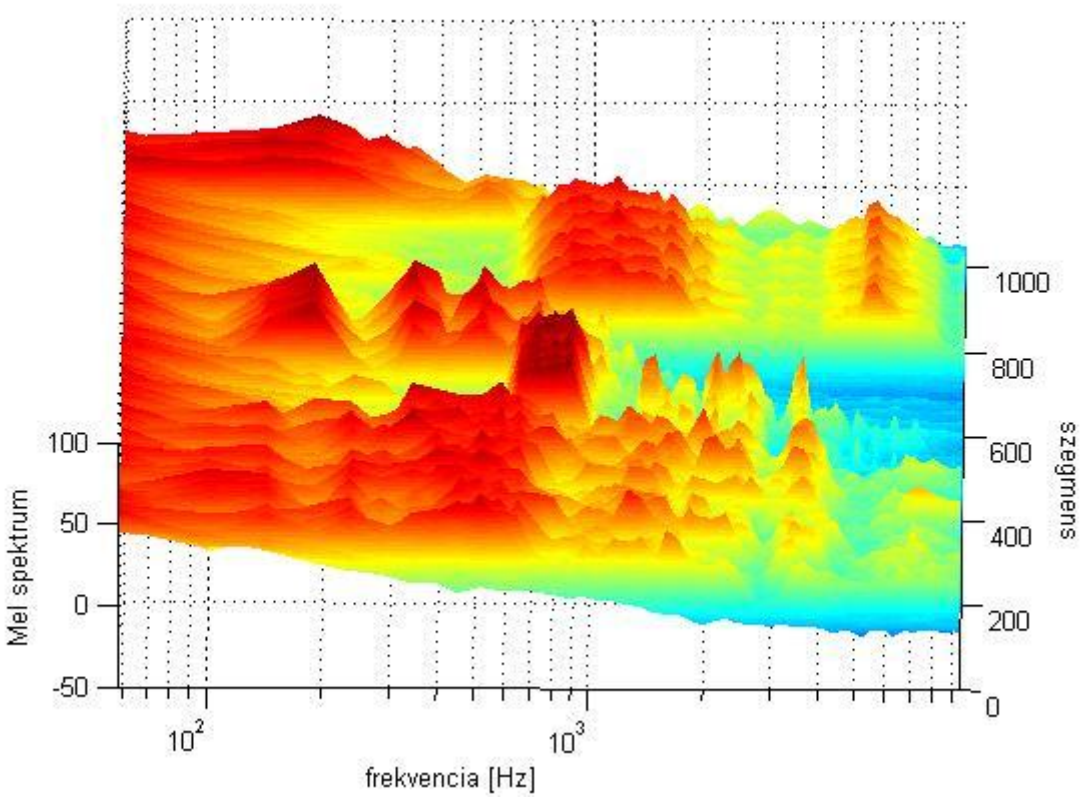
6.9 ábra A feature vektorok lineáris frekvenciaskálán

Ha ezt az ábrát a 6.10 ábrán látható módon beforgatjuk, jól elkülöníthetővé válnak a különböző hangosztályokhoz tartozó feature vektorok. Az ábrán látszik, hogy beszédből hosszabb felvételek készültek, továbbá az, hogy a kutyaugatás fedi le a legszélesebb frekvenciatartományt.



6.10 ábra A feature vektorok különböző hangosztályokra

A 6.11 ábrán a feature vektorokat logaritmikus léptékben ábrázoltam. Így jól látszik, hogy a Mel spektrum 60 Hz-től indul. Ez megfelel a Mel spektrum első sávközépi frekvenciájának. A Mel frekvenciatengely kezdőpontjának 20 Hz-et adtunk meg, de az első háromszögablak kezdőfrekvenciája még nem lesz sávközépi frekvencia, csak a háromszög csúcspontja, ami egybevág a második háromszögablak kezdőfrekvenciájával, ez a 60 Hz (lásd 4.3.2 fejezet).



