Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Laurea Magistrale in INGEGNERIA INFORMATICA

CONFRONTO DI HEAD-RELATED TRANSFER FUNCTION PERSONALIZZATE IN AMBIENTI DI REALTÀ VIRTUALE

Supervisor
STEFANO GHIDONI
Università di Padova

Co-supervisor
MICHELE GERONAZZO
Università di Aalborg

Master Candidate
DIEGO OMICIUOLO

RINGRAZIO

Abstract

I recenti sviluppi nella tecnologia audio immersiva hanno motivato una proliferazione di renderer audio binaurali utilizzati per creare contenuti audio spazializzati. I renderer binaurali cercano di fornire le corrette informazioni acustiche all'udito umano per riprodurre un'immagine sonora 3D in cuffia. La combinazione di tali renderer con i sistemi di realtà virtuale, con stimolazione multimodale e fino a sei gradi di libertà di movimento, pongono quindi nuove sfide alla valutazione della qualità dell'esperienza immersiva ed in particolare della qualità audio.
In questa tesi viene presentato una metodologia per la valutazione comparativa di diversi renderer binaurali basati su personalizzazione del contributo acustico dell'ascoltatore nella simulazione, contenuto nelle head-related transfer function (HRTF). Il metodo si basa sulla classificazione per eliminazione dei renderer durante l'esplorazione di una realtà virtuale audiovisiva in tre ambienti con caratteristiche diverse. Tale metodo viene applicato su una lista di attributi basati su dizionari di aggettivi percettivi che descrivono la qualità dell'immagine audio all'interno di una scena. I renderer messi a confronto per ogni soggetto sono personalizzati su metriche di selezione basate sulla similitudine dell'orecchio del soggetto, delle misure antropometriche della testa e la combinazione di tali informazioni.
Nei risultati ottenuti appare una chiara preferenza, riscontrabile sia dal punto di vista degli ambienti che da quello degli attributi, per quest'ultimo renderer che combina le informazioni sulle due selezioni migliori delle metriche utilizzate.

Indice

Elenco delle figure

Elenco delle tabelle

1

Introduzione

La realtà virtuale (VR dall'inglese Virtual Reality) è il termine utilizzato per indicare una realtà simulata, ovvero una percezione di un mondo reale attraverso una simulazione tramite dispositivi elettronici. I primi quindici anni del XXI secolo hanno visto un importante e rapido sviluppo della realtà virtuale. L'evoluzione delle tecnologie informatiche permette di muoverci in ambienti ricreati in tempo reale, interagendo con gli oggetti presenti in essi. L'ascesa degli smartphone con display ad alta densità e capacità grafiche 3D ha permesso di dare vita ad una generazione di dispositivi di realtà virtuale leggeri e pratici. L'industria dei videogiochi ha continuato a guidare lo sviluppo della realtà virtuale di consumo senza sosta e di conseguenza i controller di movimento e le interfacce umane naturali sono entrati a far parte delle attività quotidiane dell'interazione uomo-computer.
Si possono distinguere 2 principali tipologie [10]: *realtà virtuale immersiva* e *realtà virtuale non immersiva*. Nel primo caso l'utente si trova all'interno di un ambiente costruito intorno a sé e la simulazione avviene tramite alcune periferiche:

- *visore*: uno schermo (smartphone o un casco apposito) posto vicino agli occhi in cui riprodurre la parte visiva della simulazione, escludendo il mondo reale dalla visuale dell'utente. Questa periferica può essere utilizzata inoltre per la rilevazione dei movimenti della testa per poter replicare la stessa direzionalità nell'ambiente virtuale.

- *auricolari*: tramite questa periferica l'utente può percepire i suoni riprodotti all'interno dell'ambiente virtuale, isolato dai rumori esterni e potenzialmente rumorosi.

- *sensori*: usando questi dispositivi sarà possibile tracciare e riprodurre le stesse azioni alla parte del corpo cui essi vengono applicati (movimento di una mano, braccio,...).

Nel secondo caso invece la differenza fondamentale consiste nell'utilizzo di un monitor, che rimpiazza l'uso di un visore, che funge quindi come una finestra sul mondo virtuale tridimensionale e sulla quale potrà operare tramite degli appositi joystick o più in generale tramite un mouse e una tastiera.
Al giorno d'oggi la VR viene utilizzata in diversi campi, da quello dell'intrattenimento a quello educativo, gaming, produzione, ingegneria, simulazioni, moda, turismo. Attraverso i vari portali molte applicazioni sono state rilasciate e la gente comune può tranquillamente usufruirne tramite mobile VR[1], dando l'idea di una tecnologia in forte crescita nel prossimo futuro. Se nel 2018 i guadagni globali della realtà virtuale si sono aggirati sui 2,7 miliardi di dollari, entro il 2020 si stimano a circa 24,3 miliardi. Un'impennata straordinaria accompagnata dalla vendita mondiale di 249 milioni di visori entro il 2025.
Nella maggior parte delle applicazioni disponibili è presente una componente audio ma alcune di esse sono strettamente correlate all'ascolto e alla percezione dei suoni, come una scena di un concerto o di un gioco VR. Un esempio banale ma fortemente attuale è una scena di un ambiente di gioco VR FPS (First Person Shooting). L'utente che si trova in prima persona all'interno della simulazione necessita di una renderizzazione audio di alta qualità, fedele alla reale, tramite la quale possa localizzare le varie sorgenti audio attorno a sé anche fuori dal proprio *field of view* (FOV). Per questo motivo è importante prestare attenzione al rendering audio e alla spazializzazione dei suoni.

[1]GOOGLE Daydream: https://vr.google.com/daydream/
GOOGLE Cardboard: https://vr.google.com/cardboard/
SAMSUNG Gear VR: https://www.samsung.com/it/wearables/gear-vr-r325

1.1 Sistemi per rendering di audio spazializzato

La spazializzazione di una o più sorgenti audio consiste nella simulazione di un paesaggio sonoro tridimensionale, detto anche *soundscape*. All'interno della scena, le sorgenti assumono una posizione virtuale rispetto all'ascoltatore e possono dinamicamente muoversi intorno a lui.
Nel corso degli anni sono stati creati vari sistemi audio che implementassero la spazializzazione audio e le tecniche utilizzate dipendono dal tipo di sistema che si intende utilizzare: il tipo di dispositivi per il playback (altoparlanti, cuffie), così come il loro numero e la disposizione geometrica (sistemi stereo, sistemi surround, ...).
I principali sistemi audio sono:

- **Stereo**: il modo più semplice per implementare il suono spazializzato. Sono presenti due flussi informativi sonori ognuno dei quali destinato ad essere riprodotto da un diverso diffusore acustico posizionato frontalmente all'ascoltare, uno sulla sinistra e uno sulla destra, secondo angoli prestabiliti. Il segnale viene quindi diviso in due, canale sinistro e destro, e ciascuno è indirizzato al relativo altoparlante creando una differenze percezione della sorgente sonora. Se la posizione dell'ascoltatore è corretta frontalmente, l'utente potrà percepire una "sorgente fantasma" in mezzo ai due canali e consente l'ascolto di tutte le possibili combinazioni delle posizioni nell'arco di 180° da sinistra a destra.

- **Multicanale**: l'idea è quella di avere un canale separato per ogni direzione desiderata. L'audio quindi presenta più flussi informativi digitali, ognuno dei quali rappresenta un diverso flusso informativo sonoro. Il termine "multicanale" deriva dal fatto che il singolo flusso informativo è chiamato canale audio. Il sistema è composto da un determinato numero di altoparlanti, i quali vengono posizionati in più direzioni per riprodurre più fedelmente la direzione della sorgente sonora, sia orizzontalmente che verticalmente, tramite le leggi di *panning*[2]. Si viene a creare così l'effetto che più comunemente al giorno d'oggi viene chiamato *surround*. I sistemi home-theater commerciali si basano su questa idea.

- **Cuffie**: Sono piccole e portatili, ogni orecchio può avere un segnale separato grazie ad algoritmi specifici di rendering del suono 3D, denominati sintesi binaurale. Tuttavia le cuffie sono invasive, possono risultare scomode da indossare all'orecchio per periodi prolungati e potrebbero non avere una risposta in frequenza piatta, compromettendo la renderizzazione della spazializzazione

[2]Il Panning è la distribuzione di un segnale sonoro (a coppie monofoniche o stereofoniche) in un nuovo campo sonoro stereo o multicanale determinato da un'impostazione di controllo pan.

del suono. Tendono a fornire l'impressione di avere fonti troppo vicine e non compensano il movimento dell'ascoltatore a meno che non si utilizzi un sistema di tracciamento.

Mettendo a confronto i sistemi audio appena visti possiamo dire che i sistemi basati sull'utilizzo di cuffie presentano due vantaggi principali: primo, eliminano il riverbero dello spazio di ascolto; secondo, e più importante come già detto in precedenza, permettono di fornire segnali distinti ad ogni orecchio, il che semplifica notevolmente la progettazione delle tecniche di rendering audio 3D. Al contrario, i sistemi basati su altoparlanti soffrono di "cross-talk", cioè il suono emesso da un altoparlante sarà sempre udibile da entrambe le orecchie. Se si ignorano gli effetti dell'ambiente di ascolto, le condizioni di ascolto in cuffia possono essere grossolanamente approssimate dagli altoparlanti stereo utilizzando tecniche di cancellazione del cross-talk, che cercano di preelaborare i segnali stereo in modo tale che il suono emesso da un altoparlante venga ridotto o teoricamente cancellato all'orecchio opposto. Con queste tecniche la sorgente fantasma può essere posizionata significativamente al di fuori del segmento compreso tra i due altoparlanti e in particolare si possono produrre effetti di elevation. Il problema principale è che il risultato dipenderà dalla posizione dell'ascoltatore rispetto agli altoparlanti: la cancellazione del cross-talk si ottiene solo in prossimità del cosiddetto "sweet spot", una specifica posizione dell'ascoltatore assunta dal sistema.

1.1.1 Virtual Auditory Display

Il Virtual Auditory Display (VAD) è un sistema per generare suoni spazializzati, ovvero emessi da una certa posizione, e trasmetterli ad un ascoltatore (vedi Fig.1.1).

Un sistema audio dinamico 3D (utilizzato comunemente in sistemi VR), in cui i movimenti dell'utente cambiano l'ambiente sonoro, richiede le seguenti caratteristiche:

- **Head Tracking**: uno dei maggiori problemi per i sistemi audio 3D virtuali è il movimento della testa [11]. Per questo motivo, viene dedicata maggiore attenzione alla localizzazione della testa, che è particolarmente importante nell'ambiente virtuale 3D.

- **Bassa latenza**: idealmente il cervello non dovrebbe rilevare alcun ritardo tra il cambiamento di posizione/orientamento e il cambiamento delle proprietà sonore.

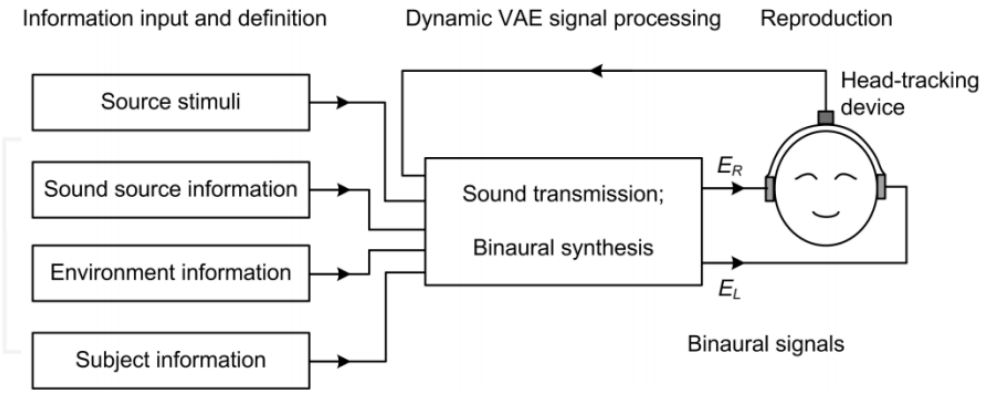


Figura 1.1: Struttura di un tipico sistema VAD dinamico [1].

- **Precisione nella rilevazione degli eventi**: questa caratteristica è ovviamente legata al tipo di evento in corso che apporterà delle modifiche. Per esempio, una traslazione della sorgente sonora dall'alto verso il basso non deve essere riportata come una rotazione intorno ad un altro asse.

Una domanda che sorge quando si considerano i sistemi di rendering di audio 3D è: abbiamo bisogno di personalizzazione? Infatti, ogni essere umano sente il suono in modo diverso. Il nostro sistema uditivo e la sua elaborazione nel cervello si è evoluto sin dalla nostra giovane età secondo la nostra morfologia. La forma della testa, delle orecchie, del busto e delle spalle sono elementi ben noti che alterano la propagazione del suono prima di entrare nel condotto uditivo e infine di colpire il timpano. Pertanto, se si vuole creare una realtà virtuale acustica spaziale immersiva, la personalizzazione del rendering gioca un ruolo importante.

1.2 Obiettivo della tesi

I sistemi di realtà virtuale con stimolazione multimodale e fino a sei gradi di libertà di movimento pongono nuove sfide alla valutazione della qualità audio, mettendo alla prova le modalità di valutazione della qualità audio e rende parzialmente inapplicabili le attuali raccomandazioni in materia di valutazione della qualità.
Basandoci sul lavoro svolto in [12], questa tesi studia gli aspetti percettivi che riguardano la qualità e la spazializzazione del suono in ambienti di realtà virtuale attraverso una serie di attributi prestabiliti. Il metodo si basa sulla classificazione per eliminazione durante l'esplorazione di una realtà virtuale audiovisiva comparando più renderer

binaurali. Inoltre viene approfondito l'impatto della personalizzazione nella qualità dell'esperienza VR [13]. La personalizzazione sarà basata su modelli di selezione per i contorni dell'orecchio e sulle misure antropometriche della testa.
La tesi è così strutturata:

- il **Capitolo 2** presenta tutto il background di questa ricerca. Il capitolo è strutturato in sei parti principali in cui vengono spiegati i concetti generali legati all'audio spazializzato, alle *head-related transfer function*(HRTF), i database di HRTF, al formato SOFA, gli aspetti qualitativi di una sorgente sonora e i sistemi VR;

- il **Capitolo 3** spiega invece le scelte effettuate per i principali componenti utilizzati durante l'esperimento. Viene descritto inoltre un lavoro precedente a tale esperimento relativo alla scelta del plugin audio;

- il **Capitolo 4** descrive l'esperimento effettuato. Si parte inizialmente dai paper su cui ci siamo basati per affrontare tale studio, si spiega poi in cosa consiste l'analisi portata, vengono esposti gli ambienti creati, mostrati nel dettaglio tutti i parametri utilizzati ed infine elencato il protocollo sperimentale;

- il **Capitolo 5** riporta tutti i risultati raccolti durante le prove di sperimentazioni con le relative considerazioni a riguardo;

- il **Capitolo 6** riassume il lavoro e i risultati di questa analisi e conclude la tesi.

2
Background

La realizzazione di un ambiente di realtà virtuale immersiva con una componente audio spazializzata richiede la comprensione dei meccanismi alla base dell'immagine sonora di un ascoltatore in uno spazio e implica la conoscenza, tra l'altro, dell'acustica, della psicoacustica, dell'udito spazializzato e dell'elaborazione digitale del segnale. Il suono spazializzato viene trasmesso in cuffia grazie alla sintesi binaurale o auralizzazione [14], definita così dal Dr. Mendel Kleiner.

2.1 Concetti generali di audio spazializzato

Per una migliore comprensione del lavoro svolto, all'interno di questa sezione vengono presentate le nozioni relative alla percezione spaziale dell'audio. Inizialmente saranno esposti gli attributi percettivi, seguiti poi dalle trasformazioni acustiche ad opera del corpo umano che permetto ad un ascoltatore di percepire il suono. Alla fine verrà spiegato come un soggetto percepisce il suono.

2.1.1 Attributi percettivi

La percezione degli eventi uditivi da parte di un ascoltatore è in relazione ai seguenti gruppi di attributi percettivi:

- **Temporali**: riverbero, ritmo e durata sono degli esempi di questo tipo;

- **Qualitativi**: intensità, timbro e intonazione sono esempi di questo tipo;
- **Spaziali**: percezione spaziale, distanza e direzione sono esempi di questo tipo.

La presenza del corpo dell'ascoltatore è responsabile della creazione di due indicatori di localizzazione binaurali rilevanti chiamati *Interaural Time Difference* (ITD) ed *Interaural Level Difference* (ILD) (vedi la sottosezione 2.1.4) e delle informazioni monoaurali come la colorazione spettrale del suono causata dalle head-related transfer functions. Questi elementi sono racchiusi nelle funzioni di trasferimento (HRTF) relative rispettivamente all'orecchio sinistro e destro. Grazie alle HRTF individuali per un soggetto specifico è possibile renderizzare e simulare un filtro personale per la fruizione dei contenuti sonori tramite cuffie: la sintesi binaurale.
Il sistema VAD, visto in precedenza, utilizza per ciascun segnale una coppia di HRTF basate sulla relazione fra sorgente ed ascoltatore, ed una coppia di headphone transfer functions (HpTFs) specifiche per le proprietà acustiche del dispositivo di riproduzione utilizzato. Per questo motivo generalmente si fa riferimento a più set di HRTF (sinistra e destra) adattate alle differenti sorgenti, ascoltatori e contesto di simulazione.

2.1.2 Trasformazioni acustiche ad opera del corpo umano

- **Testa**: essendo un oggetto rigido situato tra le orecchie, opera come un ostacolo per la propagazione del suono, causando così i due principali indicatori binaurali: ITD, a causa della distanza che deve percorrere il suono dall'orecchio più vicino alla sorgente a quello più lontano, e ILD, in quanto la testa crea un'occlusione ed attenua l'intensità sonora percepita dall'orecchio mascherato;
- **Orecchio esterno**: è suddiviso in due parti chiamate *padiglione auricolare* e *canale uditivo* che si estende fino al timpano (vedi Fig.2.1). Dopo di esso troviamo l'orecchio medio e l'orecchio interno. Il padiglione ha una forma a "bassorilievo" con caratteristiche molto diverse da un individuo all'altro ed è collegato al canale uditivo. Quest'ultimo può essere descritto approssimativamente come un tubo di larghezza costante, con pareti ad alta impedenza acustica. Il comportamento acustico del canale uditivo è facilmente comprensibile: si comporta come un risonatore unidimensionale. D'altra parte il padiglione ha effetti molto più complessi, in quanto agisce fondamentalmente come un'antenna acustica.
- **Torace e spalle**: influenzano le onde sonore incidente sotto due aspetti principali. In primo luogo, forniscono ulteriori riflessioni che si sommano con il suono

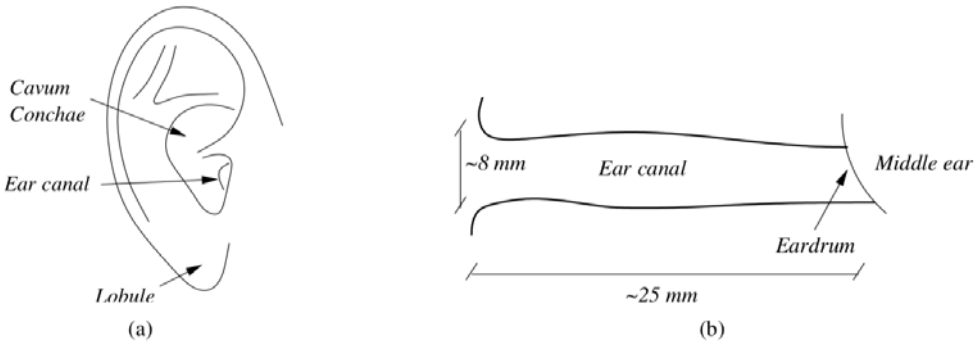


Figura 2.1: Orecchio esterno: (a) padiglione auricolare, (b) canale uditivo.

diretto. In secondo luogo, hanno un effetto occlusivo per i raggi sonori provenienti dal basso. La geometria del torace è piuttosto complicata. Tuttavia una descrizione semplificata può essere derivata considerando un tronco ellissoidale sotto una testa sferica [15] e la Fig.2.2 ne mostra un modello con i principali effetti del tronco ellissoidale sul campo sonoro dell'orecchio. Grazie al tronco ed alle spalle, un orecchio percepisce l'impulso sonoro iniziale, seguito da una serie di impulsi successivi con un ritardo direttamente proporzionale all'elevazione della sorgente.

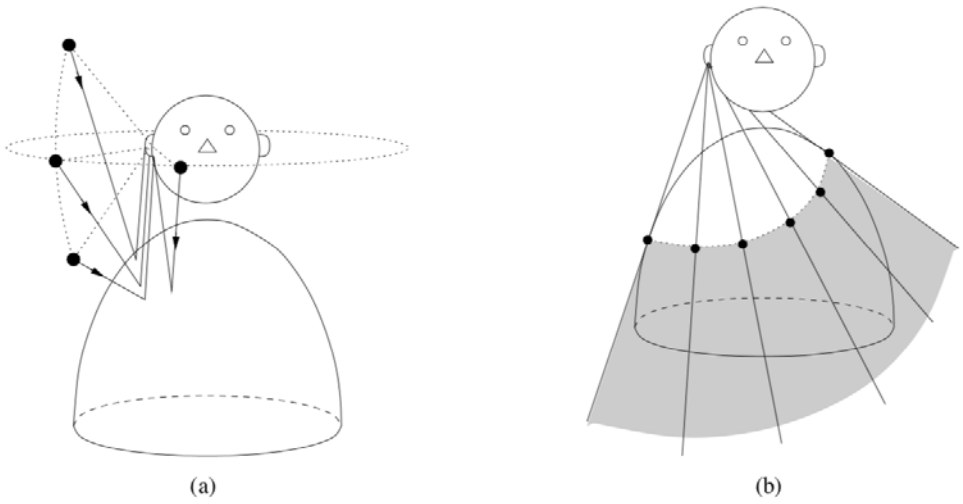


Figura 2.2: Effetti del torace e delle spalle: (a) riflessione, (b) occlusione.

2.1.3 Localizzazione di una sorgente sonora

La localizzazione di una sorgente audio in un ambiente è un processo particolarmente elaborato poiché molti effetti contrastanti e interferenti possono influenzare la percezione uditiva della posizione relativa alla sorgente sonora. In questa sezione viene fornito un breve riassunto.

- **Azimuth**: il posizionamento orizzontale delle orecchie massimizza le differenze per gli eventi sonori che si verificano intorno all'ascoltatore, senza considerare l'asse verticale. Ciò consente l'ascolto delle sorgenti sonore a livello del terreno e al di fuori del campo visivo. ITD e ILD, di cui vediamo un esempio in Fig.2.3 sono considerati i parametri chiave per la percezione dell'azimuth, i.e. sorgente sul piano orizzontale, in quella che a volte viene chiamata *Duplex Theory* della localizzazione. Le persone individuano la direzione orizzontale di un'onda sonora sfruttando l'effetto ITD più facilmente sulle basse frequenze mentre per le alte freqenze risulta essere più difficile. Questo perchè a frequenze $f > 1600Hz$ la lunghezza d'onda del suono è maggiore della distanza fra le orecchie rendendo pertanto impossibile riconoscerne la differenza di fase. Subentra quindi l'effetto ILD che è più marcato per frequenze elevate. Questi due parametri sono combinati per ottenere la percezione azimutale in tutta la gamma delle frequenze udibili.

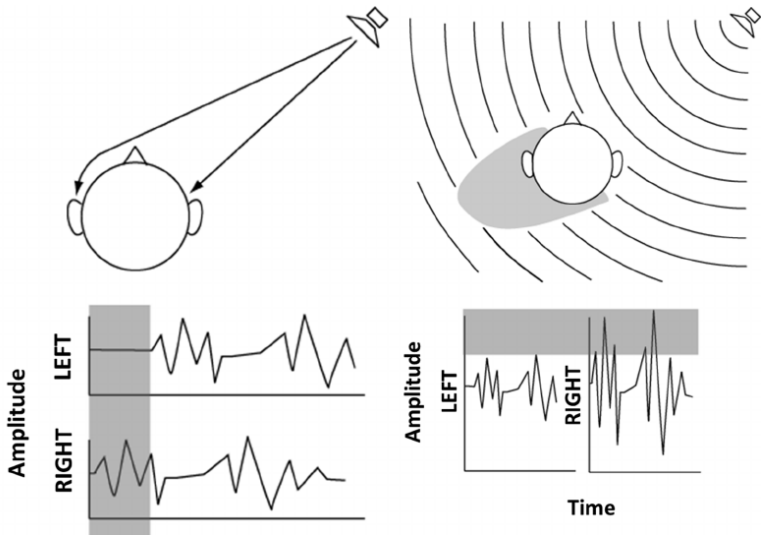


Figura 2.3: Interaural Time Difference (ITD) e Interaural Level Difference (ILD). [2]

- **Elevation**: mentre le indicazioni rilevanti per la localizzazione di una sorgente sonora nel piano orizzontale sono relativamente ben comprese, le cose si

complicano quando si considerano le elevazioni non nulle. La Fig.2.4 mostra che le sorgenti sonore situate ovunque su una superficie conica che si estende dall'orecchio di una testa sferica producono valori identici di ITD e ILD. Queste superfici sono spesso indicate come *coni di confusione*. In queste zone c'è un alta ambiguità percettiva fra suoni provenienti frontalmente e posteriormente in quanto i due valori frontali e posteriori di ITD e ILD sono molto simili. Analizzando i notch nello spettro delle frequenze si può identificare l'elevazione della sorgente sonora. Il padiglione auricolare causa delle riflessioni che causano interferenze distruttive a determinate frequenze, causandone un'attenuazione che si può notare tramite analisi spettrale. Questi notch forniscono informazioni riguardo alla dimensione verticale.

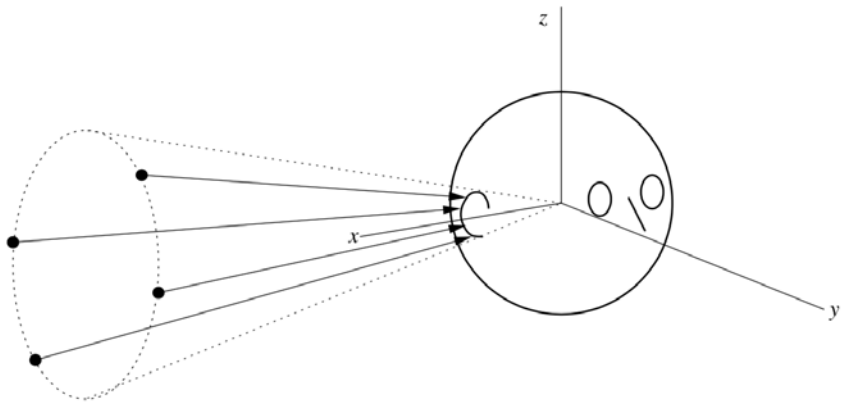


Figura 2.4: Cono di confusione.

- **Distanza**: la stima della distanza è il compito più difficile. La percezione della distanza comporta un processo di integrazione di molteplici indicatori, ognuno dei quali può essere reso inefficace dal risultato sommato di altri potenziali indicatori. In assenza di altre informazioni, l'intensità è l'indicatore primario utilizzato dagli ascoltatori, che imparano dall'esperienza per correlare lo spostamento fisico delle sorgenti sonore con corrispondenti aumenti o riduzioni di intensità. La grandezza di intensità percepita viene chiamata *loudness*. Gli incrementi di loudness (o intensità) possono funzionare efficacemente come indicazioni di distanza solo in assenza di altre informazioni, in particolare il riverbero. Quando il riverbero è presente il volume complessivo all'orecchio dell'ascoltatore non cambia molto per sorgenti molto vicine e molto lontane: la scala relativa alla distanza si applica solo al suono diretto mentre l'energia riflessa rimane approssimativamente costante.
  La stima della distanza con stimoli anecoici è di solito peggiore che negli esperimenti con condizioni di riverbero "ottimali". Molti risultati sperimentali mostra-

no una sottostima complessiva della distanza apparente di una sorgente sonora in un ambiente anecoico, che può essere spiegata dall'assenza di riverbero. Si può dire che il riverbero fornisce la "spazialità" che permette agli ascoltatori di passare da un atteggiamento di ascolto analitico ad un atteggiamento di ascolto quotidiano.
La percezione della distanza è influenzata anche dall'aspettativa o dalla familiarità con la sorgente sonora: se il soggetto sa che di solito quel suono proviene da una certa distanza, probabilmente quando lo riascolterà di nuovo sarà in grado di assegnare una determinata distanza sulla base di esperienze precedenti.

