



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**

Villamosmérnöki és Informatikai Kar

Hálózati Rendszerek és Szolgáltatások Tanszék

# HRTF kiterjesztése távolságérzet szimulációjával sugárkövetés alapon

*Készítette:*

Sgánetz Bence

*Konzulensek:*

Dr. Firtha Gergely

Dr. Fiala Péter

Budapest, 2021

# Tartalomjegyzék

<b>Kivonat</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>1. Bevezetés</b>	<b>5</b>
<b>2. HRTF halmazok előállítása</b>	<b>7</b>
2.1. Mérési környezet	7
2.2. Rögzítés	9
<b>3. Előkészületek</b>	<b>13</b>
3.1. Meglévő HRTF szettek vizsgálata	13
3.1.1. Fülek közti késleltetés a beesési szög függvényében	14
3.1.2. Fülek közti hangnyomáskülönbség a beesési szög függvényében	20
3.1.3. 3D esetek kezelése	23
<b>4. HRTF halmazok előfeldolgozása</b>	<b>25</b>
4.1. Kiegyenlítés	26
4.2. Különbségtartó kiegyenlítés	27
<b>5. Virtuális hangtér előállítása</b>	<b>30</b>
5.1. Leképezési módszerek	30
5.2. Konfiguráció	33
5.3. Áttérés fejhallgató-virtualizációra	34
5.4. További lehetőségek hangforrásokkal	36
5.5. Mozgó hangforrások kezelése	37
<b>6. Irányhallás szimulációja</b>	<b>40</b>
6.1. Hallójárat-szimuláció	41
<b>7. Validáció</b>	<b>48</b>
7.1. Elvégezhető tesztek	49
7.1.1. Irányhallási tesztek	49
7.1.2. Tömörítési teszt	51
7.1.3. Tömörítés cseréje valós időben	52
7.1.4. Hitelesítés	55

7.2. Újratervezés	56
7.2.1. HRTF placebo teszt	58
7.2.2. Távolság teszt	59
7.3. Végeredmények	60
7.3.1. Új tesztek eredményei	60
7.3.2. Tömörítések vaktesztjének eredménye	61
<b>8. Dolby Atmos támogatás</b>	<b>62</b>
8.1. Szerkezet	62
8.1.1. Objektumok elhelyezése	63
8.1.2. Objektumok hangjának előállítása	64
<b>9. Összefoglalás</b>	<b>65</b>
<b>Hivatkozások</b>	<b>67</b>

## Kivonat

A HRTF-ek (Head-Related Transfer Function, fej-átviteli függvények) sztereó fejhallgatók irányérzet-szimulációját teszik lehetővé úgy, hogy a megfelelő irányból érkező hanghullámok átviteli függvényét alkalmazzák a hangforráson. HRTF-hez használható impulzusválasz-adathalmazokat a hallójáratba helyezett mikrofonok segítségével mérhetünk, a fej körül mozgatott hangszóróból megszólaltatott gerjesztés segítségével.

A két mikrofon kizárólag a fülkagylók hatását rögzíti, a hallójáratban létrejövő torzításokat nem, pedig legnagyobb mértékben ezek alapján észleljük a hangforrások távolságát. A dobhártyáig nem tudunk mikrofont bejuttatni, ezért a különböző távolságokhoz tartozó impulzusválaszok mérése nehézkes, előállításukhoz szimulációt érdemes alkalmazni. Munkám során sugárkövetéses módszerrel állítom elő a hiányzó információt, és generálok meglévő HRTF-halmazok számára távolságfüggő korrekciós függvényeket.

A feladat elvégzéséhez komplex szimulációs szoftvert hoztam létre, mely a következő feladatok ellátására képes. A generátor alkalmazás három különböző távolságból és megfelelő irányból vett hallójárat-torzítást kombinál a feldolgozandó szett minden mérési pontjával. Az eredmények utófeldolgozásához egy impulzusválaszokhoz szánt EQ szoftvert készítettem, amely a frekvenciaválaszaikat megközelítőleg linearizálja. Az így előállt, irányt és távolságot is átadni képes szettek hatékonyságának vizsgálatához egy 3D térhangzáshoz szánt renderelőt készítettem, amelynek alapja a meglévő piaci megoldások arányos keverési módszere. A távolságot hangforrásonként számolom, mivel ez a szimulált impulzusválaszok egyszerűsége okán jól optimalizálható.

A munka eredménye egy teljes hangmotort tartalmazó függvénykönyvtár, amely képes a renderelt hangok irányát és távolságát átadni egy sztereó fejhallgatón, melynek legfőbb felhasználási módja virtuális valósághoz szánt alkalmazások realitásérzetének növelése.

## Abstract

HTRFs (Head-Related Transfer Function) make a simulated sense of direction possible on stereo headphones by applying the transfer function of the sound waves from the correct direction on the sound source. Impulse response datasets are created by placing microphones in the ear canal, and measuring a speaker with it that moves around the head.

The two microphones only measure the effect of the pinna, not the distortions created in the ear canal, which are the major factor in distance perception. Because microphones cannot be positioned near the ear drum, the impulse responses by distance are hard to measure, and simulation should be used instead. In my work, I use ray-tracing to produce the missing information and generate distance-based correction functions for existing HRTF-sets.

I created a complex simulation software for the objective, which is capable of the following tasks. The generator application combines ear canal distortion from three different distances with the appropriate direction for each measurement point of the set to be processed. To post-process the results, I created an EQ software for impulse responses that approximately linearizes their frequency responses. To examine the efficiency of the resulting sets, which could convey direction and distance, I created a 3D surround sound renderer based on the proportional mixing method of existing solutions. I calculate the distance per sound source, because it can be well optimized due to the simplicity of the simulated impulse responses.

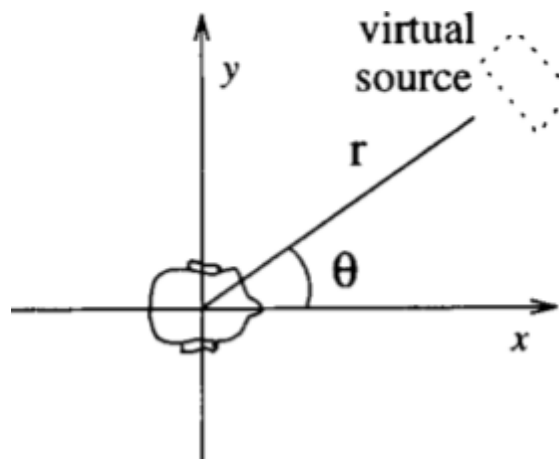
The result of this work is a class library of a complete audio engine, which is able to convey the direction and distance of rendered sounds on a stereo headset. The main use case is increasing the sense of reality of virtual reality applications.

# 1. Bevezetés

A HRTF (Head-Related Transfer Function, fej-átviteli függvény) módszer úgy tesz lehetővé térérzet-szimulációt sztereó fülhallgatókon, hogy mért hangforrások és a két fül közti átviteli függvényt alkalmazzák virtuális hangforrásokon. Egy ilyen megoldás segítségével sokat javul virtuális valósághoz készített szoftverek realitásérzete, illetve lehetővé teszi térhangzású filmek hangsávjának teljes átadását fejhallgatókon keresztül.

Hangszórók és hallójáratokba helyezett mikrofonok közt mért átviteli függvények alapján előáll egy adott sugarú gömb felületén több pont, ahol a virtualizáló szoftver elhelyezhet hangforrást. Ezen pontok halmazát nevezzük HRTF szetteknek vagy halmazoknak, a virtuális térhangzás megteremtéséhez szükséges adatokat jelentik. A pontok között különböző módszerek szerint lehetségesek átmenetek, leggyakrabban a hangforrás irányvektorát közrezáró 3 mérési pont amplitúdóit arányosan állítva. Ez a VBAP (Vector Base Amplitude Panning) módszer.<sup>[1]</sup>

Ideális esetben HRTF szettek mérését süketszobákban végzik, motorizáltan mozgatott hangszórót forgatva a mért alany vagy bábu körül. Fix hangszóró-elrendezések közepére ültetett alannyal is hasznos szettek lehet előállítani, viszont tudományos célokra, például a diplomateremben tesztelt módszer pontosítására vagy ellenőrzésére 1°-os felbontású szettek nagy segítséget nyújthatnak. Az alany körül bejárt gömb sugarát fontos közel konstans értéken tartani, hogy a késleltetések egyenletesek legyenek, mert rendereléskor a lehetséges pontok közti mozgások folytonossága érdekében bevezetett átfedés a két vagy több jel közt csak így lehet sima. Ahogy az egyenetlen távolságokban elhelyezett hangszórók is károsan hatnak az átfedések hullámfrontjára, ez a jelenség ugyanúgy megjelenik virtualizáció esetén.



1.1. ábra: A HRTF-ek koordinátarendszere.<sup>[1]</sup>

Egy HRTF pozíció 3 adattal jellemezhető. A koordinátarendszer X tengelye jelenti az alany tekintetének irányát. A tengelytől vett szögeltérés vízszintes és függőleges síkon, illetve a mérési pont - ami a későbbi virtuális hangforrás - távolsága határozza meg egyértelműen a pozíciót. Egy pozícióhoz két mérés tartozik, mindkét fülön mérjük a hangforrás átviteli függvényét. A mérőeszköz emberi alanyok esetében egy hallójáratba helyezett kis méretű mikrofon, bábuk esetében akár a hallójárat helyén elhelyezett, nagyobb és jobb minőségű mikrofon is lehet. Hangforrásként jellegzetesen koaxiális hangszórókat használnak, hogy a magas és mély hangok tökéletesen azonos irányból érkezzenek.

A dolgozat témája, hogy embereken mért HRTF szettekét egészíthessünk ki hallójárat torzításokkal. Emberi vizsgálatoknál nehézkesen oldható csak meg a hallójárat mérése, mivel ehhez a dobhártya helyén vagy közvetlen közelében kellene mikrofont elhelyezni. Ez bábuknál egyszerűen megoldható, viszont az emberi fül mélyére csak orvos segítségével, kényelmetlen folyamattal lehetséges bejuttatni. Egyetlen tesztalanyról bábút készíteni szintén nem idő- és költséghatékony. Az egyszerűség, gyorsaság, és komfort okán ezeket a méréseket leggyakrabban a hallójárat nyílásába illeszkedő mikrofonnal végzik. Ez az elrendezés magában hordozza, hogy a hallójárat teljesen kimarad a mérésből, pedig a rajta áthaladó hullámfront iránya és sugara képes változásokat előidézni a dobhártyára érkező hangképben.

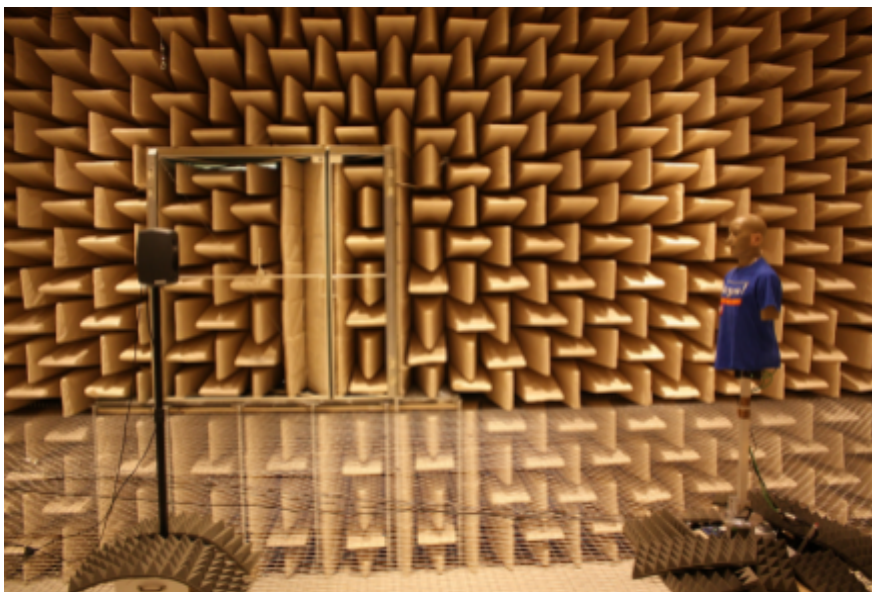
Távolságot szimuláló megoldásként ismertem meg hagyományos HRTF rögzítést több, különböző sugarú gömbön<sup>[2, 3]</sup>, hullámtér-szintézist a tetszőleges pontban elhelyezkedő hangforrás hullámfrontjának replikálásához<sup>[4]</sup>, és sugárkövetés használatát is a teljes fej szimulációjához.<sup>[5]</sup> A munkámat leginkább meghatározó kutatás a közeltéri hatások reprodukálása analitikus módon, egy távolabbi pontokat lefedő szett valós idejű módosításával.<sup>[6]</sup> A dolgozatomban ezeket a megoldásokat szeretném kombinálni, majd a minden esetben kimaradt hallójáratot sugárkövetés segítségével több szögből és távolságból szimulálni. Ezzel a céloom, hogy az elkészült virtuális térhangzásra képes hangmotor minél több jellegzetességét adja vissza az emberi hallásnak, azt vizsgálva, hogy ezzel mennyire javul a hangforrások lokalizációja.

## 2. HRTF halmazok előállítása

Ahhoz, hogy a HRTF renderelőnek minden egyes részét magam készíthessem el és a saját hallásomat leíró átviteli függvényeket használhassak a lehető legjobban személyre szabott hatás elérése érdekében, rögzítettem a füleim impulzusválaszait több irányból. Az elemzéseket leszámítva minden munkát ezen a halmazon végeztem el, a végleges szoftverbe is ezek az adatok kerültek előfeldolgozás után. A szoftverbeillesztéssel és a hangminőség javításával a *HRTF halmazok előfeldolgozása* fejezet foglalkozik, ez a rész a mérési rendszer összeállításától a nyers impulzusválaszok előállításáig mutatja be a folyamatot. A folyamat vége az a pont, ahol tudományos célokra szánt HRTF szettet ki lehet adni. Minthogy ez a fizikai valóságot írja le, például a fülkagyló valódi frekvenciaválaszát tartalmazza, nem alkalmazandó rajta semmiféle utólagos korrekció.

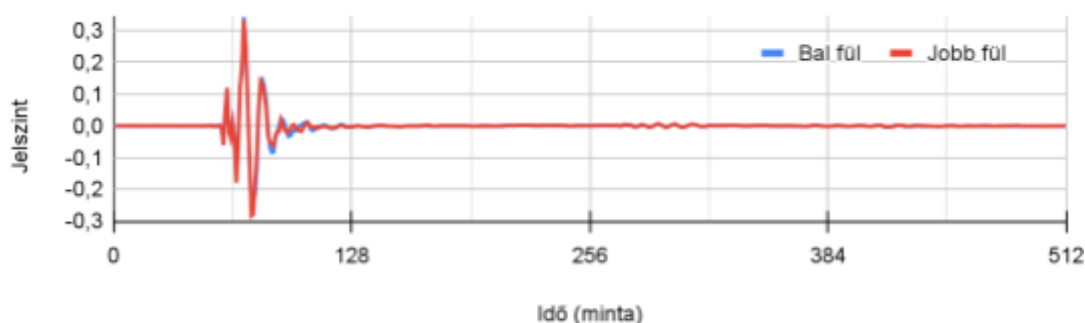
### 2.1. Mérési környezet

Legegyszerűbb esetben semmi másra nincs szükség, csak egy hangszóróra és két mikrofonra a két fülben vagy bábu esetén a hallójáratok helyén. A hangszórónak valamilyen módon meg kell kerülnie az alanyt. Kétféle automatizált körbejárás létezik: amikor a hangszórót forgatják az alany körül egy adott sugarú körön vagy gömbön, illetve amikor az alany székét forgatják. Utóbbi esetben a szerkezet olcsóbb, cserébe csak bábunál használható, embereknel szédülést, rosszullétet okozhat, illetve kizökkenheti az egyenesen tartott fejet a megfelelő mérési irányból. Az *Előkészületek* fejezetben a vizsgálataimhoz használt szettet a TU Berlin-en, a 2.1-es ábrán látható szobában készítették.



2.1. ábra: TU Berlin süketszoba a KEMAR bábu mérésekor 2011-ben.<sup>[2]</sup>





2.2. diagram: A TU Berlin KEMAR HRTF szett 1 méterről, szemből mért impulzusválasza.

A 2.1. ábrán látható, hogy a csillapításra különösen ügyeltek, még az állványok lábait is habszivacs alá temették. Minél alacsonyabb alapzajú süketszobában végzünk mérést, annál pontosabb lesz a végeredmény. Ez megfelelően magas hangerővel ellensúlyozható, de a mikrofonok érzékenysége - főleg ha kis méretűek, amelyekre élő emberek vizsgálatokor van szükség - megszab egy határt, illetve a hangszórók és erősítők kis teljesítményen produkálnak sokkal kevesebb torzítást. Ezzel az említett példa összeállítás esetében nem volt probléma, az alacsony zajszintet és az uniformitást a 0 fokos mérésük szemlélteti a 2.2. diagramon.

Az eredményen a szimmetrikus bábu két fülének minimális eltérése látszik, ez viszont olyan kis mértékű, hogy nem érdemes vele foglalkozni, valószínűleg a bábu néhány század fokos elfordulása okozta, ilyen kis változást ember nem képes hallani. A teljes mérésük 2048 minta hosszú, de már 256 után is csak visszaverődés látható, amire nincs szükség. A ténylegesen hasznos minták száma nem éri el a 64-et, mivel a kezdeti késleltetés metaadatként is elmenthető, a közvetlen impulzus lecsengése után pedig nincs szükségünk semmire, az zajnak minősül.

Ez az észrevétel egy egyszerű számítás után lényeges optimalizálásra ad lehetőséget, nem csak adattárolási szempontból, de a rögzítést is elérhetővé teszi házilag, ugyanis egy 64 minta hosszú szakasz a szett  $f_s = 44100 \text{ Hz}$ -es mintavételi frekvenciáján és  $c = 343 \frac{\text{m}}{\text{s}}$

hangsebesség esetén  $\frac{64}{f_s} \cdot c = 0.5$  méteres hullámterjedési távolságot tesz ki. Ez azt jelenti,

hogy amennyiben nem áll elő olyan felület a szobában, amelyről visszaverődve a hanghullám által megtett út kevesebb, mint fél méterrel hosszabb, mint a hangszóró-mikrofon távolság, akkor a mért eredményről a környezet hatása veszteség nélkül levágható. Egyszerűbben megfogalmazható egy kevésbé, de még így is bőségesen megengedő követelmény: a

hangszóró-mikrofon szakasz egy pontjától sem lehet semmilyen tárgy 25 cm-es távolságra. Ilyen kritérium mentén bármilyen lakószobában építhetünk mérőrendszert, én is így tettem.

A saját esetemben két hangszórót helyeztem el egymás felett, egyiket a fejemmel egy síkban, a másikat pedig úgy, hogy 45 fokos lehajlási szögben álljon a fejemhez képest. Ebben a környezetben én forogtam körbe, és az ebből fakadó pontatlanság miatt a szimmetria nem olyan mértékű, mint automatizált méréseknél, de a renderelés által nyújtott térhangzás tökéletesen érezhető, az elvárt jellemzők mind megjelentek a mérésekben.

## 2.2. Rögzítés

A mérések során átviteli függvény helyett impulzusválaszt keresünk, mivel az átviteli függvény belőle Fourier-transzformációval előállítható, a képzetes rész kihasználatlansága okán kevesebb a tárhelyigénye, és hangfájlként is könnyen kezelhető, a számítások pedig olcsóbbak rajta. Közvetlen impulzusválaszt a rendszert Dirac-deltával gerjesztve viszont nem célszerű mérni: egyetlen mintát jelent potenciálisan zajos környezetben, illetve a lejátszóeszközökben is kárt tehet.

A problémát a swept sine wave-nek, röviden sweep-nek hívott mérőjel teszi megoldhatóvá<sup>[7, 8]</sup>, mely egy frekvenciában folyamatosan növekvő szinuszhullám, két megadott frekvencia között. Ez a vizsgált frekvenciatartományt teljesen lefedi, de nem tökéletes. A Dirac-delta Fourier-transzformáltja konstans  $F(\delta(t)) = 1$  lenne, de az említett hibák miatt nem használható. Mivel a sweep jel minden pontjában más frekvenciát vizsgál, a 0 jelszint közelébe eső minták frekvenciáján pontatlanná és zajossá válik a mérés. Ez szerencsére ritka, és simítással teljesen ellensúlyozható. Mivel egy szinuszhullámról beszélünk, a sweep hossza tetszőlegesen megválasztható, a megnövekedett mérési hossz javuló zajszintet eredményez.

A generált mérőjel esetén nem lineárisan, hanem exponenciálisan nő a frekvencia, hogy a mélyebb hangok esetében pontosabb adatokkal rendelkezzen a szoftver, illetve oktávsvonként azonos energia kerüljön a gerjesztésbe. Ez a hangsúly az emberi hallás logaritmikus jellegéből adódóan előnyösebb. A generátor függvényéhez ezért először szükséges kiszámolni a frekvencia exponenciális változásának rátáját, vagyis a

$$k = \left(\frac{f_T}{f_0}\right)^{\frac{1}{T}} \quad (2.1)$$

értéket, ahol  $f_0$  a kezdeti frekvencia,  $f_T$  pedig az a frekvencia, amit a végpontig elér a sweep. A teljes jelet a következő függvény adja meg:

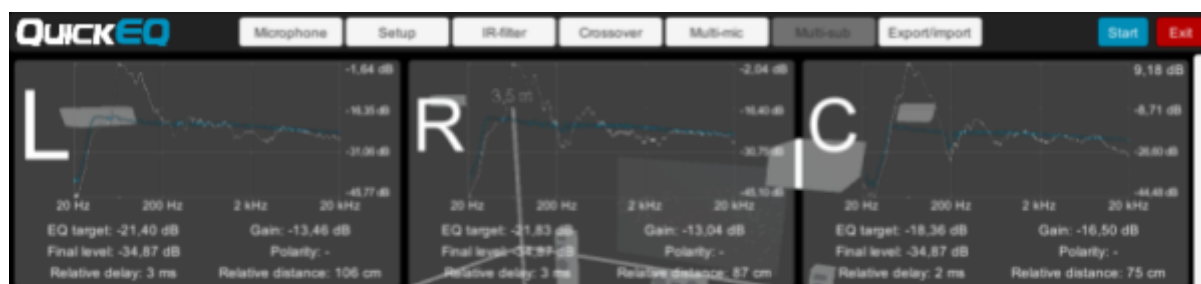
$$u(t) = \sin\left[2\pi f_0 \left(\frac{k^t - 1}{\ln(k)}\right)\right]. \quad (2.2)$$

Tetszőleges jellel való mérésből is előállítható impulzusválasz dekonvolúció segítségével. A gerjesztésválasz ( $y$ ) a gerjesztés ( $u$ ) és az impulzusválasz ( $h$ ) konvolúciója, ebből kettőt ismerünk. Fourier-térben a konvolúció szorzássá alakul, ami egy gyakori optimalizálási módszer. Ennek az inverze is működik, a konvolúció két összetevője visszakapható, amennyiben egyikük a gerjesztésválasszal egyetemben ismert, ezt hívjuk dekonvolúciónak, és egy Fourier-térbeli osztást jelent:

$$h(t) = \text{Re}\left[F^{-1}\left(\frac{F(y(t))}{F(u(t))}\right)\right]. \quad (2.3)$$

A leírtak alapján mindössze egy sweep generátorra és az azzal gerjesztett rendszer gerjesztésválaszára van szükségünk, vagyis a hangszórón kiadott sweep-et rögzítjük a mikrofonnal. Ez a feladat manuálisan is teljesíthető, de a módszer teljesen automatizálható, és felhasználási módjai sokrétűek, főleg hangrendszerek szobakorrekciójához alkalmazható. Először erre a célra hoztam létre a QuickEQ névre keresztelt szoftverem, ami az alany elhelyezését leszámítva a teljes impulzusválasz-előállítási folyamatra automatikusan képes.

A QuickEQ hangszórók kalibrálásához és egymáshoz illesztéséhez készült, de HRTF-ek előállításához is könnyen használható. Mivel egy kalibrációs szoftver, az impulzusválaszon kívül exportálható vele korrekciós szűrő, vagy akár a korrigált eredmény. Minden export csatornánként történik, ezért korrekciót nem ebben a szoftverben érdemes alkalmazni, mivel mindkét fülhöz tartozó átvitelt azonosan kell módosítani, hogy a fülek közti eltérés ne változzon. A megoldás, hogy a domináns - itt a mikrofonhoz közelebbi - fül



2.3. ábra: A QuickEQ felülete egy mérés befejeztével, bemutatva a mért és a kalibráció alkalmazása után várható eredményeket, relatív hangnyomásokat, ajánlott erősítéseket és késleltetéseket.

korrekcióját utólag alkalmazzuk a másik fül átvitelén is. Ez szintén egy konvolúciót jelent, mivel a korrekció lehetséges exportálási formátumai közül a legmegfelelőbb a FIR szűrőkhöz is használható impulzusválasz. Ennek előállításáról a *HRTF halmazok előfeldolgozása* fejezetben lesz szó, mivel a nyers adathalmaznak nem szabad utólagos javítást tartalmaznia.

