



UNIVERSITÀ DEGLI STUDI DI PARMA  
FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN  
INGEGNERIA INFORMATICA

ELABORAZIONE DI MAPPE ACUSTICHE  
TRAMITE ARRAY MICROFONICI

Relatore:  
Chiar.mo Prof. ANGELO FARINA

Correlatore:  
Dott. Ing. SIMONE CAMPANINI

Tesi di laurea di:  
LORENZO ROTTEGLIA

ANNO ACCADEMICO 2012–2013

*alla mia famiglia.*

# Indice

<b>1</b>	<b>Premessa</b>	<b>11</b>
<b>2</b>	<b>Elementi di acustica</b>	<b>14</b>
2.1	Introduzione . . . . .	14
2.2	Grandezze fondamentali . . . . .	14
2.2.1	Frequenza e Periodo . . . . .	15
2.2.2	Velocità di propagazione e lunghezza d'onda . . . . .	16
2.3	Onde acustiche . . . . .	17
2.3.1	La scala dei deciBel . . . . .	18
2.3.2	Livello di pressione . . . . .	19
2.3.3	Livello equivalente . . . . .	20
2.4	Sistema uditivo umano . . . . .	21
2.4.1	Effetti di non linearità dell'orecchio umano . . . . .	22
2.4.2	Bande frequenziali . . . . .	23
2.4.3	Filtri di ponderazione . . . . .	27
<b>3</b>	<b>Array microfonici</b>	<b>28</b>
3.1	Array sferico: <i>Eigenmike<sup>TM</sup> EM32</i> . . . . .	32
3.1.1	Composizione del sistema . . . . .	33
3.1.2	Limiti intrinseci teorici del sistema . . . . .	36
3.2	Altri array . . . . .	38
3.2.1	Array lineari . . . . .	38

3.2.2	Array planari . . . . .	38
3.2.3	Array cilindrici . . . . .	39
<b>4</b>	<b>Sintesi di microfoni virtuali</b>	<b>41</b>
4.1	Sistema di Elaborazione . . . . .	41
4.1.1	Approccio innovativo di inversione numerica . . . . .	44
<b>5</b>	<b>Ambiente di sviluppo: l'host <i>Audacity</i><sup>®</sup></b>	<b>46</b>
5.1	Libreria per la costruzione di interfacce grafiche <i>wxWidgets</i> . . . . .	48
5.2	Altre librerie . . . . .	49
5.3	Libreria di codifica audio e video: <i>FFmpeg</i> . . . . .	51
<b>6</b>	<b>Descrizione plug-in: un manuale d'uso</b>	<b>53</b>
6.1	Selezione dell'intervallo temporale da analizzare . . . . .	54
6.2	Finestra di configurazione . . . . .	55
6.3	Interfaccia principale . . . . .	57
<b>7</b>	<b>Dettagli di Implementazione</b>	<b>61</b>
7.1	Acquisizione dei dati in ingresso . . . . .	62
7.2	Organizzazione del carico computazionale . . . . .	63
7.3	Finestratura dei dati di ogni frame audio . . . . .	65
7.4	<i>Mirroring</i> dei microfoni virtuali . . . . .	65
7.5	<i>Meshing</i> della superficie mediante l'operazione di triangolazione di <i>Delaunay</i> . . . . .	67
7.6	Filtraggio in bande d'ottava . . . . .	70
7.7	Applicazione del livello di fondo scala . . . . .	72
7.8	Interpolazione dei valori acquisiti . . . . .	73
7.9	Mappatura <i>colormap</i> e <i>auto-scale</i> . . . . .	75
7.10	Sincronizzazione audio-video . . . . .	76
7.11	Esportazione dei risultati . . . . .	77

---

<b>8 Conclusioni</b>	<b>79</b>
<b>A Richiami teorici</b>	<b>81</b>
A.1 Propagazione di onde sonore . . . . .	81
A.2 Intensità, densità di energia e potenza . . . . .	83
A.3 Livelli sonori . . . . .	84
A.4 Curve di ponderazione . . . . .	86
A.5 Trasformata di Fourier . . . . .	86

# Elenco delle figure

1.1	Esempio di mappa acustica generata dal plug-in <i>Microphone Array Analyzer</i>	12
2.1	Moto delle particelle di un fluido elastico compresse e decomprese alternativamente da un pistone, che producono l' <i>onda sonora</i> . . . . .	15
2.2	Livello sonoro di una sorgente intermittente e relativo livello equivalente $L_{EQ}$ . . . . .	20
2.3	Apparato acustico umano. . . . .	21
2.4	Risposta non lineare della coclea . . . . .	22
2.5	Grafico delle curve isofoniche . . . . .	23
2.6	Grafico delle curve di ponderazione . . . . .	27
3.1	Tecniche di registrazione con array discreti di tipo INA-5 . . . . .	29
3.2	Tecniche di registrazione con array discreti di tipo OCT surround . . . . .	29
3.3	Il microfono <i>SoundField™</i> di Gerzon e Craven . . . . .	30
3.4	Diagramma polare delle armoniche <i>Ambisonics</i> . . . . .	31
3.5	Struttura del sistema <i>Eigenmike™ EM32</i> . . . . .	32
3.6	Disposizione delle capsule dell'array microfonico <i>Eigenmike™ EM32</i> . . . . .	33
3.7	Angoli di riferimento e loro direzioni relative per la descrizione sferica rispetto al sistema di riferimento antropometrico secondo la norma ISO2631. . . . .	34

3.8	Array lineare e relativo diagramma polare a 1 kHz . . . . .	38
3.9	Array planare e relativa mappa acustica che rappresenta i livelli sonori crescenti dal blu al rosso e in cui si riconosce una sorgente sonora come addensamento di rosso . . . . .	39
3.10	Array cilindrico (a), con relativa sezione verticale (b) . . . . .	40
4.1	Ogni microfono virtuale può essere sempre descritto da una somma di convoluzioni di tutti gli ingressi per un opportuno filtro FIR .	42
4.2	Possibili microfoni virtuali a ordine di direttività crescente (sulla destra) calcolati a partire da armoniche sferiche <i>Ambisonics</i> fino al terzo ordine (sulla sinistra). . . . .	43
4.3	Diagramma polare di un microfono virtuale cardioide in funzione dell'ordine di direttività. . . . .	44
5.1	Interfaccia utente dell'host utilizzato, <i>Audacity</i> <sup>®</sup> sotto diverse piattaforme. . . . .	47
5.2	Intorno per la costruzione di interfacce grafiche <i>wxFormBuilder</i> .	50
6.1	Selezione dello spezzone di materiale audio da analizzare con il <i>plug-in</i> . . . . .	54
6.2	Messaggio di errore in caso di memoria insufficiente per le operazioni di <i>precalcolo</i> . . . . .	55
6.3	Finestra di configurazione del modulo <i>Microphone Array Analyzer</i>	56
6.4	Finestra principale del modulo <i>Microphone Array Analyzer</i> . . .	58
7.1	Finestra di Hanning nel dominio del tempo per l'analisi digitale.	66
7.2	Rappresentazione schematica del processo di <i>mirroring</i> , necessario a garantire la continuità ai bordi della mappa. . . . .	67
7.3	Direzioni di puntamento delle capsule dell' <i>Eigenmike</i> <sup>TM</sup> sovrapposte ad una foto panoramica del Teatro alla Scala di Milano, svolta secondo lo schema di proiezione visto nella sezione 7.4. . . . .	68

7.4 Descrizione della proprietà dei triangoli formati con l'algoritmo di <i>Delaunay</i> . . . . .	69
---	----

# **Elenco delle tavole**

2.1	Velocità del suono in alcuni mezzi. . . . .	17
2.2	Valori di pressione e di SPL relativi a alcuni suoni comuni. . . . .	19
3.1	Coordinate sferiche antropometriche delle direzioni delle capsule fornite sia in gradi che in radianti secondo la norma ISO2631 descritta in figura 3.7 . . . . .	35

# Capitolo 1

## Premessa

Il presente progetto di tesi si pone l'obiettivo di mappare il campo acustico dinamico di un ambiente (interno o esterno) tramite una sonda, nella fattispecie un *array microfonico*. Il risultato richiesto quindi è quello di ottenere un filmato composto dalla sovrapposizione di:

**un videoclip di sfondo** ottenuto da una particolare videocamera, che rappresenta l'ambiente circostante e contenente quindi un'informazione visiva.

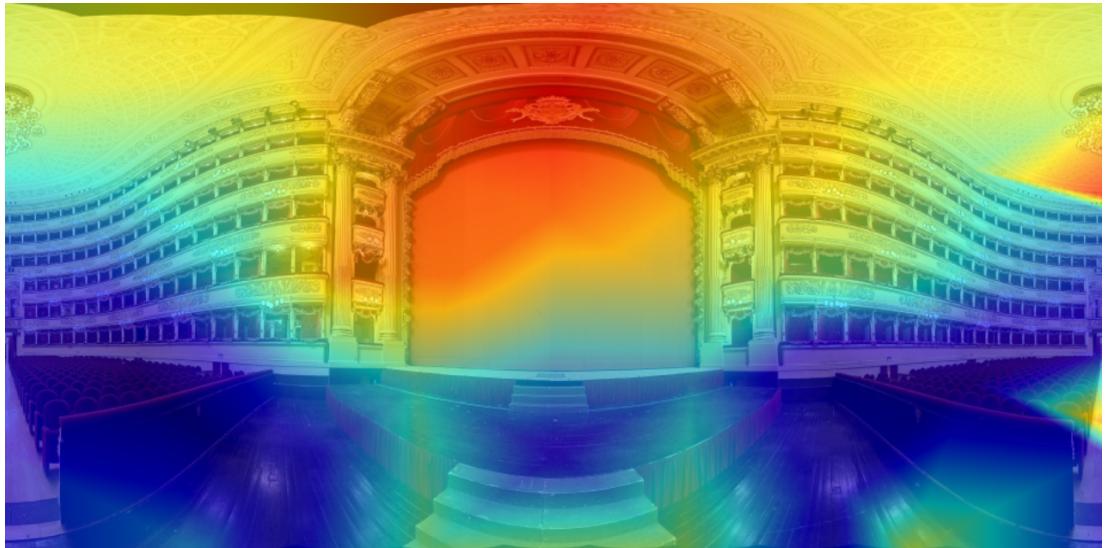
**la mappa acustica dinamica** composta di bande colorate rappresentanti i livelli sonori istantanei nel campo acustico.

**eventuali metadata** quali le posizioni delle singole capsule microfoniche dell'array, o i valori dei livelli sonori in determinate posizioni di interesse.

Il problema è stato affrontato programmando un software di nome *Microphone Array Analyzer*. Si tratta di un *plug-in* scritto per l'ambiente di editing audio *Audacity*<sup>®</sup><sup>1</sup>, software *open-source* molto versatile, dalle possibilità molto ampie per permettere gli utilizzi più diversi. Il software sviluppato in questa tesi, dialogando con l'*host Audacity*<sup>®</sup>, acquisisce i dati in ingresso e genera da essi una *mappa acustica*, composta da bande di colore che corrispondono ai diversi livelli

---

<sup>1</sup> <http://audacity.sourceforge.net/?lang=it>



**Figura 1.1:** Esempio di mappa acustica generata dal plug-in *Microphone Array Analyzer*

sonori attorno all’array sonda, posto nell’ambiente di misura.

Questo tipo di *tool* può avere numerose applicazioni tecnologiche in quanto è in grado di rendere visibile i campi sonori i quali contengono invece una informazione di tipo uditivo. La potenza di questa operazione sinestetica di traduzione di un’informazione uditiva in una visiva risiede principalmente nella maggior apprezzabilità delle grandi qualità di definizione spaziale degli array microfonici.

Dal punto di vista applicativo, il sistema sviluppato può essere di estrema utilità nella precisa individuazione di sorgenti sonore, nonchè nella misura delle loro emissioni. Basti pensare per esempio ad ambienti di tipo industriale, nei quali spesso il campo acustico è complesso e generato da molteplici e diverse sorgenti che spesso risultano difficili da individuare con precisione. Un altro esempio calzante riguarda gli ambienti molto ristretti come gli abitacoli di automobili o aereoplani il cui confort è una specifica di progetto che attualmente ricopre un notevole interesse.

In particolare, la principale espansione svolta in questa tesi è stata quella di rendere il software in grado di generare una mappa *dinamica*, riflettendo i cambiamenti del campo acustico attorno alla sonda istante per istante e sovrapponendo

questo risultato a un video di sfondo, posto in trasparenza e acquisito mediante telecamere installate appositamente sulle sonde utilizzate e raffigurante l'ambiente stesso di misura.

Di seguito, in sintesi, il contenuto:

**Capitolo 2** Richiami teorici di acustica di base e riguardo alcune proprietà fisiche del suono, ausilio fondamentale a tutta la trattazione successiva.

**Capitolo 3** Introduzione agli array microfonici e alle loro proprietà spaziali di direttività e direzionalità nella misura dei campi acustici.

**Capitolo 4** Introduzione al concetto di microfono virtuale e relativa sintesi, ovvero possibilità offerte da questo artificio, nella fattispecie riguardo alla descrizione spaziale di un campo acustico.

**Capitolo 5** Descrizione dell'ambiente software definito dal programma *host* scelto (*Audacity<sup>®</sup>*) e delle librerie esterne utilizzate.

**Capitolo 6** Breve manuale d'uso per un utilizzatore del *plug-in*. Possibili analisi e parametri significativi.

**Capitolo 7** Dettagli sul funzionamento interno, algoritmi utilizzati e flusso di lavoro.

# **Capitolo 2**

## **Elementi di acustica**

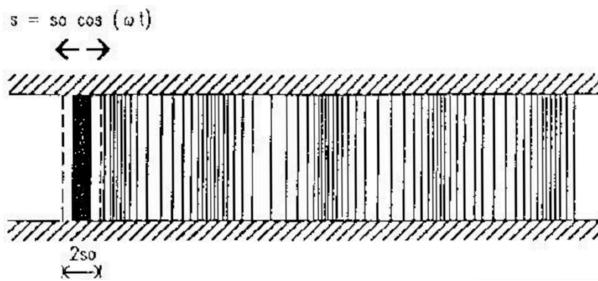
### **2.1 Introduzione**

Nel presente capitolo verranno fornite le nozioni di base riguardanti i principi fisici fondamentali della fisica del suono.

Partendo dalle grandezze fondamentali che caratterizzano l'analisi del fenomeno sonoro, si parlerà della descrizione delle diverse onde sonore per arrivare a una descrizione di campo acustico di cui si cerca di fornire una mappa visiva.

### **2.2 Grandezze fondamentali**

Il fenomeno sonoro si genera dalla variazione di pressione in un mezzo materiale elastico (fluido o solido) la quale si propaga senza trasporto di materia. Le molecole del mezzo in cui si propagano i suoni, si muovono parallelamente alla direzione di propagazione. Un'onda sonora che viaggi attraverso l'aria non è altro che una successione di rarefazioni e compressioni di piccole porzioni d'aria; ogni singola molecola trasferisce energia alle molecole adiacenti e, dopo il passaggio dell'onda, ritorna pressappoco nella sua posizione iniziale. Consideriamo l'esempio di un fluido che viene compresso dal movimento armonico di un pistone



**Figura 2.1:** Moto delle particelle di un fluido elastico compresse e decomprese alternativamente da un pistone, che producono l'*onda sonora*.

come rappresentato in figura 2.1. Le particelle del fluido adiacenti alla superficie del pistone saranno spostate dal suo movimento verso le particelle prossime ad esse e si verificherà quindi una compressione locale. Subito dopo il pistone retrocederà lasciando uno spazio libero dietro di sé che tenderà a richiamare le particelle che erano state spinte inizialmente, mentre le particelle adiacenti ad esse continueranno il loro movimento nella direzione opposta creando una regione di decompressione locale attigua al pistone. Questo andamento alternativo è responsabile della creazione del fenomeno ondulatorio, il quale viene denominato *onda sonora* nel momento in cui viene percepito dalle nostre orecchie.

### 2.2.1 Frequenza e Periodo

Riferendoci per comodità all'esempio appena visto, definiamo che il moto armonico del pistone è caratterizzato dalla frequenza  $f$  con cui la superficie piana si muove. La frequenza indica il numero di cicli compiuti dalla superficie piana in un secondo e viene espressa in *Hertz* (Hz); il tempo necessario a compiere uno di questi cicli è detto *periodo* e viene indicato con il simbolo  $T$ . Risulta intuitivo quindi capire che *frequenza* e *periodo* siano in un rapporto di proporzionalità inversa.

Se la frequenza del fenomeno è compresa tra 20 e 20 000 Hz, la perturbazione è percepibile dall'orecchio umano e si parla quindi di *fenomeno acustico* o *sonoro*.

Un suono caratterizzato da una funzione di variazione di pressione sinusoidale a *frequenza* costante, come quello dell'esempio, è detto *tono puro*. I rumori presenti in natura invece sono formati da varie componenti tonali pure sovrapposte. Queste onde sovrapposte concorreranno quindi nel formare una *forma d'onda* complessa che avrà frequenza proporzionale alle frequenze componenti.

Il nostro orecchio è particolarmente abile nel percepire le variazioni di frequenza e le relazioni tra le varie componenti frequenziali presenti in un suono. Queste componenti vengono dette *armoniche*. Se le armoniche sono a frequenze multiple intere le une delle altre, allora si avrà un effetto musicale piacevole detto *consonanza*, in caso contrario si otterrà un suono poco armonioso (addirittura in certi casi dall'altezza indeterminata) detto *dissonanza*.

### 2.2.2 Velocità di propagazione e lunghezza d'onda

Le proprietà elastiche e la massa del mezzo fluido stabiliscono la *velocità*  $c$  con cui la perturbazione si trasmette. Troviamo nella tabella 2.1 la velocità di propagazione sonora in differenti mezzi elastici.<sup>1</sup> Tenendo come riferimento sempre l'esempio del pistone oscillante di figura 2.1, definiamo come *lunghezza d'onda*  $\lambda$  la distanza tra due picchi di compressione del mezzo elastico.

Perciò, fissata la frequenza  $f$  del moto armonico del pistone,  $\lambda$  dipende dal valore della velocità di propagazione  $c_0$  secondo la relazione:

$$\lambda = \frac{c_0}{f}, \quad (2.1)$$

o alternativamente sfruttando la proporzionalità inversa tra *frequenza* e *periodo*:

$$\lambda = c_0 T. \quad (2.2)$$

---

<sup>1</sup> Dati forniti dal manuale [8]

Materiale	Densità $\rho$ [kg /m <sup>3</sup> ]	Velocità del suono $c$ [m /s]
Acciaio	7800	5000
Alluminio	2700	5820
Calcestruzzo	2000 ÷ 2600	3500 ÷ 5000
Gomma	1010 ÷ 1250	35 ÷ 230
Legno	400 ÷ 700	3300
Marmo	2600	3800
Mattoni pieni	2100	3600
Piombo	11300	1260
Rame	8900	4500
Sabbia	1600	1400 ÷ 2600
Stagno	7280	4900
Sughero	240	480
Vetro	2300 ÷ 5000	4000 ÷ 5000
Zinco	7100	3750

Tabella 2.1: Velocità del suono in alcuni mezzi.

