

UNIVERSITÁ DEGLI STUDI DI MILANO

Corso di Informatica Musicale

Dipartimento di Informatica



Interrogazione di Database di IR per caratteristiche acustiche

Relatore: Prof. Luca Andrea Ludovico

Correlatore: Dott. Giorgio Presti

Tesi di Laurea di:

Federico Schiavoni, matricola 773897

Anno Accademico 2016/2017

INDICE

1	Introduzione	1
2	Presentazione dell'argomento	2
2.1	Che cos'è una IR (Impulse Response)?.....	2
2.2	La convoluzione	3
2.3	Software commerciali	4
3	Analisi	5
3.1	Sondaggio.....	5
3.1.1	Definizione del sondaggio.....	5
3.1.2	Analisi dati	6
3.2	Features Acustiche	8
3.2.1	Correlazione (Spazialità).....	8
3.2.2	Durata	9
3.2.3	Brightness (Timbro)	9
3.2.4	Early Reflection	10
3.2.5	Timbro + Spazialità	11
3.2.6	Naturalezza.....	12
3.2.7	Low Energy	13
3.2.8	Predelay.....	14
4	Sintesi	15
4.1	Progettazione Concettuale e Logica	15
4.1.1	Schema ER	15
4.1.2	Schema ER – Ristrutturato	17
4.1.3	Traduzione.....	18
4.2	Script Extract_Features	19
4.2.1	Algoritmi per estrazione features acustiche	22
4.3	AppIr – Applicativo GUI MATLAB	27
4.3.1	Interfaccia grafica.....	27
4.3.2	Codice.....	29

5	Test	34
5.1	Utilizzo di AppIr	34
5.2	Questionario	35
6	Conclusioni e sviluppi futuri	38
6.1	Conclusioni.....	38
6.2	Sviluppi Futuri.....	38
	Bibliografia	39
	Siti consultati	40

INTRODUZIONE

Questo elaborato si pone l'obiettivo di descrivere l'implementazione di una struttura dati di Impulse Response (IR) interrogabile. La scelta di tale implementazione è motivata dal fatto che una struttura dati interrogabile tramite features acustiche renderebbe molto più semplice la scelta di una IR durante le sessioni di lavoro in ambito musicale (quali situazioni live e studio recording), ottimizzando il tempo impiegato nel lavoro.

Nel capitolo due si presentano le nozioni teoriche degli argomenti principali utilizzati per la fase di analisi e quella di implementazione della struttura. In particolare viene fatta una breve introduzione sulla natura fisica della Risposta all'Impulso (IR). Viene poi presentata la funzione di convoluzione, operazione matematica tra due funzioni. Infine vengono elencati alcuni software commerciali che utilizzano IR per la creazione di reverberi, echo e delay.

Nel capitolo tre viene descritta la fase di analisi, eseguita prima con un breve sondaggio. Una volta ottenuti i dati, questi sono stati normalizzati ed utilizzati per estrapolare le features acustiche di maggiore interesse. Vengono, infine, elencate e spiegate le suddette features relative alle IR.

Nel capitolo quattro si entra nel merito dell'implementazione della struttura dati tramite progettazione concettuale, elaborazione dello script ed applicazione in MATLAB. Viene discusso, in primis, la progettazione attraverso schema Entità – Relazione e Traduzione. In seguito verranno commentate le parti di codice dello script, necessarie per l'estrazione delle features, e suddetta applicazione sia a livello macroscopico, sia nel particolare delle singole componenti.

Infine nel quinto capitolo vengono presentati e discussi i test e i risultati ottenuti.

PRESENTAZIONE ARGOMENTI

2.1 Che cos'è una IR (Impulse Response)?

La Risposta all'Impulso o Impulse Response di un sistema dinamico è il suo output quando viene sottoposto ad un breve segnale di ingresso, ossia un impulso (Delta di Dirac). La Delta di Dirac, introdotta dal matematico Paul Dirac, è una funzione pari a zero ovunque tranne a tempo zero dove vale uno, mentre il valore del suo integrale è uguale ad 1. Siccome l'impulso è una funzione che comprende in modo equo tutte le frequenze, l'IR definisce la risposta di un Sistema Lineare Tempo-Invariante (LTI) per tutte le frequenze. Un Sistema Lineare Tempo-Invariante è un sistema lineare i cui parametri non dipendono dal tempo, ossia il sistema rimane invariato nel dominio del tempo. L'uscita $y(t)$ di un sistema dinamico lineare stazionario (LTI) a tempo continuo ad un segnale in ingresso $x(t)$ è descritta dalla convoluzione:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t - \tau) \cdot h(\tau) d\tau = \int_{-\infty}^{\infty} x(\tau) \cdot h(t - \tau) d\tau$$

dove $h(t)$ è la risposta del sistema quando l'ingresso $x(t)$ è una funzione a Delta di Dirac.

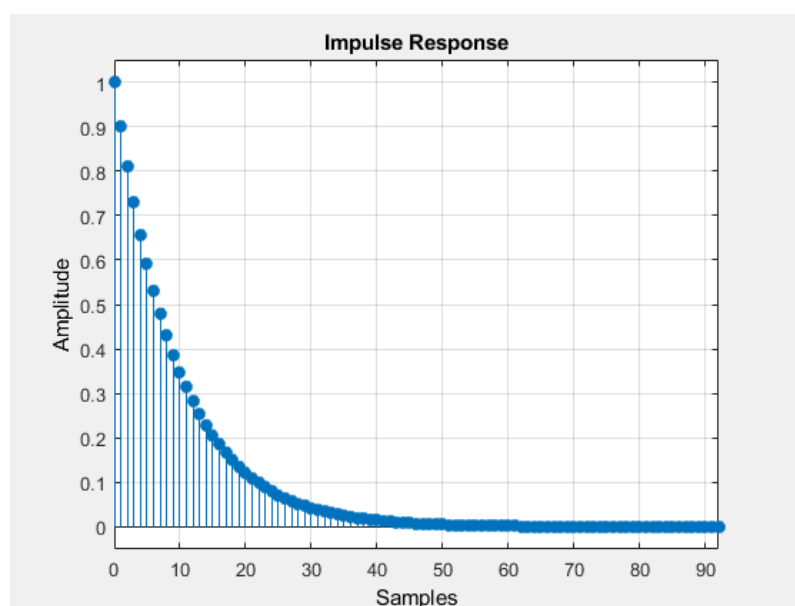


Figura 2.1: Rappresentazione di un Impulse Response in Matlab

2.2 La Convoluzione

In matematica (e, in particolare, nell'analisi funzionale) la convoluzione è un'operazione che permette di ottenere, da due funzioni x e y , una terza funzione z . La terza funzione può essere considerata come una modifica di una delle due funzioni originali e si ottiene calcolando l'integrale del prodotto tra la prima e la seconda funzione traslata di un certo valore. Nel dominio del tempo, la convoluzione corrisponde al prodotto tra lo spettro dei due segnali.

In Elaborazione Digitale dei Segnali, nella riverberazione artificiale la convoluzione è utilizzata per la risposta ad impulso di un ambiente reale ad un segnale audio digitale. In molte situazioni, le convoluzioni discrete possono essere convertite in convoluzioni circolari in modo da poter utilizzare trasformazioni rapide con una proprietà di convoluzione per implementare il calcolo.

Gli algoritmi di convoluzione veloce (Fast Convolution) più comuni utilizzano algoritmi di Trasformata Rapida di Fourier (FFT) tramite il teorema della convoluzione circolare. In particolare, la convoluzione circolare di due sequenze di lunghezza finita viene trovata prendendo una FFT di ogni sequenza, eseguendo il prodotto, e quindi eseguendo una FFT inversa.

Consideriamo due funzioni $f(t)$ e $g(t)$ definite da $\mathbb{R}$. Si definisce convoluzione di f e g la funzione nel seguente modo:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau$$

dove $\int_{-\infty}^{\infty}$ denota l'integrale definito sull'insieme dei numeri reali.

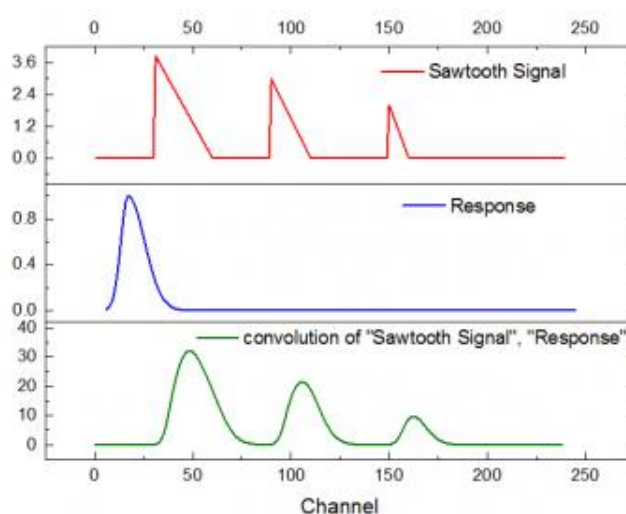


Figura 2.1.2: Convoluzione di un Segnale a Dente di Sega con un impulso

2.3 Software Commerciali

Al giorno d'oggi esistono in commercio diversi plugin software che permettono di utilizzare IR per ricreare effetti di reverberi, echo e delay tramite convoluzione. In questo modo i sound designer, ingegneri del suono e i fonici che operano in studi di registrazione possono ricreare riverberazioni di ambienti di varie dimensioni con un semplice tasto.