Mielőtt mérne, a szoftvernek szükséges megadni a hangszórók elhelyezkedését a mikrofon pozíciójához képest. Ez a korrekciók előállításához szükséges, és az alkalmazott mikrofonkalibráció függ tőle. Egy mikrofonhoz általában két kalibrációs fájl társul, egy 0 és egy 90°-os. Ez a szög a mért hangszóró irányvektora és a kapszula normálvektora közt bezárt szöget jelenti. Sztereó rendszereknél a mikrofon vízszintesen, a hangszórók felé forgatva helyezkedik el, ekkor használják a 0°-os fájlt. Hogy térhatású hangrendszerek esetében is egységes lehessen a kalibráció, a mikrofon felfelé néz, ilyenkor hasznos a 90°-os fájl.

Valós 3D térhangzásra képes hangrendszereknél a tökéletesen mikrofon felett található hangszóró ritka, ahogy HRTF szettek előállításánál is 0 vagy 1 darab mérési pontot jelent, ezért a kalibrációs fájlok közt szükséges átfedést alkalmazni a hangszóró lehajlási szögének függvényében. Ez az információ a mért eredmények vizualizálásánál hasznos, a mikrofon hatása kiegyenlíthető az ismeretében, így realisabban vethetők össze a spektrumok. A mikrofon hibáját ugyanakkor nem korrigálok a rendereléshez szánt, végleges impulzusválaszok esetében sem. Ennek oka, hogy a nyers eredményt használjam linearizált spektrummal, ami a lejátszás számára - ahol a nyers átviteli függvény érvényes, függetlenül attól, hogy mennyire torz környezetben készült - legkevésbé destruktív megoldás.

A végleges nyers eredmények a QuickEQ által exportált, az itt definiáltak szerint számolt, korrekciótól mentes impulzusválaszok, manuálisan sztereó párokba szervezve és az időbeli eltolásokat felszámolva. Az időbeli eltolások egyrészt eleve pontatlanok, mivel az általam használt mikrofont kezelő függvénykönyvtár nem képes a hangkártyával szinkronizáltan együttműködni, másrészt az *Előkészületek* fejezetben meghatároztam az ezt leíró függvényt, amit a renderelőm utólag alkalmaz.

Az így előállt fájlok már alkalmasak virtualizált tér létrehozására, viszont a hangminőség további átalakítás nélkül még nem lesz élvezhető, hiszen tartalmazzák és a renderelt hangon alkalmazzák a fülkagyló frekvenciaválaszát. Ez azért kontraproduktív, mert fejhallgató használata esetén ez a fülkagylóval ismételten keveredik, tehát a spektrum változása négyzetessé válik. A fülhallgatókat hasonló célgörbére hangolják, hiszen életünk során azt a módosított hangképet szoktuk meg, ami a fülkagylót kikerülve elveszne. Éppen

ezért fülhallgatók esetében is érvényes, hogy az ismételt spektrumbeli változás a hangképet rontja. A megoldás az impulzusválaszok kiegyenlítése, amiről a *HRTF halmazok előfeldolgoása* fejezetben esik szó.

### 3. Előkészületek

Az előkészületek során azt vizsgáltam, hogy a hallójárat nyílásában több távolságból mért szettek milyen karakterisztikákat mutatnak. Ezek alapján arról lehet következtetést levonni, hogy vehetünk-e alapul egyetlen gömbön vett mérésalmazt, amiből késleltetések és erősítések segítségével előállítható a fülkagyló hatása különböző távolságokra. Amennyiben ez a feltételezés igaznak bizonyul, egy egyszerű HRTF renderelő is kiegészíthető hangforrásonkénti hallójárat-szimulációval, nem szükséges több távolságból méréseket végezni minden egyes alanyon.

A vizsgálatok alapjául szolgáló HRTF szett a TU Berlin-en 2011-ben a KEMAR bábu segítségével 4 különböző távolságból is elkészített szabad adathalmaz. A KEMAR-nak nincs hallójárata, a mikrofon teste a középfül helyén helyezkedik el, és a hallójárat nyílásánál vágott rész túloldalán szerelhető rá a kapszula (3.1. ábra).

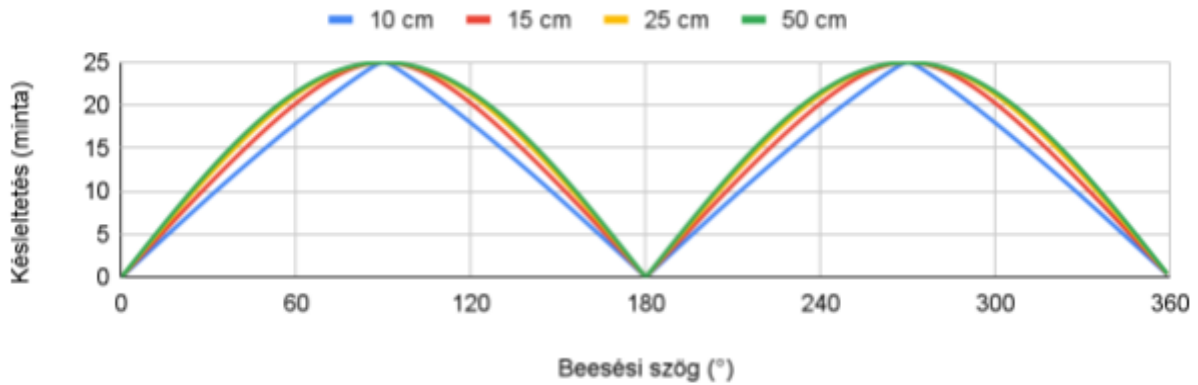
#### 3.1. Meglévő HRTF szettek vizsgálata

Egy minél szintetikusabb HRTF renderelő elkészítéséhez szükséges meghatározni a lehető legtöbb faktort, ami az ember irányhallásánál szempont. A két legegyszerűbb, triviális metrika a két fül által hallott hang közti amplitúdó- és fáziskülönbség. Ezeket a rendelkezésre álló szettekben több távolságra is ki lehet számolni. Mindkét mérőszám esetén az volt a célom, hogy előálljon egy olyan, minél egyszerűbb függvény, ami a szög- és távolságfüggés valós hatásait minél jobban megközelíti erősítéssel és késleltetéssel.

A függvények célja, hogy elég legyen egyetlen távolságból elkészíteni egy HRTF szettet, és a különböző pozíciók és távolságok között ne kizárólag átlagolással találjuk meg a köztes átviteli függvényt, illetve ne a rendelkezésre álló mérési pontok halmaza határozza



3.1. ábra: A KEMAR bábu füle és a bábuhoz illesztve.<sup>[9]</sup>



3.2. diagram: Várható késleltetések a beesési szög függvényében.

meg a renderelő korlátait. Amennyiben két, különböző távolságból, de azonos irányból vett átviteli függvény átlagát vennénk, hogy egy középük eső pontra szimuláljunk hangforrást, visszhang jelenne meg, hiszen ez valójában két, időben máshol elhelyezkedő impulzusválasz használata lenne egyszerre. Amennyiben ezt egy késleltetés helyettesíti, és az irányhallásunkért felelős karakterisztikák szimulációja is lehetségessé válik, szükségtelen lesz, hogy különböző távolságokból is végezzünk méréseket a távolságérzet átadásához.

### 3.1.1. Fülek közti késleltetés a beesési szög függvényében

Mivel a hang levegőben relatíve lassan terjed, a két fül közt könnyen mérhető időeltolás jelenik meg oldalsó beesési szögű hanghullámok esetében. A kialakuló késleltetés megállapítása nem igényel különleges mérési módot, a szabványos értékű mintavételezési frekvenciákon is akár több 10 mintás értékek mérhetőek.

Egy távolságfüggő renderelő előkészületeként nem elég csak a szögfüggését vizsgálni ennek a késleltetésnek, hanem azt is figyelembe kell venni, ha a hangforrás távolsága hozzájárul a késleltetés változásához. Először pusztán a két fül távolságát és a TU Berlin KEMAR szettjének 44.1 kHz-es mintavételezési frekvenciáját figyelembe véve a 3.2. diagramon látható előrejelzést készítettem.

A függvény a két fül távolságát veszi figyelembe, ahol a bal fül síkbeli Y tengelyen vett távolsága  $r$  sugarú körön elhelyezett hangforrásra

$$d_{YL}(\varphi) = r \cdot \sin(\varphi) - p_Y, \quad (3.1)$$

ahol  $p_Y$  a fül távolsága a fej középpontjától egy átlagos amerikai felnőtt esetében. A fejszélesség átlaga 19.4 cm, így  $p_Y = 0.097 \text{ m}$ . A jobb fülnek megfelelő távolság

$$d_{YR}(\varphi) = r \cdot \sin(\varphi) + p_Y \quad (3.2)$$

Az X tengelyen vett távolság mindkét fülnél

$$d_X(\varphi) = r \cdot \cos(\varphi), \quad (3.3)$$

így a távolságkülönbségek a

$$\Delta d(\varphi) = \left| \sqrt{d_{YL}(\varphi)^2 + d_X(\varphi)^2} - \sqrt{d_{YR}(\varphi)^2 + d_X(\varphi)^2} \right| \quad (3.4)$$

függvény szerint számolhatók. A végeredményt a  $c = 343 \frac{m}{s}$  értékű hangsebességet és az  $f_s = 44100 \text{ Hz}$  mintavételezési frekvenciát használó

$$f(\varphi) = \frac{\Delta d(\varphi)}{c} \cdot f_s \quad (3.5)$$

függvény adja meg. A függvény maximuma mindhárom távolságból

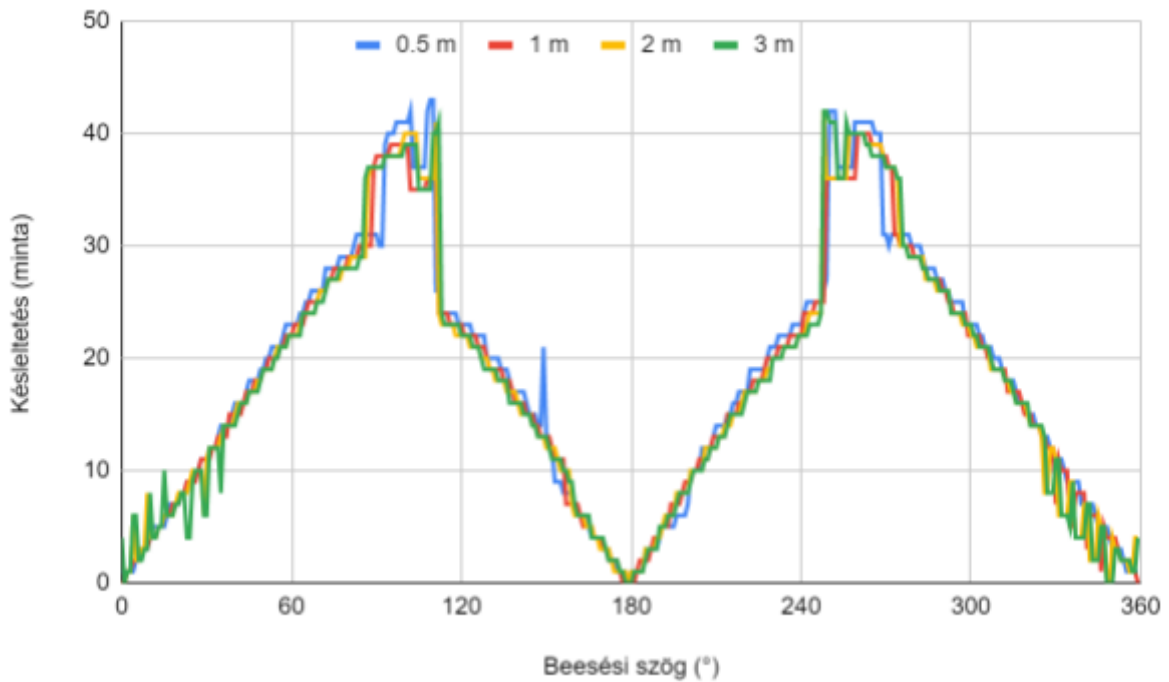
$$\Delta d\left(\frac{\pi}{2}\right) = 24.943. \quad (3.6)$$

Az eredményeken az látható, hogy a rendkívül közeli térben a késleltetés görbéje egy háromszögjelhez hasonlít, ez azonban a fej középpontjához képesti távolság, ahonnan a fül 9.7 cm-re helyezkedik el átlagosan. Mivel ez 0.3 cm-es fül-hangforrás távolságot jelent, és a fej formáját sem vettem még számításba, ez az eredmény elhagyható. A többi távolságnál már egyre pontosabban a szinuszforma jelenik meg, és bár csak a végtelenbe tartó távolság esetében tökéletes, a különbségek már 25 cm-es fejközéppont-hangforrás távolság esetében is elhanyagolhatók, így az elvárt eredmény egy távolságtól független szinuszfüggvény.

A TU Berlin KEMAR szettet négy különböző távolságból, fokenként mérték. Mivel a legkisebb távolság 0.5 m, ezért távolságtól független késleltetésekre számítottam, a  $|\sin(\varphi)| \cdot 25$  függvény szerint. Az összes vizsgált pontra vett késleltetések impulzuscsúcsok eltolásának abszolút különbségét véve a 3.3. diagramon láthatóan alakultak.

Ez az eredmény teljesen elmarad a várt formától, majdnem egy háromszögjel abszolút értékét mutatja, miközben azt csak nagyon kis távolságok esetén mutatta az előrejelzés. A 90 és 270°-os beesési szögek tartományában nagy kiugró értékek láthatók. Ezt a jelenséget a





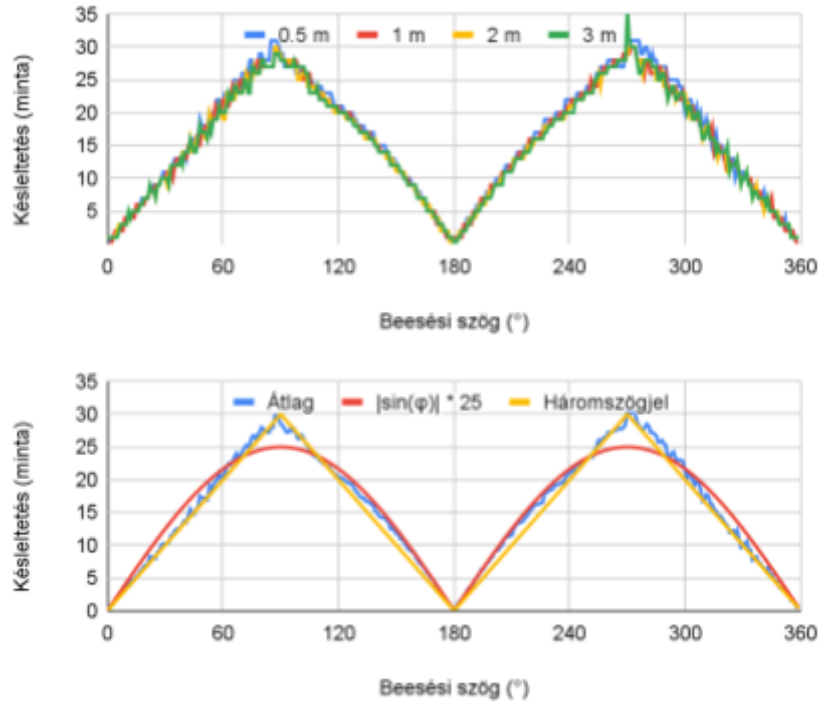
3.3. diagram: Fülek közti késleltetés a beesési szög függvényében a QU KEMAR szetten.

hangforrás körpályán való mozgatása és az emberi fej gömbszerű formája már nem indokolja. A HRTF szett mintavételezési frekvenciája 44.1 kHz, a maximális késleltetés 43 minta. Ez 0.975 ms-os késleltetésnek,  $343 \frac{m}{s}$ -os hangsebesség mellett 33.4 cm-es útnak felel meg. Egy átlagos amerikai felnőtt feje 19.4 cm széles, a kerülete pedig 57.4 cm.<sup>[10]</sup> A hivatkozott tanulmányban mért fül-fül távolságok a 25-43.5 cm tartományban helyezkedtek el. Bár az eredmény reális értékek közé esik, az előállt függvény formája eltér mind a látható megkezdett háromszögformától, mind a szinuszformától, amire számítottam.

Az impulzusválaszok megjelenítése során láthatóvá vált, hogy a kiugró késleltetést a jelszint csúcsának áthelyeződése okozta, mivel az impulzusválaszok csúcsának pozícióját



3.4. diagram: Az impulzuscsúcsokra hagyatkozó késleltetésszámítás hibájának oka.

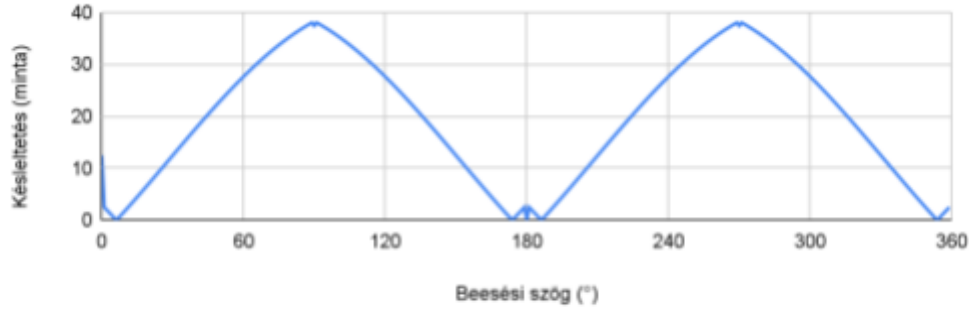


3.5. diagram: Korrigált fülek közti késleltetés a beesési szög függvényében a QU KEMAR szetten.

vettem össze. A hibát jól szemlélteti, ha egy váratlan ugrás előtti és utáni impulzusválasz releváns részeit megjelenítjük, ami a 3.4. diagramon látható.

Manuálisan összehasonlítva már nem észlelhető lényeges eltérés, de az látható, hogy a jelszint csúcserőértéke  $92^\circ$ -on még pozitív, így azt a csúcst veszi az algoritmus a késleltetésnek.  $93^\circ$ -nál ez megváltozik, már a negatív csúcs erősebb. A megoldásra az vezetett, hogy nem a csúcst, hanem az első, ahhoz képest maximum 20 dB-el kisebb pontot kerestem. Ez a módszer magában hordozza, hogy az eredmény kissé zajosabb, de a korábbi hiba megszűnik, a görbe folyamatosabb (3.5. diagram).

A korábbi kiugró értékek megszűntek az új módszertől, viszont a jelalak egyértelműen háromszög.  $114^\circ$  és  $246^\circ$  közt jobban illeszkedik a mérések átlaga a  $|\sin(\varphi)| \cdot 25$  függvényre. Ez a tartomány a fejnek azon részére esik, ahol a fülek közelebb vannak egymáshoz, a hanghullámok elhanyagolhatóan megnövekedett hosszú úton kerülnek meg a fejet. Ez az eset közelebb van ahhoz, ahogy a szinuszos alak előállt, ugyanis ott a fej megkerülését nem vettem figyelembe. Amennyiben egyetlen külső pont érintésével állítjuk elő ugyanazt a függvényt, már megjelenik a valóságban tapasztalt forma bizonyítéka a 3.6. diagramon.



3.6. diagram: Várható késleltetések a beesési szög függvényében, a fej megkerülésének figyelembevételével.

Az új várható késleltetési függvény a távolabbi fülbe tartó hanghullám útját úgy számolja, hogy a fültől egy, az X tengelyen a hangforrás felé 12.05 cm-re található pont távolságát méri előbb, majd hozzáadja a fülig tartó utat. Ez a megközelítés egy téglalapot vesz fel a fej körül, melynek szélessége és hosszúsága megegyezik az átlagos amerikai fej ugyanilyen tengelyű kiterjedéseivel. A távolabbi fülhöz tartó hanghullám a téglalap négy sarka közül a vele Y tengelyen ellentétes és X tengelyen megfelelő pontot egyenes úton érinteni fogja, kivéve, ha a beesési szög  $180^\circ$ -kal osztható.

A számításhoz bevezetett első függvény azt adja meg, hogy a távolabbi fül felé megtett úton a téglalap megfelelő sarka hol helyezkedik el az X tengelyen. Ehhez először is szükséges ismerni a megfelelő félsíkot, amit a  $hp_x(\varphi) = \text{sgn}(|\varphi - \pi| - \frac{\pi}{2})$  függvény ad meg. Ez a függvény nem felel meg annak a kitételnek, hogy csak  $180^\circ$ -kal osztható beesési szögek esetében vegyük figyelembe a fejet körülvevő téglalapot, ezért egy  $\text{sgn}(\varphi \bmod 180)$  szorzóra is szükség lesz az X tengelyi kitérés számításakor. A távolabbi fül X tengelyi kitérését meghatározó függvény az eddig definiált függvényeket felhasználva a következőképp alakul:

$$d_{Xdist}(\varphi) = r \cdot \cos(\varphi) - p_x \cdot hp_x(\varphi) \cdot \text{sgn}(\varphi \bmod 180), \quad (3.7)$$

ahol  $p_x$  az átlagos fejmélység fele, vagyis  $p_x = 0.1205 \text{ m}$ . A bal félsíkon a két fülbe érkező közvetlen hanghullámok útjának különbségét a

$$\Delta d(\varphi) = \left| \sqrt{d_{YL}(\varphi)^2 + d_X(\varphi)^2} - \left( \sqrt{d_{YR}(\varphi)^2 + d_{Xdist}(\varphi)^2} + p_x \right) \right| \quad (3.8)$$

függvény határozza meg, míg a jobb félsík esetében a  $p_x$  eltolást a bal fülre alkalmazzuk, vagyis

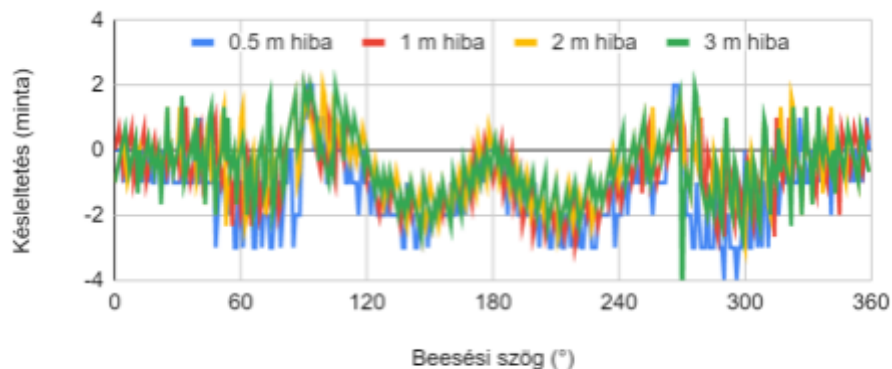
$$\Delta d(\varphi) = \left| \left( \sqrt{d_{YL}(\varphi)^2 + d_{Xdist}(\varphi)^2 + p_X} \right) - \sqrt{d_{YR}(\varphi)^2 + d_X(\varphi)^2} \right|. \quad (3.9)$$

A végső eredmény a késleltetéssel kapcsolatban, hogy háromszöggel leírható a karakterisztikája, és kizárólag a beesési szögtől függ, nem pedig a távolságtól. Ez kizárólag a közegben keletkező késleltetés, amit a HRTF szettek eredendően tartalmaznak, ismerete tömörítési lehetőségeket hordoz. Az első  $f(\varphi)$  minta az ellentétes oldalú fül impulzusválaszából elhagyható, amennyiben a konvolúció után késleltetéssel korrigáljuk. A tömörítés a memóriahasználat csökkentésén túl drasztikusan javítja definíció szerint működő konvolúciós szűrők teljesítményét, illetve FFT-alapú konvolúció használata esetén segíthet csökkenteni a szükséges ablakméretet.

A kész HRTF renderelőm ezt a tömörítési módszert alkalmazza, késleltetést a háromszögfüggvény alapján ad a másodlagos fül jelére. Hogy ez ne okozzon hibát, ennek megfelelően kellett eltávolítani az első mintákat az adathalmaz előfeldolgozása során. Bár a jeleket a csúcsertékhez képesti -20 dB-es szinttől számítom hasznosnak, az azelőtti minták alig haladják meg a -40 dB-es szintet. Ez azt jelenti, hogy az elhagyott minták miatt kevesebb, mint 1%-os hiba fordulhat elő az alkalmazott impulzusválaszban, ami az emberi hallás képességein bőven túlmutat.

Bár a minták elhagyása a spektrumban eredményezhetne nagy eltéréseket, az utólagosan ki lett egyenlítve. Erre jó indok, hogy az irányhallást a két fül közti impulzusválasz-különbség okozza, és amennyiben a két fül átviteli függvényén ugyanazt a változást eszközöljük, ugyanazt az EQ-t alkalmazzuk, ez a különbség nem változik.

A 3.7. diagramon látható, hogy a háromszögjel alkalmazása nem okoz számottevő hibát. 1 mintánál több mindössze az esetek 2%-ában veszik el a hasznosnak vélt jelből. Az



3.7. diagram: A háromszögjel késleltetésként való alkalmazásának hibája.

ilyen példák többnyire a 90 és 270°-os mérések közelében helyezkednek el, amik közül egyetlen, manuálisan kiválasztott darabot használnék egyébként is, fokenkénti részletességgel felesleges renderelni. A  $\pm 2$  mintás tartományon kívül mindössze a minták 5%-a helyezkedik el, 2-nél több levágott minta nem fordul elő, csak 2-nél több megtartott felesleges minta, de ez nem okoz problémát, mert nagyra választottam a megtartandó mintákat kiválasztó ablakot.

