



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Simulace 3D zvuku pomocí techniky binaurálního audia a doporučovacích systémů
Student:	Bc. David Bernhauer
Vedoucí:	doc. RNDr. Tomáš Skopal, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Analyzujte nejčastější přístupy k simulaci 3D zvuku, diskutujte výhody a nevýhody jednotlivých přístupů. Vyberte vhodné principy doporučovacích systémů a navrhňte s jejich použitím metodu, která podpoří generování personalizovaného binaurálního audia, s cílem zpřesnit simulaci 3D zvuku. S pomocí technologií WebGL a WebAudio implementujte webovou aplikaci (jednoduchou hru) dle navržené metody. Použitou metodu otestujte a pozorujte zlepšení či zhoršení přesnosti v čase.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdlík, CSc.
děkan

V Praze dne 20. února 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Simulace 3D zvuku pomocí techniky binaurálního audia a doporučovacích systémů

Bc. David Bernhauer

Vedoucí práce: doc. RNDr. Tomáš Skopal, Ph.D.

8. května 2017

Poděkování

Chtěl bych poděkovat panu doc. RNDr. Tomáši Skopalovi, Ph.D. za cenné rady, projevenou důvěru, trpělivost a ponechání volnosti. Dále poděkování patří panu Ing. Dominiku Štorkovi za pomoc v oblasti binaurálního audia. V neposlední řadě poděkování patří mé rodině a nejbližším přátelům za podporu během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učením technickým v Praze uzavřel licenční smlouvu o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 8. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 David Bernhauer. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Bernhauer, David. *Simulace 3D zvuku pomocí techniky binaurálního audia a doporučovacích systémů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Cílem této práce je navrhnout metodu efektivního získávání HRTF z databáze již existujících profilů. Práce analyzuje možnosti využití doporučovacíh systémů pro nalezení vhodného HRTF, které by mohly nahradit nejčastěji používané přístupy. Pomocí webové aplikace, implementované technologiemi WebGL a WebAudio, jsme ověřili efektivitu kolaborativního filtrování. Uživatelské testování prokázalo zlepšení přesnosti oproti nepersonalizovanému HRTF profilu průměrně o 10°. Přínosem této práce je nalezení rychlé metody pro relativně přesnou simulaci audio virtuální reality.

Klíčová slova binaurální audio, HRTF, doporučovací systémy, kolaborativní filtrování, virtuální realita, WebGL, Web Audio API

Abstract

The aim of this work is to design a method of efficient HRTF acquisition from the profiles already existed. The thesis analyzes the possibilities of using the recommender systems to find a suitable HRTF that could replace the most commonly used approaches. Using web applications - implemented by technologies WebGL and WebAudio we have verified the effectiveness of collaborative filtering. User testing demonstrated improvement of accuracy against an unpersonalized HRTF profile of an averagely 10 degrees. The conclusion of this work is to find a fast method for relatively accurate audio virtual reality simulation.

Keywords binaural audio, HRTF, recommender systems, collaborative filtering, virtual reality, WebGL, Web Audio API

Obsah

Úvod	1
Cíl práce	1
Motivace	1
Struktura práce	2
1 Simulace 3D zvuku	3
1.1 Jednoduché systémy prostorového zvuku	3
1.2 Systémy založené na HRTF	5
1.3 Problémy simulace	11
2 Doporučovací systémy	13
2.1 Doporučení založené na obsahu	14
2.2 Kolaborativní filtrování	19
2.3 Hodnocení kvality	25
3 Metody doporučování HRTF	27
3.1 Specifické vlastnosti HRTF	27
3.2 Současná řešení	28
3.3 Granularita doporučování	29
3.4 Model testování	32
4 Implementace	37
4.1 Technologie	37
4.2 Návrh	42
4.3 Rozhraní pro rozšíření	45
5 Testování	49
5.1 Příprava	49
5.2 Průběh testování	51
5.3 Vyhodnocení	53

Závěr	61
Zhodnocení	61
Další kroky	62
Budoucí vývoj	62
Literatura	65
A Seznam použitých zkratk	69
B Obsah přiloženého CD	71
C Profily uživatelů	73

Seznam obrázků

1.1	Jednoduchý nákres měření ve studiu	7
1.2	Kužel neurčitosti	8
1.3	Odrazy od ušního boltce	8
1.4	Efekty prostředí	9
1.5	Crosstalk-cancelled stereo	11
2.1	Přesnost a úplnost	26
3.1	Simulace vzdálenějšího objektu	32
3.2	Rozdělení obrazovky pro testování	35
4.1	MVC struktura aplikace	43
4.2	Struktura databáze	45
5.1	Tester LfAp0t, 1. fáze	55
5.2	Tester L6R7Ep, 2. fáze	55
5.3	Porovnání skóre mezi fázemi experimentu	56
5.4	Porovnání průměrné nepřesnosti během prvních 15 minut mezi fázemi experimentu.	57
5.5	Porovnání průměrné nepřesnosti mezi fázemi experimentu.	58
5.6	Porovnání nepřesnosti se zohledněním kolaborativního filtrování	59
C.1	Tester bHHdLS, 2. fáze	73
C.2	Tester hur2Zz, 1. fáze	74
C.3	Tester hur2Zz, 2. fáze	74
C.4	Tester Wn2kOj, 1. fáze	75
C.5	Tester Wn2kOj, 2. fáze	75

Seznam tabulek

3.1	Příklad diskretizace hodnocení	28
4.1	Podpora WebGL v prohlížečích	39
4.2	Podpora WebAudio v prohlížečích	40
5.1	Průměrný počet střel během prvních 15 minut	53
5.2	Průměrná přesnost pokusů ve stupních	54

Úvod

Virtuální realita je často opakované téma nejen současné multimediální scény, ale také oblasti asistivních pomůcek. Zábavní průmysl se zaměřuje především na grafickou část virtuální reality, ale k vytvoření kompletního zážitku je potřeba i simulace audio složky, která trpí nutností personifikace. Doporučovací systémy nabízejí efektivní možnost personifikace na základě jednoduchých i složitějších principů.

Cíl práce

Cílem této práce je analýza nejčastějších procesů získávání HRTF profilu, který slouží jako parametr personifikace audio virtuální reality. Dalším bodem je analýza jednoduchých doporučovacích systémů a nalezení metody, jakým způsobem propojit získávání HRTF a doporučovací systémy, za účelem nalezení kompromisu mezi rychlostí a očekávanou přesností. Praktickou částí je implementace doporučovacího systému a webové aplikace s využitím technologií WebGL a WebAudio. Samotným experimentem je pozorování zlepšení či zhoršení přesnosti.

Motivace

Především v oblasti asistivních pomůcek se očekává od simulace vysoká přesnost. Virtuální realita je z velké části určena k orientaci v prostoru, přičemž takovou možnost ocení především nevidomí. V kombinaci s dalšími nástroji, jako je head-tracking nebo rozpoznávání obrazu, může virtuální realita poskytnout nevidomým zcela jiný pohled na svět. A usnadnit každodenní činnosti bez pomoci okolí.

Tento experiment může posunout využití virtuální reality i v oblasti zábavního průmyslu, vzhledem k širšímu využití zvukových efektů a hlavně lokalizace. Některé počítačové hry již umožňují práci s HRTF simulací zvuku

i bez použití virtuální reality. Existence rychlé a snadné kalibrace parametrů HRTF může být klíčová pro rozvoj oblasti audio virtuální reality, a to i díky její jednoduchosti a dostupnosti.

Struktura práce

V první kapitole analyzujeme techniky simulace prostorového zvuku. Popisujeme výhody a nevýhody jednotlivých přístupů. Zaměřujeme se především na simulaci zvuku pomocí binaurálního audia, techniky založené na HRTF a jednotlivé principy této techniky. Diskutujeme možné problémy simulace a způsoby, jak je eliminovat.

Druhá kapitola uvádí přehled jednoduchých doporučovacích systémů a jakým způsobem můžeme hodnotit jejich kvalitu. Zaměřujeme se především na doporučovací systémy založené na obsahu a kolaborativní filtrování.

Třetí kapitola diskutuje možné přístupy využití doporučovacích systémů pro doporučování HRTF profilů. Rozebírá rozdíly mezi běžnými doporučovacími systémy jako jsou e-shopy a doporučováním HRTF profilů. Poukazuje na současná řešení využívající například neuronové sítě. Popisuje aplikaci různých modelů doporučování a testování.

Čtvrtá kapitola se zabývá implementací webové aplikace. Seznamuje čtenáře s použitými technologiemi a jejich podporou v dnešních prohlížečích. V části návrhu popisuje výběr HRTF databáze a strukturu reálné databáze i aplikace samotné. Rozebírá specifikaci rozhraní, jež musí objekty dodržovat pro integraci doporučovacího systému. To umožňuje dalším čtenářům rozšířit aplikaci a otestovat jiný doporučovací systém.

Poslední kapitola popisuje experiment samotný a jeho výsledky. Popisuje proces přípravy experimentu, jeho průběh i parametry. Interpretuje naměřená data a jejich prostřednictvím poukazuje na zajímavé jevy, které teoretické modely nepodchytí.

Simulace 3D zvuku

Každý den člověk vnímá mnoho různých zvukových vjemů a podvědomě určuje jejich zdroj a jeho pozici. Některá zvířata mají sluch natolik zdokonalen, aby jej byla schopná využít k orientaci v prostoru. U člověka k tomuto účelu běžně slouží především zrak, ale nemusí tomu tak být vždy. Ať už z důvodu, že člověk trpí zrakovým postižením či nemůže v danou situaci využít zrak (tma, telefonní rozhovor, uživatelské rozhraní založené na hlasu, ...), je třeba pro orientaci využít alternativního způsobu, kterým je například zvuk.

Filmy se staly velice rozšířeným způsobem zábavy, zprostředkovávaný především obrazovou a zvukovou informací. Původní němé filmy přinášely divákům pouze obraz. Stejně tak i počítačové hry a virtuální realita v prvních krůčcích svého vývoje přinášely pouze obrazovou informaci. S rozvojem technologií se postupně zvyšovala poptávka po rozšíření zážitku i pomocí dalších smyslů. Díky rozšíření osobních počítačů a přítomnosti zvukových karet je zvuk ideálním kandidátem ubírající se správným směrem.

Asistivní pomůcky i oblast zábavy jsou jedním z mnoha důvodů, proč se zabývat simulací prostorového zvuku. Obzvláště v případě asistivních pomůcek je nutná dostatečná přesnost simulace. Cílem není jen zprostředkovat základní obsahovou informaci, ale zajistit kompletní zvukovou informaci včetně pozice zdroje.

V první části se budeme věnovat popisu jednoduchých systémů prostorového zvuku, jenž jsou dnes široce používané, avšak pro naše potřeby málo efektivní. V druhé části se přesuneme k metodám založeným na HRTF, které se dokáží vypořádat s potřebami přesné simulace 3D zvuku. Třetí část se zaměřuje na problémy spojené se simulací prostorového zvuku.

1.1 Jednoduché systémy prostorového zvuku

S nejjednoduššími principy simulace prostorového zvuku se setkáváme především ve filmové části zábavního průmyslu. Takové systémy prostorového zvuku využívají jednoduché myšlenkové postupy nebo umožňují simulaci zvuku za

určitých omezujících podmínek (např. pozice zdroje, cenová dostupnost). Kvůli své jednoduchosti se jedná o nejčastější přístupy užívané v zábavním průmyslu. Následující přehled uvádí, některé z těch nejběžnějších přístupů, a popisuje jejich výhody a nevýhody.

1.1.1 Stereofonní systém

Základním způsobem simulace zvuku je využití stereofonního systému, tedy dvou reproduktorů umístěných nejčastěji před posluchačem. Stereofonní systém dokáže jednoduše simulovat zvuk ze dvou pozic v prostoru (umístění reproduktorů) a přibližně aproximovat libovolnou pozici mezi reproduktory, změnou poměru hlasitostí obou reproduktorů. Nevýhodou stereofonního systému je simulace pouze přibližné pozice a to pouze na spojnici mezi reproduktory.[1] Velkou výhodou je nízká cena reproduktorů, široká dostupnost a podpora různých zvukových formátů.

Pokud bychom chtěli využít metodu HRTF (vizte sekci 1.2) pro stereofonní systém reprezentovaný reproduktory, můžeme využít metodu *crosstalk-cancelled* stereo, vizte 1.2.4.1. Aby metoda byla efektivní vyžaduje znalost přesné polohy posluchače před reproduktory[1], případně je potřeba přepočítat zvukové signály dle sledovacího zařízení, což metodu činí nepoužitelnou pro převážnou většinu případů užití.

1.1.2 Multikanálový systém

Náhradou dvou větších reproduktorů za větší počet menších reproduktorů získáme možnost simulovat zvuk z většího množství pozic v prostoru. Pro nízké frekvence se použije nesměrový subwoofer. Tohoto systému využívá technologie Dolby Pro Logic. Nejčastěji se s tímto systémem setkáváme v kinosálech.[1] Vzhledem k poměrně vysoké pořizovací ceně a náročnému přemísťování celého systému se nejedná o vhodného kandidáta pro běžnou simulaci prostorového zvuku.

1.1.3 Binaurální nahrávka

Jedná se o jednu z pokročilejších metod „simulace“. Umělé figuríně (lze nahrávat i s živým posluchačem, jak uvádí [2]) vložíme do uší miniaturní mikrofony, kterými nahrajeme okolní prostředí. Zvuková informace může být doprovázena i obrazovou informací v podobě videa. Posluchač poté použije sluchátka, čímž zachycený zvuk umístí do stejné pozice, v jaké byly při nahrávání mikrofony, a má tak k dispozici podobný zážitek, jaký by měl na místě figuríny v době nahrávání scény.

Nahrávka je neměnná, tudíž se nehodí na následnou simulaci libovolného zvuku. K tomu, aby zážitek byl zcela totožný, je potřeba, aby figurína měla podobnou stavbu těla, hlavy a uší jako následný posluchač, v opačném případě bude zvuková informace (především pozice zdroje) zkreslená. Nevýhodou

tohoto přístupu je nutnost použití stereofonních sluchátek (náhrada za reproduktory je možná, avšak problematická, vizte sekci 1.1.1).[1]

1.2 Systémy založené na HRTF

HRTF je zkratka pro anglický výraz *head-related transfer function*, v českém překladu přenosová funkce vztažená k hlavě. Myšlenka HRTF je založena na poměrně jednoduchém principu. Podobně jako u binaurální nahrávky předpokládáme, že zvuk zdroje v prostoru můžeme zachytit na nějakém jiném místě v prostoru, a z tohoto místa jej následně reprodukovat dle zachycené informace. Zobecněním tohoto principu předpokládáme, že máme funkci, jejíž parametry jsou pozice zdroje vůči posluchači a jeho monofonní zvukový signál (frekvenční spektrum) a jejíž návratová hodnota je stereofonní zvukový signál, který bychom mohli zachytit s mikrofonom umístěným v uších posluchače. Takové funkci budeme říkat HRTF a slouží jako frekvenční filtr, díky němuž poznáme směr zvuku, podobně jako je tomu u ušního boltce.

Výhoda metod založených na HRTF je možnost simulovat zvuk z libovolné pozice v prostoru s poměrně velkou přesností (v závislosti na kvalitě HRTF). Nevýhodou těchto metod je potřeba zkonstruovat nebo naměřit HRTF s tím, že v případě měření je nutné naměřit všechny pozice potřebné pro simulaci (ostatní pozice budou dopočítány a nemusí být dostatečně přesné). Navíc podobně jako je tomu u binaurální nahrávky, uživatel musí použít sluchátka. V první části popisujeme možnosti získávání HRTF včetně jejich hlavních výhod a nevýhod, dle [3]. Druhá část se zaměřuje na popis samotného měření HRTF ve studiu. Třetí část popisuje jednotlivé principy HRTF a modely využívané v HRTF modelování.

1.2.1 Možnosti získávání HRTF

Poměrně snadným a levným způsobem využívaným především v zábavním průmyslu je použití nějakého standardního HRTF. Vzhledem k neexistenci žádného standardu[3] se využívá figurína člověka s mikrofony v uších, na které se naměří HRTF. Místo celé figuríny se může použít pouze umělá hlava či část figuríny (např. KEMAR). Výhodou je jednoduchý a případně i automatizovaný způsob měření HRTF z libovolného množství poloh. Alternativou je zprůměrování většího množství HRTF profilů. Nevýhodou předmětného principu je však značná nepřesnost, a z tohoto důvodu je pouze zábavní průmysl, kde není třeba přesná lokalizace zvuku, ideálním kandidátem pro použití takového HRTF profilu. Naopak použití takové HRTF v rámci asistivních pomůcek je zcela nedostatečné, právě kvůli možné nepřesnosti.

Přesným opakem předchozí metody je naměření individuálního HRTF pro každého posluchače. Takové měření může být časově náročné (vizte v 1.2.2) a pro posluchače často nepohodlné. Na druhou stranu takto naměřené HRTF přináší extrémně přesné výsledky.

Třetí metoda namísto přímého měření HRTF předpokládá závislost fyziologie člověka a HRTF. Zahrnuje tak měření částí těla a hlavy mající největší vliv na tuto funkci. K takovým antropometrickým bodům patří především velikost a tvar hlavy, tvar uší a velikost ramen. Následně z těchto hodnot je HRTF vygenerována matematickým modelem. Jednotlivé části tohoto matematického modelu jsou naznačeny v sekci 1.2.3. Podrobnosti k matematickému modelování lze nalézt na [3] a v [4]. Podstatnou výhodou této metody je možnost regulovat přesnost simulace (ne)zanedbáním jednotlivých částí. Nevýhodou je nutnost měření antropometrických bodů, které stále vyžaduje odbornou asistenci.

Poslední metoda a zároveň metoda, kterou se zabýváme v této práci, je vytvoření databáze HRTF profilů pro několik reprezentantů populace. Následně si uživatel může vybrat HRTF profil dle vlastní preference nebo mu může být přiřazen na základě získaných informací o uživateli. Od této metody si slibujeme lepší přesnost než od jediného HRTF, a naopak nižší časovou náročnost než při individualním měření.

1.2.2 Měření HRTF

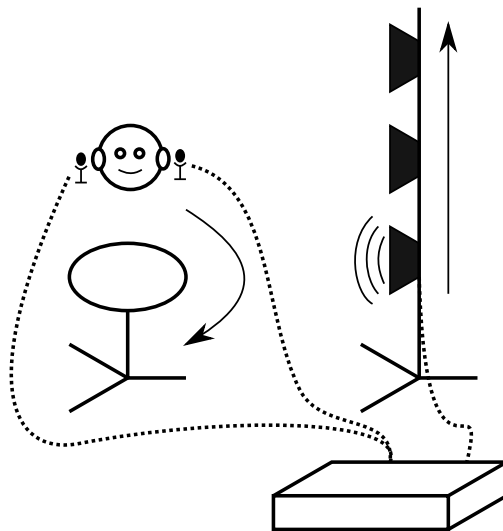
HRTF není možné měřit přímou metodou, v praxi se měří tzv. *head-related impulse response*, v českém překladu impulsová odezva vztažená k hlavě. Jedná se o skutečnou změnu signálu, a HRTF je její Fourierovou transformací.[5] Teoreticky je HRIR impulsová odezva ucha na Diracův impuls, v praxi se využívá bílý šum.[6]

Samotné měření může probíhat dvěma způsoby. Základní myšlenkou je umístění miniaturních mikrofónů do uší měřených subjektů a umístění reproduktoru do požadované polohy (sférická soustava souřadnic). Pro urychlení měření se dá nainstalovat více reproduktorů vedle sebe nebo nad sebe a posluchače posadit na židli s automatickým krokovým otočením. V takovém případě stačí postupně spustit každý reproduktor a otočit posluchačem, který musí zůstat po celou dobu na židli ve stejné poloze.[7]

Pokročilejší myšlenkou je otočení celého principu měření, vložení miniaturních reproduktorů do uší subjektu a umístění mikrofónů do všech měřených pozic. Tento přístup je náročnější, reproduktory nesmí být příliš výkonné, aby neublížili posluchači samotnému. Z tohoto důvodu je nutné zajistit zvukově odstíněnou místnost a kvalitní mikrofony. Dalším problémem mohou být odrazy od samotné místnosti, které je také nutné eliminovat před měřením. Tato varianta je mnohonásobně rychlejší, ale zároveň se jedná o poměrně drahé řešení kvůli nezbytnému vybavení.[7]

1.2.3 Základní principy a modelování HRTF

Modelování HRTF probíhá většinou na bázi několika základních principů fungování HRTF. Jednotlivé principy jsou seřazeny od jednodušších myšlenek po



Obrázek 1.1: Nákres běžného měření ve studiu, židle s automatickým krokovým natočením, 3 postupně spouštěné reproduktory.

komplexnější. Jejich kombinace dává základ pro fungování metod založených na HRTF. Některé metody simulace využívají jen některé části, jiné naopak kombinaci všech dohromady.

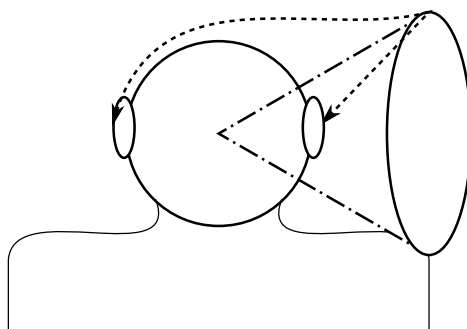
1.2.3.1 ILD a ITD

S modelováním ILD (interaural level difference) se můžeme setkat v případě pozicování zvuku stereofonním systémem. Pokud je zdroj blíže k levému (resp. pravému) uchu, pak by signál v levém (resp. pravém) uchu měl být hlasitější. Hlava samotná pak slouží jako překážka. Tento princip bývá nejčastěji doplňován modelováním ITD (interaural time difference), kdy navíc pracujeme s myšlenkou zpoždění levého a pravého kanálu v závislosti na pozici zdroje.

Samotné použití těchto dvou principů je stále nedostatečné pro určení přesné polohy. Pro zjednodušení si můžeme hlavu člověka představit jako kouli a uši budou dva protilehlé body. Pak existuje vždy kužel s vrcholem ve středu koule a kolmice, která spojuje body reprezentující uši. Body na kuželu v rámci jedné kružnice mají ILD a ITD shodné. Tomuto jevu se říká kužel neurčitosti.