Uno dei plugin più famosi per riverberazione è Altiverb di Audio Ease, plugin che permette di controllare diversi parametri come il tempo di riverberazione, la dimensione dell'ambiente, il pre-delay e attack, l'eq e il brightness dell'ambiente.

Un altro plugin per i delay e riverberi è Delay Designer di Apple. Permette anche qui di scegliere tra una serie di Risposte all'Impulso di default o di inserire IR di terze parti per poter creare ritardi e reverberi andando a modificare i parametri precedentemente elencati.

Esistono anche IR Loader, plugin utilizzati soprattutto per caricare impulsi di cabinet per chitarra e basso. Uno di questi è Pulse, plugin free della Rosen Digital che permettere di caricare fino a due Impulse Response in modo da poter avere una configurazione stereo oltre che mono. Nel plugin della Line 6 Helix Native, durante la creazione di routing del segnale, è presente uno slot di IR Loader dove è possibile caricare impulse response di terze parti o creati dall'utente in modo da poter utilizzare la simulazione di amplificazione con le proprie IR.

3.1 Sondaggio

Prima di iniziare a progettare la struttura dati ed a sviluppare l'applicazione, è stato formulato un sondaggio, in modo da capire quali caratteristiche acustiche muovessero più interesse tra gli addetti del settore audio.

Il sondaggio è stato sottoposto a ingegneri del suono, sound designer, tecnici audio di studi di registrazione e live, ricercatori accademici ed amatoriali.

Una volta ottenute le risposte, sono stati analizzati i dati in modo da utilizzarli per l'implementazione della struttura e dell'applicativo.

3.1.1 Definizione del Sondaggio

Il sondaggio è stato formulato con delle domande specifiche contenenti dati anagrafici per rendere la profilazione dei soggetti più precisa.

Dalle risposte ottenute si è potuto avere un quadro generale dei vari ambiti lavorativi, delle IR utilizzate durante le sessioni di lavoro, delle caratteristiche acustiche influenti, eventuale difficoltà sulla scelta delle IR e opinioni riguarda a una strutturazione più efficace per la gestione e archiviazione delle IR.

Nel sondaggio erano previsto risposte chiuse, a scelta multipla, e domande aperte per raccogliere ulteriori dati. Per la creazione del sondaggio è stato utilizzato Google Module. La distribuzione è avvenuta tramite gruppi e forum di social network, in particolare Audio Recording Forum di Facebook.

Vengono elencate qui di seguito le domande del sondaggio:

- In che modo impiegate le IR (ambito lavorativo)?
- Quali IR utilizzate?
- A quali caratteristiche fisiche o psicoacustiche date la priorità nella scelta delle IR da utilizzare?

- Trovi difficoltà nella scelta delle IR durante le tue sessioni di lavoro? Se “SI” quali?
- Nella scelta di IR consideri più importante come “suona” o “cosa dovrebbe suonare” (Qualità del Suono o Qualità della Simulazione)?
- Secondo le vostre esperienze lavorative, una qualche forma di organizzazione e navigazione tramite libreria potrebbe migliorare o facilitare la scelta delle IR da impiegare?
- E se “SI” in quali aspetti migliorerebbe la vostra esperienza lavorativa?

Si è passati poi alla fase di analisi dei dati.

3.1.2 Analisi dati

Il sondaggio è stato compilato da 70 utenti proveniente da diversi ambiti audio, come Ingegnere del Suono, Sound Designer, Ricercatori Accademici ed Amatori del settore. L’analisi dei dati è di rilevante importanza per avere un quadro più preciso riguardo agli ambiti lavorativi, alle IR utilizzate durante le sessioni di lavoro, alla eventuale difficoltà incontrata dai soggetti nella scelta di quest’ultime e al parere riguardo eventuale riorganizzazione dal punto di vista di archiviazione delle IR. Nella figura 3.1.2 è possibile visualizzare una tabella riassuntiva contenente le statistiche dei dati raccolti. Dall’analisi dei dati è potuto emergere un dato fondamentale per la progettazione della struttura di dati e, in seguito, sviluppo dell’applicativo: quali caratteristiche acustiche fossero di maggior importanza nella scelta di una IR durante le sessioni di lavoro. A domanda sottoposta, i soggetti del sondaggio hanno dichiarato quali caratteristiche acustiche valutavano per scegliere o meno una IR durante il mixaggio di un brano o la loro sessione di lavoro. Raccolti i dati relativi alla domanda, si è potuto classificare le varie caratteristiche acustiche di maggiore interesse dividendole tramite label quali:

- | | | |
|------------------------|--------------------|-------------------------|
| • Realismo | • Estetica | • Colore |
| • Tridimensionalità | • Naturalezza | • Profondità |
| • Spazialità | • Consistenza | • Risposta in Frequenza |
| • Evoluzione Spettrale | • Accuratezza | • Tipologia di |
| • Dinamica | • Early Reflection | Campionatura |
| • Stereofonia | • Variabile | • Qualità |
| • Tempo di | • Lunghezza | • Nitidezza |
| Riverberazione | • Densità | • Ritardo |
| • Coerenza di Fase | | |

Caratteristiche Fisiche o Psico - Acustiche	Numero di Risposte
Realismo	15
Spazialità	14
Variabile	11
Naturalezza	10
Tempo di Riverberazione	9
Estetica	7
Tipologia di Campionatura	4
Risposta in Frequenza	4
Early Reflection	3
Densità	3
Evoluzione Spettrale	1
Dinamica	1
Coerenza di Fase	1
Consistenza	1
Accuratezza	1
Qualità	1
Nitidezza	1
Ritardo	1

Figura 3.2.1: Tabella statistiche

Si è deciso di raggruppare per caratteristiche simili in label per ottenere le features acustiche che soddisfacessero le esigenze riscontrate durante il sondaggio.

È stato possibile così ottenere otto caratteristiche acustiche:

- Correlazione: misura la spazialità
- Durata: approssima RT60 e distingue cabinet per chitarra da riverberi
- Brightness: descrive qualitativamente il timbro

- Early Reflection: permette di capire la dimensione dell'ambiente campionato
- Correlazione + Brightness: combinazione tra la misura di spazialità e la qualità timbrica. Permette di calcolare l'evoluzione spettrale del segnale.
- Naturalzza: permette di percepire se una IR sia stato catturato da un ambiente reale oppure sintetizzato digitalmente
- Low Energy: permette di distinguere la tipologia delle IR, se echi, delay oppure riverberi
- Predelay: parametro riferito alla quantità di tempo che intercorre tra il suono diretto e lo sviluppo della riverberazione, prime riflessioni incluse

3.2 Features Acustiche

Verranno approfondite di seguito ognuna delle caratteristiche trovate.

3.2.1 Correlazione (Spazialità)

La Spazialità descrive la dimensione di un ambiente riprodotto da una IR. La dimensione varia a seconda di diversi fattori: se l'IR dell'ambiente è stato catturato in modo stereo oppure mono, la dimensione reale della ambiente, l'apparecchiatura utilizzata per la ripresa delle IR (il numero di microfoni utilizzati di ripresa) ecc..

Esistono diversi ambienti da cui è possibile ottenere una IR come per esempio cattedrali, hall, stanza di piccole o medie dimensioni, cabinet per chitarra e basso, teatri, reattori nucleari, ambienti e luoghi della vita quotidiana (stanze da letto, sale da bagno, salotti) ecc..

Per determinare la spazialità è stata scelta la misura di correlazione, valutata tra il canale destro e il canale sinistro: se sarà un valore vicino a “zero” molto probabilmente l'IR è di tipo stereo e rappresenterà un ambiente di grandi dimensioni; se, invece, sarà vicino ad “uno” sarà di tipo mono e potrebbe rappresentare un ambiente piccolo, per esempio un cabinet per chitarra.

3.2.2 Durata

È la caratteristica che rappresenta la durata dell'IR. È stata calcolata in questo modo: ottenuto il numero di campioni, il risultato è stato diviso per la frequenza di campionamento della risposta all'impulso.

Come scelta di unità di misura, si è deciso di utilizzare i secondi (s).

3.2.3 Brightness (Timbro)

Il Brightness è la quantità di energia dedicata alle alte frequenze. È una feature locale e può essere rappresentata nel dominio del tempo o della frequenza.

Tramite il brightness è possibile calcolare il timbro dell'IR: infatti valori alti di brightness saranno relativi ad una IR di timbrica molto chiara, mentre valori bassi della feature indicheranno un timbro piuttosto scuro.

Può essere rappresentato in Decibel (dB) oppure Hertz (Hz).

