

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

1 Premessa	1
2 Elementi di acustica	4
2.1 Introduzione	4
2.2 Grandezze fondamentali	4
2.2.1 Frequenza e Periodo	5
2.2.2 Velocità di propagazione e lunghezza d'onda	6
2.3 Onde acustiche	7
2.3.1 La scala dei deciBell	7
2.3.2 Livello di pressione	9
2.3.3 Livello equivalente	9
2.4 Sistema uditivo umano	10
2.4.1 Effetti di non linearità dell'orecchio umano	11
2.4.2 Bande frequenziali	13
2.4.3 Filtri di ponderazione	15
3 Array microfonici	16
3.1 Array sferico: Eigenmike	16
3.2 Array cilindrico	16
3.3 Array planare	16
4 Sintesi di microfoni virtuali	17
4.1 Elaborazione dei segnali microfonici	17
4.2 Sistema di Elaborazione	17
4.3 Implementazione del calcolo convolutivo	17

5 Ambiente di Sviluppo: l'host Audacity	18
5.1 Libreria per la costruzione di interfacce grafiche <code>wxWidgets</code>	20
5.2 Altre librerie	21
5.3 Libreria per elaborazione audio-video: <code>FFmpeg</code>	22
6 Descrizione plug-in: un manuale d'uso	24
6.1 Selezione dell'intervallo temporale da analizzare	24
6.2 Finestra di configurazione	26
6.3 Interfaccia principale	28
6.4 Esportazione dei risultati	30
7 Dettagli di Implementazione	31
7.1 Acquisizione dei dati in ingresso	32
7.2 Organizzazione del carico computazionale	33
7.3 Livello di fondo scala	34
7.4 <i>Mirroring</i> dei microfoni virtuali	35
7.5 <i>Meshing</i> della superficie mediante l'operazione di triangolazione di <i>Delaunay</i>	35
7.6 Interpolazione dei valori acquisiti e filtraggio in bande d'ottava .	38
7.7 Mappatura con scale cromatiche e <i>auto-scaling</i>	40
7.8 Esportazione dei risultati	41
8 Conclusioni	43
A Appendice	44
A.1 Propagazione di onde sonore	44
A.2 Intensità, densità di energia e potenza	46
A.3 Livelli sonori	47
A.4 Curve di ponderazione	49
A.5 Trasformata di Fourier	49

Elenco delle figure

1.1	Esempio di mappa acustica generata dal plug-in <i>Microphone Array Analyzer</i>	2
2.1	Moto delle particelle di un fluido elastico compresse e decomprese alternativamente da un pistone, creando l' <i>onda sonora</i>	5
2.2	Livello sonoro di una sorgente intermittente e relativo livello equivalente L_{EQ}	10
2.3	Apparato acustico umano.	11
2.4	Risposta non lineare della coclea	12
2.5	Grafico delle curve isofoniche	12
2.6	Grafico delle curve di ponderazione	15
5.1	Interfaccia utente dell'host utilizzato, <i>Audacity®</i> sotto diverse piattaforme.	19
5.2	Intorno per la costruzione di interfacce grafiche <i>wxFormBuilder</i> . .	21
6.1	Selezione dello spezzone di materiale audio da analizzare con il <i>plug-in</i>	25
6.2	Messaggio di errore in caso di memoria insufficiente per le operazioni di <i>precalcolo</i>	26
6.3	Finestra di configurazione del modulo <i>Microphone Array Analyzer</i>	27
6.4	Finestra principale del modulo <i>Microphone Array Analyzer</i>	28
7.1	Rappresentazione schematica del processo di <i>mirroring</i> , necessario a garantire la continuità ai bordi della mappa.	36

7.2	Direzioni di puntamento delle capsule dell'Eigenmike sovrapposte ad una foto panoramica del Teatro alla Scala di Milano, svolta secondo lo schema di proiezione visto nella sezione 7.4.	36
7.3	Descrizione della proprietà dei triangoli formati con l'algoritmo di <i>Delaunay</i>	38
8.1	Graphic User Interface sotto le tre diverse piattaforme principali: <i>Windows</i> , <i>Mac OS X</i> e <i>Linux</i>	43

Elenco delle tabelle

2.1	Velocità del suono in alcuni mezzi (fonte:).	6
2.2	Valori di pressione e di SPL relativi a alcuni suoni comuni.	8
2.3	Definizione delle diverse bande e relative frequenze di centro banda per le due diverse analisi descritte.	14

Capitolo 1

Premessa

Il presente progetto di tesi si pone l’obiettivo 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 videoclip composto dalla sovrapposizione di:

un videoclip di sfondo ottenuto da una particolare videocamera, che rappresenti 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®*¹, celebre software *open-source* molto versatile, dalle possibilità molto ampie per permettere gli utilizzi più disparati. Il software sviluppato in questa tesi, dialogando con l’*host Audacity®*, acquisisce i dati in ingresso e genera da essi una *mappa acustica*, composta da bande di colore che corrispondono ai diversi livelli sonori attorno all’array sonda, posto nell’ambiente di misura.

Questo tipo di *tool* può avere svariate applicazioni tecnologiche in quanto è in

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

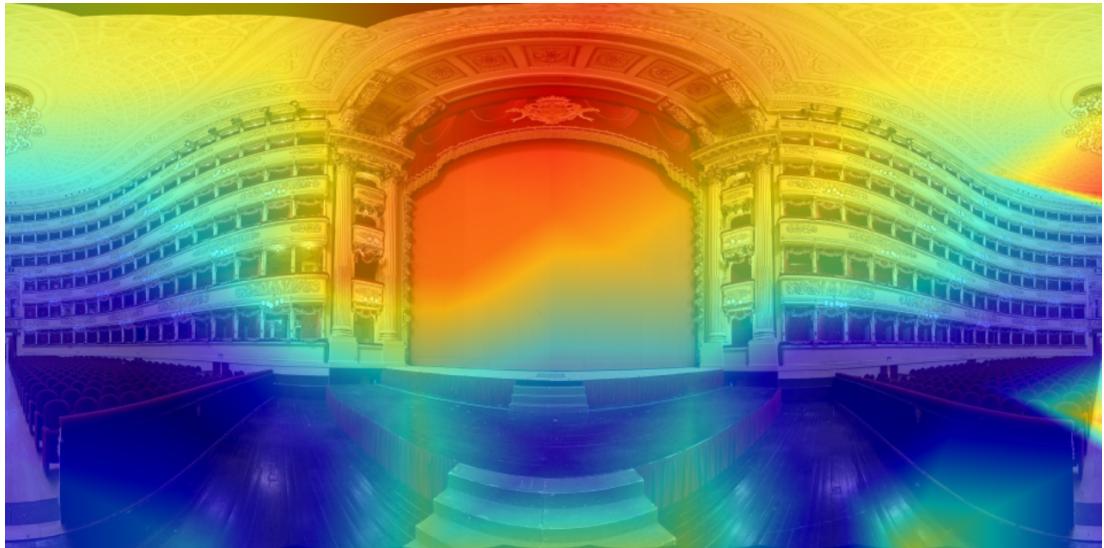


Figura 1.1: Esempio di mappa acustica generata dal plug-in *Microphone Array Analyzer*

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 varie 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 da me svolta durante 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 sulla acustica di base e riguardo alcune proprietà fisiche del suono, ausilio fondamentale a tutta la trattazione successiva.

