

UNIVERSITÀ DI PADOVA



FACOLTÀ DI INGEGNERIA

TESI DI LAUREA

SOUNDRISE: STUDIO E PROGETTAZIONE DI UN'APPLICAZIONE MULTIMODALE INTERATTIVA PER LA DIDATTICA BASATA SULL'ANALISI DI FEATURE VOCALI

Laureando: Stefano Giusto

Relatore: Prof. Antonio Rodà

Correlatore: Dott.ssa Serena Zanolla

Corso di Laurea Magistrale in Ingegneria Informatica

Data Laurea: 17 luglio 2012

Anno Accademico 2011/2012

Sommario

Nel campo della didattica rivolta ai bambini possono essere impiegati, assieme ai tradizionali metodi di insegnamento, strumenti basati su tecnologie multimediali. Alcune soluzioni propongono sistemi che permettono l'insegnamento di nozioni teoriche, ad esempio le norme di sicurezza in un campus, le regole per uno stile di vita corretto o concetti di botanica, puntando ad uno sviluppo della capacità di analisi, comprensione, memorizzazione e valutazione. Altre stimolano il parlato del bambino incoraggiandolo a ripetere parole o fonemi avvalendosi in alcuni casi di animazioni video che mostrano la parte interna ed esterna del tratto vocale. Molte delle applicazioni si presentano però sofisticate e difficili da utilizzare e non in grado di coinvolgere pienamente il bambino nell'attività didattica.

In questo lavoro viene presentato il processo di realizzazione di un'applicazione multimodale interattiva per la didattica, denominata *SoundRise*, basata sull'analisi real-time di feature vocali. *SoundRise* è proposta come un nuovo modo di insegnare ai bambini le caratteristiche della propria voce attraverso un'esplorazione delle capacità vocali, facendo modo che sia lo stesso bambino a scoprire, e poi interpretare e comprendere, il contenuto informativo che riceve attraverso un feedback grafico. Egli ricopre un ruolo attivo dal momento che è lui stesso a controllare tramite la propria voce l'evoluzione grafica del feedback. Questo offre un incentivo all'utilizzo dell'applicazione, con la possibilità di ottenere un maggior impegno e tempo dedicato all'utilizzo di *SoundRise*. L'applicazione è inoltre creata per essere utilizzabile su sistemi operativi differenti, quali Windows, Mac OS X e Linux, e sviluppata pensando anche alla possibilità di estenderne la multimodalità grazie all'integrazione con un altro sistema interattivo, la *Stanza Logo-Motoria*. Questo impiego dell'applicazione permette di rendere ancora più attivo il ruolo del bambino, consentendo un'esplorazione dell'ambiente in cui i due sistemi sono installati.

Per la progettazione è stato utilizzato l'ambiente di programmazione grafica real-time Pure Data (Pd). In appoggio agli oggetti di elaborazione audio forniti da Pd, è stata utilizzata una libreria esterna denominata *timbreID* per l'analisi del timbro vocale.

Nel primo capitolo vengono proposti sistemi multimodali per la didattica intesi come strumenti per aiutare i bambini nel processo cognitivo di nozioni teoriche, che non interessano un'analisi delle produzioni vocali, ed applicazioni che propongono invece un riconoscimento del parlato e delle vocalizzazioni per l'insegnamento ed il training.

Nella prima parte del secondo capitolo viene presentata l'idea alla base del progetto *SoundRise*, gli obiettivi di questo lavoro e gli strumenti utilizzati per lo sviluppo. Segue poi uno studio sull'estrazione delle feature ed in particolare del timbro, per il riconoscimento delle cinque vocali della lingua italiana, e una panoramica della patch Pure Data che definisce *SoundRise*, descrivendo inoltre l'interfaccia grafica ed il mapping delle feature in contenuti informativi multimediali. Nel terzo capitolo si includono i risultati dei test di studio fatti per identificare le soluzioni più adatte all'estrazione delle feature timbriche. Seguono inoltre i risultati di una fase di sperimentazione fatta per valutare l'intuitività dell'interfaccia grafica di *SoundRise* e dalla quale si sono ricavate utili indicazioni di tendenza da confrontare con le prestazioni dell'applicazione riscontrate.

Ringraziamenti

Desidero ringraziare i proff. A. Rodà, F. Avanzini e S. Canazza per il sostegno fornitomi durante la realizzazione e stesura della tesi.

Vorrei ringraziare la dott.ssa S. Zanolle per la disponibilità, l'interesse, la collaborazione e la cordialità concessi nello sviluppo di SoundRise.