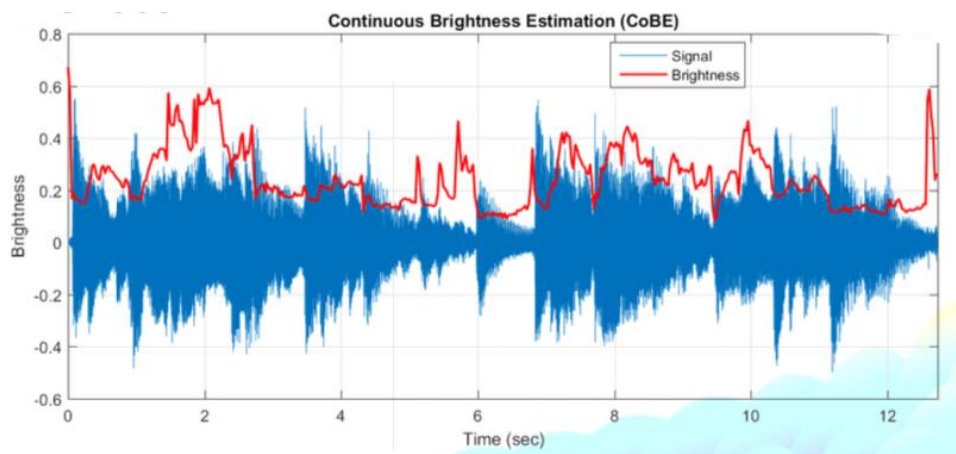


Figura 3.2.3: Grafico del Brightness tratto dall'algoritmo CoBe del Dottor Giorgio Presti

3.2.4 Early Reflection

Secondo le definizioni standard, le prime riflessioni sono suoni che arrivano all'ascoltatore dopo essere stati riflessi una o due volte da parti dello spazio di ascolto, come muri, soffitti e pavimento. Arrivano più tardi del suono diretto, spesso in un intervallo compreso tra 5 e 100 millisecondi, ma possono arrivare prima dell'inizio della piena riverberazione. Le prime riflessioni forniscono al cervello le informazioni sulla dimensione di una stanza e sul senso di distanza dei suoni in una stanza. Hanno un ruolo importante nel determinare il carattere generale e il suono della stanza.

Le early reflection sono state misurate in millisecondi (ms).

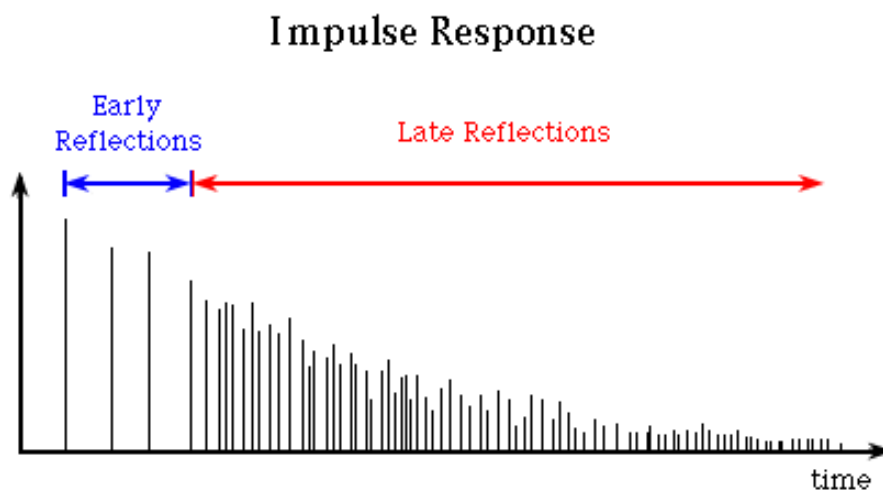


Figura 3.24: Grafo rappresentante l'Early Reflection

3.2.5 Timbro + Spazialità

La feature Timbro + Spazialità è una caratteristica acustica che unisce la spazialità e il brightness dell'impulse response.

Viene ottenuta calcolando la media del quadrato della differenza del brightness tra il canale destro e sinistro dell'impulse response.

Se si avrà un valore alto (per esempio vicino ad uno) si otterrà una differenza timbrica tra i due canali, altrimenti il valore rimarrà basso (per es. vicino a zero).

Di seguito la formula matematica:

$$\frac{(brightness(a_n) - brightness(b_n))^2}{n}$$

3.2.6 Naturalizza

La Naturalizza è una feature acustica che permette di indicare il valore di realismo nelle IR, se sia stato creato tramite metodo analogico oppure sintetizzato digitalmente.

Si è calcolata la Naturalizza tramite l'entropia, che è in grado di catturare questa differenza di segnale: i segnali generati analogicamente avranno dei valori molto più imprevedibili, mentre i segnali sintetizzati saranno più regolari.

È possibile creare un impulse response sia in modo analogico oppure digitale:

- **Metodo Analogico:** l'IR di un ambiente reale viene catturato con una strumentazione di ripresa, che comprende microfoni (il tipo e il numero è scelto in base alla ripresa che si vuole ottenere), sistema di monitoraggio e un source sound.

L'ambiente ripreso può essere di dimensioni diverse, come cattedrali, room, hall, cabinet per chitarra ecc..

- **Metodo Digitale:** viene creato una IR attraverso funzioni matematiche in un ambiente di sviluppo, come per esempio Matlab.

Questo semplifica la creazione perché non è necessario essere fisicamente nel luogo per poter ottenere una misurazione e quindi una ripresa dell'ambiente.

3.2.7 Low Energy

La Low Energy o Low Level Ratio è definita come la percentuale in cui il Root Mean Square è sotto la media.

RMS è definito come la radice quadrata del quadrato medio (la media aritmetica dei quadrati di un insieme di numeri), oppure noto come media quadratica ed è un caso particolare della media generalizzata con esponente uguale a 2.

Questa caratteristica permette di differenziare le varie IR nel caso che siano echi, delay oppure riverberi.

La Low Energy è una feature acustica globale e descrive la variazione di dinamica delle IR. È rappresentata nel dominio del tempo ed è misurata a percentuale.

Low Energy Ratio (low level ratio)

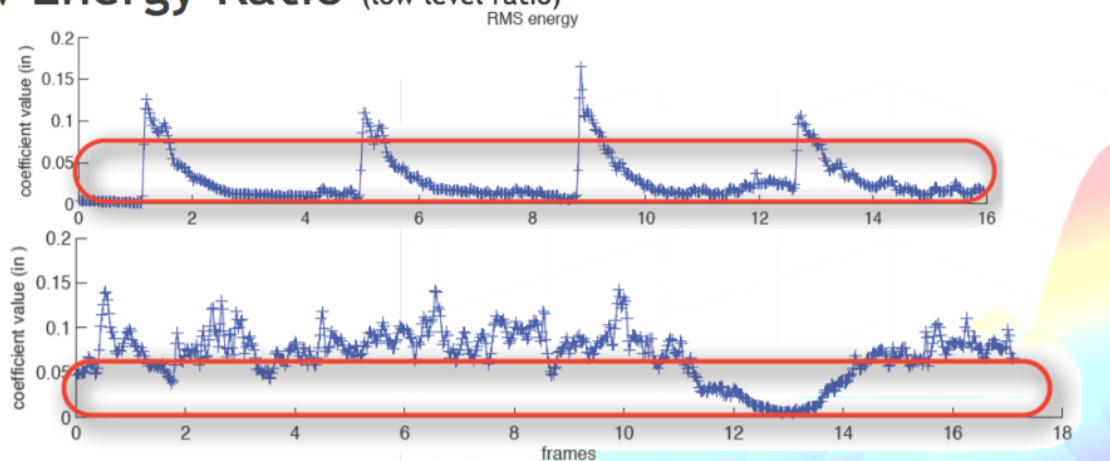


Figura 3.2.7: Grafo rappresentata la Low Energy

3.2.8 Pre – Delay

Il Pre-delay è la quantità di tempo che intercorre tra il suono diretto e lo sviluppo della riverberazione, prime riflessioni incluse.

È un parametro che si trova nei processi di riverberazione. Fondamentale per la percezione della dimensione della stanza, perché per il nostro orecchio è il tempo che il suono impiega a sbattere nelle pareti e a tornare indietro.

La regolazione accurata del parametro di pre-delay fa una grande differenza nella "clarity" di un mix musicale. Ad esempio, un pre-delay più lungo sposterà la coda del riverbero delle voci, rendendole molto più presenti e comprensibili.

Il Pre-delay è misurato in millisecondi (ms).