Capitolo 3 introduzione agli array microfonici come tecnica di misura dei campi acustici e delle loro proprietà spaziali di direttività e direzionalità.

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®*) 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 come rappresentato in Figura 2.1. Le particelle del fluido adiacenti alla superficie del pistone saranno spostate dal suo movimento verso le particelle subito adiacenti a esse e si verificherà quindi una compressione locale. Subito dopo il pistone retrocederà lasciando un spazio libero di sé che

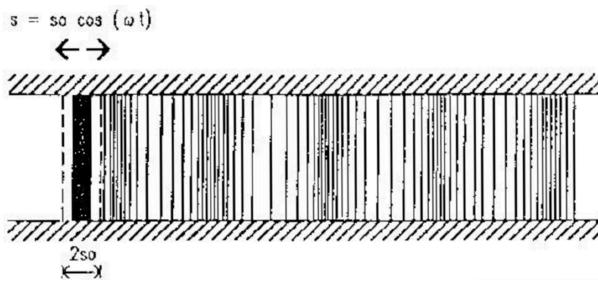


Figura 2.1: Moto delle particelle di un fluido elastico compresse e decomprese alternativamente da un pistone, creando l'*onda sonora*.

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 adiacente 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 frequen-

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

Tabella 2.1: Velocità del suono in alcuni mezzi (fonte:).

za e le relazioni tra le varie componenti frequenziali presenti in un suono. Queste componenti vengono dette *armoniche*. Se le armoniche sono 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.

Tenendo come riferimento sempre l'esempio del pistone oscillante di Figura 2.1, definiamo come *lunghezza d'onda λ* la distanza tra due picchi di compressione del mezzo elastico.

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

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

o alternativamente sfruttando la definizione ??:

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

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 deciBell

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:

$$p_0 = 20 \mu Pa \quad (2.4)$$

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

Sono stati riportati nella tabella 2.2 alcuni valori di pressione in pascal relativi a suoni comuni.

<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.

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 umano¹, viene introdotta la scala logaritmica dei *decibell*. 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:

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

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. Nel momento in cui essa viene applicata alla

¹Vedi sezione 2.4.1

variazione di pressione di un suono si parla di *Livello di Pressione* (o *Sound Pressure Level SPL*).

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):

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

o più comunemente

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

dove p_0 sia il valore di scostamento di pressione usato come riferimento riferimento, scelto dal sistema internazionale uguale a $20 \mu Pa$ che come abbiamo visto poco sopra nell'equazione 2.4 corrisponde alla soglia di udibilità a $1000 Hz$ nell'atmosfera in condizioni standard².

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:

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

²Per una spiegazione più dettagliata sui livelli sonori si veda l'appendice A.3

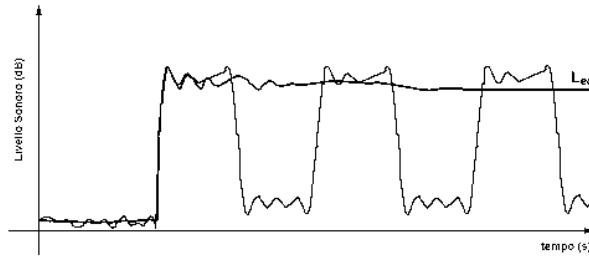


Figura 2.2: Livello sonoro di una sorgente intermittente e relativo livello equivalente L_{EQ} .

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. 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. Infatti il livello equivalente è usato nella legislazione come riferimento principale sul quale normare il rumore. 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 normare 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 dalle svariate funzioni, ognuno dei quali influenza sulla percezione acustica in maniera rilevante e addirittura alcuni degli effetti

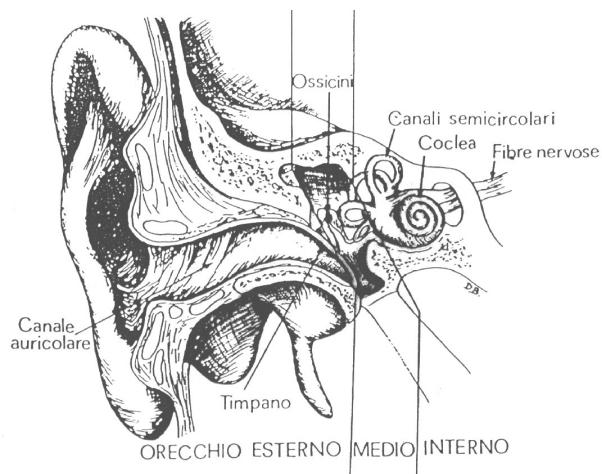


Figura 2.3: Apparato acustico umano.

legati alla sensazione uditiva, che vengono chiamati *psicoacustici*, riguardano la sola interpretazione, da parte del cervello, dei segnali elettrochimici provenienti dall'apparato uditivo.

In questa sede ci interessano solamente alcuni effetti di non linearità dell'apparato uditivo, le quali 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; è lei la responsabile di gran parte degli effetti non lineari di cui discuteremo in seguito.

Sezionando la coclea, 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, alla maniera, ad esempio, di una corda o di una frusta. Si osservi nel grafico di figura 2.4 come le basse frequenze interessino la parte terminale mentre le alte frequenze la parte iniziale. Si evince facilmente inoltre che due rumori con bande sovrapposte (in tutto o in parte) si mascherino