2.1.4 ITD e ILD

Per quasi tutte le posizioni delle sorgenti sonore nello spazio, le onde sonore si propagano e arrivano in ritardo ad un orecchio rispetto all'altro. Questo fenomeno fisico si traduce in una differenza di tempo interaurale (ITD). Questa differenza aumenta sistematicamente per le direzioni laterali, raggiunge il suo massimo e diminuisce nuovamente nella parte posteriore della testa. La posizione laterale a cui corrisponde il valore massimo dipende dalla posizione dell'orecchio, e il ritardo massimo tra le orecchie è di circa 700 μs per la testa di un adulto.
L'ITD è causato non solo dalla distanza tra le due orecchie ma, come detto in precedenza, anche dalla diffrazione e rifrazione della testa [16]. Questi effetti dipendono dalla frequenza, mentre Kuhn [17] spiega e approssima l'ITD in tre diverse gamme di frequenza tra la frequenza più bassa e la più alta udibile: a frequenze inferiori a 2 kHz, la testa è la causa principale degli effetti d'ombra e la causa del ritardo dell'onda che arriva all'orecchio controlaterale. Con l'aumentare della frequenza la diffrazione della testa aumenta per il fatto che la lunghezza d'onda è piccola rispetto alle dimensioni della testa. Di conseguenza, le onde iniziano a diffondersi intorno alla testa. L'influenza di queste onde sull'ITD, utilizzato in particolare per la localizzazione del suono nella gamma di frequenza inferiore a 2,5 kHz [18] [19], è quindi limitata. Alle alte frequenze, la differenza di livello interaurale (ILD) diventa più importante e consente una migliore localizzazione del suono [20].
Oltre all'ITD, anche l'ILD dipende dalla frequenza: è molto ridotto a frequenze inferiori a 2 kHz, il che è dovuto alle grandi lunghezze d'onda rispetto alla testa. Per lunghezze d'onda più corte, l'attenuazione all'orecchio controlaterale è maggiore ed è

influenzata dalle stesse onde. L'ILD è fortemente dipendente dalla direzione e dalla frequenza.
L'ILD può essere determinato in base alla direzione e alla frequenza in funzione delle HRTF dell'orecchio destro e sinistro

$$ILD = 20\ log_{10}\left(\frac{HRTF_L}{HRTF_R}\right) \tag{2.1}$$

La parte interessante risulta essere però la possibile stima dell'ITD partendo dalle grandezze antropometriche della testa. Infatti nel corso degli anni questo tema è stato trattato sotto vari aspetti e sono stati presentati vari modelli per una possibile stima [16]. In [21] viene presentato un modello ottimale di testa sferica stimato utilizzando l'antropometria dei soggetti, basata su semplici e robuste equazioni predittive empiriche. Tale personalizzazione riduce notevolmente gli errori angolari oggettivi, che si verificano quando si utilizza un modello generico. In [22] viene presentato un semplice modello di testa ellissoidale che può tenere conto con precisione della variazione dell'ITD e può essere adattato ai singoli ascoltatori. In [23] viene fatto un confronto sulle previsioni di ITD dei modelli di testa sferica ed ellissoidale con orecchie sfalsate con l'ITD misurato di trentasette soggetti. I parametri del modello vengono prima ottimizzati individualmente e poi viene condotta un'analisi di regressione per predire questi parametri a partire da misure antropometriche.
In [16] inoltre viene fornita una breve panoramica di come l'ILD può essere modellato da un approccio geometrico.

2.2 Head-related transfer function (HRTF)

Abbiamo visto come l'udito viene influenzato dal torso e dell'orecchio esterno sul campo sonoro del timpano. Tutti gli effetti che abbiamo esaminato sono lineari, il che significa che possono essere descritti per mezzo di funzioni di trasferimento e si combinano in modo additivo. Pertanto, la pressione sonora prodotta da una sorgente sonora arbitraria sul timpano è determinata unicamente dalla risposta impulsiva dalla sorgente al timpano. Questa è chiamata *Head-Related Impulse Response (HRIR)* e la sua trasformata di Fourier è chiamata *Head Related Transfer Function (HRTF).*
L'HRTF è una funzione di tre coordinate spaziali e frequenza. Data la forma ap-

prossimativamente sferica della testa, è consuetudine utilizzare le coordinate sferiche rappresentate in Fig.2.5, che utilizzano notazioni e convenzioni leggermente diverse rispetto alle definizioni più tradizionali.

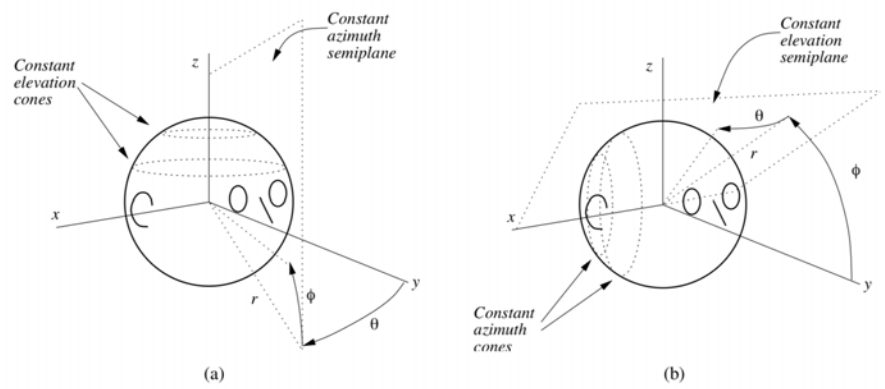


Figura 2.5: Sistemi di coordinate sferiche utilizzati nella definizione di HRTF: (a) sistema di coordinate polari verticali, e (b) sistema di coordinate polari interaurale.

In particolare, in questo contesto, le coordinate angolari verticali e orizzontali vengono definite *azimuth* ed *elevation* e sono indicate rispettivamente come θ e ϕ, mentre la coordinata radiale è denominata *raggio* e indicata come r.
Inoltre, in letteratura vengono utilizzati due diversi sistemi di coordinate sferiche. La Fig.2.5(a) mostra il sistema di coordinate polari verticali: in questo sistema l'azimuth è misurato come l'angolo tra il piano yz e un piano verticale contenente la sorgente e l'asse z, e l'elevation è misurata come l'angolo verso l'alto rispetto al piano xy. Con questa opzione, le superfici di azimuth costante sono piani che attraversano l'asse z, e le superfici di elevation costante sono coni concentrici intorno all'asse z.
In alternativa, a volte, viene usato il cosiddetto sistema di coordinate polari interaurale, mostrato in Fig.2.5(b). In questo caso l'elevation viene misurata come angolo dal piano xy ad un piano contenente la sorgente e l'asse x, e l'azimuth viene poi misurato come angolo dal piano yz. Con questa scelta, le superfici di elevation costante sono piani attraverso l'asse x, e le superfici di azimuth costante sono coni concentrici con l'asse x.
La notazione per indicare una HRTF e‘ la seguente:

$$H^{(L,R)} = (r, \theta, \phi, \omega) \tag{2.2}$$

dove (L) ed (R) identificano l'HRTF relativa all'orecchio sinistro e destro rispettivamente. Nel caso in cui $r \to +\infty$ (che in pratica significa $r > 1$ nella maggior parte delle applicazioni) si può assumere di essere a distanza sufficiente da rendere l'HRTF indipendente da r. Di conseguenza è possibile utilizzare la notazione:

$$H^{(L,R)} = (\theta, \phi, \omega) \tag{2.3}$$

Quindi, formalmente possiamo definire l'HRTF ad un orecchio come il rapporto dipendente dalla frequenza tra il *Sound Pressure Level (SPL)* $\Phi^{(L,R)}(\theta, \phi, \omega)$ con il SPL che verrebbe prodotto in campo aperto nel centro della testa $\Phi_f(\omega)$ come se l'ascoltatore fosse assente

$$H^{(X)}(\theta, \phi, \omega) = \frac{\Phi^{(X)}(\theta, \phi, \omega)}{\Phi_f(\omega)} \tag{2.4}$$

dove X è L o R. La Fig.2.6 mostrano due esempi di HRTF (solo magnitude response): tutti gli effetti esaminati in questa sezione si combinano per formare una funzione di θ e ϕ.

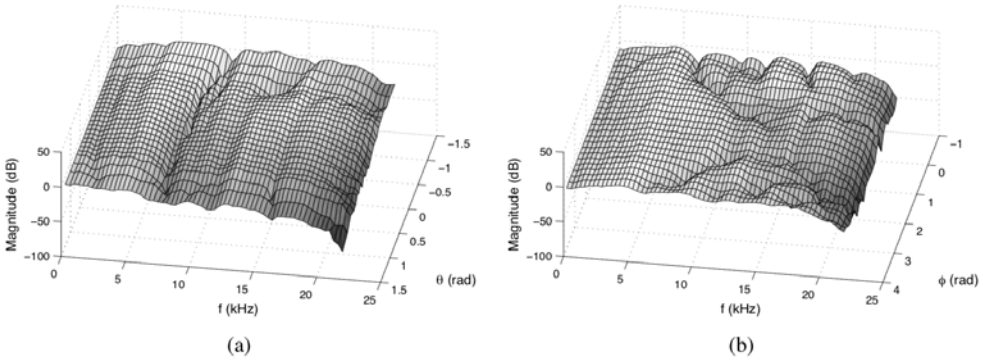


Figura 2.6: Esempio di ampiezza di HRTF (a) nel piano xy ($\theta \in [-\pi/2, \pi/2], \phi = 0$) e (b) nel piano yz ($\theta = 0$, $\phi \in [-\pi/4, \pi]$). Vengono utilizzate coordinate polari interaurali.

2.2.1 Rendering basato su HRTF

L'idea generale nei sistemi audio 3D basati su HRTF è quella di utilizzare HRIR e HRTF misurate. Dato un segnale anecoico e una posizione desiderata della sorgente sonora virtuale (θ, ϕ), i segnali sinistro e destro sono sintetizzati come segue:

1. ritardando il segnale anecoico di una quantità appropriata, al fine di introdurre l'ITD desiderato;

2. convolvere il segnale anecoico con le corrispondenti risposte all'impulso relative alla testa sinistra e destra.

La Fig.2.7 ne riporta uno schema sintetico a blocchi.

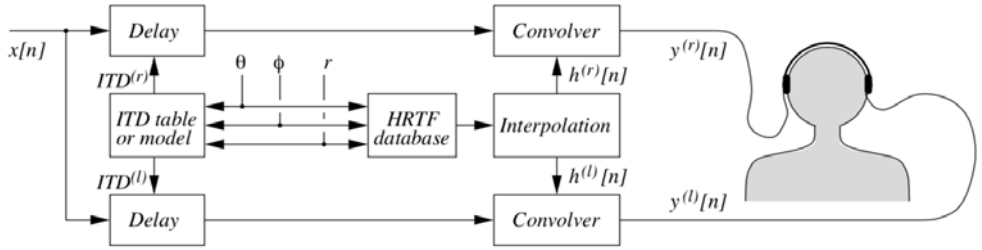


Figura 2.7: Schema a blocchi di un sistema di rendering audio 3D per cuffie basato su HRTF.

La configurazione standard per la misura di una HRTF individuale, come possiamo vedere in Fig.2.8, è formata da una camera anecoica con un certo numero di altoparlanti situati su un supporto sferico. Il raggio della sfera formata dalla composizione degli altoparlanti deve essere superiore ad un metro per evitare gli effetti di prossimità, ad intervalli prestabiliti di azimuth ed elevation. Il soggetto verrà quindi collocato al centro di questa sfera, munito di microfoni nelle orecchie. Le HRIR sono misurate riproducendo degli impulsi sonori e registrando le risposte al segnale su ogni orecchio per ciascuna posizione. In questo modo ogni utente può avere a disposizione il proprio set di HRTF individuale.

Ci sono però alcuni aspetti negativi che riguardano questo processo di acquisizione. Sicuramente non si possono trascurare effetti come i movimenti indesiderati dell'ascoltatore,che invalidano la precisione della misura, ed il limitato numero di posizioni specifiche utilizzate per estrarre i set HRTF.
Uno dei principali limiti della personalizzazione e della commercializzazione dell'audio binaurale all'interno della realtà virtuale è il duro lavoro che sta dietro la creazione delle singole HRTF, che cattura tutti gli effetti fisici creando una percezione personale dell'audio fedele alla condizione di ascolto naturale. Risulta spesso un compito oneroso da eseguire per ogni singolo soggetto dell'esperimento o di un test. Ecco perché spesso si utilizzano metodi alternativi per introdurre delle HRTF appropriate, assegnando ad un soggetto un set di HRTF personalizzate, non create appositamente

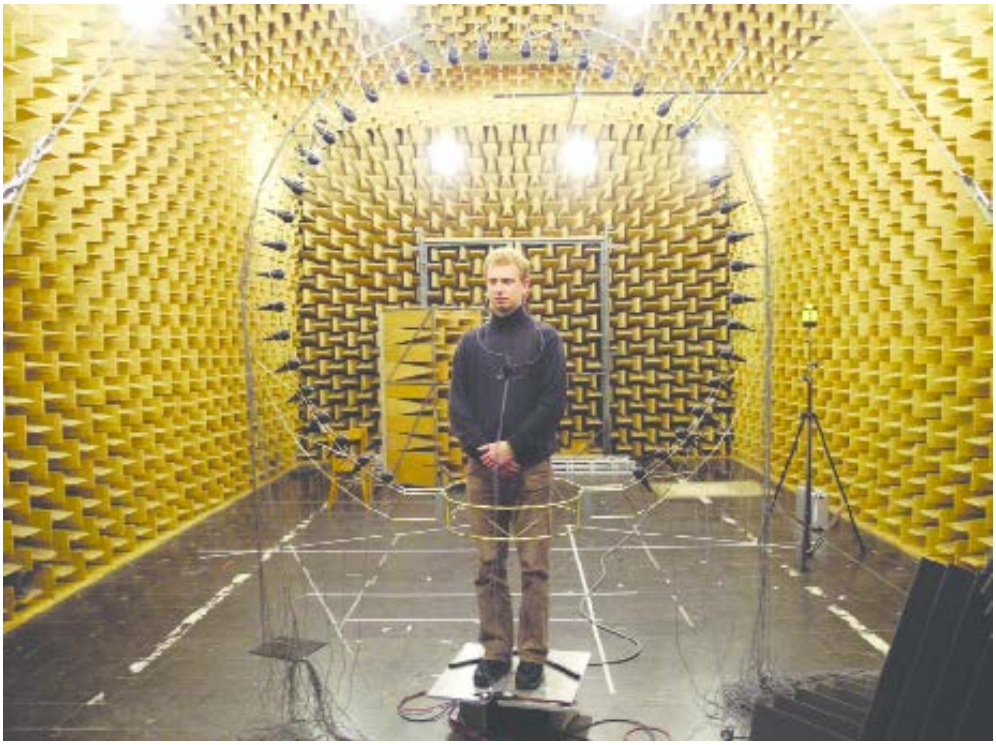


Figura 2.8: Set-up in uso per la misurazione dell'HRTF [3]

sull'individuo stesso ma da una persona che ha messo a disposizione il proprio set in un database: un compromesso tra qualità e costo della misurazione [24].
La scelta di una HRTF generalizzata all'interno di un set di HRTF disponibili in un database è basata sulla corrispondenza tra i nuovi ascoltatori e i soggetti del database già memorizzati. La parte più interessante e importante, è il metodo di come viene assegnato un particolare set di HRTF ad un soggetto invece che ad un altro set. La ricerca scientifica affronta la questione in modi diversi e ci sono diverse alternative più o meno costose che utilizzano diversi strumenti hardware e software. Il vantaggio principale di questo approccio è che l'utente può essere guidato ad una selezione autonoma del proprio miglior set di HRTF senza bisogno di attrezzature speciali e conoscenze approfondite [13].
Esempi di tecniche di selezione sono:

- **Selezione a due passi** [25]: questo metodo si basa su due step conseguenti. Solitamente nel primo passo si seleziona da un pool iniziale di HRTF un sottoinsieme, rimuovendo le peggiori dal punto di vista percettivo. Nel secondo si

sceglie la miglior corrispondenza fra il soggetto esterno ed il database di HRTF rimasto;

- **Tracciamento dei profili del padiglione auricolare** [26] [27]: si tracciano i contorni dell'orecchio e si applica un modello riflessivo in base ai raggi descritti;
- **Corrispondenza tra misurazioni antropometriche** [28] [29]: si basa sul miglior match della HRTF nel dominio antropometrico. Questi parametri corrispondono alla forma dell'orecchio esterno utilizzando le misure disponibili, ad esempio confrontando le proprie dimensioni con le sette distanze presenti nel database CIPIC;
- **Torneo DOMISO** (Determination method of OptimuM Impulse-response by Sound Orientation) [30]: La scelta viene effettuata tramite ripetute prove di ascolto di HRTF diverse ed esclusione delle peggiori, finché non si rimane con la migliore;

2.2.2 Interpolazione

Le misurazioni per l'HRTF possono essere effettuate solo in un insieme finito di posizioni e, quando una sorgente sonora in un punto intermedio deve essere utilizzata, l'HRTF deve essere *interpolata*. Se non viene applicata l'interpolazione (ad esempio, se si utilizza un approccio *nearest*), quando la posizione della sorgente cambia, nello spettro sonoro vengono generati artefatti udibili come click e rumore.
Un modo semplice per eseguire l'interpolazione direttamente sui campioni HRIR è il metodo bilineare [4], che consiste semplicemente nel calcolare la risposta in un dato punto (θ, ϕ) come media ponderata delle risposte misurate associate ai quattro punti più vicini. Più precisamente, se il corrispondente insieme di HRIR è stato misurato su una reticolo sferico (come in Fig.2.9), la stima $\widehat{h}$ dell'HRIR a un punto arbitrario (θ, ϕ)) può essere ottenuto dalla seguente equazione:

$$\widehat{h}[n] = (1 - c_\theta)(1 - c_\phi)h_1[n] + c_\theta(1 - c_\phi)h_2[n] + c_\theta c_\phi h_3[n] + (1 - c_\theta)c_\phi h_4[n] \quad (2.5)$$

dove $h_k[n]$ con $(k = 1, 2, 3, 4)$ sono le HRIR associate ai quattro punti più vicini alla posizione desiderata. I parametri c_θ e c_ϕ possono essere calcolati nel modo seguente:

$$c_\theta = \frac{\theta \bmod \theta_{grid}}{\theta_{grid}} \quad (2.6)$$

$$c_\phi = \frac{\phi \bmod \phi grid}{\phi grid} \tag{2.7}$$

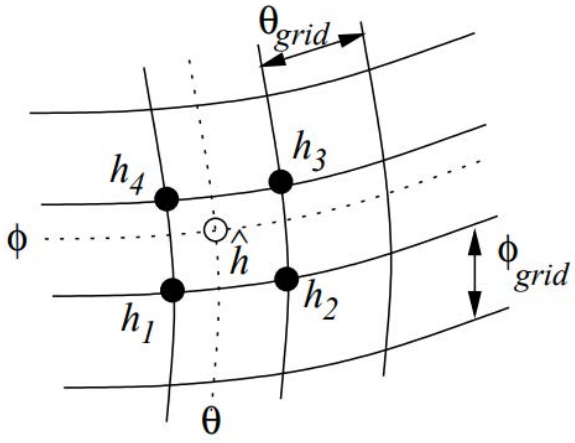


Figura 2.9: Interpretazione grafica dell'interpolazione rettangolare bilineare [4]

Una HRTF interpolata può essere ottenuta anche utilizzando tramite l'utilizzo di soli tre punti di misura [4] [5], che formano un triangolo che racchiude la posizione della sorgente desiderata. Questo approccio può essere esteso andando ad includere la distanza della sorgente attraverso l'interpolazione diretta delle misure HRTF ottenute a varie distanze. L'interpolazione tetraedrica [4] [31] è una sorta di interpolazione tridimensionale, basata sull'individuazione di quattro punti di misura che formano un tetraedro che racchiude la posizione del bersaglio.
La *triangolazione Delaunay* (DT) può essere utilizzata per determinare un insieme di punti in 2D che sono raggruppati in triangoli non sovrapposti. È ottimale che questi triangoli siano quasi equiangolari quando vengono utilizzati per l'interpolazione. DT è l'approccio migliore in questo senso, in quanto massimizza l'angolo minimo dei triangoli generati. DT crea triangoli in modo tale che la circonferenza di ciascuno di essi non contenga altri punti. In 3D, DT risulta in un tetraedro tale che la circonferenza di ogni tetraedro non contenga altri punti.

Come si può vedere da Fig.2.10, X è la posizione dell'HRTF desiderata, mentre A, B, C e D sono posizioni di determinate HRTF misurate da diverse distanze dalla sorgente. Qualsiasi punto desidearato, X, all'interno del tetraedro può essere calcolato

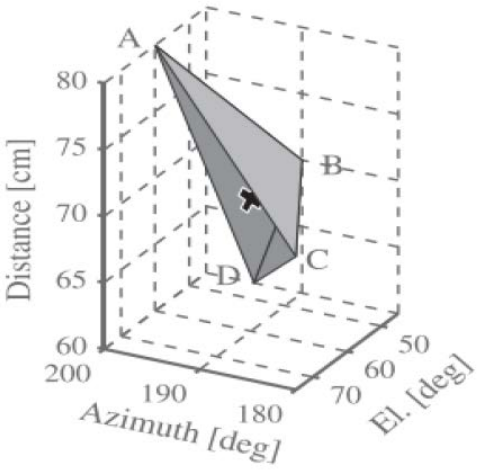


Figura 2.10: Interpretazione grafica dell'interpolazione tetraedrica [5]

come una combinazione lineare dei vertici, come mostrato dall'equazione

$$\mathbf{X} = g_1\mathbf{A} + g_2\mathbf{B} + g_3\mathbf{C} + g_4\mathbf{D} \tag{2.8}$$

dove g_i sono pesi scalari che arrivano fino a uno. I pesi g_i rappresentano le coordinate baricentriche del punto X. Per stimare l'HRTF $\widehat{\mathbf{H}}_x$ desiderata del punto X come la somma ponderata delle HRTF $\mathbf{H}_i$ misurate in A, B, C e D, possiamo usare le coordinate baricentriche come pesi per l'interpolazione, come segue:

$$\widehat{\mathbf{H}}_x = \sum_{i=1}^{4} g_i\mathbf{H}_i \tag{2.9}$$

2.3 Database di HRTF

Diversi gruppi di ricerca hanno misurato diversi dati relativi a varie HRTF e li hanno resi disponibili per altri ricercatori creando dei database. In questa sezione andremo a vedere alcuni tra i più importanti database di HRTF messi a disposizione e le loro caratteristiche principali.

2.3.1 CIPIC

Il database CIPIC [6] è un database pubblico di HRTF ad alta risoluzione spaziale creato al U.C. Davis CIPIC Interface Laboratory nel 2001. Si sono occupati di misurare HRTF ad alta risoluzione spaziale per più di 90 soggetti ed hanno reso disponibile la Release 1.0, un sottoinsieme di pubblico dominio di 45 soggetti (tra cui il manichino KEMAR con pinne grandi e piccole). Escludendo il manichino KEMAR, i 43 soggetti umani (27 uomini e 16 donne) erano studenti dell'U.C. Davis o visitatori del laboratorio. Tutte le HRTF sono state misurate con il soggetto seduto al centro di un anello di $1m$ di raggio il cui asse è stato allineato con l'asse interaurale del soggetto. La posizione della testa del soggetto non era vincolata, ma il soggetto poteva monitorare la posizione della testa. La lunghezza di ogni HRIR è di 200 campioni, corrispondente ad una durata di circa 4,5 ms. I punti nella coordinata di elevation sono stati campionate uniformemente in passi di $360/64 = 5,625°$ partendo da $-45°$ fino a $+230,625°$. Questo porta al campionamento spaziale in 1250 punti, come illustrato in Fig.2.11.

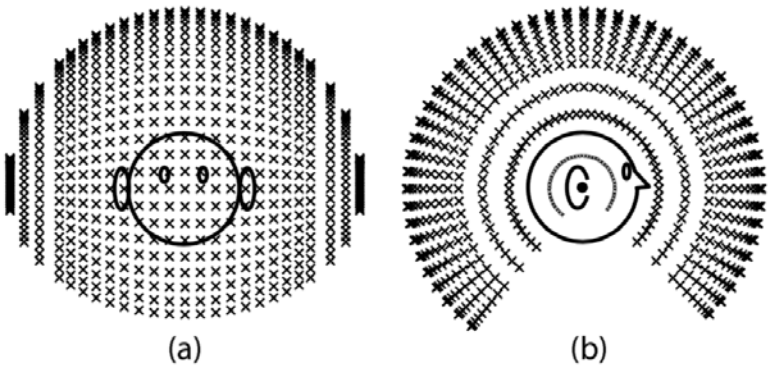
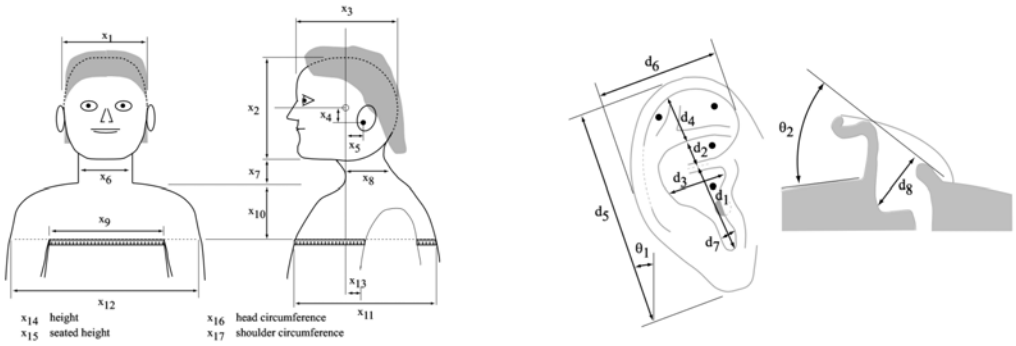


Figura 2.11: Posizioni dei punti rappresentanti le sorgenti sonore (a) frontale (b) laterale nel database CIPIC [6]

Oltre ad includere l'HRIR per 1250 direzioni per ogni orecchio di ogni soggetto, il database include una serie di misure antropometriche (vedi Fig.2.12) che possono essere utilizzate per ulteriori studi. In generale, una particolare misura è stata inclusa perché è stata ritenuta significativa e perché è stato possibile misurarla in modo affidabile e ragionevolmente facile. Inoltre come ulteriori misurazioni, sono state registrate il peso, l'età e il sesso di ogni soggetto.



(a) Misure della testa e del busto

(b) Misure dell'orecchio

Figura 2.12: Misure antropometriche raccolte nel database CIPIC [6]

I ricercatori del U.C. Davis CIPIC Interface Laboratory inoltre hanno messo a disposizione una ricca documentazione tecnica aggiuntiva e alcuni programmi utili usufruibili tramite MATLAB per la verifica dei dati.

2.3.2 MIT KEMAR database

Il database MIT KEMAR [7] è un database pubblico di HRTF creato dal MIT Media Lab of Perceptual Computing. Le misure consistono nella risposta impulsiva dell'orecchio sinistro e destro di un altoparlante montato a 1,4 metri dal KEMAR (Knowles Electronic Manikin for Acustic Research). Questo database infatti non si basa su misure effettuate su soggetti umani ma su una testa artificiale di un manichino, come possiamo vedere in Fig.2.13.
Il KEMAR è uno strumento di ricerca acustica che permette misurazioni riproducibili per stabilire le prestazioni degli apparecchi acustici e di altri apparecchi elettroacustici, nonché la qualità delle registrazioni binaurali. Questo simulatore di testa e busto (HATS) si basa sulle dimensioni medie a livello mondiale della testa e del busto umano maschile e femminile. In particolare in Fig.2.13b vengono riportate le dimensioni relative al KEMAR in generale.
Le misurazioni sono state effettuate nella camera anecoica del MIT. Il KEMAR è stato montato in posizione verticale su un supporto girevole motorizzato che poteva essere ruotato con precisione su qualsiasi azimuth tramite controllo computerizzato. Il diffusore è stato montato su un supporto a braccio che ha permesso un posizionamento preciso del diffusore a qualsiasi elevation rispetto al KEMAR. Così, le misurazioni

sono state effettuate per ciascuna elevation alla volta, impostando l'altoparlante nella posizione corretta e ruotando il KEMAR per ogni azimuth. Lo spazio sferico intorno al KEMAR è stato campionato per misure di elevation che vanno da $-40°$ a $+90°$. Per ogni livello di elevation, le posizioni di azimuth ricoprono un angolo intero $360°$ con incrementi circa di $5°$. La tabella Tab.2.1 mostra il numero di campioni e l'incremento azimuthale ad ogni elevation.

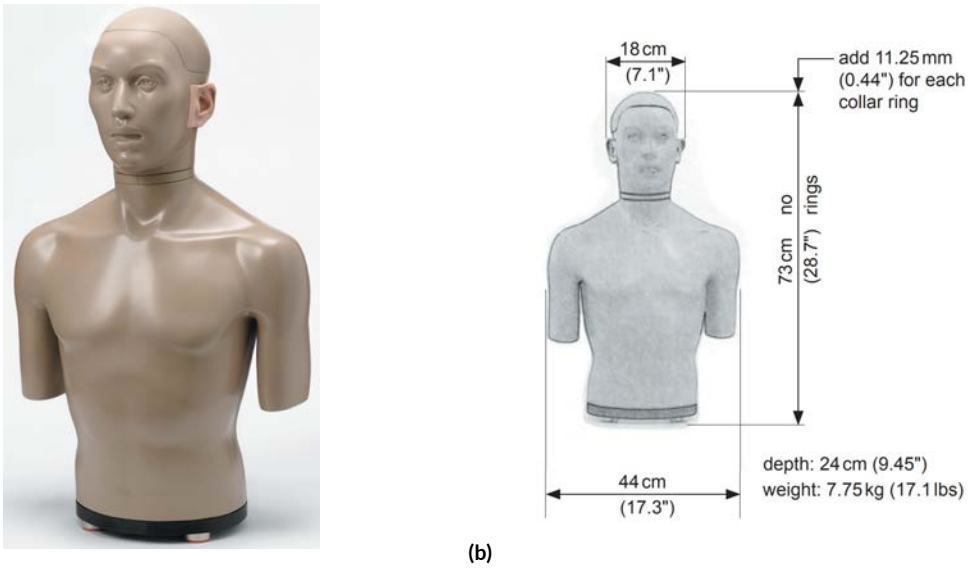


Figura 2.13: Manichino KEMAR tipo 45BA (a) aspetto reale (b) dimensioni [7]

In totale, sono state campionate 710 posizioni. Se il KEMAR fosse perfettamente simmetrico e i suoi microfoni auricolari fossero identici, sarebbe bastato campionare solo l'emisfero sinistro o destro intorno al KEMAR. Tuttavia nel corso delle misurazioni, il KEMAR montava due padiglioni differenti (esistono diversi modelli di pinna omologati per il KEMAR), in questo caso il padiglione sinistro era "normal", il padiglione destro era il modello "large red" e, di conseguenza, le risposte non erano identiche. Questo però in realtà rappresenta un bonus, perché campionando l'intera sfera sono stati ottenuti due set completi di HRTF simmetriche.