1.2.3.2 Odraz zvuku od ramen

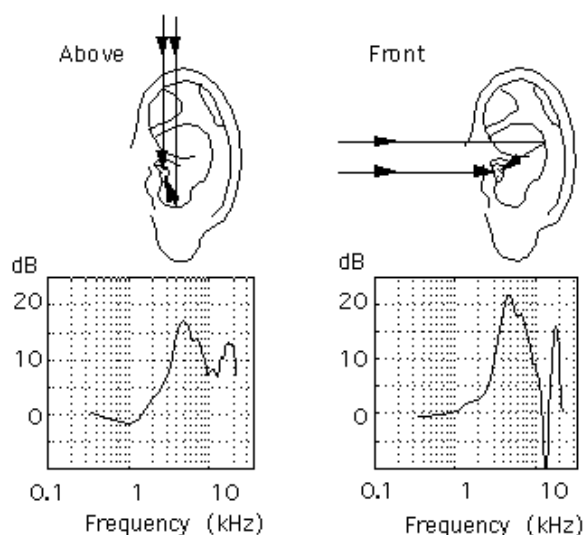
Pokud bychom chtěli rozpoznat, zda se pozice zvuku nachází nad úrovní hlavy či naopak, můžeme k tomu jednoduše využít odraz zvuku od ramen subjektu. Ucho tak zaznamená zpožděný odraz, tato informace nám však přidá informaci jen o pozici objektu v rovině mediální.



Obrázek 1.2: *Kužel neurčitosti*. Kružnice bodů, které mají k posluchači stejné hodnoty ILD a ITD.

1.2.3.3 Ušní boltce jako filtr

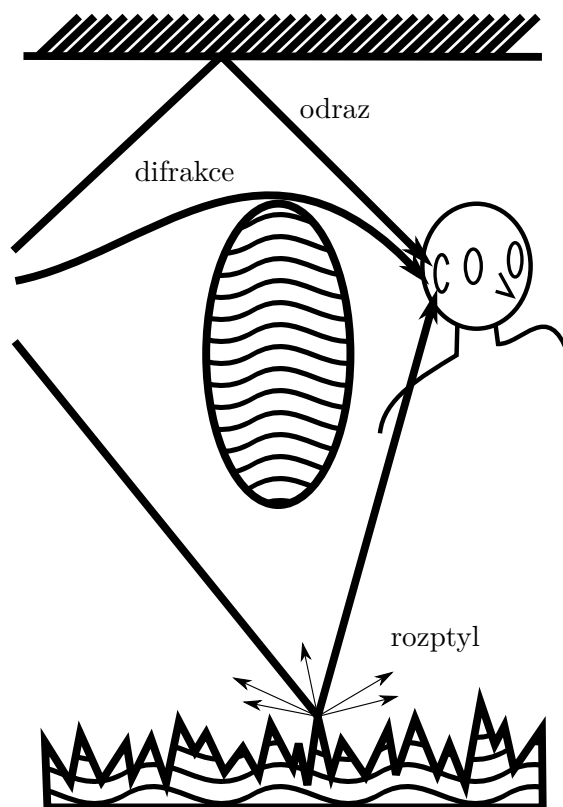
Nejkomplikovanější částí celého procesu, která zároveň nejvíce ovlivňuje vnímání pozice zvuku a je hlavní příčinou unikátnosti HRTF, je vnější ucho. Mnohonásobné odrazy od částí ušního boltce slouží jako jednoduchý frekvenční filtr. Díky vnějšímu uchu je člověk schopen lokalizovat většinu zvuků i bez nutnosti otočení hlavy. Tento efekt je nejvýraznější pro frekvence okolo 10 kHz (tzv. *pinna notch*).



Obrázek 1.3: *Odrazy od ušního boltce*. Kolem 10 kHz si můžeme všimnout velkého útlumu, který se liší pro různé směry. Převzato z [8].

1.2.3.4 Okolní prostředí

Odrazy zvuku od okolního prostředí jsou někdy zahrnuty již do nahrávaného HRTF. Musíme tak měřit HRTF již ve vybraném prostředí. To může být komplikované i z hlediska výběru prostředí. Častěji se však využívá možnosti uchopit odrazy od prostředí jako další slabší zdroj původního zvuku. V této souvislosti se bavíme o tzv. *room impulse response*, česky impulsové odezvě místnosti. V případě M subjektů a N prostředí bychom standardně potřebovali naměřit celkem $M \cdot N$ HRTF profilů, přičemž díky možnosti skládat tyto zvuky dohromady nám stačí pouze $M + N$ HRTF profilů. Navíc můžeme pracovat s RTF (room transfer function) nezávisle na posluchači.



Obrázek 1.4: Jednotlivé efekty prostředí můžeme simulovat nezávisle na posluchači aplikováním RTF.[5]

1.2.4 Matematika HRTF

Jak už jsme neznali, HRTF je Fourierovou transformací HRIR. Otázkou je, proč je potřeba převádět HRIR na HRTF. Předpokládejme, že $h_R(t)$ je impulsová odezva v rovině časové (tedy se jedná o HRIR) pro pravý zvukový kanál, resp. $h_L(t)$ pro levý zvukový kanál. Nechť $x(t)$ popisuje originální monotónní zvukový signál, poté $x_L(t)$ a $x_R(t)$, výsledný zvukový signál pro levé a pravé ucho, získáme jako konvoluci (1.1) originálního signálu a HRIR.[9]

$$x_{L,R}(t) = h_{L,R}(t) * x(t) = \int_{-\infty}^{\infty} h_{L,R}(t - \tau) x(\tau) d\tau \quad (1.1)$$

Pomocí FFT (rychlé Fourierovy transformace) převedme původní signál a HRIR do frekvenčního spektra a získáváme $X(\omega)$ jako frekvenční spektrum zdroje a $H_L(\omega)$ jako HRTF pro levý zvukový kanál. Ve frekvenční doméně je konvoluce převedena na násobení. Získáváme tedy (1.2).[9]

$$X_{L,R}(\omega) = \mathcal{F}(h_{L,R}(t) * x(t)) = H_{L,R}(\omega) X(\omega) \quad (1.2)$$

Vzhledem k diskretní podobě naměřených HRIR je místo klasické konvoluce a FFT nutné použít jejich diskretní variantu. Asymptotická časová složitost diskretní konvoluce je $\mathcal{O}(N^2)$, kde N je velikost vzorku. Asymptotická časová složitost DFT je $\mathcal{O}(N \log(N))$, samotné násobení pak pouze $\mathcal{O}(N)$. Z toho plyne, že využití DFT pro převod do frekvenční domény a zase zpět je výhodnější.

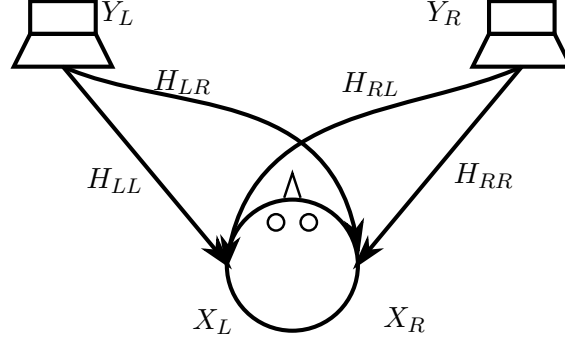
1.2.4.1 Crosstalk-cancelled stereo

Simulovat HRTF je teoreticky možné i pomocí stereofonních reproduktorů. V případě stereofonních sluchátek je problém jednodušší v tom, že pro každé ucho zvlášť přichází právě jeden zvukový signál. V případě stereofonních reproduktorů přichází do každého ucha 2 signály.

Na diagramu 1.5 vidíme základní myšlenku kombinace dvou zvukových signálů $X_L = H_{LL}Y_L + H_{RL}Y_R$. Kde Y_L je signál levého reproduktoru a Y_R signál pravého reproduktoru, a X_L výsledný signál. H_{LR} značí HRTF funkci pro levý reproduktor a pravé ucho. Zjednodušeně můžeme znázornit tuto rovnici jako násobení dvou matic, rovnice 1.3.

$$\begin{pmatrix} H_{LL} & H_{LR} \\ H_{RL} & H_{RR} \end{pmatrix} \begin{pmatrix} Y_L \\ Y_R \end{pmatrix} = \mathbb{H} \begin{pmatrix} Y_L \\ Y_R \end{pmatrix} = \begin{pmatrix} X_L \\ X_R \end{pmatrix} \quad (1.3)$$

Pokud chceme simulovat prostorový zvuk pomocí stereofonních reproduktorů, musíme nejprve spočítat očekávaný signál X_L a X_R . Poté musíme spo-

Obrázek 1.5: Náskres principu *crosstalk-cancelled* stereo.

čítat inverzi matice $\mathbb{H}$. A spočítat výstupní signály reproduktorů Y_L a Y_R dle rovnice 1.4. Stále přetrvává problém pohybujícího se posluchače.

$$\begin{pmatrix} Y_L \\ Y_R \end{pmatrix} = \mathbb{H}^{-1} \begin{pmatrix} X_L \\ X_R \end{pmatrix} \quad (1.4)$$

1.3 Problémy simulace

Následující část popisuje příklady možných problémů spojených se simulací zvuku. Některé z nich nastaly během testování aplikace, a je na ně potřeba adekvátně reagovat či s nimi počítat.

Nejčastějším problémem simulace s použitím HRTF je nepřesnost. Ta zahrnuje i přirozenou nepřesnost, kdy člověk není schopen pomocí sluchu přesně lokalizovat zdroj. Dalšími faktory ovlivňující nepřesnost je použití nevhodného HRTF profilu nebo chyba při měření HRTF (subjekt se pohnul). Dalším vedlejším faktorem je možnost *miss-clicku* (překliknutí) v rámci testovací aplikace, vizte 5.3.2.

Druhým neméně častým problémem je nemožnost lokalizace pozice vpředu a vzadu. Ta je většinou způsobená použitím špatného HRTF nebo nevhodným ovládáním testovací aplikace. Přirozený pohyb pro lokalizaci zvuku je natočení hlavy, to však bez použití zařízení pro *head tracking* není možné a může tak vznikat zmatení uživatele, jenž pohne hlavou a očekává změnu zvuku. Na měřených datech je toto prohození zřetelné a dá se případně řešit použitím absolutních hodnot kosinů. Takové řešení nám neumožňuje rozeznat, zda se jednalo o skutečně špatný nebo dobrý profil s neúmyslnou chybou.

Vzhledem k různé citlivosti funkce HRTF na různé frekvence je některé zvuky snazší lokalizovat než jiné. Občas si můžeme všimnout, že některé zvuky (např. pípající telefon) se nám může jevit jako by měl zdroj všude okolo nás. Z tohoto důvodu je důležité vybrat pro simulaci takový zvuk, který se bude dobře lokalizovat.

Jelikož testované osoby využívají k lokalizaci především zrak, nejsou zvyklé orientovat se v prostoru pouze pomocí zvuku. To může znamenat počáteční problémy a následné zlepšení, na mnoha subjektech je pozorováno průměrné zlepšení přesnosti s postupem času i v první fázi testování, vizte 5.3.2. Při porovnávání výsledků může být nutné porovnávat přibližně stejný časový úsek (nebo počet pokusů).

Doporučovací systémy

Lidé odjakživa hledali alternativy k již známým věcem. Pokud například pravěký muž snědl rostlinu a následně mu bylo nevolno, pak takovou rostlinu a jí podobné napříště již nezkoušel a varoval i své druhy. V dnešní době můžeme pro změnu řešit obdobné dilema, líbí-li se nám konkrétní píseň a rádi bychom našli další podobné nahrávky, nemusí to být zcela snadný úkol. Naopak si nevybíráme interprety našeho neoblíbeného žánru. Obecně řečeno, pokud máme s danou věcí určitou zkušenost, pak očekáváme obdobnou od podobně vypadajících věcí. Takovéto chování bychom běžně nazvali *předsudkem*, nicméně se jedná o základní způsob, jakým člověk hodnotí pro něj neznámý objekt na základě předchozích zkušeností.

Mnoho našich preferencí sdílíme s našimi přáteli či obecně zájmovou skupinou. Spoléháme, že jejich doporučení díky podobným zájmům budou odpovídat našemu očekávání. Ne vždy se musí jednat o skupinu lidí nám blízkých, internetový obchod se nám zdá důvěryhodnější, pokud si můžeme přečíst pozitivní reference. Naše rozhodování je ovlivňováno hodnocením pro nás *důvěryhodných* osob.

„Cílem doporučovacích systémů je generovat množině uživatelů smysluplná doporučení věcí nebo produktů, které by je mohly zajímat.“ [10, s. 1] Teoreticky můžeme doporučovací systémy po správném nastavení parametrů aplikovat na libovolný problém optimálního výběru a očekávat pozitivní zpětnou vazbu od uživatele. Doporučovací systémy dle [10, s. 2] řadíme do tří kategorií podle principu, na kterém fungují:

- založené na obsahu,
- kolaborativní filtrování,
- hybridní.

[11] uvádí navíc další podobné kategorie, avšak lišící se mírně ve svém významu. Tyto kategorie jsou zmíněny v sekcích 2.1.3 a 2.2.4. V dalších částech této kapitoly budeme pracovat s následujícími definicemi.

Objekt Objekty (anglicky *item*) jsou věci, které doporučujeme prostřednictvím doporučovacího systému. Objekt může být charakterizován svými vlastnostmi. Množinu všech objektů budeme označovat $\mathcal{I}$.

Uživatel Uživatel (anglicky *user*) je osoba, které jsou doporučovány objekty na základě preferencí uživatele. Množinu všech uživatelů budeme označovat $\mathcal{U}$. V některých případech není nutné pracovat s uživatelem jako s fyzickou osobu. Například pro simulaci běhu doporučovacího systému nám stačí pracovat s personou (popis chování) daného uživatele.

Hodnocení Hodnocení (anglicky *rating*) objektu i uživatelem u budeme označovat r_{ui} . Hodnocení je libovolné reálné číslo, pokud nebude řečeno jinak, potom kladná hodnota indikuje pozitivní vztah k objektu (oblíbenost, využitelnost uživatelem, cena pro uživatele) a záporná hodnota naopak negativní. Množinu všech objektů hodnocených uživatelem u budeme označovat $\mathcal{I}_u$.

2.1 Doporučení založené na obsahu

Základní podmínkou doporučvacích systémů založených na obsahu je přítomnost vlastností (anglicky *features*) doporučovaných objektů. Tyto vlastnosti mohou být obsaženy v samotném objektu (barevný histogram obrázku, frekvence slov v textu, technické parametry telefonu, ...) nebo mohou být doplněny metainformacemi dodanými uživateli či jiným zdrojem (místo a čas pořízení fotografie, hodnocení produktu významnou autoritou). Problémem metainformací je především otázka jejich důvěryhodnosti, podobně jako tomu je u kolaborativního filtrování, vizte 2.2.3.

Nejjednodušší variantou je doporučování nejpodobnějších objektů k jednomu konkrétnímu objektu (právě prohlížený produkt), což je poměrně nepraktické. V praxi se častěji používá silnější varianta, kdy uživatel může ohodnotit více objektů, a my tak můžeme pracovat s přesnějšími preferencemi. [12, s. 77] uvádí celkem 3 hlavní přístupy k získávání explicitního hodnocení.

Binární hodnocení Objekty jsou ohodnoceny binární hodnotou 1 nebo 0 s daným významem líbí nebo nelíbí (relevantní nebo nerelevantní, ano nebo ne, ...).

Hodnocení na škále Objekty jsou hodnoceny na škále diskretních čísel, případně mohou být čísla nahrazena za textové ekvivalenty (např. Likertova škála). Hodnocení může být vzestupné (hvězdičky) nebo sestupné (školní známkování).¹

¹Pokud máme sestupné hodnocení, pak v systému musíme předpokládat, že záporné hodnocení je lepší.

Textové hodnocení Objekty jsou okomentovány v přirozeném jazyce. Takový přístup již vyžaduje zpracování přirozeného jazyka, identifikaci kladných a záporných významů, což nemusí být snadný úkol (identifikace na základě jednotlivých slov může být nedostačující, například hovorový výraz „strašně dobré“ může být po slovech interpretován pozitivně i negativně). Uživateli však může textové hodnocení poskytnout mnohem detailnější hodnocení objektu, přičemž samotná interpretace je přenechána uživateli.²

Hodnocení může být také implicitní. Mezi takové řadíme například úspěšný nákup v elektronickém obchodě. Příklad negativního implicitního hodnocení může být vrácení zboží v dvoutýdenní lhůtě. Na webových stránkách budeme za implicitní hodnocení považovat odkazy, na které uživatelé klikají. Stránka, na které setrvávají déle, bude mít lepší hodnocení. Takový přístup může mít problémy s objektivitou hodnocení (v případě doby setrvání na stránce, např. nečekaný telefonát prodlužující dobu setrvání).

Následně můžeme z množiny hodnocení uživatele a vektoru vlastností objektů spočítat pomocí agregační funkce vektor preferencí. Jednoduchou agregační funkcí může být například (2.1) dle [13, s. 110], v některých případech může být vhodnější její normovaná verze (2.2), kde $\mathbf{x}_u$ je vypočtený profil uživatele (preference), $\mathbf{x}_i$ je vektor vlastností objektu i .

$$\mathbf{x}_u = \sum_{i \in \mathcal{I}_u} r_{ui} \mathbf{x}_i \quad (2.1)$$

$$\mathbf{x}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{ui} \mathbf{x}_i}{\sum_{i \in \mathcal{I}_u} r_{ui}} \quad (2.2)$$

Posledním krokem procesu³ je najít nejpodobnější objekty. Uživateli můžeme doporučit jeden konkrétní objekt nebo celou skupinu objektů na základě zvolené vzdálenostní funkce⁴ d . Objekty můžeme dle [14, s.15–17] nejjednodušeji vybrat třemi způsoby:

Rozsahový dotaz Zvolíme akceptovatelný rozsah (anglicky *range*) podobnosti. Všechny objekty spadající do tohoto rozsahu budeme uvažovat ve výběru objektů k doporučení. Formálně může být definována jako (2.3), kde r je zadaný rozsah. Nevýhodou je nutná znalost podobnostní funkce

²Některé aplikace (např. www.alza.cz) umožňují ohodnocení (textových) hodnocení, tím lze ohodnotit jeho kvalitu a vliv (uživatelé jsou prezentováni nejprve nejrelevantnější komentáře).

³Celý proces je iterativní, následně od uživatele obvykle očekáváme zpětnou vazbu ve formě dalšího hodnocení, které může sloužit pro vypočtení nových doporučení. A celý proces se opakuje.

⁴Vzdálenostní funkce je opakem podobnostní funkce. Objekty mající vzdálenost rovnou 0 jsou identické, čím větší vzdálenost, tím méně podobné si objekty jsou.

a dat, případně opakované dotazování, abychom se vyhnuli situaci, kdy parametr r bude příliš restriktivní či naopak.

$$R(\mathbf{x}_u, r) = \{i \in \mathcal{I}, d(\mathbf{x}_i, \mathbf{x}_u) \leq r\} \quad (2.3)$$

kNN dotaz Problém rozsahového dotazu můžeme vyřešit použitím kNN dotazu (2.4). Z množiny všech objektů vybereme k nejvíce podobných objektů (nejbližších sousedů) k profilu uživatele. Nevýhodou kNN dotazu je akceptace vždy k objektů nehledě na jejich podobnost.

$$kNN(\mathbf{x}_u) = \{R \subseteq \mathcal{I}, |R| = k \\ \wedge \forall i \in R, j \in \mathcal{I} - R : d(\mathbf{x}_u, \mathbf{x}_i) \leq d(\mathbf{x}_u, \mathbf{x}_j)\} \quad (2.4)$$

kRNN dotaz Reverzní kNN dotaz nám může poskytnout informaci o tom, pro které objekty by se profil uživatele nacházel ve výsledku jejich kNN dotazu. (2.5) Získáme tak i vzdálenější izolované objekty, a naopak se vyhneme podobným shlukům dat.

$$kRNN(\mathbf{x}_u) = \{R \subseteq \mathcal{I}, \forall i \in R : \mathbf{x}_u \in kNN(\mathbf{x}_i) \\ \wedge \forall j \in \mathcal{I} - R : \mathbf{x}_u \notin kNN(\mathbf{x}_j)\} \quad (2.5)$$

Následující sekce popisují, jak se vypořádat s textově orientovanými systémy a jakým způsobem převést slova a dokumenty na číselnou reprezentaci. V druhé části jsou uvedeny populární podobnostní míry uplatňované pro podobnostní vyhledávání. V poslední části uvádíme alternativní přístupy k doporučení na základě obsahu a jejich rozdíly.

2.1.1 Textově orientované doporučovací systémy

Textově orientované doporučovací systémy nebudou naší prioritou, uvádíme je pouze pro obecný přehled. Při hledání podobnosti mezi HRTF profily můžeme využít některé myšlenky. „Většina doporučovacích systémů založených na obsahu používá poměrně jednoduché modely jako je shoda klíčových slov nebo model vektorového prostoru (VSM, anglicky Vector-Space Model) se základním TF-IDF ohodnocením.“ [12, s. 81] „VSM reprezentuje dokumenty a jednotlivé dotazy jako vektor vah. Každá váha udává míru důležitosti daného termu v dokumentu nebo dotazu.“ [15, s. 3259] Termy představují důležitá slova po předzpracování, které může zahrnovat tokenizaci, odstranění příliš běžných slov, lematizaci a jiné.