6.11 ábra A feature vektorok különböző hangosztályokra

A 6.12 és a 6.13-es táblázat a forrásazonosítás végeredményeit mutatja. A következőkben csak a Mel spektrummal számolt eredményeket mutatom be. Ennek az az oka, hogy bár előzetesen FFT-vel is végeztem szimulációkat, azok eredményei sokkal rosszabbak lettek, mint a Mel spektrummal számolt módszer esetében.

Az eredményként kapott hangosztályokat összeszámoltuk a confusion mátrixban (felosztás mátrixa), amelyben összesítettük, hogy az egyik csoportból származó tanító minták alapján hányszor ismerte fel megfelelően az adott mintát, és hányszor más hangcsoportnak. Mivel 5 hangosztály van, ezért a mátrix 5x5-ös. Ennek megfelelően a mátrixban a diagonális helyén várjuk a nagyobb értékeket. A második táblázatban ugyanezek az értékek százalékos arányban szerepelnek.

darab		minek ismerte fel				
		beszéd	furulya	kürt	kutya	szék
felismerendő	beszéd	572	16	18	26	26
	furulya	14	240	24	17	6
	kürt	19	12	233	21	5
	kutya	36	15	53	385	8
	szék	24	11	46	22	102

6.12 táblázat Confusion mátrix szegmensekre

%		minek ismerte fel				
		beszéd	furulya	kürt	kutya	szék
felismerendő	beszéd	87	2	3	4	4
	furulya	5	80	8	6	2
	kürt	7	4	80	7	2
	kutya	7	3	11	77	2
	szék	12	5	22	11	50

6.13 táblázat Confusion mátrix százalékosan

A 6.12 táblázatban a szegmensekre kiszámított eredményeket számláltuk össze és írtuk be a megfelelő helyre attól függően, hogy azt az algoritmus minek ismerte fel. A módszer jó eredményeket adott, egyedül a széktolás esetében ismerte fel 50% arányban a széktolás hangjának, 12%-ban, 5%-ban, 22%-ban, és 11%-ban pedig másnak. Ebből látható, hogy még mindig széknek ismerte fel legtöbbször, de rájöttünk arra, hogy ha valamiféle többségi szavazással döntenénk a végső megítélésben, akkor jobb eredményeket kapnánk.

Az ötlet megvalósítása egy fájl feldolgozására terjedt ki, tehát a korábbi szegmensenkénti helyett a 4.4.3 alfejezetben leírt módon hangeseményenként (fájlonként) hozunk egy összegzett döntést arról, hogy a hang melyik hangosztályba tartozik. Ezt többségi szavazással valósítottuk meg. A módszer jelentősen megnövelte az eredményességet, és így már mind a hat fájlra pontosan eltalálta a megfelelő hangosztályt. Ezek eredménye a következő táblázatban látható.

darab (fájlonként)		minek ismerte fel				
		beszéd	furulya	kürt	kutya	szék
felismerendő	beszéd	6	0	0	0	0
	furulya	0	6	0	0	0
	kürt	0	0	6	0	0
	kutya	0	0	0	6	0
	szék	0	0	0	0	6

6.14 táblázat Confusion mátrix fájlokra

7 Összegzés, konklúzió

Összefoglalásképpen elmondhatjuk, hogy legelőször megterveztem a jelfeldolgozás folyamatát, majd különböző lokalizációs és forrásazonosító algoritmusokat implementáltam. A lokalizációs algoritmusok idő- vagy frekvenciatartományban késleltetik meg a mikrofonokból beérkező jelet, majd ezeket összeadva teljesítményt számoltam. A teljesítmény maximuma adja meg a feltételezett forrás pozícióját. A forrásazonosítás két részre volt osztható. Elsőként az időfüggvényekből tulajdonságvektorokat generáltunk, amelyek jól jellemzik a hangot. Ezután a tulajdonságvektorokat osztályoztuk, azaz meghatároztuk, hogy a milyen forráshoz tartozhatnak. A mérések elvégzése után kiértékeltem az eredményeket.

Arra jutottunk, hogy az implementált lokalizációs eredmények csak részben feleltek meg elvárásainknak, a nagy szórás miatt az algoritmusok ilyenformán inkább csak térrészek elkülönítésére alkalmasak.