## 2.3 Onde acustiche

I fenomeni acustici consistono quindi essenzialmente in una perturbazione di pressione che si propaga in un mezzo elastico. Ciò che caratterizza il fenomeno è l'entità di questa perturbazione rispetto a un valore di equilibrio preso come riferimento. Nel caso applicativo più frequente, la propagazione nell'aria, si prende come riferimento la pressione atmosferica.

Essendo  $P_0$  la pressione di riferimento del mezzo, la *pressione acustica istantanea* viene definita come segue:

$$p(t) = p'(t) - P_0 \quad (2.3)$$

dove  $p'(t)$  è il valore di pressione atmosferica nell'istante  $t$  in un punto dato in cui si vuole misurare la pressione acustica.

### 2.3.1 La scala dei deciBel

Il valore minimo di perturbazione di pressione  $p_0$  udibile varia da persona a persona, ma il valore standard considerato come riferimento di misura è:

$$p_0 = 20 \text{ } [\mu\text{Pa}] \quad (2.4)$$

Invece i valori massimi a cui il sistema uditivo può venire sottoposto sono dell'ordine delle centinaia di pascal (Pa). Normalmente viene considerata come soglia del dolore  $P_{max} = 20 \text{ Pa}$ . Si pensi però che alcuni rumori comuni superano di gran lunga questa soglia.

Sono stati riportati nella tabella 2.2 alcuni valori di pressione relativi a suoni comuni. Dunque è evidente che la dinamica a cui è sottoposto l'apparato acustico umano, e che quindi è necessario misurare, è molto ampia. Infatti:

$$\frac{P_{max}}{P_{min}} = \frac{200 \text{ } [Pa]}{2 \cdot 10^{-5} \text{ } [Pa]} = 10^7 \quad (2.5)$$

Per evitare quindi di utilizzare sette ordini di grandezza, nonché per rispettare le proprietà di non linearità della risposta dell'apparato acustico umano<sup>2</sup>, viene introdotta la scala logaritmica dei *deciBel*. Data una grandezza da misurare  $G$  e un suo valore di riferimento  $g_0$  rispetto al quale effettuare la misura, si definisce il *deciBel* ( $dB$ ) come:

$$\text{grandezza } G \text{ } [dB] = 10 \cdot \log \left( \frac{G}{g_0} \right) \quad (2.6)$$

Perciò il dB non rappresenta un'unità di misura ma semplicemente una scala logaritmica con cui rappresentare il valore. La scala dei deciBel può quindi essere applicata a qualunque grandezza.

---

<sup>2</sup> Vedi sezione 2.4.1

<i>Evento sonoro</i>	<i>Variazione di pressione</i> [Pa]	<i>Livello di pressione</i> (SPL) [dB]
Jet militare al decollo, 50m	200	150
Arma da fuoco, 3m	63	130
Concerto rock, 2m	20	120
Motosega, 1m	14	117
Passaggio di un Aeroplano	6,3	110
Clacson, 7m	2	100
Metropolitana, interno	1	94
Strada trafficata, 1m	$2 \cdot 10^{-1}$	80
Automobile, interno	$6 \cdot 10^{-2}$	70
Conversazione	$2 \cdot 10^{-2}$	60
Ambiente casalingo	$2 \cdot 10^{-3}$	40
Stormire delle foglie	$2 \cdot 10^{-4}$	20
<i>Soglia di udibilità</i>	$2 \cdot 10^{-5}$	0

**Tabella 2.2:** Valori di pressione e di SPL relativi a alcuni suoni comuni.

### 2.3.2 Livello di pressione

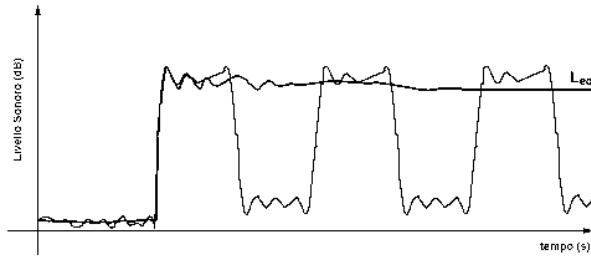
Quasi mai in acustica si fa riferimento alla pressione direttamente in pascal, si usa piuttosto misurare in dB e, per dare un significato energetico alla misura, si usa considerare il quadrato del valore di variazione di pressione. Si definisce quindi il *livello di pressione acustica L* (o SPL: Sound Pressure Level):

$$L = \log\left(\frac{p^2}{p_0^2}\right) \quad [\text{dB}] \quad (2.7)$$

o più comunemente

$$L = 20 \cdot \log\left(\frac{p}{p_0}\right) \quad [\text{dB}] \quad (2.8)$$

dove  $p_0$  sia il valore di scostamento di pressione usato come riferimento riferimento, scelto dal sistema internazionale uguale a  $20 \mu\text{Pa}$  che come abbiamo visto



**Figura 2.2:** Livello sonoro di una sorgente intermittente e relativo livello equivalente  $L_{EQ}$ .

poco sopra nell’equazione 2.4 corrisponde alla soglia di udibilità a  $1000\text{ Hz}$  nell’atmosfera in condizioni standard.<sup>3</sup>

### 2.3.3 Livello equivalente

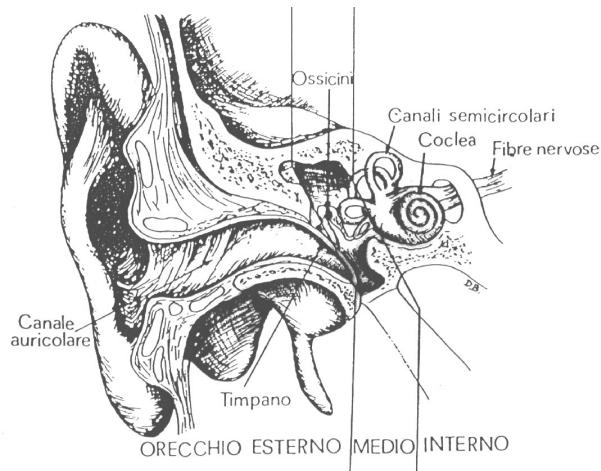
In applicazioni reali, usualmente ci si trova in presenza di sorgenti caratterizzate da un valore  $SPL$  non costante nel tempo, delle quali occorre valutare la *rumorosità*. Descrivendo il fenomeno sonoro con la funzione matematica che ne regola l’andamento del *livello di pressione* nel tempo, otteniamo una valutazione precisa in un dato istante. Questo però non ci fornisce un’informazione sulla rumorosità globale. Se ad esempio avessimo una sorgente che si accende ad intermittenza, conoscere esattamente l’andamento del tempo non aiuterebbe nel valutare il livello sonoro che la sorgente produce in un determinato periodo di tempo.

Si definisce quindi un *livello equivalente* ( $L_{EQ}$ ) che si calcola come:

$$L_{EQ} = 10 \log \left( \frac{1}{T} \int_0^T \frac{p^2(t)}{p_0^2} dt \right) \quad [\text{dB}] \quad (2.9)$$

Il livello equivalente rappresenta una sorta di media del livello sonoro sul periodo di tempo  $T$  considerato. In figura 2.2 è rappresentato l’andamento (quantitativo) del livello emesso da una sorgente intermittente ed il corrispondente livello equivalente. Come si vede dal grafico, il livello equivalente raggiunge valori sempre più stabili all’aumentare della lunghezza della finestra di integrazione considerata.

<sup>3</sup> Per una spiegazione più dettagliata sui livelli sonori si veda l’appendice A.3



**Figura 2.3:** Apparato acustico umano.

L'importanza di questo valore misurabile è quella di consentirci di quantificare il livello sonoro complessivo emesso da una sorgente in un determinato intervallo di tempo, attraverso un unico numero. Il livello equivalente, infatti, è usato nella legislazione come riferimento principale rispetto al quale emettere normative riguardanti l'inquinamento acustico. In particolare la Legge Italiana stabilisce tre intervalli di tempo diversi per effettuare le rilevazioni:

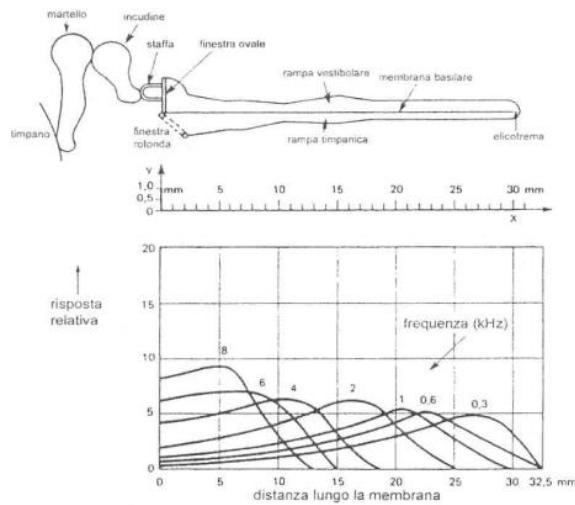
**dalle 6:00 alle 22:00** corrispondente al periodo diurno.

**dalle 22:00 alle 6:00** corrispondente al periodo notturno.

**8 ore** che corrispondono a una giornata lavorativa. Da utilizzare per regolare la rumorosità sul luogo di lavoro.

## 2.4 Sistema uditivo umano

L'apparato uditivo umano, come si evince dalla figura 2.3, è molto complesso e composto da moltissimi elementi, ognuno dei quali influisce sulla percezione acustica in maniera rilevante e addirittura alcuni degli effetti legati alla sensazione uditiva, che vengono chiamati *psicoacustici*, riguardano la sola interpretazione, da



**Figura 2.4:** Risposta non lineare della coclea

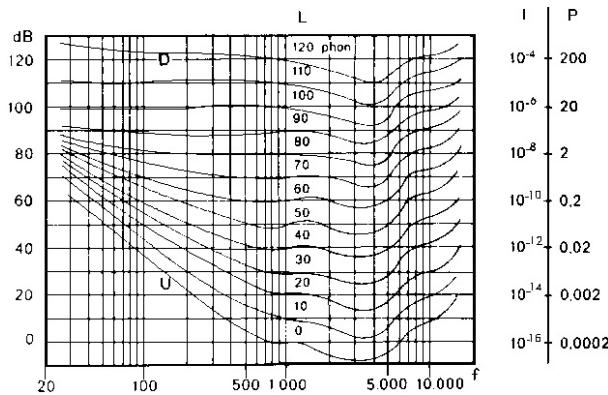
parte del cervello, dei segnali elettrochimici provenienti dall'apparato uditivo.

In questa sede ci interessano solamente alcuni effetti di non linearità dell'apparato uditivo, che comportano conseguenze fondamentali nelle modalità di analisi di un qualsiasi fenomeno sonoro, quali la descrizione mediante suddivisione in *bande frequenziali* e l'introduzione dei *filtri di ponderazione*, che verranno descritti successivamente.

### 2.4.1 Effetti di non linearità dell'orecchio umano

Uno degli organi sensoriali principali dell'apparato acustico è la *coclea* rappresentata in figura 2.4, responsabile di gran parte degli effetti non lineari della percezione uditiva umana, di cui discuteremo in seguito.

Analizzandola in sezione, si trova una sorta di doppia lamina la quale è caratterizzata da una diversa sensibilità lungo la sua estensione, a seconda delle frequenze di eccitazione del segnale acustico, similmente ad una corda o ad una frusta. Si osservi nel grafico di figura 2.4 come le basse frequenze interessino la parte ter-



**Figura 2.5:** Grafico delle curve isofoniche

minale mentre le alte frequenze la parte iniziale. Si evince facilmente inoltre che due rumori con bande sovrapposte (in tutto o in parte) si mascherino in modo tale che il segnale di maggiore intensità copra il segnale più debole, a meno che quest'ultimo non abbia una banda sufficientemente larga. Per questo ed altri motivi che non approfondiremo in questa sede, il sistema uditivo umano presenta una sensibilità meno accentuata alle frequenze molto basse (poche decine di  $\text{Hz}$ ) ed a quelle elevate (oltre i  $15 \text{ kHz}$ ). Inoltre per procurare la stessa *sensazione sonora* (che si misura in *phon*) occorrono, a frequenze diverse, livelli di pressioni sonore diverse, allo stesso modo suoni di stessa intensità ma frequenza diversa vengono percepiti dall'orecchio in modo diverso.<sup>4</sup>

Tutti questi effetti sono osservabili nel grafico in figura 2.5.

Essendo dunque tutto il sistema uditivo umano, nella sua risposta all'eccitamento acustico, molto soggetto alla frequenza dell'onda incidente, è spesso comodo analizzare i fenomeni acustici *filtrati* solo per alcune *bande* di frequenza.

## 2.4.2 Bande frequenziali

La descrizione della composizione in frequenza dei segnali sonori può essere condotta valutando il contenuto di energia sonora all'interno di prefissati intervalli

<sup>4</sup> Si veda [9] al capitolo 2.

di frequenze, le *bande di frequenza*, ognuna di esse definita all'interno dell'intervallo delimitato da una frequenza di taglio superiore  $f_s$  e da una frequenza di taglio inferiore  $f_i$ , e caratterizzato dalla propria frequenza di banda centrale  $f_c$ . A seconda del numero e della posizione delle bande che si decide di utilizzare al fine di dividere lo spettro sonoro, si possono condurre principalmente due tipi di analisi:

**analisi a banda costante:** in questo tipo di suddivisione si utilizzano bande frequenziali di uguale larghezza, nello specifico che godono della proprietà:

$$\Delta f = f_s - f_i = \text{costante} \quad (2.10)$$

**analisi a banda percentuale costante:** la larghezza delle bande utilizzate  $\Delta f$  è una percentuale costante del valore della frequenza nominale di centro-banda ( $f_c$ ) che caratterizza la banda stessa. La scelta di questo metodo di suddivisione dello spettro delle frequenze avviene in special modo perché il sistema uditivo umano risponde agli stimoli in modo più simile allo schema dell'apertura percentuale costante che a quello dell'apertura costante.

Si parla di *ottave* (o bande da 1 /1 di ottava) nel caso che le frequenze di taglio siano conformi alla legge

$$\begin{cases} f_s = 2 \cdot f_i \\ f_i = \frac{f_c}{\sqrt{2}} \\ f_s = f_c \cdot \sqrt{2} \end{cases} \quad (2.11)$$

Si parla invece di *terzi d'ottava* (bande da 1 /3 d'ottava) se invece le frequenze di taglio seguono questa legge

$$f_s = \sqrt[3]{2} \cdot f_i \quad (2.12)$$

Essendo l'intervallo frequenziale di interesse compreso tra 20 e 20 000  $Hz^5$  sono necessari “banchi” di 10 filtri (ottave) o 30 filtri (terzi d'ottava) per coprire tutto lo spettro udibile.

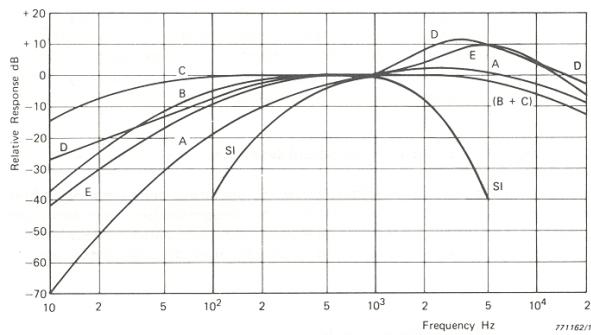
Entrambe le analisi vengono ottenute tramite un'operazione detta di *filtraggio*, che non sarà approfondita in questa sede ma di cui si può trovare la descrizione dell'implementazione al capitolo 7.6.

Nella seguente tabella sono definite le diverse bande e relative frequenze di centro-banda per le due diverse analisi descritte.

---

<sup>5</sup> Intervallo di sensibilità dell'orecchio umano, come visto nella sezione 2.2.1

<i>Bande di 1 /1 ottava</i>			<i>Bande di 1 /3 ottava</i>		
<i>fi</i>	<i>fc</i>	<i>fs</i>	<i>fi</i>	<i>fc</i>	<i>fs</i>
			22,4	25	28,2
22	31,5	44	28,2	31,5	35,5
			35,5	40	44,7
			44,7	50	56,2
44	63	88	56,2	63	70,8
			70,8	80	89,1
			89,1	100	112
88	125	177	112	125	141
			141	160	178
			178	200	224
177	250	355	224	250	282
			282	315	355
			355	400	447
355	500	710	447	500	562
			562	630	708
			708	800	891
710	1 000	1 420	891	1 000	1 122
			1 122	1 250	1 413
			1 413	1 600	1 778
1 420	2 000	2 840	1 778	2 000	2v239
			2 239	2 500	2v818
			2 818	3 150	3v548
2 840	4 000	5 680	3 548	4v000	4 467
			4 467	5 000	5v623
			5 623	6v300	7 079
5 680	8 000	11 360	7v079	8 000	8 913
			8 913	10 000	11 220
			11 220	12 500	14 130
11 360	16 000	22 720	14 130	16 000	17 780
			17 780	20 000	22 390



**Figura 2.6:** Grafico delle curve di ponderazione

### 2.4.3 Filtri di ponderazione

Come descritto nel paragrafo 2.4.1, la sensibilità dell'orecchio varia al variare della frequenza. Per tale motivo il livello di pressione *SPL* che misuriamo in realtà non corrisponde a una reale sensazione acustica, cioè variazioni del valore di *SPL* non necessariamente corrispondono a uguali variazioni nella percezione acustica (variazioni di *volume*). Per rendere più aderente alla sensazione umana e quindi rendere più intuitiva la misura di un fenomeno sonoro, occorre utilizzare dei filtri di pesatura o *ponderazione*. Quelli attualmente utilizzati sono rappresentati in figura 2.6. Analizzando il grafico notiamo le curve più importanti<sup>6</sup>:

**filtro di ponderazione “A”** : il più comunemente impiegato e il cui andamento si conforma alla risposta dell'orecchio umano a livelli medio-bassi. Il livello misurato con la ponderazione del filtro *A* viene chiamato  $dB(A)$ .

**filtro di ponderazione “C”** : impiegato per rumori molto forti o esplosioni misurate quindi in  $dB(C)$ .

---

<sup>6</sup> Per un approfondimento sulle curve di ponderazione si veda l'appendice A.4

# Capitolo 3

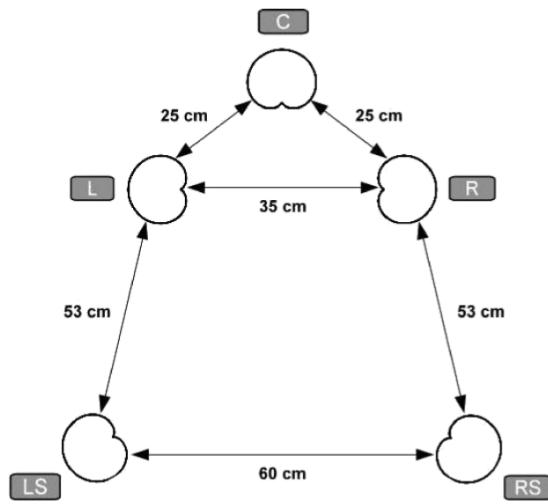
## Array microfonici

Nell'evoluzione delle tecniche di riproduzione di un ambiente acustico, si è giunti a livelli di spazializzazione notevoli che rendono l'ascolto molto più coinvolgente. Basti pensare alla diffusione della stereofonia negli anni '60 fino a quello che è ormai uno standard *de facto* della riproduzione audio domestica o cinematografica: il *Surround 5.1*. Esistono anche molte altre tecniche avanzate di riproduzione sonora in grado di avvicinarsi a una simulazione realistica del campo acustico, almeno nel punto di ascolto, quali lo *stereo dipolo binaurale* o la *Wave Field Synthesis* ma non è questa la sede per approfondire tale tema.