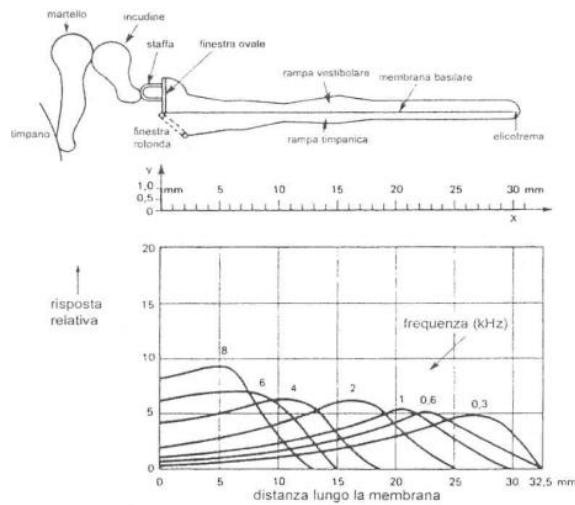


Figura 2.4: Risposta non lineare della cochllea

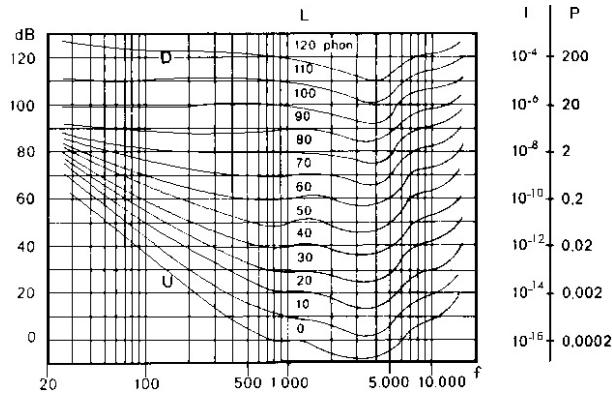


Figura 2.5: Grafico delle curve isofoniche

in modo tale che il segnale di maggiore intensità copra il segnale più debole, a meno che quest'ultimo non abbia una larghezza di banda sufficientemente larga.

Per il sopraccitato 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 Hz) ed a quelle elevate (oltre i $15kHz$). 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 diverso³. Tutti questi effetti sono osservabili nel grafico in Figura 2.5.

³Si veda [3] al capitolo 2.

Essendo dunque tutto il sistema uditivo umano molto soggetto, nella sua risposta all'eccitamento acustico, alla frequenza dell'onda incidente, è spesso comodo analizzare i fenomeni acustici 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 di frequenze, le *bande di frequenza*. Ciascuna banda è caratterizzata da una frequenza di taglio superiore f_s e da una frequenza di taglio inferiore f_i . L'analisi in frequenza può essere di due tipi:

analisi a banda costante: se $\Delta f = f_s - f_i = \text{costante}$ cioè suddivisione a bande frequenziali di uguale larghezza. Tipicamente impiegata per analisi approfondite della composizione in frequenza. Solitamente viene usata per misure nel campo delle vibrazioni delle strutture o delle macchine. Viene ottenuta con una tecnica di elaborazione matematica detta FFT (Fast Fourier Transform)⁴

analisi a banda percentuale costante da 1/1 o 1/3 di ottava: se la larghezza di banda Δf è una percentuale costante del valore della frequenza nominale di centro-banda che caratterizza la banda stessa.

Si parla di *ottave* (bande da 1/1 di ottava) nel caso di $f_s = 2 \cdot f_i$.

Si parla invece di *terzi d'ottava* (bande da 1/3 d'ottava) se $f_s = 2^{1/3} f_i$.

Tipicamente impiegata per misure acustiche. Possono essere usati “banchi” di 10 filtri (ottave) o 30 filtri (terzi), ottenuti con circuiti analogici o digitali (filtri IIR)

Nella Tabella 2.3 sono definite le diverse bande e relative frequenze di centro-banda per le due diverse analisi descritte.

<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>
11	16	12	14,1	16	17,8
			17,8	20	22,4
			22,4	25	28,2
22	31,5	44	28,2	31,5	35,5
<i>tabella</i>	<i>trop</i> po	<i>lunga</i>	<i>come</i>	<i>si</i>	<i>risolve?</i>
			35,5	40	44,7
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8
11	16	12	14,1	16	17,8
			17,8	20	22,4
			14,1	16	17,8

Tabella 2.3: Definizione delle diverse bande e relative frequenze di centro-banda per le due diverse analisi descritte.

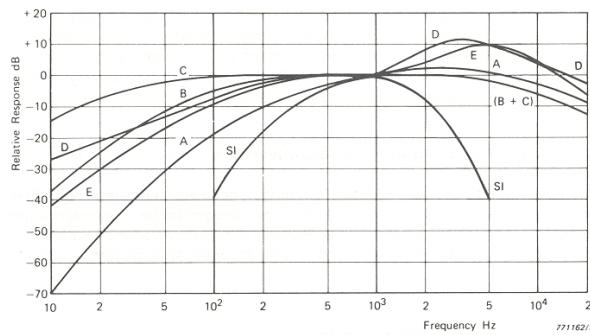


Figura 2.6: Grafico delle curve di ponderazione

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 importanti⁵:

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 $dB(C)$.

⁴si veda il capitolo ??

⁵Per un approfondimento sulle curve di ponderazione si veda l'appendice A.4

Capitolo 3

Array microfonici

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

Sferico: array microfonico costituito da 32 capsule omnidirezionali poste uniformemente su una sfera

Cilindrico: array microfonico

Planare: array microfonico

3.1 Array sferico: Eigenmike

3.2 Array cilindrico

3.3 Array planare

Capitolo 4

Sintesi di microfoni virtuali

[...]

4.1 Elaborazione dei segnali microfonici

[...]

4.2 Sistema di Elaborazione

[...]

4.3 Implementazione del calcolo convolutivo

[...]

Capitolo 5

Ambiente di Sviluppo: l'host Audacity

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 invece di sviluppare un *plug-in* che si installasse quindi su un sistema pre-esistente, in grado di svolgere tutte le funzioni di base di recording e editing audio, nonchè 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. Principalmente si era discusso sulla possibilità di sviluppare un plug-in utilizzando la SDK della Steinberg®, famosa casa di sviluppo software audio, per via della universale diffusione e implementazione su quasi tutti i tipi di host su ogni piattaforma esistente. Questo tipo di software prendono il nome di **VST**. Nonostante i grandi vantaggi di portabilità e facilità di utilizzo e di reperibilità di documentazione e supporto, non è stato possibile sviluppare il software con questa SDK. Il problema fondamentale che si è riscontrato era l'impossibilità