Elevation	Number of Measurements	Azimuth Increment
-40	56	6.43
-30	60	6.00
-20	72	5.00
-10	72	5.00
0	72	5.00
10	72	5.00
20	72	5.00
30	60	6.00
40	56	6.43
50	45	8.00
60	36	10.00
70	24	15.00
80	12	30.00
90	1	x

Tabella 2.1: Numero di misure e incremento azimuthale ad ogni elevation [7]

2.4 Il formato SOFA

Le HRTF vengono misurate da un certo numero di laboratori e sono tipicamente memorizzate nel formato di file nativo di ciascun laboratorio. Sebbene i diversi formati siano vantaggiosi per ogni laboratorio, lo scambio di tali dati è difficile a causa delle incompatibilità tra i formati. Nel corso degli anni si è cercato quindi di produrre un formato che potesse essere universale, dando vita così al formato SOFA [32].
Il formato **SOFA** (*Spatially Oriented Format for Acoustics*) ha lo scopo di rappresentare i dati spaziali in modo generale, consentendo di memorizzare non solo HRTF ma anche dati più complessi, come risposte impulsive direzionali all'interno di una stanza (DRIRs) misurate con un array di microfoni multicanale eccitati da un array di diffusori e risposte acustiche di cuffie [33].
Durante la progettazione del SOFA, sono stati stabiliti i seguenti requisiti:

- descrizione del setup di misurazione con geometria arbitraria, cioè non limitata a casi speciali come una griglia regolare o una distanza costante;

- dati autodescrittivi con una definizione coerente, cioè, tutte le informazioni richieste sul setup di misura devono essere fornite come metadati nel file;
- flessibilità per descrivere i dati di condizioni multiple (ascoltatori, distanze, ecc.) in un unico file;
- supporto parziale per file e rete;
- disponibile come file binario con compressione dei dati per un'efficiente archiviazione e trasferimento;
- convenzioni di descrizione predefinite per le configurazioni di misura più comuni.

Le specifiche SOFA mirano a soddisfare tutti questi requisiti. In poche parole, la configurazione di misura è descritta da vari oggetti e dalle loro relazioni. Le informazioni sono memorizzate in un contenitore numerico basato su *netCDF-4*. Una misura viene considerata come un'osservazione campionata discreta effettuata in un momento specifico e in una condizione specifica. Ad ogni misura vengono quindi raccolti e fatti corrispondere dei dati (ad esempio, una risposta impulsiva, IR) ed è descritta dalle sue dimensioni e metadati corrispondenti. Tutte le misure sono memorizzate in un'unica struttura di dati (ad esempio, una matrice di IR). Una descrizione coerente delle configurazioni di misura è data dalle convenzioni SOFA.
Nel nostro caso, utilizzando come esempio il file SOFA che descrive il soggetto 003 del database CIPIC, possiamo vedere in Fig.2.14 i campi che descrivono le HRTF e l'intera configurazione di misurazione.
I campi di maggior interesse per il nostro esperimento sono stati:

- *Data.IR*: contiene per ogni posizione (combinazione azimuth ed elevation) i 200 samples della IR per entrambi i canali Left e Right;
- M: numero di misurazioni (1250);
- N: numero di campioni che descrivono una misurazione (200);
- R: numero di ricevitori, ovvero il numero di canali (2);
- *SourcePosition*: definisce per ogni misurazione le posizioni di azimuth ed elevation relative ad un certo raggio r costante.

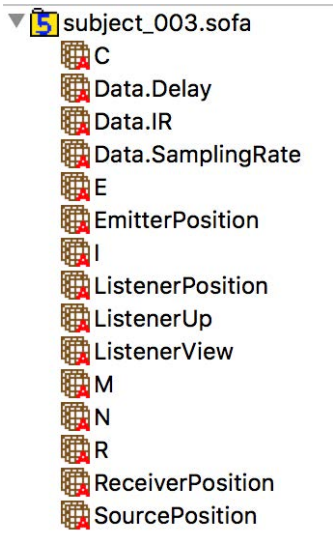


Figura 2.14: Campi all'interno di un file SOFA (ispezione tramite il software HFDView 3.0)

Si possono osservare altri campi interessanti che descrivono il setup di misurazione come il *Data.SamplingRate*, che descrive la frequenza di campionamento, l'*EmitterPosition*, che descrive la posizione del diffusore, i 3 campi *ListenerPoistion*, *ListenerUp*, *ListenerView*, che descrivono come è posizionato il soggetto ascoltatore all'interno del setup col relativo sistema di riferimento.

2.5 Attributi qualitativi per definire un'esperienza di ascolto

I sistemi audio spazializzati sono stati spesso valutati in relazione a singole qualità uditive come, ad esempio, l'accuratezza della localizzazione, la percezione della colorazione o della distanza. Il vantaggio dell'utilizzo di HRTF è ben documentato per quanto riguarda il miglioramento della precisione della localizzazione [26] [34]. Tuttavia, con l'aumento dell'uso dell'audio binaurale in scenari audiovisivi più complessi, studi cognitivi e simulazioni di realtà virtuale e aumentata, l'impatto percettivo della selezione HRTF può andare oltre la semplice localizzazione.
Definiamo innanzitutto cosa si intende per "Quality of Experience" (QoE, ovvero qualità dell'esperienza). "*QoE è il grado di piacere o disturbo di una persona la cui esperienza coinvolge un'applicazione, un servizio o un sistema. Risulta dalla valuta-*

zione della persona del soddisfacimento delle sue aspettative e dei suoi bisogni rispetto all'utilità e/o al godimento alla luce del contesto, della personalità e dello stato attuale della persona [35]".
Diversi vocabolari sensoriali descrittivi sono stati sviluppati per vari campi di interesse legati all'audio, come ad esempio acustica ambientale, altoparlanti, sistemi di registrazione multicanale e sistemi di riproduzione, nonché per sistemi di correzione ambientale, codec audio, algoritmi per la spazializzazione delle cuffie e anche per ambienti acustici virtuali (VAE) più complessi. Alcuni studi precedenti hanno spesso applicato vocabolari che sono stati inizialmente generati ad hoc dall'esperienza/conoscenza degli autori e poi ridotti applicando l'analisi dei fattori alle valutazioni degli ascoltatori di un insieme di stimoli d'esempio.
Nel corso degli anni sono stati svolti diversi studi che permettessero di stabilire un insieme di attributi percettivi direttamente collegati alle variazioni HRTF, basati su un approccio consensuale, che non è limitato dalla natura individuale della percezione audio binaurale. Gli attributi per i quali esiste un consenso globale sul valore (ad esempio, stimolo A è più forte di B) non sono considerati pertinenti all'insieme di attributi rilevanti per gli effetti HRTF. Quando si valuta la qualità di un rendering binaurale utilizzando un set HRTF non individuale, la discrepanza tra l'HRTF dell'ascoltatore e l'HRTF utilizzata per il rendering binaurale avrà un impatto di degrado. Questo degrado varierà da individuo a individuo, il che significa che non c'è modo di stabilire una percezione globale di consenso per un dato stimolo.
Nelle prossime sottosezioni vengono presentati degli esempi di studi che hanno contribuito alla creazione di un dizionario di aggettivi percettivi per la descrizione qualitativa di una scena audio. Il dizionario è stato progettato per l'ascolto binaurale e può essere utilizzato quindi in studi futuri per indagare le ragioni soggettive che portano a far sì che una persona possa preferire un set di HRTF ad un altro.

2.5.1 SAQI

Per superare le limitazioni rispetto alla rilevanza e alla completezza di altri vocabolari creati come detto in precedenza, Lindau *et al.* [8] hanno sviluppato *Spatial Audio Quality Inventory* (SAQI) per la valutazione percettiva di tutte le tecnologie audio spaziali utilizzate per la sintesi di ambienti acustici. Si tratta di un vocabolario di consenso che comprende 48 descrittori verbali delle qualità auditive che si presume

abbiano rilevanza pratica quando si confrontano i campi sonori (ri)sintetizzati con riferimenti reali o immaginari o tra loro. I 48 descrittori possono essere grossolanamente suddivisi in otto categorie (timbro, tonalità, geometria, stanza, comportamento temporale, dinamica, artefatti e impressioni generali) e devono essere considerati come descrittori di "differenze percepite rispetto a descrittore di riferimento".
Il vocabolario è stato prodotto da un Focus Group di 21 esperti di lingua tedesca per l'acustica virtuale. Altri cinque esperti hanno contribuito a verificare l'univocità di tutti i descrittori e le relative spiegazioni. Inoltre, la traduzione in inglese è stata seguita e verificata da otto esperti bilingue.
La tabella "Table I", presente in [8] a pagina 5 e seguenti, riporta tutti e 48 i descrittori prescelti. Ogni descrittore è completato da una breve descrizione scritta che chiarisce la circoscrizione e da un'etichetta in scala dicotomica, uni/bipolare, rispettivamente. Per la gestione di aspetti eventualmente trascurati o emergenti, è stata inserita nel vocabolario una categoria aperta (*'Others'*), da nominare in base agli argomenti dei test di ascolto. Sono state quindi definite cinque entità di valutazione di base che forniscono un'ontologia di tipo ideale per una scena acustica virtuale: *sorgenti in primo piano*, *sorgenti di sfondo*, l'*ambiente acustico della stanza simulata*, il *sistema di riproduzione* stesso e l'*ambiente di laboratorio*. In combinazione, queste cinque entità sono pensate per incorporare tutti i possibili oggetti di interesse, riportati nella Tab.2.2. Infine, tutte le differenze percepite possono essere definite più da vicino rispetto alla loro causa, cioè se dipendono dall'interazione dell'utente, dagli eventi di scena o da nessuno di essi.

All audible events				
Intended audible events (elements of the presented virtual scene)			Unintended audible events	
Foreground sources	Background sources	Room acoustic environment	Reproduction system	Laboratory environment

Tabella 2.2: Eventi o oggetti a cui poter indirizzare una differenza percepita. [8]

Un test SAQI inizia con il confronto uditivo tra uno stimolo di prova e un riferimento dato o immaginato. Poi al soggetto dovrebbe essere chiesto prima di tutto se ha percepito qualche differenza, perché, se il soggetto nega, il test potrebbe essere fermato a questo punto. Altrimenti, la differenza complessiva percepita può essere valutata utilizzando una scala di intensità unipolare. Dopo la valutazione, il ricercatore può opzionalmente chiedere al soggetto di indicare il comportamento temporale,

le dipendenze legate all'utente o alla scena e/o di assegnare oggetti di riferimento alla differenza percepita, il che può essere fatto utilizzando domande a scelta multipla. Queste opzioni possono essere selezionate in relazione all'interesse di ricerca o rispetto agli stimoli utilizzati. La procedura viene ripetuta per tutti gli attributi selezionati contenuti nel SAQI, potenzialmente in ordine di presentazione randomizzato, mentre gli stimoli del test sono accessibili per un confronto continuo. Infine, i soggetti dovrebbero essere invitati a specificare e valutare altre differenze che sono state potenzialmente trascurate.

2.5.2 Vocabolario formulato da Simon et al.

Simon *et al.* [9] hanno sviluppato una lista di attributi che qualificano le differenze percepite tra le HRTF, fornendo una comprensione qualitativa della varianza percettiva delle rappresentazioni binaurali non individuali. L'elenco degli attributi è stato progettato utilizzando un metodo di consultazione del *consensus vocabulary protocol* (CVP). I partecipanti hanno seguito una procedura del CVP, descrivendo le differenze percepite tra stimoli binaurali basati su estratti binauralizzati di produzioni multicanale. A ciò ha fatto seguito una riduzione lessicale automatica e una serie di riunioni del gruppo di consenso durante le quali i partecipanti hanno concordato un elenco di attributi pertinenti. Infine, l'elenco di attributi proposto è stato valutato attraverso un test di ascolto, che ha portato a otto validi attributi percettivi per descrivere le dimensioni percettive interessate dalle variazioni di set HRTF.
Gli attributi finali ottenuti devono osservare le seguenti caratteristiche:

- essere oggettivi;
- avere poca interferenza tra di loro;
- consentire la discriminazione tra gli stimoli;
- essere singolari rispetto alla combinazione di più termini;
- non essere una combinazione di sotto-attributi;
- essere precisi, ben definiti e non ambigui;
- generare consenso tra i partecipanti;
- devono riferirsi alla realtà;

- non usare il gergo;

- essere specificabile da un riferimento.

Attribute	End points	Definition
Coloration	More high frequency content More low frequency content	Feeling of a sound richer in high/medium/low frequencies
Elevation	More toward the top More toward the bottom	*Self-explanatory*
Externalization	Inside the head Outside the head	Perception of sounds located outside the head
Immersion	Immersive Non-immersive	Feeling of being located in the middle of the audio scene
Position-front/back	Front Back	*Self-explanatory*
Position-lateral	More toward the left More toward the right	*Self-explanatory*
Realism	Realistic Non-realistic	Sounds seem to come from real sources located around you
Relief	Compact Spread out	Distance between the closest sound objects and the farthest

Tabella 2.3: Elenco degli attributi, delle definizioni e dei punti finali convalidati (traduzione in inglese). [9]

Il vocabolario è stato formulato da u gruppo di 17 partecipanti (13 uomini, 4 donne) che comprendeva 7 ingegneri del suono professionisti, 9 studenti in ingegneria del suono e 1 ricercatore in suono binaurale, di età compresa tra 20-52 anni. L'elenco degli attributi finali è composto da 8 aggettivi, come possiamo vedere in Tab.2.3 con una breve descrizione. Alcuni attributi trovano corrispondenza ad attributi spaziali e generali del dizionario SAQI: l'*immersione* è simile alla *presenza* di SAQI. Tuttavia, un gran numero di attributi trovati in SAQI non sono stati convalidati nello studio di Simon *et al.* È il caso degli attributi della stanza, degli attributi temporali, della dinamica e degli artefatti. Infatti, questo studio si è concentrato esclusivamente sui confronti HRTF e non su altri effetti di trasformazione o degradazione. Questo elenco è rivolto ad esperti del suono e ingegneri del suono, con un uso preciso della definizione, e potrebbe non essere adatto per valutazioni da parte di partecipanti non pratici del campo da esaminare.

2.6 Sistemi di realtà virtuale

Il termine realtà virtuale ha assunto molte declinazioni, passando da Sensorama[1] fino ad arrivare agli odierni dispositivi mobile, e la differenza l'ha fatta spesso la tecnologia con cui vi si è avuto accesso. La tecnologia è infatti l'elemento in grado di effettuare il balzo spazio temporale tra la dimensione reale e quella virtuale, dove tutto cambia: scenario, cognizione di sé stesso, visuale, percezione del suono, interazione con l'ambiente. La grande novità odierna sta nel fatto che la tecnologia offre potenzialità nuove ed estremamente innovative, consentendo alla realtà virtuale di effettuare continui ed esponenziali balzi in avanti, come vediamo in Fig.2.15.

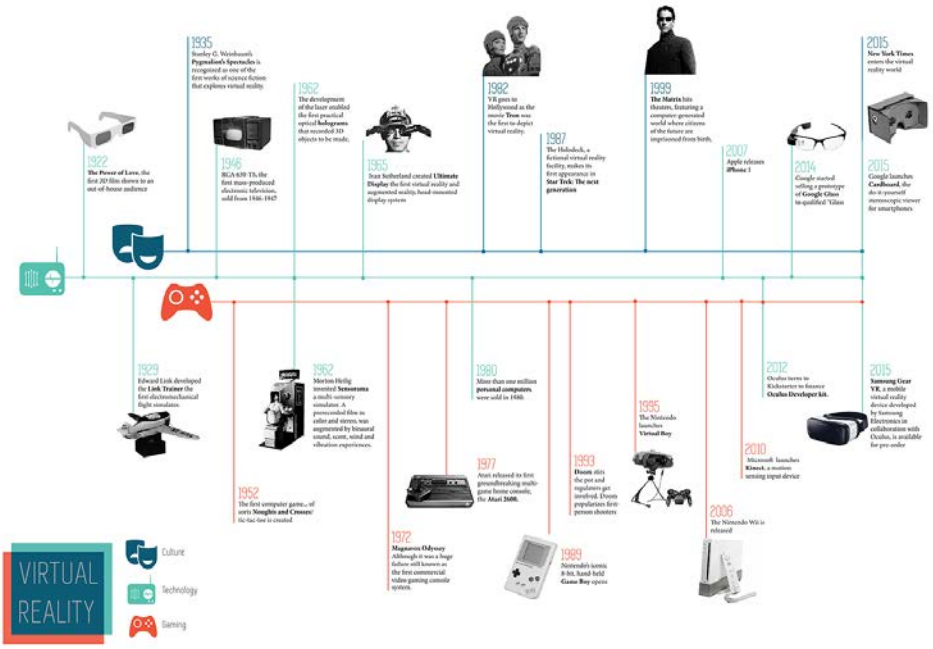


Figura 2.15: Sviluppo della tecnologia VR negli anni

Per una perfetta immersione nella realtà virtuale servono tre componenti base: un visore integrato con un display che avvolge la vista e un sistema audio, un computer (una console o uno smartphone) e un controller, per interagire con questa nuova dimensione. Per la creazione di un ambiente di realtà virtuale totalmente nuovo,

[1]La macchina Sensorama, primo vero dispositivo per la realtà virtuale, creato alla fine degli anni '50.

quindi, si devono compiere delle scelte riguardo quali dispositivi e software utilizzare per i seguenti componenti: visore, motore grafico e motore audio.

2.6.1 Visore VR

Quello dei visori per la realtà virtuale è un mercato vasto, in cui ogni prodotto fa storia a sé. Il mercato della realtà aumentata e realtà virtuale, negli ultimi anni in costante crescita, registrerà un vero e proprio balzo da qui al 2022, fino a toccare i 209,2 miliardi di dollari. Lo evidenzia il grafico elaborato da Statista sulla base di una ricerca di IDC (International Data Corporation).

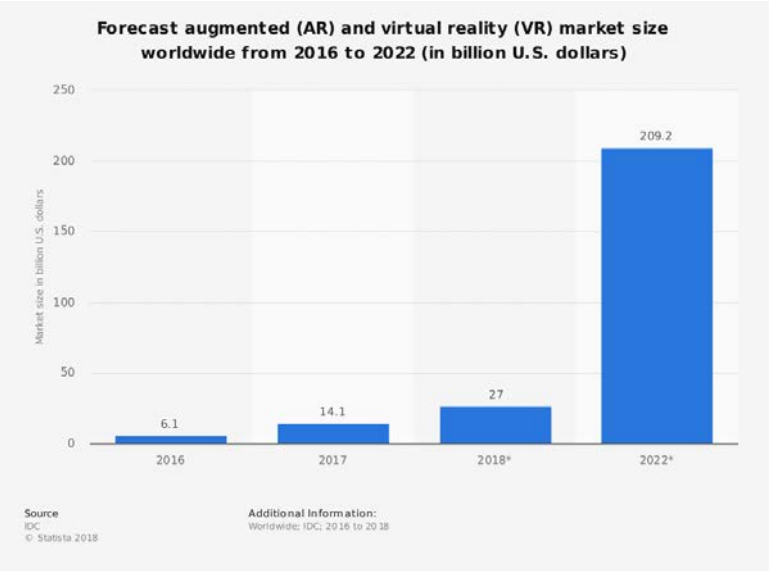


Figura 2.16: Il trend di mercato dal 2016 al 2022

Diverse funzionalità e diverse metodologie di utilizzo, rendono infatti ciascun prodotto in parte unico. Alcuni si utilizzano con lo smartphone, altri con il PC, altri ancora con piattaforme di gioco, ma non esiste attualmente un prodotto universale che permetta di soddisfare diverse esigenze. Attualmente i dispositivi di realtà virtuale si stanno diffondendo notevolmente, con la recente presentazione di molte case videoludiche come Oculus Go[2], Oculus Quest[3] e Xiaomi Mi VR[4].

[2]https://www.oculus.com/go/?locale=it_IT
[3]https://www.oculus.com/quest/?locale=it_IT
[4]https://www.mi.com/global/mivr1c

Oculus Rift

Oculus Rift[5] è una linea di visori per la realtà virtuale indossabile sul viso (in inglese HMD, *head-mounted display*) sviluppati e prodotti da Oculus VR, una divisione di Facebook Inc. Lo schermo LCD ad alta risoluzione da 7 pollici ha una profondità di colore di 24 bit per pixel ed è abilitato alla stereoscopia 3D e il campo visivo è di oltre 90 gradi in orizzontale (110 gradi di diagonale). Inoltre è munito di una serie di controller per l'interazione, di apposite cuffie audio e sensori spaziali.
Per la parte software mette a disposizione vari strumenti. Il runtime di Oculus Rift supporta ufficialmente Microsoft Windows, macOS e GNU/Linux. Il pacchetto di installazione include componenti come il driver per le cuffie (che include il driver del display Oculus e i driver del controller), il driver del sensore di Position Tracking, Oculus Service e Oculus Home Application. Il servizio runtime implementa una serie di tecniche di elaborazione volte a ridurre al minimo la latenza e in aggiunta a migliorare la scorrevolezza delle applicazioni VR. Le applicazioni per il sistema Rift sono sviluppate utilizzando l'Oculus PC SDK, un SDK proprietario gratuito disponibile per Microsoft Windows. Inoltre, è direttamente integrata con i popolari motori di gioco come Unity, Unreal Engine 4 e Cryengine. Questo permette agli sviluppatori che già conoscono questi motori di creare contenuti VR con poco o nessun codice specifico VR.

Samsung Gear VR

Il Samsung Gear VR è un HMD per la realtà virtuale sviluppato da Samsung Electronics, in collaborazione con Oculus VR, e prodotto da Samsung. Questo tipo di dispositivo si colloca sul settore del mobile VR, ovvero dove uno smartphone compatibile funge da display e processore delle cuffie, mentre l'unità Gear VR funge da controller, che contiene il campo visivo di 100 gradi. È possibile collegare un controller esterno tramite connessione bluetooth. I contenuti ovviamente possono essere sviluppati per ambiente Android o direttamente per sistema Oculus tramite la propria SDK.

[5]https://www.oculus.com/rift

HTC Vive

HTC Vive[6] è un HMD per la realtà virtuale sviluppato da HTC e Valve Corporation. Utilizza la tecnologia di tracciamento "room scale", che consente all'utente di muoversi nello spazio 3D e di utilizzare vari controller con motion-tracked per interagire con l'ambiente. Il dispositivo utilizza due pannelli OLED ad alta risoluzione e un campo visivo pari a 110 gradi. Il runtime di HTC Vive supporta ufficialmente Microsoft Windows, macOS e GNU/Linux ed usa l'SDK di Valve, ovvero Steam VR e OpenVR, che viene supportata dai maggiori motori grafici.

2.6.2 Motore grafico

Un motore grafico (in inglese "game engine") è un ambiente di sviluppo software progettato con lo scopo di costruire ambienti virtuali. La funzionalità di base tipicamente fornita da un motore di gioco include un motore di rendering ("renderer") per la grafica 2D o 3D, un motore fisico o di rilevamento delle collisioni (e la risposta alle collisioni), audio, scripting, animazione, intelligenza artificiale, networking, streaming, gestione della memoria, threading, supporto alla localizzazione, grafico di scena e può includere il supporto video per la cinematica. I motori di gioco vengono messi a disposizione dei game designer per codificare e pianificare un gioco in modo rapido e semplice, senza doverne costruire uno da zero. Che siano basati su 2D o 3D, offrono strumenti per aiutare nella creazione e nel posizionamento degli asset.

Unity Engine

Unity[7] è un motore di gioco multipiattaforma sviluppato da Unity Technologies. Il motore può essere utilizzato per creare giochi tridimensionali, bidimensionali, di realtà virtuale e di realtà aumentata, così come simulazioni ed altre esperienze. Il motore è stato adottato da industrie esterne ai videogiochi, come il cinema, industria dell'automobile, l'architettura, l'ingegneria e l'edilizia. Il motore offre un'API (*Application Programming Interface*) di scripting primario in C#, sia per l'editor Unity sotto forma di plugin, sia per i giochi stessi. L'editor Unity è supportato su Windows e macOS, con una versione dell'editor disponibile per la piattaforma Linux, anche se in fase sperimentale. Unity ha opzioni di licenza gratuita e a pagamento. La licenza

[6] https://www.vive.com/us/product/vive-pro/
[7] https://unity.com/

gratuita è per uso personale o per aziende generalmente piccole. I creatori possono sviluppare e condividere le risorse generate dagli utenti ad altri creatori di giochi attraverso l'Unity Asset Store. Ciò include risorse e ambienti 3D o 2D.

Unreal Engine

L'Unreal Engine[8] è un motore grafico sviluppato da Epic Games. La prima versione è stata realizzata per lo sparatutto in prima persona Unreal, pubblicato nel 1998 per Microsoft Windows, Linux e Mac OS. Nel corso degli anni lo sviluppo è continuato, adattando il software alle potenzialità degli hardware disponibili e portandolo ad altre piattaforme. Attualmente è disponibile la quarta generazione del motore annunciata già nel 2005 ed è disponibile gratuitamente dal 2015.

2.6.3 Motore audio

Il motore audio (in inglese "audio engine") è il componente che consiste in algoritmi relativi al caricamento, alla modifica e all'emissione del suono attraverso il sistema di diffusori predisposto. Una parte fondamentale di questo lavoro di tesi è stato scegliere il motore audio da utilizzare negli ambienti da sviluppare, che potesse permettere il più fedelmente possibile una simulazione reale del suono.

Google Resonance Audio

Google Resonance Audio [36] è un motore audio multipiattaforma sviluppato da Google e recentemente reso open source, mettendo a disposizione una SDK per sviluppi su mobile e desktop. Questo motore audio prevede la possibilità di utilizzare l'audio spazializzato basato su Ambisonic, ammettendo i formati Ambix ACN/SN3D. Sono incluse anche tecniche di codifica, manipolazione del campo sonoro e tecniche di decodifica, come le HRTF. Nello specifico Resonance Audio utilizza HRIR personalizzate derivate dal database SADIE [37]. Resonance Audio va oltre la spazializzazione 3D di base, fornendo strumenti per la modellazione accurata di ambienti sonori complessi, infatti la SDK permette:

- personalizzazione della direzionalità della sorgente sonora;
- effetti di campo vicino;

[8] https://www.unrealengine.com/en-US/

- diffusione della sorgente sonora;
- riverbero basato sulla geometria;
- occlusione;
- registrazione di file audio Ambisonic.

Google Resonance Audio non specifica il numero di sorgenti che può trattare complessivamente ma specifica che riproduce internamente tutte le sorgenti sonore in un campo sonoro Ambisonic di terzo ordine globale. Questo permette di applicare le HRTF solo una volta al campo sonoro piuttosto che alle singole sorgenti sonore al suo interno [9]. Questa ottimizzazione riduce al minimo i costi della CPU per sorgente sonora, consentendo la riproduzione di molte più sorgenti simultanee rispetto alla maggior parte delle tradizionali tecniche di spazializzazione per sorgente sonora. Tuttavia, l'ordine Ambisonic in Resonance Audio è regolabile, permettendo di controllare la risoluzione spaziale. L'utilizzo di Ambisonics di ordine superiore offre una maggiore fedeltà di uscita e una migliore localizzazione diretta della sorgente.
Per quanto riguarda riflessione e riverbero, il motore controlla la riflessione tramite un parametro che consente di controllare la forza delle prime riflessioni in una stanza, mentre il riverbero tramite 3 parametri che consentono di regolare volume degli effetti, bilanciare le frequenze e regolare la durata.

Steam Audio

Steam Audio [38] è un motore audio multipiattaforma sviluppato da Valve Corporation. L'SDK rilasciata è totalmente gratuita e non è limitata ad un particolare dispositivo VR o a Steam. Il motore audio progettato ha le seguenti caratteristiche:

- **Audio 3D per suoni diretti**: permette il rendering binaurale del suono diretto utilizzando HRTF per modellare accuratamente la direzione di una sorgente sonora rispetto all'ascoltatore;
- **Audio 3D per registrazioni Ambisonics**: permette anche di spazializzare le clip audio Ambisonics, ruotandole in base all'orientamento dell'ascoltatore nel mondo virtuale;

[9] https://www.waves.com/ambisonics-explained-guide-for-sound-engineers

- **HRTF personalizzate utilizzando file SOFA**: oltre alla sua HRTF integrata, Steam Audio può spazializzare sorgenti localizzate e Ambisonics utilizzando qualsiasi HRTF specificata dall'utente. Questi file devono essere in formato AES standard SOFA (vedi sezione 2.4) [32];

- **Occlusione e occlusione parziale**: può modellare l'occlusione utilizzando la tecnica di raycast del suono diretto anche in modo parziale;

- **Crea effetti ambientali e riverberi su misura per la scena**: analizza le dimensioni, la forma, il layout e le proprietà dei materiali delle stanze e degli oggetti della scena. Utilizza queste informazioni per calcolare automaticamente gli effetti ambientali simulando la fisica del suono;

- **Automatizzare il processo di creazione degli effetti ambientali**: utilizza un processo automatico in tempo reale o basato su pre-calcolo, in cui le proprietà audio ambientali sono calcolate (utilizzando principi fisici) in tutta la scena;

- **Supporto geometrie dinamiche**: può modellare come la geometria in movimento influisce sull'occlusione e sugli effetti ambientali;

- **Supporto per la localizzazione della testa**: Per le applicazioni VR, Steam Audio può utilizzare le informazioni di head tracking per far cambiare il campo sonoro in modo fluido e preciso quando l'ascoltatore gira o muove la testa.