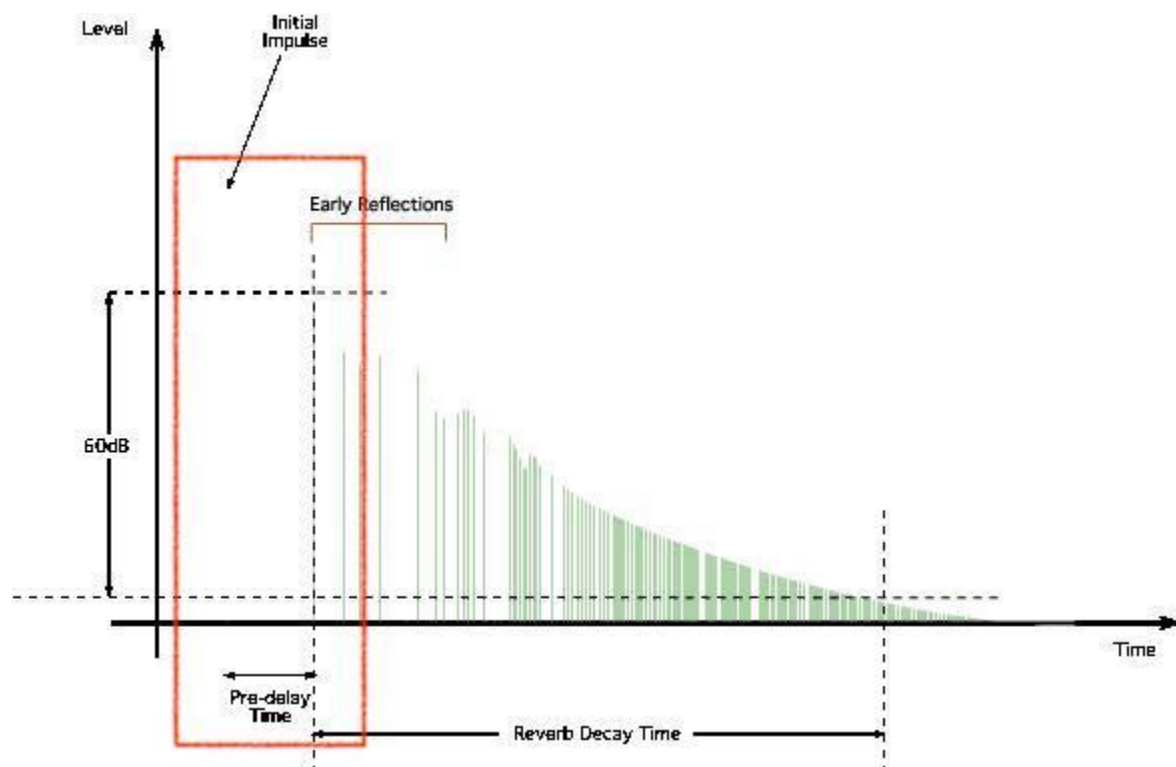


Figura 3.2.8: Grafo rappresentante il tempo di Predelay di un Reverbero

In questo capitolo si descrive la progettazione concettuale e logica dello schema ER, l'implementazione di uno script in MATLAB per l'estrazione delle features acustiche ed, infine, l'applicativo AppIr, costituito da un'interfaccia grafica ed relativo codice in MATLAB.

4.1 Progettazione Concettuale e Logica

Prima di addentrarsi nel discorso dello sviluppo relativo all'applicativo, viene presentato di seguito come è stato strutturato a livello concettuale il database per la realizzazione futura di un plugin dall'applicativo MATLAB, che permetterà all'utente di interagire in modo da poter scegliere le varie IR contenute nel dataset. Si è partiti da una progettazione concettuale dello schema ER, relativa Ristrutturazione ed infine Traduzione, dove le entità e le associazione dello schema vengono tradotte in relazioni.

4.1.1 Schema ER

Lo schema ER iniziale comprende 3 entità e 4 associazioni.

Le Entità sono le seguenti:

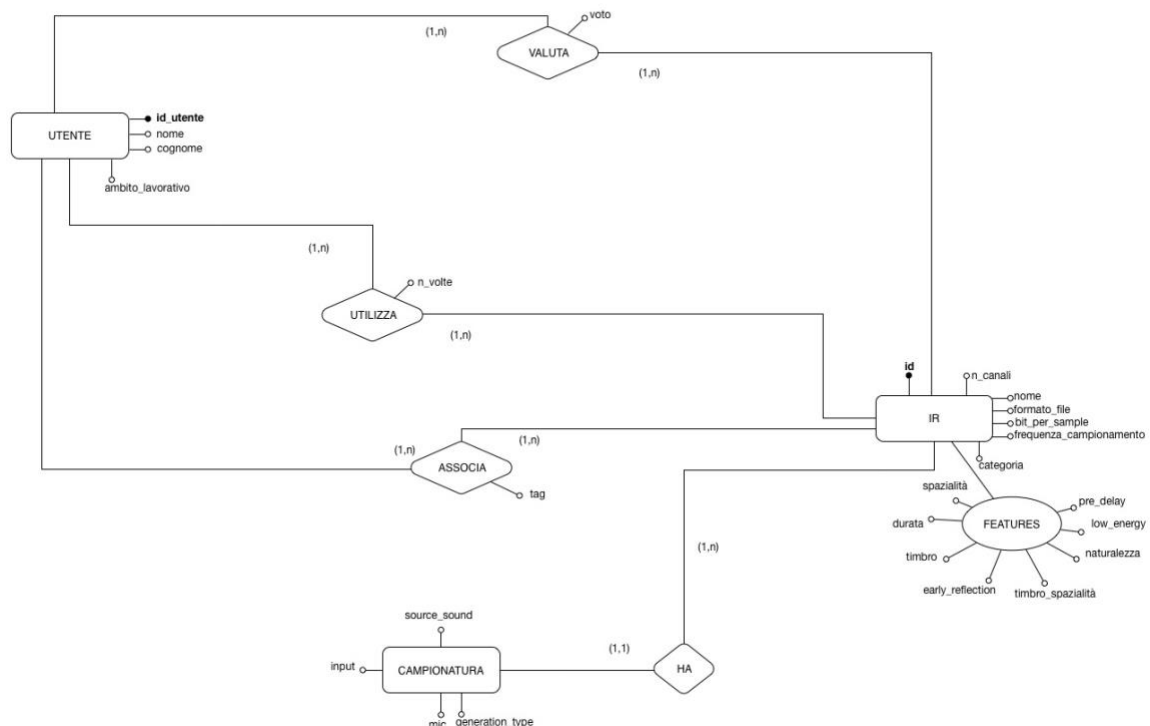
1. IR: entità forte comprendente gli attributi relativi alle varie IR del dataset. È identificata dall'identificatore **id** e comprende gli attributi nome, formato file, bit per sample, frequenza campionamento, categoria, n canali ed infine, ma non meno importante, l'attributo composto FEATURES contenente le caratteristiche acustiche elencate nel capitolo 3.
2. UTENTE: entità forte in cui sono contenuti i dati relativi all'utente. Contiene l'identificatore **id_utente** e gli attributi nome, cognome e l'ambito lavorativo.
3. CAMPIONATURA: entità debole che viene identificata da **id** dell'entità IR. Contiene gli attributi relativi alla strumentazione e al metodo di cattura delle IR. Gli attributi sono i seguenti: mic, generation type, source sound ed infine input (relativo al tipo di monitoraggio usato).

Le Associazioni presenti sono:

1. VALUTA: associazione tra entità IR ed entità UTENTE contenente l'attributo voto. Questa associazione molti a molti stabilisce che utente può dare una valutazione alle IR appena utilizzate in modo da migliorare e velocizzare la ricerca futura.
2. UTILIZZA: associazione molti a molti tra UTENTE ed entità IR contenente l'attributo n_volte. La seguente associazione indica il numero di volte che un utente ha impiegato l'IR: ogni volta che una IR viene utilizzata, aumenta il valore di n_volte di +1. Il numero di volte utilizzato renderà molto più celere la ricerca futura dell'IR già impiegata, in modo da ottimizzare il lavoro.
3. ASSOCIA: associazione di tipo molti a molti contenente l'attributo tag. Questa associazione permette ad un utente di affidare ad ogni IR un tag, in modo da semplificare la ricerca nel dataset durante le sessioni di lavoro e snellire il workflow.
4. HA: è un'associazione debole tra l'entità IR e l'entità Campionatura. Essendo debole durante il processo di Normalizzazione verrà integrata nella relazione Campionatura. Non contiene alcun attributo.

Viene presentato di seguito lo schema ER realizzato:

SCHEMA ER - STRUTTURA DATI



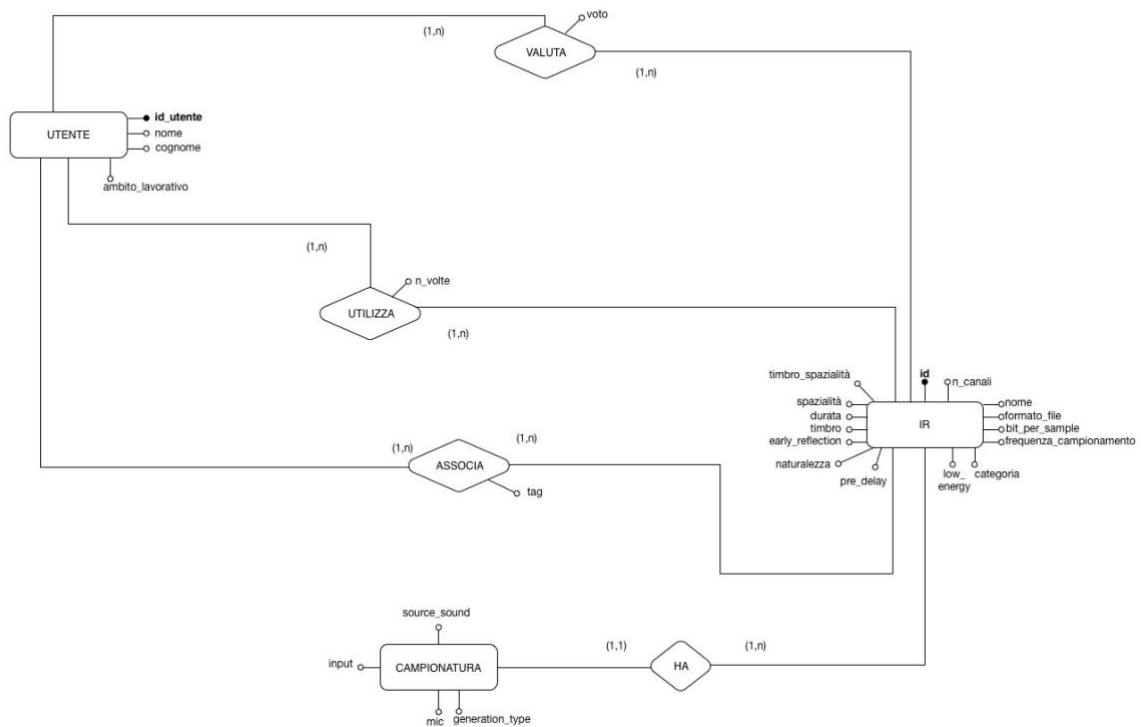
4.1.2 Schema ER – Ristrutturazione

Lo Schema ER Ristrutturato comprende le 3 entità (rispettivamente IR, UTENTE ed CAMPIONATURA) e le 4 associazioni precedenti (rispettivamente ASSOCIA, VALUTA, HA ed, infine, UTILIZZA). La fase di ristrutturazione è stata effettuata sull'attributo composto dell'entità IR.