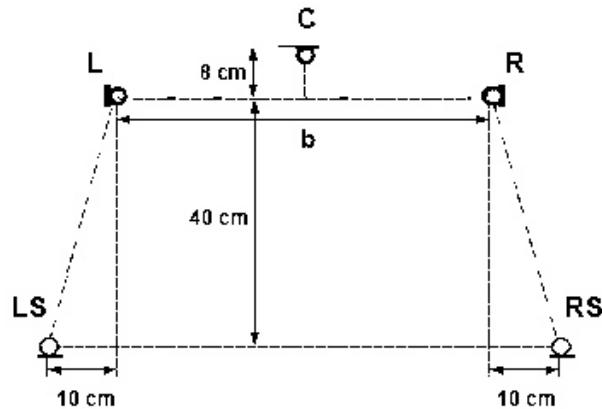
Ciò che si vuole sottolineare invece è che la ricerca in campo di registrazione ha portato allo studio di tecniche sempre più innovative per l'acquisizione del campo acustico nei più fini dettagli, quali definizione spaziale, direttività e composizione del segnale.

Inizialmente per creare la sensazione di definizione spaziale si utilizzava una tecnica semplice di *panning* che consiste nel processare il segnale registrato in modo da *proiettare* il suono su uno o sull'altro altoparlante. Parallelamente si sono fatti esperimenti di registrazione multicanale in modo da mantenere il rapporto 1:1 tra canali registrati e numero di altoparlanti, al fine di evitare il *post-processing* e quindi di modificare il segnale originale.

Si vogliono citare al riguardo gli schemi *INA-5* e *OCT-surround*, riportati



**Figura 3.1:** Tecniche di registrazione con array discreti di tipo INA-5



**Figura 3.2:** Tecniche di registrazione con array discreti di tipo OCT surround

rispettivamente in figura 3.1 e figura 3.2. Il successo di tali array risultò essere molto limitato a causa del fatto che queste tecniche non consentono un controllo in fase di registrazione; inoltre gli array forniscono un angolo di ripresa fissato non regolabile a meno di complesse strutture snodate e non danno buoni risultati in termini di missaggio dei segnali captati a breve distanza.

Un’alternativa innovativa rispetto a queste topologie venne proposta da Michael Gerzon and Peter Craven, che nel 1978 lanciarono sul mercato il microfono



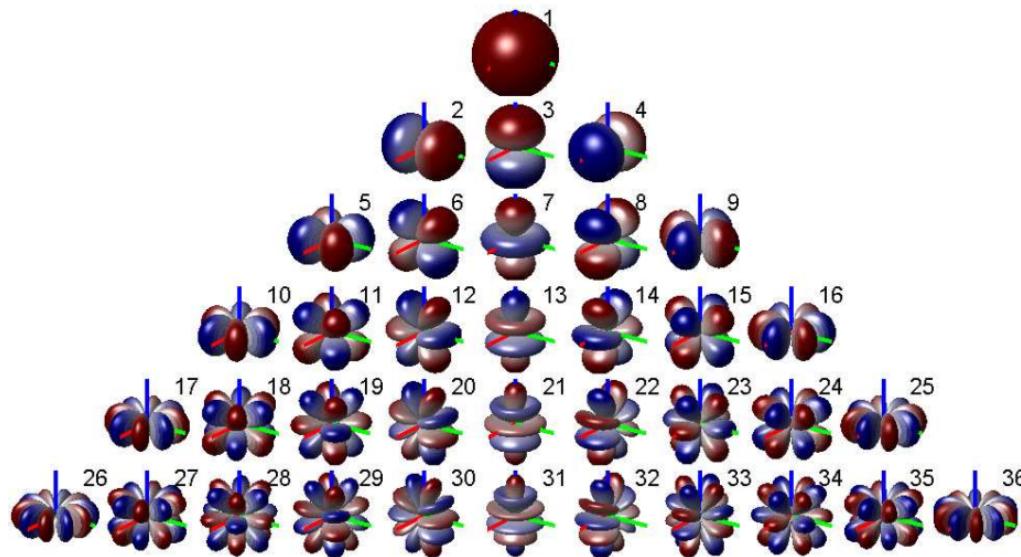
**Figura 3.3:** Il microfono *SoundField™* di Gerzon e Craven

*SoundField™*. Si tratta di un array di 4 capsule cardioidi disposte sulle facce di un tetraedro regolare come mostrato nella foto in figura 3.3.

L’approccio innovativo di tale microfono fu quello di scindere completamente i concetti di registrazione e riproduzione acquisendo dei segnali che modellassero il campo acustico nelle sue caratteristiche di pressione ma anche di velocità; solo la combinazione di questi segnali avrebbe dato luogo a informazioni acustiche significative. Questo approccio costituisce la base della codifica *Ambisonics* inventata dallo stesso Peter Craven che rappresenta tutt’oggi un importante riferimento per la ricerca su impianti audio moderni. Secondo questa teoria, i quattro segnali registrati dalle capsule del sensore tetraedrico, vengono denominati *A-format* e da essi, con un processo matematico teorizzato nella teoria *Ambisonics*, è possibile ottenere quattro segnali che caratterizzano in prima approssimazione il campo vettoriale acustico nel punto in cui è posta la sonda:

- il segnale omnidirezionale di pressione ( $W$ ),
- i tre segnali direzionali di velocità lungo i tre assi cartesiani ( $X$ ,  $Y$  e  $Z$ ).

Questi segnali così elaborati vengono chiamati *B-format*. Essi costituiscono le armoniche di ordine 0 e 1 della codifica *Ambisonics* e sono indipendenti dalla conformazione dell’array utilizzato; la caratterizzazione *B-format* è diventata quindi uno standard per rappresentare l’informazione acquisita dagli array microfonici avanzati.



**Figura 3.4:** Diagramma polare delle armoniche *Ambisonics*

Oggi esistono nuove topologie di array, dal numero di microfoni molto più elevato e dalle disposizioni più disparate, in grado di sintetizzare armoniche *Ambisonics* di ordine superiore al primo il cui diagramma polare è riportato in figura 3.4.

Combinando tra loro un set di segnali codificati in formato *Ambisonics* (*B-format*), attraverso un opportuno decoder che opera un semplice missaggio, è possibile ricreare un segnale equivalente a quello registrato da un microfono virtuale con orientazione arbitraria e direttività di ordine massimo pari all'ordine di troncamento del set di armoniche utilizzato.<sup>1</sup>

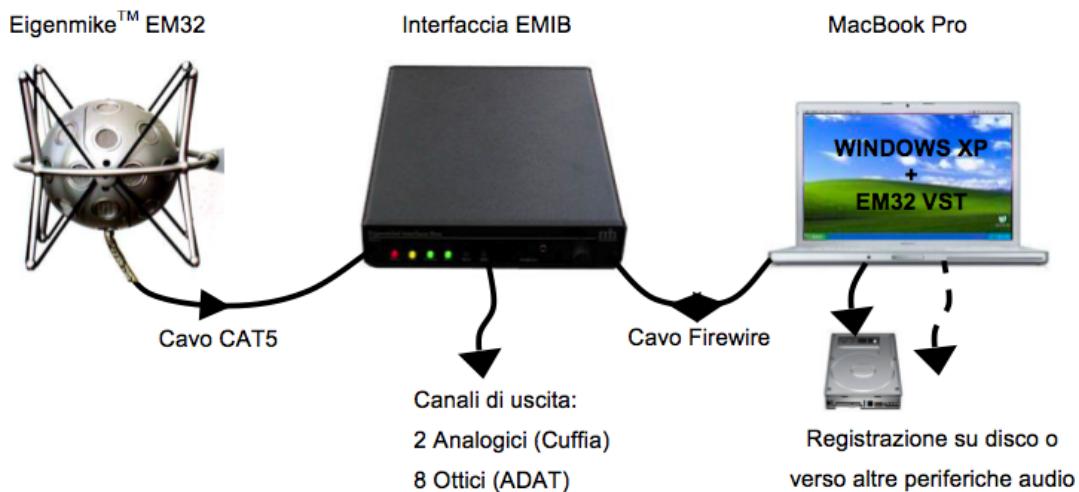
Nel presente capitolo verranno presentati i tre diversi tipi di array microfonici presi in considerazione e sui quali questa tesi è stata collaudata.

**Sferico:** *Eigenmike™ EM32* prodotto dalla *mhAcoustics*<sup>2</sup>

**Cilindrico:** array progettato dal gruppo di Acustica dell'Università degli Studi di Parma sull'hardware dell'array sferico *Eigenmike™ EM32* ma con alcune

<sup>1</sup> Vedi il capitolo 4 sulla sintesi dei microfoni virtuali.

<sup>2</sup> [http://www.mhacoustics.com/mh\\_acoustics/Eigenmike\\_microphone\\_array.html](http://www.mhacoustics.com/mh_acoustics/Eigenmike_microphone_array.html).



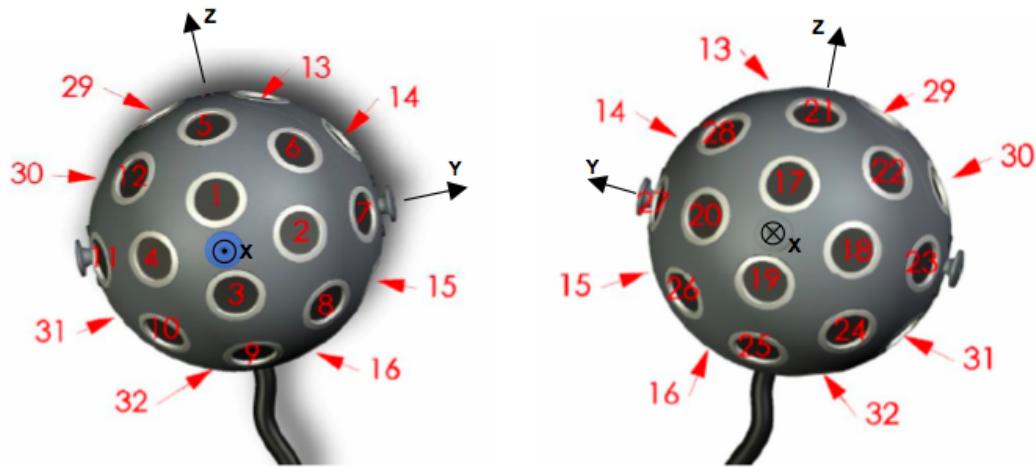
**Figura 3.5:** Struttura del sistema *Eigenmike™ EM32*

aggiunte di progetto.

**Planare:** progettato dal gruppo di acustica nel 2009 e migliorato negli anni con alcune modifiche sostanziali.

### 3.1 Array sferico: *Eigenmike™ EM32*

Si tratta del primo array microfonico ad elevato grado di integrazione utilizzato dall'Università di Parma, grazie ad un progetto di collaborazione con il Centro Ricerche RAI di Torino; viene prodotto dalla società americana *mhAcoustic*<sup>3</sup> ed è probabilmente il primo sistema così portatile della sua categoria ad essere stato commercializzato.



**Figura 3.6:** Disposizione delle capsule dell'array microfonico *Eigenmike<sup>TM</sup> EM32*

### 3.1.1 Composizione del sistema

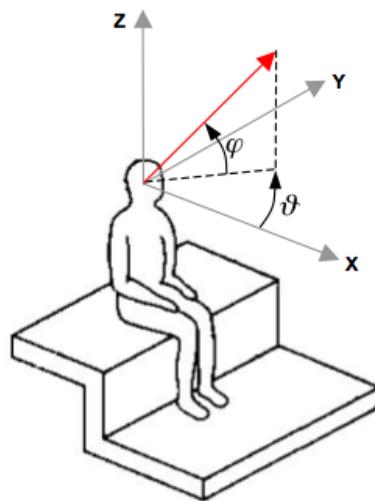
Come possiamo notare dalla figura 3.5, il sistema *Eigenmike<sup>TM</sup> EM32* viene fornito dalla casa produttrice *mhAcoustic* composto da un array microfonico sferico a 32 capsule e da una interfaccia di controllo *Eigenmike Interface Box (EMIB)*, i quali devono venire interfacciati attraverso il software corredato, con un computer che si occuperà di salvare su hard disk le sessioni di registrazione dell'array.

L'array microfonico è costituito da una sfera di alluminio anodizzato con raggio di 42 mm sulla cui superficie sono disposte in modo relativamente uniforme 32 capsule di tipo *electret* di elevata qualità. La conoscenza della disposizione precisa delle capsule è di grande importanza al fine di mappare queste ultime su una proiezione dello sfondo, come verrà spiegato al capitolo 6.3.

Possiamo intenderne il preciso assetto dall'osservazione della figura 3.6. All'interno della sfera sono alloggiati l'intera circuiteria analogica di preamplificazione con guadagno controllabile via software, nonché i convertitori A/D in grado di campionare i segnali delle capsule con risoluzione fino a 24 bit e frequenza di campionamento di 48 000 Hz. I segnali vengono quindi trasferiti all'interfaccia *EMIB* mediante un collegamento digitale garantendo l'immunità ai disturbi e

---

<sup>3</sup> [www.mhacoustics.com](http://www.mhacoustics.com)



**Figura 3.7:** Angoli di riferimento e loro direzioni relative per la descrizione sferica rispetto al sistema di riferimento antropometrico secondo la norma ISO2631.

con grande guadagno in comodità, ingombro e risparmio di cablaggi audio XLR. Lo stesso sistema di preamplificazione analogica consente anche di compensare le differenze di sensibilità esistenti tra una capsula e l'altra per ottenere scostamenti compresi tra  $\pm 0.5$  dB. La mhAcoustics caratterizza infatti la sensibilità delle capsule una a una, e fornisce all'acquirente un file di testo contenente i 32 valori che corrispondono ai guadagni in dB necessari alla compensazione. Questi valori, una volta caricati sulla EEPROM contenuta all'interno del microfono stesso, vengono automaticamente sommati ad ogni variazione del guadagno in modo da effettuare la compensazione di sensibilità in modo totalmente trasparente.

Al fine di effettuare la mappatura delle capsule ricercata, sono necessarie le coordinate sferiche descritte dagli angoli di *azimut* e *elevation*. Per questa descrizione si è fatto riferimento alla norma ISO2631 che viene descritta nella figura 3.7.

<i>mic #</i>	$\theta$ [rad]	$\varphi$ [rad]	$\theta$ [ $^{\circ}$ ]	$\varphi$ [ $^{\circ}$ ]
1	0,000	0,367	0	21
2	0,559	0,000	32	0
3	0,000	-0,367	0	-21
4	5,725	0,000	328	0
5	0,000	1.012	0	58
6	0,785	0,611	45	35
7	1,204	0,000	69	0
8	0,785	-0,611	45	-35
9	0,000	-1,012	0	-58
10	5,498	-0,611	315	-35
11	5,079	0,000	291	0
12	5,498	0,611	315	35
13	1,588	1,204	91	69
14	1,571	0,559	90	32
15	1,571	-0,541	90	-31
16	1,553	-1,204	89	-69
17	3,142	0,367	180	21
18	3,700	0,000	212	0
19	3,142	-0,367	180	-21
20	2,583	0,000	148	0
21	3,142	1,012	180	58
22	3,927	0,611	225	35
23	4,346	0,000	249	0
24	3,927	-0,611	225	-35
25	3,142	-1,012	180	-58
26	2,356	-0,611	135	-35
27	1,937	0,000	111	0
28	2,356	0,611	135	35
29	4,695	1,204	269	69
30	4,712	0,559	270	32
31	4,712	-0,559	270	-32
32	4,730	-1,204	271	-69

**Tabella 3.1:** Coordinate sferiche antropometriche delle direzioni delle capsule fornite sia in gradi che in radianti secondo la norma ISO2631 descritta in figura 3.7

### 3.1.2 Limiti intrinseci teorici del sistema

Per la morfologia dell'oggetto *array di microfoni* in questione è possibile calcolare alcuni limiti operativi intrinseci e confrontarli con quelli dichiarati dalla casa produttrice.

In primo luogo, secondo il *teorema di Shannon*, dovrebbe verificarsi un fenomeno di *aliasing spaziale* se si tentassero di decodificare segnali di una frequenza maggiore di quella limite. Per poter essere decodificata correttamente, infatti, un'onda sonora di lunghezza d'onda  $\lambda$ , dovrebbe essere campionata *nello spazio* un numero doppio di volte rispetto alla frequenza dell'onda che si vuole campionare. Si può quindi scrivere una relazione che lega la massima frequenza decodificabile prima dell'introduzione di effetti di *aliasing*, che chiameremo  $f_{alias}$ , con la distanza di campionamento, cioè la distanza minima tra le capsule  $d$ , che è funzione dell'angolo tra le capsule  $\gamma$  e il raggio  $R$  della sfera, e quindi anche del numero di capsule del microfono:

$$f_{alias} = \frac{v}{2 \cdot \lambda} = \frac{v}{2 \cdot d} \approx \frac{v}{2 \cdot R \cdot \gamma} , \quad (3.1)$$

essendo  $v$  la velocità del suono.

Osservando la disposizione delle capsule descritta in tabella 3.1 si può notare che ci sono due possibili spaziature angolari fra le capsule, abbastanza simili tra loro:

$$\begin{cases} \gamma_{1-5} = 1,012 - 0,367 = 0,645 \\ \gamma_{1-3} = 0,367 - (-0,367) = 0,734 \end{cases} \quad (3.2)$$

dove  $\gamma_{1-5}$  rappresenta la spaziatura tra le capsule 1 e 5 mentre  $\gamma_{1-3}$  tra quelle contrassegnate dai numeri 1 e 3.

Sostituendo queste spaziature nella equazione 3.1,

$$\begin{cases} f_{alias\ 1-5} \approx \frac{v}{2 \cdot R \cdot \gamma_{1-5}} = \frac{343}{2 \cdot 42 \cdot 10^{-3} \cdot 0,645} = 6,3 \text{ [kHz]} \\ f_{alias\ 1-3} \approx \frac{v}{2 \cdot R \cdot \gamma_{1-3}} = \frac{343}{2 \cdot 42 \cdot 10^{-3} \cdot 0,734} = 5,6 \text{ [kHz]} \end{cases} \quad (3.3)$$

Quindi la massima frequenza operativa sembra essere intorno ai 5,5 kHz. Tuttavia è bene precisare che tale relazione è valida sotto precise semplificazioni geometriche e quindi solamente per array lineari uniformi. La banda passante dichiarata dal costruttore, invece, è dell'ordine dei 10 kHz, quindi ampiamente superiore al valore appena calcolato.

Un'altra importante relazione emerge dalla scomposizione del campo sonoro in serie di *Fourier-Bessel* e consente di stimare il massimo ordine ( $L$ ) delle armoniche *Ambisonic* ottenibili da un microfono sferico con  $N$  capsule.

$$(L + 1)^2 \leq N \quad (3.4)$$

dalla quale deriva, essendo il numero di ordine delle armoniche strettamente non negativo,

$$L \leq \sqrt{N} - 1 = \sqrt{32} - 1 = 4,7 \quad (3.5)$$

Risulta quindi che il sensore è in grado di generare microfoni virtuali supercardioidi con direttività fino al 4° ordine, mentre la casa costruttrice ha impostato il sistema per la generazione di microfoni virtuali fino alla massima direttività del 3° ordine.