### 3.1.2. Fülek közti hangnyomáskülönbség a beesési szög függvényében

A késleltetés melletti legfontosabb metrika a csillapítás. Ennek az előrejelzése teljes szimuláció nélkül még nehezebb feladat, mint a késleltetésé, mivel a fej formája komplex, és a fülek sem középen helyezkednek el. A csillapítást javarészt nem a távolság, hanem a takarás okozza, ezért a távolságkülönbségből számított különbségen túl egy konstans csillapítást is várhatunk. Szerencsére ez a konstans csillapítás már a rögzített átviteli függvényben szerepel, így csak a távolságfüggést szükséges meghatározni. Mivel fix, általában 1 méteres távolságból mért szettekhez szeretnék távolságszimulációt társítani, ezért az ahhoz képest relatív csillapítást kell meghatározni a hangforrás távolsága függvényében.

Ennek az előrejelzése egyszerűbb, mint a késleltetésé, ugyanis nem a távolságok különbségére vagyunk kíváncsiak, hanem csak az arányukra. Ennek oka, hogy a mérés szempontjából domináns (hangforráshoz közelebbi) fülhöz képest számított relatív hangnyomásérték a csillapítás, és a csillapítás, amit Decibelben mérünk, egy arányszám. A

hangnyomás változása  $r$  távolságban  $\frac{1}{r}$  mértékű erősítésnek felel meg. Egy hangforrás két

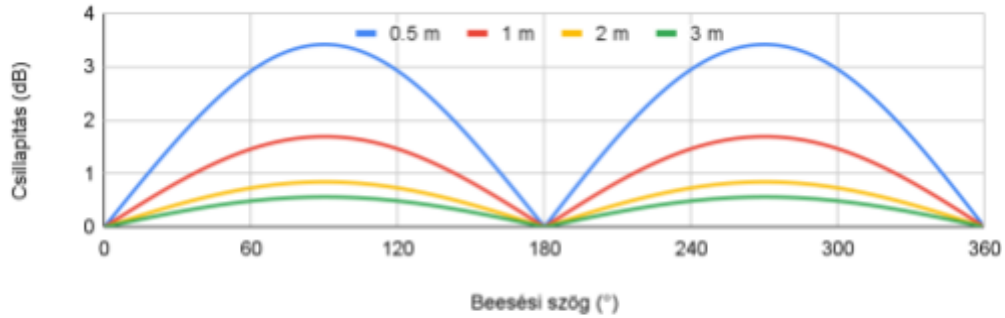
fülre vett távolságainak arányszáma  $\frac{1}{r_1} \div \frac{1}{r_2} = \frac{r_2}{r_1}$ . Mivel Decibelben számoljuk a

csillapítást az

$$x_{dB} = 20 \cdot \log_{10}\left(\frac{x}{x_0}\right) \quad (3.10)$$

képlettel, ahol az arányszám helyére illesztjük a két távolság arányát, és csak az eredmény abszolút értékére vagyunk kíváncsiak, mert rendereléskor tudjuk, hogy melyik fül a domináns, amihez képest csillapítjuk a másikat, ezért mindegy, hogy melyik fültől vett távolságot vesszük  $r_1$ -nek vagy  $r_2$ -nek. Ez a

$$\left|20 \cdot \log_{10}\left(\frac{x}{y}\right)\right| = \left|20 \cdot \log_{10}\left(\frac{y}{x}\right)\right| \quad (3.11)$$



3.8. diagram: Elvart csillapítási karakterisztika a beesési szög függvényében.

egyenletből azonosságok alapján levezethető.

Elég tehát először azt megállapítanunk, hogy mekkora a két távolság hányadosa, amit a késleltetéshez használt függvény átalakításával is megkaphatunk, ekkor a teljesen pontos

$$-A(\varphi) = \left| 20 \cdot \log_{10} \left( \frac{\sqrt{d_{YL}(\varphi)^2 + d_X(\varphi)^2}}{\sqrt{d_{YR}(\varphi)^2 + d_X(\varphi)^2}} \right) \right| \quad (3.12)$$

képlet áll elő, melynek eredménye a vizsgált HRTF szett távolságaira a 3.8. diagramon látható.

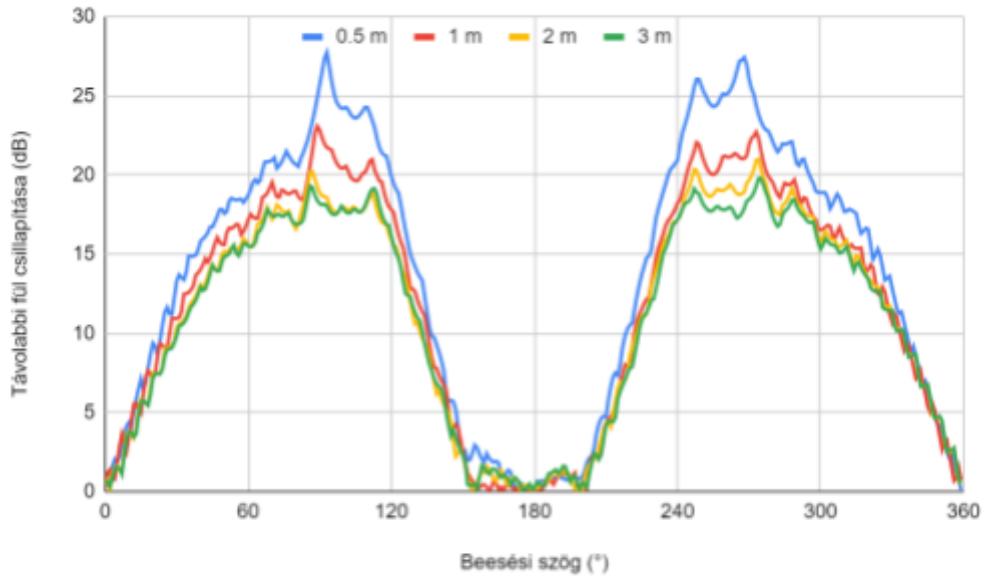
Ez a függvény annak ismeretében, hogy a végtelenbe tartó sugár esetén a függvény alakja közelít a tökéletes szinuszhoz, egyszerűsíthető. Ez a közelítés a késleltetés esetében vizsgált extrém közeltéri előrejelzéseknél is látható volt, azonban ott körpályáról nézett távolságkülönbségekről volt szó, ahol valójában azt mutatták be a szinusztól való eltérések, hogy a füleket középpontnak véve hogy torzul a körpálya, amit a hangforrás a fej körül bejár. Ebben az esetben egy egyszerűsítés nem mutatna eltérést, hiszen ha csak azt néznénk, hogy a mérés szempontjából mekkora a fülek tengelyén a távolságkülönbség, vagyis

$$d(\varphi) = \sin(\varphi) \cdot 0.194, \quad (3.13)$$

az nem függne semmilyen módon a távolságtól.

A csillapítás esetében azonban ez nem prioritás, ezért vehetjük alapul a renderelő számára is a

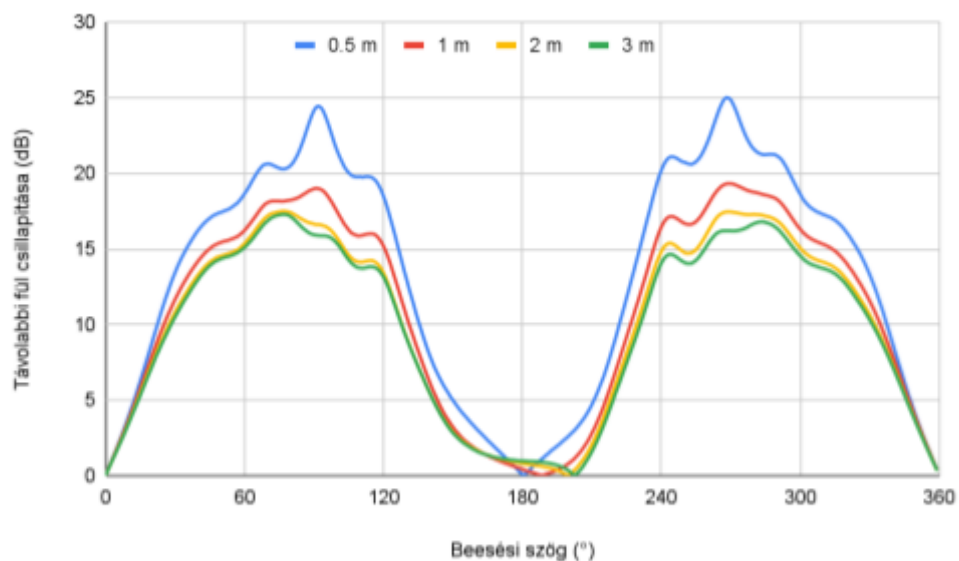
$$-A(\varphi) = \left| 20 \cdot \log_{10} \left( \frac{r+d(\varphi)+2}{r-d(\varphi)+2} \right) \right| \quad (3.14)$$



3.9. diagram: Valós csillapítási értékek, az impulzusválaszok csúcserékével számolva.

képletet, amiben nincs gyökvonás, illetve 2 helyett csak egyetlen szögfüggvényt kell számolni. Ennek a hibaaránya olyan kicsi, hogy az előrejelző diagram 180-360°-os tartományát már ez a képlet adta, és eltérés maximum az ezredes helyiértéken jelent meg.

Mivel az előrejelzés ismét figyelmen kívül hagyta, hogy a fülek között egy fej is elhelyezkedik, ezért természetes, hogy a 3.9. diagramon látható valóság nem egyezik meg vele. Ebben az esetben azonban előrelátható volt, hogy a fülek hátsó elhelyezkedése miatt a 180° környékén mért csillapítások sokkal kisebbek lesznek. Ez viszont a szintén előrejelzett konstans csillapítások miatt történik, amiket a későbbiekben ki tudunk ejteni. Fontosabb



3.10. diagram: Valós csillapítási értékek, az impulzusválaszok négyzetes közepével számolva.

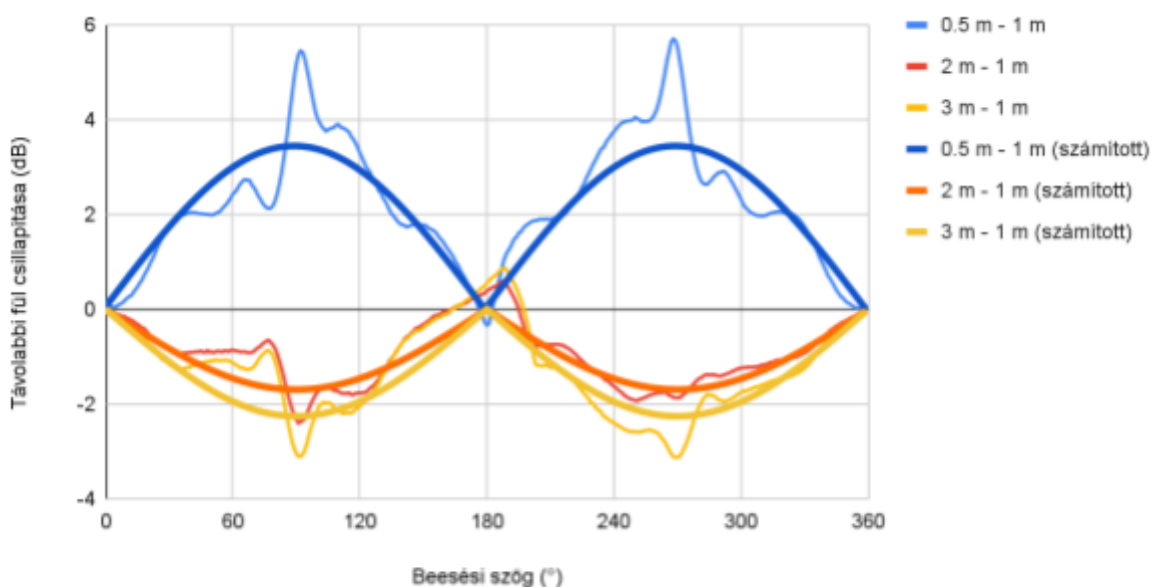
probléma, hogy az eredmény zajosnak tűnik, tehát nem építhetünk arra, hogy a csúcserkékek különbsége egy sima görbét ír le, ezért négyzetese középértékekkel újraszámoltam a csillapításokat, amik a 3.10. diagramon láthatók.

Bár a görbe szebb lett, a renderelő szempontjából nem ez a fontos, hanem az, hogy az 1 méterről mért értékekből hogyan szimulálhatjuk bármelyik más távolság csillapítását. Ez egy egyszerű kivonás, és a fej konstans fedései, amik már a mért átviteli függvényekben szerepelnek, kiesnek. Amennyiben ezt megtesszük mind a mért, mind az előre számított értékekre, utóbbinál egy kettes szorzót alkalmazva már megfelelően hasonló eredményt kapunk, amiről elmondható, hogy szimuláció híján jobb közelítést elérni nehézkes (3.11. diagram).

### 3.1.3. 3D esetek kezelése

Az előkészületek során pusztán a vízszintes síkot vizsgáltam, az alapul vett HRTF szett nem is rendelkezik magassági mérési pontokkal. A csillapítást egy egyszerűsítés során a két fül beesési szöggel súlyozott távolságára alapoztam, ez pedig garantálja, hogy 3 dimenzióba is átvihető legyen. Amennyiben a fülek által kijelölt egyenes körül forgatjuk a vízszintes síkot úgy, hogy a hangforrás irányvektora essen rá, akkor már leképezhető a kapott képletekre a számítás.

Ennek egy optimális megoldása, ha 90 fokos értékre normalizáljuk a képletet, szinusz helyett koszinusszal leírva azt, ekkor elég a térbeli szögeltéréssel számolni a fül



3.11. diagram: Mért és számított csillapítások az 1 méteres távolsághoz tartozó eredményekhez viszonyítva.



normálvektorához képest, ami már egy olcsó számítás a skaláris szorzat alapján. Ugyanez érvényes a késleltetés háromszögfüggvényére, ami, amennyiben a megfelelő oldali fülhöz viszonyítunk, egyetlen abszolút értékkel leírható lesz, hiszen csak a  $[0^\circ; 90^\circ]$  tartományban található szögekkel kell számolni.

## 4. HRTF halmazok előfeldolgozása

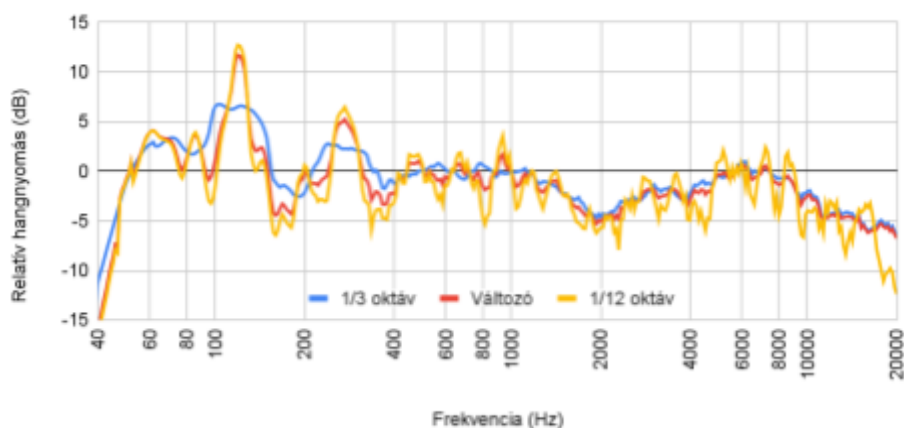
A *HRTF halmazok előállítás*a fejezetben elkészült egy olyan adathalmaz, amivel már lehetséges elkészíteni HRTF alkalmazásokat, de a hangminőség nem lenne élvezhető. A probléma a spektrum kiegyenlítésével megoldható, ezért készítettem egy szoftvert HRTF impulzusválaszokat tároló WAV fájlok korrekciójához. Az alkalmazás a QuickEQ kódján alapszik, de a bal és jobb oldali hangcsatorna együttes kezelése miatt a folyamathoz több változtatást kellett bevezetni.

Alapesetben, hangszórók korrekciójához a QuickEQ mérés után az előállt görbén a felhasználó által választott simítást alkalmaz, ami dolgozhat konstans számú oktávot átfedő ablakkal, vagy folyamatosan növekvő oktávátfedéssel. Ezeket a 4.1. diagram mutatja be. A simítás miatt kisebb mértékben kell a spektrumba nyúlni, ami azt eredményezi, hogy egységesebb marad a hangzás a szobában, nem pedig egy konkrét pont válik az egyetlen kalibrált hallgatói pozícióvá. Mivel HRTF használatakor fejhallgatóról van szó, a szűrő tökéletesen lineáris frekvenciaválaszának elérése a cél, ami hatására az nem változtatja meg a hangképet, csak irányérzetet ad hozzá. A frekvenciaválaszt az

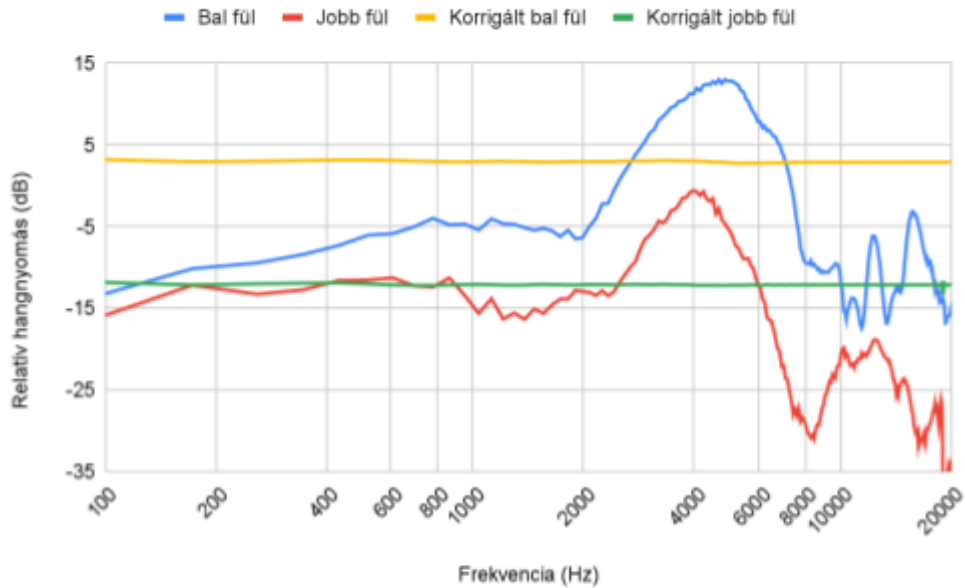
$$A(\omega) = |F[h(t)]| \quad (4.1)$$

képlet állítja elő.

Még egy fontos beállítás kiegyenlítő szűrő készítésekor a korrekciós pontok száma. Hangszórók esetében csak annyi számít, hogy legyen elég, és megfelelően pontosan tudja követni a simított görbét. Ez általában a simítás ablakméretét jelenti, például  $\frac{1}{3}$  oktávos simítás mellé  $\frac{1}{3}$  oktávonként célszerű felvenni egy pontot. Azért nem alkalmazható pontosan



4.1. diagram: Különböző simítások alkalmazása egy hangszóró frekvenciaválaszán.



4.2. diagram: A kiegyenlítés hatása, amennyiben mindkét fület egyénileg szűrjük. A vizsgált minta a TU Berlin KEMAR szettjének 45 fokos mérése 1 méterről.

a simítás, mert a túl nagy spektrumbeli leszívások korrekciója az erősítő túlzott mértékű terhelésével járna. Amennyiben a szoba geometriája lyukat eredményez a spektrumban, az adott tartomány korrekciója felesleges. Ezek miatt a QuickeEQ a +6 dB-nél nagyobb erősítésű pontokat elhagyja. HRTF szetteknel ez sem probléma, mivel a cél, hogy a spektrum minden értéke 0 dB-en álljon, ezért tetszőleges erősítéseket alkalmazhatunk. A spektrum összes pontját korrigálni kell a maximális eredmény érdekében. Mivel előfeldolgozást végzünk, nem pedig valós idejű korrekciót, a teljesítmény nem elsődleges tényező, vagyis a pontok száma tetszőleges. A generálandó EQ minden pontját úgy szabom meg, hogy az erősítés a mért relatív értékek reciproka, Decibel skálán ellentettje legyen.

#### 4.1. Kiegyenlítés

Az EQ pontosságának teszteléséhez először csak csatornánként dolgoztam fel a hangfájlokat, vagyis a bal és jobb fület különállónak tekintve korrigáltam. Az ekkor kapott, 4.2. diagramon bemutatott eredmény megfelel az elvárásoknak, előállt a szinte hibátlanul lineáris spektrum mindkét fül esetében. A szűrőt a

$$h_f(t) = F^{-1}\left(\frac{g}{A(\omega)}\right) \quad (4.2)$$

képlet állítja elő, ahol

$$g = RMS[h(t)], \quad (4.3)$$

vagyis a fül impulzusválaszának négyzetátlagáé. Ez az érték az a jelszint, amin a szűrő nem okoz hangnyomásbeli változást, ezért a spektrumban ezt a konstans értéket célszerű elérni. A konvolúció kiszámításával az

$$F[y(t)] = F[h(t) \cdot h_f(t)] = A(\omega) \cdot \frac{g}{A(\omega)} = g \quad (4.4)$$

eredmény adódik, és ez az ábrán is látszik.

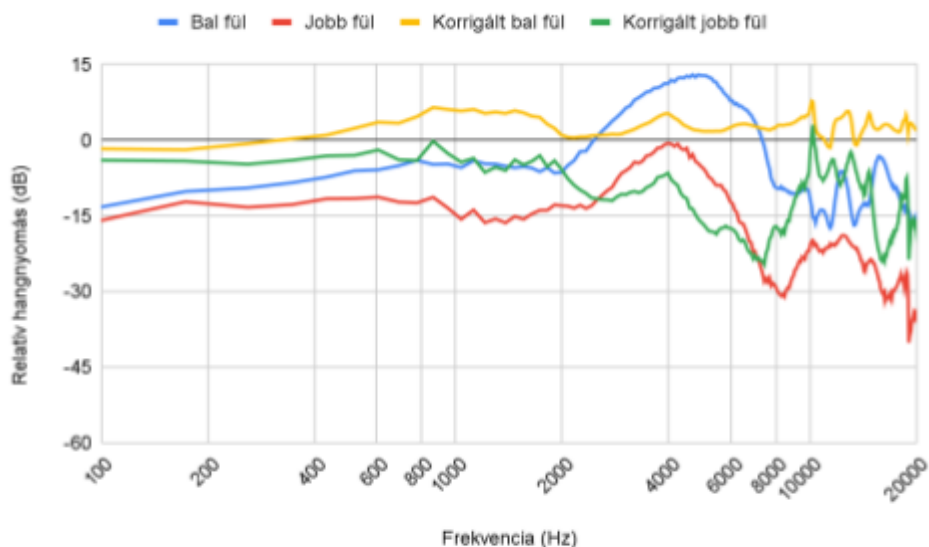
Bár elméletben a függvény értéke konstans, a késleltetés csökkentése érdekében alkalmazott fázisminimalizálás<sup>[11]</sup> okoz hibákat, amik mérhetően megjelennek. Bár az algoritmus okozhat zérusokat a spektrumban, jelen példán minimális változások jelennek csak meg: a bal fül maximális eltérése  $g$ -től 0.4 dB, a jobb oldalé pedig a 20 kHz alatt látható ponton 1.08 dB. Amennyiben csak az 1-10 kHz-es tartományt vizsgáljuk, a hibák drasztikusan lecsökkennek, rendre 0.21 és 0.07 dB-re. Ezek közül egyik érték sem számottevő, nem érdemes ezt az extra lépést kihagyni.

## 4.2. Különbségtartó kiegyenlítés

Ahogy az várható volt, a két fül közti különbséget teljesen átalakítja, ha a két fül szűrőjét egymástól függetlenül korrigáljuk. Ez a térhangzást tönkreteszi, és csak a bal-jobb irányérzet marad, ami HRTF nélkül is jelen van. A megoldás, ha a két fület ugyanúgy szűrjük. Ez történik például, ha hangszórót hallgatunk: bárhogy torzíthatjuk a jelet, hallani fogjuk, hogy melyik irányból szól, hiszen ez a torzítás a hangforráshoz tartozik, nem pedig a fülünkhöz. Ugyanígy fel lehet fogni a két fül egyszerre szűrését: ha a hangforrás jelen alkalmaznánk, az eredmény pontosan ugyanaz lenne.