A forrásazonosítás viszont nagyon jó eredményeket adott. A többségi döntésnek köszönhetően minden esetben felismerte a megfelelő hangosztályt. Megjegyzendő, hogy módszerünket elég jól elkülöníthető hangosztályokból származó hangokra teszteltük, ám azokra kiválóan szerepelt.

8 Köszönetnyilvánítás

Ezúton szeretném megragadni az alkalmat arra, hogy köszönetemet fejezzem ki mindazoknak, aki a diplomamunkám során valamilyen formán segítettek. Külön köszönettel tartozom konzulensemnek, dr. Orosz Györgynek a segítőkészségéért, és azért, hogy ösztönzött feladatom precíz megoldására.

Irodalomjegyzék

[1] *Radar*, Wikipédia, (2014 december 1.) http://hu.wikipedia.org/wiki/R%C3%A1di%C3%B3lok%C3%A1tor

[2] *Ultrahang*, Wikipédia (2014 december 1.), http://hu.wikipedia.org/wiki/Ultrahang

[3] *Mel Frequency Cepstral Coefficient (MFCC) tutorial*, Practical Cryptography (2014 december 1.) http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[4] *Incoherent Frequency Fusion for Broadband Steered Response Power Algorithms in Noisy Environments*,Daniele Salvati, Carlo Drioli, Member, IEEE, and Gian Luca Foresti, Senior Member, IEEE, IEEE Signal Processing Letters, Vol. 21, No. 5, May 2014

[5] *Accurate Indoor Ultrasonic Position Tracking*, Györke Péter, Proceedings Of The 21st Phd Mini-Symposium, Budapest University Of Technology And Economics, Building I, February 3, 2014

[6] *k legközelebbi szomszéd*, scholarpedia.org, 2014. december 1. http://scholarpedia.org/article/K-nearest_neighbor

[7] *k legközelebbi szomszéd*, 2014 december 1. http://blog.csdn.net/jianxi602/article/details/35800641

[8] J. J. Christensen - J. Hald, *Beamforming*. Brüel & Kjaer Technical Review, 2004. 20. oldal

[9] *Two Decades of Array Signal Processing Research*, IEEE Signal Processing Magazine 1996 July 72-75. oldal

[10] *Beamforming method: suppression of spatial aliasing using moving arrays*, A. Cigada, M. Lurati, F. Ripamonti, M. Vanali
(Berlin Beamforming Conference (BeBeC) 2008 February 19. and 20. 5-6. oldal
http://bebec.eu/Downloads/BeBeC2008/Papers/BeBeC-2008-19_Cigada_Lurati_etal.pdf

[11] *Delay Sum Beamforming*, The Lab Book Pages,2014. december 1. http://www.labbookpages.co.uk/audio/beamforming/delaySum.html

[12] *A fül felépítése és működése*, Debreceni Egyetem Klinikai Központ, 2014. december 1. http://egeszsegcentrum.intelliopen.hu/az-emberi-test/erzekszervek/66-a-ful-felepitese-es-mukodese

[13] *Sound Pattern Matching Using Fast Fourier Transform in Windows Phone*, Nokia, 2014. december 1. http://developer.nokia.com/community/wiki/Sound_pattern_matching_using_Fast_Fourier_Transform_in_Windows_Phone

[14] *Support vector machine*, Wikipédia, 2014 december 1, http://en.wikipedia.org/wiki/Support_vector_machine

[15] *Minimum variance adaptive beamforming applied to medical ultrasound imaging* Johan-Fredrik Synnevag Andreas Austeng, Department of Informatics, University of Oslo, P.O. Box 1080, N-0316 Oslo, Norway, Sverre Holm

[16] D. Salvati, C. Drioli, G. L, Foresti, *Incoherent Frequency Fusion for Broadband Steered Response Power Algorithms in Noisy Environments*, IEEE Signal Processing Letters, Vol. 21, No. 5, May 2014

[17] *Hangazonosítás*, Balla Gábor hangtechnikai szakértő, 2014 december 1. http://hangtechnikaiszakerto.hu/hangtechnika/hangazonositas.html