Steam Audio offre un modo semplice per modellare il percorso diretto (lineare) del suono dalla sorgente all'ascoltatore, inclusi effetti come l'attenuazione della distanza, l'occlusione, la trasmissione e il rendering binaurale basato su HRTF. Innanzitutto si può scegliere se rendirazzare il suono diretto in modo binaurale o meno. Nel momento in cui si decide di utilizzare questo metodo, viene resa disponibile la scelta dell'interpolazione da usare per ricoprire tutte le posizioni nello spazio e le interpolazioni messe a disposizioni sono la "nearest" e la "bilinear". Nel caso in cui si utilizzino HRTF personalizzate, il motore permette di caricare più file nel formato SOFA con la possibilità di cambiare la selezione anche in runtime. Si passa poi alla gestione dell'occlusione, dove sono previsti 4 modelli di gestione (*Off*, *On, No Transmission*, *On, Frequency Independent Transmission*, *On, Frequency Dependent Transmission*). Questi modelli possono essere applicati utilizzando due metodi messi a disposizione: *Raycast*, ovvero viene generato un singolo raggio dalla sorgente all'ascoltatore per determinare l'occlusione, oppure *Partial*, si generano più raggi dalla sorgente all'ascoltatore e la porzione di raggi che sono occlusi determina quanto il suono sarà

occluso. Altri parametri che permettono di operare sulla diffusione diretta del suono sono la *Direzionalità*, *Physics Based Attenuation* e *Air Absorption* per regolare l'attenuazione.
Per quanto riguarda la diffusione dei suoni indiretti, Steam Audio permette di abilitare le riflessioni, analizzando tutte le caratteristiche che compongono la scena e applicando effetti ambientali. La simulazione può essere fatta in realtime o pre-computata e viene data la possibilità di scegliere se renderizzare o meno i suoni indiretti in modo binaurale.

3

Personalizzazione di HRTF in Steam Audio

In questa sezione verrà esposto quale ambiente di lavoro è stato realizzato per affrontare questo lavoro di tesi. Verranno esposte quindi le scelte effettuate riguardanti i vari componenti principali utilizzati fornendo una motivazione di tale scelta.
Originariamente la scelta è stata fatta tra Google Resonance Audio e Steam Audio poiché si cercava un motore audio che permettesse l'implementazione dell'uso di HRTF personalizzate (inizialmente Steam Audio non lo permetteva, ufficialmente implementato dalla release 2.0.16 beta), ma come vedremo col passare del tempo le cose si sono evolute. Quindi per quanto riguarda la scelta iniziale, entrambi hanno caratteristiche promettenti ma il motore audio selezionato è stato Steam Audio Engine. Il fattore principale è stato ovviamente che Steam Audio permettesse l'implementazione di HRTF personalizzate tramite l'SDK messa a disposizione, documentando tale argomento nel forum online dove veniva spiegato quali passi effettuare e quali metodi dichiarare, mentre Google Resonance Audio non permetteva nient'altro che l'utilizzo del database di HRTF affiliato, ovvero il SADIE. Inoltre Steam Audio concede di operare su molti più parametri all'interno del plugin importato in Unity per modificare vari attributi, ottenendo così un suono maggiormente dettagliato.
Intrapresa quindi questa scelta, in collaborazione con un altro tesista, Luca Buriola, è stato sviluppato un plugin proprio basato su Steam Audio che permettesse l'utilizzo

di HRTF personalizzate partendo da un file SOFA di un qualsiasi dataset. Tuttavia, anche il team di sviluppo di Valve ha lavorato in parallelo su questo tema ed ha rilasciato la release beta Steam Audio 2.0.16 e, vedendo e testando il plugin, alla luce delle importanti implementazioni portate, non era possibile trascurare tale soluzione. Di seguito viene portato un confronto tra questi 2 plugin ed un benchmark che ci ha condotto alla scelta finale.

3.1 Confronto dei plugin

Iniziamo intanto specificando quali sono i punti in cui i due plugin messi sotto analisi differiscono.
Il plugin sviluppato da parte nostra era basato sulla release ufficiale di Steam Audio 2.0.10. Tale realease non supportava la lettura immediata dei file SOFA. Utilizzando l'API messa a disposizione dai ricercatori che hanno sviluppato il formato SOFA, con un semplice script Matlab (illustrato qui sotto) è stato possibile generare una semplice funzione che potesse ispezionare un file SOFA generale e poter estrarre da esso le informazioni a noi indispensabili. Nel nostro caso sono stati estratti i campioni registrati delle HRIR, le relative posizioni a cui fanno riferimento i campioni di HRIR e i parametri M, N, R e la frequenza di campionamento del file SOFA. Con queste informazioni sono stati generati dei file di testo, rispettivamente "*dataleftIR.txt*", "*datarightIR.txt*", "*posizione.txt*" e "*datiSOFA.txt*", che saranno messi a disposizione del plugin di Steam Audio inserendoli nella SDK di partenza.

```
SOFAfile = 'CIPIC_subject_003_hrir_final.sofa';
cd './sofa/API_MO/';
SOFAstart;
cd '../../';
hrtf=SOFAload(SOFAfile);
Fs = hrtf.Data.SamplingRate;

filePOS = fopen('posizione.txt','wt');
fileleftIR = fopen('dataleftIR.txt','wt');
filerightIR = fopen('datarightIR.txt','wt');
fileDATI = fopen('datiSOFA.txt','wt');

for ii = 1:size(hrtf.SourcePosition,1)
          ii = 1;

            hinfo.azimuth   = hrtf.SourcePosition(ii,1);
            hinfo.elevation = hrtf.SourcePosition(ii,2);
```

```
            hinfo.distance  = hrtf.SourcePosition(ii,3);

            fprintf(filePOS, '%f, %f, %f,', hinfo.azimuth, hinfo.elevation, hinfo.distance);

            % Extract hrir data
            hdataleft = squeeze(hrtf.Data.IR(ii,1,:));
            hdataright = squeeze(hrtf.Data.IR(ii,2,:));
            %check if it is coloumn-wise
            if ndims(hdataleft) > 2,
                error('Data array cannot be an N-D array.');
            end

            fprintf(fileleftIR, '%E,' , hdataleft);
            fprintf(filerightIR, '%E,' , hdataright);
end

fprintf(fileDATI, '%d,' , size(hrtf.Data.IR));
fprintf(fileDATI, '%d', hrtf.Data.SamplingRate);

fclose(filePOS);
fclose(fileleftIR);
fclose(filerightIR);
fclose(fileDATI);
```

Successivamente, seguendo le linee base indicate nel forum online di Steam [39], sono state implementate le 3 funzioni consigliate *OnLoadHrtf*, *OnLookUpHrtf*, *onUnloadHrtf*, che rispettivamente servono per integrare nella libreria Steam la possibilità di caricare una HRTF, applicare il rendering e rilasciare l'HRTF caricata in precedenza. Tutte le modifiche effettuate sono state apportate all'interno del file "*audio_engine_settings.cpp*"della libreria. All'interno della funzione *OnLookUpHrtf* viene applicata direttamente l'interpolazione e nel nostro caso è stata utilizzata la triangolazione di Delaunay (vedi sottosezione 2.2.2).

OnLoadHrtf

```
//Onload gets called when the HRTF needs to be loaded in.
void onLoadHrtf(IPLint32 numSamples, IPLint32 numSpectrumSamples,IPLFftHelper fftHelper, void* fftHelperData)
{
^^IcreatePoint();

  LOG << "onLoadHrtf" << endl;

^^I//Save the number of spectrum samples for later.
^^IhrtfSize = numSpectrumSamples;

```

```
  LOG << " hrtfSize: " << hrtfSize << endl;
  LOG << " numSamples: " << numSamples << endl;


^^I//This part comes straight from the custom HRTF blog made by Steam Audio.
^^Ifor (int i = 0; i < points.size(); i++)
   {
^^I^^I//Left ear
^^I^^Ifloat* tempLeftHrir = new float[numSamples];

^^I^^I//Zero padding
^^I^^Imemset(tempLeftHrir, 0, numSamples * sizeof(float));
^^I^^Imemcpy(tempLeftHrir, points.at(i).leftHrir, N * sizeof(float));

^^I^^Ipoints.at(i).leftHrtf = new IPLComplex[numSpectrumSamples];

^^I^^I//fft
^^I^^IfftHelper(fftHelperData,tempLeftHrir, points.at(i).leftHrtf);

^^I^^I//Right ear
^^I^^Ifloat* tempRightHrir = new float[numSamples];

^^I^^I//Zero padding
^^I^^Imemset(tempRightHrir, 0, numSamples * sizeof(float));
^^I^^Imemcpy(tempRightHrir, points.at(i).rightHrir, N * sizeof(float));

^^I^^Ipoints.at(i).rightHrtf = new IPLComplex[numSpectrumSamples];

^^I^^I//fft
^^I^^IfftHelper(fftHelperData, tempRightHrir, points.at(i).rightHrtf);

^^I^^Idelete[] tempLeftHrir;
^^I^^Idelete[] tempRightHrir;
^^I}

^^I//No longer needs hrirs after this.
^^Ifor (int i = 0; i < points.size(); i++)
   {
^^I^^Idelete[] points.at(i).leftHrir;
^^I^^Idelete[] points.at(i).rightHrir;
^^I}
}
```

OnLookupHrtf

```
//The function that is called whenever a sound needs HRTF data.
void onLookupHrtf(float* direction, IPLComplex* leftHrtf, IPLComplex* rightHrtf)
{

  LOG << "*************** INIZIO onLookupHrtf  ***************" << endl;
```

```
6
7
8   ^^I//Unity and Steam Audio use a Y-up coordinate system, SOFA uses a z-up system.
9   ^^Ifloat y = -direction[0];
10  ^^Ifloat x = -direction[2];
11  ^^Ifloat z = direction[1];
12
13     //HRTF is elevation/azimuth, therefore convert.
14  ^^Ifloat radius = sqrt(pow(x, 2) + pow(y, 2) + pow(z, 2));
15  ^^Ifloat elevation = asin(z/radius) * (180.0 / M_PI);
16  ^^Ifloat azimuth = atan2f(y, x) * (180.0 / M_PI) ;
17
18     LOG << "direction[0]: " << direction[0] << endl;
19     LOG << "direction[1]: " << direction[1] << endl;
20     LOG << "direction[2]: " << direction[2] << endl;
21
22     //atan2f returns -pi to pi. Therefore azimuth is -180 to 180. If azimuth is negative, convert it to appropriate
23  ^^Iif (azimuth < 0)
24  ^^I{
25  ^^I^^Iazimuth += 360;
26  ^^I}
27
28    LOG << " ----------- INIZIO TRIANGOLO -----------------"  << endl;
29
30       for (auto& triangle : triangles)
31  ^^I  {
32
33  ^^I^^I  const auto A = triangle.p1;
34  ^^I^^I  const auto B = triangle.p2;
35  ^^I^^I  const auto C = triangle.p3;
36
37        const double T[] = {A.x - C.x, A.y - C.y, B.x - C.x, B.y - C.y};
38  ^^I^^I  double invT[] = {T[3], -T[1], -T[2], T[0]};
39  ^^I^^I  const auto det = T[0] * T[3] - T[1] * T[2];
40
41        for (auto i = 0; i < 4; ++i)
42  ^^I^^I^^IinvT[i] /= det;
43  ^^I^^I  const double X[] = {azimuth - C.x, elevation - C.y};
44
45        // Barycentric coordinates of point X
46  ^^I^^I  auto g1 = static_cast<float>(invT[0] * X[0] + invT[2] * X[1]);
47  ^^I^^I  auto g2 = static_cast<float>(invT[1] * X[0] + invT[3] * X[1]);
48  ^^I^^I  auto g3 = 1 - g1 - g2;
49
50        // If any of the barycentric coordinate is negative, the point
51  ^^I^^I  // does not lay inside the triangle, so continue the loop.
52
53        if (g1 < 0 || g2 < 0 || g3 < 0)
54  ^^I^^I^^Icontinue;
55
56        const auto& irA = points.at(mapind[make_pair(A.x,A.y)]);
```

```
57          const auto& irB = points.at(mapind[make_pair(B.x,B.y)]);
58          const auto& irC = points.at(mapind[make_pair(C.x,C.y)]);
59
60          IPLComplex* leftHrtfBuffer = new IPLComplex[hrtfSize];
61  ^^I     IPLComplex* rightHrtfBuffer = new IPLComplex[hrtfSize];
62
63          for (int i = 0; i < hrtfSize; i++)
64          {
65
66             leftHrtfBuffer[i].real = g1 * irA.leftHrtf[i].real + g2 * irB.leftHrtf[i].real + g3 * irC.leftHrtf[i].real;
67             leftHrtfBuffer[i].imag = g1 * irA.leftHrtf[i].imag + g2 * irB.leftHrtf[i].imag + g3 * irC.leftHrtf[i].imag;
68             rightHrtfBuffer[i].real = g1 * irA.rightHrtf[i].real + g2 * irB.rightHrtf[i].real + g3 * irC.rightHrtf[i].real;
69             rightHrtfBuffer[i].imag = g1 * irA.rightHrtf[i].imag + g2 * irB.rightHrtf[i].imag + g3 * irC.rightHrtf[i].imag;
70
71          }
72
73          memcpy(leftHrtf, leftHrtfBuffer, hrtfSize * sizeof(IPLComplex));
74          memcpy(rightHrtf, rightHrtfBuffer, hrtfSize * sizeof(IPLComplex));
75
76          break;
77      }
78
79      LOG << "*************** FINE onLookupHrtf *************************" << endl;
80
81  }
```

OnUnloadHrtf

```
1   void onUnloadHrtf()
2   {
3   ^^Ifor (int i = 0; i < points.size();i++) {
4   ^^I^^Idelete[] points.at(i).leftHrtf;
5   ^^I^^Idelete[] points.at(i).rightHrtf;
6   ^^I}
7   ^^Ipoints.clear();
8   ^^Ipoints = vector<Point>();
9   }
```

La release beta 2.0.16 di Steam Audio sviluppata dal team di Valve invece permette di poter caricare all'interno del progetto Unity direttamente i file SOFA desiderati con cui si intende lavorare, andando a specificare all'interno dello Steam Audio Manager il numero totale di tali file e specificare il loro nome. Inoltre sarà permesso cambiare durante l'esecuzione il file SOFA e di conseguenza la renderizzazione derivante. Come detto in precedenza, Steam Audio Engine dà la possibilità di scegliere l'interpolazione da utilizzare tra "nearest" e "bilinear".

3.2 Benchmark

Dopo aver ispezionato ciascun plugin per capire le differenze principali che li contraddistinguono, è necessario capire quale sia il plugin migliore e più adatto su cui basare l'esperimento. Per poter trarre questa decisione è stato effettuato un benchmark diretto tra i due plugin interessati, vedendo quale dei due si comportasse meglio sotto determinate condizioni. Servendosi di Unity come motore grafico, sono stati creati 3 ambienti di prova con caratteristiche differenti dal punto di vista grafico/ambientale, in modo tale che rappresentassero situazioni diverse in cui di conseguenza il suono rispettivo si trovasse in situazioni diverse su cui essere testato.
Il *primo ambiente* di prova è uno spazio base totalmente vuoto in cui vengono inserite solamente le sorgenti audio al suo interno. Il *secondo ambiente* di prova invece riproduce l'interno di una struttura composto da 2 stanze, completamente vuote. Alle pareti che compongono tale struttura viene applicato lo stesso *Steam Audio Material* (in questo caso Generic). Il *terzo ambiente* di prova infine presenta un luogo più complesso e articolato, dove troviamo un ambiente esterno esteso e al cui interno sono situati 3 strutture composte da un numero diverso di stanze. Va specificato che l'ambiente è stato dettagliato inserendo particolari sia esterni che interni, arricchendo la scena come un ambiente di test finale. Agli oggetti appartenenti alla scena sono stati assegnati *Steam Audio Material* specifici, quindi troviamo oggetti con Brick (Mattone), Concrete (Calcestruzzo), Wood (Legno), Plastic (Plastica), Rock (Roccia) e Ground (Terreno).
Le sorgenti audio all'interno dei vari ambienti erano rappresentati da dei cubi unitari bianchi e sono state posizionate in posizioni casuali, tenute però fisse ovviamente per ogni test di benchmark. Come traccie audio sono stati utilizzati 20 brani differenti, ciascuno assegnato ad una unica sorgente.
Per ciascun plugin, su ciascun ambiente, sono stati testati i seguenti settaggi: entrambi erano impostati per l'audio 3D spazializzato, utilizzando entrambi la stessa HRTF (*subject_003.sofa* del CIPIC) e sono state variate la tipologia di interpolazione e l'abilitazione delle riflessioni, testando così le 4 combinazioni "Nearest", "Nearest + riflessioni", "Bilinear / Delaunay" e "Bilinear + riflessioni / Delaunay + riflessioni". Inoltre il numero di sorgenti sonore attive è stato variato, studiando nello specifico i casi con 1, 5, 10 e 20.
I dati sono stati raccolti direttamente tramite lo Unity Profiler che permette di vedere

in tempo reale il consumo delle risorse messe a disposizione. Nello specifico sono stati presi in considerazione i dati riguardanti la percentuale di utilizzo totale della CPU, la percentuale di utilizzo totale della CPU da parte della componente audio, il totale della memoria locale utilizzata e il totale della memoria locale utilizzata dalla componente audio. Di seguito vengono riportate le varie tabelle con i risultati ottenuti dall'analisi dei dati raccolti. Per i dati *Tot Audio Cpu* e *Tot Memoria Allocata*, poiché sono valori variabili nel tempo di analisi del test, sono stati campionati e vengono riportati il valore medio con la varianza. Come possiamo notare, i dati sistemati nelle apposite tabelle riportano già delle evidenti differenze tra i plugin analizzati ma, per una maggior comprensione, sono stati riprodotti degli appositi grafici significativi qui di seguito.

GRAFICI BENCHMARCK AMBIENTE 1

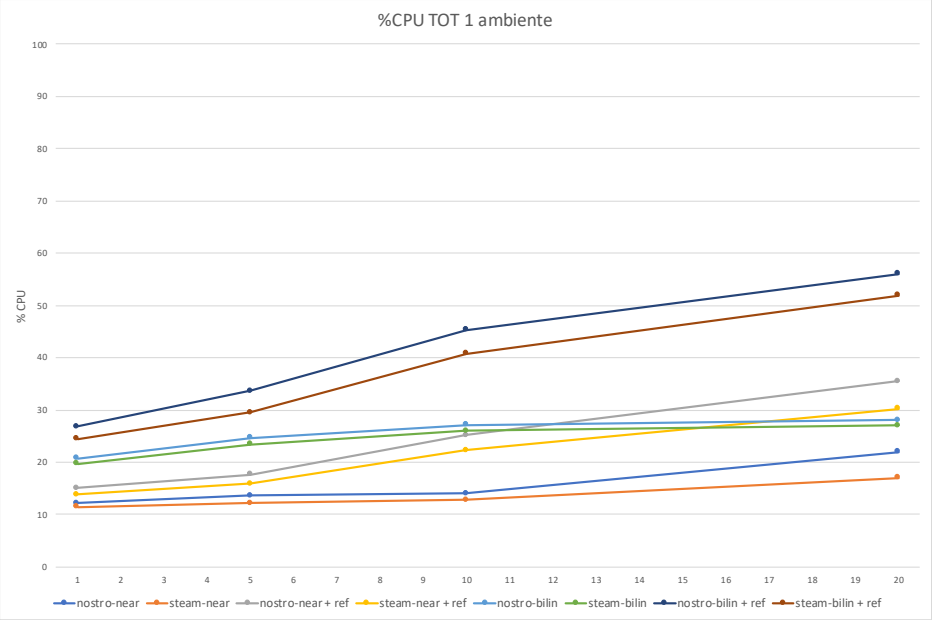


Figura 3.1: Percentuale di utilizzo della CPU in totale nel primo ambiente di benchmark

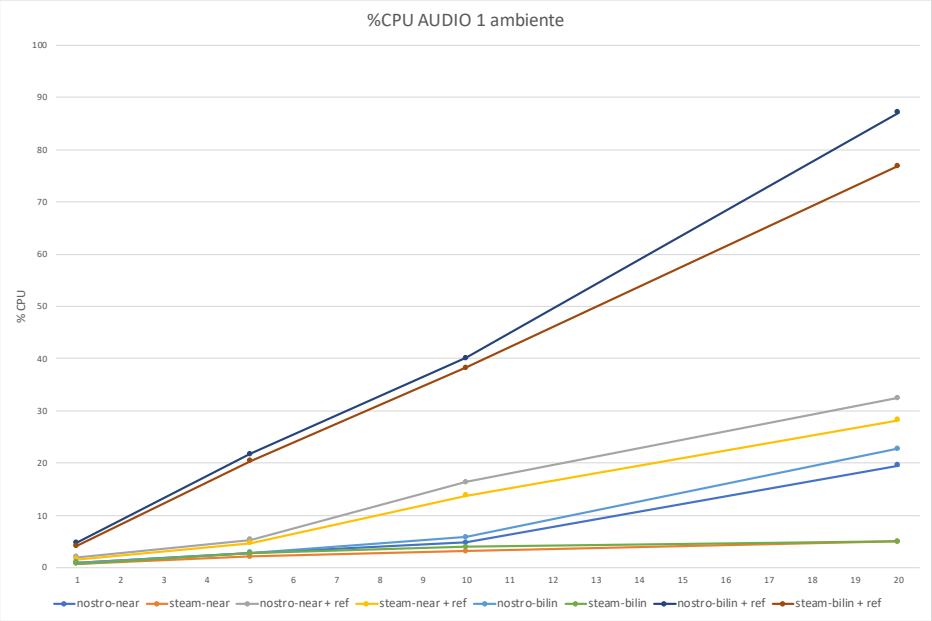


Figura 3.2: Percentuale di utilizzo della CPU dalla componente audio nel primo ambiente di benchmark

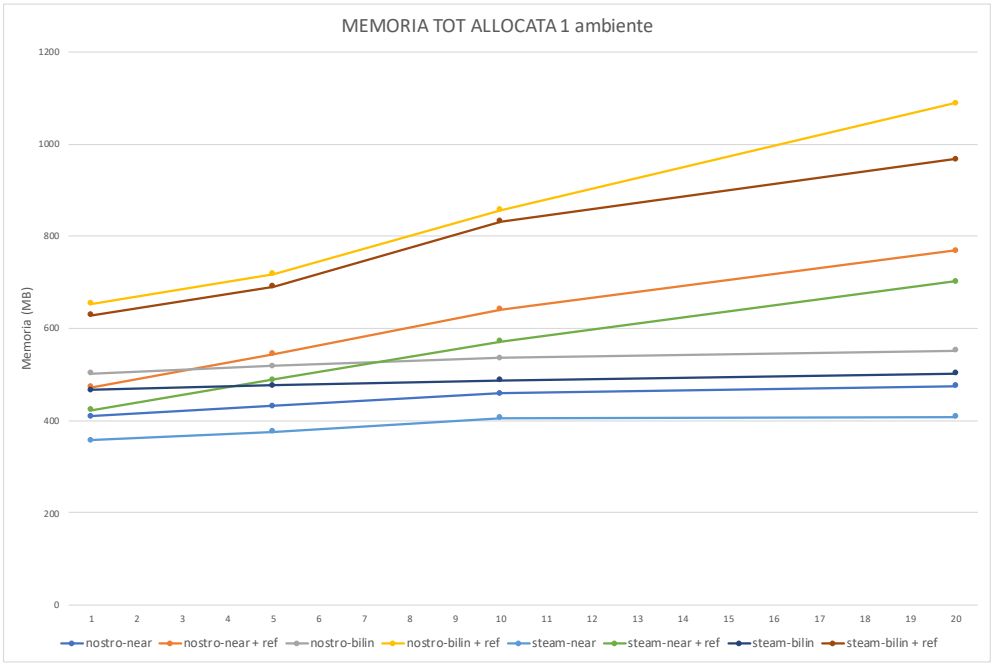


Figura 3.3: Memoria totale allocata nel primo ambiente di benchmark

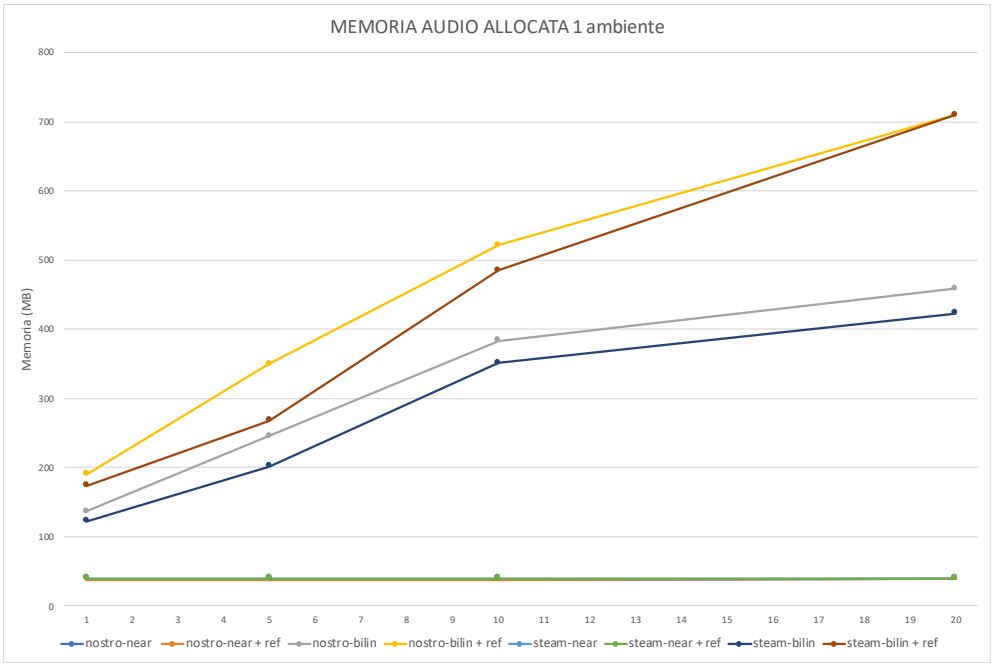


Figura 3.4: Memoria allocata per la componente audio nel primo ambiente di benchmark

GRAFICI BENCHMARCK AMBIENTE 2

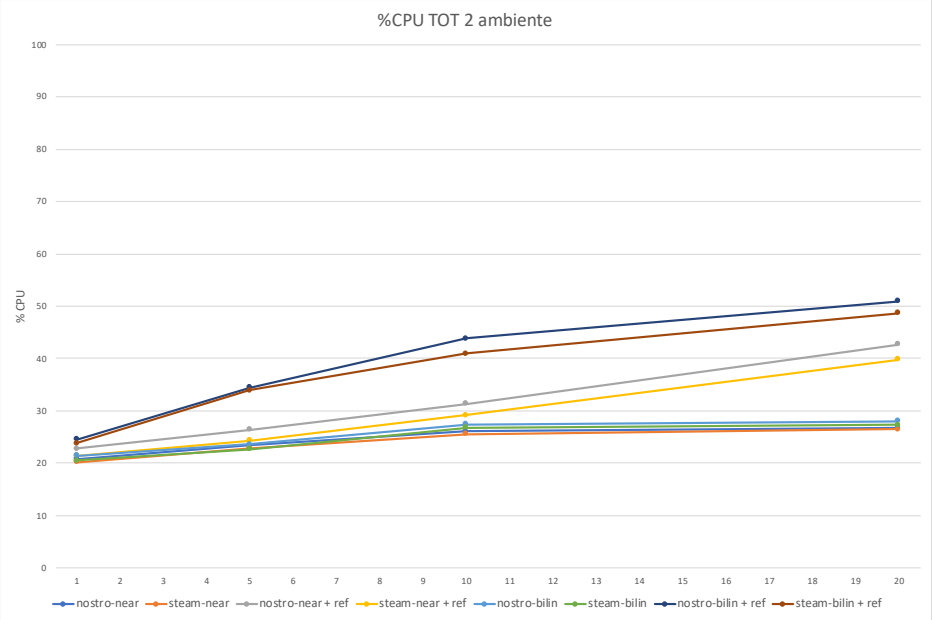


Figura 3.5: Percentuale di utilizzo della CPU in totale nel secondo ambiente di benchmark

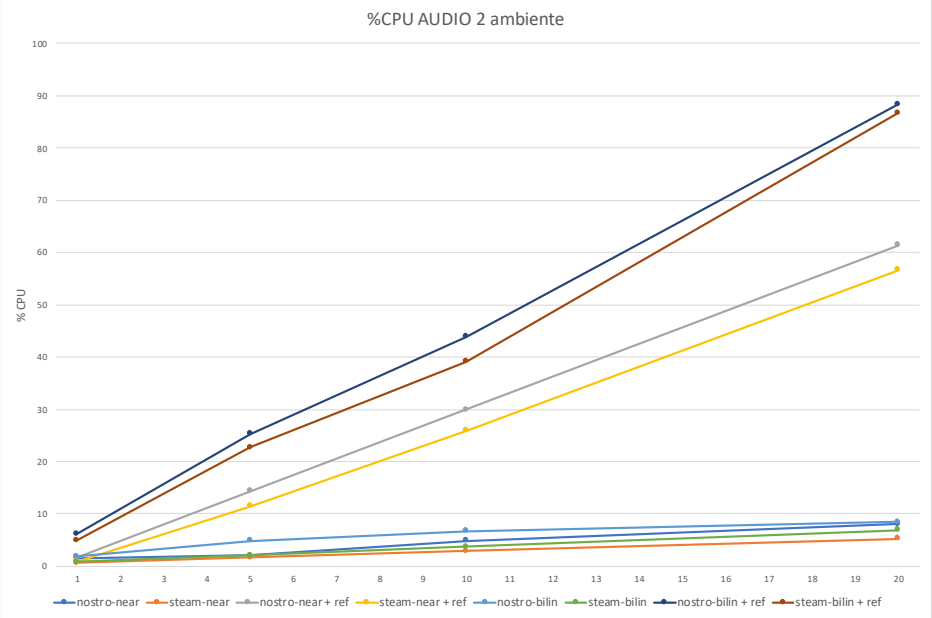


Figura 3.6: Percentuale di utilizzo della CPU dalla componente audio nel secondo ambiente di benchmark