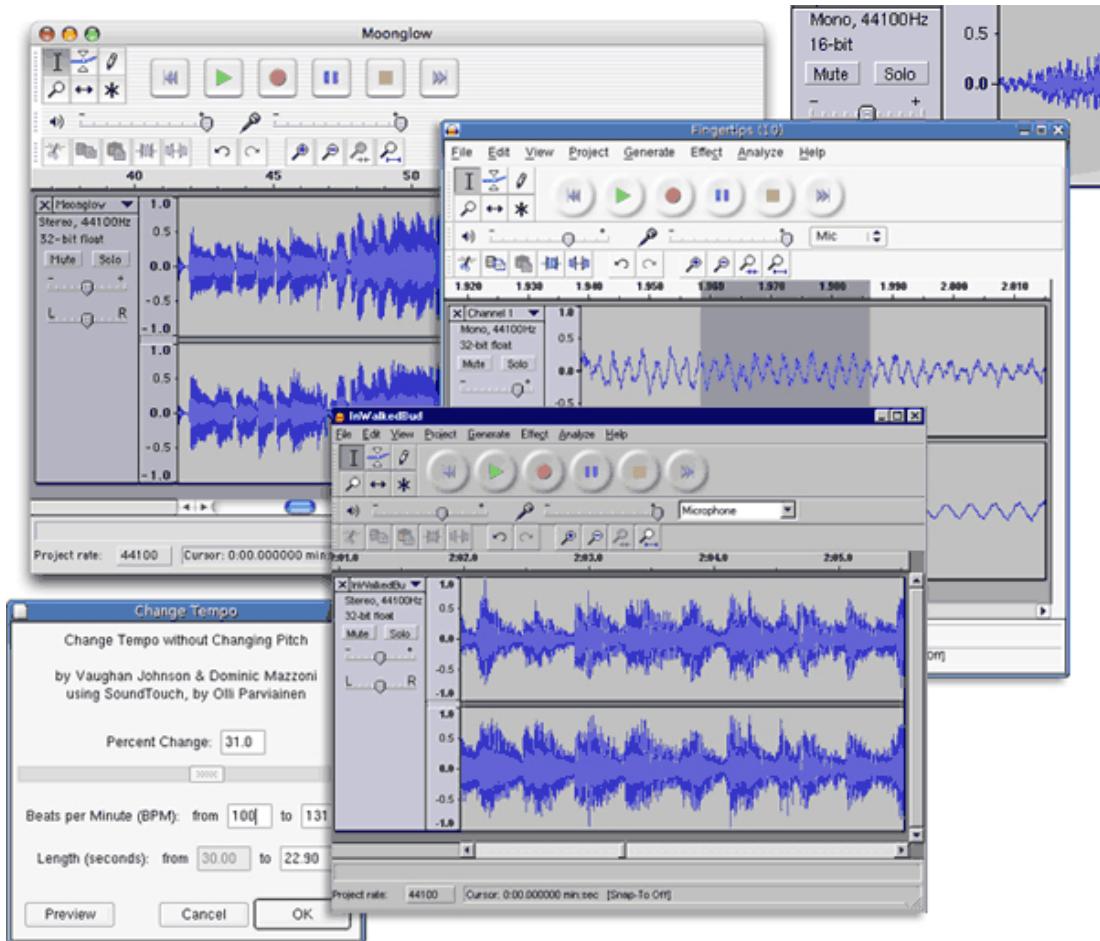


Figura 5.1: Interfaccia utente dell'host utilizzato, *Audacity®* sotto diverse piattaforme.

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 cercano sempre di lavorare a blocchetti di 256 campioni alla volta in modo da simulare il più possibile un effetto *real-time*.

Si è optato quindi per la piattaforma più permissiva, comoda e personalizzabile (in quanto progetto *open source*), *Audacity®*. 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 a arrivare alle preferenze che l'utente sta utilizzando. Inoltre si tratta di un programma nativamente multiplataforma, come richiesto, e a migliorarne ulteriormente le capacità aggiungiamo che si tratta di uno dei pochissimi programmi che gestiscono registrazioni

a N canali a piacimento e che implementano il formato audio *W64*. L'utilizzo di questo formato è a dir poco fondamentale in quanto, registrando eventi anche lunghi più di un'ora, utilizzando decine di canali, attualmente è fortemente sconsigliato l'utilizzo di formati con **header** sotto i 64 bit. *Audacity®* 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 piattaforme 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 per via della grande maturità del progetto e per la capacità di conferire alla propria GUI il look 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 in maniera nativamente multipiattaforma 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 sorta di *middle-ware* e ponendo il livello di programmazione al di sopra dei dettagli di implementazione. Per notare il livello di maturità del progetto basti pensare che vengono fornite funzionalità di *multithreading*, salvataaggio e caricamento di immagini in una varietà di formati diffusamente utilizzati, 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®* 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 nuovo motivo valido per scegliere di utilizzare proprio **wxWidget**.

Per la creazione di interfacce grafiche basate su **wxWidget**, si è individuato

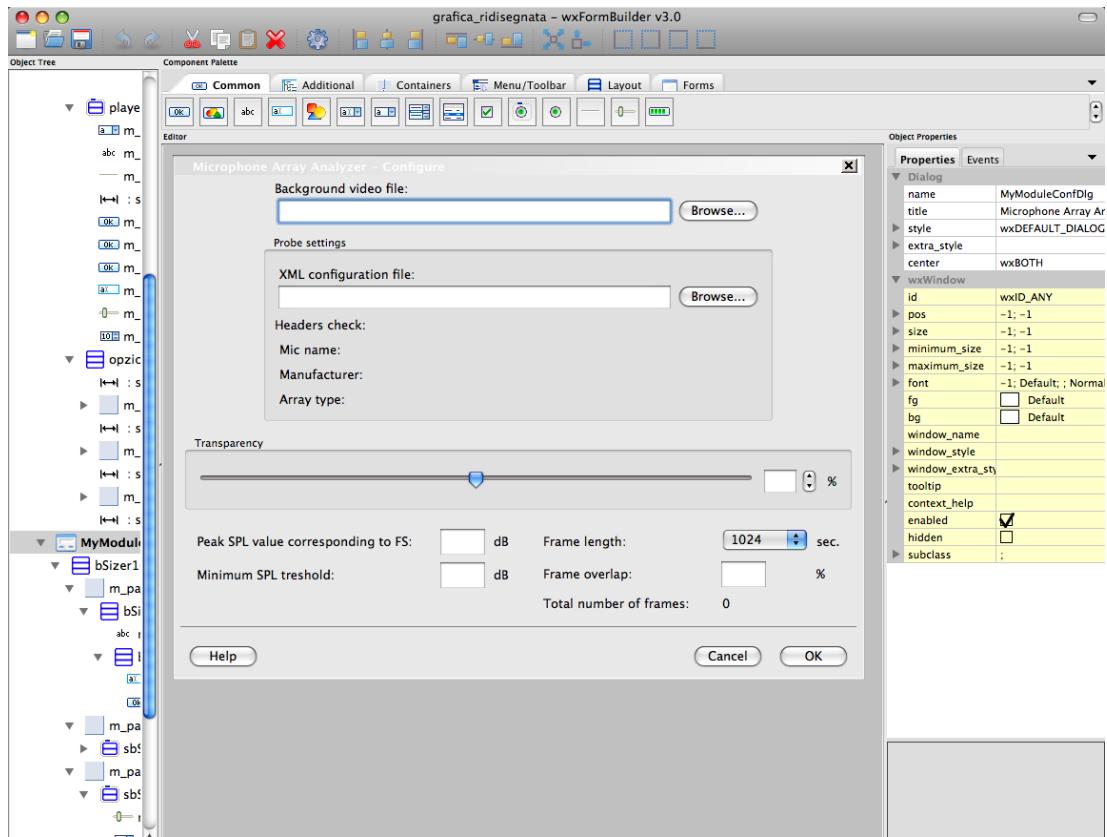


Figura 5.2: Intorno per la costruzione di interfacce grafiche wxFormBuilder

un programma *helper* di tipo *WYSIWYG*¹, di nome *wxFormBuilder*, sicuramente di grande utilità e immediatezza nonchè 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. Per chi volesse approfondire sono stati allegati a piè pagina i relativi link web dove trovare informazioni supplementari