TF-IDF ohodnocení se skládá ze dvou částí. Frekvence termu (dále jen TF, anglicky *term frequency*) v dokumentu určuje poměr četnosti termu vůči četnosti nejčastějšího termu v rámci jednoho dokumentu, frekvenci termu t_k v dokumentu d_j můžeme vyjádřit jako (2.6), kde $f_{k,j}$ je počet výskytů termu v dokumentu. Nejčastěji vyskytující se term v dokumentu má tedy $TF(t_k, d_j) = 1$.

$$TF(t_k, d_j) = \frac{f_{k,j}}{\max_z f_{z,j}} \quad (2.6)$$

Inverzní frekvence v dokumentech (dále jen IDF, anglicky *inverse document frequency*) určuje významnost termu napříč dokumenty. IDF vychází z myšlenky, že term vyskytující se pouze v jednom dokumentu bude jistě významější než běžně vyskytující se slova. Definovat ji můžeme jako (2.7), kde N je celkový počet dokumentů a n_k je počet dokumentů, ve kterých se term vyskytuje alespoň jednou.

$$IDF(t_k) = \log \frac{N}{n_k} \quad (2.7)$$

Samotné TF-IDF se spočítá jako součin těchto dvou funkcí.[12, s. 81–82]

$$TF - IDF(t_k, d_j) = TF(t_k, d_j) \cdot IDF(t_k) = \frac{f_{k,j}}{\max_z f_{z,j}} \cdot \log \frac{N}{n_k}$$

2.1.2 Vzdálenostní míry

Základem kvalitního doporučovacího systému, který je založen na obsahu, je správná volba vzdálenostní míry. Podobnostní míra určuje jak moc je daný objekt podobný, tedy vyšší hodnota je ekvivalentní vyšší podobnosti. U vzdálenostní (nebo také distanční) míry je tomu naopak, vzdálenost 0 tudíž implikuje, že se objekty neliší. Jednoduchý převod mezi podobností s_{ij} a vzdáleností d_{ij} můžeme provést například následujícím výpočtem:

$$s_{ij} = 1 - d_{ij}$$

Následující část ukazuje skromný přehled několika málo vzdáleností, které se svým významem mohou pojit k HRTF doporučování. Pro vzdálenostní míry většinou definujeme následující podmínky[16, s. 60]:

- $d_{ij} \geq 0$, vzdálenost mezi dvěma objekty je nezáporná,
- $d_{ii} = 0$, vzdálenost je 0 právě tehdy, když porovnávané objekty jsou identitické,
- $d_{ij} = d_{ji}$, vzdálenost je symetrická,
- $d_{ij} \leq d_{ik} + d_{kj}$, vzdálenost splňuje trojúhelníkovou nerovnost.

Pokud vzdálenostní míra splňuje všechny výše zmíněné podmínky, pak ji nazveme metrickou vzdáleností nebo metrikou.

Mezi základní vzdálenosti pro číselné hodnoty řadíme Euklidovskou metriku (2.8), která je speciálním případem Minkowského vzdálenosti (2.9). [16, s. 68] Takové vzdálenosti bychom využili především pro naměřené hodnoty antropometrických bodů.

$$d_{ij} = \sqrt{\sum_{k=1}^N (x_{ik} - x_{jk})^2} \quad (2.8)$$

$$d_{ij} = \sqrt[\lambda]{\sum_{k=1}^N |x_{ik} - x_{jk}|^\lambda} \quad (2.9)$$

Na HRTF profily se můžeme dívat z několika úhlů. První možnost je uvažovat HRTF pouze jako funkci. Případně se můžeme dívat na HRIR jakožto na zvuk nebo polynom. Pro přehled funkcí k tomuto tématu můžeme využít [17].

My se zaměříme na HRIR (příp. HRTF) jako časovou sekvenci. K porovnávání podobností (nejen časových) sekvencí se často užívá DTW (anglicky *dynamic time warping*). DTW se snaží o nalezení zarovnání dvou sekvencí takové, aby mělo minimální cenu. [18]

Cena c je definována jako vzdálenostní míra pro hodnoty naší sekvence. Sekvence tedy nemusí být pouze číselná. Definujme si cenovou matici $\mathbb{C} \in \mathbb{R}^{N \times M}$ jako $\mathbb{C}(n, m) = c(x_n, y_m)$, kde $X = (x_1, \dots, x_N)$ a $Y = (y_1, \dots, y_M)$ jsou dvě sekvence. [18]

Definujme si zarovnání jako sekvenci $p = (p_1, \dots, p_L)$, kde $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ s následujícími podmínkami: [18]

- $p_1 = (1, 1)$ a $p_L = (M, N)$
- $n_1 \leq n_2 \leq \dots \leq n_L$ a $m_1 \leq m_2 \leq \dots \leq m_L$
- $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$

Cena zarovnání p je definována jako $c_p(X, Y) = \sum_{l=1}^L c(x_{n_l}, y_{m_l})$. DTW vzdálenost je pak definována jako (2.10). Algoritmus pro nalezení optimálního zarovnání s asymptotickou složitostí $\mathcal{O}(MN)$ můžeme nalézt v [18].

$$\text{DTW}(X, Y) = \min\{c_p(X, Y) \mid c_p \text{ musí být zarovnání}\} \quad (2.10)$$

Další vzdálenostní míry můžeme nalézt v [17].

2.1.3 Alternativní přístupy

Podobným přístupem může být doporučování založené na znalosti konkrétní domény (anglicky *knowledge-based*). Nehledáme konkrétní podobnost mezi dvěma objekty na základě hodnocení uživatele. Místo toho hledáme jak moc je daný objekt užitečný uživateli či jak moc odpovídá očekávání uživatele dle jeho preferencí. To vyžaduje podrobnou znalost doporučovaných objektů (domény problému), vztahu objektů k preferencím uživatele (jak moc jsou pro něj užitečné) a samotné preference uživatele. Cílem tohoto druhu doporučování je zvýšit utilizaci doporučení pro uživatele.[11]

Typickým příkladem může být doporučování léčiv na základě diagnózy pacienta. Použití kolaborativního filtrování může selhat kvůli neexistenci podobných pacientů (různé příznaky, vzácná nemoc, kombinace více nemocí zároveň, ...), použití doporučovacího systému založeného na obsahu také není možné (některé léky by pro pacienta mohli být smrtící, legislativa neumožňuje na pacientovi testovat léky „náhodným způsobem“). Doporučování na základě znalosti konkrétní domény umožňuje vyfiltrovat léky, na které je pacient alergický či jsou pro pacienta z jiného důvodu nevhodná. A zároveň doporučit takové léky nebo jejich kombinaci, která maximalizuje pravděpodobnost vyléčení s minimálním dopadem vedlejších účinků na pacienta.

2.2 Kolaborativní filtrování

V rámci kolaborativního filtrování se snažíme predikovat hodnocení objektu uživatelem. Narozdíl od doporučování založeném na obsahu je kolaborativní filtrování založeno na myšlence, že podobně chovající se uživatelé mají podobné preference. Takový druh doporučovacích systémů se označuje jako doporučovací systém založený na přístupu sousedů (anglicky *neighborhood approach*).

[12, s. 111] rozlišuje dva typy kolaborativního filtrování. Kolaborativní filtrování založené na uživateli (anglicky *user-based*) hledá podobnost mezi uživateli a na základě hodnocení podobných uživatelů predikuje hodnocení. Naopak kolaborativní filtrování založené na objektech (anglicky *item-based*) využívá podobnosti mezi objekty, tedy podobně hodnocené objekty budou mít podobné hodnocení od uživatelů.⁵

Predikované ohodnocení $\hat{r}_{ui}$ objektu i uživatelem u , můžeme definovat jako agregaci reprezentovanou agregační funkcí *aggr* nad hodnocením N nejpodobnějších uživatelů (takovou množinu budeme značit $\mathcal{N}(u)$) podobně jako v [19].

$$\hat{r}_{ui} = \text{aggr}_{v \in \mathcal{N}(u)} r_{vi} \quad (2.11)$$

⁵Všechny následující výpočty budou předpokládat kolaborativní filtrování založené na uživateli. Jednoduchou záměnou získáme ekvivaletní rovnice aplikovatelné pro kolaborativní filtrování založené na objektech.

Obdobně můžeme definovat rovnici i pro kolaborativní filtrování založené na objektech. Pokud uživatel v nehodnotil objekt i , pak můžeme použít pro jeho hodnocení neutrální výchozí hodnotu nebo ho vyřadit z množiny podobných uživatelů k uživateli u . Množinu obsahující N nejpodobnějších uživatelů, kteří hodnotili objekt i , budeme označovat jako $\mathcal{N}_i(u)$.

2.2.1 Prediktivní agregační funkce

Nejjednodušší prediktivní agregační funkcí je průměrné hodnocení N nejpodobnějších uživatelů (2.12). Takový výpočet nebere v potaz větší vliv podobnějších uživatelů.[13, s. 115] Výhodou je její jednoduchost a snadný výpočet. Výpočet hodnot pro všechny kombinace uživatelů a objektů má asymptotickou časovou složitost $\mathcal{O}(|\mathcal{U}| \cdot |\mathcal{I}| \cdot N) + \mathcal{O}(|\mathcal{U}| \cdot (kNN))$, kde $\mathcal{O}(kNN)$ je asymptotická časová složitost nalezení N nejpodobnějších uživatelů pro jednoho uživatele. Především při přidání nového hodnocení r_{vj} má aktualizace predikcí časovou složitost pouze $\mathcal{O}(|kRNN(v)|)$, tedy pro uživatele z množiny $kRNN(v)$ (mají uživatele v mezi N nejpodobnějšími) a jejich hodnocení objektu j upravíme predikci.

$$\hat{r}_{ui} = \frac{1}{|\mathcal{N}_i(u)|} \sum_{v \in \mathcal{N}_i(u)} r_{vi} \quad (2.12)$$

Běžným řešením problému s necitlivostí vzhledem k podobnosti uživatelů je použití váženého průměru (2.13).[13, s. 115] Podobnost uživatele u a v budeme značit jako w_{uv} . Vzhledem k možnosti záporných vah (nepodobností), které by mohly způsobit predikci hodnocení mimo povolený rozsah, budeme normalizovat predikci součtem absolutních hodnot podobností $|w_{uv}|$. [13, s. 115] Hodnocení podobnějších uživatelů má tak větší dopad na predikovanou hodnotu.

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \cdot r_{vi}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (2.13)$$

2.2.1.1 Normalizace průměrem

Problém různého měřítka hodnocení můžeme vyřešit normalizací predikce průměrem. Někteří uživatelé mají tendenci dávat spíše kladná hodnocení a jiní naopak spíše záporná. Tohoto trendu se můžeme zbavit posunutím škály takovým způsobem, aby průměrné hodnocení uživatelem bylo pro naše výpočty ekvivalentní 0, a upravením původní rovnice (2.13).

$$h(r_{ui}) = r_{ui} - \bar{r}_u$$

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \cdot h(r_{vi})}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \right)$$

Kde $\bar{r}_u$ je průměrné hodnocení uživatele u . Po jednoduchých úpravách získáváme

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \cdot (r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} . \quad (2.14)$$

Díky normalizaci průměrem záporná hodnocení implikují negativní očekávání a naopak kladná hodnocení pozitivní očekávání. [13, s. 122]

2.2.1.2 Normalizace odchylkou

Předpokládejme dva uživatele. Uživatel A a jeho hodnocení objektů, které je zcela náhodné na škále od 1 do 5. Naproti tomu předpokládejme uživatele B , který všechny objekty hodnotí známkou 3. Oba uživatelé mají průměrné hodnocení $\bar{r}_u = 3$. Přesto pokud uživatel B bude hodnotit objekt i jinou známkou ($r_{Bi} \neq 3$), pak takové hodnocení pro nás bude mít mnohem větší význam, než stejné hodnocení od uživatele A . [13, s. 123]

Z tohoto důvodu definujeme normalizační funkci jako (2.15), kde σ_u je standardní odchylka hodnocení uživatele u . Po jednoduchých úpravách (2.13) získáváme variantu (2.16).

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u} \quad (2.15)$$

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \cdot \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} \cdot \frac{r_{vi} - \bar{r}_v}{\sigma_v}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (2.16)$$

2.2.2 Podobnostní míry

Podobně jako vzdálenostní funkce u doporučovacích systémů založených na obsahu, tak i v případě kolaborativního filtrování záleží na výběru podobnostní funkce. [12, s. 124] uvádí celkem 3 nejpoužívanější podobnostní míry založené na korelaci.

2.2.2.1 Míry založené na korelaci

Často používanou mírou v *information retrieval* je kosinová podobnost (anglicky *Cosine Vector*) mezi dvěma vektory $\mathbf{x}_a$ a $\mathbf{x}_b$:

$$\cos(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^\top \mathbf{x}_b}{\|\mathbf{x}_a\| \|\mathbf{x}_b\|} .$$

Pro potřeby kolaborativního filtrování můžeme s vektorem $\mathbf{x}_a$ pracovat jako s vektorem hodnocení daného uživatele. Samotná kosinová podobnost mezi uživateli u a v bude definována jako (2.17). [12, s. 124]

$$CV(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{i \in \mathcal{I}_v} r_{vi}^2}} \quad (2.17)$$

Občas se můžeme setkat i s variantou (2.18), která nebere v potaz rozdíl objektů, jež druhý uživatel nehodnotil. Oba uživatele pak porovnáváme pouze pomocí objektů, které hodnotili oba zároveň.

$$CV_{alternative}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} r_{ui}^2 \sum_{i \in \mathcal{I}_{uv}} r_{vi}^2}} \quad (2.18)$$

Další populární podobnostní mírou je Pearsonova korelace (anglicky *Pearson Correlation*), která odstraňuje vliv průměru a rozptylu v hodnocení. Definována je jako (2.19). Je nutné podotknout, že se nejedná o kosinovou podobnost na normalizovaných datech, ale o naši zmiňovanou variantu kosinové podobnosti (2.18). [12, s. 125]

$$PC(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u) (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (2.19)$$

Vzhledem k různé škále hodnocení uživatelů definujeme *Adjusted Cosine* míru (volným překladem přizpůsobenou kosinovou míru) pro hledání podobnosti objektů. Je odvozena z Pearsonovi korelace, ale místo normalizace průměrem hodnocení objektu $\bar{r}_i$, je hodnocení normalizováno průměrem hodnocení uživatele $\bar{r}_u$.

$$AC(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u) (r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}} \quad (2.20)$$

2.2.2.2 Další podobnostní míry

[12] uvádí další 2 podobnostní míry, o kterých můžeme uvažovat. První z těchto je *Mean Squared Difference* (volně přeloženo střední čtvereční rozdíl), která určuje podobnost uživatelů jako inverzní hodnotu průměrného čtverečního rozdílu mezi hodnocením uživatelů. MSD má slabší vyjadřovací schopnosti oproti PC, jelikož nemůže zachytit negativní korelaci.

$$MSD(u, v) = \frac{|\mathcal{I}_{uv}|}{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - r_{vi})^2} \quad (2.21)$$

Další známou mírou je Spearmanova pořadová korelace (anglicky *Spearman Rank Correlation*). Místo samotného hodnocení objektů výpočet používá pořadí daného objektu v závislosti na hodnocení. Hodnota k_{ui} značí pořadí

objektu i v seznamu hodnocení uživatele u . Výhoda tohoto principu je, že nevyžaduje normalizaci dat. Nevýhodou pro diskrétní hodnocení s malou škálou může být mnoho stejných hodnocení, a tedy nutnost pracovat pro mnoho objektů se stejným pořadím.[12, s. 126]

$$\text{SRC}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u) (k_{vi} - \bar{k}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (k_{vi} - \bar{k}_v)^2}} \quad (2.22)$$

2.2.3 Problémy spojené s kolaborativním filtrováním

Získání predikovaných hodnot může být výpočetně náročné na základě zvolené prediktivní agregační funkce a podobnostní funkce. [12, s. 119–120] uvádí porovnání časové složitosti na základě výběru mezi doporučováním založeném na uživateli nebo na objektech. Časová složitost samotného doporučování je stejná pro oba modely $\mathcal{O}(|\mathcal{I}|k)$, tedy pro každý objekt v databázi potřebujeme spočítat doporučení, ve kterém pracujeme s k nejpodobnějšími uživateli nebo objekty (předpokládáme uložení informace o k nejpodobnějších uživateli nebo objektech v databázi). Trénovací nebo také offline část pak má složitost $\mathcal{O}(|\mathcal{U}|^2 \max_u |\mathcal{I}_u|)$ pro doporučování založené na uživateli a $\mathcal{O}(|\mathcal{I}|^2 \max_i |\mathcal{U}_i|)$ pro doporučování založené na objektech. V praxi můžeme předpokládat, že počet uživatelů bude řádově větší než počet doporučovaných objektů (a bude i nadále růst). Pokud počet uživatelů doporučovacího systému přesáhne počet doporučovaných objektů, pak je výhodné použít doporučovací systém založený na objektech.

Kolaborativní filtrování pracuje na základě statistických dat o uživateli. Pokud náš systém bude používat nový uživatel (resp. objekt), pak se dá předpokládat, že pro takového uživatele nebudeme mít k dispozici žádné informace. Tomuto problému se říká *cold-start* problém.[10, s. 7–8] Řešením může být aplikování alternativního přístupu vizte 2.2.4 nebo explicitní specifikace preferencí uživatele (resp. objektu). Variantou může být i získání hodnocení objektů na základě několika náhodných testů (pokus-omyl), kterou ovšem nemůžeme použít ve všech případech.

Vzhledem k rozšířenosti doporučovacích systémů v oblasti komerčních stránek a e-shopů jsou doporučovací systémy pod náporom útočníků. Jejich cílem může být zvýšení pravděpodobnosti doporučení vlastního produktu nebo naopak snížení pravděpodobnosti doporučení konkurenčního produktu. Útočníci během takových útoků běžně vytvářejí virtuální profily uživatelů a na základě znalostí o doporučovacím systému se snaží ovlivnit jeho výsledky. Útok průměrným hodnocením (anglicky *average attack*) předpokládá znalost průměrného hodnocení objektů (resp. uživatelů). Virtuální profil pak ohodnotí všechny produkty dle rovnoměrného rozdělení s daným průměrem. Vlastní produkt získá nadprůměrné hodnocení nebo naopak konkurenčnímu produktu

bude hodnocení sníženo.[10, s. 8] Řešením těchto problémů může být cena za doporučení, kdy uživatelé mohou hodnotit objekt pouze tehdy, pokud si objekt zakoupili. Také technické zajištění unikátnosti uživatele (IP adresa, kombinace parametrů detekovatelných přes prohlížeč uživatele, ...) může být alternativou.

Aby mohly doporučovací systémy pracovat co nejlépe, sbírají od uživatelů maximum možných dat. Takové systematické sbírání dat může podléhat zákonným regulacím a především musí být taková data dostatečně zabezpečena před zneužitím. Zároveň by měl doporučovací systém s takovými daty pracovat šetrně a obezřetně. Uživatel by měl mít přehled o tom, jaká data jsou sbírána a jakým způsobem budou používána.

2.2.4 Alternativní přístupy

[11, s.12–13] identifikuje další 3 principy podobné nebo jinak využívající kolaborativní filtrování. Prvním z těchto principů je *demografické* doporučování, které se pokouší doporučovat objekty na základě demografického profilu uživatele. Nejjednodušší variantou, používanou hojně na webu, je personalizace stránek (překlad, doporučení nejbližších kamenných obchodů, nabídka produktů dostupných v lokalitě, ...). Složitější systémy mohou využívat sofistikovanější přístupy, které zohledňují věk, národnost, náboženství a jiné atributy uživatele, na jejichž základě doporučují nebo naopak nedoporučují objekty (hinduismus – hovězí maso).

Doporučování založené na komunitách (anglicky *community-based*) využívá informaci o přátelích uživatele a na základě jejich preferencí jsou doporučovány objekty. Systém využívá předpokladu, že uživatelé se již sami seskupují do zájmových skupin a v rámci těchto skupin zaujímají stejné či podobné stanovisko. Uživatelé navíc spoléhají více na doporučení svých známých než na doporučení anonymních lidí.[20] Doporučovací systémy založené na komunitách jsou méně náchylné na *cold-start*, jelikož se nehledá podobnost mezi uživateli.[11, s. 13] Problém *cold-start* stále přetrvává při přidávání nových objektů.

Hybridní systémy nejsou alternativou jen pro kolaborativní systémy, ale pro všechny druhy doporučovacích systémů. Základní myšlenkou je podchytit problémy jednotlivých druhů doporučovacích systémů a vykompenzovat je aplikováním dalších přístupů. Aplikováním demografického doporučování můžeme vyfiltrovat doporučované objekty, které jsou pro uživatele zcela nepřijatelné (náboženské či etické aspekty). Přidáním komunitního doporučování můžeme přidat uživateli druhý vzorek uživatelů, jimž bude spíše věřit, a tím se vyvarovat problému *cold-start*. Tomuto se můžeme vyhnout i aplikováním doporučování založeném na obsahu, a tak uživateli doporučovat již kladně označené podobné objekty.

2.3 Hodnocení kvality

„Kvalita doporučovacíh systémů může být vyhodnocena porovnáním doporučení k testovací sadě známých hodnocení uživateli.“ [10, s. 7] Nejčastěji používanou metrikou je střední absolutní chyba (anglicky *mean-absolute error*), definována jako průměrný absolutní rozdíl mezi predikovanou hodnotou a hodnocením uživatele, kde $\mathcal{R}$ je množina všech hodnocení.

$$MAE = \frac{\sum_{\{u,i\} \in \mathcal{R}} |\hat{r}_{ui} - r_{ui}|}{|\mathcal{R}|} \quad (2.23)$$

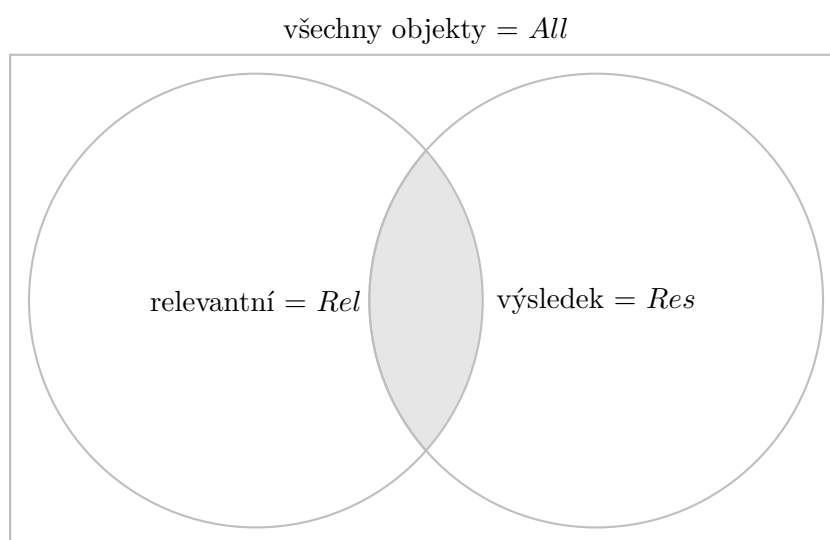
Související metrikou pro měření kvality doporučovacíh systémů je také střední kvadratická chyba nebo také její odmocnina RMSE (anglicky *root mean squared error*).

$$RMSE = \sqrt{\frac{\sum_{\{u,i\} \in \mathcal{R}} (\hat{r}_{ui} - r_{ui})^2}{|\mathcal{R}|}} \quad (2.24)$$

Dalšími indikátory mohou být metriky používané pro *information retrieval* jako jsou přesnost (anglicky *precision*) a úplnost (anglicky *recall*). Za výsledek doporučovacího systému můžeme považovat množinu objektů, jejichž predikovaná hodnota je vyšší (resp. nižší) než rozsah r . Relevantní výsledek pak můžeme definovat jako množinu objektů, s hodnocením vyšším (resp. nižším) než definovaný rozsah r . Přesnost (2.25) definujeme jako poměr relevantních výsledků k celkovému počtu výsledků získaných. Úplnost (2.26) definujeme jako poměr získaných relevantních výsledků k celkovému počtu všech relevantních objektů. [21, s. 2–4]

$$Precision = \frac{|Rel \cap Res|}{|Res|} \quad (2.25)$$

$$Recall = \frac{|Rel \cap Res|}{|Rel|} \quad (2.26)$$



Obrázek 2.1: Přesnost a úplnost.