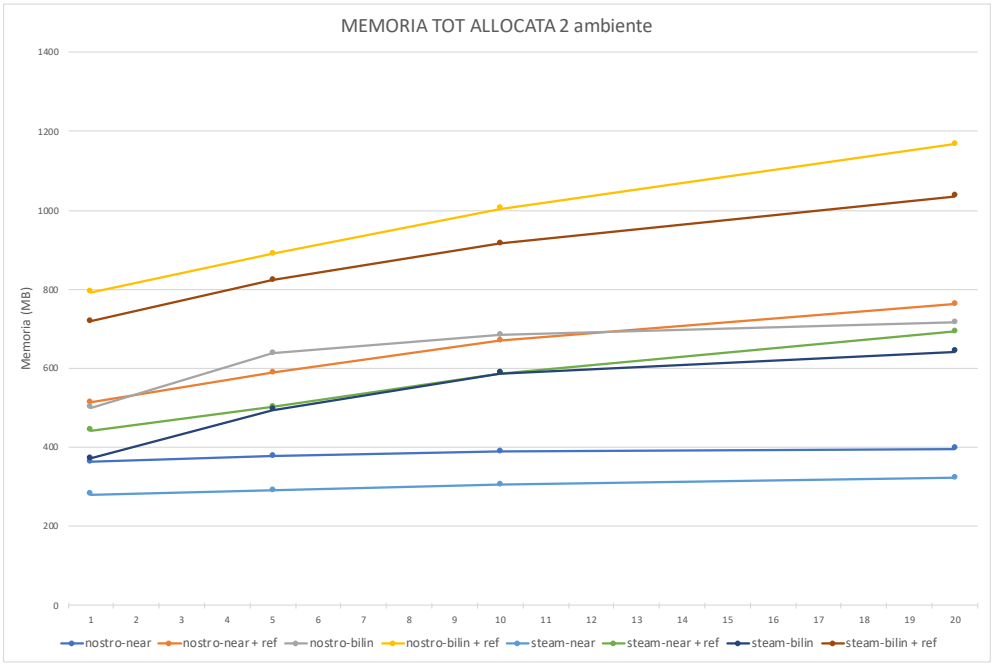


Figura 3.7: Memoria totale allocata nel secondo ambiente di benchmark

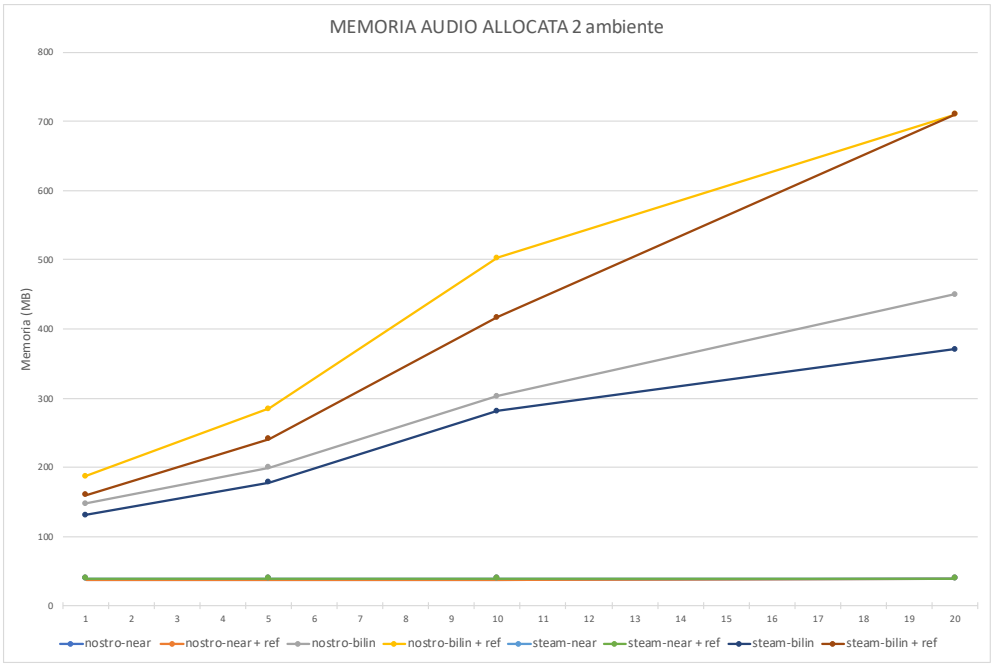


Figura 3.8: Memoria allocata per la componente audio nel secondo ambiente di benchmark

GRAFICI BENCHMARCK AMBIENTE 3

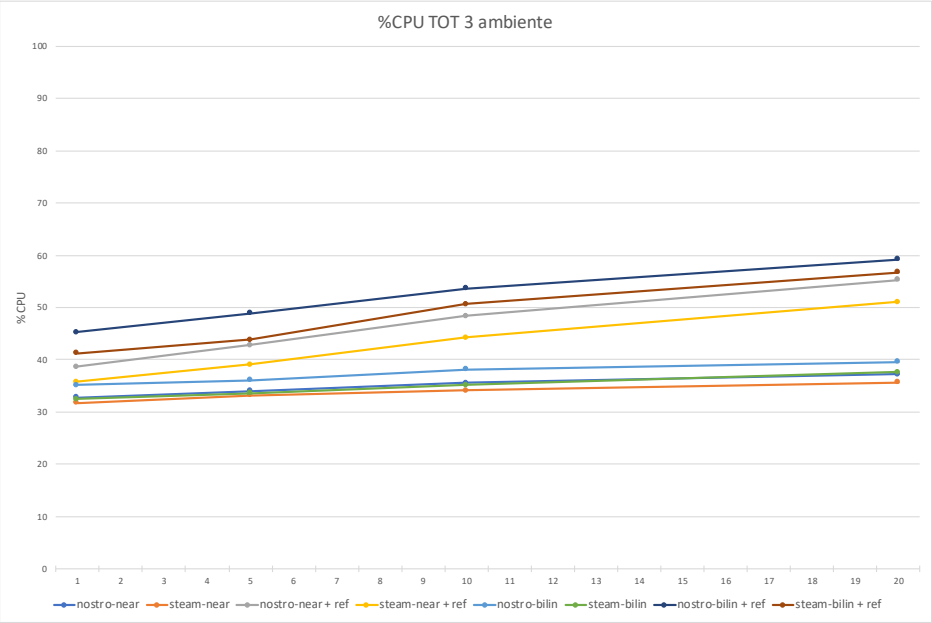


Figura 3.9: Percentuale di utilizzo della CPU in totale nel terzo ambiente di benchmark

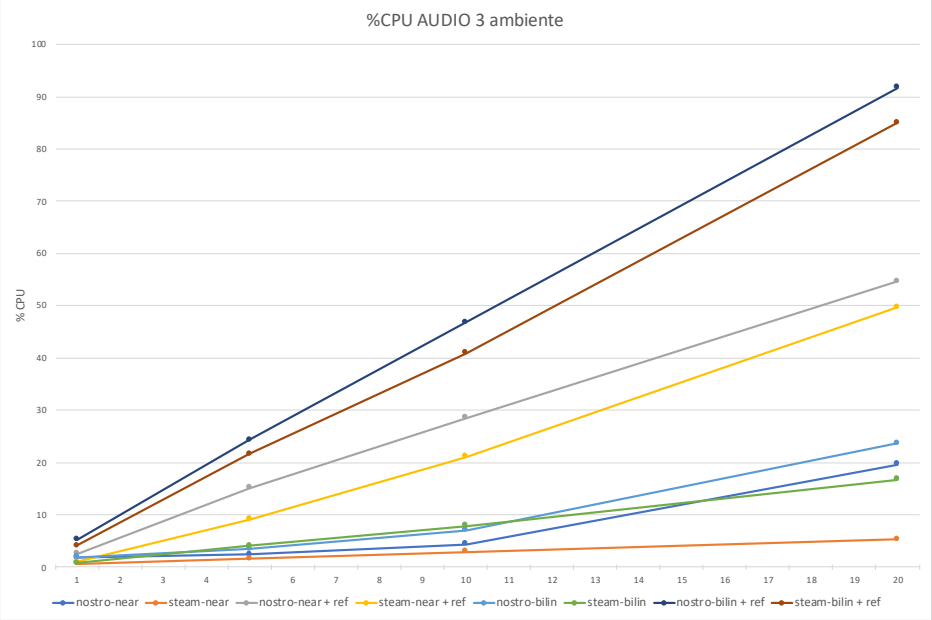


Figura 3.10: Percentuale di utilizzo della CPU dalla componente audio nel terzo ambiente di benchmark

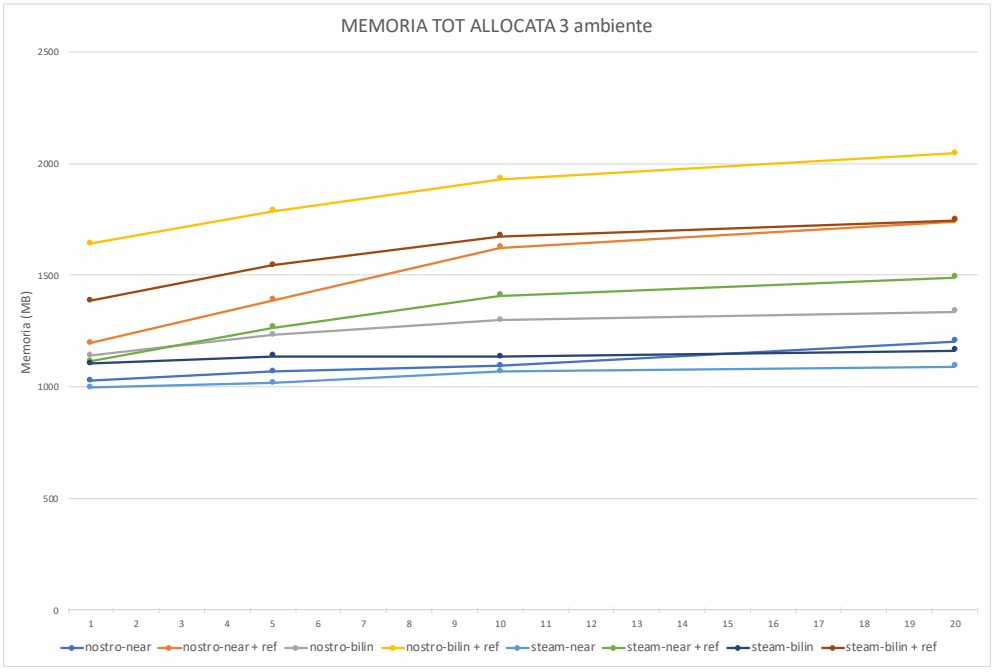


Figura 3.11: Memoria totale allocata nel terzo ambiente di benchmark

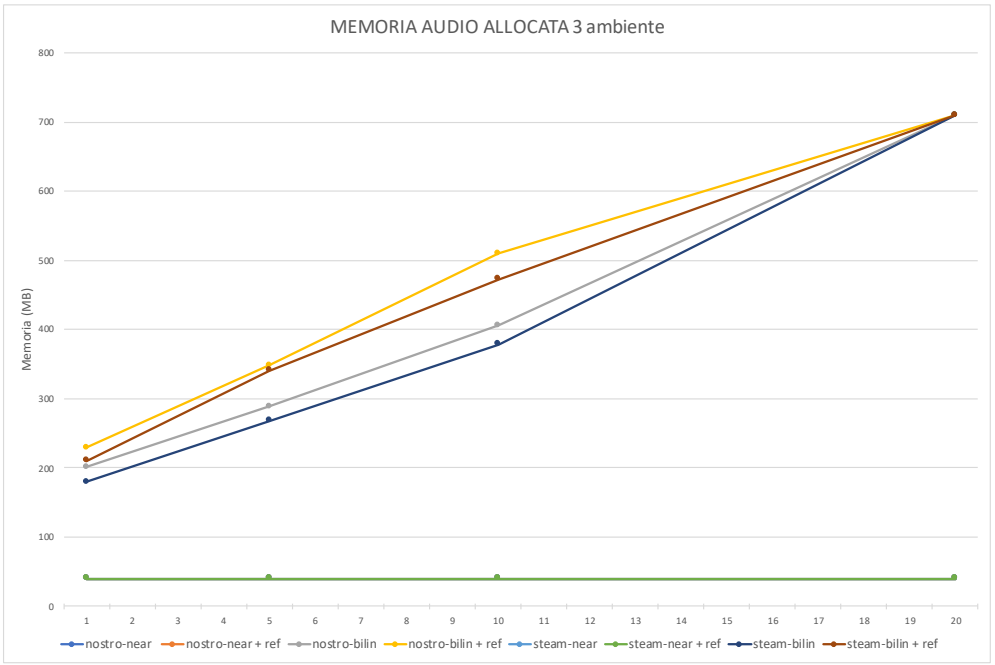


Figura 3.12: Memoria allocata per la componente audio nel terzo ambiente di benchmark

Quello che si può subito notare in generale è che entrambi i plugin in analisi si comportano allo stesso modo, poiché entrambi utilizzano un core centrale Steam Audio, ma il plugin da noi sviluppato tende a richiedere un utilizzo maggiore di risorse, nel caso di utilizzo della CPU leggermente mentre per quanto riguarda l'allocazione della memoria più considerevole. Entrambi hanno degli andamenti crescenti costanti con l'aumentare del numero delle sorgenti in riproduzione, tranne nei casi di allocazione di memoria per la componente audio con interpolazione "Nearest", dove la memoria viene tenuta ad un valore costante. Notiamo in oltre che per la memoria audio si può incorrere in situazioni di saturazione con valori elevati di numero di sorgenti.
Un altro accorgimento che si nota subito è come incida il suono indiretto, in questo caso il fenomeno della riflessione, che va a richiedere un notevole sforzo alle risorse in gioco rispetto ai casi simili di riproduzione del solo suono diretto. Si nota inoltre che con l'aumento della complessità della scena e dei particolari usati (dettagli, materiali,...), al sistema viene richiesta una maggiore quantità di memoria per la gestione dell'ambiente.
Andando quindi ad esaminare velocemente più da vicino il terzo ambiente di benchmark, poiché è quello che alla fine più si avvicina alla realtà degli ambienti da sviluppare per l'esperimento, soprattutto fino ad un massimo di 5 sorgenti sonore, vediamo nei grafici di Fig.3.9, Fig.3.10, Fig.3.11 e Fig.3.12 la riconferma di come il plugin di Steam Audio 2.0.16 sfrutti in modo migliore le risorse messe a disposizione dal sistema.
Quindi alla fine di questa analisi, la decisione presa è stata quella di utilizzare la release 2.0.16 di Steam Audio come motore audio da integrare negli ambienti Unity. Anche se non di molto, Steam Audio offre una maggior efficacia delle risorse utilizzata e una maggior stabilità nell'utilizzo. Cosa soprattutto da non sottovalutare nel nostro caso, offre la possibilità di caricare da subito svariati file SOFA notoriamente contenenti HRTF diverse (il plugin nostro invece non supportava ancora questa funzione) e la possibilità di cambiare in realtime tali file, con la conseguenza di cambiare anche il rendering derivante, altra funzionalità che non supportata dal nostro plugin. L'interpolazione usata per l'esperimento sarà la bilinear utilizzata da Steam Audio 2.0.16.

4

Qualità dell'esperienza: impatto della personalizzazione

Il vantaggio dell'utilizzo di HRTF nell'audio binaurale è ben documentato per quanto riguarda il miglioramento della precisione di localizzazione [34][40]. Tuttavia, con l'aumento dell'uso dell'audio binaurale in scenari più complessi, studi cognitivi e simulazioni di realtà virtuale e aumentata, l'impatto percettivo della selezione HRTF non è del tutto chiaro, mancando rigorose valutazioni della qualità d'ascolto e dell'esperienza in VR.

4.1 Lavori correlati

I vari studi svolti a riguardo valutano renderer derivanti dalla spazializzazione delle HRTF racchiuse nel formato SOFA attraverso criteri diversi ma un fattore ovviamente li accomuna: la valutazione di questi test porta ad una classifica, cioè ottenere una graduatoria finale che identifichi quale sia il renderer più adatto al soggetto. Un primo caso importante si tratta dello studio riportato in [41] condotto da Olli Rummukainen *et al.*. Lo studio tratta sempre il tema della localizzazione. Ai soggetti vengono sottoposti 5 diversi renderer derivanti dalla stessa HRTF ma differenziati per diversi parametri di latenza. Rummukainen in questo studio introduce il concetto di clas-

sifica basato su un metodo alternativo per valutare i sistemi di riproduzione audio, ovvero misurare le prestazioni dei partecipanti. Il compito è quello di individuare una sorgente sonora e raggiungere a piedi la sua posizione il più velocemente possibile.
Un altro esperimento interessante preso in considerazione è quello condotto da Timo Hedke *et al.* in [42]. In questo studio sono stati analizzati casi precedenti sull'esperienza dei giochi per computer in cui viene usato una tipologia di suono surround che riporta un effetto positivo per quanto riguarda divertimento, interesse e sensazione di presenza rispetto ad un renderer stereo del suono. È stato quindi condotto uno studio su diversi utenti che indaga l'impatto dell'audio binaurale sulla QoE percepita in un gioco per computer. Dopo aver sostenuto al computer la prova sperimentale su una condizione, il soggetto ha compilato un questionario articolato in 29 quesiti che misura la QoE di gioco attraverso il Mean Opinion Score (MOS), il questionario sull'esperienza di gioco e questionari realizzati a mano per diversi aspetti della percezione uditiva, oltre che per la presenza.
In [43] Reardon *et al.* si concentrano sulla definizione di una procedura per la valutazione e la caratterizzazione delle tecnologie binaurali. La prima fase del test tratta la valutazione quantitativa ed è incentrata sulla valutazione fondamentale dell'immagine uditiva 3D del renderer binaurale: esternalizzazione, confusioni fronte/retro e su/giù e localizzazione. La seconda fase, nota come valutazione qualitativa, si concentra sugli attributi più generali dell'immagine uditiva: ai soggetti è stato chiesto di valutare discretamente su una scala da 1 a 5 gli attributi per le clip audio surround rese binaurali. La fase finale consiste in una valutazione delle preferenze, dove l'utente è in grado di classificare i renderer presentati da meno preferiti a più preferiti. Questa classificazione viene successivamente utilizzata per studiare le correlazioni tra gli attributi di qualità del suono e le preferenze dell'ascoltatore.
Vediamo ora invece i due studi principali che hanno portato alla luce l'analisi condotta in questa tesi. Partiamo dalla ricerca condotta da Rummukainen in [12] dove avviene una valutazione qualitativa su dei renderer audio in ambienti di realtà virtuale. In questo esperimento viene adattata la classica metodologia di test a stimolo multiplo alla realtà virtuale e vengono aggiunte funzionalità di monitoraggio comportamentale. Il metodo si basa sulla classificazione per eliminazione durante l'esplorazione di una realtà virtuale audiovisiva. Il metodo di valutazione proposto consente l'immersione in scene virtuali multimodali, consentendo la valutazione comparativa di più renderer binaurali.

Un aspetto importante della valutazione critica dell'audio è la seguente possibilità di passare da un renderer all'altro senza dover fare affidamento sulla memoria uditiva dei partecipanti. È stata sviluppata una piattaforma per la valutazione in tempo reale dei renderer binaurali che permette ai partecipanti di passare da una condizione all'altra, senza interruzione dell'input sensoriale audiovisivo. Poiché i soggetti che hanno sostenuto l'esperienza non sono posizionati in un luogo statico, l'interfaccia di controllo del test è stata implementata all'interno dell'ambiente VR stesso, consentendo la piena libertà di movimento senza essere costretti a tornare in un luogo specifico per interagire con l'interfaccia dell'esperimento. L'interfaccia è stata progettata in modo tale da poter essere installata ovunque nella scena VR. Essa presenta al suo interno 4 pulsanti nominati con delle lettere, con i relativi pulsanti per la cancellazione, che identificano e attivano i renderer predisposti per l'analisi. I renderer utilizzati sono:

- *Renderer 1* usa un modello strutturale per produrre HRTF per ogni oggetto audio in tempo reale data la posizione e l'orientamento della testa dell'ascoltatore e le posizioni dell'oggetto audio;
- *Renderer 2* usa una HTRF selezionata da un database e sceglie la coppia più vicina per ogni oggetto audio della scena;
- *Renderer 3* utilizza altoparlanti virtuali ed una HRTF dummy-head con l'aggiunta di riverbero artificiale;
- *Stereo Mix* è un renderer stereo statico di tutti gli oggetti audio di una data scena e non è influenzato dalla posizione o dall'orientamento dell'ascoltatore. Questo renderer può essere considerato come una condizione di riferimento.

Questi renderer sono stati classificati dai soggetti in 3 diversi ambienti VR in cui l'utente può muoversi liberamente all'interno di uno spazio tracciato di $2mx3m$. Le differenze principali tra le 3 scene sono:

- *Ambiente 1*: scena di un ristorante con tre oggetti audio, ovvero chitarra, chiacchiere ambientali e apertura di bottiglie. Tutti gli oggetti audio si trovano al di fuori dello spazio tracciato, ma relativamente vicino;
- *Ambiente 2*: scena di un soggiorno con un pianoforte suonato all'interno dello spazio tracciato.

- *Ambiente 3*: Scena esterna con quattro oggetti audio distanti (taglio di alberi, anatre, aerei e uccelli). Tutti gli oggetti audio sono al di fuori dello spazio tracciato.

Questa metodologia di classificazione però valuta i renderer solo su un fattore di realismo degli stimoli utilizzati. Per questo è stato preso in considerazione il secondo studio, la ricerca svolta da Geronazzo *et al.* in [40]. Questo articolo indaga la connessione tra la sensibilità di localizzazione verticale di segnali da parte dell'ascoltatore e la QoE, la qualità audio spaziale e l'attenzione. Sono state fornite agli utenti delle HRTF personalizzate che sono state selezionate individualmente sulla base dei dati antropometrici dell'orecchio esterno. Vengono proposti due esperimenti VR con display montato sulla testa: un test di screening volto a valutare la performance di localizzazione dei partecipanti con HRTF per una sorgente audio spazializzata non visibile, e un'esplorazione di 2 minuti di una scena VR con cinque sorgenti audiovisive sia in condizioni di ascolto non spazializzate (panning stereo 2D) che spazializzate (rendering HRTF in campo libero). Il secondo esperimento è composto da tre prove, una per ogni condizione audio (*Stereo*, *Generic HRTF*, *Custom HRTF*), in ordine casuale. Dopo ogni prova, è stata concessa una pausa e ai soggetti è stato chiesto di compilare un questionario composto da 8 domande in cui viene chiesto di valutare su una scala di 7 punti degli attributi sulla QoE che sono: *Esternalizzazione* (più internalizzato - più esternalizzato), *Reattività* (ritardo inferiore - ritardo maggiore), *Naturalezza* (minore naturalezza - maggiore naturalezza), *Presenza* (bassa - alta), *Attenzione audio* (poco - molto), *Attenzione visiva* (poco - molto), *Realismo* (meno realistico - più realistico) ed infine *Elevation* (Sì - No). Attraverso questo questionario è stato quindi possibile descrivere ed ottenere informazioni riguardanti la scena VR che andassero oltre un semplice parere sull'esperienza generale.

4.2 Esperimento svolto

Basandoci sui lavori condotti in [12] e [40], l'obiettivo di questa tesi è quello di formulare una procedura di valutazione, attraverso una classifica per eliminazione, della qualità audio di una esperienza di realtà virtuale che utilizza 6 diversi renderer binaurali messi a confronto. La classifica che un soggetto sottoposto al test deve comporre però non è basata sull'esperienza generale, ma andrà a valutare singolarmente i 6

renderer individuati per una lista di attributi scelti e per ciascun ambiente di realtà virtuale.

4.2.1 Ambienti VR creati

Gli ambienti sono creati si diversificano tra loro per come sono composti e le situazioni che riproducono si ispirano agli ambienti usati in [12].
Con questi 3 ambienti si è cercato di riprodurre delle scene che si distinguessero e fossero uniche tra di loro in modo che rappresentassero scene differenti per numero di sorgenti presenti nella scena, numero di stanze e sorgenti per stanza, ambiente interno o esterno.
L'*Ambiente 1*, raffigurato in Fig.4.1 e Fig.4.2, inscena un ampio salotto al cui interno è presente un piano in riproduzione con un pezzo di musica classica.

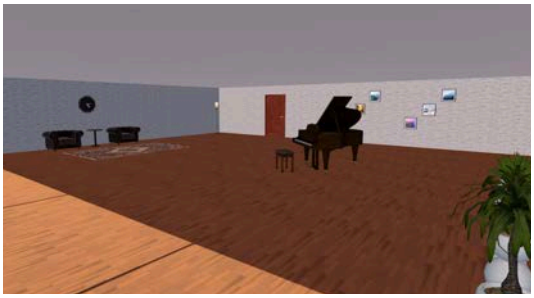


Figura 4.1: Illustrazione del primo ambiente VR

Figura 4.2: Prospettiva dall'alto del primo ambiente VR. In rosso la posizione della sorgente audio

L'*Ambiente 2* inscena un ufficio retrò articolato in 3 stanze: la stanza (vedi Fig.4.3) in cui il soggetto si trova all'inizio rappresenta la sala d'ingresso, comunicante con entrambe le altre due stanze, in cui è presente una radio d'epoca che sta trasmettendo. Da qui si può passare alla seconda stanza che rappresenta la sala d'attesa/segreteria e la sorgente audio è data dal telefono presente sulla scrivania. Per una maggiore comprensione si può consultare la Fig.4.4. Nell'ultima stanza invece le sorgenti audio presenti solo il videoproiettore e il ventilatore al soffitto, come si può vedere in Fig.4.5. In Fig.4.6 mostra la vista dall'alto generale della mappa dell'Ambiente 2.
L'*Ambiente 3* invece cambia totalmente scenario poiché rappresenta un scena esterna. Il perimetro dell'area in cui è possibile muoversi è delimitata da una recinzione al cui

Figura 4.3: Illustrazione del secondo ambiente VR, stanza con radio

Figura 4.4: Illustrazione del secondo ambiente VR, stanza con telefono

Figura 4.5: Illustrazione del secondo ambiente VR, stanza con proiettore e ventilatore

Figura 4.6: Prospettiva dall'alto del secondo ambiente VR. In rosso le posizioni della sorgenti audio

interno si trova un'officina. Le fonti audio che troviamo in questo ambiente sono il compressore all'interno della struttura, il motore della macchina, la sbarra d'entrata e degli uccellini situati sugli alberi esterni alla recinzione vicino all'officina. La Fig.4.8 illustra la scena dell'Ambiente 3 con una visuale dall'alto.

Figura 4.7: Illustrazione del terzo ambiente VR

Figura 4.8: Prospettiva dall'alto del terzo ambiente VR. In rosso le posizioni della sorgenti audio

Le sorgenti audio presenti all'interno degli ambienti sviluppati sono state impostate con dei valori di intensità sonora di riferimento (distanza 1 m) riportati in Tab.4.1

SORGENTE AUDIO	INTENSITÀ (dB)
Pianoforte	68
Telefono	55
Radio	80
Proiettore	50
Ventilatore	35
Auto	83
Compressore	70
Uccellini	50
Sbarra	40

Tabella 4.1: Intensita'(dB SPL) utilizzate per le sorgenti audio all'interno degli ambienti VR.

4.2.2 Interfaccia e movimento

Normalmente, dispositivi come il mouse e/o la tastiera possono essere utilizzate come controller per tali test di valutazione. Tuttavia, l'esperimento viene svolto all'interno di una camera silente e l'utilizzo del HMD non rende facile usufruire di tali dispositivi da parte del soggetto. Il Samsung Gear VR mette a disposizione un controller mobile collegato tramite Bluetooth con lo smartphone utilizzato come display, ed è stato quindi configurato come dispositivo di interazione durante l'esperimento. Del controller in dotazione, illustrato in Fig.4.9, viene utilizzato il touchpad "a" per il movimento ed il pulsante a grilletto "b" per interagire con l'interfaccia.

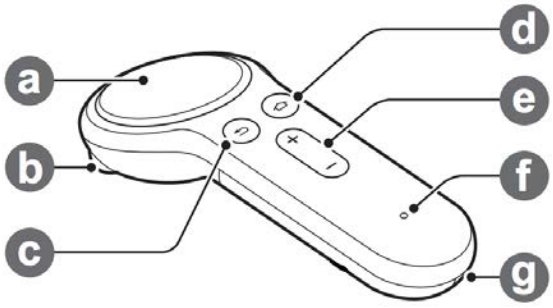


Figura 4.9: Controller per Samsung Gear VR

Poiché i partecipanti non saranno posizionati in un luogo statico all'interno dell' ambiente VR, l'interfaccia di controllo del test è implementata all'interno dell'ambiente

VR stesso, consentendo la piena libertà di movimento senza essere costretti a tornare in un luogo specifico per interagire con l'interfaccia dell'esperimento. L'interfaccia implementata è sempre in sovrimpressione durante la navigazione della scena ed è stata incorporata in modo totalmente minimale ed ininfluente per l'impatto visivo, come possiamo vedere in Fig.4.10. La composizione prevede un tasto unico centrale a forma di X posto sul lato alto della visuale e la cui funzione sarà quella della cancellazione del renderer scelto. I 6 tasti posti sul lato basso invece, nominati con una lettera dalla A alla F, identificano i render sottoposti alla prova. Il renderer attivo al momento è notificato in rosso mentre il renderer selezionato dal laser puntatore del telecomando è notificato in rosa. Per cancellare un renderer l'utente lo deve selezionare e renderlo attivo tramite la pressione del grilletto, poi premere il pulsante X in posto in alto che disabiliterà tale renderer facendolo scomparire anche dalla visuale. Dopodiché si attiverà automaticamente un altro renderer tra i disponibili, prontamente notificato in rosso. Per cambiare i renderer è possibile cambiare o con il puntatore laser e con il grilletto oppure con il tasto "SU" del touchpad è possibile fare un cambio sequenziale rapido.