¹(*What You See Is What You Get*)

FFT algoritmo² massicciamente utilizzato in acustica come sempre in materia di elaborazione di segnali. In questo caso si è utilizzata una implementazione propria dello stesso codice sorgente dell'host *Audacity®*.

Triangle++ wrapper C++ per la libreria Triangle che implementa il famoso algoritmo della *triangolazione di Delaunay* utile a partizionare una superficie con sotto-superfici di area triangolare³.

libSndFile libreria gratuita per la lettura-scrittura e elaborazione di file audio⁴.

TinyXML libreria di ausilio alla lettura e scrittura su file xml per scambio di dati tra programmi⁵. 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.

5.3 Libreria per elaborazione audio-video: **FFmpeg**

Essendo lo scopo della tesi in oggetto, lo sviluppo di una mappa acustica dinamica ottenuta dalla sovrapposizione di due video generati dall'elaborazione audio, è facile intendere quanto sia 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, ancora una volta, nel bundle di *Audacity®*: **FFmpeg**⁶. Si tratta di una soluzione completa e, ancora una volta, *cross-platform*, per registrare convertire e elaborare *stream* audio e video.

A Sua volta, 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

²<https://code.google.com/p/audacity/source/browse/audacity-src/trunk/src/FFT.cpp>

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

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

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

⁶<http://www.ffmpeg.org/about.html>

molte *core multimedia utilities*, **libswresample** la quale è in grado di effettuare operazioni di **resampling** audio e video nonchè 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 nonchè per esportare i risultati nella forma di frame singoli oppure di un video intero. Di questo argomento si discuterà meglio in seguito ⁷.

⁷si confronti la sezione ??

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 possibilità che l'utente si trova a sfruttare 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 ora.

6.1 Selezione dell'intervallo temporale da analizzare

Prima ancora 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

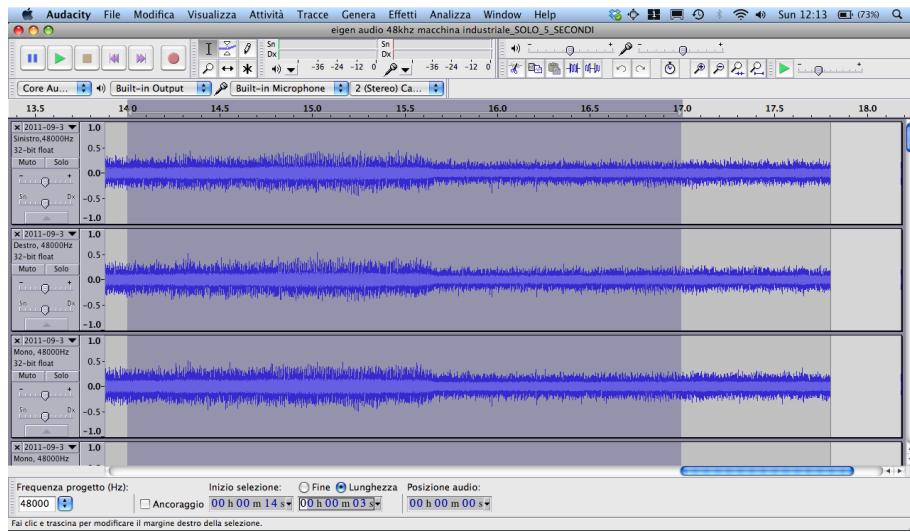


Figura 6.1: Selezione dello spezzone di materiale audio da analizzare con il *plug-in*.

essere necessario selezionare solo una parte, in modo da focalizzarsi nell'analisi su un particolare istante, all'occorrenza anche molto breve. Si pensi per esempio all'analisi delle prime riflessioni in una sala da concerto; analisi nella quale si è interessati a studiare il comportamento di un ambiente sotto l'aspetto degli echi e altri effetti che possono essere fastidiosi per alcuni utilizzi dell'ambiente stesso.

A questo scopo è necessario selezionare la porzione di audio di interesse, come indicato in figura 6.1. Questa è la procedura con la quale *Audacity®* gestisce il materiale audio da editare attraverso i *plug-ins* (che nel menù del programma vengono denominati *effects*).

In alternativa è possibile selezionare anche l'intera registrazione, consapevoli che essendo l'analisi compiuta nel *precalcolo* computazionalmente molto onerosa, nonché esigente per quanto riguarda l'utilizzo delle *risorse*, si va incontro alla possibilità che il *precalcolo* si interrompa prima di aver analizzato tutti i frame, debitamente all'esaurimento delle risorse a disposizione. In tal caso verrà presentato un messaggio di avviso, come da figura 6.2, e il programma passerà alla visualizzazione dei soli risultati analizzati. Si tenga presente, a titolo di riferimento, che su un normale laptop con processore *dual core* da 2.0 GHz con 2 Gb di *RAM*, l'elaborazione più lunga complessivamente effettuabile consta di circa 60 frames e impiega 2 min per il *precalcolo*.