Metody doporučování HRTF

Cílem práce je ověřit možnost využití doporučovacích systémů jako alternativu k běžně používaným metodám získávání HRTF (vizte 1.2.1). Kombinací metody HRTF a doporučovacích systémů můžeme teoreticky dosáhnout relativně přesné simulace prostorového zvuku bez potřeby individuálního měření HRTF. V následující kapitole rozebíráme jednotlivé metody doporučování objektů a také model (aplikaci) sloužící jako rozhraní pro hodnocení objektů. Následující přehled by tak měl být především inspirací k dalším experimentům a motivovat k rozšíření této práce.

3.1 Specifické vlastnosti HRTF

Každý specifický problém vyžaduje precizní nastavení parametrů doporučovacího systému a správnou volbu jednotlivých technik. Následující sekce rozebírá jednotlivé odlišnosti v doporučování HRTF oproti běžným doporučovacím systémům.

Naším doporučovaným objektem bude HRTF nebo její podmnožina. HRTF jsme definovali jako funkci pozice zdroje relativní vůči posluchači (definována sférickými souřadnicemi) a frekvence, tedy funkci 4 proměnných. Některé databáze mohou poskytovat další statistické informace o osobě, které HRTF profil patří. Mezi takové metainformace můžeme řadit fyziologické míry (antropometrické body) člověka, jako je šířka ramen, obvod hlavy kolem uší nebo jednotlivé části ušního boltce.

Běžné doporučovací systémy většinou doporučují nové objekty k prozkoumání. Námi navrhovaný doporučovací systém musí doporučovat nejlepší možné objekty z množiny všech objektů. Vzhledem k tomuto faktu a problému s nepřesností (vizte 3.1.1) budeme muset predikované hodnocení počítat nejen na základě podobných objektů, ale i na základě již provedeného hodnocení.

3.1.1 Hodnocení

Hodnocení jednotlivých profilů nemůže probíhat pouze explicitní formou. Takové hodnocení by bylo značně zkreslené a pro uživatele matoucí. Hodnocení bude muset probíhat implicitní formou, kterou můžeme chápat jako kalibraci zvukového zařízení. Hodnocení HRTF profilu budeme reprezentovat jako nepřesnost simulace. Nepřesnost simulace pro naše potřeby bude implicitně získaný úhel mezi skutečnou pozicí zdroje a uživateli simulovanou pozicí zdroje. Z takové reprezentace vyplývá, že hodnocení jednotlivých HRTF profilů bude reálné číslo. Teoreticky můžeme převést reálné hodnocení na diskrétní škálu pomocí mapování jednotlivých intervalů vizte tabulku 3.1, navíc není nutné použít lineární škálování, ale naopak můžeme preferovat určitý rozsah hodnot. V našem případě jsou pro nás zcela nepoužitelné profily HRTF s nepřesností větší než 90 stupňů. Pro naše potřeby však můžeme a budeme pracovat s původními reálnými hodnotami.

Tabulka 3.1: Ukázka nelineárního přeškálování reálných hodnot na diskrétní známkování

Hodnocení	1	2	3	4	5
Interval [°]	$\langle 0 - 5 \rangle$	$\langle 5 - 15 \rangle$	$\langle 15 - 45 \rangle$	$\langle 45 - 90 \rangle$	$\langle 90 - 180 \rangle$

Zároveň si musíme uvědomit, že neexistuje absolutně správné hodnocení HRTF profilů. Nepřesnost lokalizace zvuku nemusí být zapříčiněna pouze špatným HRTF profilem, ale samotnou nepřesností uživatele. Vyrušení, nechtěné označení výsledku a jiné problémy mohou ovlivnit hodnocení, vizte 5.3.2. Ač získáváme hodnocení implicitně pomocí testovací aplikace, nemáme jistotu, že takové hodnocení je skutečně objektivní a nejedná se o statistickou odchylku.

3.2 Současná řešení

Mnoho studií se zabývá personifikací HRTF profilů pomocí hledání vztahu mezi antropometrií uživatele a jeho HRTF profilem, jak uvádí [22]. V [22] je představena možnost simulace HRTF profilu za využití neuronových sítí.

Hlavní myšlenkou tohoto přístupu je práce se sníženou dimenzionalitou HRTF profilu. Vnořené neuronové sítě neslouží jen jako základ pro výběr HRTF profilu, ale také jako způsob přizpůsobení existujících profilů. Otestování tohoto modelu je implementováno použitím virtuálního uživatele, který určí simulovanou pozici (reprezentována jako Gaussovský proces s šumem). Teoretické výsledky dosahují průměrné nepřesnosti od 1° do 5°.[22]

Nevýhodou této metody zůstává použití pouze virtuálního uživatele. V části 3.4 rozebereme možné přístupy uživatelského testování, výhody, nevýhody a předpokládané problémy, které by i tuto metodu mohly ovlivnit.

V dalším experimentu [23] otestovali podobnou metodu již na skutečných uživateli. K testování využili jednoduchou aplikaci implementovanou v Matlabu, kde uživatel označil předpokládané místo simulace zvuku. Nepřesnost predikce pro reálné uživatele je již mnohem větší než v předchozím experimentu, ale můžeme pozorovat praktické zlepšení oproti nepersonifikovanému profilu.

3.3 Granularita doporučování

Předpokládejme doporučovací systém pro obchod s elektronikou, můžeme se pak rozhodovat mezi doporučováním celých počítačových sestav nebo jednotlivých komponent. Stejně tak se můžeme rozhodnout zda doporučovat na základě podobnosti nebo s využitím kolaborativního filtrování. Následující části rozebírají jednotlivé granularity doporučování HRTF i aplikování příslušných doporučovacích systémů.

3.3.1 Doporučování celých HRTF

Základní myšlenkou je pracovat s HRTF jako s kompletním balíkem a jednotlivé balíky doporučovat uživatelům. Výhodou tohoto principu je především jeho jednoduchost, přičemž máme jistotu, že existuje alespoň jedna osoba, pro kterou je daný profil ideální.

3.3.1.1 Doporučování založené na obsahu

[5] uvádí, že přesný 3D počítačový model hlavy lze využít k modelování HRTF. Pokud celý proces otočíme a budeme předpokládat znalost HRTF a vlastníkovy antropometrie hlavy (přidružené informace), můžeme se pokusit rekonstruovat 3D model hlavy uživatele. Ten bude sloužit jako preference uživatele a podle něj budeme hledat nejbližší HRTF profil. Pro naše potřeby není nutné pracovat s kompletním modelem, ale pouze s některými antropometrickými body.

Nevýhodou této metody je potřeba použití HRTF databáze s antropometrickými informacemi o HRTF profilech. Před použitím dat je nutné jejich předzpracování, kde se pokusíme odfiltrovat lineárně závislé parametry, a tak snížit dimenzi problému. Následná aplikace doporučovacího systému nám pomůže efektivně najít hodnoty antropometrických bodů pro uživatele.

Se znalostí přibližných antropometrických bodů uživatele můžeme najít nejpodobnější HRTF profil na základě podobnostní funkce. Při návrhu podobnostní funkce musíme zohlednit, které parametry mají významnější vliv při výběru ideálního profilu.

3.3.1.2 Kolaborativní filtrování

Předpokládejme, že můžeme posluchače rozdělit do unifikovaných skupin, podobně jako máme konfekční (či mezinárodní) velikosti. Rovněž předpokládejme, že pro každou takovou skupinu máme HRTF profil, který může reprezentovat celou skupinu. Formálně můžeme pro každou skupinu definovat HRTF profil tak, že použití jiného povede ke zhoršení přesnosti.

Výhodou této metody je nezávislost na antropometrických bodech posluchače. Obecně předpokládáme pouze podobnou stavbu vnějšího ucha a hlavy mezi podobnými uživateli. Nevýhodou je počáteční problém *cold-start*, který můžeme odstranit aplikováním dalších principů doporučovacích systémů, nebo v našem případě i počátečním náhodným doporučováním.

3.3.2 Doporučování založené na částech HRTF

Místo doporučování kompletních HRTF profilů můžeme HRTF profil rozdělit na jednotlivé komponenty a ty následně doporučovat. Umožníme tím vytvořit nové HRTF profily, které vzniknou kombinací několika různých HRTF profilů. Nevýhodou takového přístupu je však složitost celého procesu a vysoká pravděpodobnost vytvoření naprosto nerealistických HRTF profilů.

Jak se ukázalo při testování, některé vzorky působily na posluchače posunutím zdroje nebo kompletní transformací virtuálního prostoru. Nejčastěji se transformace projevovala zúžením nebo naopak rozšířením 180° intervalu. Pozice úplně vlevo a úplně vpravo byly lokalizovány jinde, než by posluchač očekával. Tohoto jevu by se dalo využít složením dvou různých HRTF, které deformují posluchačův virtuální prostor takovým způsobem, že právě kombinace částí by tento efekt eliminovala.

Předpokládejme HRTF profil A , který deformuje pouze pravou část prostoru, a naopak profil B deformující část levou. Použitím levé části profilu A a pravé části profilu B deformaci obou eliminujeme. Problémem stále zůstává vysoká složitost samotného HRTF sestávající se celkem ze 4 dimenzí, přičemž v našem příkladu jsme se omezili pouze na jednu složku.

Samotné HRTF tedy můžeme rozdělit pouze podle jedné složky nebo podle celé množiny. Následující přehled uvádí přehled složek a jejich intuitivní význam. V seznamu neuvažujeme různé posluchače (jimž profil patří) jako samostatnou dimenzi.

Frekvence Uživateli budeme doporučovat jednotlivé profily nejdříve podle některé vybrané frekvence. Po nalezení ideálního profilu pro danou frekvenci budeme hledat profily pro frekvence sousední. Můžeme předpokládat, že sousední frekvence budou mít podobnou charakteristiku. Nebo naopak můžeme pokračovat se zcela jinou frekvencí, a poté na základě dalších znalostí o HRTF interpolovat predikce pro mezilehlé frekvence.

Vzdálenost Některé HRTF databáze se zaměřují pouze na simulaci správného směru a neřeší vzdálenost simulovaného subjektu.⁶ Naopak jiné zaznamenávají i různé vzdálenosti od posluchače.⁷ Pro potřeby ověření funkčnosti doporučovacích systémů aktuálně není nezbytně nutné pracovat zvláště se vzdáleností od posluchače. Lokalizace zvuku pomocí vnějšího ucha závisí především na úhlu odrazu v rámci ušního boltce a úhel zdroje měříme od středu hlavy. Z toho plyne, že změna vzdálenosti způsobí změnu úhlu dopadu. Tudíž různé vzdálenosti můžeme simulovat zeslabením (případně zesílením) původního signálu. Rozdíly v ITD, ILD a úhlu dopadu můžeme simulovat použitím profilů pro různé úhly po rozdělení profilu dle levého a pravého kanálu.

Azimut Asi nejpodstatnější variantou je rozdělení profilu dle úhlu v horizontální rovině. Jak bylo zmíněno výše, v praktickém testu se ukázal jistý druh deformace zvukového prostoru, kdy se testovaným subjektům zdál prostor před nimi menší nebo naopak větší.

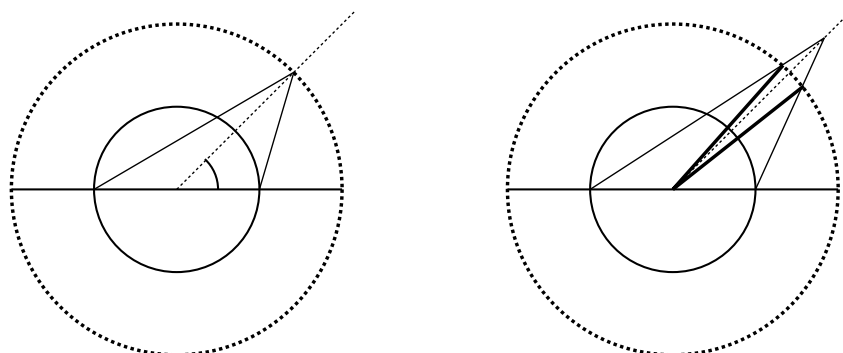
Elevace Dává smysl testovat zvláště zvuky nacházející se nad úrovní hlavy a zvláště zvuky pod úrovní hlavy. Jelikož takové směry můžeme rozpoznávat například díky odrazům od ramen, budou profily pro tyto pozice spíše závislé na parametrech ovlivňující tuto dimenzi. Tedy HRTF osoby s podobnou stavbou ramen nám poslouží pro simulaci zvuků nad úrovní hlavy.

Zvukové kanály Nad levým a pravým kanálem neuvažujeme jako nad standardní dimenzí. Pro většinu případů chceme použít levý i pravý kanál ze stejného HRTF profilu. Naopak pro změnu vzdálenosti (vizte výše) potřebujeme použít různé úhly i kanály. Vizte diagram 3.1, kde pro levý kanál potřebujeme HRTF profil pod úhlem α_L a pro pravý kanál naopak pod úhlem α_R .

Avšak nevýhodou je potřeba detailnějšího měření, což v našem případě znamená větší počet hodnocení pro více směrů (dimenzí). Pokud bychom se omezili pouze na doporučování různých profilů z různých směrů, mohli bychom rozdělit sférický prostor na několik významných oblastí a doporučovat profily právě pro tyto oblasti. Tím bychom snížili počet potřebných dimenzí. Při takovém dělení je nutné zohlednit fakt, že pro určité směry potřebujeme větší přesnost (vpředu, vlevo, vpravo, vzadu, ...). Pro tyto směry mohou oblasti na kulové ploše vypadat jako soustředné kružnice odpovídající daným přesnostem.

⁶CIPIC Database dostupná z <http://interface.cipic.ucdavis.edu/sound/hrtf.html>

⁷Listen HRTF Database dostupná z <http://recherche.ircam.fr/equipes/salles/listen/index.html>



Obrázek 3.1: *Použití rozdělení levého a pravého kanálu pro simulaci vzdálenějšího zdroje zvuku.* Plný kruh označuje posluchače, tečkovaný kruh naměřenou vzdálenost. Levý diagram ukazuje simulaci zvuku pro existující profil. Pravý diagram ukazuje situaci po změně vzdálenosti, zvýrazněné jsou nově simulované úhly pro levý a pravý kanál zvlášť.

3.3.3 Modifikace HRTF profilů

Další principy založené především na regresních modelech [22], [23] využívají pro individualizaci HRTF také možnost editovat již existující HRTF profily (samotné změny, kombinace více HRTF profilů). Vzhledem k vysoké dimenzi (vizte 3.3.2) se tyto principy snaží o snížení dimenze a tedy interpolaci vnitřních bodů.

Takové modifikace mohou být poměrně komplikované a pro potřeby našeho testování se jedná o přílišné detaily, které bychom mohli vykompenzovat použitím kvalitní HRTF databáze. Tedy databáze, která bude rovnoměrně zastupovat celou populaci.

3.4 Model testování

Jak již bylo zmíněno v úvodu, nejen nalezení správné kombinace doporučovacíh systémů je otázkou našeho zkoumání, ale je jím i způsob testování a tedy získávání hodnocení jednotlivých profilů. Cílem takového doporučování může být zlepšení přesnosti pro asistivní pomůcky i zábavní průmysl, z toho důvodu je důležitý i výběr testerů.

Výběr osob pro otestování našeho modelu je základním kamenem testování. Využití virtuálního uživatele nebo osoby je pro ověření reálného nasazení nedostačující. Ideálním kandidátem je tedy samotný uživatel takové aplikace. V případě asistivních pomůcek se jedná o nevidomé, u kterých předpokládáme větší smysl pro orientaci v prostoru. Pro oblast zábavního průmyslu může být uživatelem prakticky kdokoli, přičemž pro osoby bez zrakového postižení bude

mnohem obtížnější se orientovat v prostoru jen pomocí sluchu.

3.4.1 Testované pozice

Experimenty uvedené v 3.2 pracují s testováním přesnosti „všech“ pozic (případně podmnožinou konkrétních bodů v prostoru). Takové experimenty mohou být komplikované na testování, především z hlediska označení simulované polohy uživatelem.

K dispozici můžeme mít videozáznam nebo sledovací zařízení a uživatele ukazujícího v prostoru bod simulace. Tento bod může případně označit na počítači pomocí grafického rozhraní a virtuální mapy. Takové hledání může být nepřesné, jelikož se uživatel nemusí přesně orientovat v grafické reprezentaci nebo si jen nedokáže převést zvukovou informaci na grafickou. Výhodné naopak může být použití nástrojů pro virtuální realitu, které dokáží zaznamenávat přesně polohu ruky (nejlépe relativně k pozici hlavy).

Alternativou k všesměrovému testování může být testování pouze jednoho směru, a tím je především poloha přímo před posluchačem. Cílem uživatele pak může být nasměrovat se čelem ke zdroji zvuku. To může být implementováno v reálném nebo virtuálním prostoru. Oproti předchozí variantě nemusí být uživatel rušen nutností označovat polohu, ta může být zaznamenána automaticky v aplikaci nebo pomocí úhloměru na otočné židli. Takový způsob reflektuje reálné navigování v prostoru, uživatel najde zdroj zvuku a vydá se k němu.

Kombinací obou principů můžeme získat měření pro více směrů s lokalizací podobnou předchozímu případu. Uživatel dostane za úkol nasměrovat se zcela vlevo (či jiný snadno určitelný směr). Tento způsob podporuje myšlenku, že má smysl přesně rekonstruovat HRTF pouze pro významné pozice ve virtuálním prostoru, jelikož ostatní pozice člověk nemusí být schopen přesně určit ani v prostoru reálném.

3.4.2 Způsob testování

Hlavní otázkou této části práce je nabídnout varianty uživatelských rozhraní k získání HRTF profilu pomocí doporučovacích profilů. Na výběr máme jak lokální testování, tak vzdálené pomocí běžně dostupných technologií nebo moderních VR setů. Následující analýza popisuje jednotlivé varianty, jejich výhody a nevýhody.

3.4.2.1 Testování v prostoru

Pro testování může posloužit upravená laboratoř určená původně k nahrávání HRTF profilů. Uživatele posadíme doprostřed místnosti a umístíme sluchátka do uší. Následně pouštíme uživateli simulované zvuky a on pomocí ruky (slovním vyjádřením, zaznačením do mapy, ...) zaměří zdroj zvuku. Spočítáme

3. METODY DOPORUČOVÁNÍ HRTF

rozdíl simulované pozice a skutečné pozice. Pro zaznamenání může posloužit jednoduchý videozáznam ze tří různých směrů (shora, zleva a zepředu).

Výhodou tohoto testování je možnost testovat přesnost na libovolnou pozici v prostoru. Případně je možná i automatizace využitím videotechniky. Metoda vyžaduje, aby se posluchač nepohyboval, což znemožňuje uživateli efektivně ukázat pozici zvuku vzadu. Problémem může být i nutkání posluchače pohnout hlavou směrem ke zdroji, pro lepší lokalizaci zvuku. Problém s pohybujiícím se posluchačem můžeme vyřešit i přidáním sledovacího zařízení (anglicky *head tracking*) a přepočítáváním zvuku v reálném čase.

Pokud se při testování omezíme pouze na horizontální rovinu a testování pozice vpředu, můžeme pak umístit posluchače na speciálně upravenou otočnou židli (s pevným umístěním hlavy). Taková židle může být vybavena manuálním nebo automatickým úhloměrem. Cílem uživatele pak bude otočit se, aby byl čelem ke zdroji zvuku. Problémem může být nepřírozený pohyb hlavy, avšak přesnost měření bude větší.

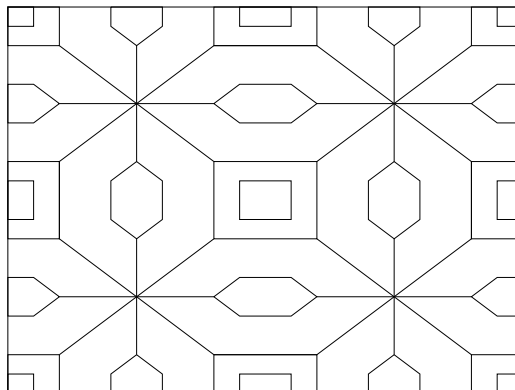
Přítomnost uživatele na testování může být nepohodlná. Navíc těžko podporuje integraci s doporučovacími systémy. Pokud by takové testování bylo i časově náročné je praktičtější přímé naměření HRTF profilu. Z tohoto důvodu nebudeme tento způsob dále uvažovat.

3.4.2.2 Aplikace

Nejjednodušší aplikací, díky které mohou uživatelé hodnotit profil, je grafické rozhraní s virtuálním prostorem, kde uživatel může označit pozici zdroje. Tento přístup má hned několik nevýhod. Uživatel nejprve musí převést informaci z virtuálního zvukového prostoru do reálného prostoru, a následně z reálného prostoru do virtuálního grafického prostoru. Druhou překážkou může být neustálé vyrušování posluchače nutností přeskakovat mezi dvěma prostory. Pro mnoho uživatelů je totiž příjemnější během simulace zavřít oči. Výhodou naopak je testování libovolné pozice v prostoru.

Pokud bychom chtěli uživateli usnadnit orientaci v grafickém virtuálním prostoru, pak můžeme grafické rozhraní reprezentovat jako monitor (případně více monitorů dohromady) rozdělený na jednotlivé sektory. Cílem uživatele je určit, v kterém sektoru se simuluje zvuk. Oblasti simulace nemusí být symetrické ani stejně velké, naopak mohou odpovídat očekávaným přesnostem, jak ukazuje diagram 3.2. Nevýhodou tohoto přístupu je možnost testování pouze zobrazitelných oblastí, simulace musí být přepočítána vzhledem k vzdálenosti uživatele a monitoru. Uživatel stále musí přeskakovat mezi dvěma prostory. Uživatelské hodnocení (přesnost) pak můžeme redukovat na binární hodnocení 0 a 1.