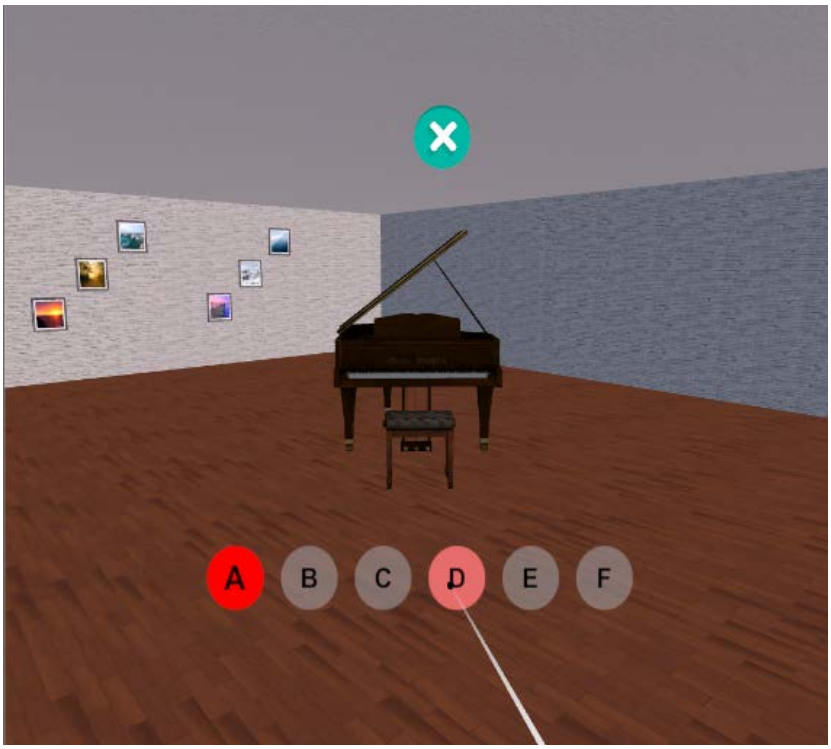


Figura 4.10: Visuale interfaccia integrata

Per quanto riguarda il movimento, è stato implementato un sistema di movimento basato sulla posizione rilevata sul touchpad. Questo permette quindi, con una velocità costante, di muoversi secondo le direzioni frontali e laterali. In più ovviamente, rilevando il movimento a 360° della testa, è possibile seguire la direzione frontale mirata.

4.2.3 Dizionario degli attributi utilizzato

La lista degli attributi utilizzati per la valutazione del sistema di riproduzione del suono binaurale e dell'acustica virtuale è basata sulle liste di attributi Simon (2016) [9] e Lindau (2014) [8].
Di seguito vengono elencati gli attributi con una breve descrizione:

- **REALISMO / NATURALEZZA**: come il suono sia il più possibile simile a quello che si sente nella realta;
- **IMMERSIONE**: percezione di "essere in scena", o "presenza spaziale". In altre parole, l'impressione di essere all'interno della scena virtuale o di essere integrati spazialmente nella scena virtuale;
- **ARTEFATTI**: individuazione di eventi sonori chiaramente indesiderati. Il render audio produce dei suoni distorti e/o rovinati che non corrispondono alla realtà, quindi innaturali;
- **RELIEF**: capacità di distinguere la distanza tra gli oggetti sonori più e gli oggetti sonori più lontani;
- **CHIAREZZA**: capacità di individuare distintamente le sorgenti sonore all'interno dell'ambiente virtuale.

Sono stati selezionati questi 5 attributi per rendere l'esperimento più scorrevole e non usare interamente i dizionari visti in precedenza, poichè sono molto più lunghi e articolati.

4.2.4 Renderer sottoposti all'analisi

Innanzitutto va spiegato che cosa si intende per HRTF manipolata o modificata nel nostro caso di studio. La modifica dell'ITD di una certa HRTF, che comporta la

sostituzione di tale grandezza con un altro ITD, si può apportare al set di HRTF desiderato, nel dominio delle frequenze, mediante la ricostruzione a fase minima e applicando poi il delay. In pratica si calcola l'ITD da applicare e lo si va a sommare poi convertendolo in fase alle HRTF da manipolare portate a fase minima. Nel nostro caso è stato scelto di combinare le risposte in frequenza del modello scelto tramite ITD visto in [23] e del modello estratto tramite i contorni dell'orecchio visto [44]. Nominando quindi con H_h l'HRTF del modello da cui prendere l'ITD e H_c l'HRTF basata sui contorni dell'orecchio, otteniamo quindi l'HRTF combinata nominata con H_s dalla seguente formula:

$$A_s(\omega) = \begin{cases} A_h(\omega) & \text{if } \omega < \omega_l \\ A_h(\omega) + \frac{A_c(\omega) - A_h(\omega)}{\omega_h - \omega_l}(\omega - \omega_l) & \text{if } \omega_l < \omega < \omega_h \\ A_c(\omega) & \text{if } \omega > \omega_h \end{cases} \tag{4.1}$$

dove $A_s(\omega) = \log|H_s(\omega)|$, $A_h(\omega) = \log|H_h(\omega)|$ e $A_c(\omega) = \log|H_c(\omega)|$. I valori di frequenza di soglia utilizzati sono $w_l = 250Hz$ e $w_h = 1000Hz$.
I 6 renderer che vengono sottoposti alla prova per ogni soggetto sono quindi:

1. **KEMAR**: viene utilizzato il file mit_kemar_large_pinna;
2. **CIPIC_ear**: soggetto individuato dalla selezione dell'orecchio;
3. **CIPIC_raggio_ottimo**: soggetto individuato dalla selezione del raggio ottimale;
4. **CIPIC_cross_correlation**: soggetto individuato dalla selezione dell'ITD calcolato con cross-correlation;
5. **CIPIC_ear & CIPIC_raggio_ottimo**: HRTF manipolata tra il soggetto CIPIC dell'orecchio e il soggetto CIPIC del raggio ottimale;
6. **CIPIC_ear & CIPIC_cross_correlation**: HRTF manipolata tra il soggetto CIPIC dell'orecchio e il soggetto CIPIC dell'ITD calcolato con cross-correlation.

Tra questi 6 render verrà stilata una classifica da ogni soggetto, per ogni attributo descrittivo e per ogni ambiente

4.3 Setup

Come visore per la realtà virtuale è stato scelto un Samsung Gear VR accoppiato ad un Samsung Galaxy S8. Tra i vari visori è stato scelto questo poichè tale strumento è stato fornito direttamente dal laboratorio di Informatica Musicale di Padova. Tale visore però non viene utilizzato come visore "stand-alone" ma viene utilizzato in parallelo ad un pc che si occupa della renderizzazione dell'ambiente. Inizialmente l'esperimento veniva sviluppato come un'applicazione nativa per Oculus / Samsung Gear VR ma si sono riscontrati poi dei problemi di gestione da parte del motore audio scelto (Steam Audio), che non veniva supportato pienamente nella versione mobile. È stata trovata una soluzione a questo problema andando a utilizzare il software VRidge 2.0 prodotto da RiftCat. Questa app abilita la tecnologia RiftCat VRidge su uno smartphone Android, trasformandolo in uno schermo per la realtà virtuale. Il sistema usa la potenza computazionale di un PC desktop per eseguire applicazioni di ambienti in VR per PC e poi ne effettua lo streaming sullo smartphone via WiFi. Questo permette quindi al nostro visore di simulare altri visori supportati dall'applicazione, nel caso nostro HTC Vive, implementato con SteamVR.
Per quanto riguarda la scelta del motore grafico la scelta è ricaduta su Unity Engine. Unity era già stato utilizzato in precedenza per un lavoro antecedente a questo (vedi sezione 3.1). Esso infatti permette di sviluppare direttamente applicazioni per varie piattaforme e nel nostro caso non si sono riscontrati problemi a dover passare dalla piattaforma Oculus / Android iniziale a quella successiva Windows.

4.3.1 Sistema

Per la parte di elaborazione foto e utilizzo del plugin Matlab è stato utilizzato un MacBook Pro (Retina, 13-inch, Early 2015) con le seguenti specifiche:

- OS: MacOS High Sierra 10.13.6;
- processore: Intel Core i5-5287U, dual-core 2,9 GHz;
- RAM: 16 GB, 1867 MHz DDR3;
- GPU: Intel Iris Graphics 6100, 1536 MB VR;
- SSD: 256 GB (scritt.: 640 MB/s, lett.: 1,3 GB/s).

Per quanto riguarda la parte grafica usata per renderizzare gli ambienti VR da trasmettere poi tramite collegamento wireless allo smartphone inserito nel visore Samsung Gear VR è stato utilizzato un PC con:

- OS: Windows 10 Home (64-bit);
- processore: Intel Core i7-3770k, quad-core 3,4 GHz;
- RAM: 8 GB, Kingston DDR3 1600 MHz;
- GPU: nVidia GeForce GTX-1060, VRAM 3 GB 1708 MHz GDDR5;
- HDD: WD (WD5000AAKX) Blu Hard Disk 500 GB, 7200 RPM, SATA 6 GB/s, 16 MB Cache, 3.5 ";

4.3.2 Strumenti

Oltre al già citato Samsung Gear VR (SM-R324NZAAITV) usato come visore, usato in coppia con uno smartphone Samsung Galaxy S8 ed il controller in dotazione, per completare l'headset sul soggetto come cuffie sono state utilizzate le Hefio ONE .
Per quanto riguarda l'uscita audio, è stata collegata una scheda audio esterna al PC e per svolgere tale compito è stata utilizzata una MOTU-896mk3.
Inoltre per la trasmissione del segnale video allo smartphone nel visore, è stato utilizzato un comune modem router che permette l'utilizzo di una rete WiFi su banda 5 GHz, poiché permette il raggiungimento di prestazioni migliori in fase di trasferimento dati.
Le foto utilizzate nel plugin Matlab, per l'acquisizione dei dati, sono state realizzate con un Samsung Galaxy S7.

4.3.3 Software

Di seguito vengono specificati tutti i software utilizzati per lo svolgimento di tale esperimento.
Le cuffie Hefio, che permettono un'ottimizzazione personale per offrire un suono fedele e preciso come in natura in condizioni acustiche ideali, calibrano la riproduzione sonora per l'acustica dei singoli canali auricolari misurando la risposta in frequenza

delle cuffie al timpano. Per tale scopo è stato utilizzato il software in dotazione di Hefio, su sistema Android. La risposta in frequenza calcolata e la relativa curva di equalizzazione elaborata sono state scaricate tramite il software Windows di Hefio e applicato tramite il software EqualizerAPO, come illustrato in Fig.4.11.

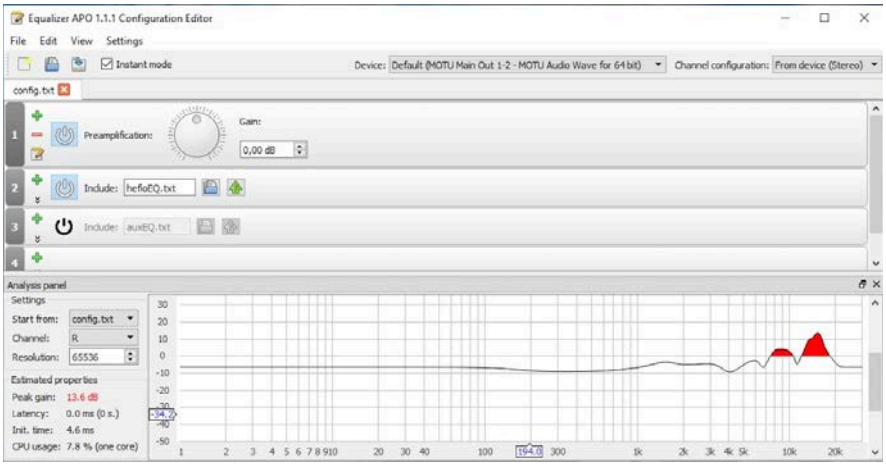


Figura 4.11: Schermata del software EqualizerAPO per applicare la curva personale generata dal software Hefio.

Per la trasmissione del segnale video è stato utilizzato il software Riftcat VRidge 2.0 per Windows 10 e la rispettiva applicazione sviluppata Oculus/Samsung GearVR. Questo ci permette di utilizzare il visore GearVR simulando un HTC Vive.
Come motore grafico (per la creazione di ambienti, interfaccia, implementazione esperimento) è stato utilizzato Unity nella versione 2018.3.0f2 (64-bit).
Per la parte audio, il motore audio utilizzato è Steam Audio nella versione 2.0.16 Beta.

4.3.4 Parametri

Specifichiamo ora i valori utilizzati per i parametri fondamentali durante lo svolgimento di questo esperimento.
Partendo dalla trasmissione video, per il software Riftcat VRidge 2.0 sono stati settati i valori di 40 Mbps , 60 FPS e l'impostazione di "previeni la perdita di frame", come vediamo in Fig.4.12.
Passiamo ora ai parametri utilizzati all'interno di Unity. Per quanto riguarda i parametri generale di progetto, sono stati utilizzati tutti i valori di default tranne per i parametri audio in cui alla voce *Spatializer Plugin* è stato ovviamente specificato

Figura 4.12: Schermata del software Vridge 2.0 per la trasmissione del segnale video al visore con le impostazioni indicate.

"Steam Audio Spatializer", mentre per la voce *DSP Buffer Size* è stato utilizzato il parametro "Good Latency". Va specificato che è stato condotto un test di latenza su tutto il setup per vedere effettivamente il tempo di risposta del nostra sistema montato. I dati raccolti mostrano che il setup utilizzato per gli esperimenti è soggetto ad una latenza pari a 90 *ms* circa. Come riportato in [45], la ricerca ha dimostrato che la presenza di una latenza significativa in un ambiente uditivo head-tracked ha un impatto sulle prestazioni di localizzazione e degrada la qualità dell'esperienza riportata. Va specificato però che ulteriori studi hanno suggerito che le latenze di $150 - 500$ *ms* in ambienti audio hanno un impatto minimo sulle prestazioni di localizzazione o sulla latenza percepita, quindi nel nostro caso può essere considerato un valore accettabile per il nostro scopo.
Specifichiamo ora invece i parametri di maggior interesse, ovvero quelli inerenti alle sorgenti audio negli ambienti: le componenti cui bisogna porre attenzione sono *Audio Source*, di base in Unity, e *Steam Audio Source*, che viene introdotta dal plugin di Valve. Per una maggior comprensione sono riportate delle figure illustrative, Fig.4.13 e Fig.4.14.
Per quanto riguarda la componente *Audio Source*, i parametri che sono stati modificati sono:

- **Spatial Blend**: valore impostato a 1 (3D), in modo che la spazializzazione del suono avvenga in tutti e 3 gli assi;
- **Doppler Level**: valore impostato a 0, per disattivare tale effetto;

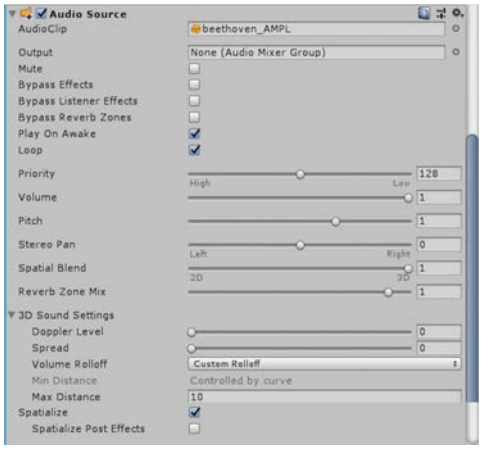


Figura 4.13: Impostazioni dei parametri della componente Audio Source

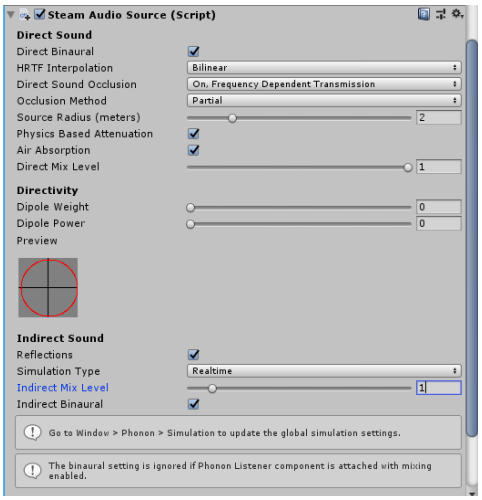


Figura 4.14: Impostazioni dei parametri della componente Steam Audio Source

- **Spread**: valore impostato a 0;
- **Volume Rolloff**: è stato specificato il valore "Custom Rolloff" e la curva di apprendimento è stata resa lineare e azzerata. Questo perchè (come suggerisce la documentazione di Steam) di adeguare i volumi in base alla distanza se ne occuperà il plugin Steam, come vedremo in seguito;
- **Spatialize**: campo attivato per abilitare la spazializzazione.

Per quanto riguarda la componente *Steam Audio Source*, i parametri che sono stati modificati sono:

- **Direct Binaural**: campo attivato per abilitare il renderer binaurale;
- **HRTF Interpolation**: è stato specificato il valore "Bilinear", per utilizzare l'interpolazione bilineare;
- **Direct Sound Occlusion**: è stato specificato il valore "On, Frequency Dependent Transmission";
- **Occlusion Method**: è stato specificato il valore "Partial", per l'effetto spiegato in precedenza;
- **Physics Based Attenuation**: campo attivato per abilitare l'attenuazione simile a quella reale (si occupa di regolare i volumi all'interno dell'ambiente);

- **Air Absorption**: campo attivato per abilitare assorbimento nell'aria del suono (si occupa di regolare i volumi all'interno dell'ambiente);
- **Reflections**: campo attivato per abilitare le riflessioni all'interno dell' amibiente virtuale;
- **Simulation Type**: è stato specificato il valore "Realtime", in modo tale che la simulazione venga fatta al momento e non precomputata;
- **Indirect Binaural**: campo attivato per abilitare il renderer binaurale anche con in suoni indiretti.

Le ultime modifiche apportate per la parte audio all'interno di Unity riguardano il *Steam Audio Manager*, di cui viene riportata una illustrazione in Fig.4.15. Oltre a specificare il numero di renderer da utilizzare nel campo *Size* ed i rispettivi nomi dei file SOFA che identificano tali renderer, il preset di simulazione è stato impostato ad "High" per ottenere una simulazione percettiva di alto livello più possibile simile a quella naturale. Per attivare tutte queste impostazioni va ricordato che bisogna pre-esportare la scena ed esportare tutti gli oggetti in scena collegati al plugin Steam tramite le apposite funzioni.

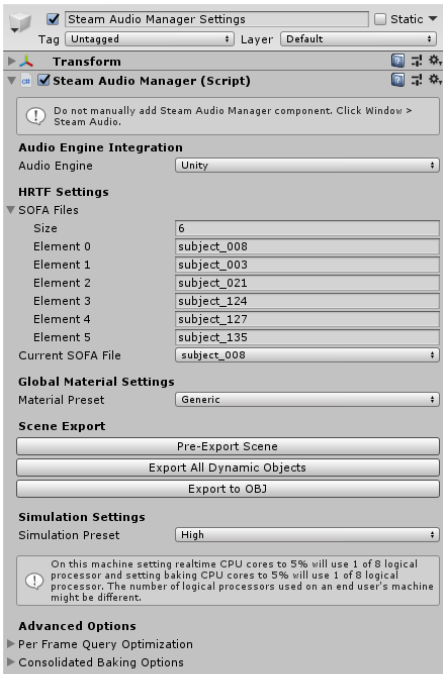


Figura 4.15: Impostazioni dei parametri di Steam Audio Manager

4.4 Protocollo sperimentale

L'esperimento è stato svolto in collaborazione con un altro tesista, Luca Buriola, che ha lavorato ad uno studio strettamente collegato [46] a quello qui presentato. Riprendendo il lavoro svolto in [27], lo scopo della tesi da lui sviluppata è concentrato sul creare un software per la selezione delle HRTF non individuali secondo le caratteristiche antropometriche dell'utente. Sono state apportate quindi delle modifiche al plugin sviluppato in Matlab, come illustrato in Fig.4.16, che permettono di tenere in considerazione altri fattori rilevanti per la selezione di una HRTF.

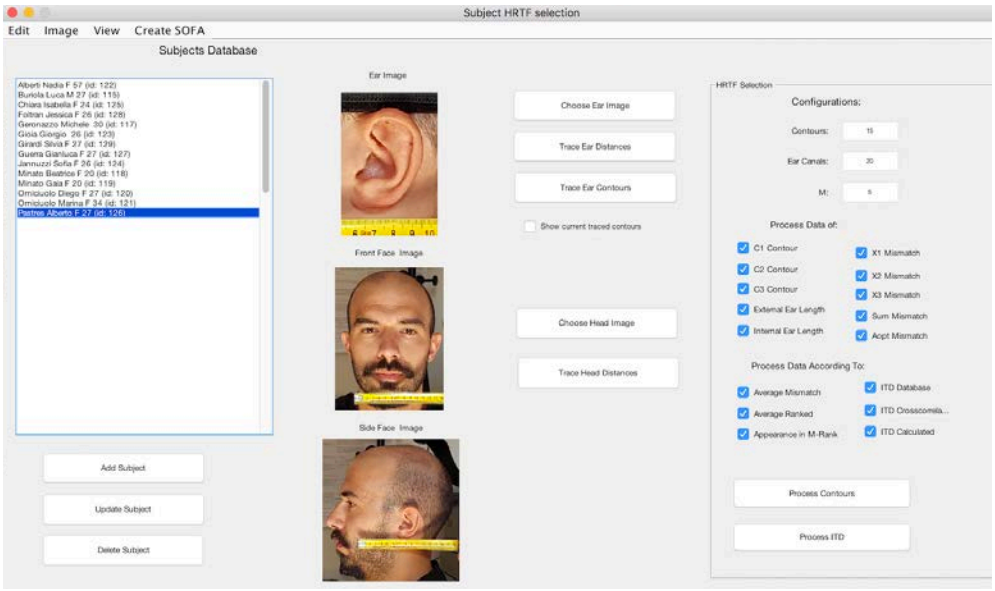


Figura 4.16: Gui modificata del plugin Matlab sviluppato

Oltre alla selezione già presente basata sui tracciamenti dei contorni dell'orecchio C_1, con le metriche implementate, sono state implementate le tecniche di selezione relative ad un set di HRTF generalizzate che si basano sulle misure antropometriche della testa del soggetto sperimentale. Le misure in questo caso sono X_1, X_2 e X_3 e la derivante misura del raggio ottimo calcolato con

$$a_{opt} = 0.44 \cdot X_1 + 0.23 \cdot X_3 + 3.2 \tag{4.2}$$

Inoltre è stato supportato il calcolo per metodi di estrazioni dell'ITD tramite cross-correlation [47] e tramite calcolo diretto del modello sferico [23]. Viene data la possi-

bilità di scegliere direttamente anche l'ITD presente nel database CIPIC.
L'ultima importante modifica apportata riguarda la generazione dei file SOFA. Si vuole generare un file che utilizzi le informazioni della pinna per la localizzazione verticale, e che utilizzi un ITD basato sulla similitudine antropometrica della testa per la localizzazione orizzontale. Viene usata come base l'HRTF del soggetto selezionato dalla metrica per i contorni dell'orecchio, andando a modificare il suo ITD. Si sceglie quindi il soggetto più simile sulla base antropometrica all'utente attraverso la metrica preferita e si estrae uno dei tre ITD pre-calcolati. A questo punto, l'ITD selezionato viene inserito nell'HRTF attraverso la tecnica spiegata in precedenza. Ottenuti i campioni finali di HRTF, viene generato il file SOFA utilizzando una funzione nella libreria[1] Matlab appositamente creata e messa a disposizione dal team di standardazzazione del formato SOFA.
Generati i file SOFA si passa ai test veri e propri di localizzazione e audio quality.

4.4.1 Test di localizzazione

In questo esperimento, svolto da Luca Buriola, è stato chiesto ai soggetti di localizzare delle sorgenti sonore provenienti da diverse direzioni. Il test è diviso in due macro parti. Nella prima parte (LONG) il soggetto deve individuare 73 posizioni. Questa operazione viene ripetuta 3 volte sia per il Render KEMAR, sia per il Renderer CIPIC_ear. La seconda parte (SHORT) prevede invece la localizzazione di 25 posizioni distinte. Questa operazione viene effettuata per ciascun renderer. I risultati vengono raccolti in appositi file *.txt* dove vengono salvati posizione di provenienza e posizione mirata.

4.4.2 Test di qualità dell'audio

Passiamo quindi alla spiegazione di come si svolge un test. All'inizio dell'esperimento, al soggetto che deve sottoporsi ai test, vengono scattate due foto (una frontale ed una laterale) che serviranno all'interno del plugin Matlab per l'acquisizione dei dati. Il processo è stato standardizzato per ogni soggetto in modo che i dati raccolti abbiano tutti la stessa validità. Le foto sono state scattate a 50 *cm* di distanza utilizzando un Samsung Galaxy S7 posizionato sopra un cavalletto trepiedi. Il soggetto, seduto

[1] https://www.sofaconventions.org/mediawiki/index.php/Software_and_APIs

su una sedia fissata, indossa un visore nel quale vede, per facilitare il corretto posizionamento della testa per lo scatto fotografico, un asse orizzontale ed un asse verticale. Una volta centrati gli assi grazie ad un mirino fisso, viene scattata la prima foto (laterale) valente per il tracciamento dei contorni dell'orecchio e per le misure antropometriche X_2 e X_3. Dopodiché ruotando di 90° a sinistra e ricentrando gli assi, si procede allo scatto frontale valente per la misura antropometrica X_1. A questo punto si passa al plugin Matlab dove viene creato il profilo del soggetto, inserite le foto e, una volta fissate le misure $Pixel-to-meter$ di riferimento per le foto, vengono tracciate le misure delle grandezze richieste (contorni e misure antropometriche). Si procede quindi all'elaborazione dei dati richiesti (metriche selezionate). Per ogni soggetto si può decidere se elaborare solo la parte relativa agli studi svolti sull'orecchio oppure quelli riguardanti l'ITD di un soggetto con le rispettive selezioni implementate nella GUI. Finito il processing è possibile osservare i risultati ottenuti tramite gli appositi grafici generati in cui vengono mostrati i soggetti CIPIC che più si avvicinano alle richieste dell'utente. È possibile dunque generare i file SOFA desiderati attraverso l'apposito menù. Sono stati generati per ciascun soggetto i file SOFA con le caratteristiche descritte in precedenza per il *Render 5* e il *Render 6*. Questo chiude la parte legata al plugin Matlab.
Si passa ora alla parte di realtà virtuale. La prima parte sperimentale è composta da test di localizzazione, di cui possiamo vedere la scena in Fig.4.17.

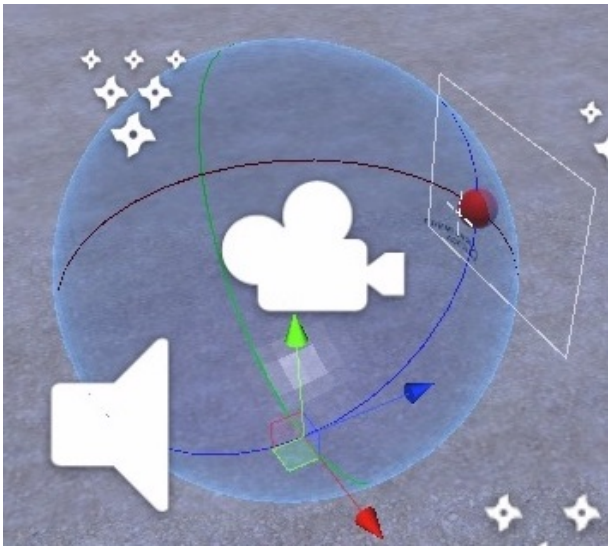


Figura 4.17: Scena Unity per il test di localizzazione

I 6 file SOFA che identificano i rispettivi diversi renderer vengono caricati nel progetto Unity. Nel frattempo il soggetto viene posto all'interno della camera silente, come riportato in Fig.4.18, istruito sul test e su come si svolge. Le cuffie Hefio che vengono fatte indossare sono state calibrate con la curva di equalizzazione personale del soggetto, applicando tale equalizzazione direttamente dal PC tramite il software dedicato. Per prendere confidenza con il setup al soggetto viene fatto affrontare un trial di apprendimento per la localizzazione. Dopo una leggera pausa comincia la prima parte di test LONG. Alla fine delle 3 prove (con delle piccole pause di defaticamento tra l'una e l'altra) il soggetto compila un questionario NASA che misura la prestazione umana in un contesto di alta automazioni o VR. Si procede poi in modo identico con il secondo renderer di prova. A questo punto l'esperimento viene fermato e vengono lasciati passare alcuni giorni che consentono di alleggerire il carico di sforzo sul soggetto e fa concede anche all'utente di non avere un apprendimento in fase di prestazione sul test di localizzazione.

Figura 4.18: Condizioni dell'esperimento: il soggetto si posiziona al centro della camera insonorizzata con la luce spenta per tutta la durata del test.