Si ringraziano i bambini e il personale della Scuola Primaria Elisa Frinta di Gorizia (Go) per la collaborazione durante i test dell'applicazione.

Ringrazio la mia famiglia per l'immenso sostegno ed affetto datomi e per aver creduto in me sempre.

I miei amici e i compagni di università che mi hanno accompagnato in questa mia fase della vita rendendo gli anni universitari un ricordo ancor più bello.

Un ringraziamento particolare a tutti coloro che hanno condiviso con me corsi, esami, progetti, tesine, laboratori e serate padovane. Ringrazio Marco, compare di tesi, per i copiosi e "impossibili" consigli ed aiuti dati.

Indice

Sommario	i
Ringraziamenti	iii
1 Applicazioni multimodali a fini didattici	1
2 Progettazione e sviluppo dell'applicazione SoundRise in Pure Data	7
2.1 Idea di fondo	7
2.2 Specifiche base di SoundRise	9
2.3 Strumenti di sviluppo utilizzati	10
2.4 Estrazione e analisi della durata e dell'intensità della voce	10
2.4.1 L'oggetto [env~]	10
2.5 Estrazione e analisi dell'altezza della voce	11
2.5.1 L'oggetto [fiddle~]	11
2.6 Analisi del timbro vocale	12
2.6.1 Studio del centroide spettrale e delle formanti	12
2.6.2 L'external [sFormCENTROID]	14
2.6.3 Soluzione adottata: timbreID	15
2.6.3.1 Analisi del cepstrum - Gli oggetti [mfcc~] e [bfcc~] . . .	17
2.6.3.2 Identificazione delle vocali	18
2.7 Overview della patch Pd SoundRise	20
2.7.1 Dettagli della patch	20
2.7.2 SoundRiseCore	21
2.7.3 La patch <i>pd settings</i>	24
2.8 Interazione con l'utente	27
2.8.1 Interfaccia grafica	29
2.8.2 Rappresentazione grafica	31
2.9 Impiego di SoundRise	34
2.9.1 Modalità di utilizzo	35
2.9.1.1 La Stanza Logo-Motoria	35
2.9.1.2 Integrazione con la Stanza Logo-Motoria	36

2.9.1.3	Utilizzo dell'applicazione	37
3	Valutazioni	41
3.1	Analisi di <i>timbreID</i>	41
3.1.1	Risultati	46
3.2	Test di Usabilità per misurare l'intuitività dell'interfaccia di <i>SoundRise</i>	49
4	Conclusioni	67
	Appendici	73
A	Strumenti software	73
A.1	Pure Data	73
A.2	EyesWeb	75

Elenco delle figure

2.1	Mappatura delle caratteristiche vocali nelle caratteristiche grafiche.	9
2.2	Schema a blocchi del sistema di estrazione delle feature vocali.	10
2.3	timbreID in una configurazione di training.	16
2.4	Patch di identificazione delle vocali <i>a</i> , <i>e</i> ed <i>o</i>	19
2.5	Contenuto della patch SoundRise.pd.	21
2.6	Ingresso audio.	22
2.7	Filtraggio del segnale audio di ingresso.	23
2.8	Patch per il riconoscimento delle vocali: <i>pd vowel</i>	24
2.9	Blocco relativo all'analisi dell'ampiezza e della durata del segnale vocale.	25
2.10	Estrazione del pitch del segnale audio in ingresso	26
2.11	Blocco relativo all'analisi dell'altezza del segnale vocale.	27
2.12	Blocco relativo alla rappresentazione grafica delle vocali, dell'intensità e del pitch.	28
2.13	Blocco relativo alla rappresentazione grafica della durata.	29
2.14	Sezione relativa al caricamento dell'immagine di sfondo della finestra grafica	30
2.15	Console di controllo di SoundRise	31
2.16	Esempio di schermate della finestra grafica: (a) modalità analisi intensità, (b) modalità analisi pitch, (c) modalità analisi durata, (d) modalità analisi timbro	33
2.17	Indice delle zone in cui è divisa la stanza analizzata tramite il sistema Stanza Logo-Motoria.	36
2.18	Blocchi della patch EyesWeb per l'invio dell'indice della zona della stanza.	38
2.19	Oggetti della patch <i>SoundRise.pd</i> per la ricezione dell'indice della zona della stanza.	39
2.20	(a) Foto dell'ambiente in cui vengono integrate la <i>Stanza Logo-Motoria</i> e <i>SoundRise</i> (b) webcam per la cattura della posizione dell'utente (c) <i>SoundRise</i> in esecuzione su un PC.	40
3.1	Patch per la valutazione degli external [bfcc~] e [mfcc~]	42
3.2	Patch per la valutazione degli external [bfcc~] e [mfcc~]	43
3.3	Patch che combina le misure di [bfcc~], [mfcc~] e [specCentroid~] in un unico vettore di feature.	44
3.4	Patch che combina le misure di [bfcc~], [mfcc~] e [specRolloff~] in un unico vettore di feature	45
3.5	Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra [bfcc~] e [mfcc~].	46