**Figura 3.8:** Array lineare e relativo diagramma polare a 1 kHz

## 3.2 Altri array

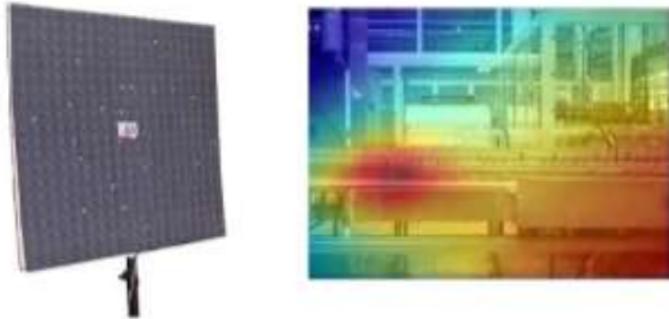
Esistono altre topologie di array avanzati di cui in questo capitolo si fornirà una breve descrizione e si elencheranno le principali differenze con l'array sferico appena descritto.

### 3.2.1 Array lineari

Gli array lineari sono i più semplici e i più diffusi dal punto di vista commerciale. I loro utilizzi sono plurimi e diversificati, dall'individuazione di sorgenti sonore alle misure acustiche attraverso la sintesi di un microfono virtuale ultradirettivo lungo l'asse dell'array. La disposizione lineare però produce inevitabilmente una risposta spaziale dell'array non isotropa che non consente quindi di ottenere una ricostruzione precisa del campo sonoro tridimensionale.

### 3.2.2 Array planari

Gli array planari hanno un campo di ripresa sonora di circa  $180^\circ$  sia sul piano orizzontale che verticale e sono indicati soprattutto per ottenere immagini sonore tipo *acoustic camera*, utili nell'analisi rumoristica generalmente in ambito industriale. Questa topologia consente di ottenere focalizzazioni della direttività su una griglia rettangolare di punti disposti frontalmente rispetto al pannello. Il

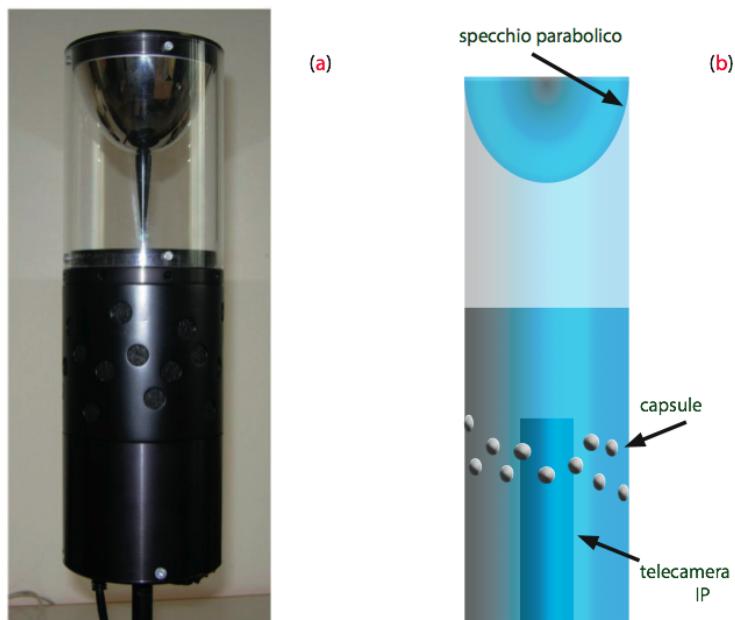


**Figura 3.9:** Array planare e relativa mappa acustica che rappresenta i livelli sonori crescenti dal blu al rosso e in cui si riconosce una sorgente sonora come addensamento di rosso

modello planare costruito dal gruppo di Acustica del Dipartimento di Ingegneria Industriale dell’Università degli studi di Parma, presenta una disposizione delle capsule secondo una spirale logaritmica modificata: questa figura geometrica dovrebbe portare a una risoluzione acustica migliore rispetto alla disposizione delle capsule casuale.

### 3.2.3 Array cilindrici

Questo tipo di array si pone l’obiettivo di scansionare principalmente l’asse orizzontale a scapito della risoluzione sull’asse verticale, un po’ a immagine e somiglianza della naturale percezione dell’apparato acustico umano e di conseguenza anche più utile nella maggioranza delle applicazioni. L’array cilindrico fornisce quindi la possibilità di sintetizzare microfoni virtuali a direttività arbitraria con una zona di puntamento e focalizzazione di  $360^\circ$  sul piano orizzontale e di circa  $90^\circ$  sul piano verticale ( $\pm 45^\circ$  dall’orizzonte). L’array costruito dal gruppo di ricerca sopracitato è inoltre dotato di una telecamera IP che, puntando ad uno specchio per visione omnidirezionale, consente di avere un’immagine di servizio da utilizzarsi come background per la mappa acustica o per il sistema di puntamento



**Figura 3.10:** Array cilindrico (a), con relativa sezione verticale (b)

real-time dei microfoni virtuali.<sup>4</sup>

<sup>4</sup> Si tratta di un progetto parallelo di cui è consultabile una pubblicazione [3] della società di acustica AIDA srl di Parma in collaborazione con la RAI

# Capitolo 4

## Sintesi di microfoni virtuali

L'utilizzo di array di microfoni avanzati, come abbiamo visto, genera numerose possibilità di analisi, tuttavia i segnali registrati dalle capsule devono essere interpretati e non hanno una valenza di per sé. Come abbiamo visto, la codifica interpretativa tradizionale consiste nel generare le armoniche sferiche *Ambisonics* fino al massimo ordine possibile a seconda del tipo di array.

A partire da questi segnali, con opportune combinazioni e elaborazioni, si può fornire in uscita un set di registrazioni che simulano microfoni, chiamati *virtuali* appunto, dalla direzione e apertura polare arbitrarie.

La parte dell'array microfonico che si occupa di passare da segnali *raw* (o *B-format*) ai microfoni virtuali è il *sistema di elaborazione*

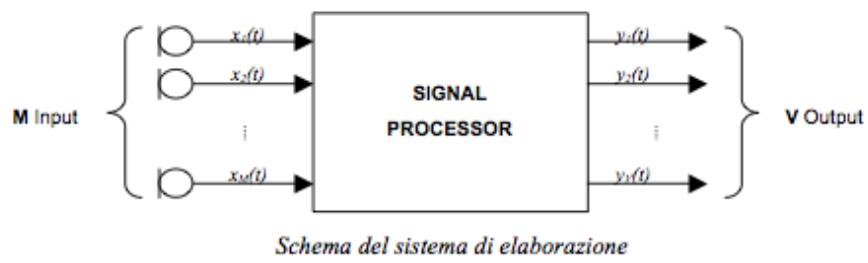
### 4.1 Sistema di Elaborazione

Il sistema di elaborazione dei segnali *raw* delle capsule di un array microfonico è dunque di primaria importanza per dare significato alle registrazioni ottenute. Implementato solitamente da processori digitali dedicati o in alternativa da sistemi virtualizzati software generano dei segnali digitali esprimibili come sommatoria dei contributi di ogni capsula filtrati opportunamente. L'*i-esimo* segnale

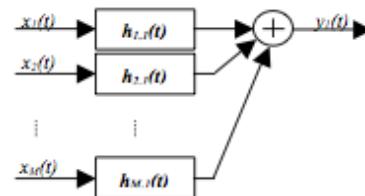
di uscita risponderà quindi alla legge generica

$$y_i(t) = \sum_{m=1}^M x_m(t) * h_{m,i}(t) \quad (4.1)$$

Il processing compiuto da questo componente del sistema *array microfonico*



*Schema del sistema di elaborazione*

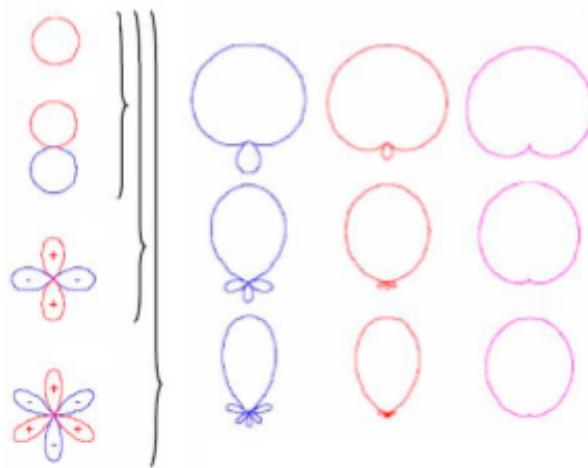


*Schema di un singolo canale di uscita del sistema di elaborazione*

**Figura 4.1:** Ogni microfono virtuale può essere sempre descritto da una somma di convoluzioni di tutti gli ingressi per un opportuno filtro FIR

non costituisce dunque un problema oneroso, infatti si tratta di una somma di convoluzioni facilmente implementabile su un calcolatore con algoritmi noti. La cosa interessante in questa fase consiste nel capire come generare i microfoni voluti cioè quali filtri utilizzare per ottenere questo risultato.

Come possiamo vedere dalla figura 4.2, combinando le armoniche sferiche *Ambisonics* di ordine dallo 0 al 3 della colonna di sinistra, è possibile generare microfoni virtuali supercardioidi con direttività crescente dall'ordine 1 all'ordine 3.



**Figura 4.2:** Possibili microfoni virtuali a ordine di direttività crescente (sulla destra) calcolati a partire da armoniche sferiche *Ambisonics* fino al terzo ordine (sulla sinistra).

La direttività di un microfono virtuale di ordine  $h$ , espressa in coordinate sferiche, segue l'espressione:

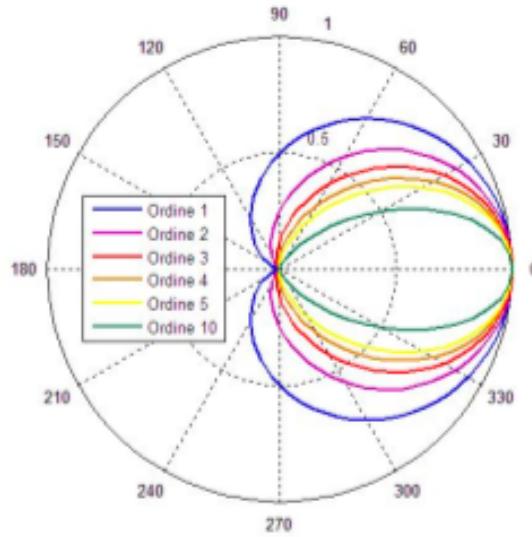
$$D_H(\vartheta, \varphi, h) = \text{sign}(D_1(\vartheta, \varphi)) \cdot |D_1(\vartheta, \varphi)|^h, \quad (4.2)$$

dove  $D_1(\vartheta, \varphi)$  è la direttività del microfono virtuale del primo ordine che è pari a:

$$D_1(\vartheta, \varphi) = P + G \cdot \cos \vartheta \cdot \cos \varphi \quad (4.3)$$

Questo approccio costituisce la tradizione e la storia di questo tipo di codifiche fin dagli anni '70, quando vennero introdotte, con la sola miglioria dell'incremento di ordine di armoniche raggiungibile. Essendo però nella maggior parte dei casi precisa volontà dell'utilizzatore di questi sistemi codificare un determinato set di microfoni virtuali noto a priori, può essere conveniente evitare di passare attraverso la codifica *Ambisonics* troppo teorica e soggetta a moltissimi scostamenti dovuti alle non idealità dei sistemi.

Si è provato a seguire una strada alternativa, definendo direttamente le uscite del sistema di elaborazione come segnali monofonici corrispondenti ad un set di microfoni con l'orientazione e direttività cercate.



**Figura 4.3:** Diagramma polare di un microfono virtuale cardioide in funzione dell'ordine di direttività.

#### 4.1.1 Approccio innovativo di inversione numerica

Dalle considerazioni fatte sullo stato dell'arte nell'elaborazione dei segnali degli array microfonici emerge che le tecniche utilizzate sono basate esclusivamente su modelli teorici fortemente semplificati della struttura fisica del microfono e delle caratteristiche delle capsule.

L'approccio innovativo proposto prevede di caratterizzare l'intero array microfonico sperimentalmente, determinando, mediante misure di risposta all'impulso, la funzione di trasferimento tra un certo numero di onde piane incidenti sull'array microfonico da diverse direzioni e i segnali elettrici generati dalle singole capsule.

Elaborando mediante opportune tecniche di inversione numerica la matrice di risposte all'impulso ottenuta, si vuole determinare una matrice di filtri che, convoluti con i segnali delle capsule microfoniche, permetta di ottenere i segnali di un set di microfoni virtuali con direttività arbitraria.

La scelta dei coefficienti che caratterizzano i filtri viene fatta mediante metodi numerici che utilizzano tecniche di inversione basate sulla minimizzazione dell'errore ai minimi quadrati.

I risultati ottenuti saranno quindi *massimamente simili* a quelli desiderati.<sup>1</sup>

---

<sup>1</sup> Si veda [4] e [7] per una descrizione più accurata del procedimento di generazione dei filtri di inversione qui riassunto. Inoltre si veda l'articolo [5] per il processo di regolarizzazione usato sui filtri.

# Capitolo 5

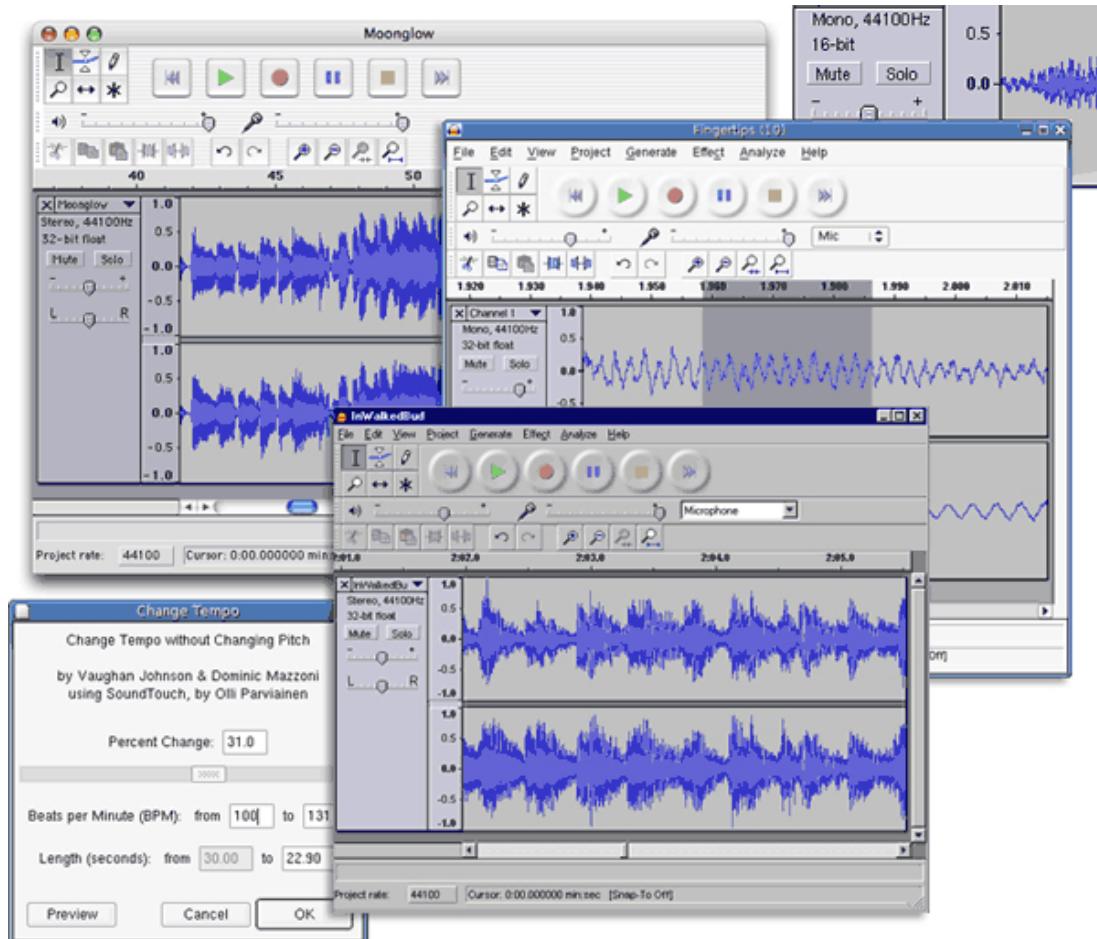
## Ambiente di sviluppo: l'host

### *Audacity<sup>®</sup>*

Nell'approccio al problema presentato, si sono studiate e discusse diverse possibilità di sviluppo. Una prima possibilità presa in considerazione era quella di sviluppare un software stand-alone, completo, in grado quindi di effettuare autonomamente tutte le operazioni necessarie alla creazione della mappa, a partire dalla registrazione del materiale audio proveniente dall'array microfonico, fino alla sintesi della mappa stessa.

Per massimizzare la granularità del progetto e per sfruttare risorse già disponibili gratuitamente si è preferito sviluppare un *plug-in* che si installasse su un sistema pre-esistente, in grado di svolgere tutte le funzioni di base di recording e editing audio, nonché di poter essere in un futuro espandibile con altre funzioni attraverso l'uso di altri *plug-in* creando quindi una *suite* multifunzionale.

A questo punto si sono presi in considerazione diversi tipi di plug-in e SDK tra le più diffuse e utilizzate, che avessero le caratteristiche cercate di compatibilità multipiattaforma. Inizialmente si era valutata la possibilità di sviluppare un *plug-in* utilizzando l'interfaccia VST (Virtual Studio Technology) della Steinberg<sup>®</sup>, per



**Figura 5.1:** Interfaccia utente dell'host utilizzato, *Audacity*<sup>®</sup> sotto diverse piattaforme.

via della universale diffusione e implementazione su quasi tutti i tipi di host su ogni piattaforma esistente.

Nonostante gli indiscutibili pregi di portabilità, facilità di utilizzo e di reperibilità di documentazione e supporto, non è stato ritenuto vantaggioso sviluppare il software con questa SDK.

Il problema fondamentale che si è riscontrato è stato l'impossibilità di effettuare analisi globali nel dominio del tempo, come una riscalatura o una semplice normalizzazione per via della rigida impostazione di questi tipi di *plug-in*, i quali sono pensati per lavorare a blocchetti di 256 campioni alla volta in modo da si-

mulare il più possibile un effetto *real-time*. Inoltre sarebbe venuto a mancare la possibilità di portare il software sui sistemi GNU/Linux, sui quali la tecnologia VST è scarsamente supportata.