Il secondo giorno di test il soggetto riprende con i test di localizzazione SHORT con tutti e 6 i renderer. Viene compilato il questionario NASA nei casi rispettivi LONG. Si passa ora la parte di valutazione di audio quality.
La parte sperimentale di audio quality inizia anch'essa con un trial di apprendimento per il soggetto. Questo trial serve all'utente perché prenda confidenza innanzitutto

con un ambiente di realtà virtuale verosimile, con l'interfaccia creata per la selezione dei renderer e l'interazione con il controller per le gesture riguardo il movimento e la selezione. Al termine del trial, dopo una breve pausa, inizia il primo test di valutazione. All'interno del primo ambiente, libero di fare ogni movimento, al soggetto viene chiesto di valutare prima il REALISMO. Il soggetto prima ascolta tutti e 6 i renderer e poi andrà a comporre la classifica desiderata con il metodo di cancellazione. Al termine della valutazione dei renderer viene fatta una pausa di defaticamento. Di seguito in modo analogo vengono valutati IMMERSIONE, ARTEFATTI, RELIEF e CHIAREZZA ovviamente con prove distinte seguendo lo stesso ordine. Al termine delle prove del primo ambiente si passa successivamente alle prove del secondo e del terzo ambiente. Si ottengono così 5 classifiche distinte per ogni ambiente che valutano i 6 renderer per i 5 rispettivi attributi secondo il gradimento espresso dai soggetti.

5 Risultati

In questo capitolo verranno esposti ed analizzati i dati raccolti nel corso delle prove sperimentali effettuate dai soggetti.

5.1 Soggetti

Ai test hanno preso parte 15 soggetti (8 femmine, 7 maschi) di un'età compresa dai 20 ai 57 anni (età media di 29,3 anni con SD = 8,7). I soggetti per la maggior parte sono studenti universitari e/o lavoratori ma non sono esperti nel settore audio. Nessuno dei soggetti ha riportato problemi di udito noti che potessero interferire con lo svolgimento richiesto dalle prove ed inoltre il 45% dei partecipanti aveva già avuto esperienze di realtà virtuale in precedenza.
I soggetti, una volta posizionati all'interno della camera silente e con il setup pronto, dopo aver affrontato la parte relativa alla localizzazione sono stati sottoposti ai test di audio quality. Con il semplice ambiente di trial hanno preso confidenza, riuscendo a gestire il sistema messo disposizione. Durante la prova di valutazione è stato richiesto ai soggetti di muoversi liberamente all'interno dell'ambiente e di ascoltare tutti i renderer a disposizione prima di procedere con l'eliminazione.
I renderer messi a disposizione sono stati mantenuti fissi per tutte le prove nel seguente ordine:

- **A** → Kemar, nel dettaglio viene utilizzato il *mit_kemar_large_pinna.sofa*;
- **B** → Cipic_ear, derivante dalla selezione dell'orecchio;
- **C** → Cipic_aopt, derivante dalla selezione del raggio ottimo;
- **D** → Cipic_cross, derivante dalla selezione dell'ITD calcolato con cross-correlation;
- **E** → Ear_x_aopt, HRTF manipolata con le caratteristiche di B e C;
- **F** → Ear_x_cross, HRTF manipolata con le caratteristiche di B e D;

Per completare il test di audio quality un soggetto ha impiegato in media 45 minuti.

5.2 Risultati

Presentiamo ora i dati raccolti nelle varie prove sperimentali durante le giornate di test. I dati della classifica per eliminazione vengono analizzati con un metodo di valutazione a punteggio applicato in base alla posizione occupata al momento della cancellazione. I punti che vengono assegnati ai vari renderer sono

POSIZIONE	PUNTI ASSEGNATI
1°	6
2°	5
3°	4
4°	3
5°	2
6°	1

Tabella 5.1: Sistema di valutazione a punteggio assegnato per la classifica.

I dati sono stati sistemati in apposite tabelle che presentano le classifiche parziali risultanti dalla valutazione dei soggetti per ciascun attributo su ogni ambiente.

5.2.1 Analisi per ambiente

Iniziamo analizzando i dati raccolti nel primo ambiente. Come possiamo vedere dalla Tab.A.25 il renderer che riscontra un maggior gradimento nella valutazione di ciascun attributo è il renderer Ear_x_aopt tranne nel caso della valutazione dell'attributo "REALISMO" in cui il più valutato risulta il renderer Ear_x_cross. Di seguito troviamo il renderer Cipic_aopt e i renderer Cipic_cross separati da una leggera differenza. Infine troviamo i renderer Cipic_ear e Kemar che si distaccano in modo sensibile dagli altri casi.
Quindi andando ad esaminare i risultati parziali del primo ambiente derivanti dalla somma dei precedenti, come possiamo vedere nella Tab.A.26 ed il rispettivo grafico in Fig.A.1, il renderer Ear_x_aopt risulta essere il migliore e subito dopo troviamo il renderer Ear_x_cross, con una valutazione leggermente inferiore. Al terzo posto si classifica il renderer Cipic_aopt, poi il Cipic_cross ed il Cipic_ear, con una differenza equidistribuita. Infine ovviamente troviamo il renderer Kemar.
Passando all'analisi del secondo ambiente, troviamo una situazione del tutto analoga nelle classifiche che formano i vari renderer per i vari attributi, come possiamo vedere in Tab.A.28. Notiamo solamente che, tranne nel caso dell'attributo "CHIAREZZA", la differenza tra i renderer Ear_x_aopt ed Ear_x_cross è praticamente nulla nella valutazione degli attributi "REALISMO", "RELIF" e "IMMERSIONE". Per quanto riguarda le altre valutazione di classifica, le posizioni rimangono delineate allo stesso modo visto in precedenza. Ad ulteriore conferma possiamo vedere la Tab.A.28 e la Fig.A.2 che riportano la classifica parziale del secondo ambiente.
Vediamo infine i risultati derivanti dalle valutazioni fatte dai soggetti nel terzo ambiente. Ancora una volta troviamo una situazione del tutto simile alle due precedenti, dove troviamo solamente una differenza leggermente più marcata tra i renderer Ear_x_aopt ed Ear_x_cross nelle prime due posizioni, mentre si accorcia la distanza tra le posizioni 4 e 5 tra i renderer Cipic_cross e Cipic_ear. Le posizioni finali comunque alla fine rimangono invariate, come vediamo in Tab.A.30 e Fig.A.3, mantenendo così la stessa linea di gradimento vista per gli altri ambienti.
Per una maggiore caratterizzazione degli attributi, con i dati raccolti è stata fatto una analisi nel conteggio del renderer che si è classificato al primo posto per ogni attributo. L'andamento predominante rimane sempre simile alle classifiche generali appena viste con i renderer Ear_x_aopt e Ear_x_cross che riscontrano un maggior gradimento

mentre i renderer Cipic_ear e Kemar non vengono mai considerati come migliori. Notiamo però che negli ambienti interni, ovvero Ambiente 1 e Ambiente 2, per gli attributi "REALISMO" e "IMMERSIONE" i renderer Cipic_aopt e Cipic_cross si collocano al livello dei migliori renderer Ear_x_aopt e Ear_x_cross, come possiamo vedere nei grafici in Fig.A.4, Fig.A.5, Fig.A.8 e Fig.A.9. Andando a confrontare questi risultati con quelli ottenuti nel test di localizzazione in [46], questa classificazione per i renderer Cipic_aopt e Cipic_cross è stata effettuata da soggetti che corrispondono ai soggetti con un elevato errore di localizzazione di una sorgente sonora sul piano verticale e/o azimuthale. Gli ambienti 1 e 2, essendo ambienti interni e quindi soggetti a riflessione, tendono così a dare un'impressione alterata e questi soggetti quindi non distinguono in modo adeguato la spazializzazione del suono, accettando quindi anche un renderer meno definito. Nel terzo ambiente invece questo non succede poiché essendo esterno non è soggetto a riflessioni.

5.2.2 Discussione

Innanzitutto, come si nota dalla sottosezione precedente, gli attributi valutati alla fine sono stati ridotti a 4: infatti l'attributo "ARTEFATTI" non è stato incluso nell'analisi poiché i soggetti, durante le sessioni di test, non hanno mai riscontrato degli artefatti prodotti dai renderer. Durante la valutazione di questo attributo è stato chiesto di posizionarsi possibilmente in posti con molte e forti riflessioni, di compiere movimenti rapidi e di cambiare i renderer in movimento, senza produrre però alcun effetto. Probabilmente questo può dipendere dal fatto che gli utenti non sono esperti nel settore audio e quindi tendono a non rilevare delle sottili imperfezioni prodotte renderer. Sicuramente questa risulta essere comunque un'ottima notizia per i SOFA utilizzati provenienti dai dataset CIPIC e KEMAR e soprattutto per i SOFA prodotti tramite il plugin Matlab per dare vita alle HRTF manipolate combinando due renderer risultanti dalle metriche di selezione che abbiamo visto: questo sottolinea l'assoluta affidabilità della qualità dei SOFA combinata con il motore audio poiché i file utilizzati non producono dei suoni distorti e/o rovinati che non corrispondono alla realtà. Un'altra nota da riportare, di un comportamento che si è verificato durante i test, è la difficoltà riscontrata dai soggetti nel trovare una differenza tra i renderer E ed F, ovvero i due renderer prodotti tramite il software Matlab derivanti dalla manipolazione di HRTF vista nella sottosezione 4.2.4. Come possiamo vedere dalla formula

4.1, i campioni di una HRTF che effettivamente vengono modificati risultano essere una minima parte mentre il vero cambiamento è l'introduzione dell'ITD, che risulta essere quasi del tutto simile con le HRTF selezionate dalle due metriche utilizzate per la scelta del soggetto CIPIC da usare per i renderer C e D. Questo giustifica il comportamento dei soggetti che durante le sessioni di test, qualora si imbattessero nell'eliminazione di uno dei renderer che coinvolgevano le HRTF manipolate, di conseguenza eliminavano entrambi e sottolinea le conseguenti posizioni ravvicinate nelle classifiche prodotte con uno scarto di punteggio così ravvicinato.
Come ultima nota sul comportamento osservato dei soggetti durante le fasi di test che hanno portato a queste valutazioni, si fa presente che un soggetto solitamente tende a stilare una classifica sempre simile nella valutazione dei diversi attributi all'interno dei vari ambienti. Tranne alcuni casi, i soggetti identificano in tutte le classifiche lo stesso renderer peggiore e lo stesso renderer migliore, con una possibile leggera variazione dei renderer che occupano le posizioni di mezzo.
Vediamo ora quindi i risultati finali prodotti dalla valutazione dei renderer per ciascun attributo nei 3 ambienti di realtà virtuale. La classifica finale viene riportata in Tab.5.2 ed il conseguente grafico illustrato in Fig.5.1.
Il renderer che riporta la peggior valutazione è il renderer Kemar. All'unanimità dei soggetti in ogni classifica è stato sempre il primo eliminato poiché ha una qualità troppo inferiore per le caratteristiche richieste in questo tipo di analisi, soprattutto nella valutazione degli attributi nell'ambiente 3 che rappresenta una scena esterna: questo renderer infatti è stato definito troppo piatto per poter distinguere in modo nitido le varie fonti audio e le loro distanze e manca di tono e colorazione per quanto riguarda la valutazione di attributi come "REALISMO" e "IMMERSIONE".
Successivamente troviamo il renderer Cipic_ear, che identifica il file SOFA che rappresenta il soggetto CIPIC selezionato dalla similitudine del contorno C_1 dell'orecchio. Nonostante l'effettivo miglioramento dal punto di visto qualitativo, nella maggior parte dei casi nella valutazione degli attributi, soprattutto negli ambienti 2 e 3 non offre un'adeguata spazializzazione dei suoni prodotti dalle sorgenti audio presenti negli ambienti: infatti nella valutazione degli attributi "RELIEF" e "CHIAREZZA" non era semplice il poter rapportare le giuste distanze tra i suoni e poterle identificare in modo nitido all'interno degli ambienti.
I renderer che sono rimasti (Cipic_aopt, Cipic_cross, Ear_x_aopt ed Ear_x_cross) va sottolineato che sono renderer che hanno riscosso un livello di attenzione mag-

giore nella valutazione, raggiungendo ciascuno la prima posizione nelle classifiche parziali stilate dai soggetti, chi più frequentemente chi meno. Questo avviene grazie alla personalizzazione delle HRTF basata sulle misure antropometriche della testa per la localizzazione orizzontale del suono, che è più facilmente utilizzabile anche da utenti meno esperti. Il successivo posto nella classifica viene occupato dal renderer Cipic_cross, che identifica il file SOFA associato al soggeto CIPIC derivante dalla selezione dell'ITD calcolcato con la tecnica della cross-correlation. Al terzo posto troviamo il renderer Cipic_aopt, che identifica il file SOFA del soggetto CIPIC selezionato dalla metrica delle misure antropometriche con raggio ottimo più simile al soggettp testato. I renderer Cipic_cross e Cipic_aopt, con un gradimento maggiore per quest'ultimo, offrono una buona spazializzazione e renderizzazione delle fonti audio negli ambienti interni (Ambiente 1 e 2), perdendo però di qualità nell'ambiente esterno.
Infine troviamo i renderer Ear_x_cross ed Ear_x_aopt, rispettivamente secondo e primo nella valutazione finale. Questi due renderer, che identificano i file SOFA prodotti con il plugin Matlab, sono stati valutati dai soggetti come i migliori, offrendo la miglior spazializzazione e la miglior renderizzazione per gli attributi percettivi valutati, soprattutto per quanto riguarda il terzo ambiente e gli attributi "RELIEF" e "CHIAREZZA", in cui era possibile distinguere in modo ottimo le fonti audio e collocandole ad una distanza adeguata secondo il suono percepito. Nella maggior parte dei casi di soggetti che avevano preferito un renderer antropometrico (Cipic_aopt o Cipic_cross) nei primi due ambienti, la valutazione migliore nel terzo ambiente è ricaduta nei renderer renderer Ear_x_cross o Ear_x_aopt per i motivi sopracitati. Quindi, alla luce dei risultati ottenuti, possiamo affermare che in un ambiente di realtà virtuale, il renderer audio binaurale che offre una migliore qualità a livello percettivo e che più si avvicina alla realtà è quello offerto dall'utilizzo di una HRTF che combina i campioni del soggetto CIPIC individuato dalla selezione dell'orecchio e il soggetto CIPIC con un raggio ottimale più simile al soggetto testato.

TOTALE AMBIENTI 1 - 2 - 3		
RENDERER	**PUNTI**	**CLASS.**
A	180	**6**
B	498	**5**
C	735	**3**
D	617	**4**
E	886	**1**
F	864	**2**

Tabella 5.2: Classifica finale di tutti gli ambienti valutati nel corso dei test.

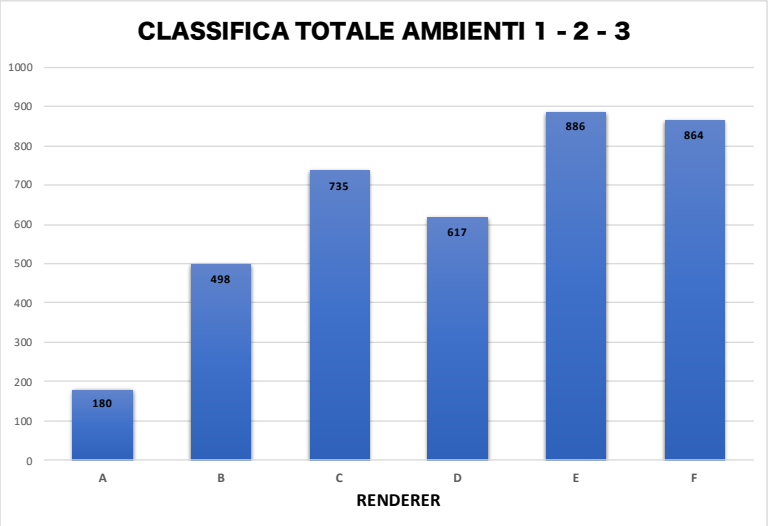


Figura 5.1: Grafico della classifica finale dei tre ambienti

6
Conclusioni

In questa tesi è stato proposto un metodo di valutazione di HRTF basato su una classificazione per eliminazione come in [12] per un confronto di HRTF. La valutazione non viene effettuata a livello generale ma su un numero limitato di attributi percettivi prestabiliti derivanti dai dizionari SAQI [8] e Lindau [9]. Le HRTF messe a confronto sono il risultato di una elaborazione di un software Matalb sviluppato per assegnare all'utente delle HRTF personalizzate su diverse metriche di selezione (contorno orecchio, misure antropometriche e calcolo dell'ITD).
La valutazione effettuata dai soggetti sui diversi attributi percettivi sottoposti all'analisi avviene all'interno di 3 diversi ambienti di realtà virtuale immersivi con caratteristiche di base diverse, ovvero ambiente interno o esterno e la presenza di una o più sorgenti. Per effettuare tale classificazione per eliminazione è stata implementata un'interfaccia che potesse permettere all'utente di poter cambiare in tempo reale i vari renderer messi a disposizione mantenendo inalterata la posizione corrente negli ambienti.
Dai risultati derivanti dai dati raccolti nelle sessioni di test, è stato dimostrato che le HRTF cui viene introdotto l'ITD e combinano i campioni di due HRTF personalizzate migliorano la percezione uditiva all'interno di ambiente di realtà virtuale, ricreando un udito quanto più simile alla condizione di ascolto individuale. Tali renderer sono stati i più valutati dai soggetti tra i renderer messi a disposizione. Sono state riscontrate delle differenze con i renderer che utilizzano i file SOFA del dataset KEMAR e

CIPIC che venivano utilizzati in [27].
In conclusione, i risultati indicano che la qualità audio per il VR unita ad i risultati derivanti dal test di localizzazione svolto in parallelo ottengono i migliori risultati con una HRTF manipolata cui viene introdotto l'ITD che più si avvicina a quello soggettivo.
Lavori futuri sicuramente devono concentrarsi sul fatto di portare altri confronti di questo tipo poiché in letteratura sul campo qualitativo ci sono ancora molte lacune, con la possibilità di diversificare i renderer cui sottoporre questo tipo di test. Inoltre il questionario qualitativo può essere esteso prevedendo l'introduzione di nuovi attributi percettivi da valutare.

A

Appendice A

DATI BENCHMARCK AMBIENTE 1

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	11,5	12,7	12,8	17
Tot Audio Cpu (%)	0,77 ± 0,01	2,11 ± 0,13	3,15 ± 0,11	5,00 ± 0,06
Tot Memoria Allocata (MB)	358 ± 0,52	375,71 ± 0,87	396,45,45 ± 0,50	407,91 ± 0,06
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,7

Tabella A.1: Caso Steam Audio 2.0.16 con "Nearest" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	12,2	13,6	14	22
Tot Audio Cpu (%)	0,91 ± 0,01	2,75 ± 0,09	4,84 ± 1,59	19,63 ± 4,22
Tot Memoria Allocata (MB)	409,22 ± 1,53	431,61 ± 4,88	459,43 ± 0,58	474,81 ± 1,27
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.2: Caso plugin sviluppato con "Nearest" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	13,8	15,9	22,3	30,3
Tot Audio Cpu (%)	1,59± 0,02	4,71 ± 0,16	13,82 ± 0,23	28,30 ± 0,33
Tot Memoria Allocata (MB)	424 ± 0,64	489,39 ± 0,92	572,1 ± 1,61	702,43 ± 1,26
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,7

Tabella A.3: Caso Steam Audio 2.0.16 con "Nearest + riflessioni" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	15,1	17,7	25,2	35,5
Tot Audio Cpu (%)	2,01 ± 0,21	5,35 ± 0,79	16,44 ± 1,49	32,51 ± 5,72
Tot Memoria Allocata (MB)	473,11 ± 1,03	544,31 ± 1,38	641,93 ± 1,22	769,59 ± 1,55
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.4: Caso plugin sviluppato con "Nearest + riflessioni" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	19,7	23,4	26	27,1
Tot Audio Cpu (%)	0,81 ± 0,01	2,78 ± 0,08	4,00 ± 0,04	5,08 ± 0,05
Tot Memoria Allocata (MB)	466,78 ± 2,37	476,92 ± 3,84	487,27 ± 3,64	503,26 ± 4,33
Tot Audio Memoria (MB)	122,7	201,7	351,1	423,4

Tabella A.5: Caso Steam Audio 2.0.16 con "Bilinear" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	20,8	24,7	27,2	28,1
Tot Audio Cpu (%)	0,98 ± 0,03	2,87 ± 0,02	5,86 ± 0,11	22,79 ± 3,39
Tot Memoria Allocata (MB)	503,22 ± 1,34	519,32 ± 2,87	535,81 ± 3,22	552,32 ± 3,10
Tot Audio Memoria (MB)	136,3	245,4	383,3	459,1

Tabella A.6: Caso plugin sviluppato con "Delaunay" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	24,5	29,5	40,8	51,9
Tot Audio Cpu (%)	4,2 ± 0,04	20,43 ± 0,67	38,26 ± 0,66	76,92 ± 4,25
Tot Memoria Allocata (MB)	629,14 ± 1,45	691,03 ± 1,87	832,03 ± 2,19	967,22 ± 3,14
Tot Audio Memoria (MB)	173,5	267,6	485,1	710,0

Tabella A.7: Caso Steam Audio 2.0.16 con "Bilinear + riflessioni" nell'ambiente 1 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	26,8	33,7	45,3	56,1
Tot Audio Cpu (%)	4,78 ± 0,20	21,78 ± 0,83	40,14 ± 1,91	87,07 ± 6, 78
Tot Memoria Allocata (MB)	655,56 ± 1,73	718,34 ± 2,34	857,11 ± 3,21	1088,22 ± 4,21
Tot Audio Memoria (MB)	190,0	349,3	521,8	710,0

Tabella A.8: Caso plugin sviluppato con "Delaunay + riflessioni" nell'ambiente 1 di benchmark.

DATI BENCHMARCK AMBIENTE 2

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	20,2	22,9	25,6	26,5
Tot Audio Cpu (%)	0,65 ± 0,05	1,64 ± 0,02	2,88 ± 0,03	5,23 ± 0,06
Tot Memoria Allocata (MB)	281,41 ± 1,87	291,23 ± 2,35	305,73 ± 3,87	323,6 ± 3,57
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.9: Caso Steam Audio 2.0.16 con "Nearest" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	20,8	23,4	26,1	26,7
Tot Audio Cpu (%)	1,57 ± 0,02	2,08 ± 0,03	4,91 ± 0,06	8,03 ± 0,12
Tot Memoria Allocata (MB)	363,23 ± 1, 25	378,44 ± 1,85	389,33 ± 2,31	396,82 ± 2,56
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,7

Tabella A.10: Caso plugin sviluppato con "Nearest" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	21,5	24,3	29,2	39,8
Tot Audio Cpu (%)	0,95 ± 0,13	11,52 ± 0,09	25,9 ± 0,05	56,73 ± 0,11
Tot Memoria Allocata (MB)	443,1 ± 1,31	502,91 ± 1,95	587,75 ± 2,67	692,33 ± 3,37
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.11: Caso Steam Audio 2.0.16 con "Nearest + riflessioni" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	22,8	26,4	31,4	42,7
Tot Audio Cpu (%)	1,73 ± 0,12	14,38 ± 0,11	29,9 ± 0,26	61,43 ± 0,22
Tot Memoria Allocata (MB)	513,49 ± 1, 82	588,18 ± 2,54	669,63 ± 2,81	761,82 ± 3,16
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,7

Tabella A.12: Caso plugin sviluppato con "Nearest + riflessioni" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	20,5	22,7	26,8	27,3
Tot Audio Cpu (%)	0,88 ± 0,02	2,08 ± 0,02	3,76 ± 0,03	6,97 ± 0,12
Tot Memoria Allocata (MB)	372,45 ± 1,89	494,38 ± 2,48	586,97 ± 2,94	642,37 ± 3,25
Tot Audio Memoria (MB)	131,3	178,3	281,3	371,3

Tabella A.13: Caso Steam Audio 2.0.16 con "Bilinear" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	21,5	23,6	27,4	28,1
Tot Audio Cpu (%)	1,88 ± 0,05	4,82 ± 0,19	6,73 ± 0,06	8,52 ± 0,07
Tot Memoria Allocata (MB)	500,71 ± 1,21	637,61 ± 1,22	684,36 ± 1,78	715,61 ± 2,75
Tot Audio Memoria (MB)	147,7	199,3	303,3	450,3

Tabella A.14: Caso plugin sviluppato con "Delaunay" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	23,8	33,9	41	48,7
Tot Audio Cpu (%)	4,98 ± 0,03	22,76 ± 1,72	39,13 ± 2,8	86,7 ± 7,57
Tot Memoria Allocata (MB)	718,32 ± 2,56	822,45 ± 3,40	915,87 ± 3,56	1035,82 ± 4,88
Tot Audio Memoria (MB)	160,0	241,0	417,0	710,0

Tabella A.15: Caso Steam Audio 2.0.16 con "Bilinear + riflessioni" nell'ambiente 2 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	24,5	34,5	43,8	51
Tot Audio Cpu (%)	6,20 ± 0,32	25,29 ± 0,98	43,88 ± 1,56	88,38 ± 9,32
Tot Memoria Allocata (MB)	729,91 ± 1,86	889,31 ± 2,99	1003,77 ± 3,16	1167,45 ± 4,87
Tot Audio Memoria (MB)	186,90	285,0	503,0	710,0

Tabella A.16: Caso plugin sviluppato con "Delaunay + riflessioni" nell'ambiente 2 di benchmark.

DATI BENCHMARCK AMBIENTE 3

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	31,8	33,2	34,1	35,7
Tot Audio Cpu (%)	0,68 ± 0,01	1,62 ± 0,02	2,97 ± 0,03	5,33 ± 0,07
Tot Memoria Allocata (MB)	997,48 ± 3,67	1016,45 ± 2,88	1069,39 ± 2,91	1091,21 ± 3,21
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.17: Caso Steam Audio 2.0.16 con "Nearest" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	32,8	34,0	35,6	37,2
Tot Audio Cpu (%)	1,85 ± 0,02	2,45 ± 0,04	4,41 ± 0,10	19,7 ± 0,12
Tot Memoria Allocata (MB)	1026,33 ± 3,67	1068 ± 2,91	1094,55 ± 3,33	1204,55 ± 4,21
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.18: Caso plugin sviluppato con "Nearest" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	35,8	39,1	44,3	51,1
Tot Audio Cpu (%)	0,93 ± 0,08	9,18 ± 0,09	21,12 ± 0,05	49,70 ± 0,17
Tot Memoria Allocata (MB)	1114,82 ± 4,21	1265,41 ± 5,63	1408 ± 5,95	1491,22 ± 6,43
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.19: Caso Steam Audio 2.0.16 con "Nearest + riflessioni" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	38,7	42,8	48,4	55,3
Tot Audio Cpu (%)	2,51 ± 0,12	15,11 ± 2,14	28,58 ± 2,54	54,7 ± 3,92
Tot Memoria Allocata (MB)	1196,28 ± 4,31	1387,71 ± 5,49	1623,39 ± 4,98	1741,21 ± 6,77
Tot Audio Memoria (MB)	39,6	39,6	39,6	39,6

Tabella A.20: Caso plugin sviluppato con "Nearest + riflessioni" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	32,5	33,6	35,2	37,7
Tot Audio Cpu (%)	0,75 ± 0,01	4,12 ± 0,04	7,92 ± 0,43	29,76 ± 0,58
Tot Memoria Allocata (MB)	1103,66 ± 1,27	1137,15 ± 1,66	1164,15 ± 2,71	1202,01 ± 3,01
Tot Audio Memoria (MB)	179,7	267,8	378,8	710,0

Tabella A.21: Caso Steam Audio 2.0.16 con "Bilinear" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	35,2	36,1	38,2	39,6
Tot Audio Cpu (%)	1,9 ± 0,03	3,45 ± 0,05	7,05 ± 0,12	19,7 ± 0,23
Tot Memoria Allocata (MB)	1139,81 ± 1,98	1232,67 ± 2,51	1298,29 ± 2,78	1338,21 ± 3,22
Tot Audio Memoria (MB)	200,9	288,8	405,8	710,0

Tabella A.22: Caso plugin sviluppato con "Delaunay" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	41,3	43,9	50,7	56,8
Tot Audio Cpu (%)	4,07 ± 0,09	21,67 ± 2,07	40,97 ± 2,42	84,98 ± 7,31
Tot Memoria Allocata (MB)	1385,26 ± 1,83	1545,71 ± 1,91	1675,71 ± 2,53	1746,39 ± 3,91
Tot Audio Memoria (MB)	210,0	304,9	472,4	710,0

Tabella A.23: Caso Steam Audio 2.0.16 con "Bilinear + riflessioni" nell'ambiente 3 di benchmark.

	1 sorgente	5 sorgenti	10 sorgenti	20 sorgenti
Tot Cpu utilizzata (%)	45,3	48,9	53,7	59,2
Tot Audio Cpu (%)	5,24 ± 0,13	24,29 ± 4,29	46,80 ± 3,78	91,7 ± 9,61
Tot Memoria Allocata (MB)	1641,77 ± 2,36	1788,33 ± 3,16	1932,26 ± 3,72	2045,33 ± 5,31
Tot Audio Memoria (MB)	229,0	348,0	510,4	710,0

Tabella A.24: Caso plugin sviluppato con "Delaunay + riflessioni" nell'ambiente 3 di benchmark.

REALISMO	
A	15
B	41
C	60
D	54
E	70
F	75

IMMERSIONE	
A	15
B	42
C	65
D	48
E	74
F	71

RELIEF	
A	15
B	42
C	58
D	53
E	76
F	71

CHIAREZZA	
A	15
B	37
C	62
D	54
E	75
F	72

Tabella A.25: Classifica risultante per ciascun attributo valutato nel primo ambiente.

TOTALE AMBIENTE 1		
RENDERER	PUNTI	CLASS.
A	60	**6**
B	162	**5**
C	245	**3**
D	209	**4**
E	295	**1**
F	289	**2**

Tabella A.26: Classifica complessiva di tutti gli attributi nel primo ambiente.