3.6	Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra [bfcc~] e [mfcc~] combinati con il centroide.	47
3.7	Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra [bfcc~] e [mfcc~] combinati con la frequenza di roll-off. . . .	48
3.8	Valori complessivi di correttezza in percentuale, ottenuti considerando l'identificazione di tutte le vocali, dati da [bfcc~] e da [mfcc~] combinato con la frequenza di roll-off e con il centroide.	49
3.9	Istogramma che mostra la percentuale di risposte corrette date dai bambini alla Domanda 1 per tutte le feature analizzate individualmente.	51
3.10	Istogramma che mostra la percentuale di combinazioni di feature identificate dai bambini come risposta alla Domanda 1 per tutte le feature analizzate contemporaneamente. I = Intensità, A = Ampiezza, D = Durata, T = Timbro.	52
3.11	Istogramma che mostra la percentuale di risposte corrette date dai bambini alla Domanda 2 per tutte le feature analizzate individualmente.	53

Elenco delle tabelle

2.1	Frequenza delle prime tre formanti delle vocali italiane; le ultime due righe si riferiscono alla versione aperta di /e/ e /o/	13
2.2	Correttezza sFormCENTROID	15
2.3	Associazione vocale-indice identificativo	22
2.4	Associazione vocale-colore	34
3.1	Risultati [bfcc~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	54
3.2	Risultati [mfcc~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	55
3.3	Risultati [bfcc~] con 40 coefficienti: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	56
3.4	Risultati [bfcc~] combinato con [specCentroid~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	57
3.5	Risultati [bfcc~] combinato con [specCentroid~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	58
3.6	Risultati [mfcc~] combinato con [specCentroid~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	59
3.7	Risultati [mfcc~] combinato con [specCentroid~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	60
3.8	Risultati [bfcc~] combinato con [specRolloff~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	61
3.9	Risultati [bfcc~] combinato con [specRolloff~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	62
3.10	Risultati [mfcc~] combinato con [specRolloff~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	63
3.11	Risultati [mfcc~] combinato con [specRolloff~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino	64
3.12	Risultati del test di Usabilità di SoundRise	65

Capitolo 1

Applicazioni multimodali a fini didattici

Il rapido sviluppo delle tecnologie e la loro disponibilità sta consentendo l'utilizzo di nuove forme di approcci alla didattica basate sull'utilizzo di tecnologie multimediali. Queste offrono forme di intervento ed interazione che invitano l'utente ad esercitarsi con la pratica e a ricevere feedback opportuni in risposta al suo modo di interagire e si presentano come uno dei modi più promettenti tramite cui l'istruzione può essere personalizzata. Non è sempre facile, attraverso le tradizionali forme di interazione, fornire un approccio all'insegnamento e un feedback sufficienti. L'assistenza data dalla tecnologia può offrire soluzioni adeguate, permettendo a molti bambini e adulti di interagire con il loro mondo nonostante alcune limitazioni espressive nel linguaggio. Alcuni prodotti, che mirano a perseguire tali obiettivi, si presentano molto sofisticati e alquanto difficili da utilizzare, il che fa capire la necessità di renderli più user friendly e, cosa importante, devono costituire una tecnologia capace di mantenere l'impegno, la capacità interattiva, la flessibilità e la creatività dell'utente[1].

Giochi per computer, visti come piattaforme per lo studio, possono stimolare un bambino ad aumentare le proprie motivazioni nell'apprendimento attraverso un'esperienza di gioco. Si parla di *game-based learning* (GBL). In [2] viene proposto un framework per replicare il processo di apprendimento cognitivo in unità GBL. Il modello proposto permette a studenti delle scuole elementari di accedere a tre tipi di processi cognitivi: *ricordare e comprendere, analizzare e applicare* e *valutare*. Rispetto all'apprendimento tradizionale attraverso pagine web di testo, GBL risulta una piattaforma più attraente ed interessante che permette all'utente di ottenere un miglioramento nei risultati. Invece di essere visti come piattaforme di intrattenimento, i giochi per computer vengono invece considerati un canale di distribuzione di contenuti digitali. Il framework proposto mira a replicare il processo cognitivo in un gioco per computer con il fine di trasmettere conoscenze di base riguardanti la sicurezza nel campus e un buono stile di vita, pro-