Si è optato quindi per una piattaforma più flessibile, comoda e personalizzabile (in quanto progetto *open source*), *Audacity*<sup>®</sup>. Questo *host* fornisce ai *plug-in* che ne espandono le funzionalità (i *moduli*) tutte le informazioni di cui dispone, dall'intero *workspace* contenente il materiale audio, fino ad arrivare alle preferenze che l'utente sta utilizzando. Inoltre si tratta di un programma nativo multipiattaforma, come richiesto, e a migliorarne ulteriormente le capacità si noti che si tratta di uno dei pochissimi programmi in grado di gestire file audio multicanale e che implementano il formato audio *W64*. L'utilizzo di questo formato è fondamentale in quanto può essere necessario dover registrare eventi di lunghezza anche superiore a un'ora, utilizzando decine di canali. Infatti, con l'utilizzo di formati audio dotati di **header** inferiori ai 64 bit, non si supera la decina di minuti di acquisizione, se il numero di canali va oltre i 2 della normale codifica stereo o al massimo i 6 o 7 per le codifiche di tipo *surround*, per un problema di spazio di indirizzamento. Nel formato wav per esempio il campo dell'header riferito ai dati potrà indirizzare solo 4 GB ( $2^{32} - 1$  bit), mentre il W64, come suggerisce il nome, avrà una lunghezza massima di  $2^{64} - 1$  bit. *Audacity*<sup>®</sup> consente inoltre la registrazione a 16, 24 o 32 bit float con una frequenza di campionamento fino a 96 KHz.

## 5.1 Libreria per la costruzione di interfacce grafiche **wxWidgets**

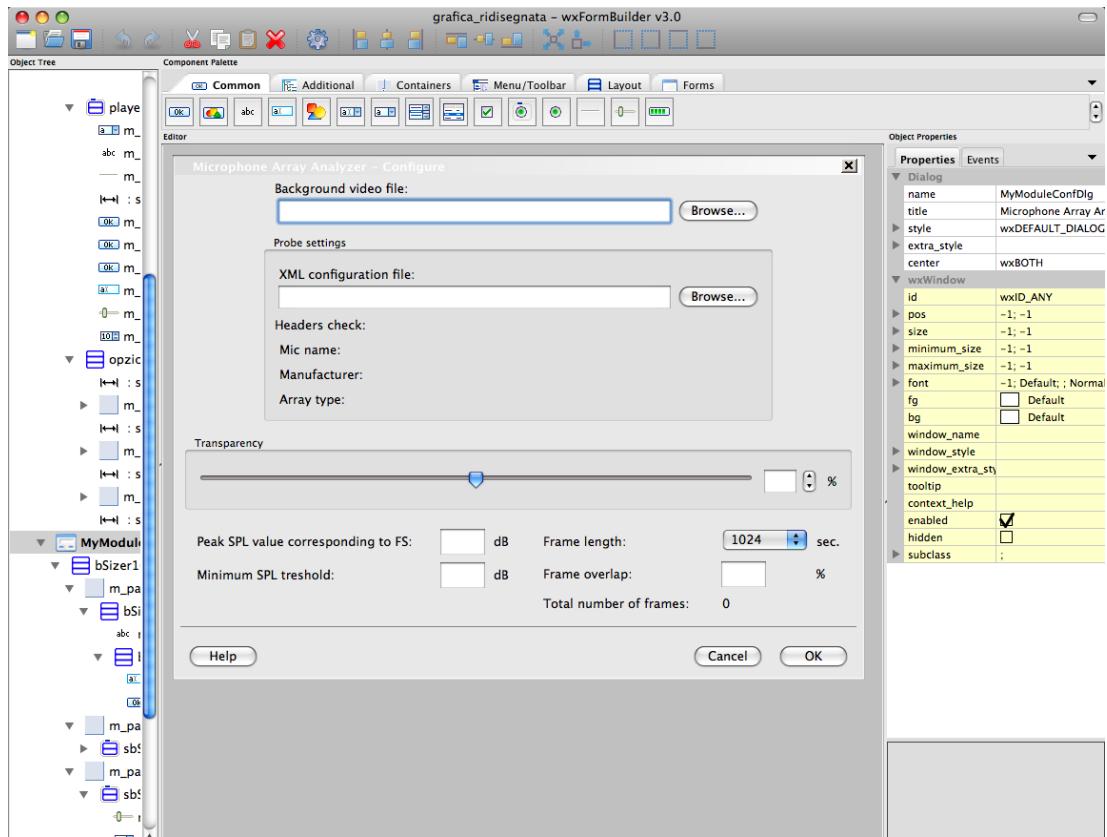
Le **wxWidgets** sono librerie C++ che consentono al programmatore di creare e gestire facilmente interfacce grafiche compilabili sulla maggior parte delle piatta-

forme più diffuse sia a 32 bit che a 64 bit. Questo pacchetto non è l'unico della sua tipologia; esistono infatti alternative come Tk, GTK, etc., ma tra le altre **wxWidgets** spicca sicuramente grazie alla maturità del progetto e alla capacità di conferire alla propria **GUI** l'aspetto tipico del sistema operativo sul quale il codice verrà compilato, grazie all'utilizzo delle *API* della piattaforma invece di una emulazione della interfaccia grafica. Grazie all'utilizzo di **wxWidgets**, si ha accesso a una serie di funzioni avanzate già implementate e contenute in un **wrapper** che ne *incapsula* e nasconde al programmatore la dipendenza dalla singola piattaforma, fungendo da una sorta di *middle-ware* e ponendo il livello di programmazione al di sopra dei dettagli di implementazione dipendenti da una specifica piattaforma. Per notare il livello di maturità del progetto basti pensare che vengono fornite funzionalità di *multithreading*, salvataggio e caricamento di immagini nei formati più diffusi, supporto per *database*, *HTML*, *drag-and-drop*, *streams* multimediali, gestione della **clipboard** e altro ancora. Non è un caso che la stessa interfaccia grafica di *Audacity*<sup>®</sup> faccia utilizzo di queste librerie. Possiamo quindi individuare anche nella coerenza di interfaccia con il proprio host (su qualunque piattaforma si trovi a dover funzionare il programma) un motivo valido per scegliere di utilizzare proprio **wxWidget**.

Per la creazione di interfacce grafiche basate su **wxWidget**, si è individuato un editor di tipo visuale, di nome **wxFomBuilder**, sicuramente di grande utilità e immediatezza nonché di estrema completezza, tanto che si è riusciti a implementare quasi interamente l'interfaccia con questo programma.

## 5.2 Altre librerie

Durante lo sviluppo, si è ampiamente fatto uso di algoritmi e librerie esterne, in modo da sfruttare i risultati noti più importanti (nel caso di algoritmi matematici) e per gestire in maniera trasparente alcune *features* avanzate del software. Ne elenchiamo alcune tra le più importanti con una breve descrizione.



**Figura 5.2:** Intorno per la costruzione di interfacce grafiche wxFormBuilder

**FFT** algoritmo<sup>1</sup> utilizzato in modo massiccio in acustica in materia di elaborazione di segnali. In questo caso si è utilizzata una implementazione propria dello stesso codice sorgente dell'host *Audacity*<sup>®</sup>.

**Triangle++** *wrapper* scritto nel linguaggio C++ per la libreria Triangle che implementa il famoso algoritmo della *triangolazione di Delaunay*<sup>2</sup> utile a partizionare una superficie con sotto-superfici di area triangolare.<sup>3</sup>

**libSndFile** libreria gratuita per la lettura-scrittura e elaborazione di file audio.<sup>4</sup>

<sup>1</sup> <https://code.google.com/p/audacity/source/browse/audacity-src/trunk/src/FFT.cpp>

<sup>2</sup> Si veda il testo [6].

<sup>3</sup> <http://www.complexfoundry.com/piyush/scripts/triangle/>

<sup>4</sup> <http://www.mega-nerd.com/libsndfile/>

**TinyXML** libreria di ausilio alla lettura e scrittura su file `xml` per scambio di dati tra programmi.<sup>5</sup> In questo caso è stato usato un foglio `xml` come file di configurazione per la caratterizzazione dell'array microfonico utilizzato.

Di queste librerie e del loro utilizzo si parlerà più approfonditamente nei capitoli seguenti sull'implementazione, mentre alla libreria **FFmpeg** si è dedicata la seguente apposita sezione .

### 5.3 Libreria di codifica audio e video: **FFmpeg**

Per sviluppare una mappa acustica dinamica ottenuta dalla sovrapposizione di due video generati dall'elaborazione audio, è di vitale importanza l'elaborazione audio-video. Per gestire le problematiche legate alla lettura, codifica, interpretazione, sincronizzazione dei due flussi audio video si è deciso di ricorrere a una delle librerie più largamente utilizzate, nonché già presenti nel bundle di *Audacity®* : **FFmpeg**.<sup>6</sup>

Si tratta di una soluzione completa e, *cross-platform*, per registrare, convertire e elaborare *stream* audio e video.

Questa suite contiene le librerie `libavformat`, per interpretare i formati di *multimedia container* con i `muxers` e `demuxers` contenuti al suo interno, `libavcodec`, per l'utilizzo dei `codec` audio e video, la libreria `libavutil` contenente molte *core multimedia utilities*, `libswwresample` la quale è in grado di effettuare operazioni di `resampling` audio e video nonché di conversione del formato dei sample, in maniera altamente ottimizzata.

**FFmpeg** è gratuita e rilasciata sotto la licenza *GPL*. Le *routines* di questa libreria sono state utilizzate per decodificare il video di background nonché per

---

<sup>5</sup> <http://www.grinninglizard.com/tinyxml/>

<sup>6</sup> <http://www.ffmpeg.org/about.html>

esportare i risultati nella forma di frame singoli oppure di un video intero. Di questo argomento si discuterà meglio in seguito.<sup>7</sup>

---

<sup>7</sup> Si confronti la sezione 7

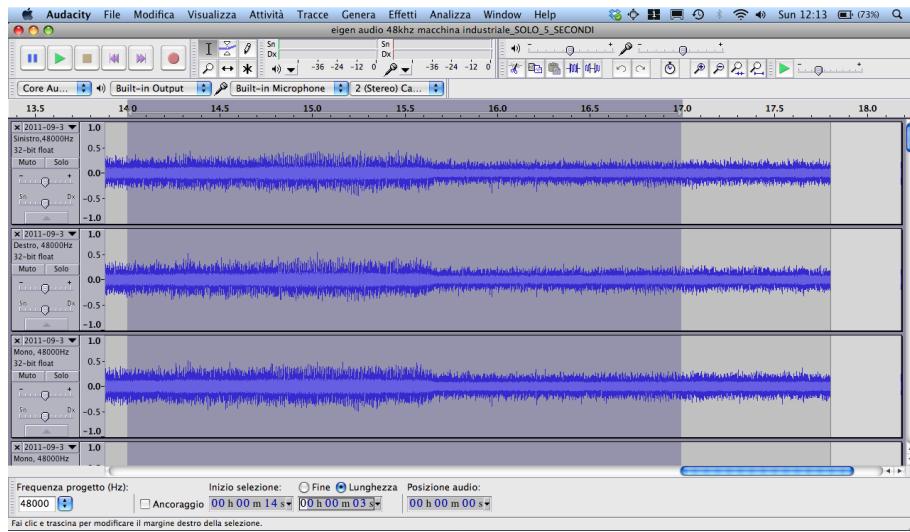
# Capitolo 6

## Descrizione plug-in: un manuale d'uso

Le operazioni compiute dal software qui descritto si dividono sostanzialmente in tre parti:

- una prima parte di *configurazione*
- una parte di *precalcolo*
- una parte di *customizzazione* dei risultati tramite l'utilizzo di varie opzioni fornite dalla *GUI*

In questo capitolo si vogliono descrivere le funzionalità a disposizione dell'utente, nell'utilizzo del *plug-in*, sia in fase di configurazione che in fase di presentazione e analisi dei risultati, mentre la parte di calcolo verrà descritta più approfonditamente nel capitolo 7 alla luce dei parametri utilizzati e presentati.



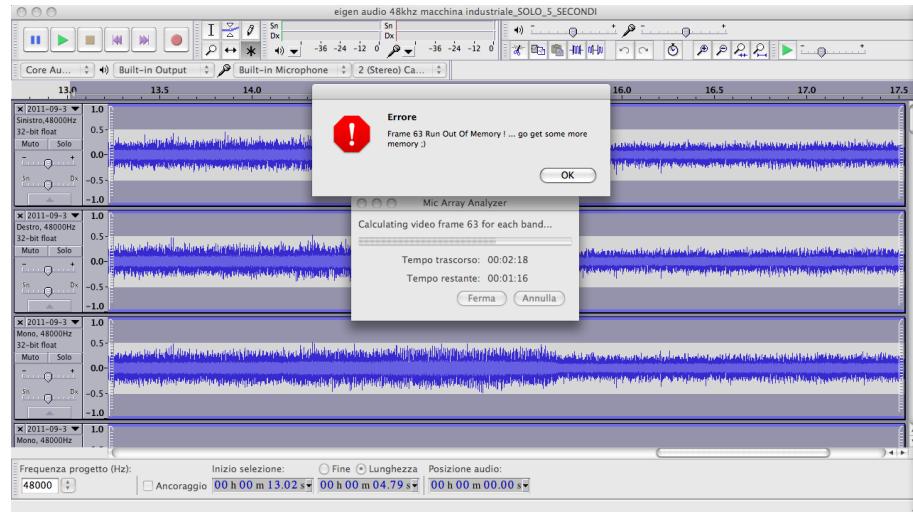
**Figura 6.1:** Selezione dello spezzone di materiale audio da analizzare con il *plug-in*.

## 6.1 Selezione dell'intervallo temporale da analizzare

Prima di lanciare il *plug-in*, è necessario caricare il file audio della registrazione multicanale acquisita con l'*array microfonico* nell'host *Audacity®*.

Di tutta la registrazione, che può in alcuni casi durare anche qualche ora, potrebbe essere necessario selezionare solo una parte, in modo da focalizzarsi nell'analisi su un particolare intervallo di tempo, all'occorrenza anche molto breve. Si pensi per esempio all'analisi delle prime riflessioni in una sala da concerto: in questo tipo di studio, si è interessati al comportamento di un ambiente sotto l'aspetto degli echi e altri effetti. A questo scopo è necessario selezionare la porzione di audio di interesse, come indicato in figura 6.1. Così facendo si limita il materiale audio che *Audacity®* deve editare attraverso i *plug-in* (denominati *effects*).

In alternativa è possibile selezionare anche l'intera registrazione, ben sapendo che l'analisi compiuta nel *precalcolo* è computazionalmente molto onerosa, nonché esigente per quanto riguarda l'utilizzo delle *risorse*, perciò va incontro alla possibilità che in questa fase il programma si interrompa prima di aver analizzato



**Figura 6.2:** Messaggio di errore in caso di memoria insufficiente per le operazioni di *precalcolo*

tutti i frame. In tal caso verrà presentato un messaggio di avviso, come da figura 6.2, e il programma passerà alla visualizzazione dei soli risultati analizzati fino a quel momento. Si tenga presente che su un normale laptop con processore *dual core* da  $2.0\text{ GHz}$  con  $2\text{ GB}$  di *RAM*, l'elaborazione più lunga complessivamente effettuabile consta di circa 60 frames e impiega circa 90 secondi per il *precalcolo*.

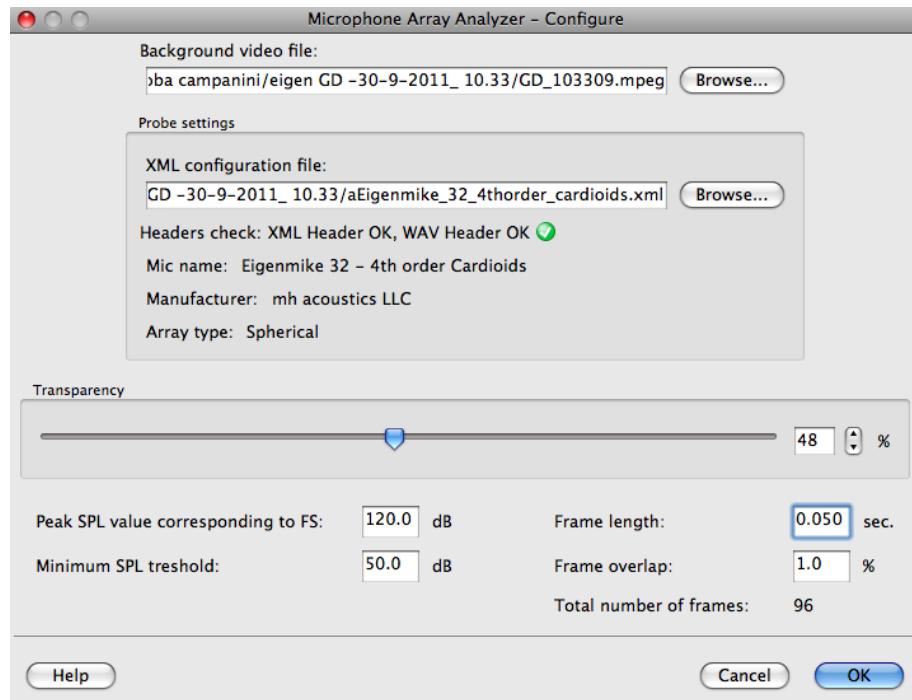
Sarà sempre possibile costruire un video a partire dai singoli frames esportati in diverse elaborazioni del plug-in, operando su spezzoni audio consecutivi.

## 6.2 Finestra di configurazione

Una volta selezionato il materiale audio da analizzare, dal menu *Effect* di *Audacity*<sup>®</sup> è possibile accedere al *plug-in* in oggetto, alla voce *Microphone Array Analyzer*.

La prima finestra che appare all'utente è mostrata in figura 6.3. Si tratta di una finestra di dialogo dedicata all'inserimento di alcuni parametri per l'analisi del materiale audio precedentemente selezionato.

Attraverso questa finestra l'utente dovrà specificare:



**Figura 6.3:** Finestra di configurazione del modulo *Microphone Array Analyzer*

**un videoclip dell’ambiente di background** acquisito, con una videocamera adatta allo scopo<sup>1</sup>, durante la registrazione dell’evento sonoro, su cui mappare i livelli *SPL* con le bande di colore;

**un file di configurazione** in formato *xml* contenente la mappatura in *pixel* delle posizioni delle singole capsule microfoniche sulla immagine di background. Il file contiene anche una serie di informazioni sull’array stesso, come nome e tipologia, casa produttrice ecc,

**il livello di trasparenza** della colormap al sovrapporsi al video di background;

**il livello min SPL** cioè il limite inferiore della scala di *dB* che si vuole analizzare; I pixel eventualmente presenti nella *color-map* con valori SPL che non superassero questa soglia verrebbero disegnati completamente trasparenti;

<sup>1</sup> Per una descrizione delle telecamere nei singoli array si ritorni al capitolo 3

**livello di fondo scala FS** utile per effettuare una sorta di taratura che adeguerà verso quel limite tutti gli altri valori misurati per dare loro una verosimiglianza fisica;

**frame length** dello spezzone audio elementare in cui si vuole suddividere il materiale importato per effettuarne l'analisi. Corrisponde alla lunghezza del frame del video finale di output;

**frame overlap** cioè la percentuale con cui ogni frame si sovrappone nel tempo a quello a lui immediatamente precedente in modo da aggiungere un certo grado di *smoothing* per controbilanciare l'operazione di finestratura<sup>2</sup> svolta dal *plug-in* in seguito alla suddivisione del materiale di analisi;

La coerenza dei dati verrà verificata automaticamente e, in caso di rilevamento di incongruenze, verrà visualizzato un messaggio di errore con una breve spiegazione.