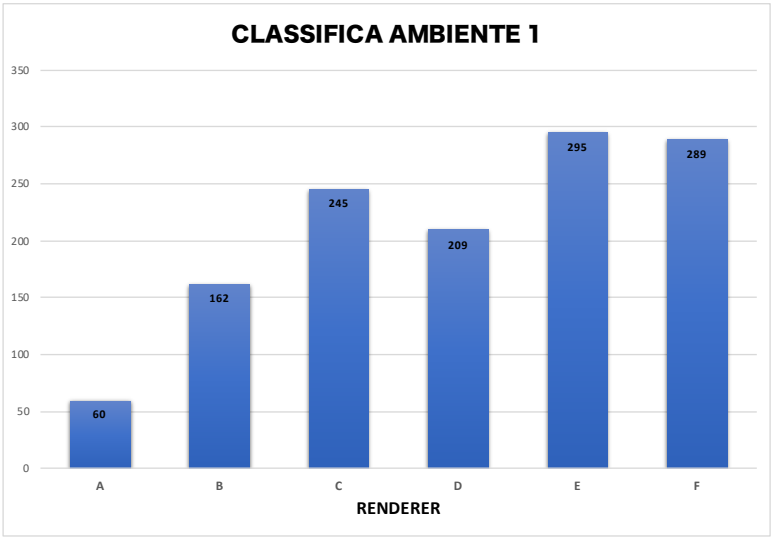


Figura A.1: Grafico della classifica complessiva del primo ambiente

REALISMO	
A	15
B	42
C	63
D	50
E	72
F	73

IMMERSIONE	
A	15
B	37
C	61
D	57
E	72
F	73

RELIEF	
A	15
B	42
C	61
D	52
E	73
F	72

CHIAREZZA	
A	15
B	44
C	63
D	52
E	73
F	68

Tabella A.27: Classifica risultante per ciascun attributo valutato nel secondo ambiente.

TOTALE AMBIENTE 2		
RENDERER	PUNTI	CLASS.
A	60	6
B	165	5
C	248	3
D	211	4
E	290	1
F	286	2

Tabella A.28: Classifica complessiva di tutti gli attributi nel secondo ambiente.

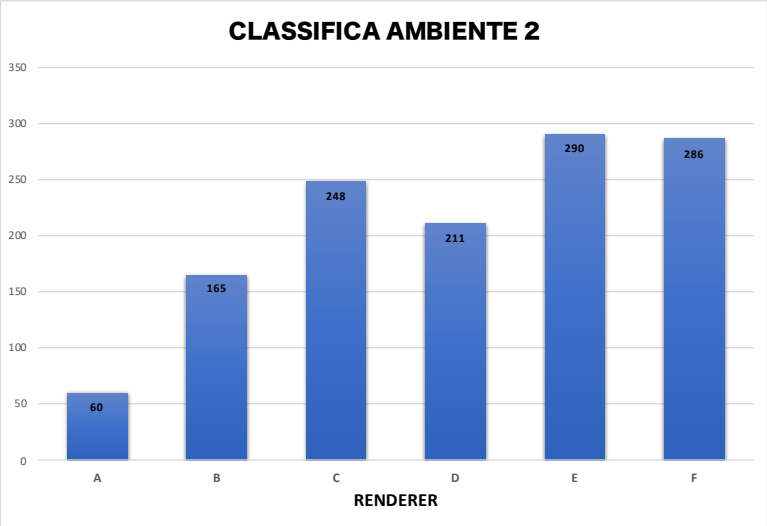


Figura A.2: Grafico della classifica complessiva del secondo ambiente

REALISMO	
A	15
B	46
C	59
D	54
E	71
F	70

IMMERSIONE	
A	15
B	43
C	63
D	49
E	75
F	70

RELIEF	
A	15
B	41
C	58
D	46
E	79
F	76

CHIAREZZA	
A	15
B	41
C	62
D	48
E	76
F	73

Tabella A.29: Classifica risultante per ciascun attributo valutato nel terzo ambiente.

TOTALE AMBIENTE 3		
RENDERER	PUNTI	CLASS.
A	60	6
B	171	5
C	242	3
D	197	4
E	301	1
F	289	2

Tabella A.30: Classifica complessiva di tutti gli attributi nel terzo ambiente.

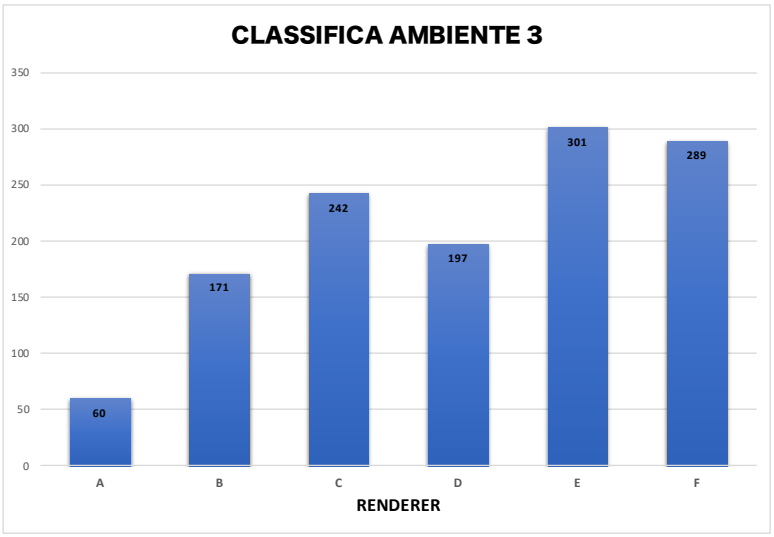


Figura A.3: Grafico della classifica complessiva del terzo ambiente

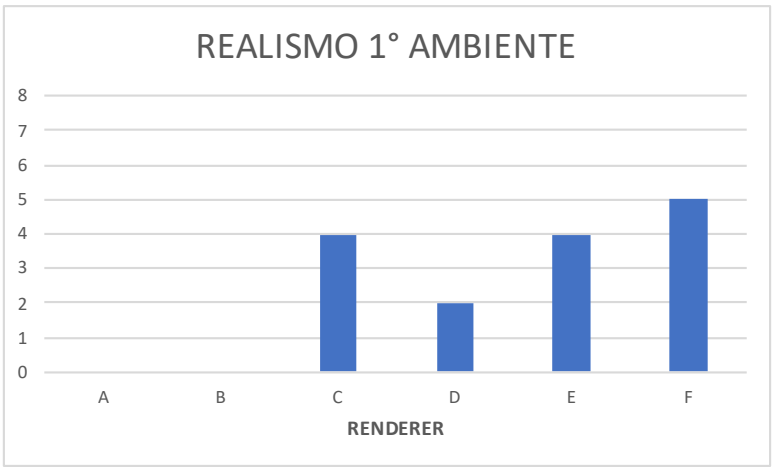


Figura A.4: Grafico della classifica del primo posto REALISMO del primo ambiente

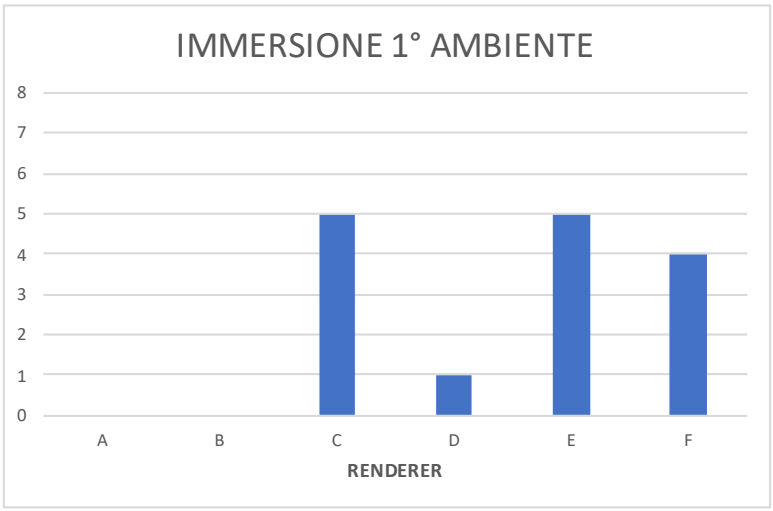


Figura A.5: Grafico della classifica del primo posto IMERSIONE del primo ambiente

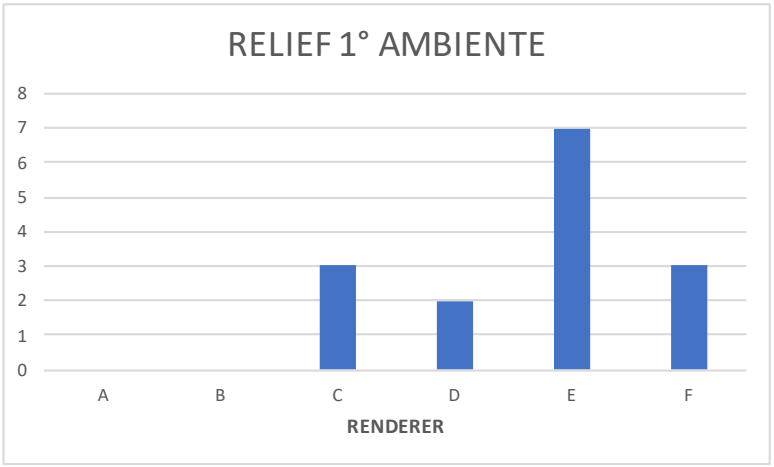


Figura A.6: Grafico della classifica del primo posto RELIEF del primo ambiente

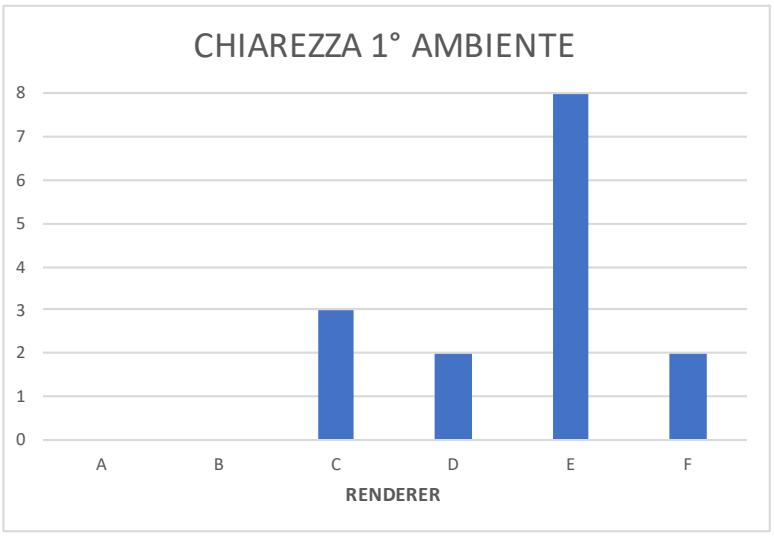


Figura A.7: Grafico della classifica del primo posto CHIAREZZA del primo ambiente

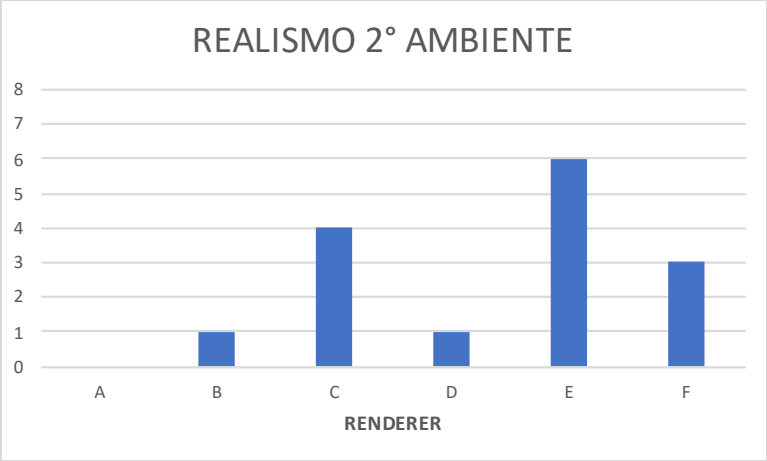


Figura A.8: Grafico della classifica del primo posto REALISMO del secondo ambiente

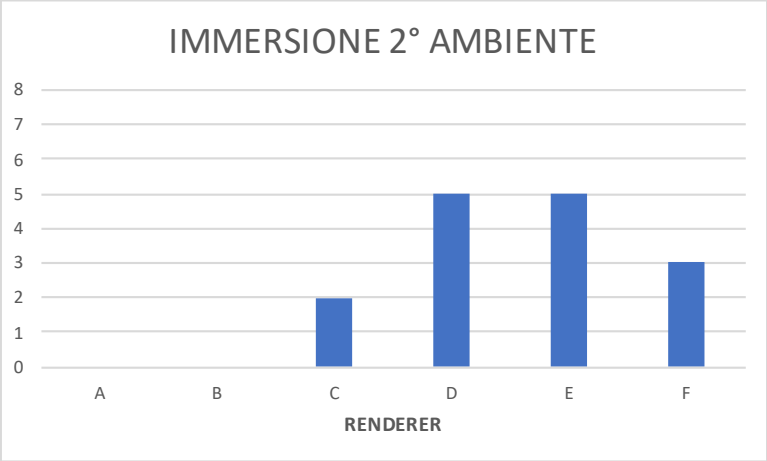


Figura A.9: Grafico della classifica del primo posto IMMERSIONE del secondo ambiente

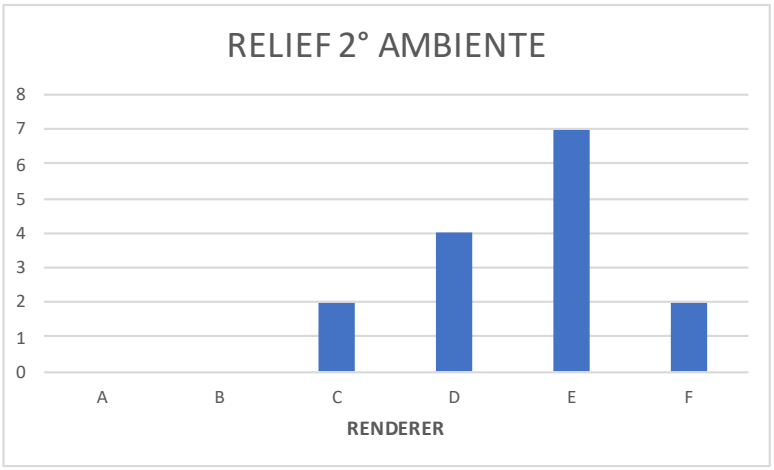


Figura A.10: Grafico della classifica del primo posto RELIEF del secondo ambiente

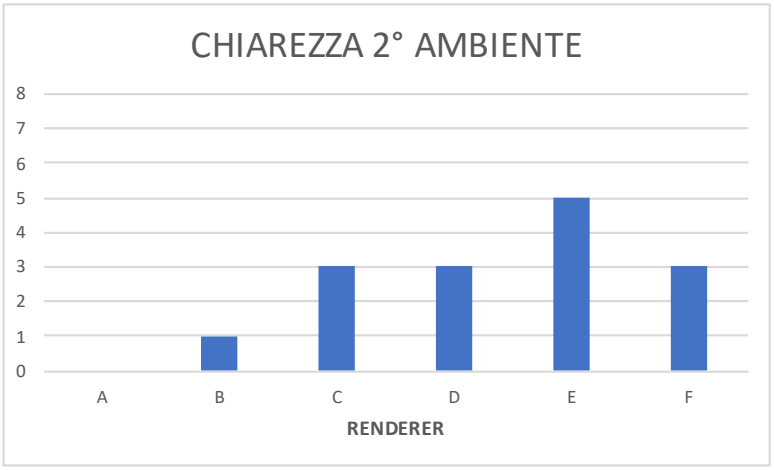


Figura A.11: Grafico della classifica del primo posto CHIAREZZA del secondo ambiente

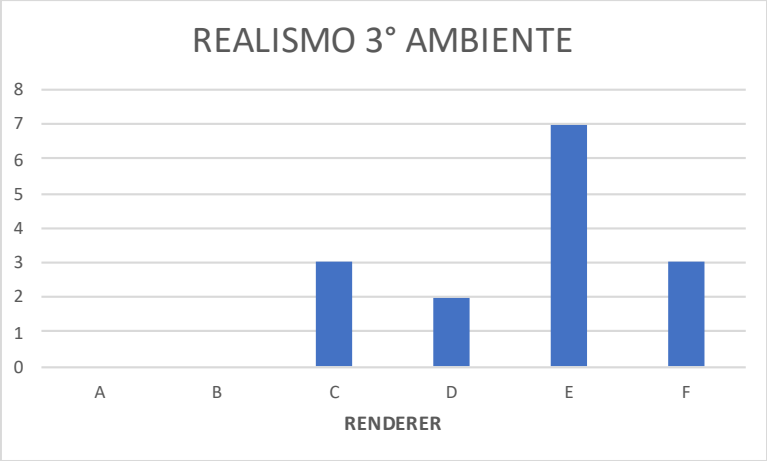


Figura A.12: Grafico della classifica del primo posto REALISMO del terzo ambiente

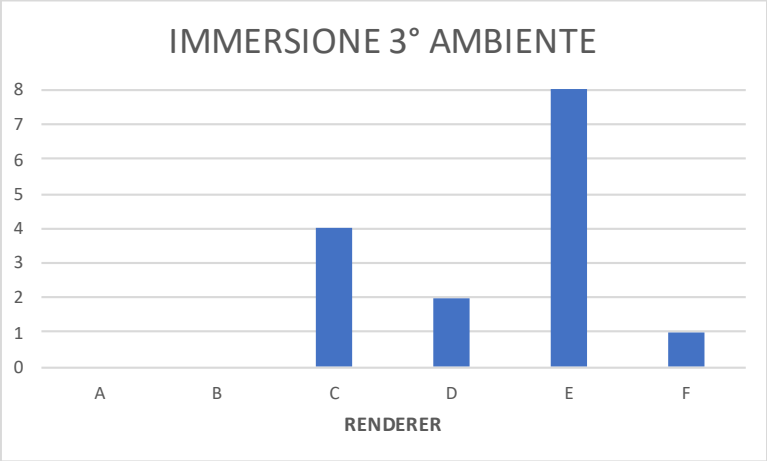


Figura A.13: Grafico della classifica del primo posto IMMERSIONE del terzo ambiente

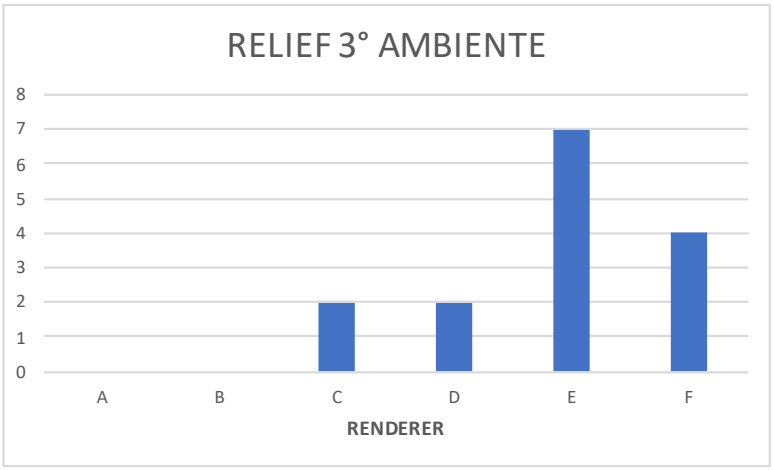


Figura A.14: Grafico della classifica del primo posto RELIEF del terzo ambiente

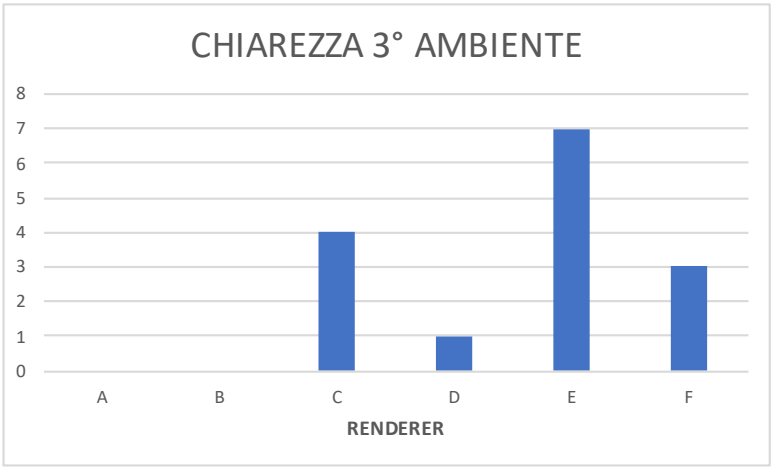


Figura A.15: Grafico della classifica del primo posto CHIAREZZA del terzo ambiente

Bibliografia

[1] B. Xie and X. Zhong, "Head-related transfer functions and virtual auditory display," *Soundscape Semiotics*, 2014.

[2] L. Sun, X. Zhong, and W. Yost, "Dynamic binaural sound source localization with interaural time difference cues: Artificial listeners," *Journal of the Acoustical Society of America*, vol. 137, pp. 2226–2226, 04 2015.

[3] J. Fels, "Binaural techniques – past and present," 08 2019.

[4] F. P. Freeland, L. Biscainho, and P. Diniz, "Efficient hrtf interpolation in 3d moving sound," 06 2002.

[5] H. Gamper, "Head-related transfer function interpolation in azimuth, elevation, and distance," *The Journal of the Acoustical Society of America*, vol. 134, no. 6, pp. EL547–EL553, 2013. [Online]. Available: https://doi.org/10.1121/1.4828983

[6] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The cipic hrtf database," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, Oct 2001, pp. 99–102.

[7] W. G. Gardner and K. D. Martin, "Hrtf measurements of a kemar," *The Journal of the Acoustical Society of America*, vol. 97, no. 6, pp. 3907–3908, 1995. [Online]. Available: https://doi.org/10.1121/1.412407

[8] A. Lindau, V. Erbes, S. Lepa, H.-J. Maempel, F. Brinkmann, and S. Weinzierl, "A spatial audio quality inventory (saqi)," *Acta Acustica united with Acustica*, vol. 100, 10 2014.

[9] L. S. R. Simon, N. Zacharov, and B. F. G. Katz, "Perceptual attributes for the comparison of head-related transfer functions," *The Journal of the Acoustical*

Society of America, vol. 140, no. 5, pp. 3623–3632, 2016. [Online]. Available: https://doi.org/10.1121/1.4966115

[10] G. Melacca and S. Invitto, "La realtà virtuale. strumento per elicitare processi neurocognitivi per il trattamento in ambito riabilitativo," 2016. [Online]. Available: http://siba-ese.unile.it/index.php/psychofenia/article/view/16142/13933

[11] Y. Deldjoo and R. E. Atani, "A low-cost infrared-optical head tracking solution for virtual 3d audio environment using the nintendo wii-remote," *Entertainment Computing*, vol. 12, pp. 9–27, 2016.

[12] O. Rummukainen, T. Robotham, S. Schlecht, A. Plinge, J. Herre, and E. Habets, "Audio quality evaluation in virtual reality: multiple stimulus ranking with behavior tracking," 08 2018.

[13] M. Geronazzo, E. Peruch, F. Prandoni, and F. Avanzini, "Applying a single-notch metric to image-guided head-related transfer function selection for improved vertical localization," *J. Audio Eng. Soc*, vol. 67, no. 6, pp. 414–428, 2019. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=20483

[14] M. Kleiner, B.-I. Dalenbäck, and P. Svensson, "Auralization-an overview," *J. Audio Eng. Soc*, vol. 41, no. 11, pp. 861–875, 1993. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=6976

[15] V. Algazi, R. Duda, R. Duraiswami, N. Gumerov, and Z. Tang, "Approximating the head-related transfer function using simple geometric models of the head and torso," *The Journal of the Acoustical Society of America*, vol. 112, pp. 2053–64, 12 2002.

[16] R. Bomhardt, "Anthropometric individualization of head-related transfer functions," Ph.D. dissertation, 09 2017.

[17] G. F. Kuhn, "Model for the interaural time differences in the azimuthal plane," *The Journal of the Acoustical Society of America*, vol. 62, no. 1, pp. 157–167, 1977. [Online]. Available: https://doi.org/10.1121/1.381498

[18] D. J. Kistler and F. L. Wightman, "A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction," *The Journal of the Acoustical Society of America*, vol. 91, no. 3, pp. 1637–1647, 1992. [Online]. Available: https://doi.org/10.1121/1.402444

[19] F. L. Wightman and D. J. Kistler, "The dominant role of low-frequency interaural time differences in sound localization," *The Journal of the Acoustical Society of America*, vol. 91, no. 3, pp. 1648–1661, 1992. [Online]. Available: https://doi.org/10.1121/1.402445

[20] A. Kulkarni, S. K. Isabelle, and H. S. Colburn, "Sensitivity of human subjects to head-related transfer-function phase spectra," *The Journal of the Acoustical Society of America*, vol. 105, no. 5, pp. 2821–2840, 1999. [Online]. Available: https://doi.org/10.1121/1.426898

[21] V. Algazi, C. Avendano, and R. O. Duda, "Estimation of a spherical-head model from anthropometry," *AES: Journal of the Audio Engineering Society*, vol. 49, 04 2002.

[22] R. O. Duda, C. Avendano, and V. Algazi, "An adaptable ellipsoidal head model for the interaural time difference," vol. 2, 04 1999, pp. 965 – 968 vol.2.

[23] H. Bahu and D. Romblom, "Optimization and prediction of the spherical and ellipsoidal itd model parameters using offset ears," 07 2018.

[24] M. Geronazzo, S. Spagnol, and F. Avanzini, "Mixed structural modeling of head-related transfer functions for customized binaural audio delivery," 07 2013.

[25] L. Sarlat, O. Warusfel, and I. Viaud-Delmon, "Ventriloquism aftereffects occur in the rear hemisphere," *Neuroscience letters*, vol. 404, pp. 324–9, 10 2006.

[26] M. Geronazzo, S. Spagnol, A. Bedin, and F. Avanzini, "Enhancing vertical localization with image-guided selection of non-individual head-related transfer functions," 05 2014.

[27] M. Geronazzo, E. Peruch, F. Prandoni, and F. Avanzini, "Improving elevation perception with a tool for image-guided head-related transfer function selection," 2017.

[28] D. Schönstein, "Hrtf selection for binaural synthesis from a database using morphological parameters," 2010.

[29] D. N. Zotkin, J. Hwang, R. Duraiswaini, and L. S. Davis, "Hrtf personalization using anthropometric measurements," *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, pp. 157–160, 2003.

[30] S. Xu, Z. Li, and G. Salvendy, "Individualization of head-related transfer function for three-dimensional virtual auditory display: A review," 07 2007, pp. 397–407.

[31] H. Hugeng, J. Anggara, and D. Gunawan, "Enhanced three-dimensional hrirs interpolation for virtual auditory space," *Proceedings of ICSigSys 2017*, pp. 35–39, 05 2017.

[32] P. Majdak, Y. Iwaya, T. Carpentier, R. Nicol, M. Parmentier, A. Roginska, Y. Suzuki, K. Watanabe, H. Wierstorf, H. Ziegelwanger, and M. Noisternig, "Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions," 05 2013.

[33] B. Boren, M. Geronazzo, P. Majdak, and E. Choueiri, "Phona: A public dataset of measured headphone transfer functions," 10 2014.

[34] C. Mendonça, G. Campos, P. Dias, J. Vieira, J. P. Ferreira, and J. A. Santos, "On the improvement of localization accuracy with non-individualized hrtf-based sounds," *Journal of the Audio Engineering Society*, vol. 60, pp. 821–830, 10 2012.

[35] O. Rummukainen, "Reproducing reality: Perception and quality in immersive audiovisual environments," Ph.D. dissertation, 12 2016.

[36] Google resonance audio engine. [Online]. Available: https://resonance-audio.github.io/resonance-audio/

[37] G. Kearney and T. Doyle, "An hrtf database for virtual loudspeaker rendering," in *Audio Engineering Society Convention 139*, Oct 2015. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17980

[38] Steam audio engine. [Online]. Available: https://valvesoftware.github.io/steam-audio/

[39] Custom hrtf steam audio integration. [Online]. Available: https://steamcommunity.com/games/596420/announcements/detail/1306452631300493638

[40] M. Geronazzo, E. Sikström, J. Kleimola, F. Avanzini, A. de Götzen, and S. Serafin, "The impact of an accurate vertical localization with hrtfs on short explorations of immersive virtual reality scenarios," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct 2018, pp. 90–97.

[41] O. Rummukainen, S. Schlecht, A. Plinge, and E. Habets, "Evaluating binaural reproduction systems from behavioral patterns in a virtual reality — a case study with impaired binaural cues and tracking latency," 10 2017.

[42] T. Hedke, J. C. Ahrens, J. Beyer, and S. Möller, "Impact of spatial audio presentation on the quality of experience of computer games," 2017.

[43] G. Reardon, A. Genovese, M. Gospodarek, C. Jerez, P. Flanagan, A. Roginska, S. Calle, and G. Zalles, "Evaluation of binaural renderers: A methodology," 10 2017.

[44] S. Spagnol, M. Geronazzo, and F. Avanzini, "On the relation between pinna reflection patterns and head-related transfer function features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 508–519, March 2013.

[45] C. Schissler, A. Nicholls, and R. Mehra, "Efficient hrtf-based spatial audio for area and volumetric sources," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 4, pp. 1356–1366, April 2016.

[46] L. Buriola, "Personalizzazione di head-related transfer function basata su misure antropometriche."

[47] A. Andreopoulou and B. Katz, "Identification of perceptually relevant methods of inter-aural time difference estimation," *The Journal of the Acoustical Society of America*, vol. 142, pp. 588–598, 08 2017.