Abychom se vyhnuli přeskakování mezi prostory, musíme vytvořit testovací aplikaci takovým způsobem, aby v ní uživatel rovnou mohl označit pozici zdroje. K takovému principu můžeme využít standardního postupu používa-



Obrázek 3.2: Následující schéma ukazuje možné rozdělení monitoru na vícero oblastí, přičemž pro důležité body máme k dispozici zvýraznění přesnosti v podobě zdvojené plochy. Větší efektivity u této metody můžeme dosáhnout použitím větší obrazovky či mnoha monitorů.

ného v počítačových hrách typu FPS⁸. Cílem uživatele je pomocí myši zaměřit zdroj zvuku a označit jeho pozici. Nevýhodou této metody je možnost označit pouze směr vpředu, případně lze rozšířit o určení výšky nebo dalších významných směrů. Na druhou stranu uživatel může nepřerušeně testovat a dokonce lokalizovat zvuk otáčením hlavy (bez *head-tracking* je otáčení hlavy simulováno pohybem myši).

V případě, že chceme cíleně měřit jen konkrétní pozici, můžeme aplikaci vytvořit opačným způsobem. Uživatel zůstává nehybný a zdroj zvuku se pohybuje. Uživatel provede akci (kliknutí myši, stisk klávesy) pokud se domnívá, že je zdroj v měřené pozici. Tato metoda nám dává výhodu v testování různých okolních směrů, protože uživatel nemá možnost ovlivnit pohyb zdroje a přizpůsobit si model dle svého pohodlí (rozhlížení se, lokalizace zvuku pomocí odhadování středu mezi levou a pravou stranou, ...).

3.4.2.3 Sety pro virtuální realitu

S rozšířením zařízení pro virtuální realitu (dále jen VR) můžeme uvažovat i o jiných způsobech, které mohou být pro uživatele přirozenější. VR zařízení nám mohou umožnit sledovat otočení a naklonění hlavy. Takové vylepšení dnes poskytuje i chytrý mobilní telefon s kompasem a gyroskopem. Umístění mobilního telefonu na hlavu uživatele můžeme zajistit například pomocí Cardboardu⁹. Během simulace bude potřeba vypnout obrazovou informaci nebo ji nahradit za virtuální prostor, přičemž zdroj zvuku nebude pro uživatele viditelný.

⁸anglicky *first-person shooter*, akční hra z pohledu první osoby

⁹<https://vr.google.com/cardboard/>

Cílem uživatele tak může být skutečným pohybem hlavy najít zdroj zvuku, který buď jednoduše označí (stiskem tlačítka) nebo se na něj po dobu 5 sekund dívá. Tato metoda je velice závislá na přesnosti sledovacího zařízení, před měřením je tedy nutná co nejpřesnější kalibrace. Druhou možností jak označit simulovanou polohu je využití ovládacího kontroleru s vlastním lokalizačním zařízením, uživatel pak může na cíl jednoduše ukázat a nemusí se tak otáčet.

Nevýhodou může být vyšší cena, kvůli které nemusí být VR sety použitelné pro širokou veřejnost. S expanzí VR aplikací se dá očekávat větší poptávka a tedy i možnost využití VR pro nalezení vhodného HRTF profilu.

3.4.3 Vyhodnocení

Vzhledem k absenci absolutně správného hodnocení profilu uživatelem, není možné použít klasické přístupy vyhodnocování kvality doporučovacího systému. Jako ukazatel kvality doporučovacího systému využijeme průměrnou přesnost simulace.

Od doporučovacího systému očekáváme rostoucí zlepšení přesnosti s rostoucím počtem hodnocení. Navíc předpokládáme, že použití doporučovacího systému bude mít za následek nižší průměrnou nepřesnost, než použití ne-personalizovaného HRTF profilu. Pro srovnání nám budou sloužit výsledky experimentu [23] s reálným uživatelem. Průměrné zlepšení v rámci všech uživatelů bude dalším sledovaným ukazatelem.

Implementace

Na základě rozboru metod uvedených v předchozí kapitole jsme se rozhodli využít kolaborativního filtrování pro doporučování celých bloků HRTF. Vybrali jsme si implementaci jednoduchého virtuálního prostoru ve stylu FPS, to umožní uživateli představit si virtuální prostor a nabídnout standardní ovládání. Pohyb ve virtuálnímu prostoru je omezen pouze na horizontální rovinu, uživateli je umožněno rozhlížet se doleva a doprava. Prostorový zvuk budeme simulovat pomocí aplikace za využití technologií WebGL a WebAudio. V této kapitole rozebereme použité technologie a objektový návrh celé aplikace.

4.1 Technologie

Abychom mohli poskytnout aplikaci maximálnímu množství uživatelů, rozhodli jsme se vytvořit doporučovací systém jako webovou aplikaci. Díky tomu mohli někteří testéři vyzkoušet simulaci 3D zvuku v pohodlí svého domova. Aplikaci pohání technologie WebGL a WebAudio, o autentizaci uživatelů a perzistenci dat se stará platforma Google Firebase¹⁰.

4.1.1 HTML5 a JavaScript

K využití technologií WebGL a WebAudio je jako základ nezbytné použití HTML5 a JavaScriptu. HTML5 je nejnovější verze specifikace pro tvorbu webových stránek. Pro naše potřeby přidává 2 základní koncepty, které využíváme, a tím jsou elementy `canvas` a `audio`.^[24]

Element `canvas` slouží jako základní kontejner pro vykreslování grafického obsahu. Pokud chceme do takového kontejneru vykreslovat grafiku, nejprve získáme kontext (vizte zdrojový kód 4.1). K dispozici máme na výběr dva druhy kontextů `2d` a `webgl` (v případě, že prohlížeč podporuje WebGL API). Specifikace `2d` kontextu spadá stejně jako HTML5 pod W3C.^[25] Specifikace

¹⁰<https://firebase.google.com/>

`webgl` kontextu je zajišťována Khronos Group a je založena na specifikaci OpenGL ES.[26]

Zdrojový kód 4.1: Získání 2D grafického kontextu

```
var canvas = document.getElementById("application");  
var context = canvas.getContext("2d");
```

Druhým pro nás podstatným elementem je `audio`, díky němuž můžeme posluchači reprodukovat přehrávaný zvuk. Tento nejprve upravíme pomocí konvolučního filtru, dle příslušného HRTF profilu. Pro simulaci 3D zvuku je potřeba využít ekvivalentu ke grafickému kontextu, a tím je `AudioContext`.

JavaScriptové kódy obsluhující naši aplikaci splňují požadavky definované specifikací ECMAScript 5 ve striktním módu. Verze 5 je použita především z důvodu vyšší podpory běžných prohlížečů než novější verze 6.[27][28] Po četných problémech s implementací by bylo výhodnější implementovat aplikaci v TypeScriptu, abychom vynutili i dodržování datových typů. Bohužel aplikace již byla těsně před dokončením. Aplikace tak splňuje alespoň požadavky vyžadované striktním módem.

4.1.2 WebGL

WebGL je grafické rozšíření pro HTML5 založené na OpenGL SE 2.0. Pokud prohlížeč podporuje WebGL (v některých prohlížečích je nutné explicitně povolit výpočty na GPU), pak získáme z elementu `canvas` WebGL kontext, který můžeme využít pro nízkoúrovňovou práci s grafikou a především GPU. WebGL kontext nám umožňuje nejen vykreslování základních grafických primitiv, ale také nahrání vlastních vertex či fragment shaderů.[26]

Nevýhodou práce přímo s WebGL kontextem je příliš nízká úroveň programování. Pro většinu aplikací nám stačí použít jednoduché shadery, implementované transformační matice a pomocné funkce k nim vztahované. Pro tyto účely je vhodnější využít WebGL frameworků jako je `three.js`¹¹ nebo `babylon.js`¹². `babylon.js` se prezentuje více jako WebGL engine určený pro tvorbu her, zároveň obsahuje implementované různé funkcionality, včetně integrace WebAudio.[29] Framework `three.js` se prezentuje spíše jako grafická knihovna nad WebGL, přičemž existuje spousta rozšíření, které lze využít. Abychom se vyhnuli problémům s konflikty mezi naším nastavením WebAudio a nastavením poskytovaným `babylon.js` frameworkem, použili jsme v práci `three.js` framework a některá další rozšíření (importování modelů, ...).

Výhodou použití frameworků místo manuální práce s WebGL je především možnost nestarat se o používanou verzi WebGL a podporu prohlížečů. Tabulka 4.1 ukazuje verzi prohlížeče, která jako první podporovala specifikaci a ukazuje procento uživatelů s alespoň částečně kompatibilním prohlížečem, pro fungování WebGL a WebGL 2.0.

¹¹<https://threejs.org/>

¹²<https://www.babylonjs.com/>

Tabulka 4.1: Podpora specifikací WebGL a WebGL 2.0 ze strany prohlížečů. Procentuální zastoupení všech uživatelů dle podporovaných prohlížečů.[30][31]

Prohlížeč	WebGL	[%]	WebGL 2.0	[%]
Chrome	únor 2012	27,21	leden 2017	22,96
Firefox	březen 2011	6,56	leden 2017	5,10
Opera	listopad 2012	0,93	leden 2017	0,61
IE ¹³	říjen 2013	5,06		
Safari	říjen 2014	12,05		
Bez podpory ¹⁴		8,14		71,31

4.1.2.1 three.js

Knihovna **three.js** poskytuje vyšší vyjadřovací schopnosti než samotné použití WebGL. Umožňuje tak programátorovi pracovat přímo s objekty grafické scény, byť i stále může pracovat s nízkoúrovňovými funkcemi. Jednoduše lze vkládat do scény zdroje světla a různé druhy kamer. **three.js** poskytuje skrze rozšíření také podporu pro jednodušší práci s WebAudio, případně s WebVR.[32]

Výhodou implementace pomocí knihovny **three.js** je především možnost rychlé konverze do formátu pro WebVR. I bez podpory WebVR má programátor k dispozici efekty, jejichž aplikováním může 3D zobrazení provést metodou anaglyfu, *parallax barrier*, *pepper's ghost* nebo stereo zobrazením, které se dá využít při zobrazení na mobilním telefonu s Cardboardem.[32]

4.1.3 WebAudio

WebAudio je zvukové rozšíření pro HTML5. Umožňuje nám přehrávat zvukové nahrávky přímo v prohlížeči uživatele. Web Audio API nám poskytuje rozhraní nejen pro práci s monofonními nebo stereofonními nahrávkami, ale poskytuje nám práci i s vícekanálovými systémy pomocí **ChannelMergerNode** a **ChannelSplitterNode**. [33] Pokud bychom pracovali s 5.1 audio nahrávkou, pak můžeme využít toto rozhraní a upravit zvuk tak, aby byl použitelný pro simulaci v rámci HRTF (monofonní nahrávka).

Kromě běžných nástrojů pro práci s audiem, jako je přehrávání či kombinace efektů se zvukovou nahrávkou, nám Web Audio API nabízí rozhraní **PannerNode**, jež dokáže zprostředkovat simulaci prostorového zvuku metodou HRTF.[33] Nevýhodou je vyšší využití CPU kvůli nutnosti přepočítávání konvoluce vstupního zvukového signálu a HRTF profilu.[34] Navíc nám jeho rozhraní neumožňuje manuálně nastavit HRTF profil.[35] Část implementace [35] využijeme i v naší aplikaci.

¹³Zahrnuje jak prohlížeče IE, tak prohlížeč Edge

¹⁴Do celkového počtu jsou zahrnuty i prohlížeče zde neuvedené

Výhodou přístupu vlastní implementace není jen možnost použít vlastní HRTF profily. Díky WebGL můžeme případný výpočet konvoluce přesunout na GPU, a tím zrychlit algoritmus a ušetřit drahé výpočty na CPU. Jelikož v naší aplikaci simulujeme vždy právě jeden zvuk, nemusíme se zabývat výpočetní náročností. V případě použití více zdrojů bychom měli uvažovat i o této variantě.

Tabulka 4.2: Procentuální zastoupení uživatelů, kteří mohou využít Web Audio API, rozdělené dle prohlížečů.[36]

Prohlížeč	Edge	Firefox	Chrome	Safari	Opera	Bez podpory ¹⁵
[%]	1,58	5,79	26,2	11,82	0,56	21,6

4.1.4 WebVR

WebVR specifikace v současnou chvíli je pouze v experimentální fázi. Využít můžeme předběžné vydání WebVR 1.1, které může být implementováno v některých prohlížečích, ale nejedná se o finální specifikaci. Aktuálně vyvíjená a zatím nestabilní specifikace je WebVR 2.0.[37] Jelikož neexistuje žádná stabilní verze této specifikace je moudré rozhodnutí implementaci testovacího rozhraní podporující VR zařízení odložit.

Využít rozhraní WebVR chceme především z důvodu získání informací o pozici hlavy a případného ovladače. Samotné zobrazení pro Cardboard nebo jiný druh 3D obrazové simulace zvládneme i využitím vhodných filtrů frameworku `three.js`. [32]

4.1.5 Google Firebase

Žádná aplikace se neobejde bez perzistentního úložiště dat. Google Firebase (dále jen Firebase) poskytuje jednoduchou realtime databázi založenou na formátu JSON. Pro účely testování byl použit program Spark, který zdarma umožňuje ukládat až 1 GB dat při paralelním připojení až 100 klientů. S omezením na množství přenesených dat v rozsahu 10 GB za měsíc.[38]

Firebase poskytuje i další služby, mezi které patří autentizace uživatele, lokální úložiště dat nebo jednoduchý hosting. Na takové platformě můžeme provozovat kompletní aplikaci, případně pro offline fázi doporučovacího systému můžeme využít integrované cloudové funkce.[39]

4.1.5.1 Autentizace

Vzhledem k potřebě identifikace uživatelů jednotlivých hodnocení využijeme možnost autentizace prostřednictvím Firebase. Díky tomu získáme i kontrolu nad zapisovanými daty, jelikož Firebase nám umožňuje nastavit pravidla

¹⁵Ve výběru nejsou uvedeny všechny prohlížeče (mobilní, ...).

pro zápis a čtení jednotlivých částí realtime databáze. Každému uživateli je přiřazen unikátní identifikátor, jehož pomocí se identifikuje pro ostatní části Firebase.[40]

Autentizace je možná pomocí Google účtu, Facebook účtu nebo libovolného jiného systému podporující standard OAuth 2.0 nebo OpenID Connect. Pro naši práci využijeme autentizaci pouze prostřednictvím služby Google (většina testerů jsou studenti FIT ČVUT a mají k dispozici školní Google účet). Firebase při přihlášení požaduje po uživateli zobrazení základních informací, mezi které patří emailová adresa, jméno, příjmení a případně profilový obrázek.

4.1.5.2 Databáze

Nejdůležitějším prvkem Firebase je realtime databáze založená na formátu JSON. Pokud se rozhodneme pro použití Firebase máme na výběr z několika rozhraní (Web, REST, iOS, C++, Unity, ...). Pro nás bude klíčové použití JavaScriptové knihovny pro webové aplikace. Narozdíl od RESTového přístupu již obsahuje systém pro práci s událostmi, jinak musíme aktualizace v reálném čase simulovat opakovanými dotazy (tzv. *polling*).

Jelikož se jedná o JSON objektovou databázi, pak pro získání určitého objektu je nutné stáhnout všechny hodnoty, kterými mohou být vnořené objekty. Takové stahování bude pro větší databáze náročnější, proto se musíme snažit o nízké zanoření objektů. V tom nám mohou pomoci i křížové reference, které je nutné simulovat ručně.[40]

Výhodou využití JavaScriptové knihovny pro webové aplikace je automatická podpora offline obsluhy. Pokud by došlo k odpojení klienta, pak aplikace bude pokračovat v offline režimu a po opětovném připojení se provedou opožděné změny serveru. Další výhodou je transakční zpracování požadavků, a s tím spojené zachování integrity dat.[40]

Alternativou k Firebase realtime databázi je projekt RethinkDB¹⁶. Jedná se o open-source řešení s podobnými vlastnostmi jako u Firebase realtime databáze. Administrace databáze je přístupná přes webové rozhraní a samotné řešení je navrženo tak, aby bylo možné spustit databázi na více pracovních stanicích zároveň, a tak zvýšit výkon celé aplikace. Navíc RethinkDB nabízí podporu pro Map-Reduce úlohy, což otevírá prostor pro další experimenty.

4.1.5.3 Další možnosti

Pokud bychom chtěli využít plného potenciálu Firebase utilit, pak můžeme naši aplikaci rozšířit o kompletní zpracovávání na straně Firebase. Díky integrovanému webovému serveru nemusíme řešit hosting pro naši aplikaci. Práce s webovým serverem probíhá přes webové rozhraní nebo pomocí Firebase CLI.

¹⁶<https://www.rethinkdb.com/>

Pro větší objemy dat, jako jsou HRTF profily, grafické modely a případné zvukové nahrávky, můžeme využít cloudového datového úložiště.[40]

Při implementaci realtime doporučovacího systému je potřeba rychlá reakce na změny jednotlivých hodnocení. V závislosti na zvolené podobnostní a agregační funkci je potřeba změnit různá data. Takové změny by pro uživatele mohly být výpočetně náročné a navíc bychom umožnili uživateli měnit i data mimo svůj rozsah působnosti. Využitím cloudových funkcí se můžeme vyhnout oběma problémům. Výpočet by probíhal asynchronně a data by zůstala zapouzdřená.

4.2 Návrh

Testovací aplikace byla navržena pro snadnější aktualizaci a úpravu jako MVC webová aplikace. Doporučovací systém funguje nad daty CIPIC HRTF databáze poskytující informace o „vlastníkovi“ HRTF profilu. Následující sekce popisuje výběr dat, strukturu aplikace a návrh JSON databáze pro zaznamenávání průběhu testování.

4.2.1 HRTF databáze

Pro náš doporučovací systém jsme si vybrali HRTF profily poskytnuté z CIPIC HRTF databáze. Výhodou této databáze jsou metainformace o nahrávaných subjektech. V případě potřeby jsme schopni pracovat i s doporučovacím systémem založeným na obsahu.

Databáze obsahuje celkem 43 HRTF profilů reálných lidí, navíc zde nalezneme HRTF profily umělé figuríny KEMAR. Pro každý tento profil je HRIR naměřeno pro 50 různých elevací rovnoměrně od -45° do $+230.625^\circ$. A celkem 25 azimutů, rovnoměrně rozmístěných od -45° do $+45^\circ$ a speciálně rozmístěné pro krajní polohy. Dohromady pro obě dvě uši se jedná o 2500 měření.[41]

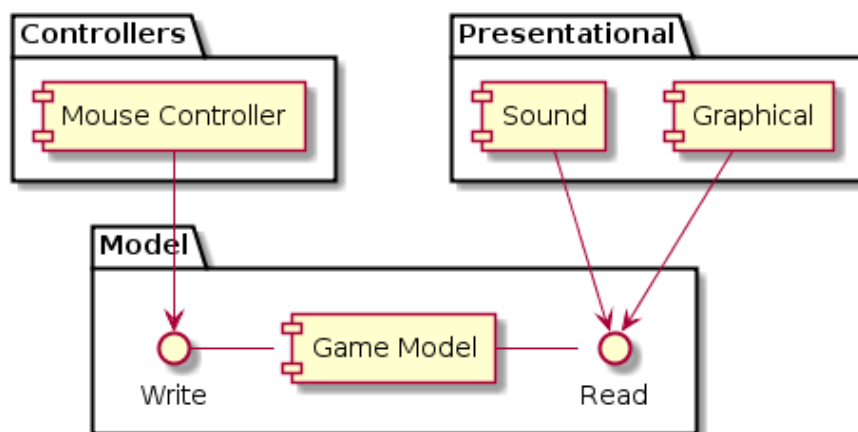
Přestože díky vysokému počtu měřených pozic máme virtuální prostor hustě pokryt, je nutné pro neměřené pozice nalézt HRTF. Mohli bychom použít metodu nejbližšího souseda nebo průměrování. Avšak dle [42] pro všechny ostatní simulované pozice využijeme interpolaci pomocí Delaunay triangulace, která má lepší výsledky.

Každý uživatel je identifikován 17 antropometrickými body týkající se těla a hlavy. Dále je k dispozici dalších 10 antropometrických bodů zaměřených na vnitřní ucho.[41] Tyto informace můžeme využít pro další experimenty spojené s doporučováním HRTF profilů na základě obsahu.

4.2.2 Struktura

Aby testovací rozhraní mohlo sloužit i pro další experimenty, je nutné dobře navrhnout jeho strukturu. Pro naši aplikaci jsme se rozhodli pro standardní

MVC strukturu. Jak ukazuje 4.1, oproti klasickým webovým aplikacím máme dva prezentační systémy (grafický a zvukový).



Obrázek 4.1: Na diagramu si můžeme všimnout netradičně dvou prezentačních složek. Díky MVC struktuře implementace ovládání pomocí VR zařízení zahrnuje pouze vytvoření nového kontroleru.

4.2.2.1 Kolaborativní filtrování

Jedním z cílů této práce je vytvoření knihovny pro kolaborativní filtrování, kterou bude možné použít pro podobné problémy jako je doporučování HRTF profilů. To zahrnuje opakované hodnocení již ohodnocených objektů a případnou možnost online aktualizace doporučovacího systému (doporučování na základě aktuálních změn v databázi).

Ač samotná třída byla navržena pro doporučování založeném na uživateli, záměnou parametrů a zviditelněním některých privátních prediktivních metod umožníme doporučování založené na objektech. Pro výpočet podobnosti byla použita kosinová podobnost. Její výhodou je nízká časová složitost pro online aktualizaci $\mathcal{O}(|\mathcal{U}|)$.

Samotné opakované hodnocení je agregováno průměrem všech hodnocení. Jelikož samotný průměr, jak se ukázalo, nemusí být vždy ideálním řešením (uživatelé dělají chyby, neorientují se pořádně ve virtuálním audio prostoru, ...) je implementováno ignorování minimálního a maximálního hodnocení během každých pěti pokusů.