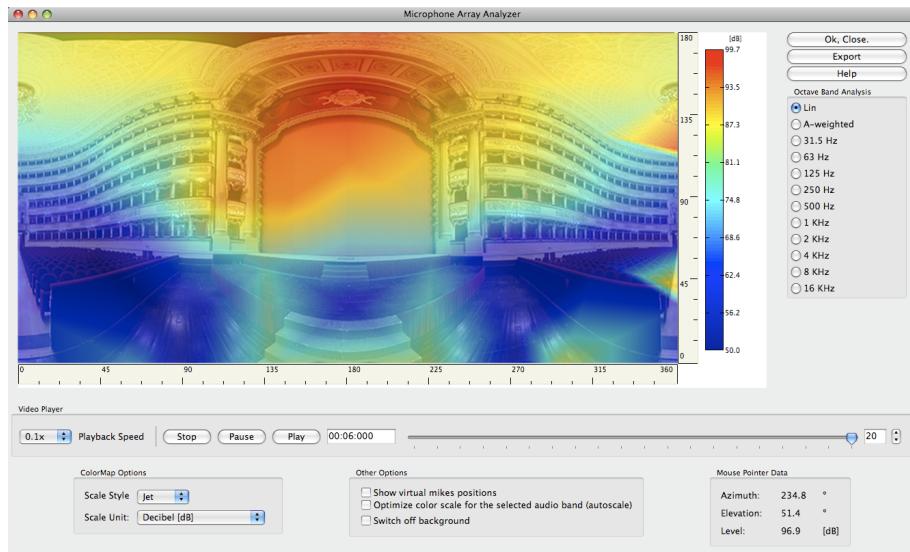
## 6.3 Interfaccia principale

Come possiamo notare in figura 6.4, la finestra principale del *plug-in* mostra un video player contenente un video di output preliminare, ottenuto utilizzando un set di opzioni di default. Si ha però accesso anche a tutte queste opzioni direttamente dagli altri pannelli della finestra, potendo quindi modificare ciascuna di queste opzioni e di conseguenza il video di output. Si procede ora con la descrizione, una per una, di queste configurazioni ulteriormente disponibili:

**Selettore di banda di frequenze** secondo cui filtrare i risultati e osservare quindi al meglio il comportamento delle onde sonore in una banda particolare. Si può quindi discriminare l'emissione di energia sonora in una particolare regione di spettro. Questo tipo di analisi può essere molto utile

---

<sup>2</sup> Descritta al capitolo 7.3



**Figura 6.4:** Finestra principale del modulo *Microphone Array Analyzer*

quando si cerca di studiare un campo acustico dovendo discriminare diverse sorgenti sonore o particolari riflessioni di uno stesso impulso. Grazie al *filtro passa-banda* è possibile eliminare dall’analisi del campo acustico tutto il materiale sonoro di minor interesse evidenziando quindi l’oggetto di attenzione in fase di test.

**Selettore di scala di colore** nella modalità di un semplice menù a tendina che fornisce una gamma di tre possibilità cromatiche, *hot* o *cold*, intuitivamente colori caldi o freddi, e *jet* che invece passa dai freddi ai caldi mano a mano che aumenta il valore SPL che si vuole rappresentare. Grazie a questa opzione è possibile variare la gradazione di colore usata per rappresentare i valori numerici sulla mappa, in modo da massimizzare il contrasto con il video sullo sfondo e rendere il grafico di più comoda interpretazione. Naturalmente, è possibile cambiare opzione anche durante la riproduzione video, come nel caso del cambio di *filtro passa-banda* in modo da analizzare con due scale (o filtri) diversi due parti diverse del video.

**Selettore di unità di misura** con cui rappresentare i livelli sonori. La scelta può ricadere in una delle quattro possibilità offerte dal menù a tendina:

decibel ( $dB$ ), pascal ( $Pa$ ), radice quadrata e cubica della pressione ( $\sqrt{Pa}$  e  $\sqrt[3]{Pa}$ )

**Mostra/nascondi i microfoni virtuali** consente, selezionando l'apposita *checkbox*, di visualizzare sulla colormap le posizioni dei microfoni virtuali contrassegnati da una piccola croce bianca.

**Abilita/disabilita autoscale** permette di modificare i valori di massimo e minimo dei livelli SPL associati alla scala di colore rispettivamente considerando il massimo e minimo rispetto al materiale audio filtrato nella banda di frequenze attualmente selezionata, o semplicemente utilizzando i valori pre-filtraggio nel caso in cui la funzione di autoscale non sia selezionata. Questo permette di apprezzare maggiormente i piccoli cambiamenti di livello di pressione misurati in punti vicini tra loro, al fine di notare maggiormente la direzione di provenienza di una sorgente piuttosto che di una riflessione particolare all'interno di un campo acustico complesso. Sarà sufficiente selezionare la banda di frequenze in cui risiede il suono che si vuole evidenziare per ottenere il risultato voluto.<sup>3</sup>

**Disabilita/abilita lo sfondo** in modo da poter vedere correttamente i colori della mappa acustica senza *l'interferenza* del video di background visibile in trasparenza. Per esempio potrebbe essere utile in casi complessi poter visionare solo alcuni frame, *spegnendo* il video di background, per poterlo poi riaccendere in seguito.

Oltre a queste opzioni con cui personalizzare i risultati ottenuti, è presente nell'interfaccia grafica anche un ultimo pannello non interattivo che fornisce l'informazione riguardante il livello sonoro, nell'unità di misura scelta, relativo alla

---

<sup>3</sup> Si noti che attraverso la medesima operazione è anche possibile effettuare l'analisi opposta e cioè risalire alla banda frequenziale di maggior rilievo in  $dB$  per quanto riguarda il rumore di una sorgente di cui si conosce la direzione, magari nel caso in cui ci si trovi all'interno di un campo acustico complesso in cui è impossibile isolare la sola sorgente interessata.

posizione del mouse sulla mappa e quindi a una coordinata sferica nel già citato<sup>4</sup> sistema di riferimento antropometrico centrato sull'array microfonico.

Al termine delle operazioni è possibile esportare i risultati tramite l'apposito bottone *Esporta*.

---

<sup>4</sup> Secondo la norma ISO2631, come descritto al capitolo 3.1.1 e in figura 3.7

# Capitolo 7

## Dettagli di Implementazione

In questo capitolo si analizzeranno i dettagli del lavoro computazionale che il tracciamento della mappa comporta. La spiegazione, analogamente al flusso di implementazione, segue i seguenti step:

1. Acquisizione dei dati;
2. Organizzazione del carico computazionale;
3. Finestratura sui frame audio;
4. Mirroring dei microfoni virtuali, per garantire la continuità ai bordi della mappa;
5. Copertura dell'intera area della mappa con superfici di forma triangolare (*meshing*);
6. Filtraggio in bande d'ottava;
7. Scalatura delle ampiezze dei segnali risultanti in funzione del fondo scala specificato nella finestra di configurazione;
8. Interpolazione dei dati;
9. Applicazione della *color-map* in relazione all'opzione di *auto-scaling*;

10. Sincronizzazione audio-video;

11. Esportazione dei risultati.

Al capitolo 4 si è già analizzato lo step preliminare che consiste nella sintesi della matrice dei filtri che consentono di ottenere i microfoni virtuali. Sarà quindi necessario, a monte dell'utilizzo del *plug-in*, effettuare la convoluzione tra i dati registrati dall'array e i filtri sintetizzati attraverso un modulo esterno, come per esempio il convolutore della suite *Aurora*<sup>1</sup>, di nome *Convolver*.<sup>2</sup>

Si procederà di seguito con la spiegazione in dettaglio dei singoli passaggi di implementazione.

## 7.1 Acquisizione dei dati in ingresso

Per il corretto funzionamento del *plug-in*, abbiamo visto, nel capitolo precedente alla sezione 6.2, che sono necessari i seguenti dati in ingresso:

- una **registrazione multicanale** effettuata con l'array microfonico in uso e prefiltrata in modo che i segnali rappresentino già le uscite dei microfoni virtuali;
- un **videoclip dell'ambiente di background**;
- le **coordinate** in *pixel* delle posizioni dei singoli microfoni virtuali sulla immagine di background.

Di tutti questi dati viene controllata la coerenza in fase di configurazione.<sup>3</sup>

---

<sup>1</sup> Si veda [2]

<sup>2</sup> Scaricabile gratuitamente in versione per Windows, Linux o Mac OS, dal sito  
<http://pcfarina.eng.unipr.it/Public/Aurora-for-Audacity>

<sup>3</sup> Di nuovo si confronti il capitolo 6.2

Per quanto riguarda la **registrazione multicanale** direttamente proveniente dall'array, è lo stesso host *Audacity*<sup>®</sup> a farsi carico di rendere disponibile la registrazione al *plug-in* tramite l'utilizzo di apposite *routines*.

La seconda componente fondamentale per la costruzione della mappa acustica è il **videoclip di background**. Essendo a conoscenza del *path* nel *filesystem*, fornитoci in fase di configurazione dall'utente, utilizziamo alcune *routines* basate sulle *API* della suite **FFmpeg** per acquisire il video e salvarlo in memoria diviso in singoli frames (quindi immagini di tipo bitmap o per meglio dire **wxBitmap**). A tale scopo è stata creata una struttura dati complessa composta da due classi fondamentali:

- la classe **Video** che si occupa di modellizzare il video finale di output;
- la classe **VideoFrame** la quale implementa la struttura logica dei singoli frame del video e contiene tutte le informazioni di pressione, livelli *SPL*, le varie bitmap di sfondo e di mappe di colore per ogni banda frequenziale, nonché le informazioni temporali di *timestamp* riguardanti la sincronizzazione tra audio e relativa mappa acustica e video.

Infine le **coordinate** dei microfoni virtuali vengono lette dal file **xml** attraverso le *routines* della libreria **tinyXml** e salvate in una struttura appositamente creata chiamata **MikesCoordinates**.

## 7.2 Organizzazione del carico computazionale

A questo punto, avendo caricato in memoria tutti i dati necessari, è possibile iniziare l'elaborazione vera e propria. Tutta la parte che segue riguarda una descrizione puramente implementativa, che chiameremo fase di *precalcolo*, la qua-

le dal punto di vista dell’utente corrisponde a una semplice *progress bar* che comunicherà lo stato di avanzamento del programma durante l’elaborazione.

Trattandosi dell’elaborazione di un video, si è pensato di parallelizzare il *pre-calcolo* in modo da approfittare della ripetitività del flusso di lavoro che deve analizzare i dati dividendoli in piccoli *chunk* temporali corrispondenti ai frame video che si vorranno ottenere.

Si procede dunque all’allocazione dinamica di svariati *thread* paralleli che correranno alla formazione del video finale, calcolando un singolo frame ciascuno. Per non sovraccaricare la CPU con un eccessivo numero di thread, si è avuta la cura di lanciare i singoli thread in modo che non fossero mai in esecuzione più di uno per *core* contemporaneamente.

Nell’implementare questa parallelizzazione, si è utilizzata la classe fornita da `wxWidgets` chiamata `wxThread` in modo da mantenere lo stile di programmazione multipiattaforma prefissato. Si tratta di una implementazione che aderisce fedelmente allo standard `POSIX`, ma che contiene alcune classi avanzate in grado di semplificare abbastanza la gestione delle corse critiche e della mutua esclusione tra gli accessi concorrenti.

Ogni *thread* dunque si incaricherà di memorizzare in una sua struttura appropriata i dati riguardanti il suo *chunk* di audio e procederà con l’esecuzione di tutti i passi che verranno descritti nelle prossime sezioni. Alla fine del proprio iter procedurale andrà a salvare i suoi risultati in una struttura `VideoFrame` che verrà aggiunta (nella posizione coerente con il numero di frame assegnato al thread stesso) alla struttura condivisa `video`, naturalmente rispettando i principi che regolano l’accesso concorrente. Il main thread, dopo aver *aspettato* tutti gli altri thread, procederà a liberare la memoria e aggiungere informazioni globali derivanti dall’analisi dell’intero video (come il massimo assoluto di fondoscala).

## 7.3 Finestratura dei dati di ogni frame audio

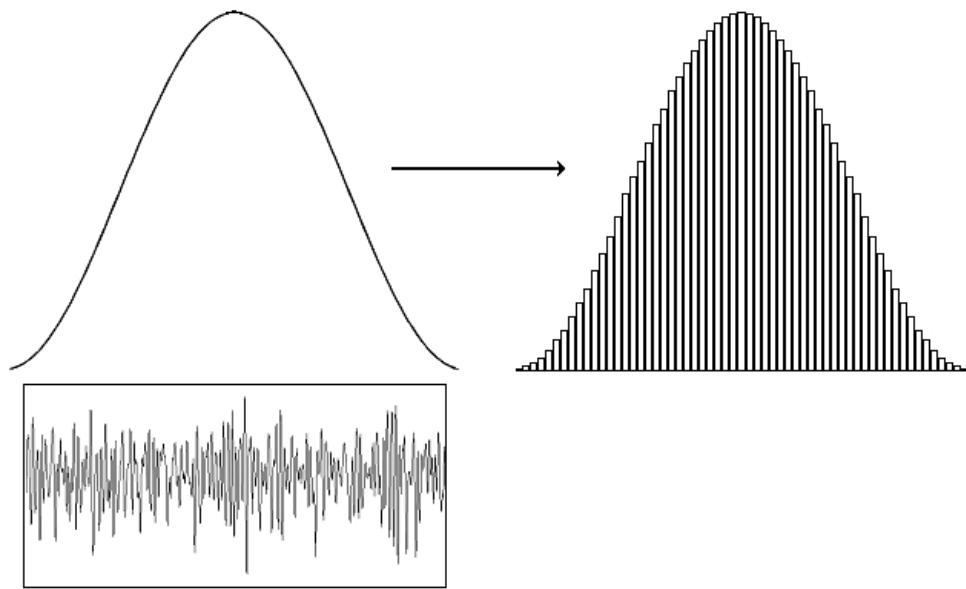
La tecnica di *finestratura* viene utilizzata per pesare i *chunk* di dati acquisiti nel dominio del tempo al fine di minimizzare gli effetti di bordo che darebbero luogo a *leakage* nell'analisi spettrale. Utilizzando correttamente le funzioni di finestratura è possibile invece aumentare la risoluzione del risultato che si otterrà nel dominio della frequenza durante l'analisi spettrale da effettuare più avanti nella trattazione. Nell'operare un'analisi di tipo digitale attraverso FFT per misurare il contenuto frequenziale dei dati acquisiti, si dovrà basare l'analisi sul set di dati finito dato dai samples audio selezionati. La trasformata FFT invece assume che il set finito sia un periodo di un segnale periodico, infatti per questo algoritmo, sia il dominio del tempo che della frequenza sono di topologia circolare. Perciò i due estremi dello spezzone di audio analizzato verranno interpretati come se fossero connessi l'uno all'altro risultando in effetti di discontinuità ad alta frequenza sui bordi. Per minimizzare questo effetto si ricorre a una funzione di finestratura del segnale misurato nel dominio del tempo in modo da forzare i due estremi a incontrarsi a zero dando luogo quindi a un segnale che non presenti brusche transizioni sui bordi.

Per l'implementazione della finestratura si è scelto di utilizzare la finestra di Hanning che assegna un peso  $w(n)$  in funzione del numero di *samples* sul totale  $N$  secondo la legge

$$w(n) = \frac{1}{2} \left[ 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right], \quad 0 \leq n \leq N-1. \quad (7.1)$$

## 7.4 *Mirroring* dei microfoni virtuali

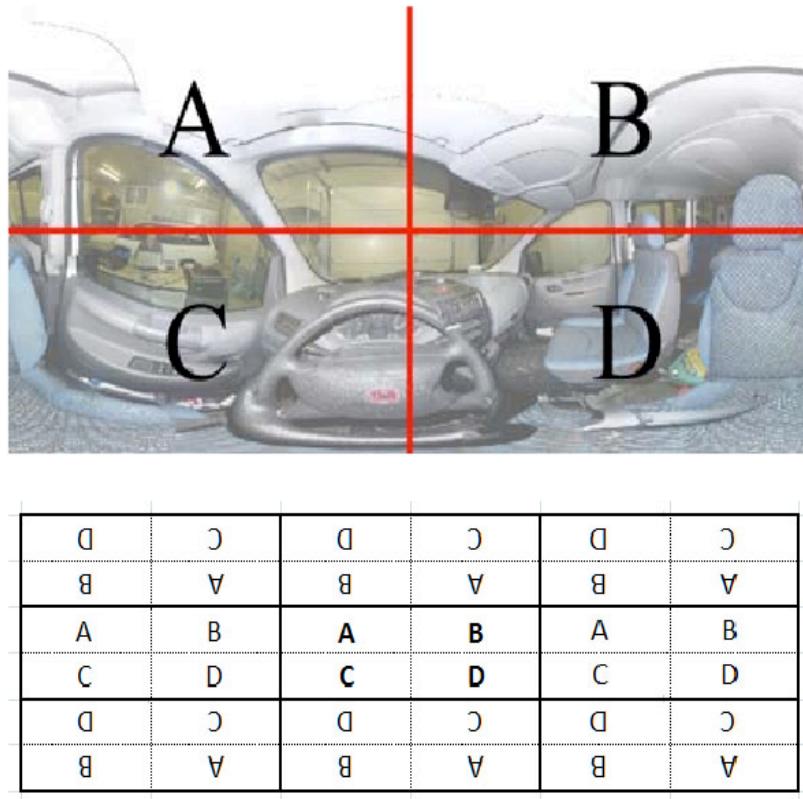
La registrazione effettuata con un array sferico non produce una mappa dei livelli sonori come proiezione cilindrica di una mappa sferica, come ad esempio una mappa ottenibile con un procedimento simile a quello usato per la rappresentazione



**Figura 7.1:** Finestra di Hanning nel dominio del tempo per l'analisi digitale.

della superficie terrestre sul planisfero. Muovendo dall'esempio del planisfero, è noto che, essendo la terra sferica, muovendosi lungo i bordi del planisfero si avrà una certa continuità della mappa, ovvero uscendo da nord si rientrerà da sud, uscendo da est si rientrerà da ovest etc. Nel caso particolare di tracciamento della mappa sonora, si è proceduto prolungando la mappa stessa oltre i propri bordi. Per fare ciò si è effettuata una estensione della mappa *specchiando* i punti relativi ai microfoni virtuali sintetizzati seguendo lo schema proposto nel documento [1] e riportato in Figura 7.2.