Egy közös szűrő előállításához két módszert vettem fel: egyik a mérési ponthoz közelebbi fül szűrése, és az előállt szűrő alkalmazása a másik fülön is, másik pedig a két fül spektrumainak átlaga alapján készült szűrő. Azért választottam az utóbbit, mert amennyiben a közelebbi fül tökéletesen korrigált hangot hallana, a másik fülel együtt már nem állna fenn az állítás, hacsak nem tökéletesen azonos a két fül torzítása, ami nem esélyes. A szűrőt tehát úgy állítottam elő, hogy a két fület átlagoltam:

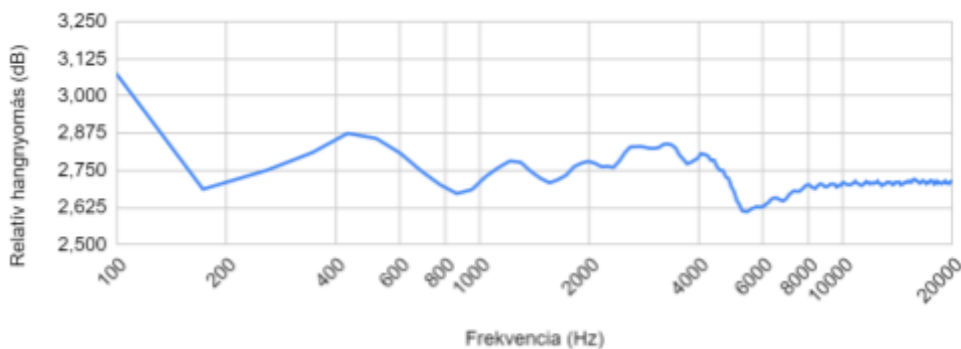
$$h_f(t) = F^{-1}\left(\frac{g_L + g_R}{A_L(\omega) + A_R(\omega)}\right). \quad (4.5)$$



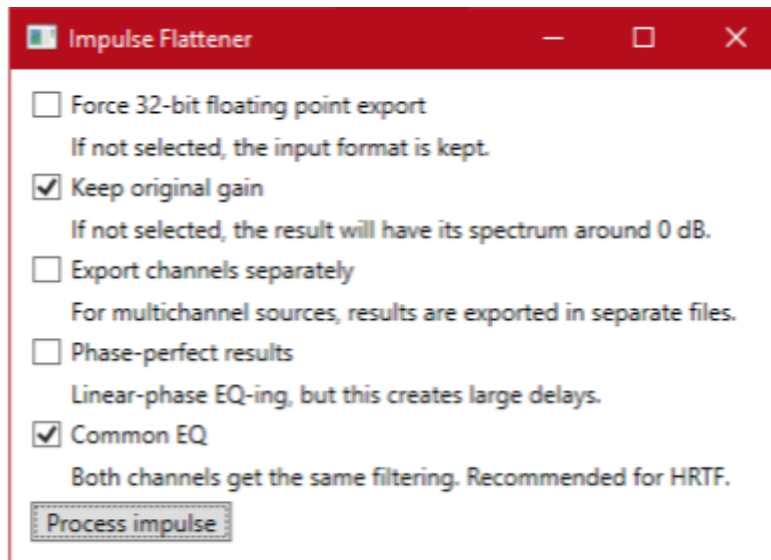
4.3. diagram: Közös szűrő alkalmazásának eredménye mindkét fülre. A vizsgált minta a TU Berlin KEMAR szettjének 45 fokos mérése 1 méterről.

Mivel a vizsgált mérésen a bal fül lényegesen hangosabb a beesési szög miatt, a közös szűrő létrejöttkor az dominált, éppen ezért a bal fül korrekció után lényegesen egyenletesebb, mint a jobb. Ez az eredmény a 4.3. diagramon látszik. Megfigyelhető, hogy mindkét fül esetében a korrekció és a kiindulási állapot különbsége egyenlő, ahogy az elvárt egy olyan szűrőtől, ami kizárólag a spektrum befolyásolására jött létre.

Bár az eredményként kapott spektrumok a kiindulási állapothoz képest közelítenek a lineárishoz, a bal fül maximális eltérése a célszinttől 9.82 dB, a jobb fülé 11.59 dB. Ezek nagyon magas értékek, és önmagukban könnyen hallható változásokat okozhatnak. Ezek a számok nem javulnak sokat a vizsgált tartomány szűkítésével sem, 1 és 10 kHz közt a bal fül esetében 6.12 dB-re csökken a hiba, a jobb fül esetében változatlan. Ha azonban a két kapott korrigált impulzusválaszt összegezzük (4.4. diagram), a spektrumuk minimális eltérést mutat



4.4. diagram: A két fül összegzett jelének spektruma.



4.5. ábra: Az impulzusválaszokat linearizáló szoftverem kezelőfelülete.

a lineáris állapottól, 200 Hz felett egy 0.25 dB-es tartományt fed le, ami alig rosszabb, mint a fűlek külön kezelésekor előállt értékek, és továbbra is elhanyagolható.

Mivel két jel konvolúciója diszkrét időben a két jel mintaszámának összegénél 1 mintával rövidebb, de a HRTF impulzusválaszokat tipikusan a gyors Fourier-transzformációval végzett konvolúciók miatt egy 2 hatványnak megfelelő mintaszámmal tárolják, az impulzusok spektrumát linearizáló szoftverem a bemeneti mintaszám kétszeresének megfelelő hosszal exportálja az eredményt. Ez a saját HRTF halmazom esetében azt jelenti, hogy a 64 mintás mérési eredmények 128 mintásra nőnek korrekció után.

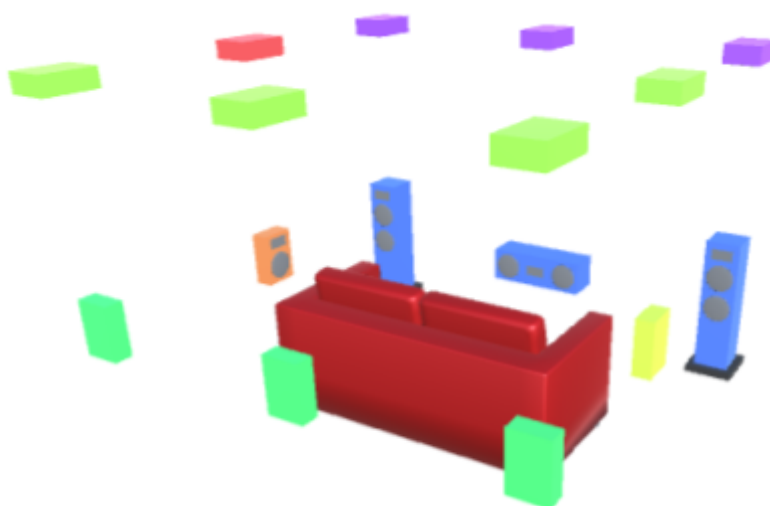
## 5. Virtuális hangtér előállítása

A HRTF renderelőt olyan módon építettem fel, hogy köztes állapotként megjelenik a hangtér virtuális hangszórókra történő leképezése, amelyeket egyesével egy HRTF halmaz mérési pontjaival feleltetek meg. Ez a köztes állapot önmagában használható 3D térhangzás előállításához hangszórók segítségével, és a hangforrások keverése is könnyű rajtuk, mivel HRTF esetében ritkán használt keverési módszereket alkalmazva már hallójárat-szimuláció nélkül is alkalmas lenne távolságérzet átadására.

### 5.1. Leképezési módszerek

Hangszórókra keverésnél 3 fő módszer csoport állhat elő: irányított, távolságalapú, és arányos.<sup>[12]</sup> A HRTF renderelők leggyakrabban egy irányított megoldást alkalmaznak, például a *Fej-átviteli függvények* fejezetben ismertetett VBAP-ot, ami egy gömbfelületre illeszti az elérhető leképezési pontokat, megkeresi a hangforrás irányvektorát közrezáró háromszöget, és azok között arányosan állítja az erősítést. Ez az irányérzet átadásához tökéletes, és jóval pontosabb lokalizáció jellemzi, mint a később ismertetett két módszert, viszont azért zártam ki, mert a gömbön belül átúszó hangokat nem képes jól kezelni, amennyiben a hangforrás irányvektora hirtelen közel ellentétes irányúvá válik.

A távolságalapú módszert legkönnyebben szintén gömbfelületre illesztett pontokkal lehet kezelni. A hangforrás pozícióját, amennyiben a hallgatótól vett távolsága meghaladja a gömb sugarát, a gömb felületére kell illeszteni. Ezt követi minden virtuális hangszóró



5.1. ábra: HRTF renderelésre kijelölt virtuális hangszórók. Minden hangszóró az alkalmazott szett egy mérési pontjának felel meg. Vízszintes síkon és felette 45 fokos lehajlási szögnek megfelelő magasságban 45 fokként vettem fel renderelési pontokat az optimális teljesítmény érdekében. Az optimális teljesítmény 1% CPU-használatot jelent egy Intel Core i5-7300HQ egy magján. Az ábrán az alkalmazott leképezési módszer szemléltetése okán helyeztem el a hangszórókat gömb helyett egy kocka felületén.

súlyozása a hangforrástól vett távolsága alapján, ez a függvény  $w_i = r - d_i$ , ahol  $r$  a gömb sugara,  $d_i$  pedig a hangforrás távolsága az adott hangszórótól. A súlyok jelentik azt az erősítést, amivel az adott hangszórókból megszólal a hangforrás jele, viszont az összegük elég magas tud lenni, ezért normalizálni kell, hogy az intenzitás - az erősítések négyzetösszege - konstans 1 maradjon, ne befolyásolja az érzett hangerőt. Ebben az esetben a normalizálás minden súly osztását jelenti a súlyok négyzetösszegének gyökével. Ezek nyomán a végső erősítés egy leképezési ponton:

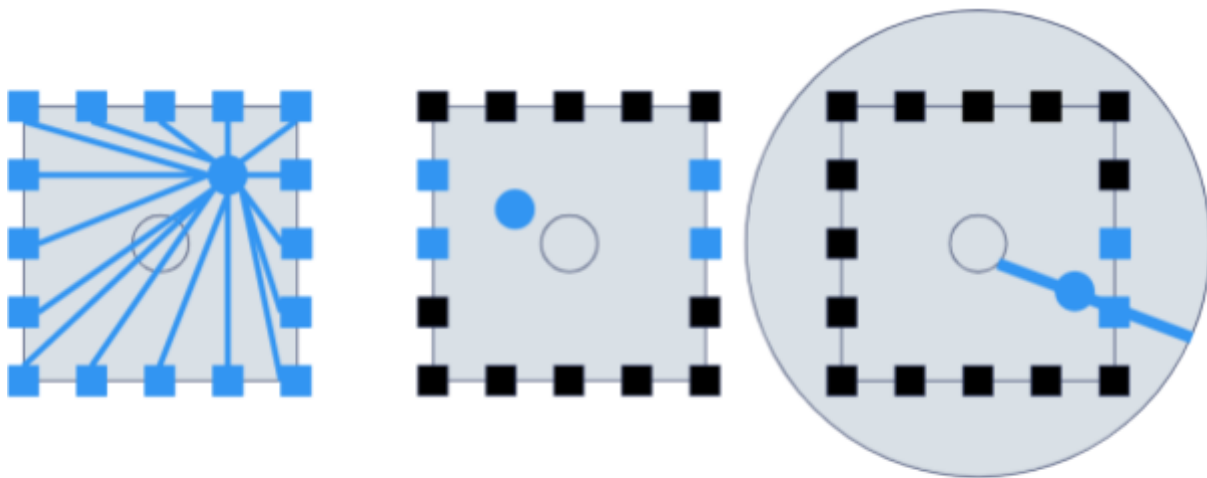
$$g_i = \frac{w_i}{\sqrt{\sum w_i^2}}. \quad (5.1)$$

Belátható, hogy

$$\sum g_i^2 = 1. \quad (5.2)$$

Ez a módszer tökéletesen képes a gömbön belül bármely pontban elhelyezni hangot, bármilyen irányba is mozogjon. Az elvetésének kisebbik oka, hogy minden hangszóróra keverni kell minden egyes hangforrást, ami nem optimális a teljesítmény szempontjából. A domináns indok, hogy a hibátlan eredményhez tökéletes késleltetésnek kell társulnia minden egyes fej-átviteli függvényhez. Mivel manuálisan dolgoztam fel az időzítési hibával rögzített méréseket, ez nem adott.

A harmadik, a renderelőben alkalmazott módszer az arányos, ami a hangszórókat egy kockára illesztve mindhárom dimenzióban megállapítja a legközelebbi lehetséges



5.2. ábra: Leképezési módszerek vizualizációja, egy szoba falára szerelt hangszórókat alapul véve. Az ábrák balról jobbra: távolságalapú, arányos, irányított.



keverési pontokat, és azok távolságának arányát veszi erősítésnek a vizsgált tengelyen. Ez 2 dimenzióban képzelhető el legkönnyebben, az 5.2. ábra is úgy mutatja be. Ekkor a hangforrástól előre-hátra és balra-jobbra tengelyen is kijelölhető a legközelebbi hangszóró, vagyis kialakul egy olyan téglalap, ami közrezárja a hangforrást. Ekkor mindkét tengelyen megállapítható, hogy arányosan hol helyezkedik el a leképezendő pont, például X tengely esetében

$$r_x = \frac{p_x}{w}, \quad (5.3)$$

ahol  $p_x$  a hangforrás pozíciója az X tengelyen, feltételezve, hogy a virtuális tér bal oldalán ez az érték 0, jobb oldalán pedig  $w$ , vagyis a tér szélessége. Hogy túlerősítés ne forduljon elő,  $r_x$  értékét a  $[0; 1]$  tartományra kell normalizálni. Ekkor a jobb oldali hangszórók súlyának egyik szorzótényezője

$$w_x = r_x, \quad (5.4)$$

bal oldalon pedig

$$w_x = 1 - r_x. \quad (5.5)$$

Ezt a számítást minden tengelyen elvégezve és az eredményeket a megfelelő hangszórók esetében összeszorozva megkapjuk azok súlyát.

A konstans intenzitás fenntartása érdekében a végső erősítés a súlyok gyöke, ez 3 dimenzióban

$$g_i = \sqrt{w_x \cdot w_y \cdot w_z}. \quad (5.6)$$

Az intenzitás akkor konstans, ha az erősítések négyzetösszege 1. Ez az erősítések képlete esetében igaz, belátható tetszőleges dimenzióra rekurzióval. A felhasznált képlet

$$f(c) = \sqrt{r_i \cdot c^2} + \sqrt{(1 - r_i) \cdot c^2} = c, \quad (5.7)$$

az első vizsgált tengely esetében  $c = 1$ . A következő tengelyen szintén előáll egy  $r_{i+1}$  és egy  $1 - r_{i+1}$  aránypár, ezekre kiszámítható

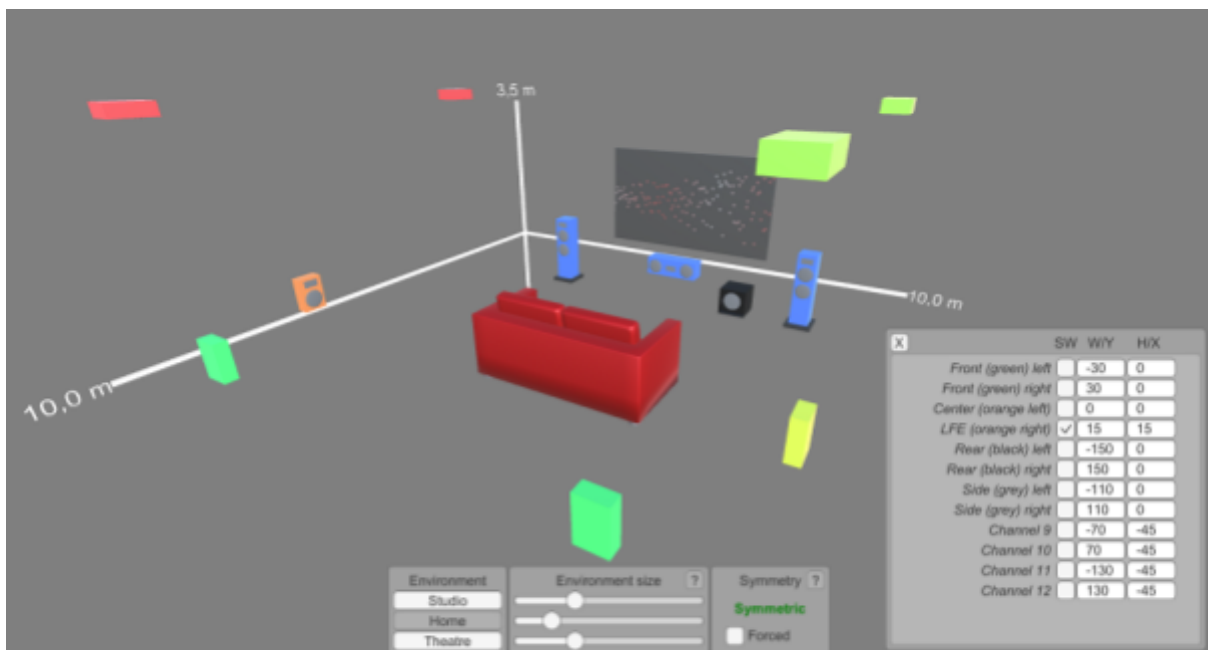
$$f(r_{i+1}) + f(1 - r_{i+1}) = 1, \quad (5.8)$$

vagyis kétszer annyi hangszórót a leírtaknak megfelelően súlyozva, a korábbi szorzótényezőket megtartva az intenzitás változatlan.

Ezt a módszert azért alkalmazzák hangszórókon és moziba szánt megoldások esetében, mert nem csak a szoba egy pontján hallható a megfelelő pozicionálás, hanem elég nagy számú hangszóróval szinte mindenhol, hiszen csak a hangrendszer egy kis része szól. Ez már úgy átvihető fejhallgatóra, hogy nincs szükség minden esetben tökéletesen tárolt és alkalmazott időzítésekre. Közeltér esetében a távolságalapú módszerhez hasonlóan nincs látványos hibája, az átfedések fül mellett is pontosak. A módszer használata egyetlen megkötéssel rendelkezik a hangszórók elhelyezésével kapcsolatban, mégpedig, hogy a Z tengelyre szimmetrikus szobát vár el. Ennek oka, hogy téglatestek zárják közre a hangforrást, ne pedig szabad alakzatok. Különböző vízszintes síkok elrendezése lehet más, és mozik esetében költségoptimalizálás érdekében így is tesznek, viszont virtuális térben nem érdemes ilyen könnyítést alkalmazni. Mivel egy hangforrás megszólaltatásához maximum 8 hangszóró szükséges, a módszer teljesítménye jobb, mint a távolságalapúnak, de rosszabb, mint az irányítottnak.

## 5.2. Konfiguráció

Bár az arányos módszert alkalmazom, a renderelőmet azzal a céllal készítettem, hogy tetszőleges hangszóróelrendezést támogasson. A grafikus felületen fogd-és-vidd módszerrel



5.3. ábra: Hangszóróelrendezés konfigurációja a renderelő szoftverben.

térben áthelyezhető egy hangszóró, vagy manuálisan megadhatók az irányvektorának szögei az Y és X tengely körül. Hangszórókat a szimmetria támogatása érdekében páronként enged hozzáadni, ezért páratlan elrendezés úgy jöhet létre, ha az egyik kimenet az LFE (Low Frequency Effects), vagyis a mélynyomó hangsáv kiadására van állítva. Ez egy közös hangsáv, minden LFE-nek jelölt kimeneten ugyanaz jelenik meg. Alapértelmezetten minden hangforrás 120 Hz alatti részét megszólaltatja, ez a szűrés kikapcsolható.

A felület visszajelzést ad, hogy a beállított elrendezés szimmetrikus-e. Amennyiben igen, a zöld “Szimmetrikus” (Symmetric) felirat ad visszajelzést, hogy az arányos keverési módszer alkalmazásának feltétele teljesül, és ezért azt is fogja használni. Hogy a szimmetria módosítások után is garantáltan megmaradjon, az “Erőltetett” (Forced) opciót bejelölve a fogd-és-vidd módszerrel mozgatott hangszórók a párjukon is alkalmazzák a módosítást ellentétes irányban, illetve a manuálisan megadott szögeket szintén párokon alkalmazzák.

Amennyiben a használt elrendezés bármely oknál fogva mégsem szimmetrikus, a kimenet nem némul el, hanem megváltozik az alkalmazott keverőmotor, mégpedig az irányított és távolságalapú módszerek közt egy középútra, ami mind fejhallgatón, mind hangszórón elfogadható kompromisszumot nyújt. Az aszimmetrikus renderelő használatát egy vörös “Aszimmetrikus” (Asymmetric) felirat jelzi, az alkalmazott módszere pedig a

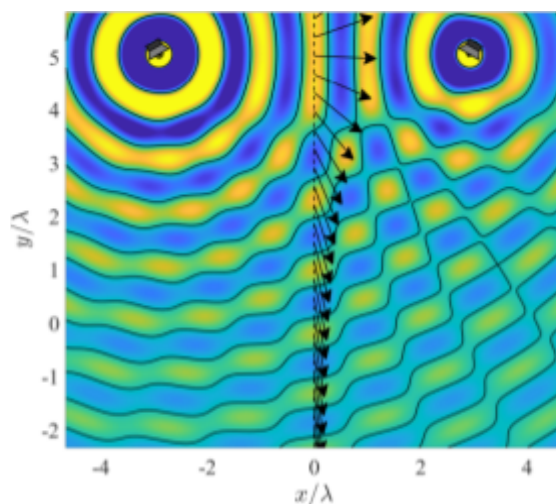
$$w_i = (\pi - \varphi_i)^8 \quad (5.9)$$

súlyfüggvénnyel módosított és 3 legközelebbi csatornára korlátozott átalakítása a távolságalapú megoldásnak. A súlyfüggvényben  $\varphi_i$  a hangforrás és a hangszóró irányvektora által bezárt szög.

A konfiguráló felület lehetővé teszi a környezet méretének megadását, ami egy téglatest oldalhosszainak beállítása. Ennek kizárólag a szimmetrikus renderelőnél van hatása, hiszen aszimmetrikus esetben csak a bezárt szögtől függenek a lekeverési erősítések. Mivel a szimmetrikus keverő tengelyenkénti értékeinek a környezet megfelelő oldalhossza osztója, ezért a környezet mérete valójában azt határozza meg, hogy a virtuális térben mi az a kijelölt térfogat, amin belül a hangok mozgása a szobán belüli átfedést eredményez.

### 5.3. Áttérés fejhallgató-virtualizációra

Az arányos módszer alkalmasságát a valós térben hullámtér-szintézis (5.4. ábra) képes igazolni. A sztereó párok, amik ebben az esetben a kezelt dimenziók, úgy változtatják meg a



5.4. ábra: Két hangforrást megszólaltató VBAP hullámtere, ha a források közt kizárólag a hangnyomás különbözik. Látható a hullámfront módosult fázisa a várható hallgatói pozíciókban.<sup>[4]</sup>

hullámfrontot, mintha a hangforrás valójában a két hangszóró közt lenne. Ebből következik, hogy a hallgatóhoz már egy olyan hanghullám érkezik, mintha a renderelt pozícióból valóban hang szólalt volna meg.

Amennyiben rendelkezésünkre áll egy olyan HRTF szett, ahol nem gömbfelületen, hanem egy kocka pontjain, például egy 3D térhangzásra képes moziterem segítségével készültek a mérések, a hangszórókhöz szánt renderelő átalakítása HRTF renderelővé pusztán a mért átviteli függvények alkalmazását jelenti. Ekkor minden egyes hangszóró utófeldolgozása úgy történik, hogy a fülhallgató mindkét oldalára a hangszóró kimenetét a megfelelő oldali fej-átviteli függvénnyel konvolúciót képezve keveri a renderelő. Ekkor minden hangszóró egy elkülönülten hallható térbeli pont fejhallgatón keresztül, és a virtuális térhangzás összeállt.

A gömbön felvett HRTF szettek apró problémát okoznak, de ez könnyen kezelhető. Mivel az arányos keverő egy kocka felületén működik, de a renderelhető pontok egy gömbön helyezkednek el, elég úgy torzítani a teret, hogy a gömb szerinti pozíciók a kockához képest arányosan ugyanott legyenek, ekkor a renderelés úgy viselkedik, mintha gömbre lennének felvéve a hangszórók, és emellett is működik az arányos algoritmus. Ezt úgy implementáltam, hogy az irányvektort szoroztam annak kocka- és gömbfelületre normalizált verzióinak arányával. Kockafelületre normalizálás a maximális oldalhosszal történő osztást jelent, hogy a legnagyobb komponens hossza 1-re változzon, ezzel elhelyezve a pontot egy 2 egység oldalhosszúságú kocka felületén, vagyis

$$v_{nc} = \frac{v}{\max(|v_x|, |v_y|, |v_z|)}, \quad (5.10)$$

gömbre pedig a következőképp tudunk normalizálni:

$$v_{ns} = \frac{v}{|v|}. \quad (5.11)$$

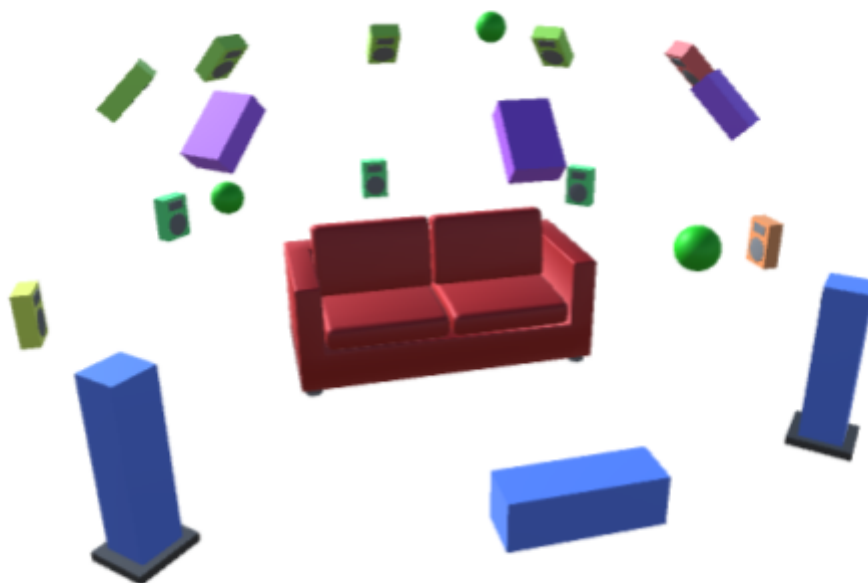
A végleges irányvektor, amivel a HRTF renderelőnek dolgoznia kell, az

$$v' = v \cdot \frac{v_{nc}}{v_{ns}} = v \cdot \frac{|v|}{\max(|v_x|, |v_y|, |v_z|)}, \quad (5.12)$$

így már a renderelt pontok úgy viselkednek, mintha egységnyi távolságra lennének a hallgatótól. Ezt a jelenséget a szoftver a hangszórók gömbön elhelyezésével is szemlélteti (5.5. ábra).