Jelikož doporučujeme i již vyzkoušené HRTF profily je predikovaná hodnota vypočtena jako (4.2), přičemž c_{ui} je počet hodnocení HRTF profilu i uživatelem u . Díky parametru α tak zajistíme, že čím častěji uživatel hodnotil

daný HRTF profil, tím více je jeho průměrné hodnocení důvěryhodnější.

$$\alpha = \left(\frac{3}{2}\right)^{-c_{ui}} \quad (4.1)$$

$$\hat{r}_{ui} = \alpha \left(\bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} (r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \right) + (1 - \alpha) r_{ui} \quad (4.2)$$

Z obavy možného překročení limitů služby Firebase není online synchronizace doporučovacího systému implementována v praxi. Využívá se pouze online aktualizace na straně uživatele a synchronizace s databází probíhá pouze při spuštění a pomocí objektu `FirestoreLogger`, který umožňuje pouze transakční zápis do databáze. Ze stejného důvodu není implementováno průběžné ukládání N nejpodobnějších sousedů, které se přepočítává při každém dotazu. Naopak počítání podobnosti se ukládá do mezivýsledků a díky online aktualizaci získání podobnosti dvou uživatelů má konstantní časovou složitost.

Výběr podobnostní funkce Jako podobnostní funkci pro porovnávání uživatelů jsme zvolili kosinovou podobnost (4.3). Její výhodou je rychlý přepočet při změně hodnocení v databázi, který má asymptotickou časovou složitost $\mathcal{O}(|\mathcal{U}|)$. Pro každého uživatele musíme přepočítat hodnotu v konstantním čase.

$$\text{CosSim}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{i \in \mathcal{I}_v} r_{vi}^2}} = \frac{C_{uv}}{\sqrt{C_{uu} C_{vv}}} \quad (4.3)$$

Při změně hodnocení objektu i uživatelem u , můžeme nové hodnoty koeficientů C_{uv}^{new} a C_{uu}^{new} přepočítat dle rovnic (4.4) a (4.5). Pokud se jedná o první hodnocení r_{ui} pak staré hodnocení $r_{ui}^{old} = 0$.

$$C_{uv}^{new} = C_{uv}^{old} + (r_{ui}^{new} - r_{ui}^{old}) r_{vi} \quad (4.4)$$

$$C_{uu}^{new} = C_{uu}^{old} + (r_{ui}^{new})^2 - (r_{ui}^{old})^2 \quad (4.5)$$

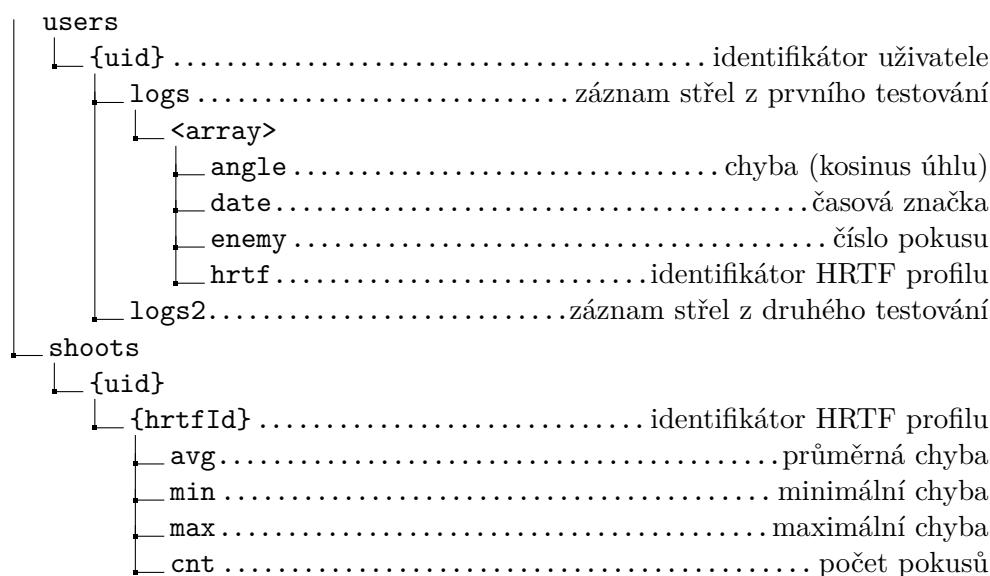
Nejbližší sousedé Pro výběr podobných uživatelů jsme se rozhodli pro model kNN dotazování. Druhým krokem je zvolení vhodného k , tedy počet nejpodobnějších uživatelů, které budeme hledat. Pokud zvolíme $k = 1$, pak při vyhledávání nejpodobnějších uživatelů k uživateli u najdeme vždy pouze uživatele u ($\text{CosSim}(u, u) = 1$). Naopak při výběru všech uživatelů by nám vzrostla časová složitost, přičemž jejich příspěvek by byl minimální. Nakonec jsme se rozhodli pro $k = \sqrt{|\mathcal{U}|} \approx 7$, které se v předběžných testech ukázalo jako obstojné.

4.2.3 Databáze

Oproti relačním databázím se v případě JSON databází zaměřujeme především na minimalizaci hloubky a seskupení dat takovým způsobem, abychom zbytečně nemuseli načítat velké objemy dat. Druhým cílem našeho návrhu je jednoduchý přístup k položkám databáze.

Pro účely záznamu testování je pro každého uživatele zaznamenáván průběh testování. Tímto způsobem uživatel přistupuje pouze ke své části logů a není nutné stahovat databázi všech uživatelů. Základní zaznamenávané atributy můžeme vidět na diagramu 4.2. V druhé části testování jsme rozšířili seznam o atribut určující, zda uživatel daný HRTF profil preferuje.

Abychom při každém spuštění doporučovacího systému nemuseli rekonstruovat tabulku doporučení, vytvořili jsme paralelně část databáze, která slouží jako 2D mapa, jejímž klíčem je dvojice identifikátor uživatele a HRTF profilu. Udržíme si statistické informace o průměrném hodnocení, minimu, maximu a počtu hodnocení. Celkový počet uložených dat je minimální možný, tedy $\mathcal{O}(|\mathcal{U}| \cdot |\mathcal{I}|)$.



Obrázek 4.2: Následující diagram ukazuje základní strukturu databáze, zobrazeny jsou pouze ty nejdůležitější parametry.

4.3 Rozhraní pro rozšíření

Modulární koncepce testovací aplikace umožňuje rozšíření různými druhy doporučovacích systémů. Můžeme vytvořit i jiné druhy adapterů pro různé druhy

databázi. Následující sekce popisuje rozhraní, které musí taková rozšíření implementovat, aby je bylo možné využít v rámci naší aplikace.

4.3.1 Logování

Logování je základním kamenem pro aktualizaci dat v našem doporučovacím systému. Každý zásah je zalogován a může být dále zpracováván doporučovacím systémem. Aby aplikace fungovala bez prodlení, jsou jednotlivé loggery spouštěny asynchronně.

Vlastní implementace loggeru musí implementovat dvě metody. Metoda `notify` přijímá jako parametr index aktuálně doporučovaného HRTF profilu. Metoda `log` přijímá celkem 3 parametry. Prvním parametrem je kosinus úhlu mezi skutečnou pozicí a pozicí označenou uživatelem, druhý parametr značí pořadové číslo pokusu a třetím parametrem je binární hodnota označující, zda uživatel tento HRTF profil preferuje či naopak.

Zdrojový kód 4.2: Zaregistrování a použití loggeru

```
var restLogger = new RestLogger( urlPrefix );
// Doporucovací system notifikuje zmenu HRTF profilu
recommender.registerListener( restLogger );
// Model loguje vsechny pokusy uzivatele
model.registerLogger( restLogger );
```

4.3.2 Doporučování

Aplikace umožňuje zaměnit jednoduše i doporučovací systém. Toho jsme využili při první fázi testování, kde byl použit `RandomRecommender` doporučovací systém, který doporučoval HRTF profily zcela náhodně. Pro druhou fázi testování jsme již využili `CollaborativeFilteringRecommender`, jenž využíval jádro kolaborativního filtrování.

Nově vytvořené doporučovací systémy musí splňovat rozhraní, které tvoří 2 metody. Pro doporučování HRTF profilů je nezbytné implementovat metodu `recommend`, jejíž návratová hodnota je index doporučovaného HRTF profilu. Tato metoda zároveň musí asynchronně notifikovat všechny posluchače (anglicky *listener*) o změně doporučovaného profilu. Druhá implementovaná požadovaná metoda `registerListener` slouží právě k zaregistrování všech posluchačů, což jsou převážně loggery. Jako první parametr očekává logger, tedy objekt splňující rozhraní 4.3.1.

Implementace doporučovacího systému může obsahovat ještě rozhraní typické pro loggery. Dává smysl, aby doporučovací systém byl notifikován o změně hodnocení. V tomto případě není nutné, aby se registroval jako svůj vlastní posluchač. Ukázka zdrojového kódu 4.3 dává jednoduchou představu, jak na to.

Zdrojový kód 4.3: Vytvoření doporučovacího systému a jeho prolinkování v aplikaci.

```
var recommender = new ContentRecommender(  
    hrtfData.length  
);  
// Nas recommender implementuje i logovani  
model.registerLogger(recommender);  
// Zaregistrujeme posluchace  
recommender.registerListener(consoleLogger);  
recommender.registerListener(firebaseLogger);  
// Pripojime doporučovani k zvukovému rozhraní  
soundSystem = SoundSystemFactory.create(  
    hrtfData ,  
    recommender  
);
```

Jelikož doporučování profilů probíhá synchronně, musí být takové doporučování relativně rychlé. Alternativním řešením může být asynchronní předvýpočet doporučovaných objektů. Takový přístup má nevýhodu, že zpožďuje reakci na hodnocení uživatelem. Problém zpoždění se dá eliminovat invalidací vypočtených hodnot, pokud již existuje hodnota pracující s aktualizovanými daty.

Testování

Poslední fází našeho experimentu je testování doporučovacího systému pro HRTF s reálnými uživateli. Cílem je ověřit možnost aplikace doporučovacích systémů za účelem zvýšení přesnosti uživatele oproti nepersonifikovanému HRTF profilu.

V první části se zaměříme na přípravu experimentu, podmínky a parametry testování. Druhá část popisuje průběh testování, včetně statistiky testerů. Třetí část interpretuje výsledky měření, jejich zdůvodnění a pohled na individuální problémy testerů.

5.1 Příprava

Experiment byl rozdělen do dvou fází testování. První fáze byla naplánována a provedena v týdnu od 3. 4. 2017. Během první fáze testeři pracovali s náhodným doporučováním a určovali pozici zdroje pouze v horizontální rovině. Výsledky této fáze sloužily jako počáteční data pro doporučovací systém (tím jsme se vyhnuli *cold-start* problému) a také jako měřítko průměrné nepřesnosti při použití nepersonifikovaného HRTF profilu.

Druhá fáze proběhla o dva týdny později, tedy v týdnu od 17. 4. 2017. Uživatelé tak měli čas si odpočinout a především jsme částečně eliminovali problém průběžného zlepšování (vizte 5.3.2). Během této fáze byly uživatelům doporučovány HRTF profily na základě predikce kolaborativního filtrování. Uživatelům byla navíc zpřístupněna možnost označit HRTF profil jako preferovaný. Ten by se měl použít v případě, že si je uživatel aktuálním pokusem (resp. přesností) zcela jist. Pro doporučovací systém měl takový pokus větší váhu.

5.1.1 Aplikace

Finální aplikace byla nahrána na server <http://vyuka.bernhauer.cz/dip/>, aby byla dostupná i testerům mimo Prahu. Při prvním načtení je uživatel pře-

směřován na autentizační stránku a teprve po úspěšném přihlášení je vpuštěn do testovací aplikace. Před jejím spuštěním je třeba vyčkat na stažení a zpracování všech HRTF profilů. Přihlášení uživatele a načtení dat trvá v průměru 60 sekund.

Samotná aplikace je rozdělena do dvou částí. První část zahrnuje 5 výstřelů (pokusů), během kterých uživatel může lokalizovat zdroj zvuku jako červenou postavu v prostoru. Uživatel nemá k dispozici žádný způsob grafického zaměřování. V druhé části grafická reprezentace zdroje zmizí a je možné jej lokalizovat pouze pomocí sluchu. Každý výstřel zaznamená nepřesnost a zdroj zvuku zcela zmizí po dobu přibližně 500 milisekund.

Uživatel má možnost rozhlížet se pohybem myši vlevo a vpravo. Označení pozice zdroje zvuku provádí levým nebo pravým tlačítkem myši. Přičemž pravé tlačítko má zároveň funkci preferovaného HRTF profilu. Takové označení má tak vícenásobnou váhu oproti standardním výstřelům levým tlačítkem.

Aplikace funguje ve většině moderních prohlížečů podporujících WebGL (s akcelerací na GPU) a Web Audio API. Ukončí se pouhým zavřením okna webového prohlížeče. Fungování aplikace nebude narušeno ani dočasným výpadkem internetového připojení, ani případnou aktualizací stránky. Opětovné načtení stránky způsobí pouze restartování stavu aplikace, a uživatel má tak opět prvních 5 výstřelů s viditelným zdrojem zvuku.

5.1.2 Parametry testování

Simulace prostorového zvuku metodou HRTF pracuje na principu simulace zvukového signálu v místě nahrávání HRTF profilu, tedy po modifikaci zvuku odrazy od vnějšího ucha. Pokud chceme přesně simulovat takový zvuk, je nutné reprodukovat zvuk v místě nahrávání HRTF. K běžnému prostorovému zážitku nám postačuje použití libovolných sluchátek. Aby simulace byla co nejpřesnější, probíhaly všechny testy pomocí sluchátek do uší („pecky“ nebo „špunty“). Tím byly zajištěny ekvivalentní podmínky simulace pro všechny účastníky testování. Testeři mohli použít svá sluchátka do uší nebo si půjčit erární sluchátka do uší značky Phillips, model SHE2550/10.

První fáze testování byla prováděna v kontrolovaném prostředí, neboť se dá předpokládat, že simulaci bude člověk využívat v běžném prostředí, ve kterém může být vyrušován. Byla vybrána místnost se střední úrovní hluku, studentské respirium v budově A fakulty stavební ČVUT na 12. patře. Tento faktor nám přidával především přirozený šum, protože simulace jednoho zvuku v tiché místnosti zní poněkud uměle.

Hlavním důvodem prezenčního testování je vysoká pravděpodobnost zmatení z orientace v prostoru pouze pomocí sluchu. Základním cílem dozoru tak bylo kromě seznámení s testovací aplikací také kontrola, že testeři zvládají lokalizaci zvuku a ovládání aplikace. Proto i testeři mimo Prahu byli většinou kontrolováni pomocí reálné databáze, aby mohli být včas upozorněni

na případné problémy. Druhá fáze již probíhala bez kontroly a testeré mohli testování provést kdekoli a kdykoli.

5.1.3 Podporované prohlížeče

Aplikace byla otestována na celkem 3 používaných webových prohlížečích. Otestované prohlížeče jsou Google Chrome 58 (64bit), Mozilla Firefox 51 (32bit), Opera 44. Ve všech prohlížečích může být nutné povolit nastavení akcelerace pomocí GPU. Další prohlížeče mohou být kompatibilní, v případě, že zcela nebo alespoň v experimentálním režimu podporují technologie WebGL a WebAudio.

5.2 Průběh testování

Následující sekce popisuje průběh testování a případné připomínky testerů k testování. Po příchodu testera na první fázi testování tomuto byl vysvětlen základní cíl práce, sděleny základní pokyny a doporučeny tipy, jakým způsobem se orientovat ve virtuálním prostoru. Samotné testování probíhalo přibližně 15 minut. Tester byl v případě nutnosti upozorněn na fakt, že se zcela neorientuje v prostoru a míří zády ke zdroji zvuku. Po testování proběhla diskuze zaměřená především na pocit ze simulovaného zvuku, případné problémy a postřehy z testování.

Na konci první fáze testeré dostali základní informace o druhé fázi. Před začátkem druhé fáze byla testerům rozeslána internetová adresa, na které mohli najít testovací rozhraní, kopii pokynů a novinky v implementaci. Testování probíhalo opět přibližně 15 minut. Poté mohli testeré případně doplnit své pocity, zejména pokud se týká lepšího pocitového vnímání z prostorového zvuku.

5.2.1 Pokyny před testováním

Uživatelům byla nejprve vysvětlena technika binaurálního audia a zároveň proveden průzkum znalosti takového řešení. Jen několik málo testerů znalo tuto techniku před samotným testováním. Následně byli seznámeni s cílem této diplomové práce.

Testery (obzvláště znalé binaurálního audia) bylo nutné informovat o použití sluchátek do uší, a byl vysvětlen i důvod. Přestože simulace prostorového zvuku metodou HRTF je možná s použitím libovolných stereofonních sluchátek, z hlediska přesnosti je potřeba simulovat zvuk ze stejného místa, ze kterého byl zvuk nahráván. Testeré se přihlásili do aplikace, připojili svá sluchátka a nastavili si hlasitost přehrávaného audia na příjemnou úroveň.

Následovalo představení samotného principu fungování aplikace. Testerům bylo vysvětleno základní ovládání a cíl hry. Prvních pět výstřelů, během kterých viděli zdroj zvuku jako červenou postavu, mohli využít k seznámení se

s ovládáním a především se naučit orientaci ve virtuálním prostoru pomocí zvuku. Po těchto pěti výstřelech byl spuštěn 15 minutový odpočet.

Doporučením bylo využití prvního výstřelu pro pochopení ovládání, a zbylé 4 výstřely pak využít následujícím způsobem. Uživatel zavře oči a pokusí se nalézt zdroj zvuku pomocí sluchu. Pokud si myslí, že zdroj zvuku našel, otevře oči a ověří si, že se zdroj zvuku nachází na obrazovce. Tak si jednoduše ověří, že správně rozpoznává směr vpředu a vzadu.

5.2.2 Gamifikace

Abychom učinili naši aplikaci zábavnější a více motivovali testery, implementovali jsme do aplikaci jako hru s gamifikačním prvkem. Cílem testerů tak bylo získat co nejvyšší skóre během 15 minut hraní a porazit co nejvíce testerů. Maximalizace skóre vyžadovala nejen co nejvyšší přesnost, ale také relativně rychlou reakci. Takový model více odpovídá realitě, kdy se nesnažíme pouze lokalizovat zvuk, ale lokalizovat ho v co nejkratším čase co nejpřesněji. Skóre uživatele u pro fázi f si můžeme definovat jako (5.1), kde R_u^f je množina všech pokusů v rámci prvních 15 minut, α_r je chyba pokusu r .

$$score_u^f = \sum_{r \in R_u^f} \cos(\alpha_r) \quad (5.1)$$

V případě kratší nebo delší hry můžeme skóre definovat také jako (5.2), kde t_r je časová známka pokusu r v minutách.

$$score_u^f = \frac{15}{\max t_r - \min t_r} \sum_{r \in R_u^f} \cos(\alpha_r) \quad (5.2)$$

5.2.3 Testeři

Testování se v první fázi zúčastnilo celkem 40 osob. Druhou fázi pak dokončilo 24 osob. Věková skupina testerů obsahovala populaci ve věku 18 až 30 let. Většina testerů byli studenti fakulty informačních technologií. Přibližně 12% testovaných subjektů byly ženy. Žádný tester netrpěl vážnějším zrakovým postižením zcela znemožňující orientaci v prostoru pomocí zraku.

5.2.3.1 Subjektivní hodnocení

Testeři měli po každé fázi prostor vyjádřit svůj pocit k samotné simulaci. Během první fáze často testeři zaznamenali problém s lokalizací zvuku, kdy nemohli rozpoznat levou a pravou stranu, protože úhel mezi pozicemi zcela vlevo a zcela vpravo byl deformován, a tak se jevil jako menší nebo větší. V některých případech zcela nevhodných profilů docházelo k pocitu, že se zvuk nachází uvnitř hlavy (byla zcela ztracena informace o pozici).

Po druhé fázi se většina testerů shodla na tom, že simulace jim byla mnohem příjemnější a že se cítili mnohem jistější ve svých střelách. Tento pocit potvrzují i výsledky. Naopak tato jistota často způsobila tzv. *miss-clicky*, které mohou ovlivnit výslednou statistiku.

5.3 Vyhodnocení

Následující část interpretuje výsledky získané nejen měřením z první a druhé fáze experimentu, ale také subjektivní hodnocení testerů. Mezi první a druhou fází očekáváme zlepšení přesnosti a pocitu z virtuálního audio prostoru.

5.3.1 Základní statistiky

Podívejme se na problém nejdříve z globálního měřítka. Tabulka 5.1 ukazuje změnu průměrného počtu střel během prvních 15 minut. Můžeme si všimnout mírného nárůstu, ale především velkého poklesu výběrového rozptylu. Během první fáze testování byl mezi uživateli propastný rozdíl mezi minimálním (~20) a maximálním (~80) počtem provedených pokusů během 15 minut. Zvýšení počtu pokusů může indikovat pozitivní změnu vnímání, kdy testeři mají lepší pocit z virtuálního prostoru. Ve výsledku to znamená přibližně jeden pokus za 15 sekund.

Tabulka 5.1: Průměrný počet střel během prvních 15 minut

	průměr	výběrový rozptyl
bez personifikace	50,707	689,162
s kolaborativním filtrováním	53,573	429,384

Na druhou stranu nám zvýšení frekvence pokusů neindikuje žádnou informaci o změně přesnosti, přestože pocit uživatele je pro nás důležitý. Tabulka 5.2 nám ukazuje průměrnou přesnost v jednotlivých fázích. U prvních 5 pokusů tester viděl zdroj zvuku, a snažil se tak (ne vždy) o co největší přesnost, což bylo zkomplikováno faktem, že k dispozici nebyl žádný způsob grafického zaměřování. Tato hodnota nám tedy označuje pomyslnou hranici přesnosti, pod kterou se dostaneme jen velmi těžší.

Už na první pohled vidíme, že doporučování s využitím kolaborativního filtrování je v průměru až o polovinu přesnější než využití nepersonifikovaného doporučování (simulováno náhodnými HRTF profily). Naši teorii podporuje i fakt, že se snížil výběrový rozptyl, tedy jednotlivé pokusy byly blíže průměru, než v případě pokusu s nepersonifikovaným HRTF.