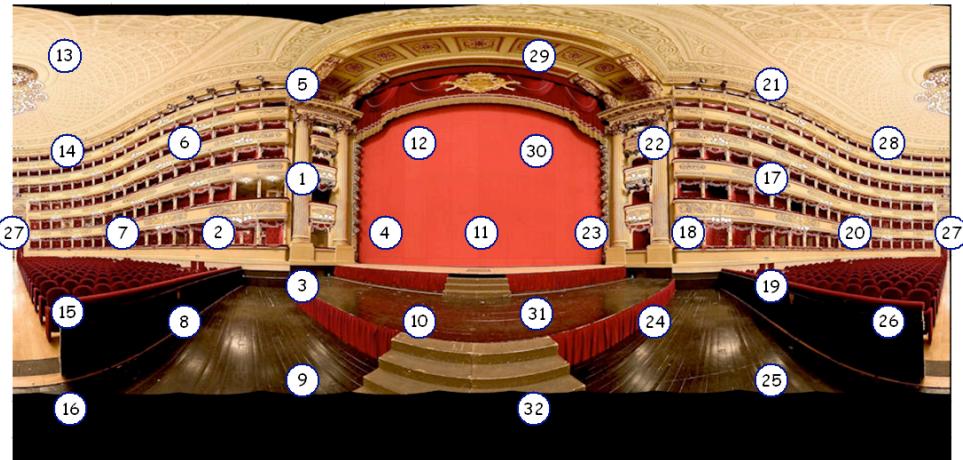
Dalla figura si osserva che i quattro quadranti in cui può essere scomposta la fotografia sferica, A, B, C e D, debbano essere *copiati e specchiati* verticalmente e/o orizzontalmente attorno alla fotografia, per imporre la condizione di continuità; più precisamente si è proceduto copiando le posizioni e i livelli registrati dai microfoni virtuali in modo da riuscire a coprire con delle *mesh* triangolari anche i bordi della mappa, così come meglio spiegato nel prossimo paragrafo.



**Figura 7.2:** Rappresentazione schematica del processo di *mirroring*, necessario a garantire la continuità ai bordi della mappa.

## 7.5 *Meshing* della superficie mediante l'operazione di triangolazione di *Delaunay*

Per eseguire l'interpolazione di una serie di livelli noti di una funzione in due variabili, l'ascissa e l'ordinata, occorre suddividere l'intero piano in aree più piccole di forma poligonale, aventi come vertici tre o più coppie di coordinate in cui i livelli siano noti; occorre poi ipotizzare, per ciascuna di queste aree, che i livelli ai vertici facciano tutti parte di un'unica funzione notevole, ed infine calcolare i valori che tale funzione assume in tutti i punti compresi quelli di ognuna delle sottoaree in cui si è suddivisa la superficie. Questo primo passaggio è detto *me-*



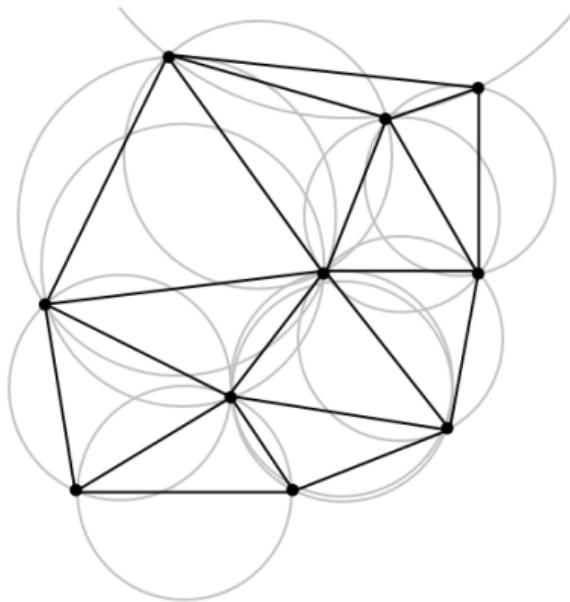
**Figura 7.3:** Direzioni di puntamento delle capsule dell'*Eigenmike<sup>TM</sup>* sovrapposte ad una foto panoramica del Teatro alla Scala di Milano, svolta secondo lo schema di proiezione visto nella sezione 7.4.

*shing* della superficie, mentre le sottoaree che vanno a ricoprire l'intera superficie prendono il nome di *mesh*.

Il primo passo da fare è stabilire il metodo da seguire per determinare la griglia, ovvero la forma geometrica delle *mesh* in cui l'intero piano xy andrà suddiviso. La prima ipotesi potrebbe essere la suddivisione in aree di forma quadrata (come i meridiani e i paralleli del planisfero): questa ipotesi è valida se i microfoni virtuali, che costituiscono la griglia dei punti in cui i livelli sonori sono noti, sono disposti su una griglia uniforme.

Come si può notare dalla Figura 7.3, le posizioni delle capsule dell'*Eigenmike<sup>TM</sup>* non lo sono, e se anche così fosse, lo sviluppo del plug-in sotto l'ipotesi di griglia uniforme avrebbe portato alla realizzazione di un software utilizzabile per un numero limitato di disposizioni dei microfoni virtuali.

La seconda ipotesi è quella di suddividere l'intero piano xy in superfici di forma triangolare; in questo modo, anche nel caso in cui le posizioni dei microfoni fossero disposte nella maniera più casuale possibile, si potrà sempre determinare un insieme continuo di triangoli che copra l'intera superficie. Nel 1925 è stato,



**Figura 7.4:** Descrizione della proprietà dei triangoli formati con l'algoritmo di *Delaunay*.

infatti, dimostrato che ogni superficie può essere partizionata attraverso superfici di forma triangolare, ma questo può richiedere un numero infinito di triangoli. Per effettuare questa operazione si è scelto quindi di utilizzare una triangolazione particolare detta di *Delaunay* che è definita come segue:

*Una triangolazione di un insieme finito di punti  $P \subset R^2$  viene detta di Delaunay se il cerchio circoscritto ad ogni triangolo è vuoto, ovvero nessun punto di  $P$  vi giace all'interno.*

Sono diversi gli algoritmi che consentono di determinare, dato un insieme di punti sparsi su un piano, la triangolazione di Delaunay. I principali, con complessità differente in funzione del numero di punti da triangolare, sono:

- l'algoritmo incrementale;
- l'algoritmo *dividi et impera*;
- l'algoritmo Convex Hull.

Per il presente lavoro di tesi si è deciso di utilizzare una libreria esterna, denominata **Triangle++**, per implementare l'algoritmo di *Delaunay*, essendo l'implementazione particolarmente impegnativa. **Triangle++** è il nome del wrapper C++ della libreria C **Triangle**<sup>4</sup>; esso fornisce la definizione di una classe molto semplice da utilizzare, la quale, dato un insieme di punti di cui si vuole conoscere la triangolazione, svolge autonomamente il calcolo delle mesh facendo uso di tutti e tre gli algoritmi sopracitati.

## 7.6 Filtraggio in bande d'ottava

L'algoritmo di filtraggio consiste dei seguenti passaggi:

1. Calcolo della trasformata discreta per passare nel dominio della frequenza
2. Calcolo dell'*autospettro* per ottenere dei valori di potenza reali
3. Accumulazione delle righe spettrali per le singole bande

Per passare al dominio della frequenza e ottenere uno spettro di potenza (punti 1 e 2), essendo una operazione abbastanza comune, *Audacity*<sup>®</sup> fornisce una sua implementazione dell'algoritmo **FFT** incapsulato in una routine di nome **Power-Spectrum** che si occupa di fornire direttamente lo spettro di potenza a partire da un segnale reale nel dominio del tempo. Si otterrà quindi una rappresentazione dello spettro di potenza composta da un numero  $N$  di righe spettrali, corrispondente alla frequenza di *Nyquist*, cioè alla metà dei campioni analizzati:

$$N = \frac{N_l}{2} + 1, \quad (7.2)$$

dove  $N_l$  è il numero di samples audio che costituiscono il frame. Inoltre per il *teorema di Shannon* la banda passante  $B$  rappresentabile dallo spettro di potenza

---

<sup>4</sup> Si veda [11]

sarà :

$$B = \frac{N_r}{2}, \quad (7.3)$$

dove  $N_r$  è la sample rate di campionamento della registrazione

Quindi ad ogni riga spettrale così calcolata corrisponderà un intervallo di frequenze  $\Delta f$  calcolato come segue:

$$\Delta f = \frac{B}{N} \quad (7.4)$$

A questo punto si prosegue accumulando le potenze contenute nelle singole righe spettrali per ogni banda analizzata nonchè per tutto lo spettro ad esclusione delle due bande più basse in cui l'array non può avere una buona risoluzione (analisi *lineare*) e infine si calcola una versione dell'analisi lineare misurata in dB(A).<sup>5</sup> I livelli vengono sommati su un array che contiene un accumulatore per banda. Su ogni accumulatore verranno sommate le righe spettrali che cadono nella banda di pertinenza, cioè il cui indice di riga spettrale  $n$  soddisfa la relazione

$$n \cdot \Delta f \in [f_i, f_s] , \quad (7.5)$$

dove  $f_i$  e  $f_s$  sono le frequenze di bordo che caratterizzano la banda stessa.

Per quanto riguarda l'accumulatore denominato *Lineare* esso comprende tutte le bande ad eccezione delle prime due le quali si è reputato non contenessero altro che rumore in quanto fuori dalla zona di risoluzione frequenziale dell'array.

Il valore pesato A invece le comprende proprio perchè a seguito della pesatura con la curva di ponderazione, sono proprio quelle frequenze le più penalizzate in quanto anche il sistema uditivo umano risponde male ai segnali all'estremità grave dello spettro udibile.

Per l'implementazione della pesatura con la curva A si è utilizzato l'approccio analitico codificando la funzione che definisce tale curva, cioè che rappresenta il peso  $A$  dipendente dalla frequenza della riga spettrale come nella relazione

$$A = 10 \log \left( \frac{3.5041384 \cdot 10^{16} \cdot f^8 \cdot (122194.217^2 + f^2)^{-2}}{(20.598997^2 + f^2)^2 \cdot (107.65265^2 + f^2) \cdot (737.86223^2 + f^2)} \right) \quad (7.6)$$

---

<sup>5</sup> Per i filtri di ponderazione si vedano la sezione 2.4.3 e l'appendice A.4

Tutti questi valori vengono calcolati per ogni microfono virtuale.

## 7.7 Applicazione del livello di fondo scala

Il livello di fondo scala ( $FS$ ) specificato dall'utente tramite la finestra di configurazione del *plug-in*, è inteso come livello massimo con il quale fornire un riferimento di pressione acustica al sensore. Posto che non sono ancora stati scoperti metodi efficaci per la taratura degli array microfonici, si dimostra comunque necessario effettuare una sorta di *adeguamento* dei risultati per ottenere una mappa di significato fisico dal punto di vista quantitativo dei livelli sonori e non solo una corretta interpretazione proporzionale di questi. Il problema è di natura elettroacustica e di elaborazione digitale dei segnali audio: il file *wav* come altri standard di questo tipo, rappresentano il suono analogico con un numero finito di *samples* digitali, per poi assegnare a ognuno un valore *quantizzato* compreso in  $[-1, 1]$ . Ora, risulta chiaro che questi non possono essere direttamente la misura dei valori di pressione acustica incidente sulla capsula microfonica, ma sono invece rappresentativi del segnale elettrico in cui essa viene tradotta dal sensore.

Per ritradurre l'informazione in pascal e ottenere quindi un informazione dal significato acustico preciso, è necessaria una operazione di *taratura* dello strumento che permetta di misurare un segnale a frequenza e SPL note a priori, in modo da poter mappare la risposta del sensore ed effettuare una opportuna calibrazione applicando un guadagno in ingresso. Non si è ancora riusciti a riprodurre questo procedimento su un sensore così complesso come un array microfonico, perciò si è escogitato uno stratagemma intuitivo che permetta di effettuare questo adeguamento facilmente, anche se senza pretesa di precisione assoluta. Non si tratta infatti di un'operazione di taratura dello strumento, se non di un semplice *adeguamento* dei risultati a una condizione di *verosimiglianza fisica*.

Si rende necessario quindi misurare, durante la registrazione con l'array microfonico, un segnale di riferimento, per esempio attraverso un fonometro calibrato, dal quale risalire al livello SPL di fondoscala della registrazione. Una volta individuato il picco massimo in dB sarà sufficiente fornire questo valore al *plug-in* in fase di configurazione e il software si occuperà di computare questo semplice adeguamento dei risultati senza variare i rapporti energetici tra i contenuti frequentiali alle varie bande o nelle diverse zone del campo acustico.

Infine, dopo l'operazione di filtraggio e scalatura, è necessario creare un *frame audio* cioè una mappa statica che rappresenti il contenuto del campo acustico nell'intervallo non infinitesimo<sup>6</sup> di un frame del video che si sta mano a mano costruendo.

Per fare ciò procediamo a interpolare i risultati fino ad ora calcolati in corrispondenza dei microfoni virtuali

## 7.8 Interpolazione dei valori acquisiti

Quando si sia calcolata una ipotetica partizione del piano *xy* in *mesh* di forma triangolare, all'interno di ciascuna delle quali ci interessi conoscere i valori assunti dalla funzione incognita dei livelli sonori, occorre scegliere, tra le tante possibili, una e una sola funzione da utilizzare come stima approssimativa di quella incognita, imponendo che soddisfi i livelli noti presenti ai tre vertici della mesh. La soluzione più semplice è quella di scegliere come funzione approssimante l'equazione di un piano passante per tre punti generici. Il metodo di interpolazione che ne deriva va sotto il nome di *interpolazione bilineare*.

Sapendo che l'equazione generica di un piano può essere scritta nella forma:

---

<sup>6</sup> Selezionato dall'utente come spiegato al paragrafo 6.2

$$z = A \cdot x + B \cdot y + C \quad (7.7)$$

e chiamando  $(x_{1i}, y_{1i}, z_{1i})$ ,  $(x_{2i}, y_{2i}, z_{2i})$  e  $(x_{3i}, y_{3i}, z_{3i})$  le coordinate degli unici tre punti noti della funzione incognita dei livelli sonori, dove i valori  $z_{1i}$ ,  $z_{2i}$  e  $z_{3i}$  corrispondono proprio ai livelli misurati in corrispondenza dei tre vertici della  $i$ -esima mesh, l'imposizione del passaggio del piano per i tre punti corrisponde all'equazione matriciale:

$$\begin{pmatrix} x_{1i} & y_{1i} & 1 \\ x_{2i} & y_{2i} & 1 \\ x_{3i} & y_{3i} & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} z_{1i} \\ z_{2i} \\ z_{3i} \end{pmatrix} \quad (7.8)$$

L'equazione 7.8 rappresenta esattamente l'approccio utilizzato dalla classe `TriangularMesh`, definita all'interno del codice del *plug-in* progettato, per il calcolo dei coefficienti  $A$ ,  $B$ ,  $C$  e  $\det(M)$  che servono per l'interpolazione. Quando si desidererà conoscere il livello sonoro in una certa posizione di coordinate (in pixel) che cadono entro una certa *mesh*, si dovrà chiamare la funzione membro dell'oggetto che rappresenta la *mesh* di interesse e che, implementando la 7.8, ci restituirà il valore interpolato.

Questa ipotesi sarebbe corretta nel caso in cui ad ogni vertice di ciascuna mesh fosse possibile associare un unico livello sonoro. Come si è visto nei precedenti capitoli, invece, il *plug-in* consente di effettuare una analisi in bande d'ottava<sup>7</sup> dei livelli sonori, ovvero non si avrà una sola mappa dei livelli, bensì tante mappe quante sono le possibili bande d'ottava selezionabili. Per ovviare a questa situazione occorre usare come terza coordinata di ciascun vertice il numero del microfono virtuale posizionato nelle stesse coordinate del vertice (anzichè il livello sonoro) e memorizzare una matrice dei livelli sonori tale che, confrontando il

---

<sup>7</sup> Si veda la sezione 2.4.2 per la definizione delle bande utilizzate

numero del microfono virtuale con il numero identificativo della banda selezionata, fornisca il livello sonoro corretto per la determinazione dei coefficienti  $A$ ,  $B$ ,  $C$  e  $\det(M)$  necessari alla costruzione della mappa per la particolare banda selezionata.

## 7.9 Mappatura *colormap* e *auto-scale*

La scala di colore che si utilizza è una funzione in una incognita a tre variabili dipendenti: l'incognita è il livello sonoro in un dato punto ed i tre valori di uscita della funzione coincidono con i tre canali *R* (*red*), *G* (*green*) e *B* (*blue*) di una interfaccia video. A seconda della scala di colore disiderata, sarà utilizzata una *routine* diversa per determinare il valore delle tre componenti cromatiche del pixel interessato, il quale sarà successivamente inserito in una *mappa RGB* (una bitmap, appunto). Per assegnare ai valori di *SPL* un valore cromatico è necessario stabilire gli estremi *SPL* da rappresentare per poi successivamente effettuare tutta la scalatura tra i livelli in modo proporzionale. Infatti le relazioni tra i valori *R*, *G* e *B* e il livello che si vuole mappare sono parametrizzate in funzione dei valori di minimo e massimo rappresentabili. Non si tratta di un'operazione banalissima in quanto per definire univocamente questi valori è necessario analizzare tutto il video alla ricerca degli estremi. Questo comporta la perdita della possibilità di lavorare in tempo reale con una registrazione; sarà invece necessario analizzare un audio pre-registrato. Dopo aver calcolato tutti i livelli corrispondenti ai microfoni virtuali, filtrati per ogni banda, ed effettuata questa operazione per ogni singolo frame<sup>8</sup>, il programma è in grado di ottenere e salvare in una struttura dati i valori di *SPL* massimi e minimi per ogni banda e per ogni frame. In questo modo sarà possibile individuare gli estremi cercati.

---

<sup>8</sup> Cioè in seguito al precalcolo descritto in questo capitolo

Con la funzione di `auto-scale` si modificano i valori minimi e massimi entro i quali verrà adattata la scala colorata in base alla banda frequenziale selezionata dall'utente.

Nel caso in cui sia stata abilitata la funzione `auto-scale`<sup>9</sup>, saranno modificati i valori *RMS* di minimo e massimo dei segnali prodotti dai microfoni virtuali limitando la ricerca in ogni frame ai soli valori riguardanti la banda selezionata; se il livello minimo così calcolato dovesse essere inferiore a quello di soglia stabilito dall'utente<sup>10</sup>, si considererà come valore minimo il valore di soglia.

Nel caso in cui invece la funzione `auto-scale` sia disabilitata, il livello minimo verrà assunto pari a quello di soglia inserito dall'utente, mentre quello massimo verrà assunto pari al massimo assoluto tra i livelli *RMS* prodotti dai vari microfoni virtuali in tutte le bande e fra tutti i singoli frame del video di output.

## 7.10 Sincronizzazione audio-video

In realtà è fuorviante parlare di audio-video in quanto anche l'audio, dopo essere stato elaborato, a questo punto della trattazione è stato tradotto in una *color-map* da trattare anch'essa come *video*. Chiaramente è necessario che i due stream scorrono in sincronia. Per ottenere questo risultato si è utilizzato un metodo che va ad agire a livello del singolo frame. Come abbiamo visto all'inizio del capitolo, alla sezione 7.1, sia lo stream audio che lo stream video vengono salvati in memoria suddivisi nei loro singoli frame. Questi frame non sono però necessariamente della stessa lunghezza cioè i due stream facilmente non avranno la stessa frame rate. Si rende dunque necessario ricavare l'informazione di *timestamp* di ogni frame: al momento dell'importazione tramite FFmpeg si salva il valore della frame rate del video di background, mentre il valore della lunghezza del frame

---

<sup>9</sup> Con il *checkbox* descritto nella sezione 6.3

<sup>10</sup> Si veda il paragrafo 6.2

(in numero di *audio samples*) viene inserito dall’utente in fase di configurazione. Dopo aver effettuato opportune conversioni si può risalire alla timestamp di ogni frame sia audio che video e confrontarle. Dato che il numero totale di frame non sarà quasi mai lo stesso, si decide di dare la precedenza allo stream audio, cioè ai frame di colormap, e di porre in sfondo il frame video tra quelli disponibili in memoria che più si avvicina alla timestamp della colormap visualizzata in quel momento.