#### 5.4. További lehetőségek hangforrásokkal

Az elkészült hangmotor nem csak hangforrások kezelését teszi lehetővé. A hallgató ugyanúgy objektumként mozgatható, pozícióval és tekinteti elfordulással is rendelkezik. Minden szereplőhöz készítettem Unity wrappert, vagyis a teljes függvénykönyvtár használható a játékmotorban, pontosan ugyanolyan módon, mint a motor gyári renderelője. A két hangmotor akár együtt is használható, például az egyszerű monó (pl. narrátor) vagy sztereó (pl. zene) sávok annak segítségével jobb teljesítménnyel keverhetők.



5.5. ábra: A HRTF renderelő végleges állapotában, 3 virtuális hangforrással. A demóalkalmazásban tetszőleges számú hangforrás hozzáadható, és billentyűzettel, egérrel, vagy controllerrel különböző alakzatokban vagy szabadon mozgathatók.

A hangforrások hallgatókhoz tartoznak, így többféle tér is virtualizálható egyszerre. A környezeti beállítások, mint a hangszórók helye és a szoba mérete szintén a hallgató beállításai. Egy hangforrás alapbeállításai a használt hangfájl, lejátszási állapot, ismétlődés engedélyezése, a némítás, hangerő, illetve lejátszási sebesség, ami hangmagasságot is változtatja, nem korrigálja.

Egy hangforrás 1D és 3D térben is kezelhető, vagyis a bal-jobb oldal között állítható arányban vagy térben pontosan renderelve. A két mód közt átfedéssel átmenetet lehet képezni, a módok keverésének hangerejének aránya állítható. Hangszórókat használó környezetekben a távolságot a hangerő csökkenése adja vissza. Ehhez választható négyféle függvény:

- a fizikailag helyes  $\frac{1}{d}$ , ahol  $d$  a hangforrás és a hallgató távolsága,
- a kisebb különbségeket okozó  $\frac{1}{1+\ln(d)}$ ,
- a lineáris  $\frac{r-d}{r}$ , ahol  $r$  a maximális renderelési távolság, ez szintén állítható,
- konstans 1 a távolságátadás kikapcsolására.

A távolságfüggvények a lineáris kivételével 1 egység távolság alatt konstans 1 értéket vesznek fel, hogy a közeltéren ne lépjen fel túlerősítés. Állítható még az objektum kiterjedése, ami annyit tesz, hogy az összes hangszóróból megszólal valamilyen hangerővel, míg maximális kiterjedés esetében a terem minden hangszóróján egyenlő jelerősséggel szól.

## 5.5. Mozgó hangforrások kezelése

Virtuális környezetekben nem feltétlenül konstans a hangforrások relatív pozíciója, ezért foglalkozni kell azzal az esettel, ha a pozíciók megváltoznak. A hangmotorommal ez kétféle módon kezelhető, az első az úsztatás, ami erősen kontrollált környezetekben kezelhető. Ha ismert a renderelt időszáv elején és végén minden objektum pozíciója, akkor megfelelő megoldás mindkét pozícióban a részlet renderelése, és az elejétől a végéig átúszni a két eredmény közt, ekkor a renderelt részlet alatt a hangforrás pozíciót vált.

Az átfedéseket konstans intenzitású módszerrel érdemes kezelni, amikor a kezdeti állapot amplitúdója az idő szerint

$$A(t) = \frac{t}{T}, \quad (5.13)$$

a végállapoté pedig

$$A_T(t) = \frac{T-t}{T}. \quad (5.14)$$

Ekkor, ha a kezdeti pozíció kapott renderelés függvénye  $f(t)$ , a végpozíció pedig  $f_T(t)$ , a végleges keverési függvény

$$\frac{f(t) \cdot t + f_T(t) \cdot (1-t)}{T}. \quad (5.15)$$

Mivel a legtöbb szoftver esetén a hangképzés külön szálon fut konstans időközzel, ezért néhány folytonos mozgás esetében előfordulhat, hogy több audio szeletet is renderelni kell, mire a pozíció frissül. Ezt az oldja meg tökéletesen, ha a beláthatóan folytonos mozgások külön szálon frissülnek, de realisabb az audio keretek megfelelően rövidre választása. Az alapbeállítást 240 mintán szabtam meg, ami a gyakran használt 48 kHz-es mintavételezési frekvencián másodpercenként 200 frissítést jelent. Ez kellően gyorsan képes lekövetni a virtuális világ változásait, és a teljesítményre sincs nagy hatással: egy gyengének mondható Intel Core i5-7300HQ processzoron akár 1000 hangforrás 12 csatornán valós időben renderelhető. A teljesítményigény az ablakméret csökkenésével drasztikusan nő, és bár a szoftver képes 1 mintánként is újraszámolni a virtuális tér állapotát, ezt valós időben nem érdemes megtenni.

A hangforrások beállítása közt megjelenik a Doppler-effektus mértéke, mely ideális környezetben a frekvenciának egy  $\frac{c-v}{c}$ -s szorzója lenne. A  $v$  sebesség az objektum elmozdulásának deriváltja, diszkrét időben

$$v = |r[t] - r[t - 1]|, \quad (5.16)$$

azonban az előző bekezdésben leírt esetről, vagyis ha két képfrissítés között több audio keret készül el, akkor a derivált értéke, így a sebesség is 0, vagyis nem lenne számolható a Doppler-effektus. Hogy minimális legyen a hiba, de közel valós időben kövesse a mozgást az effektus, a hangmotor a kiszámolt sebességek mozgóátlagát veszi. Az átlagolás mértékét az effektus beállított szintje határozza meg.

A konkrét implementációban a Doppler-effektushoz meghatározott szorzó a lejátszási sebességet befolyásolja. Ez kihat a hangmagasságra is, pontosan olyan mértékben változik a

frekvencia is, hiszen  $n$ -szeresére növekvő sebesség esetén, vagyis a hullámforma  $\frac{1}{n}$  méretűre csökkenésekor a hullámhossz is így változik:

$$\lambda = \lambda_0 \cdot \frac{c-v}{c}, \quad (5.17)$$

ami az effektus képlete. Hogy ne csak minták helyeződjenek át és nagymértékű torzítások lépjenek fel, a hullámforma újraszámításához a Catmull-Rom spline-t választottam. A választás oka, hogy folytonos, éles törésektől mentes eredményt állít elő, és az összes megadott pontot, vagyis a kiindulási mintákat érinti.



## 6. Irányhallás szimulációja

A HRTF szimuláció utolsó eleme a távolságképző szűrő. Ez hangforrásonként mindkét fülhöz érkező jelet kiegészíti a korábban említett, de nem implementált jelenségekkel. Alkalmazza a hallójáráthoz tartozó átviteli függvényeket, illetve a távolságtól függő hangerő-csillapítást, mindössze ez a két jelenség nem oldható meg a HRTF szettel vagy egy hangszóró-renderelő segítségével.

A csillapítás mértékét az *Előkészületek* fejezetben az

$$f(\varphi, r) = \left| 20 \cdot \log_{10} \left( \frac{r+d(\varphi)+2}{r-d(\varphi)+2} \right) \right| \quad (6.1)$$

képlet határozta meg, ahol

$$d(\varphi) = \sin(\varphi) \cdot 0.194. \quad (6.2)$$

Mivel a valóságban mértekre a függvény kétszerese illeszkedett, ezért a 20-as szorzó 40-re nőtt, illetve kezdeti optimalizálásként a 2-es osztók is áthelyezhetők a  $d(\varphi)$  függvénybe. A szűrő által alkalmazandó képlet a hangerőkülönbség meghatározására

$$f_r(\varphi, r) = f(\varphi, r) - f(\varphi, 1) \quad (6.3)$$

lenne, ami átrendezhető minimális számú és a CPU számára könnyebb műveletekre, így a használt függvény végső iterációja

$$f_r(\varphi, r) = \left| 40 \cdot \log_{10} \left( \frac{[r+d(\varphi)] \cdot [1-d(\varphi)]}{[r-d(\varphi)] \cdot [1+d(\varphi)]} \right) \right|. \quad (6.4)$$

Mivel a HRTF renderelést már a hallgatóhoz tartozó objektum végzi, és az összes hangforrás hangszórókra történő keverése után alkalmazza a hangszórókon az átviteli függvényeket, ebben a leképezési sorrendben kis teljesítményigényű megoldás a távolságképző beiktatására nem létezhet. A két lehetséges módosítási pont a hangforrásból elinduló jel és a teljes környezetre előálló lekevert hangkép, ami egy hangsávot jelent minden egyes hangszóróhoz. A lejátszott hang módosítása hiábavaló, az nem befolyásolja a hallott irányt, mivel nem jut belőle különböző feldolgozás bal és jobb fülre, ami jelen esetben elvárás.

Marad tehát az a módosítás, hogy a bal és jobb fülnek szánt szűrt jelet még a csatornákra keverés előtt közvetlenül elkészíti a renderelő, és azt keveri a megfelelő oldali

hangszórókra. Így az extra teljesítményigény objektumonként mindössze két konvolúció, aminél nem is lehetne kevesebb. Ez a mondat úgy tűnhet, hogy sérti az előző bekezdésben leírt “kis teljesítményigényű megoldás [...] nem létezhet” állítást, de ez nem igaz, ugyanis nem vehető tökéletes megoldásnak, csak közelítésnek.

Nem közvetlenül a fülekre alkalmazza a megfelelő impulzusválaszokat, mert ez tényleg nem illeszthető be helyesen a jelútba, csak a hangszórókra. Ahhoz, hogy a végső sztereó jelre kerüljön rá a hallójárat kiegészítést, a HRTF renderelés után kellene egyetlen, a környezetben kezelt hangforrás szűrőjét alkalmazni. Ez egy olyan durva elvárás, ami miatt minden egyes kezelt hangot külön kellene a teljes jelfeldolgozási folyamaton végigvinni, majd a legvégén összekeverni. Alkalmazott környezetben, például egy VR-alkalmazásnál, ahol a hangzásnak minimális CPU-időt érdemes szánni, nem fér bele, hogy minden egyes újabb hangforrás jelentős erőforrásigény-növekedéssel járjon. Egy demó vagy validálás okán járható út, de a legtöbb esetben nem ajánlott.

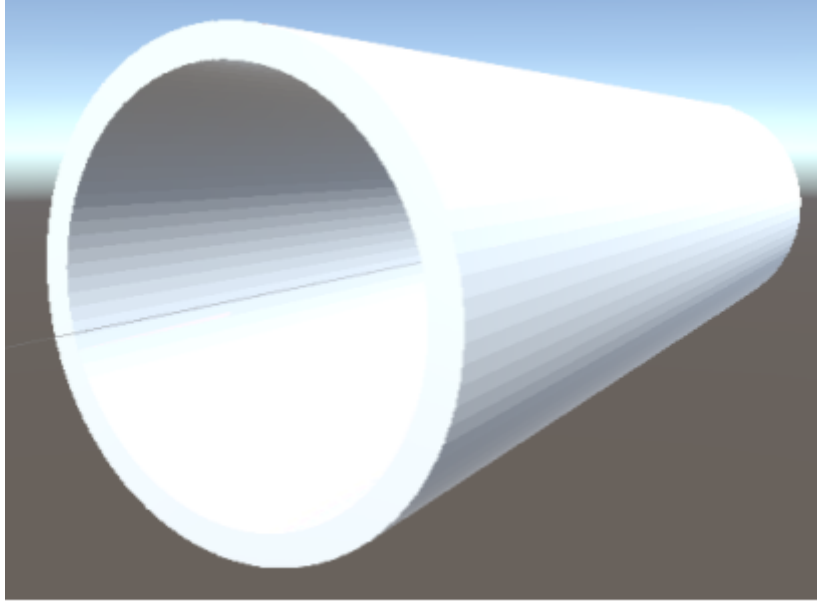
### 6.1. Hallójárat-szimuláció

A hallójárat torzításának szimulációja sugárkövetéssel történt, geometriai visszaverődéseket alkalmazva. Mivel ebben az esetben a sugarakat ellenkező irányból elindítva is ugyanazt az utat járják be, a dobhártyának megfeleltetett pont bocsátja ki őket, nem pedig a hangforrás. Ettől lehetségessé válik egy olyan optimalizáció, hogy az összes hangforrás számára elindulnak a sugarak, és annál a hangforrásnál történik meg a kezelése, amelyikbe becsapódnak. Ilyen módszerrel az algoritmus lépésszáma független a sugárkövetés célpontjainak, vagyis a szimulálandó hangforrási pozíciók számától.

A szimuláció környezete mindössze a hangforrásokból és az egyik fül hallójáratának megfeleltetett hengerből áll. Az átlagos hallójárat hossza  $l = 2.5 \text{ cm}$ , átmérője  $d = 0.7 \text{ cm}$  <sup>[13]</sup>, ezek lettek a henger dimenziói is. A hengerfal vastagsága  $x = 0.5 \text{ cm}$ , ami funkcionális célt nem szolgál, és az átmérőn felül értendő. Sugárkövetéshez a Unity játékmotort használtam, ahol az adott pontból induló sugár betalálásának vizsgálata és a pont normálvektorának meghatározása adott, ezekből a geometriai visszaverődés már kiszámítható, amihez a Unity a

$$v' = (-2 \cdot n \cdot v) \cdot v + n \tag{6.5}$$

képletet nyújtja.



6.1. ábra: A generált hengermodell, ami a hallójárat egyszerűsítése.

A henger nem egy modell, hanem elnyújtott kockákból generáltam. A két megoldás eredménye azonos, kockákkal viszont könnyebb dolgozni, és futás közben a szerkesztőből manipulálhatók. A kört, ami mentén a kezdetben egységnyi oldalhosszú kockákat elhelyeztem,  $n$  darab osztópontra a

$$p_i = \left( \sin\left(2\pi \cdot \frac{i}{n}\right) \cdot \left(1 + \frac{x}{2}\right) \cdot \frac{d}{2} \quad \cos\left(2\pi \cdot \frac{i}{n}\right) \cdot \left(1 + \frac{x}{2}\right) \cdot \frac{d}{2} \quad \frac{l}{2} \right) \quad (6.6)$$

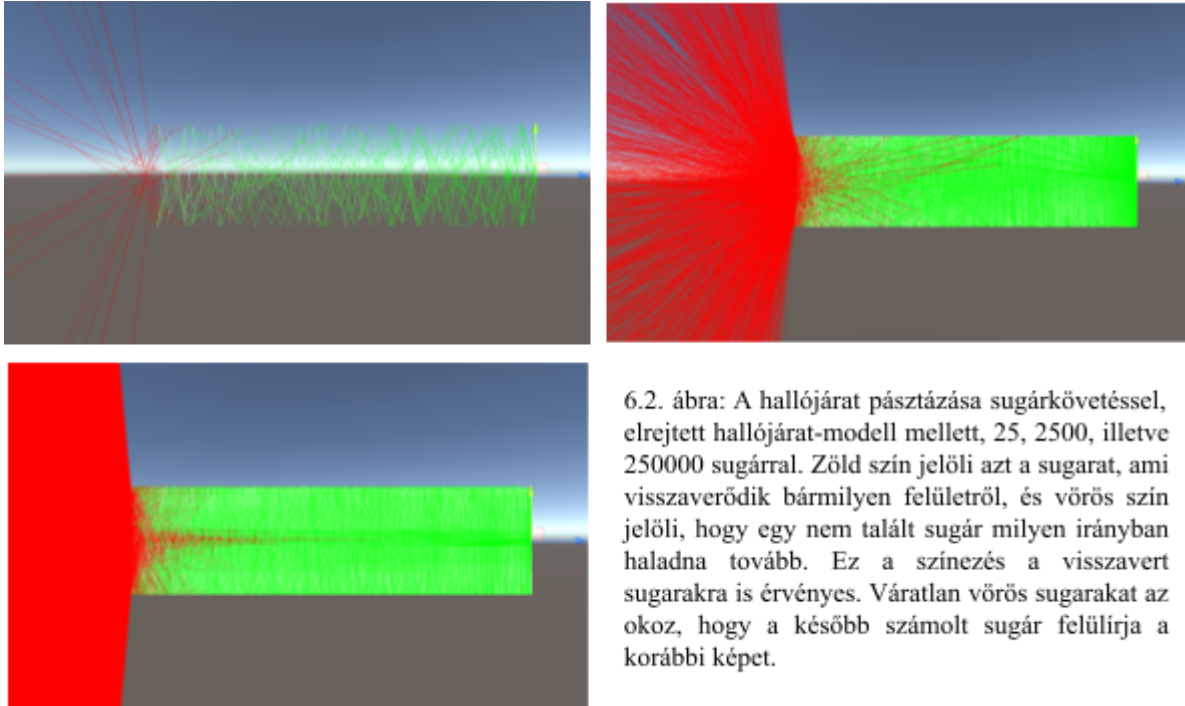
vektorok határozzák meg. Az elforgatásuk  $z$  tengelyen radiánban

$$r_z = \frac{-2\pi \cdot i}{n}, \quad (6.7)$$

a méretüket skálázó konstans vektor pedig

$$s_x = \left( \frac{2\pi}{n} \cdot \frac{d}{2} + \frac{x}{2} \quad x \quad l \right). \quad (6.8)$$

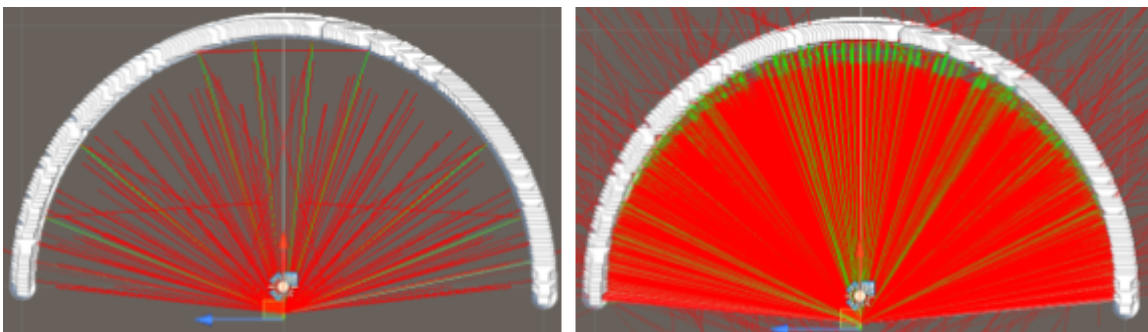
A hangforrásokat szimbolizáló pontokat 1 cm átmérőjű gömbökként vettem fel. Ennek oka, hogy a sugarak számára egy olyan elfogadási tartományt határozzak meg, amit ha érintenek, találatnak minősülnek. Amennyiben egy sugár eltalál egy hangforrást, a találati pont megváltozik a hangforrás helyére, hogy pontosan a dobhártya-hangforrás távolság álljon elő, vagyis pontszerűen viselkedjen.



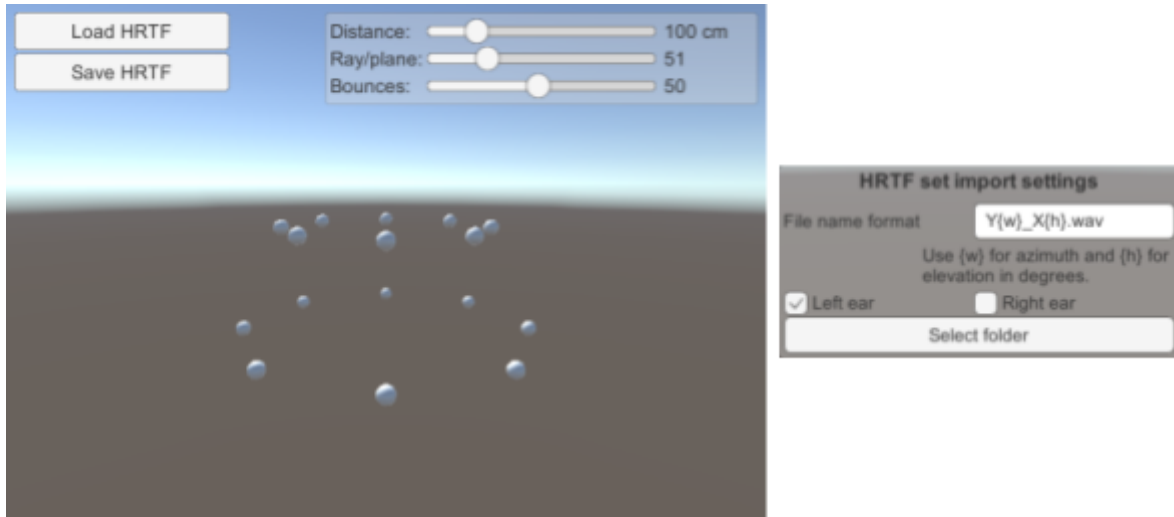
6.2. ábra: A hallójárat pásztázása sugárkövetéssel, elrejtett hallójárat-modell mellett, 25, 2500, illetve 250000 sugárral. Zöld szín jelöli azt a sugarat, ami visszaverődik bármilyen felületről, és vörös szín jelöli, hogy egy nem talált sugár milyen irányban haladna tovább. Ez a színezés a visszavert sugarakra is érvényes. Váratlan vörös sugarakat az okoz, hogy a később számolt sugár felülírja a korábbi képet.

A henger egyik végét a dobhártya mellett közvetlenül egy lappal lezárom, ez nem engedi a sugarakat kilépni a hallójárat nyílásával ellentétes irányba. A sugarakat függőleges és vízszintes tengelyen megadott felosztás szerint indítom el, ezért a kibocsátott sugarak száma minden esetben a felosztás négyzete. A 6.2. ábrán látható, hogy az 500-as érték már a teljes lehetséges térrészt kitölti a képernyőn, viszont nem elég alapos. Későbbi tesztek igazolják, hogy alapos szimulációt a 3000-es érték, vagyis 9 millió sugár eredményez.

Amennyiben a 6.3. ábrán látható módon egy hangforrást eltalál egy sugár, megkezdődik az impulzusválasz előállítás. Ennek kiinduló állapota egy 2048 minta hosszú, 0-ra inicializált tömb. A beérkező sugarak az általuk megtett  $s$  út, a hangsebesség



6.3. ábra: 30-ra és 100-ra állított osztási köz. Míg előbbi esetben összesen 8 hangforrást talált el sugár, utóbbinál majdnem mind a 180 darabot.

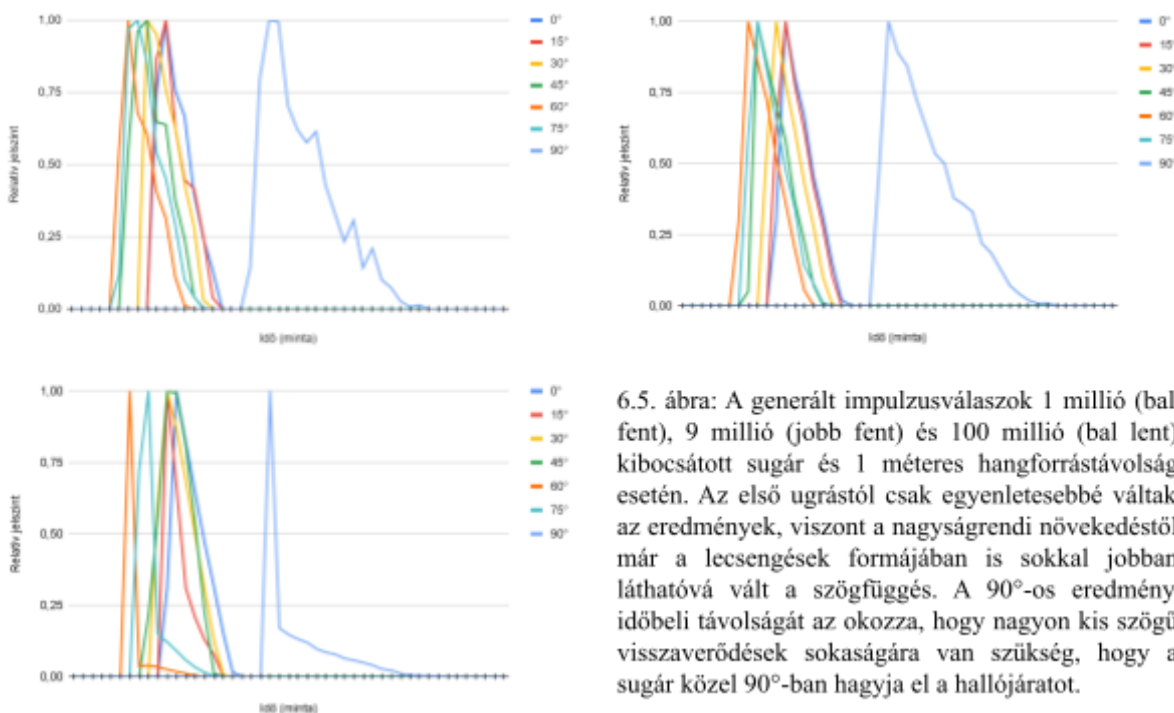


6.4. ábra: HRTF szettekhez hallójárat szimulációkat készítő szoftver felülete és importálási beállításai.