Pokud porovnáme naše výsledky s výsledky experimentu [23], kde průměrná chyba pro nepersonifikovanou simulaci byla 39,2 se směrodatnou odchylkou 22,1, vidíme, že výsledky nejsou příliš rozdílné. V případě [23] dosahují sice horšího průměru, ale menšího rozptylu hodnot. Bohužel autoři neu-

vádějí, jakým způsobem vyřešili problém pokusů, jejichž výsledkem je nepřesnost větší než 90° . Na poli individualizovaného doporučování jsou výsledky již srovnatelné, přičemž [23] dosáhli nepřesnosti $23,3^\circ$ a směrodatné odchylky $9,82^\circ$.

Tabulka 5.2: Průměrná přesnost pokusů ve stupních

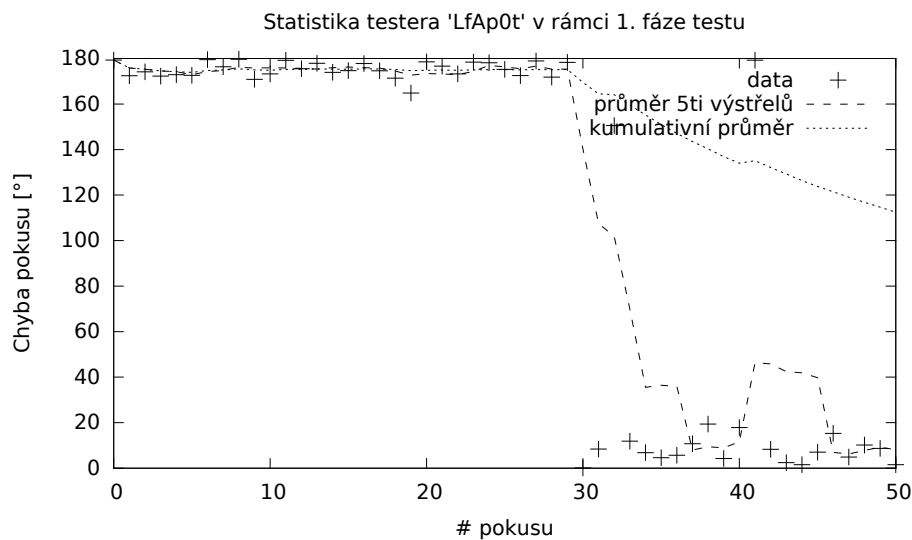
	průměr $[\circ]$	výběrový rozptyl $[\circ]$
bez personifikace	29,712	1958,596
s kolaborativním filtrováním	13,995	298,773
prvních 5 střel	5,653	16,791

5.3.2 Problémy

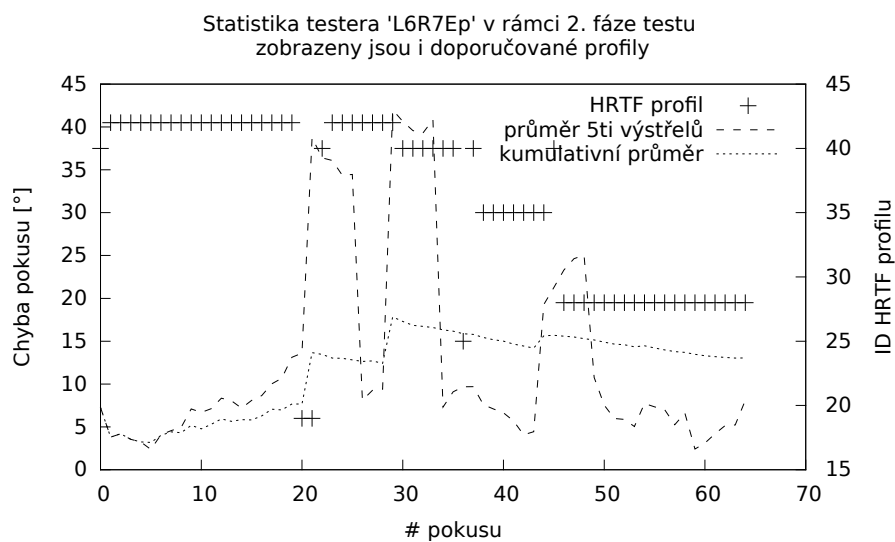
Už v předchozí části jsme zmínili problém prahu přesnosti. Každá testovací metoda má určitý práh přesnosti, který nejsme fyzicky schopni překonat. Tento může být dán jak technickými prostředky simulace, tak biologickými vlastnostmi testera. Pro detailnější zjištění tohoto prahu bychom museli provést řadu experimentů, mezi které patří i schopnost člověka v reálném prostředí rozlišovat směr pouhým otáčením hlavy (těla). Pro náš experiment jsme si jako takový práh zvolili hodnotu naměřenou při grafických testech, kdy uživatelé nebyli schopni určit přesný střed postavičky.

Uživatelé, jenž testovali metodu, v běžných situacích sluch k orientaci v prostoru nevyužívají. Proto i jejich schopnost navigace pomocí sluchu se zdokonalovala v čase. Někteří testeři preferovali mít během testování zavřené oči a soustředit se pouze na zvuk, jiní naopak. Nejvýrazněji byl tento problém vidět na vzdáleném (před)testerovi LfAp0t, který první polovinu testování označoval pozici na opačné straně, vizte graf 5.1. Okolo 30. pokusu si již zvykl na orientaci pomocí sluchu a opravil tak své pokusy.

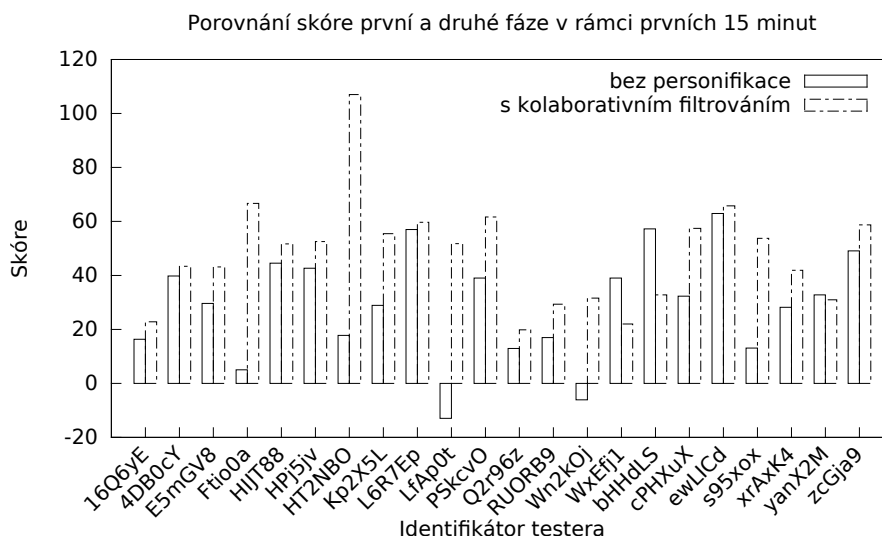
Po druhé fázi testování se testeři ozývali s připomínkami ohledně testování, a jedním z problémů, které řešili, byly takzvané *miss-clicky*, tedy nechtěná označení pozice zvuku. Takové označení bychom do naší statistiky neměli započítávat, a proto v naší implementaci vždy po každém 5. pokusu odečítáme od průměru minimální a maximální naměřenou chybu. Navzdory chybě v implementaci, která byla nalezena až po testování, tyto *miss-clicky* neovlivnily výrazně výsledky. Význam tohoto problému můžeme vidět na testerovi L6R7Ep, který během druhé fáze udělal dvě neúmyslné chyby, vizte graf 5.2. Po první chybě je uživateli skokově nabídnut jiný profil, právě z důvodu okamžitého započítání nepřesnosti do průměru. Při druhé (neúmyslné) chybě již mu je nalezen nový profil. A poslední neúmyslná chyba jen způsobí další skok.



Obrázek 5.1: Následující graf ukazuje nepřesnost jednotlivých pokusů pro testera LfAp0t v první fázi experimentu. Můžeme si všimnout počátečního zmatení z virtuálního prostoru a následnou sebekorekci.



Obrázek 5.2: Vliv miss-clicku na doporučování HRTF profilu.



Obrázek 5.3: Porovnání testovaných subjektů a změny jejich skóre mezi oběma fázemi. Vyšší hodnota je lepší.

5.3.3 Detailní pohled

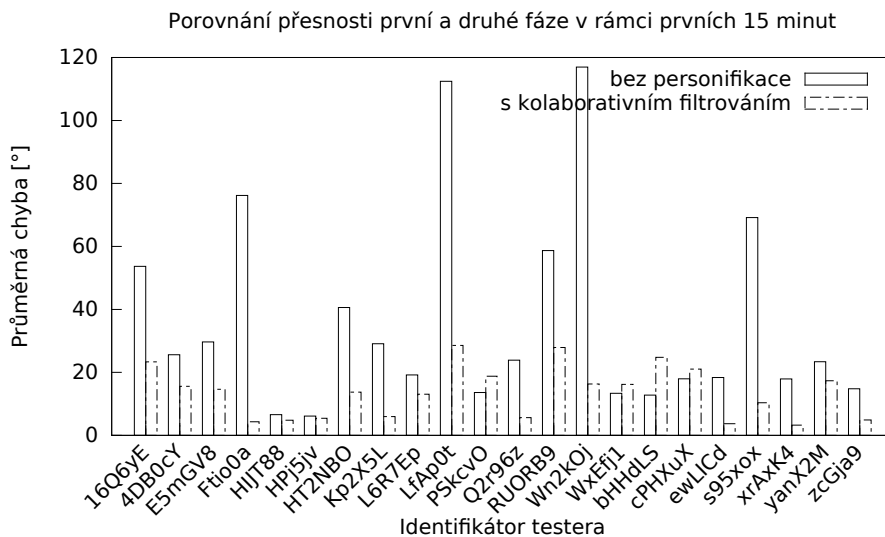
Souhrnné statistiky jsou sice v pořádku a podporují naši teorii, ale musíme ověřit, zda se nejedná pouze o lokální trend jednoho uživatele. Následující část popisuje detailní pohled na věc. V první části se zaměříme na porovnání v rámci našeho bodovacího systému, který je základem gamifikační funkce. Druhá a třetí část se zabývá standardním hodnocením. To pracuje s klasickou chybou jakožto úhlem mezi předpokládanou a skutečnou pozicí.

5.3.3.1 Statistika založená na skóre

Podívejme se nejprve na detailní statistiky k naší navržené gamifikační verzi. Graf 5.3 zobrazuje skóre jednotlivých hráčů získané v první a druhé fázi experimentu. Můžeme si všimnout radikálního zlepšení u většiny hráčů při druhé fázi testování.

Zlepšení na základě postupného přizpůsobování orientace v prostoru pomocí sluchu jsme eliminovali časovým odstupem druhé fáze. Skóre zahrnuje celkem 2 složky, přesnost (kosinus úhlu) a počet pokusů během 15 minut. Vzhledem k malé změně průměrného počtu pokusů (vizte 5.3.1) můžeme předpokládat, že jde především o zlepšení přesnosti.

Zaměříme se nyní na testery **WxEf j1** a **bHHdLS**, kteří mají znatelné zhoršení oproti první fázi. Změnu přesnosti u testera **yanX2M** nebudeme považovat za statisticky významnou. Individuálním pohledem na testera **bHHdLS** se zabýváme v 5.3.4.3. Funkce kosinus ve funkci (5.1) mění škálování hodnocení na



Obrázek 5.4: Porovnání průměrné nepřesnosti během prvních 15 minut mezi fázemi experimentu.

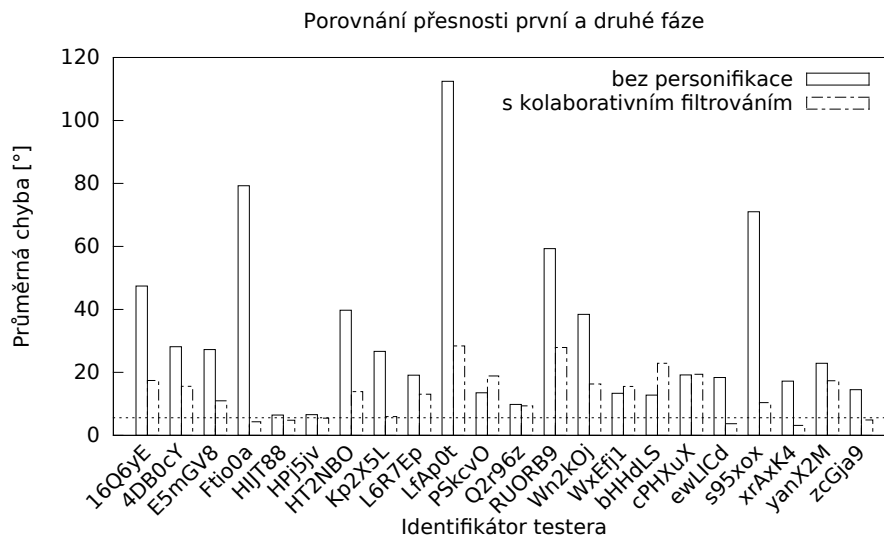
nelineární. Hodnoty blízké zcela přesnému (resp. nepřesnému) hodnocení mají menší vzdálenost. Naopak pokusy blízké levé či pravé straně mají vzdálenost větší. Proto se zkusme podívat na stejný problém, ale jako sledovanou hodnotu použijme průměrnou přesnost během prvních 15 minut.

Graf 5.4 potvrzuje naši domněnku, že změna skóre (a přesnosti) u testera **yanX2M** je statisticky nevýznamná. Pokud měříme přímo přesnost, tester je na tom lépe než v případě herního bodování. Bohužel problém s testery **WxEfj1** a **bHHdLS** jsme nevyřešili.

5.3.3.2 Průměrná přesnost

V předchozí části jsme se zaměřili na detailní pohled z hlediska herního hodnocení, které se počítá pouze z prvních 15 minut. Mnoho uživatelů však testovalo déle než 15 minut. Mohlo by být zajímavé podívat se na kompletní statistiku testerů a všechny jejich pokusy. Například u uživatele **LfAp0t** a **Wn2k0j** očekáváme snížení nepřesnosti v první fázi, jelikož oba měli ze začátku problém s orientací ve virtuálním prostoru.

Na grafu 5.5 vidíme, že ani využití celého intervalu testování nám nepomůže vysvětlit zhoršení přesnosti. Můžeme tedy konstatovat, že průměrná přesnost jednotlivých hráčů nemusí být vždy lepší. To platí alespoň z hlediska celého testování, v 5.3.3.3 se zaměříme na hodnocení kvality kolaborativního filtrování, při zohlednění jeho fungování.



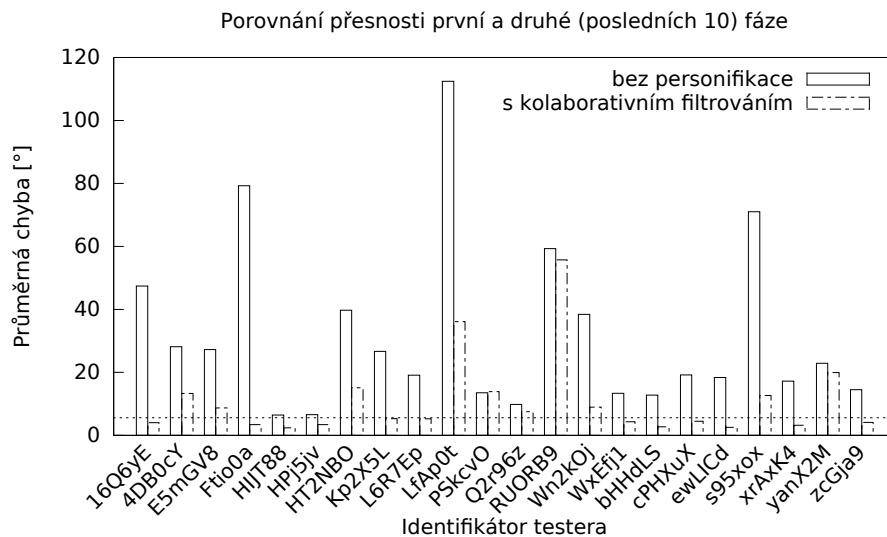
Obrázek 5.5: Porovnání průměrné nepřesnosti (všech pokusů) mezi fázemi experimentu. Zvýrazněnou máme hodnotu průměrné nepřesnosti během prvních 5ti pokusů.

5.3.3.3 Specifika kolaborativního filtrování

Bohužel změna úhlu pohledu na problém nám nepomůže vyřešit, zda se skutečně jedná o zhoršení nebo pouze nevýznamnou statistickou odchylku. Podívejme se na princip fungování kolaborativního filtrování. Kolaborativní filtrování funguje obstojně, pokud jsou dobře nastavené parametry a máme dostatek dat.

Kolaborativní filtrování tedy vyžaduje čas, aby našel vhodný HRTF profil k doporučení. Obzvláště při problému *cold-start* bude tento faktor viditelnější. Teoreticky je tedy možné, že u subjektů **WxEfj1** a **bHHdLS** kolaborativní filtrování konverguje pomaleji nebo nastaly problémy během testování.

Graf 5.6 zobrazuje porovnání první fáze testování s posledními 10 pokusy fáze druhé, průměrně se jedná o 20% všech pokusů. Můžeme si všimnout výrazného zlepšení mezi všemi uživateli. Pokud se zaměříme na celkové statistiky posledních 10 pokusů v druhé fázi, získáme průměrnou nepřesnost $6,635^\circ$ s výběrovým rozptylem $2,243^\circ$. To je výrazné zlepšení oproti globální statistice a výrazné zlepšení oproti [23]. Důvodem takového zlepšení je především zaměření experimentu [23] na modelování HRTF pro více směrů, zatímco náš experiment se zaměřuje pouze na směr vpředu.



Obrázek 5.6: Porovnání nepřesnosti se zohledněním kolaborativního filtrování

5.3.4 Individuální statistiky

Následující část popisuje statistiky individuálních testerů. Jedná se o speciální případy popisující zvláště zajímavé situace.

5.3.4.1 Tester hur2Zz

Subjekt s identifikátorem **hur2Zz** se přihlásil do první fáze testování pod účtem **jzr1i0**. Není tudíž zahrnut v našich komplexních statistikách. Zajímavostí tohoto případu je především potřeba vyrovnat se s problémem *cold-start*. Sice existuje alespoň částečný profil, který by měl odpovídat stejnému uživateli, nicméně přihlášený uživatel žádná data k dispozici nemá.

Na grafu C.3 můžeme vidět poměrně rychlou konvergenci kolaborativního filtrování a to včetně dočasného vychýlení. Oproti C.2 pozorujeme klesající tendenci. Průměrná přesnost druhé fáze je lepší, než ve fázi první.

5.3.4.2 Tester Wn2kOj

Subjekt **Wn2kOj** popisuje celkem 2 typické problémy. Prvním problémem, kterého si můžeme všimnout na grafu C.4, je zmatení z orientace ve virtuálním prostoru pomocí sluchu. Přibližně prvních 10 pokusů bylo mířeno přesně opačným směrem. Uživatel se díky samoopravnému mechanismu otočil.

Druhým problémem, kterého si můžeme všimnout ve druhé fázi (graf C.5), je vliv *miss-clicku*, případně opět zmatení, kdy původní doporučený profil, vykazoval nepřesnost v řádech jednotek stupňů. Po jedné chybě je původní profil zaměněn za profil s identifikátorem 25. Ten sice nevykazuje tak dobré

hodnoty jako původní profil, ale stále se jedná lepší profil než nepersonifikovaný.

5.3.4.3 Tester bHHdLS

Jedním z testerů, kteří se v průměru zhoršili, je i tester **bHHdLS**. Na grafu C.1 si můžeme všimnout náhodného zkoušení různých profilů. Teprve okolo 20 pokusu kolaborativní filtrování konverguje k HRTF profilu s identifikátorem 14. Ten má průměr 5ti posledních výstřelů pod 5° , ale kumulativní průměr trpí počátečním hledáním. Proto jak píšeme v 5.3.3.3 je potřeba srovnávat přesnost pouze na intervalu, kde kolaborativní filtrování konverguje k ideálnímu profilu.

Závěr

Analyzovali jsme nejčastější přístupy k simulaci 3D zvuku. Identifikovali jsme dvě nejčastěji používané metody získávání HRTF profilů. Přičemž jsme narazili na problém, že nahrávání personalizované HRTF je časově náročné a využití jednoho standardního HRTF profilu naopak rychlé, ale nepřesné.

Na základě analýzy jednoduchých doporučovacích systémů jsme navrhli několik metod pro podporu generování personifikovaného binaurálního audia. Dále jsme navrhli několik postupů testování přesnosti HRTF profilu a to včetně hodnocení kvality doporučování HRTF.

Naimplementovali jsme webovou aplikaci, která implementuje kolaborativní filtrování. Technikou gamifikace poskytuje ideální prostředí pro testování. Díky využití moderních technologií WebGL a WebAudio jsme naimplementovali jednoduchou hru, která je tak dostupná velkému množství zájemců.

Integraci s kolaborativním filtrováním jsme otestovali na reálných uživateliích ve dvou fázích. Uživatelé si vyzkoušeli jak testování v kontrolovaném prostředí, tak i možnost domácího měření, které je základem pro komerční rozšíření této metody.

Zhodnocení

Z jednoduchých doporučovacích systémů jsme si vybrali kolaborativní filtrování pro jednoduchost myšlenky. Výsledky z testování kolaborativního filtrování mohou sloužit pro návrh podobnostní funkce, kterou použijeme v doporučování založeném na obsahu. Takto nám mohou poskytnout podrobnější znalosti a informace o parametrech HRTF, abychom byli schopni určit jejich závislost.

Navržené metody v 3. kapitole mohou sloužit pro další rozvoj modelování HRTF využitím nejen doporučovacích systémů. Vybraná metoda reflektuje současné požadavky a reálně dostupné možnosti.

Implementovaná aplikace je s ohledem na technologie WebGL a WebAudio použitelná v praxi, bez nutnosti investice do cenově náročnějších VR setů.

Časová náročnost celého procesu netrvá více než 15 minut, což je v porovnání s hodinami během nahrávání ve studiu nepatrný kompromis.

Většinu problémů jsme identifikovali ještě před testováním, a tak se nám je podařilo ve většině případů úspěšně eliminovat. Za použití kolaborativního filtrování a vybraného modelu jsme dosáhli průměrné nepřesnosti $14,0^\circ \pm 17,3^\circ$. Pokud jsme se zaměřili pouze na posledních 10 pokusů, pak se průměrná nepřesnost snížila na $6,6^\circ \pm 1,5^\circ$. Za což může především konvergence kolaborativního filtrování k suboptimálnímu nebo optimálnímu profilu.