I sotto-attributi contenuti nell'attributo FEATURES sono stati inseriti direttamente nell'entità IR, in modo da poterlo eliminare.

Rappresentazione dello schema ER – Ristrutturato:

SCHEMA ER - STRUTTURA DATI



4.1.3 Traduzione

Nella fase di Traduzione di uno schema ER, le singole entità ed associazioni vengono tradotte in modo da essere rappresentate tramite Relazioni.

Le Relazioni risultanti da un'entità sarà una relazione contenente tutti gli attributi di quella singola entità. Nel caso di un'entità debole (vedi CAMPIONATURA), conterrà un identificatore esterno oltre a i suoi attributi.

Per le Associazioni, invece, la relazione risultante avrà come attributi gli identificatori delle Entità in associazione, più eventuali attributi dell'associazione stessa.

Qui di seguito viene mostrata la Traduzione delle Entità ed Associazioni dello schema ER.

SCHEMA LOGICO RELAZIONALE

Entità

IR (**id**, nome, n_canali, formato_file, bit_per_sample, frequenza_campionamento, categoria, spazialità, durata, timbro, early_reflection, timbro_spazialità, low_energy, naturalezza, predelay)

UTENTE (**id_utente**, nome, cognome, ambito_lavorativo)

CAMPIONATURA (**id_{IR}**, mic, generation_type, source_sound, input)

Associazioni

VALUTA (**id_{IR}**, **id_utente_{UTENTE}**, voto)

UTILIZZA (**id_{IR}**, **id_utente_{UTENTE}**, n_volte)

ASSOCIA (**id_{IR}**, **id_utente_{UTENTE}**, tag)

4.2 Script Extract_Features

In questo paragrafo si descrive l'implementazione di uno script in Matlab per l'estrazione delle features acustiche presentate nei paragrafi del Capitolo 3. Dopo una breve panoramica dello script in generale, verranno elencati gli algoritmi implementati per ciascuna delle features interessate.

Breve panoramica dello script

Lo script Extract_Features permette di estrapolare le caratteristiche acustiche dalle IR in formato wav presenti in una determinata directory. Nella prima parte del codice, avviene il setup dell'ambiente lavorativo che consiste nell'aggiungere il percorso cartella contenente le IR ed inizializzare le variabili necessarie per la selezione dei file, il numero di features ed, infine, l'inizializzazione di una matrice che conterrà i dati delle caratteristiche ottenuti dagli algoritmi implementati per estrapolare quest'ultime dai file wav. Nel caso che la cartella non contenga alcuna IR, verrà stampato un messaggio di errore.

```
% Matlab Script per l'estrazione di features di IR
% Creato da Federico Schiavoni
% SETUP DELL'AMBIENTE LAVORATIVO
% aggiungo path a matlab

% Cartella in cui cercare gli IR
percorso_cartella = '/Users/Fred/Desktop/Universit#/Tesi
Laurea/struttura_dati/IR/dataset/';

% Ottengo la lista degli IR in formato .wav
lista_file = dir([percorso_cartella '*.wav']);

% Conteggio dei file trovati
n_impulsi = length(lista_file);

% Numero delle features
n_features = 8;

%Prealloco matrice per contenere i dati in futuro
matrice_features = [];

% SE NON TROVO IR NELLA CARTELLA DO UN ERRORE
if( ~n_impulsi )
    error(['No files found in ' percorso_cartella ]);
end
```

Codice MATLAB: setup ambiente lavorativo

La seconda parte di codice prevede un ciclo “for” in cui viene selezionato una IR alla volta dalla cartella. Il file wav viene valutato se è formato stereo oppure mono e letto attraverso la funzione **audioread()**; in seguito viene passato nelle otto funzioni per estrapolare le features. Mano a mano che le features vengono estrapolate, i dati relativi vengono salvati in un array che, a fine ciclo, andrà a popolare la matrice precedentemente allocata.

```
% Utilizzo un ciclo FOR per passare in rassegna le IR della cartella
for i = 1:n_impulsi

    % estraggo nome e percorso dell'ir corrente da analizzare
    ir = [percorso_cartella, lista_file(i).name];

    %prevelo il file IR
    [camp,freq_camp] = audioread(ir);

    %calcolo il numero di canali per capire se ir È stereo o mono per
    %futura comodità nell'analisi

    n_canali = size(camp,2); %numero canali campione audio

    %creo vettore double per mettere dati temporaneamente
    %creo variabile j per indicizzare

    array_dati = [];
    j = 1;

    %Calcolo Spazialità

    space = space_correlation(camp,n_canali);
    array_dati(j) = space;
    j = j + 1;

    %Calcolo Durata

    tempo = durata(camp,freq_camp);
    array_dati(j) = tempo;
    j = j + 1;

    %Calcolo Timbro ( Brightness )

    bright = brightness(camp,freq_camp);
    array_dati(j) = median(bright);
    j = j + 1;

    %Calcolo Early Reflection

    e_r = early_reflection(camp,freq_camp);
    array_dati(j) = e_r;
    j = j + 1;
```

Codice MATLAB: inizio ciclo “for” e prime funzioni relative alle features acustiche

Una volta terminato il ciclo, i dati estrapolati vengono salvati in un file matrice .mat contenente due variabili: la variabile “matrice_features” che i dati relativi alle otto features per ogni IR presente nella cartella; la variabile “lista_file” contenente i nomi e percorsi relativi delle IR.

```
%Calcolo Timbro + Spazialità

t_s = timbro_space(camp,n_canali,freq_camp);
array_dati(j) = t_s;
j = j + 1;

%Calcolo Naturalezza
%manca algoritmo

natural = naturalezza(camp);
array_dati(j) = natural;
j = j + 1;

%Calcolo Low Energy

low_en = low_energy(camp);
array_dati(j) = low_en;
j = j + 1;

%Calcolo Pre Delay
%Manca algoritmo

pdl = predelay(camp,freq_camp);
array_dati(j) = pdl;


%Riempio matrice preallocata con array contenente dati
% n_righe = numero di file
% n_colonne = numero di features

for l = 1:n_features
    matrice_features(i,l) = array_dati(l);
end

end

%salvo i risultati in un file .mat

save('matrice.mat', 'matrice_features', 'lista_file');
```

Codice MATLAB: ultime funzioni relative alle features acustiche e salvataggio matrice.mat

4.2.1 Algoritmi per estrazione features acustiche

Vengono presentati ora gli algoritmi dedicati all'estrazione delle features acustiche. Ogni algoritmo è implementato in una funzione tramite MATLAB. Durante l'esecuzione, le funzioni vengono richiamate dallo script e per ognuna viene emesso un output di dati relativo ad essa; infine il risultato viene salvato nella locazione di un array precedentemente inizializzato (contenuto nello script `extract_feature`).

Correlazione (Spazialità)

Per ottenere la spazialità si è utilizzata la funzione di correlazione “**corr()**”. Per prima cosa sono stati passati come input i parametri **camp**, dove rispettivamente rappresentano l'IR. A seconda del numero di canali, la correlazione avverrà sui canali destro e sinistro (impulse response stereo) oppure la correlazione risulterà essere uguale ad uno (impulse response mono).

```
function [ space ] = space_correlation( camp )
%Calcola la spazialità dell'IR tra canale destro e sinistro.
%Se l'IR È mono calcola la spazialità sul solo canale ( risultato =
1 )

    %calcolo numero di canali
    n_canali = size(camp,2);

    if( n_canali == 2 ) %vuol dire che il segnale È stereo
        left_channel = camp(:,1);
        right_channel = camp(:,2);
        space = corr(left_channel,right_channel);
        %al massimo come valore prendo tra 0 e 1 niente negativi
        space = max(space,0);
    else
        space = 1; %il canale È mono
    end
end
```

Codice MATLAB: funzione `space_correlation` per calcolare la spazialità di un ir.