## 7.11 Esportazione dei risultati

La struttura dati **Video** conterrà le seguenti informazioni generali:

- il livello di trasparenza delle bande colorate rispetto allo sfondo;
- l’*aspect ratio* proprio del video (in *pixel*);
- il numero totale dei frame del video;
- i valori di *SPL* massimo e minimo assoluti per ogni banda.

Inoltre conterrà una struttura di tipo **hash-map** contenenti i singoli **VideoFrame** contrassegnati dal numero di frame come chiave unica. Ogni **VideoFrame** dal canto suo, conterrà le informazioni seguenti:

- *timestamp* del frame ottenuta dal progetto *Audacity*<sup>®</sup> al momento del salvataggio del *chunck* audio;
- immagine **bitmap** di background corrispondente al frame più vicino alla timestamp di cui sopra;
- una matrice contenente per ogni canale *virtuale* registrato (cioè per ogni microfono virtuale ricavato dai filtri di inversione) i valori SPL filtrati se-

condo le 10 bande frequenziali con l'aggiunta del valore filtrato A (dB(A)) e del filtraggio lineare;

- una matrice contenente il valore SPL per ogni pixel dell'immagine, ottenuto a seguito dell'interpolazione dei valori dei singoli microfoni virtuali;
- i valori di massimo e minimo SPL per ognuna delle 10 bande e per gli altri 2 filtri descritti (dB(A) e *lineare*).

A questo punto, ogni `videoFrame` può essere salvato su disco rigido con una funzione della libreria `wxWidgets` che garantirà quindi un'operazione ad hoc funzionante su ogni piattaforma. Parallelamente si procede all'esportazione dell'intero video, con le impostazioni attuali di banda selezionata, auto-scale, color-scale è unità di misura scelte, tramite una chiamata al programma esterno `FFmpeg` presente in ogni installazione di *Audacity*<sup>®</sup> che genererà il video a partire dai singoli frame appena esportati.

# Capitolo 8

## Conclusioni

In questo progetto di tesi si è lavorato alla programmazione di un software che fosse in grado di elaborare registrazioni multicanale prefiltrate, provenienti da un array microfonico generico, congiuntamente a una registrazione video dello stesso evento, in modo da generare in output una mappa sonora dinamica che fornisse una indicazione qualitativa dell’evoluzione del campo acustico. Come scelta di progetto è stata mantenuta la linea di implementare il software in forma di plug-in *Audacity®*, piattaforma che permette ai suoi *plug-in* o *moduli* di accedere a moltissime proprietà dell’host e dati del workspace, a differenza degli altri standard diffusi di plug-in come per esempio i *VST* e gli *Audiounits*.

Il plug-in progettato ai fini di questo progetto è funzionante ed altamente interattivo, permettendo diverse tipologie di realizzazione della mappa grazie alla possibilità di configurare numerose opzioni, quali lo stile della scala di colore, la percentuale di trasparenza della mappa sovrapposta al video di background, l’unità di misura dei livelli mostrati, i valori di fondo scala, la lunghezza del frame video, la percentuale di overlap tra i singoli frame etc.

Dal confronto degli output del modulo in oggetto con altri, calcolati dallo script Matlab™ descritto nel documento [1], si è inoltre riscontrata una certa coerenza tra i risultati ottenuti; ben sapendo che non è ancora stato inventato un metodo realmente applicabile per tarare una misura effettuata con un array

microfonico, si conclude che i risultati ottenuti con l'uso del *plug-in* siano corretti.

In futuro, se si riuscisse a mettere a punto questo metodo di taratura per array microfonici, sarebbe possibile utilizzare il plugin in modo molto efficace in campo rumoristico per effettuare misure di livelli sonori.

# Appendice A

## Richiami teorici

### A.1 Propagazione di onde sonore

Un'onda sonora<sup>1</sup>, come risaputo, è una perturbazione della pressione atmosferica avente la proprietà di propagarsi; affinché questo possa avvenire occorrono due elementi indispensabili: una causa ed un mezzo. Il primo potrà essere un sistema meccanico oscillante con sufficiente frequenza<sup>2</sup>, mentre il secondo è, in generale, un fluido, di norma l'aria.

La descrizione fisica della propagazione delle onde suddette avviene tramite tre leggi fondamentali:

**L'equazione di Eulero.** Siano  $p$  la pressione sonora,  $\mathbf{u}$  il vettore velocità istantanea di una particella di fluido e  $\rho_0$  la densità dell'aria in condizioni normali, allora si ha:

$$\nabla p = -\rho_0 \frac{\partial \mathbf{u}}{\partial t}. \quad (\text{A.1})$$

---

<sup>1</sup> Vedi [10], capitolo 3.

<sup>2</sup> Un'onda di pressione perché venga percepita dall'apparato uditivo umano deve avere frequenza compresa tra 20 Hz e 20 000 Hz circa, a seconda dell'individuo.

La relazione A.1 altro non è che l'equivalente fluidodinamico della seconda legge newtoniana della dinamica, da cui deriva, ed esprime in sostanza le condizioni di equilibrio dinamico delle forze agenti su un elemento di fluido.

**L'equazione di continuità.** Se si esprime con  $\delta$  la variazione relativa di densità, o *condensazione*<sup>3</sup>, del mezzo e con  $\mathbf{s}$  il vettore spostamento, per un generico elemento di fluido vale la legge

$$\delta = -\nabla \cdot \mathbf{s}, \quad (\text{A.2})$$

che sintetizza il principio secondo cui il flusso netto di massa, ossia la differenza tra quello entrante e quello uscente, deve essere uguagliato dalla variazione di massa di fluido contenuta dall'elemento.

**L'equazione di stato termodinamico.** Ipotizzando ragionevolmente variazioni di stato talmente rapide da poter essere ritenute adiabatiche ed indicando con  $p_0$  il valore della pressione sonora a riposo e con  $\gamma$  il rapporto tra i calori specifici del gas, rispettivamente, a pressione e volume costante, il comportamento termodinamico del fluido in questione è dato da

$$\frac{1}{p_0} \frac{\partial p}{\partial t} = \gamma \frac{\partial \delta}{\partial t}. \quad (\text{A.3})$$

Dalle relazioni A.1, A.2 e A.3, dopo qualche passaggio, si ricava l'*equazione di propagazione delle onde sonore*:

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}, \quad (\text{A.4})$$

---

<sup>3</sup>È data da

$$\delta = \frac{\rho - \rho_0}{\rho}$$

dove  $\rho$  è densità dell'elemento di fluido considerato.

in cui  $c$  è la velocità di propagazione dell'onda sonora nel gas, definita come

$$c = \sqrt{\gamma \frac{p_0}{\rho_0}}. \quad (\text{A.5})$$

## A.2 Intensità, densità di energia e potenza

Di seguito si riportano le definizioni di alcune grandezze fondamentali<sup>4</sup>:

**Intensità sonora.** È la quantità di energia che fluisce, nell'unità di tempo, attraverso una superficie di area unitaria perpendicolare alla direzione di propagazione dell'onda. L'*intensità istantanea* è data da

$$I = p_m u_m \cos^2 \left[ \omega \left( t - \frac{x}{c} \right) \right], \quad (\text{A.6})$$

dove  $p_m$  e  $u_m$  sono i valori di picco di pressione e velocità relative alla direzione di propagazione considerata, mentre l'*intensità media* dalla relazione

$$\bar{I} = \frac{p_{\text{rms}}}{\rho_0 c} \quad \left[ \frac{\text{W}}{\text{m}^2} \right]. \quad (\text{A.7})$$

La quasi totalità delle sorgenti reali, però, non irradia energia uniformemente in tutte le direzioni, tale caratteristica viene misurata da una grandezza denominata *direttività* ed indicata con  $Q$ ; si ha

$$Q = \frac{I_{\theta,\phi}}{\bar{I}}, \quad (\text{A.8})$$

dove  $I_{\theta,\phi}$  è l'intensità irradiata nella direzione specificata, in coordinate sferiche, dagli angoli  $\theta$  e  $\phi$ .

---

<sup>4</sup> Per i dettagli si rimanda sempre a [10].

**Densità di energia sonora.** È l'energia contenute in un'unità di volume del mezzo ed è data da

$$D = \frac{\bar{I}}{c^2} = \frac{p_{\text{rms}}}{\rho_0 c^2} \quad \left[ \frac{\text{W} \cdot \text{s}}{\text{m}^3} \right]. \quad (\text{A.9})$$

**Potenza sonora.** Supponendo che tutta l'energia sonora fluente nel mezzo sia prodotta da una sola e ben identificabile sorgente, l'energia irradiata nell'unità di tempo da quest'ultima rappresenta la sua potenza sonora  $W$ . Se  $S$  è una superficie immaginaria inviluppante la sorgente ed  $I_s$  l'intensità sonora rilevata su un area elementare  $dS$  della superficie, risulta

$$W = \int_S I_s dS. \quad (\text{A.10})$$

### A.3 Livelli sonori

A causa dell'enorme capacità dinamica dell'orecchio umano difficilmente viene adottata la scala lineare per rappresentare le grandezze acustiche: dovendo trattare con un campo di valori estremamente esteso e disperso<sup>5</sup> è universalmente preferita la scala dei deciBel, codificata nella norma ISO 1648. Premettendo che un valore espresso in deciBel implica *sempre* un rapporto con un altro valore di riferimento, di seguito si riportano le principali grandezze impiegate:

**Livello di pressione sonora.**

$$\begin{aligned} L_p &= 10 \log \frac{p^2}{p_0^2} = \\ &= 20 \log \frac{p}{p_0} \quad [\text{dB}], \end{aligned} \quad (\text{A.11})$$

dove  $p_0 = 20 \mu\text{Pa}$ . La A.11 si può anche riscrivere nel seguente modo, dopo aver sostituito il valore di  $p_0$ :

---

<sup>5</sup> La minima pressione percepibile da un individuo normo udente è di  $20 \mu\text{Pa}$ , la soglia del dolore è, invece, attorno ai  $60 \text{ Pa}$ .

$$L_p = 10 \log p^2 + 94 \quad [\text{dB}]. \quad (\text{A.12})$$

**Livello di potenza sonora.**

$$L_W = 10 \log \frac{W}{W_0} \quad [\text{dB}], \quad (\text{A.13})$$

dove  $W_0 = 1 \text{ pW}$ . Sostituendo quest'ultimo valore nella A.13 si ricava

$$L_W = 10 \log W + 120 \quad [\text{dB}]. \quad (\text{A.14})$$

**Livello di intensità sonora.**

$$L_I = 10 \log \frac{I}{I_0} \quad [\text{dB}], \quad (\text{A.15})$$

dove  $I_0 = 1 \text{ pW/m}^2$ . È largamente in uso esprimere anche la direttività in decibel: in questo caso il parametro definito dall'equazione A.8 prende il nome di *Directivity Index* ed è così definito

$$\begin{aligned} DI &= 10 \log Q = \\ &= L_{I_{\theta,\phi}} - L_{\bar{I}}. \end{aligned} \quad (\text{A.16})$$

Se  $\rho_0 c$  vale 400 rayl risulteranno uguali  $\rho_0 c I_0$  e  $p_0^2$ , cioè sarà verificata la relazione

$$L_I = L_p. \quad (\text{A.17})$$

Tale situazione si avrebbe a pressione atmosferica e temperatura di 39°C; alla temperatura di 20°C il valore dell'impedenza acustica sale a 415 rayl, tuttavia l'errore commesso applicando ugualmente la A.17 è comunemente ritenuto trascurabile.

Se l'intensità sonora  $I$  è uniforme su una superficie  $S$  che circoscrive una sorgente di potenza  $W$  si può scrivere:

$$10 \log \frac{W}{W_0} = 10 \log \frac{I}{I_0} + 10 \log \frac{S}{S_0} \quad [\text{dB}]; \quad (\text{A.18})$$

se, poi,  $S_0 = S = 1 \text{ m}^2$ , risulta esattamente

$$L_W = L_I. \quad (\text{A.19})$$

## A.4 Curve di ponderazione

La sensibilità dell'orecchio umano non è costante in tutta la banda udibile, ma presenta un massimo nella regione attorno a 4 kHz ed ha i valori minimi in corrispondenza degli estremi, come evidenziato dalle curve isofoniche elaborate da Fletcher e Munson.<sup>6</sup> Questo ha portato all'ideazione delle unità di misura *percettive* (la scala dei *phon*) ed ha suggerito l'adozione negli strumenti di misura di un filtraggio dei livelli rilevati in modo da renderli coerenti alla reale sensibilità umana. Il suddetto filtraggio è codificato nelle cosiddette *curve di ponderazione*, le quali non fanno altro che indicare l'alterazione della risposta in frequenza dello strumento, ossia il fattore con cui debbono essere pesate le frequenze della banda audio. In figura 2.6 sono riportate le curve di ponderazione A, B e C che sono, sostanzialmente, le curve isofoniche a 40, 70 e 100 phon rispettivamente.

## A.5 Trasformata di Fourier

In questa sede si vuole riportare solo la definizione operativa. Se  $x(t)$  è una funzione

---

<sup>6</sup> Si veda [9] al capitolo 2.

- limitata
- tale da presentare un numero finito di discontinuità
- assolutamente integrabile

allora la sua trasformata secondo Fourier è data dalla relazione

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt. \quad (\text{A.20})$$

La trasformata inversa del segnale  $X(f)$ , invece si può calcolare mediante l'equazione

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df. \quad (\text{A.21})$$

In generale affinchè esista la trasformata di Fourier del segnale  $x(t)$ , per il teorema di Plancherel è sufficiente che quest'ultimo abbia energia finita, caratteristica, questa, di tutti i segnali fisici.

# Bibliografia

- [1] M. BINELLI, A. VENTURI, A. AMENDOLA, E A. FARINA, *Experimental analysis of spatial properties of the sound field inside a car employing a spherical microphone array*, in 130<sup>th</sup> AES Convention, London (UK), mag. 2011.
- [2] S. CAMPANINI E A. FARINA, *A new Audacity feature: room objective acoustical parameters calculation module*, in Linux Audio Conference, Parma, Italy, April 2009.
- [3] A. FARINA, A. CAPRA, L. CHIESI, E L. SCOPECE, *A spherical microphone array for synthesizing virtual directive microphones in live broadcasting and in post production* , in AES 40<sup>th</sup> International Conference, Tokyo, Japan, october 2010.
- [4] O. KIRKEBY E P. A. N. J., *Digital Filter Design for Inversion Problems in Sound Reproduction* , AES, 47, No. 7/8 (1999).
- [5] O. KIRKEBY, P. A. NELSON, H. HAMADA, E F. ORDUNA-BUSTAMANTE, *Fast Deconvolution of Multichannel Systems Using Regularization* , in IEEE Transaction On Speech and Audio Processing, vol. VOL. 6, no. 2, March 1998.
- [6] D. T. LEE E B. J. SCHACHTER, *Two Algorithms for Constructing a Delaunay Triangulation* , International Journal of Computer and Information Sciences, 9 (1980).

- [7] S. G. NORCROSS, G. A. SOULODRE, E M. C. LAVOIE, *Evaluation of Inverse Filtering Techniques for Room/Speaker Equalization*, in 113<sup>th</sup> AES Convention, Los Angels, CA, USA, October 2002.
- [8] J. R. PIERCE, *La scienza del suono*, Zanichelli, 1988.
- [9] R. SPAGNOLO (A CURA DI), *Manuale di acustica applicata*, UTET, Torino, Italy, 3 ed., gen. 2005. ISBN 88-7750-710-1.
- [10] S. CINGOLANI E R. SPAGNOLO (A CURA DI), *Acustica musicale e architettonica*, UTET, Torino, Italy, 1 ed., dic. 2005. ISBN 88-7750-941-4.
- [11] J. R. SHEWCHUK, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*, in Applied Computational Geometry: Towards Geometric Engineering, M. C. Lin e D. Manocha, cur., vol. 1148 di *Lecture Notes in Computer Science*, Springer-Verlag, mag. 1996, pp. 203–222. From the First ACM Workshop on Applied Computational Geometry.

*Vorrei cogliere l'occasione fornитами in queste righe per ringraziare innanzitutto il Professor Angelo Farina, per avermi fatto così appassionare alla sua materia, nonostante gli toccasse l'ultima lezione del venerdì pomeriggio.. da 4 ore! Sono state le lezioni più belle che ho seguito durante la mia lunga (fin troppo) carriera scolastica.*

*Vorrei ringraziare anche i suoi collaboratori, Andrea Capra, Andrea Venturi, Luca Novarini, Enrico Armelloni, Christian Varani, Lorenzo Chiesi, per avermi sopportato lungamente e per essere tutti quanti persone così piacevoli e di cui ho grande stima.*

*Un grazie particolare va a Simone Campanini per il supporto infinito, la pazienza e la presenza, soprattutto nell'ultimo periodo quando 'il gioco s'è fatto duro'. Non mi ha negato il suo aiuto nemmeno il giorno del suo compleanno! Instancabile!*

*Inoltre vorrei ringraziare tutti gli amici che mi sono stati vicino durante questi anni all'università, primo fra tutti il mio grande amico l'ingegner Bolo, che mi ha seguito e tenuto d'occhio fin dai tempi dell'asilo portandomi fino alla laurea. Incrollabile! Grazie anche a Giorgio e Buro, Mich e Dave (GB) per le grandi e preziose risate durante le giornate tra le mura universitarie, i viaggi in macchina giornalieri, i pranzi ing, la impagabile amicizia. Grazie anche alla prof. di geometria del primo anno perché ci fa ridere ancora oggi.*

*Un sentito grazie alla mia famiglia che mi ha sempre sostenuto al massimo delle proprie possibilità senza risparmiarsi mai e con una fiducia incrollabile, nonostante qualche lamentela saltuaria. Grazie alla Ceci che so che è quella che ha fatto più fatica ma non me l'ha fatto pesare troppo nonostante tutto.*

*Ringrazio i miei quattro Nonni, i miei trentuno zii e i miei quarantaquattro cugini per essere stati la mia casa e il mio habitat e senza dei quali nulla sarebbe com'è.*

*Ringrazio in ordine sparso Michael Jackson, Al Jarreau, Pino Daniele, Marcus Miller, i Quintorigo, Bobby McFerrin, Stevie Wonder, Elio e Le Storie Tese, Ray Charles, Stefano Bollani, i Waterjets , Blues Brothers, Mario Biondi, De André, i TOTO, Dave Matthews Band, i Mr. Big, Muse, Chick Corea, Sergio Caputo, Maurizio Parascandolo, Sting, Mike Lemus, i System of a Down, FaberNoster, i Tenacious D, Dumbo Gets Mad, i Tower of Power.. qualcosa degli*

*Oliver Onions.. per essermi stati sempre nelle orecchie e, come dice Campanini,  
'per avermi tenuto vivo'.*

*Un grazie speciale a Sara; a lei è toccato di raccogliermi come un gatto e trascinarmi fino alla laurea.*

*Grazie.*

*Grazie anche alle tante persone che non ho nominato (ci vorrebbe un'altra tesi!)  
ma che porto nel cuore.*

*Un ultimo grazie ad Africa, sempre per ultima, piccola e nera ma sempre una  
sicurezza.*

Typesetted on 2013, September 27 with L<sup>A</sup>T<sub>E</sub>X 2 <sub>$\varepsilon$</sub>