Sarà sempre possibile costruire un video a partire dai singoli frames esportati

6.2. Finestra di configurazione

26

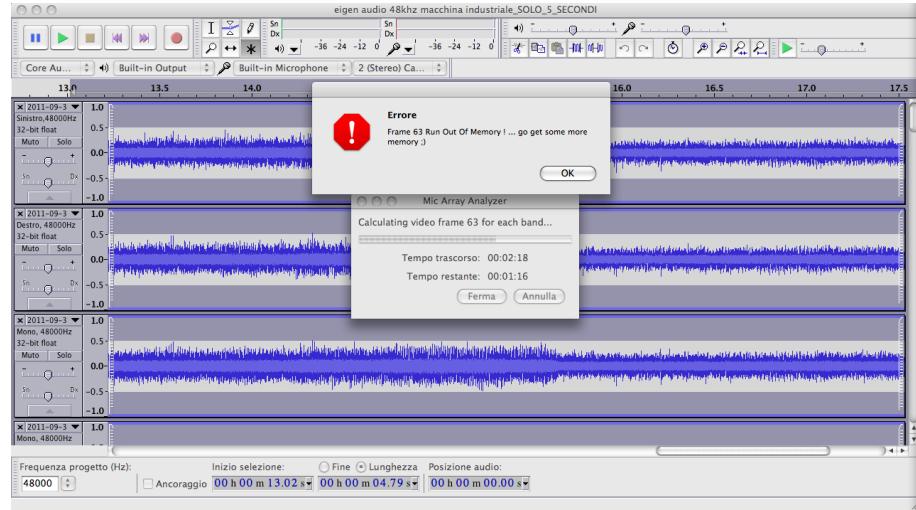


Figura 6.2: Messaggio di errore in caso di memoria insufficiente per le operazioni di *precalcolo*

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®* 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 atta all'inserimento di alcuni parametri per l'analisi del materiale audio precedentemente selezionato.

Attraverso questa finestra l'utente dovrà specificare:

Un videoclip dell'ambiente di background acquisito, con una videocamera adatta allo scopo¹, durante la registrazione dell'evento sonoro, su cui mappare i livelli *SPL* con le bande di colore.

Un file *xml* contenente la mappatura in *pixel* delle posizioni delle singole capsule microfoniche sulla immagine di background, nonché il *filepath* della matrice di filtri matematici (FIR²) utilizzati per sintetizzare tali microfoni

¹per una descrizione delle telecamere nei singoli array si ritorni al capitolo 3

²vedi appendice ??

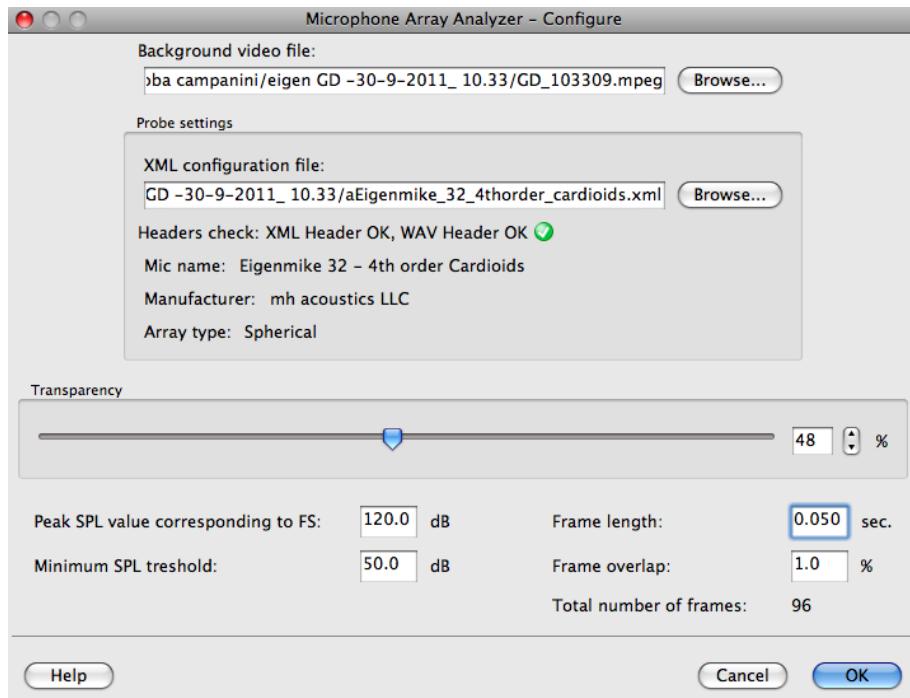


Figura 6.3: Finestra di configurazione del modulo *Microphone Array Analyzer*

nella maniera descritta al capitolo 5.3 e la *frame rate* caratteristica della matrice di filtri, che deve essere la stessa del materiale audio importata nel progetto *Audacity®*. 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.

i livelli di FS e min SPL cioè i limiti della scala di *dB* che si vuole analizzare.

I pixel eventualmente presenti nella *mappacolor-map* con valori SPL che eccedessero queste soglie, verrebbero disegnati completamente trasparenti.

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

la 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 svolta a monte

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

di *spezzettamento* del materiale di analisi; operazione che introdurrebbe una certa *scattosità* del video di output.

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. Andiamo ora a descrivere una per una 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. Questo tipo di analisi è molto utile 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 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 mentre il video sta riproducendosi, 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), radici 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.

abilità/disabilità 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 voluto³.

disabilità/abilità 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 sarebbe utile, in un momento del video in cui il campo acustico evolve in maniera complicata, poter visionare solo alcuni frame, *spegnendo* il video di background, per poterlo poi riaccendere per la rimanente parte del video.

Oltre a questi pannelli di opzioni con cui personalizzare i risultati ottenuti, è presente nell'interfaccia grafica anche un ultimo pannello, il quale non permette di selezionare nulla e non accetta input di nessun genere, ma fornisce l'importante informazione riguardante la posizione del mouse, aggiornata ad ogni movimento

³si 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.

dello stesso sul display, e il livello sonoro corrispondente a quella determinata posizione.

Al termine delle operazioni, quando ci si ritenga soddisfatti del video ottenuto è possibile esportare i risultati tramite l'apposito bottone *Esporta*.

6.4 Esportazione dei risultati

[...]

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. Sintesi dei microfoni virtuali mediante convoluzione;
4. Scalatura delle ampiezze dei segnali risultanti in funzione del fondo scala specificato nella finestra di configurazione;
5. Mirroring dei microfoni virtuali, per garantire la continuità ai bordi della mappa;
6. Copertura dell'intera area della mappa con superfici di forma triangolare (*meshing*);
7. Interpolazione dei dati e filtraggio in bande d'ottava;
8. Applicazione della *color-map* in relazione all'opzione di *auto-scaling*;
9. Sincronizzazione audio-video.

Nei capitoli precedenti si è già analizzato lo step della sintesi dei microfoni virtuali mediante convoluzione (punto 2). Si procederà di seguito con la spiegazione in dettaglio degli altri passaggi.

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,
- Un **videoclip dell'ambiente di background**,
- Una **matrice di filtri** in grado di sintetizzare i microfoni virtuali a partire dalle capsule dell'array,
- 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 configurazione¹.

Per quanto riguarda la **registrazione multicanale** direttamente proveniente dall'array, è lo stesso host *Audacity®* 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*, forniti in fase di configurazione dall'utente, utilizziamo alcune *routines* basate sulle *API* fornite dalla suite *ffmpeg* per acquisire il video, 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.