Durata

La durata è stata valutata come “durata dell’IR” o “durata del campione”. Nella funzione vengono passati come parametri **camp** e **fc** (frequenza di campionamento). Una volta ottenuto il numero di campioni delle IR tramite la funzione **length()**, è stato diviso per la frequenza di campionamento relativa.

```
function [ tempo ] = durata( camp, fc )
%Passati campione e frequenza di campionamento dell'IR la funzione (
%attraverso la funzione length ) calcola la durata del campione IR

%calcolo il numero di campioni dell'IR

n_campioni = length(camp);

%calcolo la durata dell'IR

tempo = n_campioni * ( 1 / fc ); %fc = freq di campionamento

end
```

Codice MATLAB: funzione per calcolo durata di un ir.

Brightness

Per calcolare la feature brightness è stato utilizzato una versione semplificata dell’algoritmo CoBE del Dottor Giorgio Presti, in quanto in questo algoritmo non viene considerata la finestra temporale presente nell’originale. Passati in input i parametri **camp** e **fs**, per prima cosa viene calcolato l’involuppo e salvato nella variabile **env**. Successivamente viene filtrato il segnale **hf**, in modo da ottenere lo spettro contenente solo le high frequencies necessarie per il calcolo. Infine viene salvato nella variabile **br** il rapporto tra i due segnali.

```
function [ br ] = brightness(camp,fs)
% Calcolo quanto È "chiaro" l'IR preso in esame

[env, ~] = envelope(camp,0.01*fs,'rms');
b = [1,-1];
a = 1;
[hf, ~] = envelope(filter(b,a,camp),0.01*fs,'rms');
br = hf./env;

%br = min(br,1);
% prendo il valore assoluto
br = abs(br);
br = min(br,2);
br = mean(br,'omitnan');

end
```

Codice MATLAB: funzione per calcolo brightness di un ir.

Early Reflection

Una volta reso mono l'IR (tramite la funzione **mean**) e calcolato il numero di campioni, per l'analisi della feature è stato necessario ottenere due variabile: **s** ed **e** (rispettivamente start ed ending del segnale). Si è potuto procedere al calcolo dell'early reflection tramite la funzione **rms (Root Mean Square)**.

```
function [ early ] = early_reflection(camp,fs)
%calcolo delle early_reflection

%rendo il campione mono
camp = mean(camp,2);

%calcolo il num di campioni
len = size(camp,1);

%calcolo start and ending per analisi early
s = round(0.001*fs);
e = round(len/15);

%prendo valore max
s = max(s,1);
e = max(e,s+1);

%calcolo early_reflection
er = camp(s:e);
early = rms( er );

end
```

Codice MATLAB: funzione per calcolare Early Reflection

Timbro + Spazialità

Questa feature è stata valutata calcolando la media del quadrato della differenza del brightness tra il canale sinistro e il canale destro del campione. Se l'IR è stereo verrà valutata la brightness per ogni canale e calcolata la differenza tra i due; se è mono come risultato si otterrà zero.

```
function [ result ] = timbro_space( camp, fs )

%calcolo il numero di canali
n_canali = size(camp,2);

if( n_canali == 2 )
    left_channel = brightness(camp(:,1),fs);
    right_channel = brightness(camp(:,2),fs);
    result = setdiff(left_channel,right_channel);
else
    result = 0; %il canale è mono
end
end
```

Codice MATLAB: funzione per calcolare timbro + spazialità

Naturalezza

È stata valutata utilizzando la funzione di entropia. Nella teoria dell'informazione, quanto è d'impedimento alla chiarezza e univocità del messaggio; maggiore è l'entropia, minore è la quantità di informazione. Nel calcolo della feature si è considerato la quantità di informazioni o meglio elementi per stabilire la naturalezza di un impulse response: se sarà ricco di informazioni molto probabilmente sarà una IR la più fedele possibile all'ambiente originale.

```
function [ n ] = naturalezza( ir )
%NATURALEZZA calcolata attraverso l'entropia

n = entropy(ir);

end
```

Codice MATLAB: funzione per calcolare la naturalezza di un ir.

Low Energy

È valutata come la percentuale di tempo in cui l'RMS è al di sotto della media. Per ottenere l'algoritmo per prima cosa è stato calcolata la media dell'IR passando il parametro di input tramite la funzione **mean()** ed utilizzando 'omitnan' per omettere i valori Nan. In seguito sono stati valutati gli n campioni dell'IR al di sotto della media. Infine è stata calcolata la somma dei campioni al di sotto della media e diviso per il numero di campioni totali dell'IR.

```
function [ low_en ] = low_energy(camp)
%Rapporto tra l'energia totale e l'energia sotto la media

%ometto i valori nan
media_camp = mean(camp, 'omitnan');
count = 1 * (camp < media_camp);
len = size(camp,1);
low_en = sum(count, 'omitnan')/len;

%ottengo media del low energy fra i due canali
low_en = mean(low_en,2);

end
```

Codice MATLAB: funzione per calcolare la low energy

Pre-Delay

Per calcolare il Pre-Delay si procede trovando il tempo in cui occorrono i primi picchi dell'involuppo d'ampiezza, dopo il segnale diretto (rappresentato dal primo picco in assoluto).

Si è utilizzata la funzione **find()** per trovare i picchi del segnale.

```
function [ pdl ] = predelay( ir, fs )
%PREDELAY

%creo una window size
wsiz = round(0.0005*fs);
%rendo il campione mono
ir = mean(ir,2);

%calcolo l'involuppo prendendo il campione, la window size ed
rms

[e,~]=envelope(ir,wsiz,'rms');
e = filter([1,-1], [1], e);
e = max(0,-diff(sign(e)));
peaks = find(e)/fs;

np = numel(peaks);

if np>0
    pdl = peaks(min(2,np));
else
    pdl = 0;
end

end
```

Codice MATLAB: funzione per calcolare il Pre-delay

4.3 AppIr – Applicativo GUI MATLAB

In questo paragrafo verrà presentata l'applicativo denominato AppIr, sviluppato in ambiente MATLAB. Come primo argomento trattato, verrà presa in esame l'interfaccia grafica dell'applicativo; successivamente verrà descritto il codice MATLAB necessario al suo corretto funzionamento.

4.3.1 Interfaccia grafica

L'interfaccia grafica dell'applicazione AppIr è pensata tenendo sempre conto dei dati ottenuti dal questionario. Per implementarla è stato usato GUIDE: è un ambiente di sviluppo GUI che fornisce gli strumenti per progettare interfacce utente destinate ad applicazioni personalizzate. Il Layout Editor di GUIDE consente di progettare graficamente la propria GUI tramite semplici passaggi. Una volta scelte le varie componenti tramite un menù grafico, GUIDE genera automaticamente il codice MATLAB per costruire l'UI, che verrà poi modificato per programmare il comportamento dell'applicazione.

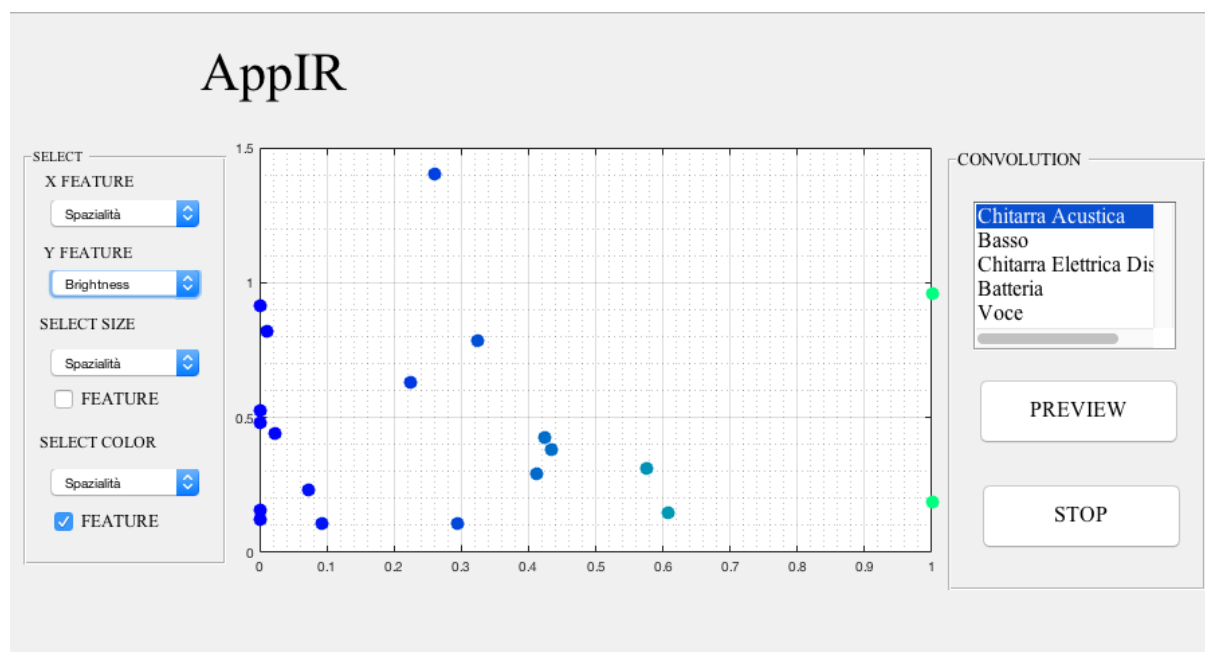


Figura 4.3.1: Interfaccia Grafica dell'applicazione AppIr

Inizializzata l'applicazione, apparirà un'interfaccia grafica composta da un grafo cartesiano su cui saranno plottati le IR del dataset, 4 menu a tendina, 2 check box, una lista di scelta e infine due pulsanti per compiere le azioni di “PREVIEW” e “STOP”.