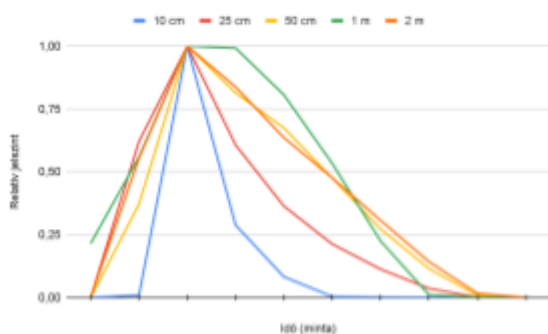
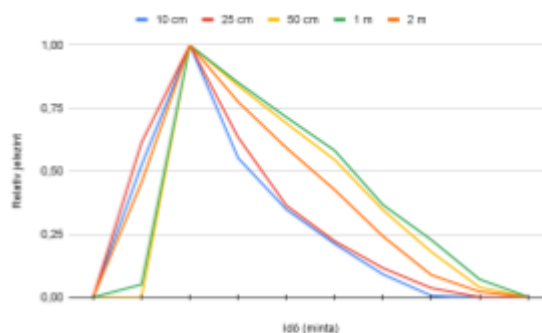
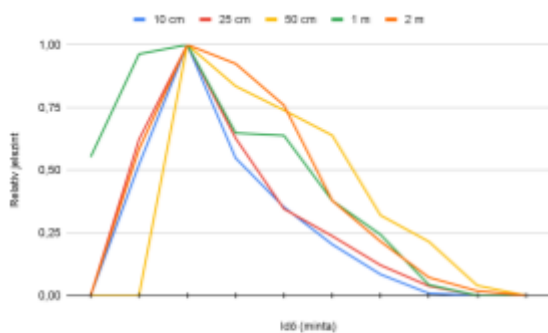
szobahőmérsékletű (25.4 °C-os), tökéletesen száraz levegőben  $c = 346.74 \frac{m}{s}$  értéke, és a feldolgozott HRTF szett  $f_s$  mintavételezési frekvenciája függvényében az időnek megfelelő

$$t = \lfloor \frac{d}{c} \cdot f_s \rfloor \quad (6.9)$$

pozícióban módosítják az eddig előállt impulzusválaszt. Egy beérkező sugár a  $t$ . mintában a hullámterjedési hangnyomásnak megfelelő  $\frac{1}{d}$  értékkel növeli az értéket. Az összes sugár kiszámolása után előáll a hallójárat impulzusválasza.



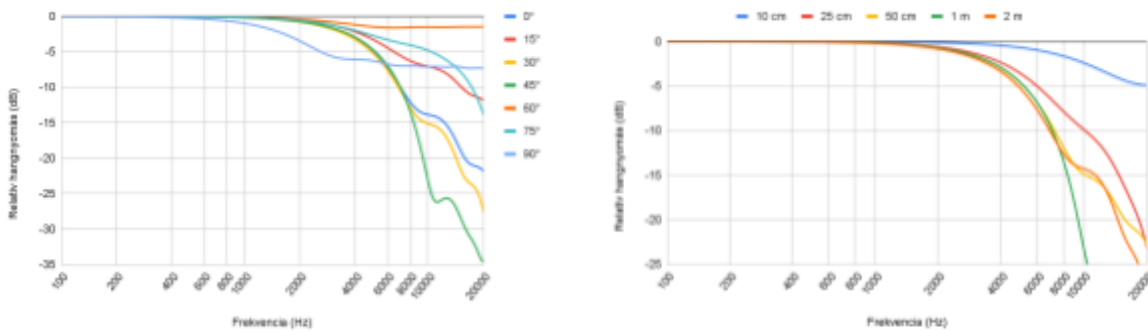
6.5. ábra: A generált impulzusválaszok 1 millió (bal fent), 9 millió (jobb fent) és 100 millió (bal lent) kibocsátott sugár és 1 méteres hangforrástávolság esetén. Az első ugrástól csak egyenletesebbé váltak az eredmények, viszont a nagyságrendi növekedéstől már a lecsengések formájában is sokkal jobban láthatóvá vált a szögfüggés. A 90°-os eredmény időbeli távolságát az okozza, hogy nagyon kis szögű visszaverődések sokaságára van szükség, hogy a sugár közel 90°-ban hagyja el a hallójáratot.



6.6. ábra: Generált impulzusválaszok 45 fokos beesési szögben, különböző távolságokból, eltávolított késleltetéssel. Az 1 millió sugarat kibocsátó szimuláció (bal fent) zajos eredményt ad, de 9 millió sugár esetében (jobb fent) már közel egységesek az 50 cm és afeletti szimulációk. 100 millió sugár használata már kiemeli a 10 és 25 cm-es távolság közti különbséget is, amit az előző két eredmény nem, az 1 méteres eredmény formája pedig betudható az alacsony mintavételezési frekvenciának.

A hallójárat impulzusválaszok generálásához egy olyan szoftvert készítettem, ami meglévő HRTF szettek alapján dolgozik. Bár az algoritmus pusztán a beállításoktól és a megszabott pontoktól függ, ez az importálási lehetőség magával vonzza, hogy ne csak a hallójárat impulzusválaszokat lehessen exportálni, hanem azok kész konvolúcióját a kiindulási szettel. További előny, hogy csak a betöltött HRTF szett számára releváns szögekben áll elő hallójárat-szimuláció. A szög itt a kiválasztott fül hallójáratából kifelé mutató vektorral bezárt szöget jelenti. Beállítható paraméter a hangforrások távolsága, a tengelyenkénti osztásszám, ami a sugarak kibocsátásakor ennyi részre osztja a kört mindkét tengelyen, illetve egy sugár visszaverődéseinek maximális száma. A korábban 2048 mintára maximalizált impulzusválasz minden paraméter maximalizálása esetén is bőven elegendő tartalékkal rendelkezik, 14.63 méteres távolságot megtett sugarakat is képes kezelni 48 kHz mintavételezési frekvencián. Az alkalmazott mintavételezési frekvencia az importált szettől függ, a szett vizsgált oldala pedig a kiválasztott fültől.

10, 25, 50, 100, és 200 cm-es távolságokra állítottam elő 15 fokonként hallójárat szimulációkat, a generált impulzusválaszok formáját a 6.5. ábra mutatja. A köztér azért hangsúlyosabb, mert drasztikus különbségek az eredményekben csak 50 cm-es hangforrástávolság alatt lettek. Ezt a tapasztalatot a 6.6. és a 6.7. ábrák szemléltetik. Mivel 9 millió kibocsátott sugár követése a számítógépen távolságértékenként 3 percig tartott, ilyen módon teszteltem. A 100 millió sugárból számolt adathalmazok előállítására 25 percet



6.7. ábra: A szimulált impulzusválaszok spektruma, beesési szög függvényében 1 méteren (balra) és távolság függvényében 45 fokos beesési szögben (jobbra), 100 millió sugarat használó szimulációval.

vett igénybe, a végleges szoftver ezeket az impulzusválaszokat használja. Az eredmények importálására és tömörítésére - üres kezdeti- és végminták levágására - egy importáló alkalmazást készítettem. Az impulzusválaszok egy minimális méretű tömbbe kerülnek, és egy olyan konvolúciós algoritmust használnak szűréskor, ami definíció szerint, de a 0 értékű minták esetében műveletet nem végezve működik. Ez a megoldás sokkal jobb teljesítményt nyújt, mint más módszerek, hiszen a szűrők hasznos hossza mindössze 6-18 minta.

A pontos implementáció a fejezet elején említett, mindkét oldali virtuális hangszórókat külön szűrő módszerrel történt. A távolságszimulációt objektumonként állíthatóvá tettem: amennyiben kikapcsolt az állapota, szimmetrikus rendszerekben az átlagoló módszerrel keverődik az adott objektum, viszont bekapcsolt állapotban erőltetetten irányított renderelés történik. Ahogy az arányos keverő egy síkban 4 csatorna közt állítja az erősítést, a távolságképző szűrő pontosan ugyanezekkel a függvényekkel, egyik tengelyen a fülhöz képest vett beesési szöggel, másik tengelyen a távolsággal számol. Így előáll 4 kijelölt  $I_i(t)$  impulzusválasz és az azokhoz tartozó  $A_i$  erősítés, amikből a konvolúciós szűrő által alkalmazott impulzusválaszt az

$$I_e(t) = \Sigma(A_i \cdot I_i(t)) \quad (6.10)$$

képlet adja meg.

A hangforrás irányával ellentétes oldalt kétféle módon is próbáltam szűrni. Az először kipróbált módszer a hangforrás oldalán alkalmazott szűrőkkel egyenértékű késleltetés alkalmazása, hogy a két fül egymáshoz képesti késleltetése ne változzon. Mivel a késleltetés nem jár torzítással, a hangforrás mozgása a fej egyik oldaláról a másikra éles, hallható változással járt. Bár a sugárkövetéssel a fel körül elhajló hullámokat nem lehet kezelni, így a

módszer a két oldalnak nem képes külön impulzusválaszt előállítani, megpróbálkoztam a két oldal egyenlő szűrésével, a távolságból fakadó erősítési eltérést megtartva. Ez azt jelenti, hogy a konvolúciót már egyedül a jefforráson, egyszer szükséges elvégezni, ami a legaggályosabb teljesítményprobléma megoldása. Ettől a lépéstől megszűnt a mozgás közben előforduló éles váltások jelensége, miközben a szűrő hatása változatlan.

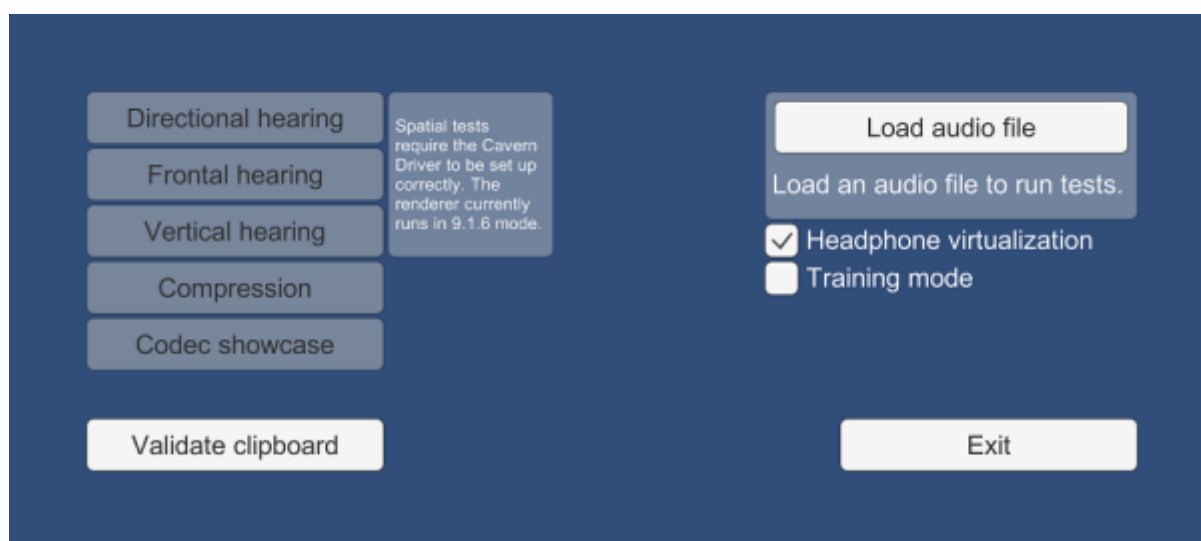
A legfőbb következtetés a szimulációk eredményéből, hogy a hallójárat hatása a beesési szögától és a távolságtól függő aluláteresző vagy high-shelf (magastartományt egyenletesen erősítő) szűrőként írható le. A kétféle hatás, ami a 6.7. ábrán látható, gyakran együtt is megjelenik. Az eredmény a hallójárat karakterisztikáját replikáló szűrő, ami az arányos keverő fizikailag is igazolt modelljéhez közelítő hatásra képes. Az erős irányítottság miatt azonban a közeleteren mozgó hangforrások nagy változásokat okoznak hangerőben, hiszen egy hirtelen szögváltozás az ellentétes fülre ugrást eredményezhet. Éppen ezért a diplomatervből levont eredmény az, hogy a hallójárat valóban fontos az irányhallás szempontjából, nagy hatása van a hallott hangra, viszont léteznek alaposan alátámasztott és pontosabb megoldások, mint az arányos keverés alapú HRTF vagy a hullámtér-szintézis.



## 7. Validáció

A munkám validálása különös nehézséget jelent, ugyanis a hallás egy szinte teljesen szubjektív érzékszerv, és nem csak az auditív memóriánk rendkívül rossz [14], de a fülkagyló alakja, így a megfelelő HRTF is egyénenként változik. Amit meg tudok határozni - az elméletek helyességének matematikai és fizikai bizonyításán túl -, az az, hogy a hallgató számára előidézi-e a kívánt hallást egy-egy HRTF szett. Ehhez egy tesztelő szoftvert készítettem el, mely a hang egy tengelyen vett pontos irányát kérdezi a felhasználótól. Az irányhalláson túl egy további tesztet is implementáltam, ami szerintem érdekes eredményeket hordoz, és a megvalósítása során váratlan ötleteket tudtam alkalmazni, ez a tömörítések hatékonyságát - vagyis ember általi érzékelhetőségét - hivatott értékelni.

A Nuance névre keresztelt szoftver indításakor a 7.1-es ábrán látható módválasztó- és beállító képernyő nyílik meg a felhasználó előtt. Mielőtt a játékoknak nevezett tesztek bármelyikét el lehetne indítani, egy hangfájlt szükséges betölteni, hiszen auditív vizsgálatot csak hanganyag segítségével lehet végezni. A támogatott formátum egyedül a RIFF WAVE, amit azért választottam, mert rendkívül egyszerű azt teljesen támogató implementációt készíteni, illetve bármilyen konvertálóprogram támogatja. A tömörítésekkel kapcsolatos játékhoz érdemes eleve ebben - vagy bármilyen veszteségmentes tömörítésben - kiadott zenezámot választani, de a formátumok vizsgálatai a későbbiekben bizonyítani fogják, hogy ez nem feltétlenül szükséges. Látható még egy “vágólap hitelesítése” gomb is, ezzel lehet igazolni, hogy egy szövegesen kapott eredmény módosíthatatlan-e, ugyanis minden játék végén a megjelenő kimenethez tartozik egy ellenőrző összeg, aminek célját a 7.1.4. fejezetben írtam le.



7.1. ábra: A Nuance indításakor megjelenő módválasztó- és beállító képernyő.

A fájlbetöltő panel alatt két kapcsolható beállítás szerepel: a HRTF-et erőltető “fejhallgató virtualizáció”, illetve a barátságosabb tesztek indító “tréning mód”. A szoftver alaphelyzetben a 3D térhangzást és HRTF-et támogató API-m beállításait veszi át, ez azonban alapértelmezetten (amennyiben a szoftver nincs telepítve) 5.1-es hangszórókiosztást jelent. Természetesen a beállítási felületén is aktiválható a virtualizáció, viszont a felülírás lehetősége szükségtelenné teszi a telepítését. További előnye az itt szereplő beállításnak, vagyis a HRTF engedélyezése és kerülése közti váltásnak, hogy az erre épülő tesztekhez gyorsan be lehet kapcsolni, illetve a tömörítések tesztjéhez, ahol hátrányt okozna, gyorsan ki lehet kapcsolni.

A tréning mód könnyítésekkel teszi tanulhatóvá a tesztek, természetesen ezáltal érvénytelenítve az eredményeket. Tréning módban minden kör után látja a felhasználó a pontos eredményét, illetve a bukást okozó hibák (például az engedélyezettnél több tévesztés a tömörítések tesztjében) nem vonják magukkal a teszt idő előtti befejeztét. Az így végzett tesztek is ellenőrző összeggel ellátott kimenettel zárulnak, azonban szerepel bennük a tréning mód ténye, azt kivágva pedig az ellenőrző érvénytelennek jelzi őket, vagyis ez az opció nem növeli meg a csalás valószínűségét.

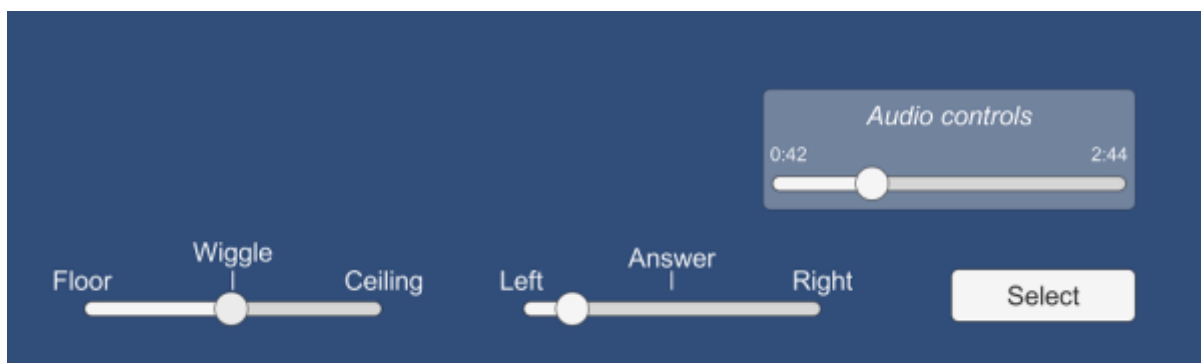
## **7.1. Elvégezhető tesztek**

A Nuance több játékmóddal is rendelkezik, ezek közül három az irányhallás vizsgálatára terjed ki, a negyedik azt állapítja meg, hogy tömörítések közt hall-e az alany különbséget, illetve szerepel egy extra demó is a módok között, ami interaktívan mutatja be, hogyan képes a szoftver lejátszás közben azonnal váltani többféle tömörítés között.

### **7.1.1. Irányhallási tesztek**

A szoftver eredendő célja az irányhallás vizsgálata, ezáltal egy HRTF szett működésének igazolása vagy éppen cáfolata. Ezt úgy éri el, hogy a fej körül egy véletlenszerű ponton elhelyez egy hangforrást, és a pont helyének meghatározása a cél. Hogy a teszt ne legyen túl bonyolult, ne kelljen a felhasználónak például gömbfelületen irányt jelölnie, mindhárom lehetséges dimenzióhoz külön tesztet társítottam. Ez lehetővé tette, hogy egy ilyen teszt mindössze két csúszkából álljon, az egyiket a tippelt irányt lehet beállítani, a másikon pedig a tesztől független mozgást lehet végezni a hangforráson, ami rásegítheti a játékost a megoldásra.

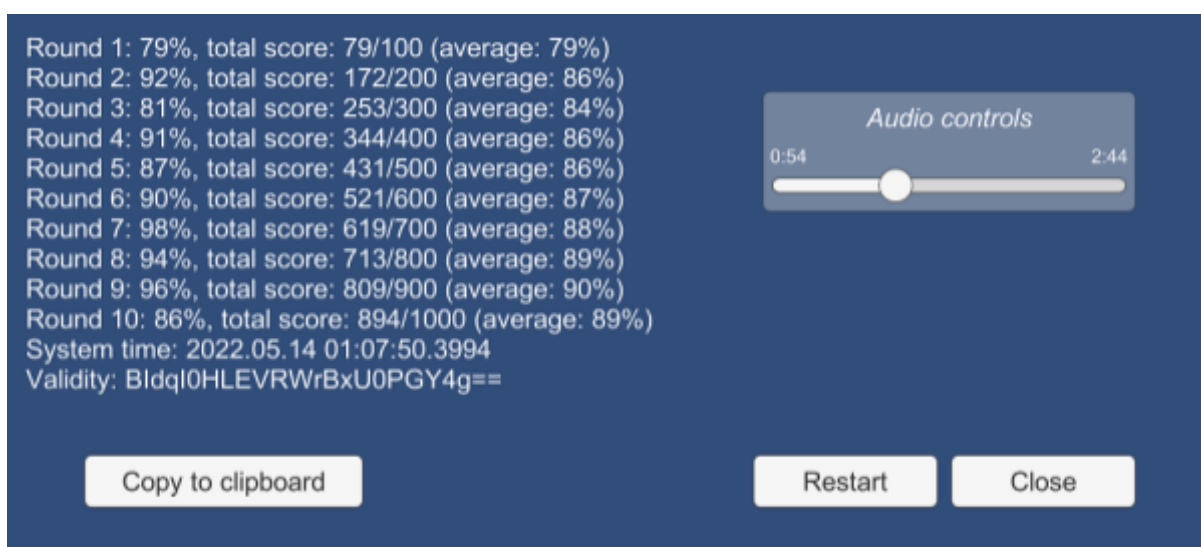
Egy teszt végén egyetlen irányú hallásunk pontosságát kaphatjuk meg, ez lehet bal-jobb, előre-hátra, illetve fent-lent. Azért is előnyös ez az elválasztás, mert vakon az ember



7.2. ábra: Irányhallás vizsgálata a bal-jobb tengelyen. Az elmozdítás függőlegesen engedélyezett.

rendkívül könnyen meghatározza a bal-jobb tengelyi irányát egy hangnak, viszont a másik kettőhöz rendszerint vizuális segítségre van szükség. Ez a teszt közben kizárt, hiszen vaktesztről beszélünk, erre jelent megoldást a “wiggle” csúszka. Hogy az érzékelés esélyét növeljem, de a megoldást ne változtassam meg közben, egy független tengelyen tetszőleges mozgást engedélyezek. Ez a szélességi tesztnél a függőleges tengely, a további tesztek esetében pedig a szélességi tengely. Szándékaim szerint ez redundanciát is biztosít, ha a HRTF irányai közül néhány hibás, mivel a hangforrás áthelyezhető olyan egyenesre, ahol a szűrők a helyes hatást érik el.

Amennyiben tréning módban fut a játék, minden kör után megjelenik az elért eredmény. Minden körben 100 pont szerezhető, ez a találat százalékos pontossága, ahol 100% a telitalálat, 0% pedig a teljesen ellentétes oldalra tolt csúszka. Körönként átlagolódik a haladás, és végeredményként 10 kör átlagát vehetjük. Éles módban, vagyis kikapcsolt tréning mód mellett a 7.3. ábrán látható záróképernyőn jelennek meg az eredmények, miután mind a



7.3. ábra: Az irányhallási tesztet záró képernyő. A pontatlanságok (például  $79+92=172$ ) nem hibák, hanem kerekített értékek.



7.4. ábra: Tömörítésekkel kapcsolatos hallásvizsgálat kettős vaktesztként implementálva.

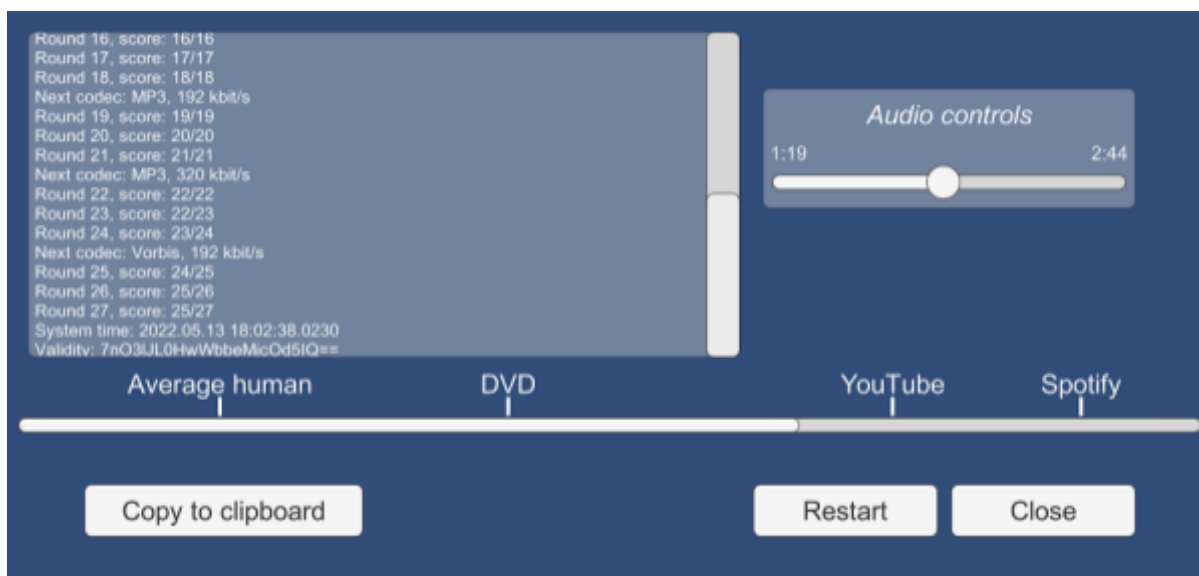
10 feladatra választ adott a játékos. Ez továbbítás és független hitelesítés céljából vágólapra másolható, illetve a teszt megismételhető vagy bezárható. Hogy a menüben tallózott zenével irányhallást lehessen tesztelni, a szoftver monóba keveri, és úgy helyezi el a térben. Egy kellemesebb záróképernyőért ez nem maradandó, a játék végére visszavált sztereóra.

### 7.1.2. Tömörítési teszt

Puszta érdeklődésen túl reális felhasználási módja lehet egy tömörítések megkülönböztetéséről szóló kettős vaktesztnek, hogy meghatározzam, melyik a játékos számára még legkisebb bitráta, amiben úgy tárolhatja a zenéit, hogy az számára még veszteségmentesnek tűnjön.