Další kroky

Testování druhé části až na jednu výjimku probíhalo se stejnými uživateli a již získanými daty. Netestovali jsme tedy chování kolaborativního filtrování při *cold-start* problému. Ač tento jeden případ dopadl dobře, nemůžeme rozhodnout o chování v obecném případě.

Na druhou stranu zopakováním celého experimentu, tedy použitím dvou fází, kdy doporučovací systém v první fázi doporučuje profily náhodně a v druhé již využívá kolaborativní filtrování, můžeme získat stejné výsledky. Časová náročnost takového měření bude sice dvojnásobná, ale 30 minut je akceptovatelná doba.

Přesto by dalším krokem mělo být otestování tohoto modelu na zcela nových uživateli. S opravením implementační chyby v benevolentnějším průměrování. A případné navržení alternativní metody.

Budoucí vývoj

Jelikož se jedná o poměrně neprozkoumanou oblast existuje několik směrů, kterými se můžeme vydat. Jako základ nám poslouží analýza testovacích modelů a analýza využití doporučovacích systémů v kombinaci s HRTF profily.

- Seznam doporučovacích systémů není zcela úplný a je tak otázkou dalších experimentů, které doporučovací systémy budou vhodné pro procházení HRTF databází. Velký potenciál je ve využití doporučovacích systémů založených na obsahu, které mohou odstranit problémy kolaborativního filtrování.
- Z hlediska implementace stojí za zmínku úprava kolaborativního filtrování. Prohození uživatelsky zaměřeného algoritmu na objektově zaměřený může urychlit doporučování, jelikož počet doporučovaných HRTF profilů zůstává stejný. Analýza takového systému může být dalším bodem zkoumání, neboť oproti klasickým doporučovacím systémům nemusí být matice hodnocení řídká. V takovém případě může být výpočetní část doporučovacího systému přenesena na server.

- Efektivita některých výpočtů by mohla být také urychlena přenesením z CPU na GPU díky využití technologie WebGL. Taková optimalizace bude nezbytná v případě simulace více zdrojů zvuků.
- Z důvodu rozšíření VR setů nebo případné specifikace WebVR, může dojít k rozšíření možnosti hodnotit (simulovat) i jiné pozice v prostoru. Navíc tím odstraníme některé současné problémy.

Oblastí dalšího vývoje je nepřehledné množství a samotné téma virtuální reality je značně populární. Tato práce představuje kombinaci 3 různých technologií (binaurální audio, kolaborativní filtrování, webové technologie), které spolupracují a poskytují tak jednoduchý nicméně nad očekávání efektivní způsob simulace zvuku ve virtuálním prostředí.

Literatura

- [1] DUDA, R. O.: Simple Spatial Audio Systems [online]. 1996–2000, [cit. 2017-03-01]. Dostupné z: <http://interface.cipic.ucdavis.edu/sound/tutorial/simplesys.html>
- [2] NÝVLT, V.; KUŽNÍK, J.: Nová sluchátka Ambeo umožní nahrávat videa s prostorovým zvukem [online]. 2017, [cit. 2017-03-01]. Dostupné z: http://technet.idnes.cz/sluchatka-s-binauralnimi-mikrofony-sennheiser-ambeo-pc4-/tec_audio.aspx?c=A170106_171917_tec_audio_nyv
- [3] DUDA, R. O.: HRTF-Based Systems. 1996–2000, [cit. 2017-03-02]. Dostupné z: <http://interface.cipic.ucdavis.edu/sound/tutorial/hrtfsys.html>
- [4] DYUSSEKENOVA, S.: *Modelování HRTF*. Diplomová práce, České vysoké učení technické v Praze, Fakulta elektrotechnická, Praha, 2015.
- [5] DURAI SWAMI, R.: Introduction to HRTFs [online presentation]. [cit. 2017-03-03]. Dostupné z: http://www.umi.acs.umd.edu/~ramani/cmssc828d_audio/HRTF_INTR0.pdf
- [6] XIE, B.: *Head-related transfer function and virtual auditory display*. J. Ross Publishing, druhé vydání, ISBN 978-1-60427-070-9.
- [7] HADRAVA, J.; BERNHAUER, D.: The Audio Component of a Virtual Reality. In *17th Conference of Czech and Slovak Physicists Proceedings*, Košice: Slovak Physical Society, 2012, ISBN 978-80-970625-4-5, s. 167–168.
- [8] DUDA, R. O.: Psychoacoustics of Spatial Hearing. 1996–2000, [cit. 2017-03-02]. Dostupné z: <http://interface.cipic.ucdavis.edu/sound/tutorial/psych.html>

- [9] POTISK, T.: Head-Related Transfer Function. 2015, [cit. 2017-03-14]. Dostupné z: http://mafijs.fmf.uni-lj.si/seminar/files/2014_2015/Seminar_Ia_Head-Related__Transfer_Function_Tilen_Potisk.pdf
- [10] Prem MELVILLE, V. S.: Recommender Systems. In *Encyclopedia of Machine Learning*, Springer, 2010, ISBN 978-0-387-30768-8.
- [11] RICCI, F.; ROKACH, L.; SHAPIRA, B.: Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, kapitola 1, Springer Science+Business Media, 2011, ISBN 978-0-387-85820-3.
- [12] LOPS, P.; de GEMMIS, M.; SEMERARO, G.: Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, kapitola 3, Springer Science+Business Media, 2011, ISBN 978-0-387-85820-3.
- [13] DESROSIERS, C.; KARYPIS, G.: A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*, kapitola 4, Springer, 2011, ISBN 978-0-387-85820-3.
- [14] ZEZULA, P.; AMATO, G.; DOHNAL, V.; aj.: *Similarity Search The Metric Space Approach* [online]. Springer, 2006, ISBN 978-0-387-29151-2.
- [15] MELUCCI, M.: Vector-Space Model. In *Encyclopedia of Database Systems*, Springer, 2009, ISBN 978-0-387-39940-9.
- [16] BANDYOPADHYAY, S.; SAHA, S.: Similarity Measures. In *Unsupervised Classification*, kapitola 3, Springer, 2013, ISBN 978-3-642-32451-2.
- [17] DEZA, M. M.; DEZA, E.: *Encyclopedia of Distances*. Springer, 2009, ISBN 978-3-642-00233-5.
- [18] MÜLLER, M.: *Information Retrieval for Music and Motion*. Springer, ISBN 978-3-540-74047-6.
- [19] DEKHTYAR, A.: Collaborative Filtering and Recommender Systems [online]. 2009, [cit. 2017-07-08]. Dostupné z: <http://users.csc.calpoly.edu/~dekhtyar/466-Fall2010/lectures/lec11.466.pdf>
- [20] SINHA, R. R.; SWEARINGEN, K.: Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001, citováno v [11].
- [21] GROSSMAN, D. A.; FRIEDER, O.: *Information Retrieval: Algorithms and Heuristics* [online]. Springer, druhé vydání, 2004, ISBN 978-1-4020-3005-5.

-
- [22] LUO, Y.; ZOTKIN, D. N.; DURAI SWAMI, R.: Virtual Autoencoder based Recommendation System for Individualizing Head-Related Transfer Functions. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.
- [23] LUO, Y.; ZOTKIN, D. N.; DURAI SWAMI, R.: Gaussian Process Models for HRTF based Sound-Source Localization and Active-Learning. In *IEEE Journal of Selected Topics in Signal Processing, March 2014*, břez-en 2014.
- [24] W3C Working Group: *HTML5 Differences from HTML4* [online]. prosinec 2014, [cit. 2017-04-10]. Dostupné z: <https://www.w3.org/TR/html5-diff/>
- [25] W3C Working Group: *A vocabulary and associated APIs for HTML and XHTML* [online]. říjen 2014, [cit. 2017-04-10]. Dostupné z: <https://www.w3.org/TR/html5/scripting-1.html#the-canvas-element>
- [26] Khronos Group: *WebGL Specification (Working Draft)* [online]. duben 2017, [cit. 2017-04-10]. Dostupné z: <https://www.khronos.org/registry/webgl/specs/latest/1.0/>
- [27] DEVERIA, A.: Can I use ECMAScript 5? [online]. [cit. 2017-03-07]. Dostupné z: <http://caniuse.com/#search=es5>
- [28] DEVERIA, A.: Can I use ECMAScript 6? [online]. [cit. 2017-03-07]. Dostupné z: <http://caniuse.com/#search=es6>
- [29] CATUHE, D.; ROUSSET, D.: Babylon.JS - Specifications [online]. [cit. 2017-04-12]. Dostupné z: <https://www.babylonjs.com/#specifications>
- [30] DEVERIA, A.: Can I use WebGL? [online]. [cit. 2017-04-27]. Dostupné z: <http://caniuse.com/#feat=webgl>
- [31] DEVERIA, A.: Can I use WebGL 2.0? [online]. [cit. 2017-04-27]. Dostupné z: <http://caniuse.com/#feat=webgl>
- [32] three.js authors: *three.js docs* [online]. [cit. 2017-04-12]. Dostupné z: <https://threejs.org/docs/>
- [33] W3C: *Web Audio API*. prosinec 2015, [cit. 2017-04-12]. Dostupné z: <https://www.w3.org/TR/webaudio/>
- [34] ADENOT, P.: *Web Audio API performance and debugging notes* [online]. [cit. 2017-04-12]. Dostupné z: <https://padenot.github.io/web-audio-perf/#pannernode-when-panningmodel--hrtf>

- [35] WOŹNIAK, T.: Implementing Binaural (HRTF) Panner Node with Web Audio API [online]. 2015, [cit. 2017-02-27]. Dostupné z: <https://codeandsound.wordpress.com/2015/04/08/implementing-binaural-hrtf-panner-node-with-web-audio-api/>
- [36] DEVERIA, A.: Can I use Web Audio API? [online]. [cit. 2017-04-27]. Dostupné z: <http://caniuse.com/#feat=audio-api>
- [37] WebVR Community Group: *WebVR*. duben 2017, [cit. 2017-04-21]. Dostupné z: <https://w3c.github.io/webvr/spec/latest/>
- [38] Pricing | Firebase [online]. [cit. 2017-03-19]. Dostupné z: <https://firebase.google.com/pricing/>
- [39] Firebase [platform]. [cit. 2017-02-04]. Dostupné z: <https://firebase.google.com/>
- [40] Firebase Documentation [online]. [cit. 2017-04-27]. Dostupné z: <https://firebase.google.com/docs/>
- [41] ALGAZI, V. R.; DUDA, R. O.; THOMPSON, D. M.; aj.: The CIPIC HRTF Database. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2001*, 2001. Dostupné z: http://interface.cipic.ucdavis.edu/data/doc/CIPIC_HRTF_Database.pdf
- [42] GAMPER, H.: Head-related transfer function interpolation in azimuth, elevation, and distance [online]. listopad 2013. Dostupné z: <http://asa.scitation.org/doi/pdf/10.1121/1.4828983>

Seznam použitých zkratek

- AC** Adjusted Cosine – podobnostní míra využívaná v kolaborativním filtrování
- API** Application Programming Interface – rozhraní pro programování aplikací
- CF** Collaborative Filtering – kolaborativní filtrování
- CLI** Command Line Interface – příkazový řádek
- CPU** Central Processing Unit – centrální procesorová jednotka
- CV** Cosine Vector – podobnostní míra využívaná v kolaborativním filtrování
- DFT** Discrete Fourier Transform – diskrétní Fourierova transformace
- DTW** Dynamic Time Warping – vzdálenostní míra využívaná v information retrieval
- FFT** Fast Fourier Transform – rychlá Fourierova transformace
- FPS** First Person Shooter – akční hra z pohledu první osoby
- GPU** Graphic Processing Unit – grafický procesor
- HRIR** Head-Related Impulse Response – impulsová odezva vztažená k hlavě
- HRTF** Head-Related Transfer Function – přenosová funkce vztažená k hlavě
- HTML** HyperText Markup Language – značkovací jazyk pro implementaci webových stránek
- IDF** Inverse Document Frequency – inverzní frekvence termu v dokumentech
- ILD** Interaural Level Difference – rozdíl hlasitostí mezi pravým a levým uchem

A. SEZNAM POUŽITÝCH ZKRATEK

ITD Interaural Time Difference – rozdíl času zaznamenání zvuku mezi pravým a levým uchem

JSON JavaScript Object Notation – formát pro výměnu dat založený na JavaScriptu

KEMAR Knowles Electronic Manikin for Acoustic Research – umělá hlava pro výzkumy v oblasti akustiky

kNN k Nearest Neighbor - k nejbližších sousedů

kRNN k Reverse Nearest Neighbor - sousedé, jímž jsem mezi k nejbližšími sousedy

MSD Mean Square Difference – střední čtvereční rozdíl

MSE Mean Square Error – střední čtvereční chyba

MVC Model-View-Controller – architektonický vzor pro návrh aplikací

PC Pearson Correlation – podobnostní míra využívaná v kolaborativním filtrování

RMSE Root Mean Square Error – míra využívaná k hodnocení kvality doporučovacích systémů

RTF Room Transfer Function – přenosová funkce vztažená k prostředí

SRC Spearman Ranking Correlation – podobnostní míra využívaná v kolaborativním filtrování

TF Term Frequency – normovaná frekvence termu v dokumentu

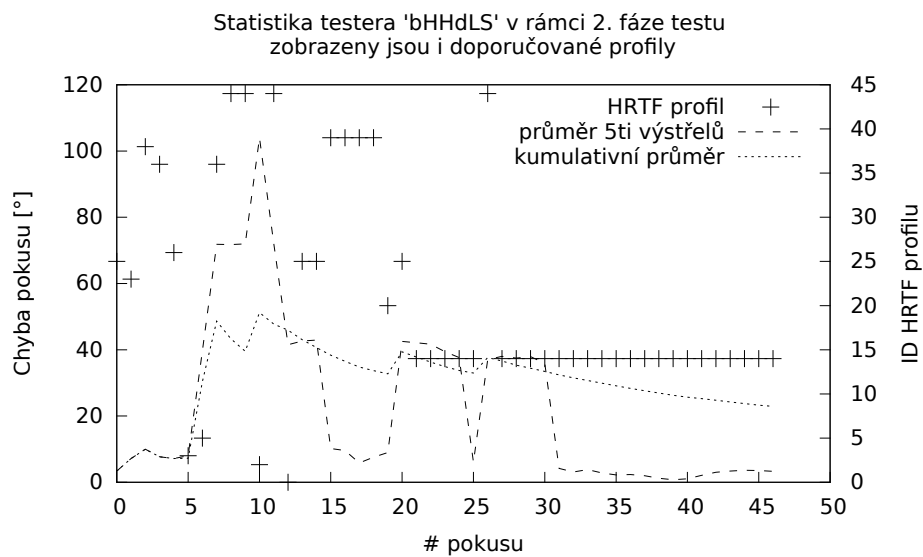
VR Virtual Reality – virtuální realita

VSM Vector-Space Model – model využívaný v doporučování textových dokumentů založeném na obsahu

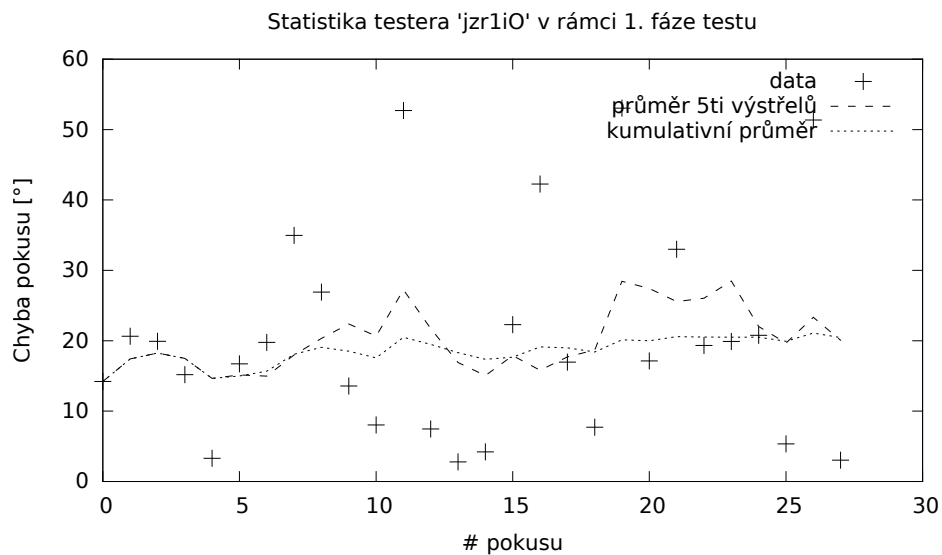
Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
_ impl.....	zdrojové kódy implementace
_ thesis	zdrojová forma práce ve formátu L ^A T _E X
_ plots.....	zdrojové kódy grafů a zpracování naměřených dat
text	text práce
_ thesis.pdf	text práce ve formátu PDF
_ thesis.ps	text práce ve formátu PS
data	
_ measured.json	databáze s naměřenými daty
_ plots.....	vygenerované grafy

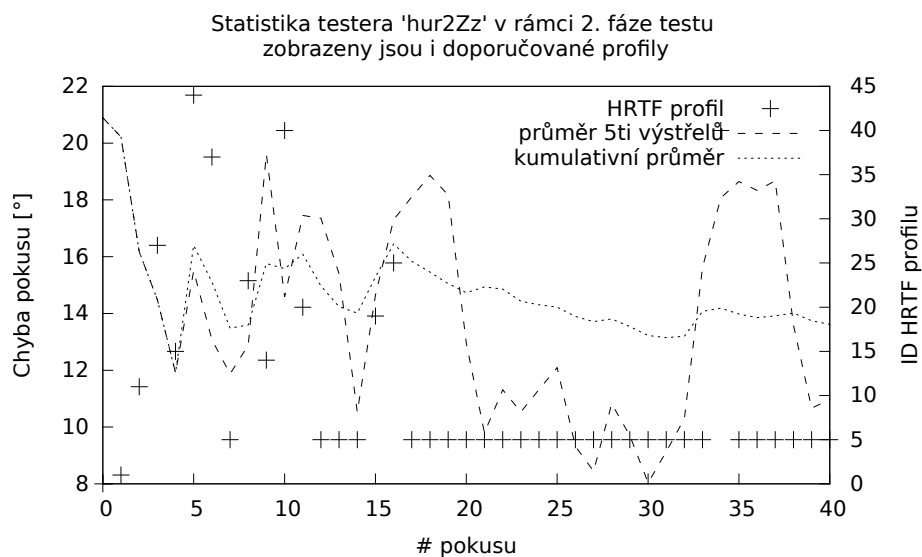
Profily uživatelů



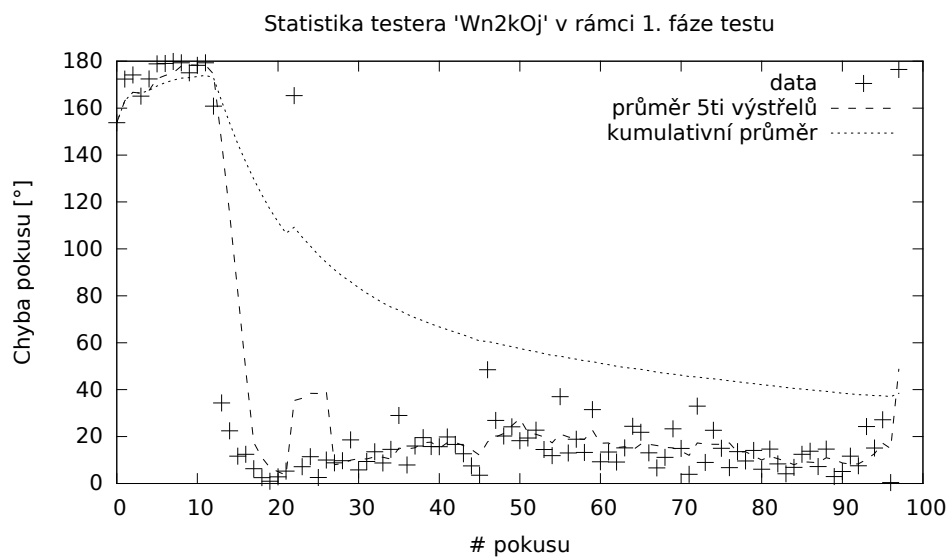
Obrázek C.1: Profil testera bHHdLS v druhé části testování. Místo nepřesností pokusů je uveden výběr HRTF profilu.



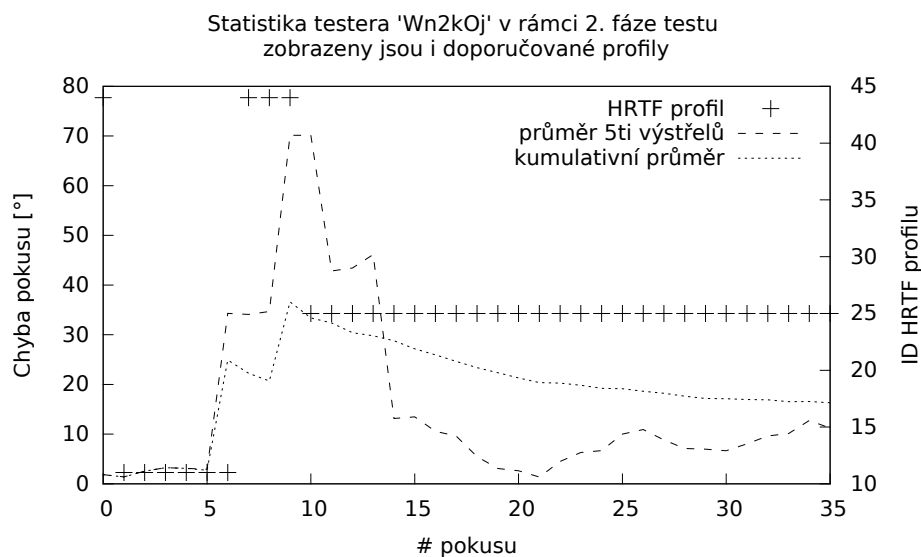
Obrázek C.2: Profil testera hur2Zz v první části testování.



Obrázek C.3: Profil testera hur2Zz v druhé části testování. Místo nepřesností pokusů je uveden výběr HRTF profilu.



Obrázek C.4: Profil testera Wn2kOj v první části testování.



Obrázek C.5: Profil testera Wn2kOj v druhé části testování. Místo nepřesností pokusů je uveden výběr HRTF profilu.