Ora verranno spiegati i componenti dell'interfaccia grafica.

Nel pannello SELECT sono presenti:

- **X FEATURE:** menù a tendina che permette di scegliere una tra le otto features acustiche e di posizionarle sull'asse delle x. Con l'asse delle y impostata, si otterranno le coordinate per plottare le n IR a seconda delle caratteristiche scelte.
- **Y FEATURE:** menù a tendina che permette di scegliere una tra le otto features acustiche e di posizionarle sull'asse delle y. Con l'asse delle x impostata, si otterranno le coordinate per plottare le n IR a seconda delle caratteristiche scelte.
- **SELECT SIZE:** menù a tendina che permette di scegliere la grandezza relativi alle IR. La grandezza verrà stabilita dalla feature acustica scelta; più sarà di grandi dimensioni, maggiore sarà il valore della caratteristica acustica per quella medesima.
- **SELECT COLOR:** menù a tendina che permette di scegliere il colore relativo alle IR. Il colore verrà stabilito dalla feature acustica scelta. La colormap utilizzata è “winter”: i colori vanno dal verde chiaro fino al blu scuro. Più avrà un colore chiaro, maggiore sarà il valore della caratteristica acustica per quella medesima.
- **CHECKBOX SIZE FEATURE:** quando è selezionata, verrà visualizzata la grandezza relativa alla feature scelta. Se la check box non sarà selezionata, verrà mostrato un valore di default.
- **CHECKBOX COLOR FEATURE:** quando è selezionata, verrà visualizzato il colore relativo alla feature scelta. Se la check box non sarà selezionata, verrà mostrato un valore di default.

Nel pannello CONVOLUTION sono presenti:

- **MENÙ A LISTA:** permette di scegliere tra un setlist di sample audio. È possibile scegliere tra CHITARRA ACUSTICA, BASSO, CHITARRA ELETTRICA DISTORTA, BATTERIA e VOCE (Femminile).
- **BUTTON PREVIEW:** premendo il pulsante è possibile effettuare la convoluzione tra l'IR scelta sul grafo e il sample audio dalla menù a lista. Verrà poi riprodotto via audio il sample convoluto.
- **BUTTON STOP:** premendo il pulsante, fa terminare la riproduzione audio del sample convoluto.

Questo sistema è stato scelto per poter avere quattro caratteristiche contemporaneamente visualizzate, aiutando e rendendo più veloce la scelta delle IR durante l'utilizzo dell'applicazione.

4.3.2 Codice

Una volta salvata l'interfaccia grafica, GUIDE automaticamente crea una struttura default di codice riservata alle **callback function** delle varie componenti dell'interfaccia grafica.

In programmazione, un callback (o, in italiano, **richiamo**) è, in genere, una funzione, o un "blocco di codice" che viene passata come parametro ad un'altra funzione. In particolare, quando ci si riferisce alla callback richiamata da una funzione, la callback viene passata come argomento ad un parametro della funzione **chiamante**. In questo modo la chiamante può realizzare un compito specifico (quello svolto dalla callback) che non è, molto spesso, noto al momento della scrittura del codice. Queste callback function entrano in funzione quando c'è un'interazione tra utente ed interfaccia grafica per ogni componente. Ogni callback contiene le variabili hObject e handles. La prima variabile serve per prelevare eventuale oggetto creato da un'azione dell'utente, mentre la handles è una struttura dati dove è possibile salvare variabile che verranno utilizzate al di fuori della funzione stessa.

A titolo d'esempio viene mostrato il codice relativo alla callback function del pulsante di "PREVIEW":

```
function Preview_Pushbutton_Callback(hObject, eventdata, handles)
% hObject      handle to Preview_Pushbutton (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)

%carico indice.mat e salvo indice che mi servirà per pescare ir
selezionata
%su plot con datacursor.
load('indice.mat')
handles.index = index;

%recupero percorso_cartella e name da lista
percorso_cartella = handles.lista_ir.folder;
name = handles.lista_ir(index).name;

%localizzo ir selezionato
ir = fullfile(percorso_cartella,name);

%Import sample wav
[x,fs] = audioread(handles.wav);

%Import Impulse Response
h = audioread(ir);

%Find n_channels
n_canali_ir = size(h,2);

% Convolution Stereo or Mono case
if n_canali_ir == 2
    %Stereo IR
    h_Left = h(:,1);
    h_Right = h(:,2);

%Left and Right Channels of Sample wav
    x_Left = x(:,1);
    x_Right = x(:,2);

    % %Perform Convolution for Stereo IR
    out_L = fconv(x_Left, h_Left);
    out_R = fconv(x_Right, h_Right);

    %Create Stereo Output
    out = [out_L, out_R];
else
    out = fconv(x,h);
end

handles.player=audioplayer(out,fs);
play(handles.player);

%save and update handles
guidata(hObject,handles);

```

In questa parte di codice, appena premuto il pulsante, viene caricato il file “indice.mat” (contenente l’indice delle IR selezionato sul grafo) e recuperato tramite l’utilizzo di quest’ultimo assieme al percorso e al nome contenuti nel file “lista_ir.mat”. Una volta recuperato l’IR necessaria, viene letto il file sample precedentemente selezionato dalla menù a lista con la funzione **audioread()**. Viene valutato poi il numero di canali dell’IR: se sarà stereo verrà effettuata una convoluzione per il canale destro e il canale sinistro del sample

audio tramite la funzione **fconv** (che verrà spiegata di seguito) ed infine creato un oggetto “out” contenente i due canali convoluti; se l’IR sarà mono, verrà effettuata una **fconv** direttamente tra il sample e l’IR. Sarà poi riprodotto via audio l’oggetto player tramite la funzione **audioplayer()**. Infine saranno aggiornati i dati tramite la funzione **guidata()**.

Ora tratteremo la Fast Convolution, che è stata implementata con la funzione **fconv()**. Si è deciso di utilizzarla poiché l’impiego della funzione default di MATLAB **conv()** impiegava troppo tempo di calcolo per effettuare la convoluzione. Viene ora presentato la parte di codice dedicata:

```
function [ y ] = fconv(x, h)
%FCONV Fast Convolution
% [y] = FCONV(x, h) effettua la convoluzione di x e h, e
normalizza l'output
%      a +/-1.
%
%      x = input vector
%      h = input vector

Ly=length(x)+length(h)-1;
% Find smallest power of 2 that is > Ly
Ly2=pow2(nextpow2(Ly));
% Fast Fourier transform
X=fft(x, Ly2);
% Fast Fourier transform
H=fft(h, Ly2);
Y=X.*H;
% Inverse fast Fourier transform
y=real(ifft(Y, Ly2));
% Take just the first N elements
y=y(1:1:Ly);
% Normalize the output
y=y/max(abs(y));

end
```

Codice Matlab: Fast Convolution

Le prime righe di codice cercano la più piccola potenza di 2 che sia maggiore della variabile Ly. Successivamente viene effettuata la Fast Fourier Transform ai vettori X ed H, che rappresentano, oltre all’IR, i canali sinistro e destro per il sample audio se la convoluzione è stereo, altrimenti sample audio e relativa IR nel caso in cui sia mono. Viene poi effettuata la Fast Fourier inversa e normalizzato l’output. L’implementazione di questa funzione ottimizza i tempi di calcolo ed elaborazione nella fase di convoluzione.

Verranno trattate ora le funzioni `scaleSize()` e `scaleColor()` implementate entrambe per normalizzare i dati delle features acustiche durante la scelta di size e color. Non utilizzando queste due funzioni si introdurrebbero eventuali valori minori di zero, che potrebbero comportare delle anomalie durante la creazione del plot sul grafo.

```
function [ size ] = scaleSize( size_feature )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

size_feature = 1+size_feature-min(size_feature);
size_feature = 200*size_feature/max(size_feature);

size = size_feature;

end
```

Codice Matlab: funzione `scalarSize`

Per la creazione del cursore sul grafo è stata utilizzata la funzione **`datacursormode`** e creata una funzione personalizzata, per il popup, visualizzabile una volta scelta l'IR sul grafo.

```
function txt = myupdatefcn(empty,event_obj)
% Display the position of the data cursor
% obj          Currently not used (empty)
% event_obj    Handle to event object
% output_txt   Data cursor text string (string or cell array of
strings).

load('matrice.mat')
index = get(event_obj, 'DataIndex');
%salvo l'indice corrente dell'ir selezionato
save('indice.mat','index');

name = lista_file(index).name;

txt = { ['IR n°: ', num2str(index)], ...
        ['Name: ', name] };
```

Una volta caricata la `matrice.mat` con la funzione **`load`**, viene salvato nella variabile `index` l'indice dell'IR, salvato nel file `indice.mat` ed infine recuperato il nome per essere visualizzato sul grafo cartesiano.