A 7.4. ábrán látható, hogy ez a teszt még egyszerűbb felhasználói felülettel rendelkezik, mint az irányhallásról szóló játékok. Egy kettős, más néven ABX vakteszt úgy működik, hogy a felhasználó három mintát kap, azaz háromféle tömörítés közül választhat. A feladata, hogy meghatározza, A és B közül melyik egyezik meg X-szel. A vakteszt azért kettős, mert nemcsak A-ról és B-ről nem tudni, hogy melyik a tömörítetlen és melyik a tömörített, hanem mert X-ről sem tudni, hogy melyik. A teszt bármennyi ideig húzható, illetve billentyűzettel is lehet váltani a tömörítések között. Szintén megengedett a szabad ugrálás a betöltött zenében, az egyetlen cél az azonos tömörítés megtalálása.

Minden támogatott tömörítés kiválasztott bitrátaiból 3 véletlenszerű ABX teszt fut. 3/3-as eredménynél garantált a továbbjutás a következő tömörítésre, de 2/3-nál már a kiesés veszélye áll fenn: az első ilyen eredményt még elnézem, a másodikat már nem, akkor véget ér a teszt, és megjelenik a záróképernyő. Ha még ennyi találat sem gyűlt össze, a kiesés garantált.



7.5. ábra: Tömörítési tesztet záró képernyő.

A 7.5. ábra bemutatja a teszthez tartozó záróképernyőt. Hasonlóan az irányhallási tesztekhez tartozó 7.3. ábrához, itt is megjelenik a kimenet (ami a hossza miatt már görgethető), annak másolása, az újraindítás, és a bezárás. Ami újdonság, hogy az eredményt egy szemléletes skálán is elhelyezi. Ezen megjelenik a teszt várható eredménye “átlagember” felirattal, a DVD hangját jelentő AC-3 formátum, illetve a YouTube és Spotify szintjét jelző, rendre 128 és 256 kbit/s bitrátájú Opus kódolás. A kimeneten látható, hogy milyen sorrendben következnek egymás után a tesztelt kódolások. Ez a sorrend konzisztens, a meghatározása pedig objektív módon történt, amiről a 7.1.3. fejezetben írtam.

### 7.1.3. Tömörítés cseréje valós időben

Egy olyan módszert találtam ki, aminek segítségével valós időben is könnyen és hatékonyan lehet váltani különböző kodekek és bitrátáik között. Hogy ne kelljen egyesével implementálni a kódolásokat, ne essen a valós idejű tömörítés miatt a teljesítmény, ki legyen egyenlítő az erősítésük (ami sok tömörítés, például az MP3 során megváltozik), és átmenettel se kelljen köztük foglalkozni, egy drasztikus egyszerűsítés mellett döntöttem.

A szoftver szemszögéből igazából az összes tömörítés egyetlen konvolúció, a kódolás vagy bitráta megváltoztatása mindössze a megfelelő impulzusválasz kiválasztását jelenti. Hogy a lejátszott hang felbontása ne okozzon problémát, előbb újramintavételezek 48 kHz-re, majd ezután alkalmazom a szűrőt.



7.6. ábra: Tömörítésváltó demó.

Az impulzusválaszok elkészítéséhez három szoftvert is használtam, első lépésként Audacity-ben kézzel rajzoltam egy Dirac-deltát, majd ezt exportálva megkaptam a veszteségmentes tömörítéseket reprezentáló szűrőt, elvégre

$$f(x) * \delta(x) = f(x). \quad (7.1)$$

Hogy egy tömörítést reprezentáló impulzusválaszt megkapjak, az exportált Dirac-deltát FFmpeg segítségével különböző kodekekkel és bitrátákkal tömörítettem, majd visszaalakítottam őket WAV fájlkká, hogy könnyedén importálni lehessen őket. A választott kódolások blokkonként működnek, és a pontosságot (amit például a mantisszára vagy exponensre szánt bitek száma ír le) sávonként határozzák meg, ezáltal a Dirac-delta, amiben a frekvenciák teljesítményeloszlása egyenlő, a lehető legnehezebb feladat nekik.

Lett tehát egy impulzusválaszom a kiválasztott tömörítések legrosszabb eseteiről, amik segítségével egy konvolúcióval el tudom érni ugyanazt a hatást, mintha valós időben lenne tömörítve a kiválasztott hanganyag. Kezdetben 1 másodperces konvolúciót terveztem, hátha illesztési hibák is kerülnek az eredményekbe, de mivel ez nem történt meg, 512 mintára rövidítettem őket, aminek során számottevő (0.01 dB-n túli) változás kizárólag 20 Hz alatt történt, az pedig az emberi hallás határai miatt elhanyagolható.

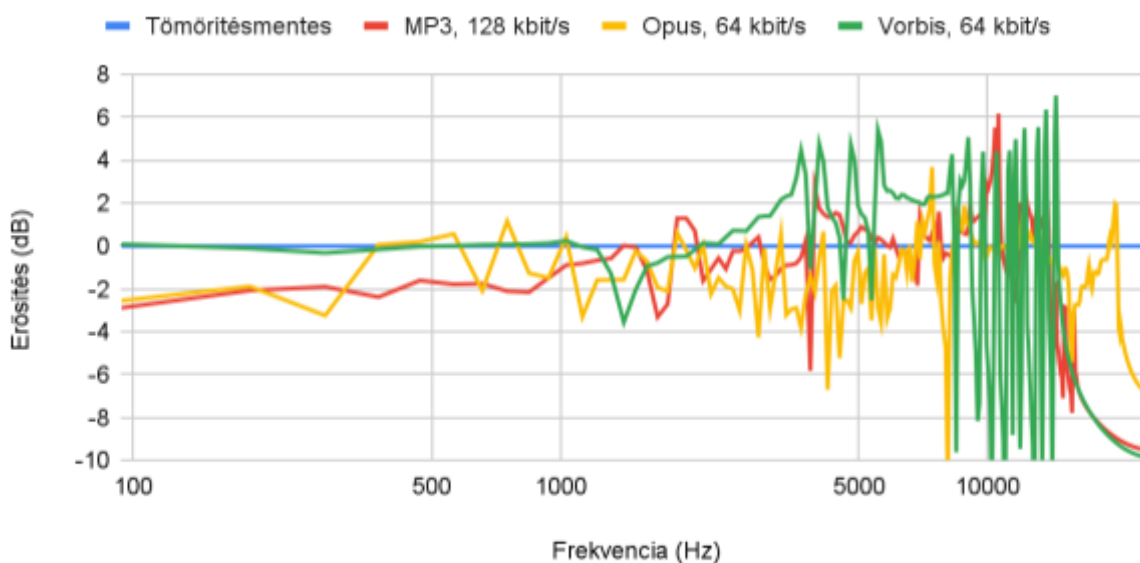
Hogy a tömörítések észlelését vizsgáló játékban ne kelljen minden egyes kódolásra elvégezni a tesztet, hanem álljon meg azon a ponton, ahol a hallgató már biztos bukná a

további tesztek, egy objektív sorrend felállítása volt szükséges. Ezt úgy oldottam meg, hogy minden impulzusválaszhoz meghatároztam egy értéket, amit “pusztítás mértékének” hívok, és ez egy olyan összehasonlítható mérőszám, ami megadja, mennyiben tér el a tömörített hang a veszteségmentestől:

$$p = \Sigma |x_i - 1|, \quad (7.2)$$

ahol  $x$  az impulzusválasz Fourier-transzformáltja. Mielőtt azonban importáltam volna az impulzusválaszokat, közös átlagos jelszintre kellett hoznom az átengedett tartományt, hiszen a hangerőkülönbség megkerülné a teszt lényegét, a hangminőségbeli különbség meghatározását. Ehhez az Equalizer APO szoftver szűrővizualizáló felületén meghatároztam az átlagos erősítést, és Audacityben úgy korrigáltam a szűrőket, hogy ez 0 dB-re essen.

A különböző tömörítések hatását a 7.7. diagramon szemléltetem. Ellenőrzésképp és referenciának a tömörítésmentes (vagy veszteségmentesen tömörített) kódolásokat jelképező Dirac-deltát választottam, ami a várható konstans 0 dB-es értéket vette fel. Ezen felül, hogy még átlátható legyen az ábra, összesen háromféle tömörítést választottam, a változó bitrátájú formátumok legrosszabb eseteit jelenítettem meg. Látható rajtuk a mélyfrekvenciás hangok lecsökkent jelszintje, az aluláteresztés, illetve az is, hogy a legmodernebb kódolás (Opus) már a legkisebb bitrátán is képes lefedni a teljes emberi hallástartományt. Éppen ezért történt, hogy amíg az objektív rangsor szerinti két legrosszabb szereplő rendre a legkisebb méretet eredményező Vorbis és MP3, addig az Opus be sem került a legrosszabb 3-ba.



7.7. diagram: Válogatott tömörítések hatása a tömörített hang spektrumára.

#### 7.1.4. Hitelesítés

Hogy sok tesztelőm legyen, és minél reprezentatívabb legyen a kutatás, az internet népét is segítségül hívtam az ismerősökön túl. Mivel az emberek hajlandók úgy felfogni egy eredményt, hogy nem a HRTF hibás, hanem az ő hallásuk, fel kellett arra készülnöm, hogy esetleg kézzel átírhatják az eredményt. Ennek elkerülése érdekében minden kimenet végén egy ellenőrző összeg található, ami a szövegben található számok egy fekete doboz által adott hash-e. Mivel nem feltételezem, hogy egy hamis eredmény miatt bárki is visszafejtené a szoftvert, úgy döntöttem, hogy obfuszkálás és/vagy C++ DLL-hívások helyett bőven megteszi a security by obscurity módszere is.

Egy kapott eredmény hitelesítése nagyon könnyű: a vágólapra kell másolni, majd a tesztsoftverben a “vágólap hitelesítése” gombra kattintva meg is jelenik mellette a “Valid!” (érvényes) vagy “Invalid!” (érvénytelen) jelzés. Ez úgy áll elő, hogy a szövegben található számokat (az ellenőrző összegén kívül) összegyűjti, majd ugyanazt a hash függvényt futtatja rajtuk, ami játékok végén is meghatározza a hitelesítő karaktersorozatot. Ha ez megegyezik a beolvasottal, a teszt hiteles volt, nem történt csalás.

A tréning módban történt tesztekre az volt az eredeti tervem, hogy a hash függvény minimális változtatásával (például egy összeadás cseréje kivonásra) álljon elő az ellenőrző összeg, és semmi ne változzon a kimeneten. Ez azért lett volna rossz döntés, mert hihetnék azt, hogy nyugodtan beküldhetnek tréning eredményeket, mert ugyanúgy hiteles. Az ellenőrző összeg törlése sem vezetett volna célra. Végül egy olyan elrettentés vezetett célra, hogy a tréning módban végzett tesztek elején megjelenik a “Nuance 1.0 training report” felirat. Ebben szerepel az 1-es és 0-s szám, amit már felhasznál a hash függvény, így a sor törlése azzal járna, hogy a teszt érvénytelenné válik.

Más ABX teszteken, például a foobar2000 ABX Comparator esetében, a kimenet része az adott teszten eltöltött idő, illetve a tesztelt fájl neve és hash-e. Ezeket nem tartottam fontosnak, és nem azért nem implementáltam őket, mert kihívást jelentene, hanem csak a helyet foglalná a záróképernyőn. Az irányhallási teszten teljesen mindegy a hangsáv és a tippeket is nagyon gyorsan szokták megjelölni, így csalókat nem lenne képes megfogni, a tömörítési teszten pedig a csaláshoz kitalált fájlok maximum a második körig jutnának, mivel utána már nincs emberi hallástartományba eső vágási frekvenciája egyik kódolásnak sem.



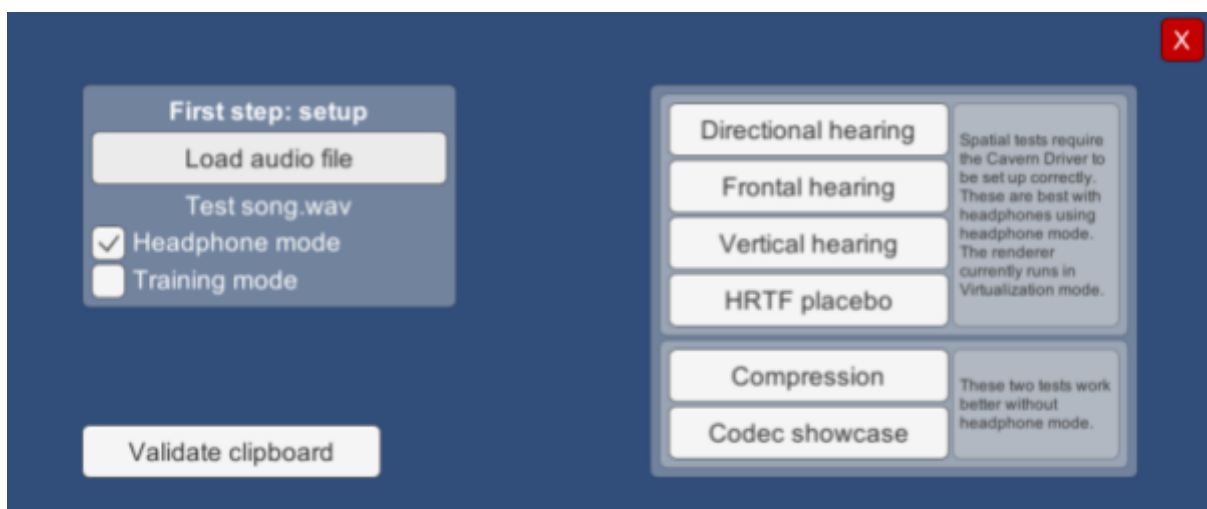
## 7.2. Újratervezés

Az első napon meghívott 8 tesztelő javaslatai és a teszt levezénylésének nehézségei a szoftver megjelenésének alapos áttervezését, a kétes eredmények pedig a tesztek újragondolását vonták magukkal. Kifogásolt jelenség volt például, hogy a kör számáról nem adott visszajelzést a játék, és emiatt, ha a hang alig változott, azt hitték, hogy csak nem kattintottak az elküldés gombra. A menü elrendezése miatt jobbról balra helyezkedtek el az egymás után használandó kezelőszervek, amikkel szinte senki nem barátkozott szívesen.

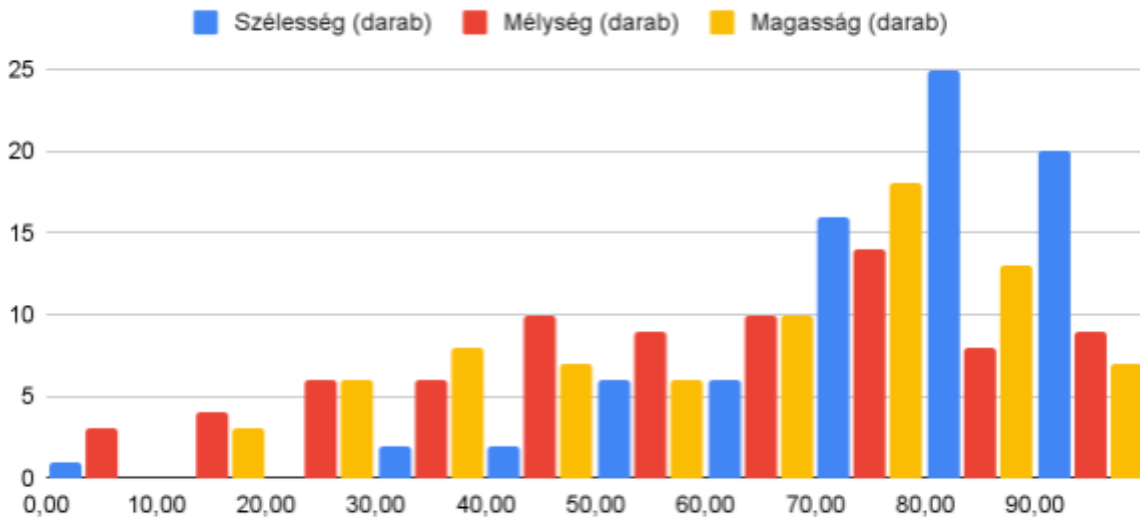
Utolsó és legnagyobb probléma volt, amint a kiválasztott tesztbe csöppenve a tesztelők semmiféle instrukciót nem kaptak, a játék nem magyarázta el a feladatát, így azt nekem kellett. Emiatt, és a korábban említett körszámláló hiánya miatt került be a játék elején egy rövid leírás a feladatról, illetve amennyiben nem tréning módban fut a teszt, úgy a helyén a forduló száma, a tömörítési tesztben pedig - ahol nem fix az utolsó kör száma - a jelenleg tesztelt tömörítésből hátralévő körök száma.

Az eredmények nagyon a határon mozogtak, és ehhez hozzájárult az irányhallási tesztek rossz tervezése. Binárisan eldönthetővé kellett volna tenni őket, illetve az eredmény sokkal több információt lenne képes mondani, ha a tippelt és valós pozíciót is tartalmazza. Így különböző statisztikai feltételezésekből lehetett csak elfogadási tartományokat felállítani, de ezek teljesen abszurd számokat adtak:

Egy kör után egyedül azt tudjuk, hogy milyen pontosan találta el a játékos a pozíciót az adott tengelyen, vagyis egyenértékű információ vele a megjelölt és helyes pozíció normalizált különbségének abszolút értéke. Ha tippelgetést feltételezünk, vagyis mindkét



7.8. ábra: Újratervezett, egyszerűbb és informatívabb főmenü.



7.9. diagram: Az irányhallási tesztek találati pontosságainak hisztogramja az első 7 tesztelő esetében.

valószínűségi változó egyenletes eloszlású és véletlenszerű, akkor a következő valószínűségszámítási probléma áll elő: legyen  $X$  és  $Y$  egy ennek megfelelő valószínűségi változó például  $[-1; 1]$  tartományban, ahol azokat az eseményeket vizsgáljuk, hogy

$$A: \operatorname{sgn}(X) = \operatorname{sgn}(Y), \quad (7.3)$$

azaz a tipp iránya és a helyes irány is megegyezik, illetve

$$B: |X - Y| \leq C, \quad (7.4)$$

ahol  $C$  egy olyan maximális eltérés, ami alatt legalább 90% annak az esélye, hogy a válasz helyes, vagyis az előjel megegyezik. Ezt az értéket kerestem, és a szimulációval is igazolt megoldás nem tetszett:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{2C - 2C^2}{2C - C^2} \geq 90\%, \quad (7.5)$$

ami  $C$ -re átrendezve  $C < \frac{2}{11}$ . Mivel ez eltérés, az eredményekre vetítve ez azt jelenti,

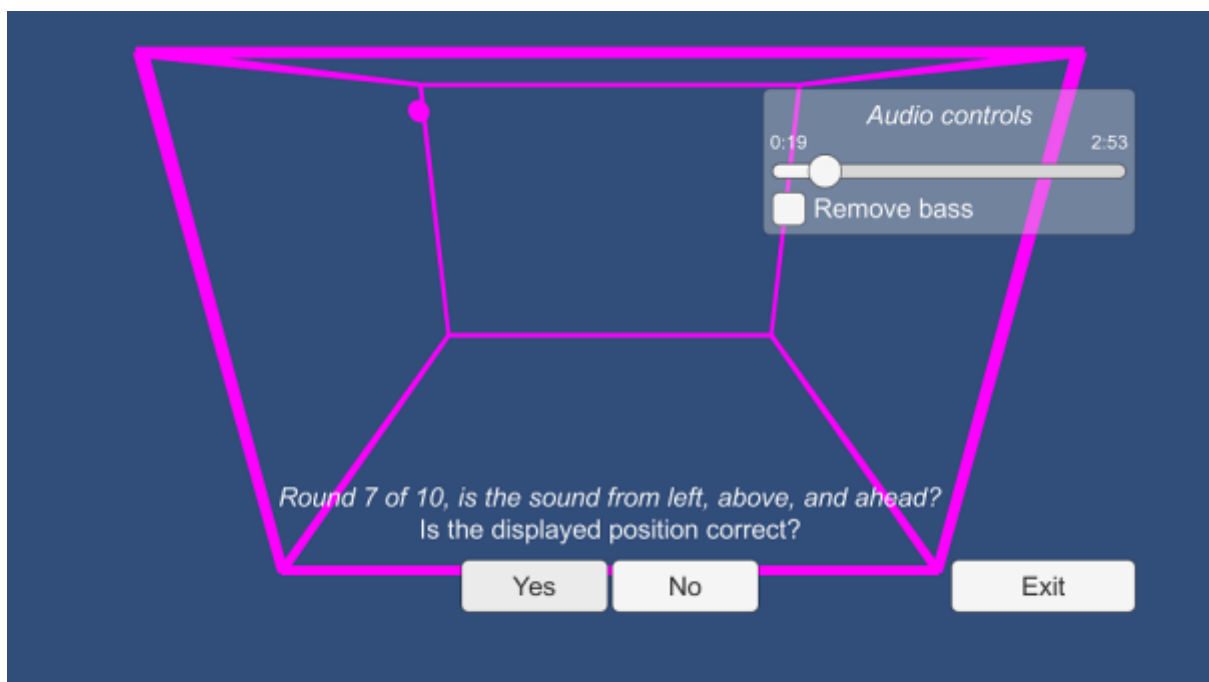
hogy a legalább  $\frac{9}{11}$  értékű, azaz legalább a 82% értékű találatokat szabadna elfogadnom. A probléma az, hogy ilyen átlagot a tesztelők egyike sem ért el, olyan evidens teszten sem, mint a bal-jobb irányú hallás. Mivel statisztikai alapon nem sikerült sok értéket kinyernem a tesztek közül, mindenképpen új tesztet kellett készítenem.

Ennek ellenére első ránézésre az eredmények hisztogramon elhelyezése, amit a 7.9. diagram mutat be, reménykedésre ad okot: az eredmények mindhárom teszten a 100%-os eredmény irányába tolódtak el, vagyis a tesztelők látványosan pontosabban határozták meg az irányokat, mint amit a vakon kattintgatás eredményezett volna, ami az egyenletes eloszlás.

### 7.2.1. HRTF placebo teszt

Az irányhallási tesztek a egyáltalán nem voltak barátságosak, nem szívesen ültetnék le senkit percekig két csúszkát tologatni. A tesztnek ezért mindenképp egyetlen eldöntendő kérdést kell feltennie az iránnyal kapcsolatban. Hogy rövid tesztek legyenek nagy áteresztő képességgel, úgy döntöttem, hogy a teszt mindhárom dimenzióban választ egy irányt, és oda helyezi el a hangforrást, majd feltesz egy kérdést az irányokat felsorolva, hogy onnan hallani-e a hangot. Kiíráskor az irányok közül az esetek felében az egyik helyet rosszat mond, és azt kell eldönteni, hogy jó vagy rossz a kiírás. A teszt nevének azért választottam azt, hogy HRTF placebo, mert segít eldönteni, hogy placebo volt-e az előző tesztek pozitív felé húzó eredménye.

Szintén grafikus segédlet, hogy egy virtuális téglatestben megjelenik a hangforrás, utalva a térbeli pozíciójára, olvasás helyett ez is kijelzi a feladatot. Ez azonban csak másodlagos kijelző, hiszen a fix kameraállás (hátról, középen, fentről nézve a téglatestre) nem teszi minden esetben egyértelművé a hangforrás felajánlott helyét. Egy auditív segédletet is elhelyeztem lehetőségként, ez a basszusok levágása, ami egy egyszerű másodrendű



7.10. ábra: HRTF placebo teszt futás közben, egy pozíciót felajánlva, amiről el kell dönteni, hogy valós-e.

Butterworth felüláteresztő szűrő 120 Hz-en, hátha néhány játékos jobban tud a segítségével lokalizálni.

A kijelzések letagadnak egy fontos tény a tesztről: a hang a szoba oldalhosszának 25%-át lefedő, véletlenszerű inklinációjú elliptikus pályán mozog. A korábbi tesztekkel ellentétben biztosra mentem, hogy egyértelmű legyen a megoldás, vagyis holtteret alkalmaztam közeltéren, így a helyes pozíció oda nem kerülhet. A holtter miatt az elliptikus pálya nem okozhatja azt, hogy másik nyolcadba kerül a hang, esetleg egyetlen képkockára a határát érintheti. Mivel nagyon gyorsan, 6,28 másodperc alatt megtesz egy kört, ez egyáltalán nem probléma. Az új verzió első tesztelői beszámoltak arról, hogy a jelenség feltűnő, de ez volt a célom.