¹ancora come già detto al capitolo 6.2

Per quanto riguarda la **matrice di filtri**, abbiamo visto² che viene fornita al plugin sotto forma di file .wav, che viene caricato in memoria *heap* durante la fase di *precalcolo*³ per poi venire *deallocated* al momento della presentazione dei risultati.

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 quale 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 *precalcolo* 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 concorreranno alla formazione del video finale, calcolando un singolo frame ciascuno. Per l'implementazione di 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**, in modo da non stravolgere le abitudini implementative del programmatore, 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 apposita 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

²al capitolo 5.3

³dove ne parli?

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 thread⁴ libererà la memoria inutilizzata e aggiungerà informazioni globali derivanti dall'analisi dell'intero video (come il massimo assoluto di fondoscala).

7.3 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 considerato dei segnali ottenuti dalla sintesi per convoluzione dei microfoni virtuali. Per imporre il FS desiderato non è sufficiente una semplice amplificazione o attenuazione dei segnali di uscita, in quanto l'utente potrebbe aver selezionato uno spezzone casuale del segnale registrato, che quindi potrebbe non comprendere la regione di audio che causa il raggiungimento del livello di FS da parte dei segnali risultanti dalla convoluzione. Per tenere conto di tale possibilità, si utilizza un algoritmo che svolge le seguenti funzioni:

1. determina su quale canale n del progetto *Audacity®* si ha il picco massimo di ampiezza tra tutti i canali registrati e ne memorizza il valore ABS_{MAX-dB} ;
2. determina il picco massimo di ampiezza dello stesso canale n , considerando solo il segnale entro la selezione, e ne memorizza il valore REL_{MAX-dB} ;
3. determina il nuovo livello di FS corretto sfruttando la relazione:

$$FS_{FIX_{dB}} = FS_{USER_{dB}} - ABS_{MAX-dB} - REL_{MAX-dB} \quad (7.1)$$

In tal modo, il livello di *FS* specificato dall'utente sarà riscalato in relazione al rapporto tra l'ampiezza massima assoluta del segnale, cioè quella determinata lungo la sua intera durata, e l'ampiezza massima relativa allo spezzone selezionato. Grazie all'introduzione di questo algoritmo il *plug-in* non deve calcolare l'intera convoluzione per poi, dopo aver applicato la correzione del FS, doverne

⁴operazione che nello standard POSIX prende il nome di `join`

scartare la maggior parte, ma viene svolta l'analisi solo sullo spezzone desiderato (??).

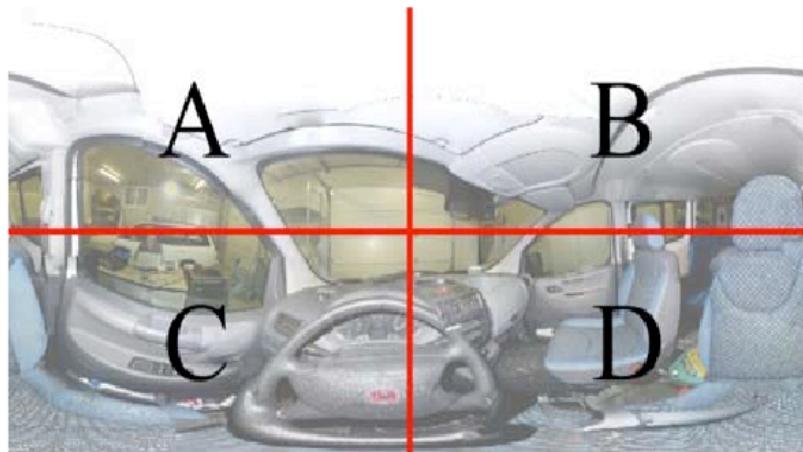
7.4 *Mirroring* dei microfoni virtuali

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 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 7.1.

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.

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



			D	C	C	D	C
		B	A	B	A	B	A
A	B	A	B	A	A	B	A
C	D	C	D	C	C	D	C
D	C	D	C	C	D	B	A
C	B	A	B	A	B	C	D
A	B	A	B	A	B	C	D

Figura 7.1: Rappresentazione schematica del processo di *mirroring*, necessario a garantire la continuità ai bordi della mappa.

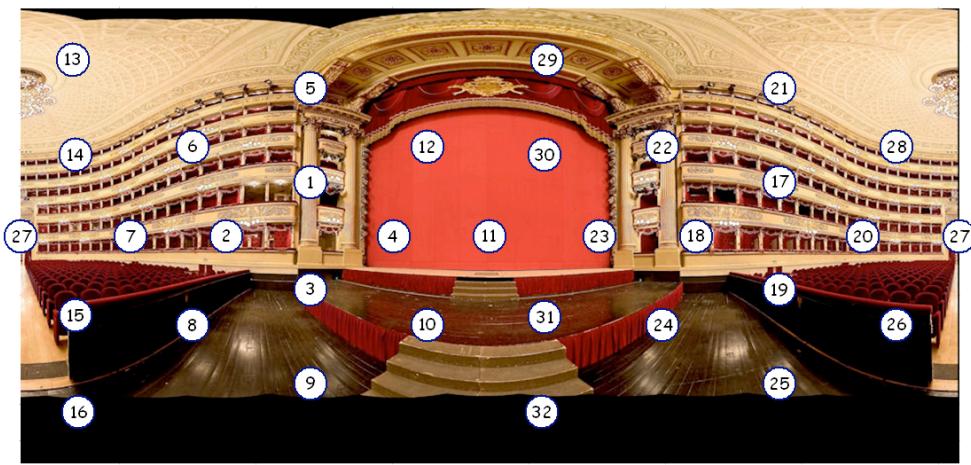


Figura 7.2: Direzioni di puntamento delle capsule dell'Eigenmike sovrapposte ad una foto panoramica del Teatro alla Scala di Milano, svolta secondo lo schema di proiezione visto nella sezione 7.4.

sottoaree in cui si è suddivisa la superficie. Questo primo passaggio è detto *meshing* 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 7.2, le posizioni delle capsule dell’Eigen-mike® purtroppo 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, 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’implemen-

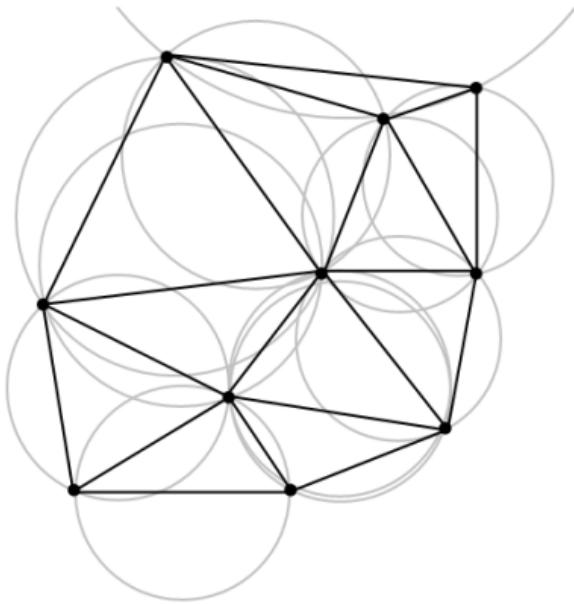


Figura 7.3: Descrizione della proprietà dei triangoli formati con l'algoritmo di *Delaunay*.