Viene presentato infine la prima parte di codice della `OpeningFunction`: qui viene inizializzata l'applicazione, caricando la matrice e preparando le variabili `handles` che serviranno come variabili di default durante il lancio di `AppIr`.

```
function AppIr_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to AppIr (see VARARGIN)

%carico la matrice contenente le features degli ir
load('matrice.mat')
handles.matrice_features = matrice_features;

%carico la lista contenente percorso e nomi degli ir da
utilizzare
handles.lista_ir = lista_file;

%default size
handles.size = 100;

%default color
handles.color = [0,1,0];

%default check_size and check_color
handles.check_size = 0;
handles.check_color = 0;

%assegno a x e y due valori di default ( space_correlation )

handles.x = handles.matrice_features(:,1);
handles.y = handles.matrice_features(:,1);

%creo lo scatter plot
scatter(handles.x,handles.y,handles.size,handles.color,'filled'
);
grid on;
grid minor;
box on;

%abilito creo un oggetto datacursormode e setto updatefcn
dcm_obj = datacursormode();
set(dcm_obj,'UpdateFcn',@myupdatefcn);

%LOAD SAMPLES AND RELATIVE PATH
handles.percorso_samples =
'/Users/Fred/Documents/MATLAB/Tesi/App/sample/';
handles.lista_samples = dir([handles.percorso_samples
'*.*wav']);
% Find number of files in the folder
n_samples = length(handles.lista_samples);
% Error if i don't find any file
if( ~n_samples )
    error(['No files found in ' handles.percorso_samples ]);
end
%Set default sample .wav
handles.wav = [handles.percorso_samples,
handles.lista_samples(1).name];

% Choose default command line output for AppIr
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

Codice MATLAB: Opening Function dell'applicazione AppIr

5.1 Utilizzo di AppIr

Si è passati alla fase di test di AppIr. Ai candidati è stata sottoposta l'applicazione e, dopo una breve spiegazione dell'interfaccia grafica e del suo compito, è stato chiesto di testarla al fine valutarne l'approccio diretto e l'esperienza avuta utilizzando l'applicazione, sia negativa che positiva.

Durante il test, ogni candidato è stato dotato di:

- un paio di cuffie: necessarie per l'ascolto della fase di convoluzione, in modo da poter effettuare un paragone tra le IR, sia stereo che mono, contenuti nel dataset e rendere il migliore possibile l'esperienza di utilizzo di AppIr.
- Un computer con installato Matlab per poter far girare l'applicativo correttamente.
- Un questionario di otto domande da compilare una volta terminato l'utilizzo di AppIr.

Il candidato ha potuto così esplorare l'applicazione nei minimi particolari, utilizzando i vari menù delle features acustiche per creare lo scatterplot desiderato; selezionare le varie IR sul grafo e effettuare un paragone in tempo reale con i vari sample audio a disposizione forniti dallo sviluppatore, visualizzando per ogni IR le caratteristiche acustiche desiderate. Per aver una maggiore accuratezza il test è stato svolto su 5 file audio differenti, una voce cantata, una chitarra acustica arpeggiata, una chitarra distorta, un riff di basso e infine un fill di batteria precedentemente campionati, in modo da fornire un'ampia scelta durante i paragoni in tempo reale delle varie IR proposte. Questo perché AppIr è un prototipo: nella versione finale sarà l'audio in ingresso a fornire materiale di preview.

5.2 Questionario

Una volta terminato il tempo di test è stata richiesta la compilazione di un questionario. Vengono elencate ora le domande a cui candidati hanno dovuto rispondere:

1. In che modo impiegate le IR (Ingegnere Del Suono, Ricerca Accademica, Sound Design, Amatore)?
2. Quali IR utilizzate durante le vostre sessioni di lavoro (Riverbero Naturale, Riverbero Artificiale, Riproduzione dell'Equalizzatore, Simulazione di Altoparlanti e Dispositivi)?
3. L'interfaccia grafica dell'applicazione AppIr ti sembra funzionale per il compito che devi svolgere?
4. Hai avuto difficoltà nell'utilizzo dell'applicazione?
5. Credi che per come è strutturata e per come funziona l'applicazione sia utile nelle tue sessioni di lavoro?
6. Hai riscontrato degli errori, bug o altri comportamenti imprevisti durante l'utilizzo dell'applicazione? Se "sì" potresti indicare quali?
7. Hai suggerimenti su come migliorare l'applicazione in modo da essere il più funzionale possibile nelle tue sessioni di lavoro?
8. Se questa applicazione fosse implementata come plugin e utilizzabile all'interno di un Daw la useresti? Per quale motivo?

Risultati

Si è svolto il questionario su un campione di 10 persone provenienti da diversi ambiti lavorativi, ovvero Ingegnere del suono, Sound Designer, Ricercatori Accademici ed Amatori.

Per il 90% dei candidati la tipologia di IR maggiormente impiegate è risultata essere Riverbero Artificiale e Naturale, mentre il restante 10% ha affermato di impiegare anche IR relative alla Riproduzione dell'Eq e Simulazione di Altoparlanti e Dispositivi.

Il 100% dei candidati ha potuto confermare la funzionalità effettiva e il compito che l'applicazione si era proposta, raggiungendo così un riscontro molto positivo tra i soggetti.

Durante la fase di test, il 20% dei candidati ha riscontrato delle difficoltà iniziali nell'utilizzo, ma dopo qualche minuti di utilizzo è risultata comprensibile e chiara. Questo, molto probabilmente, dovuto ai primi momenti di ambientazione con l'interfaccia grafica che il soggetto ha riscontrato ad inizio test.

Nei test non sono stati riscontrati bug o errori che abbiano pregiudicato il buon funzionamento dell'applicazione, anche se sono state suggerite eventuali modifiche (come una visualizzazione più dettagliata delle caratteristiche delle IR, implementazione di un loop per l'ascolto dei paragoni, semplice tutorial iniziale per ambientarsi meglio con l'interfaccia grafica) per una migliore ottimizzazione dell'applicazione AppIr.

Infine si è raggiunta la totalità e il consenso da parte del 100% dei candidati riguardo l'utilizzo dell'applicazione, giustificando che l'impiego di AppIr renderebbe più facile e snello il workflow nella scelta di IR da impiegare durante le sessioni di lavoro.

Tipologia Impulse Response	Percentuale
Riverbero Naturale ed Artificiale	90 %
Riverbero Naturale, Artificiale, Riproduzione dell'Eq e Simulazione di Altoparlanti	10 %

Difficoltà nell'utilizzo di AppIr	Percentuale
Utenti che hanno incontrato difficoltà iniziali nell'utilizzo	20 %
Utenti che non hanno avuto difficoltà iniziali nell'utilizzo	80 %

AppIr è risultato funzionale per il compito prefissato	Percentuale
Utenti con giudizio positivo	100 %
Utenti con giudizio negativo	0 %

Figura 5.2: Risultati Questionario Finale

CONCLUSIONI E SVILUPPI FUTURI

6.1 Conclusioni

Nella fase di test si è notato come, ad un approccio iniziale, l'interfaccia grafica dell'applicazione abbia richiesto alcuni minuti di ambientazione da parte dei candidati. Nonostante questo episodio, i candidati hanno saputo utilizzare l'applicazione a piena funzionalità, dando un riscontro positivo riguardo alla progettazione ed implementazione di quest'ultima. Per alcuni candidati, la sola visualizzazione grafica delle IR non è stata sufficiente ed ha portato a suggerimenti per una migliore implementazione futura dall'applicazione. In linea generale, la totalità dei candidati hanno giudicato in maniera molto positiva il lavoro svolto riguardo alla progettazione e, cosa di rilevante importanza, allo scopo che si era prefissato ad inizio tesi: l'applicazione permette un preascolto ed un paragone tra le diverse IR, facilitando la scelta in ambito lavorativo e velocizzando il workflow di mixaggio di un brano musicale.

6.2 Sviluppi futuri

Il problema di ambientazione iniziale, espresso nelle conclusioni, potrebbe essere risolto tramite la creazione di un mini tutorial, dove vengono spiegate le varie componenti grafica dell'interfaccia e il loro corretto utilizzo. In più, l'aggiunta di un loop renderebbe ancora più funzionale la fase di preascolto e paragone tra le varie IR.

Sviluppi futuri dell'applicazione sono legati all'implementazione di un Plug-in VST/AU che possa rispecchiare le caratteristiche descritte in questo elaborato, aggiungendo la funzione preascolto sul materiale audio appena registrato in un DAW, oltre all'impiego di tag e voti in modo da rendere ancora più veloce la ricerca delle IR recentemente usate o di maggiore interesse.

BIBLIOGRAFIA

- [1] Alberto Bertoni, Paola Campadelli and Giuliano Grossi. "Introduzione all'elaborazione dei segnali" 30/09/2010
- [2] Laboratorio di Informatica Musicale modulo 8: Manipolazione del Segnale Audio – Music Inf. Retrieval 29/03/2017
- [3] Presti, G., and Davide Andrea Mauro. "Continuous brightness estimation (CoBE): Implementation and its possible applications." (2013).

[1] <http://www.acousticfrontiers.com/early-reflections-101/> data: 11/01/2018

[2] <https://it.mathworks.com/help/matlab/ref/conv.html> data: 15/02/2018

[3] <https://it.mathworks.com/matlabcentral/fileexchange/5110-fast-convolution?focused=5057142&tab=function> data: 25/02/2018

[4] <https://it.mathworks.com/discovery/matlab-gui.html> data: 12/03/2018