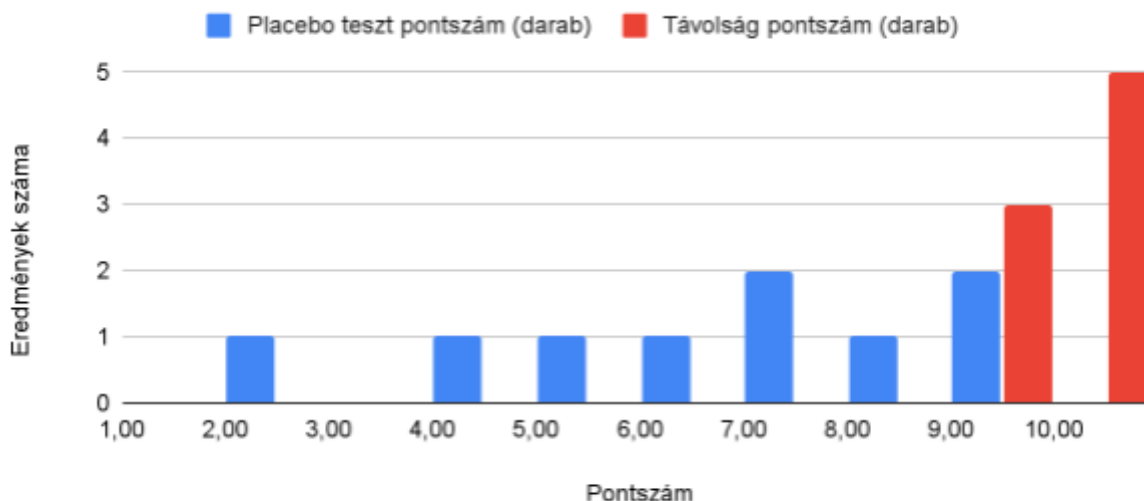
### 7.2.2. Távolság teszt

Rengeteg tesztet készítettem, épp csak a dolgozat témája maradt ki: ez a távolság átadásának próbája. Ha ezen a teszten jó eredményt érnek el a tesztelők, a diplomatervem témájaként előállított távolságot átadó szűrőkészlet működik, függetlenül attól, hogy a melléktermékként elkészült saját HRTF szett működik-e vagy sem.

A teszt úgy működik, hogy véletlenszerű irányban a vízszintes síkon elhelyez egy hangot. Az irány tartománya korlátos, a bal vagy jobb fül normálvektorától maximum 45°-kal térhet el. Ezt azért kötöttem ki, hogy ne kelljen felesleges feladatokkal stresszelni a tesztelőt, itt éppen a tengely meghatározásával, miközben mégis szükséges valamennyi véletlenszerűség, hogy ne tanuljanak rá a HRTF szett néhány irányhoz tartozó jellegzetes torzítására.



7.11. ábra: Távolsághallás vaktesztje futás közben.



7.12. diagram: Az újragondolt tesztek eredménye, sikertelen HRTF teszttel és sikeres távolság teszttel.

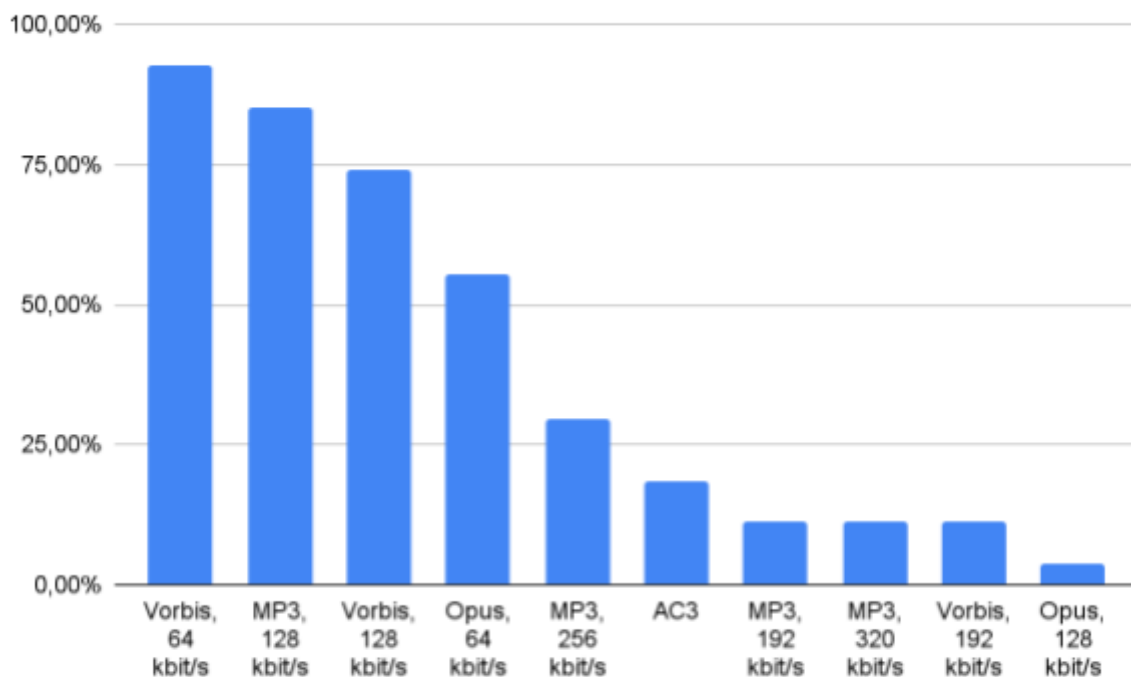
Ebben a tesztben két távolság és egy irány generálódik véletlenszerűen minden körben. A felhasználó az A és B verzió kiválasztásával elhelyezheti a hangforrást egyik vagy másik távolságban, feladata a távolabbi verzió megjelölése, amit egy elküldés gombbal nyomatékosíthat. Az A/B/Enter gombok a tömörítés testjéhez hasonlóan működnek ebben a játékban is.

## 7.3. Végeredmények

### 7.3.1. Új tesztek eredményei

A javított tesztek eredménye már egyértelmű, és a 7.12. ábráról mindkét eredmény leolvasható: a HRTF szűrőkészletem tényleg placeboként jelentkezett a tesztelők többségénél, viszont a távolságszimuláció, ami a munkám lényegi része, nagyon jól sikerült, és a vakteszten is bizonyított.

Ezúttal a tesztelőket megkérdeztem, hogy a HRTF virtualizáló demóban, ahol vizuálisan látják, honnan kellene hallani a hangforrást, és szabadon mozgathatják azt, helyesen hallják-e az irányokat. Egy olyan javaslat érkezett, hogy a hátsó irányokat jelölhetné visszhang, amit meg fogok a jövőben próbálni. Mindössze egyetlen ember válaszolt nemlegesen, a többiekről a placebo teszten derült ki, hogy a vizuális inger hatására élték meg így. A távolsághallás teszt eredménye pozitív, 9/10-es pontszámnál rosszabb eredmény nem született. Mivel a HRTF-hez szükség van vizuális ingerre, a távolsághoz nincs, a szoftvercsomag játékokban, filmekben, illetve virtuális valóságban való használatra alkalmasnak tűnik.



7.13. diagram: Tömörítésekről szóló vakteszt eredménye, az adott tömörítést meghatározni képes tesztlők arányaival.

### 7.3.2. Tömörítések vaktesztjének eredménye

Amikor megírtam a tömörítésekről szóló tesztet, egy olyan eredményt vártam, ami egyértelműen meghatározza azt a tömörítést, ami a legtöbb ember hallásának a határa. Ekkor lenne egy minimális méretű, nem hallhatóan degradált formátum, amiről tudom, hogy ha azt használom a szoftvereimben, illetve az archívumomban, akkor érdemi károsodást nem szenvednek a fájlok. Sajnos a teszt nem ezt a választ adta.

Szinte egyenletesen oszlik el, hogy ki melyik tömörítésnél bukta el a tesztet. Mivel egy kodeket 3-ból 3-szor el kellett találni, ez nem olyan, mint a placebo teszt, ahol az elvárt eredmény így néz ki. Típpeléssel exponenciálisan nehezedik minden egymást követő tömörítés elérése, a 7.13. ábrán látható 128 kbit/s-os Opus-hoz jutás esélye így 0,00000009%. Az emberek közt tehát közel egyenletesen oszlik el, hogy a felsorolt kódolásokból ki melyiket képes még meghatározni.

Ha mindenképpen meg kellene jelölnöm egy kódolást, amit csak nagyon kevés ember lenne képes felismerni, vagyis bőven alkalmazható az említett tömörítési célokra, az a 128 kbit/s bitrátájú Opus, ami a YouTube jelenlegi tömörítési formátuma.

## 8. Dolby Atmos támogatás

A kiírt feladatokon túl mindenképpen szerettem volna egy gyakorlati felhasználási módot a szoftvernek, ami nem igényel egyedi integrációs megoldást, például injektálást játékokba. 3D filmhangok lejátszására és újrakeverésére esett a választásom, mivel a HRTF megoldásom egy működőképes hangmotor, illetve a felmerülő formátumok egyike teljesen és nyilvánosan dokumentált<sup>[15, 16]</sup>, ez a Dolby AC-3 with Joint Object Coding, piaci nevén a Dolby Digital Plus Atmos. Mivel a hangsáv szinte minden esetben a hozzá tartozó képi anyaggal egy konténerben van tárolva, egy ilyen formátumot is támogatnom kellett, a választásom a demók által leggyakrabban használt Matroskára (mkv) esett.

A filmekhez használt 3D térhangzású kódolások kétféle hangforrásból állnak: ágyakból, amiknek egy adott konkrét hangszóróból kell megszólalniuk (ezek a hagyományos csatornák), illetve objektumokból, amikhez egy térbeli pozíció társul, és szabadon mozognak a film során. Mivel az ágyak akár mennyezeti hangszórók is lehetnek, ezért egyszerűen objektumként kezelem ezeket a hangforrásokat is, csak fix térbeli pozícióval. Ezt az 5.4. fejezetben említett demószoftverem már képes keverni, így pusztán az objektumok pályáját és hangját kell számára átadni a tömörített filmhangból.

### 8.1. Szerkezet

Ez az Atmos verzió nem önálló kódolás, hanem egy hagyományos 5.1 vagy 7.1 keverésből áll elő különböző szűrők és mátrixok alapján. Hogy ez visszafelé kompatibilis legyen, az alap formátum (AC-3) kihasználatlan mezőiben egy kiegészítő formátumot helyeztek el. Ezek a mezők az *auxdata* és a *skipfld*, mindkettő későbbi meghatározás céljával került a szabványba. Hogy ezeket a mezőket, illetve a legfontosabb metaadatokat (például forrás csatornák száma, mintavételi frekvencia, stb.) kinyerjem, egy teljes AC-3 dekódert meg kellett írnom, csak ezután kezdhettem neki a 3D információk dekódolásának.

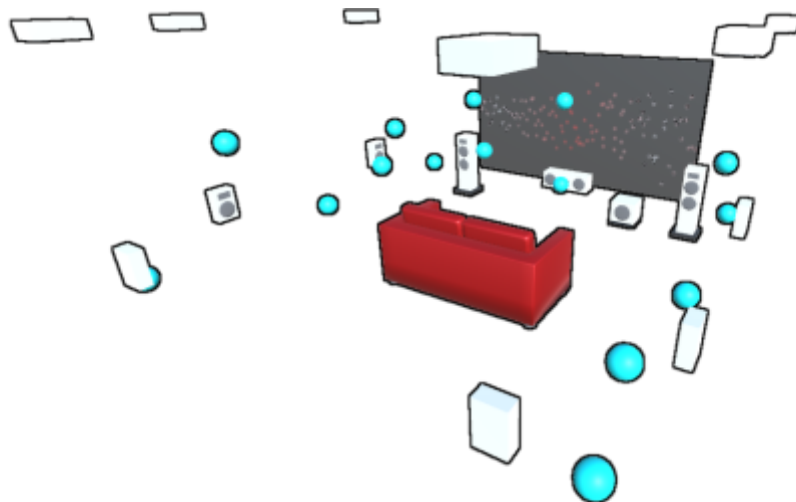
Ezek a mezők egy Extensible Metadata Delivery Format (EMDF) típusú konténer tartalmaznak, amit például a Matroska formátumban használt bináris XML-hez (Extensible Binary Meta Language, EBML)<sup>[17]</sup> vagy az MXF archiváló konténer Key-Length-Value (KLV)<sup>[18]</sup> szerkezetéhez lehet hasonlítani: egymást követik az azonosító - méret - adat hármast tartalmazó blokkok, melyek további, ugyanilyen blokkokat tartalmazhatnak, míg végül egy fa áll elő a metaadatokból.

### 8.1.1. Objektumok elhelyezése

Egy EMDF blokk többféle csomagot is tartalmazhat, de ezek közül az Atmoshoz csak kettőre van szükség: egyik az Object Audio Metadata (OAMD), ami az objektumok pozícióját tartalmazza, másik a Joint Object Coding (JOC), ez az objektumok hanghullámát előállító információk halmaza. A kettő közül az OAMD egyszerűbb, háromféle objektumhalmazt tartalmaz: egyénileg definiált ágyakat, előre meghatározott kiosztású ágyakat (ennek segítségével 3 bit akár 30 objektumot is létrehozhat és elhelyezhet), illetve a szabadon mozgó objektumokat.

Az első két kategóriát könnyű implementálni, hiszen táblákban rögzíthetők a meghatározott csatornapozíciók. A gyakorlatban csak egyénileg definiált ágyakkal és szabadon mozgó objektumokkal találkoztam, a leghatékonyabb megoldást egy hanganyag sem használta azok közül, amiket kipróbáltam. Ennek valószínűleg az az oka, hogy a konkrét pozíciók, amik a kiosztásokhoz tartoznak, nem szerepelnek a szabványban, így az enkóderek készítői elkerülték a módszer használatát.

A szabad objektumok kétfélek lehetnek: szobához vagy megjelenítőhöz horgonyozottak. Bár implementáltam mindkét esetet megfelelő transzformációkkal a saját szoftverem koordináta-rendszerébe, utóbbit nem tudtam tesztelni, mivel ilyen típusúnak jelzett hangforrást egy anyagban sem találtam. Az objektumok elhelyezése minden tengelyen 64 pozícióban lehetséges, további tömörítés céljából a különbségi kódolás is támogatott. Az ilyen kis felbontású mozgások darabosnak tűnhetnek, de ez két okból sem gond: egyrészt egyik falon sem fordulhat elő egymás mellett 64 hangszóró a valóságban, másrészt az OAMD



8.1. ábra: Az Adam-projekt című film objektumainak megjelenítése egy virtuális szoba terében. A demószoftver fehéren jelzi az aktív kimeneti hangcsatornákat, vagyis a példa esetében a teljes 9.1.6-os csatornakiosztású elrendezés megszólal.

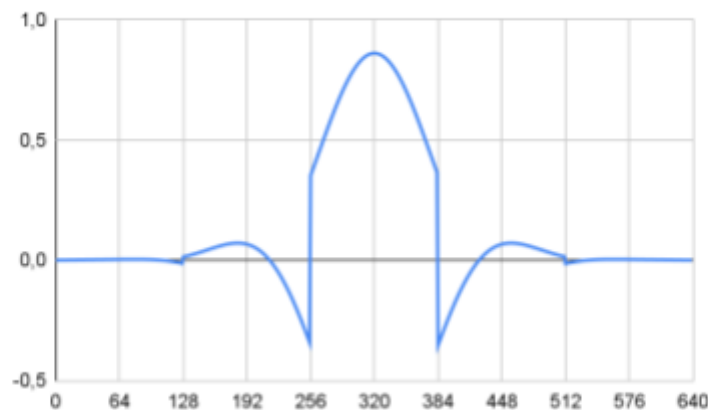


tartalmaz információt arról is, hogy a hangok hogyan végezzenek átmenetet két pozíciófrissítés között. Bár megadható olyan kiegészítő információ, ami a hangforrások helyzetének pontosságát ötszörözi, ezt a lehetőséget egy anyag sem támogatja, ami az előbb leírtak miatt várható volt és érthető. A kódolás akár 64 mintánként képes lenne frissíteni a pozíciókat, de a bevett gyakorlat (azaz minden vizsgált anyagban így volt), hogy szinkronizációs keretként, 1536 mintánként egyszer frissítik, és a maximális, ugyanekkora terjedelmű átúszást alkalmazzák.

### 8.1.2. Objektumok hangjának előállítás

Az objektumok hangmintáit meghatározó Joint Object Coding valójában egy keverési mátrix. Hagyományosan egy keverési mátrix egyik dimenziója a bemeneti csatornák, másik pedig a kimeneti csatornák, és azt határozza meg, hogy melyik bemenet milyen jelszinten keverődik az adott kimenetre. A JOC esetében is hasonló történik, azonban a mátrix háromdimenziós, ugyanis a bemenetet sávokra bontja szét.

A bemenetet sávokká bontó szűrő a szabvány része, így állandó, az ETSI dokumentumából emeltem át a konkrét implementációt. A teljesítmény kezdetben nem volt valós idejű lejátszásra alkalmas, ezért alaposan meg kellett változtatnom a másolt algoritmust, hogy optimális legyen. Mivel az alkalmazott transzformációt (quadrature mirror filter bank) nem ismerem és magyar szakirodalomban nem is találtam meg, egy olyan optimalizálási módszert alkalmaztam, amit csak utolsó eshetőségként ajánlott bevetni hangfeldolgozásban: többszálásítottam a problémát, minden csatorna “szeletelése” külön szálon zajlik. Ezzel a változtatással már valós időben előállítható a keverési mátrix alapján minden objektum hangja, lehetővé téve a hangsáv lejátszását.

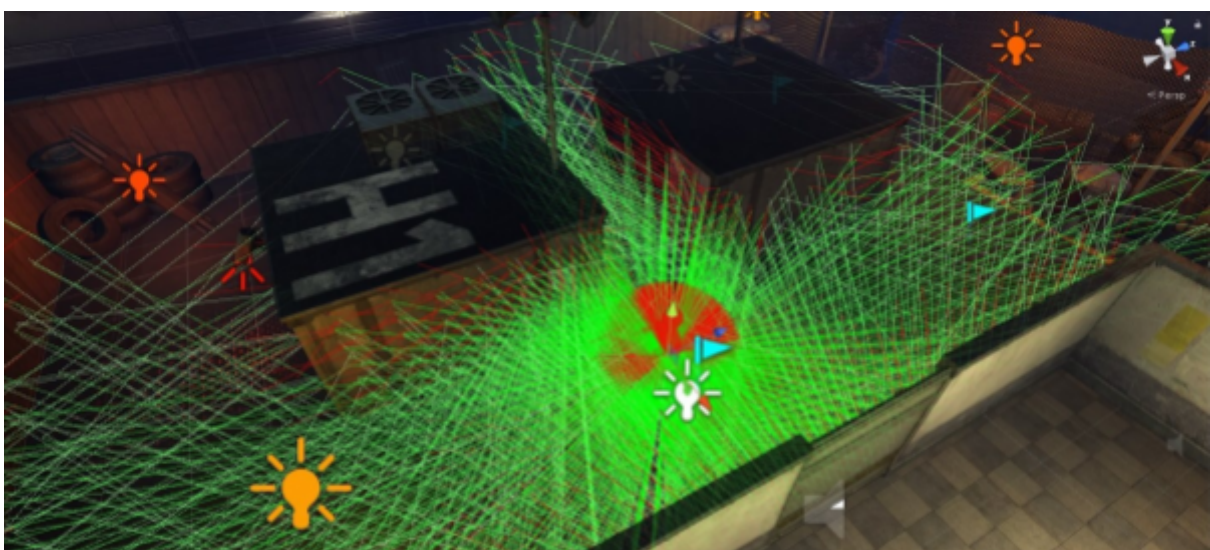


8.2. ábra: A Dolby Atmos szűrőkészlete (ETSI TS 103 420 V1.2.1, 7.4. fejezetben megjelölt tömb értékei). A tükör szűrőbank szimmetriája jól látható, a függőleges rácsvonalak a szűrők határát jelzik.

## 9. Összefoglalás

A diplomatervemben az emberi hallójárat hangra mért hatásainak szimulációját vizsgáltam és ültettem át egy gyakorlatban is használható hangmotorba. Ehhez kiindulási pontként egy HRTF segítségével virtuális fejhallgató térhangzást létrehozó szoftvert fejlesztettem, amihez megmértem és elkészítettem a saját fejemhez tartozó HRTF halmazt. A halmaz elkészítése és egy tudományos célra szánt halmaz elemzése során megismertem a fej és fülkagyló hatásait a beérkező hanghullámokra. Előállt a beesési szög és a távolság függvényében a távolabbi fül csillapítását és az időbeli különbségeket leíró függvény. Vizsgáltam és alkalmaztam annak a lehetőségét, hogy a HRTF méréseket bal-jobb fül párokban egyenlően korrigálva milyen hangzásbeli javulás érhető el. Az impulzusválaszok korrigálására egy állítható szoftvert készítettem, ami közel lineáris frekvenciaválaszú, de továbbra is HRTF-ként viselkedő verzióját készíti el a mért eredményeknek.

A demóhoz egy hangszórókhöz is használható audio renderelőt készítettem, amiben különböző keverési módszereket megvizsgálva, a megfelelőt kiválasztva virtuális hangforrások helyezhetők el és mozgathatók a térben. Nem hangforrásonként számoltam ki a HRTF eredményét, hanem a virtuális hangszórókon alkalmaztam az átviteli függvényeket. A hangszórókhöz is gyakran használt arányos keverési módszer érte el a legjobb eredményeket mind fizikai, mind virtuális környezetben. Mindkét esetben fizikai egyenletekkel levezethető, hogy közel ideális környezetben valós eredményt ad. Ezt hangszórók esetében azok elhelyezésének pontos megadása, szimmetriája, és a hangrendszer szobakorrekciója - amihez



9.1. ábra: A visszaverődéseket vizsgáló impulzusválasz-előállító alrendszer nem csak HRTF szimulációjához, de fizikailag helyes visszhangok előállításához is felhasználható. Az ábrán az ezt megvalósító szűrő által bejárt útvonalak láthatók a Unity példajátékát átalakítva.

szintén fejlesztettem szoftvert - jelenti, fejhallgatóknál pedig a HRTF halmazok pontossága, és a hallgató füléhez megfelelő hasonlósága a kritikus tényező.

Hogy elég legyen egyetlen sugarú gömbben rögzített HRTF halmazok alapján is távolságot átadni az arányos keverő képességein túl, vagyis a mérés sugaránál nagyobb távolságokat is virtualizálva, sugárkövetéses módszerrel a hallójárat szimulációját végeztem el. A fejlesztett szoftver beállítható távolságban és sugárszámmal, illetve visszaverődési korláttal egyszerre több szögből vett hallójárat-átviteli függvényt képes előállítani, ezeken túl a kiindulási HRTF szett adott távolságot átadó verzióját is exportálja. Az eredményeket többféle módon elemeztem és összevetettem, ezek alapján feltételezést tettem a hallójárat hangra mért hatásainak mibenlétére. A szimulátor alapjául egy visszaverődések és sugárbeérkezések alapján impulzusválaszokat generálni képes rendszert készítettem, aminek más gyakorlati alkalmazása is létezik, például a 9.1. ábrán bemutatott valós idejű, fizikailag helyes visszhang.

A szimulált hallójárat-átviteli függvényeket a HRTF-ek előtt alkalmaztam, ezzel egy közös, de fülenként a távolság alapján különböző hangerejű, a megfelelő beesési szöget és távolságot szimuláló szűrő állt elő. Bár a várt hatást elérte, nem jelenti azt a minőséget, mint a diplomatervezés közben felfedezett arányos keverő alapján elkészített renderelővel keresztezett HRTF szűrő, vagy akár a hivatkozott hullámtér-szintézisen alapuló módszer, illetve nehezen bizonyítható, hogy az eredmények a valóságnak megfelelnek, mint ahogy a két említett megoldás esetében ez megtörtént. A dolgozat tanulsága a hallójárat hallásra gyakorolt hatásának megismerése, és egy pontos, ámde nem a címben szereplő módszerre alapuló fejhallgató-virtualizáló szoftver elkészülte.

## Hivatkozások

- [1] V. Pukki, *Virtual Sound Source Positioning Using Vector Base Amplitude Panning*, 1997.
- [2] H. Wierstorf, M. Geier, A. Raake, S. Spors, *A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances*, 2011.
- [3] H. Wierstorf, M. Geier, A. Raake, S. Spors, *A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances*, 2011. [Online] Available:  
<http://docs.twoears.eu/en/latest/database/impulse-responses/hrirs/#wierstorf2011>  
[Hozzáférés dátuma: 16 05 2022]
- [4] G. Firtha, *A Generalized Wave Field Synthesis Framework with Application for Moving Virtual Sources*, 2019.
- [5] D. Sung, N. Hahn, K. Lee, *Individualized HRTFs Simulation using Multiple Source Ray Tracing Method*, 2013.
- [6] S. Spagnol, E. Tavazzi, F. Avanzini, *Distance rendering and perception of nearby virtual sound sources with a near-field filter model*, 2016.
- [7] S. Müller, P. Massarani, *Transfer-Function Measurement with Sweeps - Director's Cut Including Previously Unreleased Material and Some Corrections*, 2001.
- [8] S. Li, J. Peissig, *Measurement of Head-Related Transfer Functions: A Review*, 2020.
- [9] GOMeasure, *Head & torso simulators* [Online] Available:  
<https://www.gomeasure.dk/produkt-kategori/audio/head-torso-simulators/>  
[Hozzáférés dátuma: 28 10 2021]
- [10] S. H. Shin, C. L. Istook, *Analysis of Human Head Shapes in the United States*, 2006.
- [11] A. D. Smith, R. J. Ferguson, *Minimum-phase signal calculation using the real cepstrum*, 2014.
- [12] J. Riedmiller, *Dolby AC-4 - Next-Generation Audio*, 2016. [Performance]
- [13] E. Tarsoly, T. Mészáros, *Funkcionális anatómia gyógytornászok számára*, 2011.
- [14] E. Winer, J. Bradley, P. Crum, J. Johnston, *Audio Myths - Defining What Affects Audio Reproduction*, 2009. [Performance]

- [15] ETSI TS 102 366 V1.4.1 - *Digital Audio Compression (AC-3, Enhanced AC-3) Standard*, 2017.
- [16] ETSI TS 103 420 V1.2.1 - *Backwards-compatible object audio carriage using Enhanced AC-3*, 2018.
- [17] A. Noé, *Matroska File Format*, 2009.
- [18] P. Ferreira, *MXF - a technical overview*, 2010.