tazione particolarmente impegnativa. `Triangle++` è il nome del wrapper C++ della libreria C `Triangle`⁵; 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 Interpolazione dei valori acquisiti e filtraggio in bande d'ottava

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'equa-

⁵Si veda [5]

zione 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:

$$z = A \cdot x + B \cdot y + C \quad (7.2)$$

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:

$$\begin{pmatrix} x_{1i} & y_{1i} & 1 \\ x_{2i} & y_{2i} & 1 \\ x_{3i} & y_{3i} & 1 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} z_{1i} \\ z_{2i} \\ z_{3i} \end{pmatrix} \quad (7.3)$$

L'equazione 7.3 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 7.3, 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, ma ciò non può avvenire. Come si è visto nei precedenti capitoli, infatti, il *plug-in* consente di effettuare una analisi in bande d'ottava⁶ dei livelli sonori, ovvero si avranno non una sola mappa dei livelli, bensì tante mappe quante sono le possibili bande d'ottava selezionabili. Per risolvere occorre usare come terza coordinata di ciascun vertice il numero del microfono virtuale posizionato nelle stesse coordinate del vertice (anziché il livello sonoro) e memorizzare una matrice dei livelli sonori tale che, confrontando il numero del microfono virtuale con il numero identificativo della banda selezionata, fornisca il livello sonoro corretto per la determinazione dei coefficienti A ,

⁶si veda la sezione 2.4.2 per la definizione delle bande utilizzate

B , C e $\det(M)$ necessari per la costruzione della mappa della particolare banda selezionata. Il processo di filtraggio è stato affrontato in modo trasparente grazie all'utilizzo di una classe già progettata per la suite di plug-in **Aurora** per *Audacity®*⁷ la quale già disponeva di tutte le funzioni membro necessarie ad implementare i filtri in bande d'ottava, come definiti secondo norma IEC-1260. Infine, dopo l'operazione di filtraggio, è necessario creare un *frame audio* cioè una mappa statica che rappresenti il contenuto del campo acustico nell'intervallo non infinitesimo di un frame⁸ del video che si sta andando a costruire; occorre qui un calcolo del valore L_{EQ} ⁹ dei segnali filtrati.

7.7 Mappatura con scale cromatiche e *auto-scaling*

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 *RGB* del pixel interessato, il quale sarà successivamente inserito in una *mappa RGB* (una bitmap, appunto). Per assegnare ai valori di *SPL* un valore *RGB* 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 (a meno di accontentarsi di approssimazioni¹⁰), 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 frame¹¹, il programma è in grado di ottenere e salvare

⁷si veda [2]

⁸selezionato dall'utente come al paragrafo ??

⁹definito alla sezione 2.3.3

¹⁰si potrebbe per esempio...

¹¹cioè in seguito al precalcolo descritto in questo capitolo

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.

Con la funzione di `autoscale` 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`¹², 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'utente¹³, 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.

7.8 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®* al momento del salvataggio del *chunck* audio,

¹²con il *checkbox* descritto nella sezione ??

¹³si veda il paragrafo 6.2

- 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 secondo le 10 bande frequenziali con l'aggiunta del valore filtrato A ($dB(A)$) e del filtraggio lineare in banda stretta¹⁴
- 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 filtraggi 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 piattaforma¹⁵. [...]

¹⁴???

¹⁵per ottenere il video — ???

Capitolo 8

Conclusioni

Il plug-in progettato in questa sede è funzionante ed altamente interattivo, con svariate tipologie di realizzazione della mappa grazie alla possibilità di configurare 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 calcolate 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 scoperto un metodo realmente applicabile per tarare una misura effettuata con un array microfonico, si desume che i risultati ottenuti con l'uso del *plug-in* siano corretti.

figura delle tre piattaforme

[...]

Figura 8.1: Graphic User Interface sotto le tre diverse piattaforme principali: *Windows*, *Mac OS X* e *Linux*.

Appendice A

Appendice

In questa appendice finale si trovano tutte le definizioni teoriche indispensabili alla comprensione dell'elaborato qui presentato ma il cui studio è considerato prerequisito e quindi noto. Viene dunque riportato un breve riassunto delle nozioni fondamentali in modo da fornire un quadro generale ristretto ma comunque sufficiente per affrontare serenamente la lettura.

A.1 Propagazione di onde sonore

Un'onda sonora¹, 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 frequenza², mentre il secondo è, in generale, un fluido, di norma l'aria.

La descrizione fisica della propagazione delle onde suddette avviene tramite 3 leggi fondamentali che di seguito si riassumono:

L'equazione di Eulero. Siano p la pressione sonora, \mathbf{u} il vettore velocità istantanea di una particella di fluido e ρ_0 la densità dell'aria in condizioni normali, allora si ha:

¹Vedi [4], capitolo 3.

²Un'onda di pressione perché venga percepita dall'apparato uditivo umano deve avere frequenza compresa tra 20 Hz e 20000 Hz circa, a seconda dell'individuo.

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

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 δ la variazione relativa di densità, o *condensazione*³, 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 γ 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*:

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

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

³È data da

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

dove ρ è densità dell'elemento di fluido considerato.

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

A.2 Intensità, densità di energia e potenza

Di seguito si riportano le definizioni di alcune grandezze fondamentali⁴:

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 θ e ϕ .

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

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

⁴Per i dettagli si rimanda sempre a [4].

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

Per via 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 disperso⁵ 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 :

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

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

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

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

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.⁶ 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

L'argomento è enorme, in questa sede se ne vuole solo riportare la definizione operativa. Se $x(t)$ è una funzione

- 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

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

⁶Si veda [3] al capitolo 2.

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 130th 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, 2009.
- [3] R. SPAGNOLO (A CURA DI), *Manuale di acustica applicata*, UTET, Torino, Italy, 3 ed., gen. 2005. ISBN 88-7750-710-1.
- [4] S. CINGOLANI E R. SPAGNOLO (A CURA DI), *Acustica musicale e architettonica*, UTET, Torino, Italy, 1 ed., dic. 2005. ISBN 88-7750-941-4.
- [5] 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.

Se proprio devo ringraziare...

Typesetted on 2013 September 27 with L^AT_EX 2 _{ε}