poste come scenario di gioco. Il processo viene reso simile ad un processo di risoluzione di un problema in cui sono presenti compiti da assegnare all'utente, obiettivi, che è necessario portare a termine per finire il gioco. Il feedback dato è una sorta di rinforzo che rappresenta una forma di ricompensa (feedback positivo) nei confronti di un comportamento desiderabile assunto. L'utente riceverà un feedback positivo, o negativo, se una particolare conoscenza concettuale è applicata in modo appropriato, o inappropriato, alla risoluzione del problema. La conoscenza viene presentata come un tipo di strumento di gioco da utilizzare per completare una missione. Obiettivi, feedback e strumenti sono infatti le tre maggiori componenti di un gioco per computer [2].

Oltre all'apprendimento di nozioni teoriche, sistemi basati su modalità di gioco permettono di stimolare il parlato del bambino. In ambito educativo, infatti, courseware multimediali possono essere usati per bambini che presentano qualche difficoltà di apprendimento che incidono sull'ascolto, sul parlato, sulla lettura e scrittura o sulle abilità matematiche. L'uso di courseware, basati sul gioco, può aiutare questi bambini ad esplorare le loro capacità permettendogli di apprendere in base ai propri ritmi. Confrontati con i sistemi di apprendimento tradizionali, tali sistemi sono in grado di suscitare una maggiore attrazione da parte degli studenti. Il courseware *Playtime Learning* [3] è stato definito per studenti con difficoltà di apprendimento, concentrando lo sviluppo sull'impiego di materiali che possono stimolare le capacità di pensiero dei bambini. Il sistema è stato testato su un gruppo composto da alcuni studenti e insegnanti di una scuola. All'utente veniva chiesto ad esempio di completare una serie di operazioni matematiche o di ripetere numeri o di riordinare le lettere per ricomporre una parola nel modo corretto. Le attività del courseware *Playtime* oltre che a svolgere esercizi di ascolto e interazione, incoraggiano il bambino ad usare la propria voce ripetendo informazioni verbali.

Design-A-Plant [4] è un ambiente didattico che interessa l'insegnamento di concetti riguardanti l'anatomia e fisiologia botanica ed è rivolto agli studenti delle scuole medie. Include un agente pedagogico animato chiamato *Herman-the-Bug* che fornisce un feedback sulla scelta fatta dallo studente nel processo di creazione di piante. Da consigli allo studente su come realizzare graficamente piante a partire da una libreria di strutture vegetali come radici e steli. L'obiettivo è quello di creare piante che crescano in un dato ambiente naturale in determinate condizioni, quali la quantità di luce solare disponibile. Come lo studente risolve i task legati alla creazione delle piante, l'agente fornisce loro consigli di anatomia e fisiologia botanica. I risultati del test di tale programma [5] mostrano come, rispetto ad una situazione in cui non si ha la possibilità di interagire con un agente, lo studente presenti meno difficoltà nel risolvere i problemi proposti

dal programma ed inoltre ricevi un'esperienza didattica più interessante.

Questi sistemi per la didattica non offrono un'analisi del parlato come strumento per potenziare l'efficacia del metodo. Il sistema *visiBabble* [6], al contrario, elabora le vocalizzazioni infantili in tempo reale. Esso presenta un'ottica più vicina al lavoro sviluppato in questa tesi, per quella che è l'idea di analisi ed elaborazione della produzione vocale come base per l'interazione con l'utente. Il sistema risponde alle produzioni di sillabe da parte del bambino con animazioni colorate e registra l'analisi acustico-fonetica. È in grado di monitorare semplici aspetti del segnale acustico in tempo reale basandosi sullo spettrogramma. Permette di rafforzare la produzione di sillabe caratterizzanti il linguaggio e lo sviluppo cognitivo. Lo scopo di *visiBabble* consiste nell'incrementare le vocalizzazioni dei bambini e nell'aumentare la raffinatezza e la varietà di tali vocalizzazioni rinforzando le produzioni sillabiche rispetto a quelle non-sillabiche. Il programma viene usato dal bambino come un giocattolo o da un professionista come uno strumento clinico o di ricerca. Il sistema prototipo *visiBabble* include un notebook, un microfono e un software che esegue funzioni di individuazione dei punti in un enunciato intorno ai quali estrarre informazioni sulle feature distintive di fondo. Fornisce inoltre un feedback su monitor, dando così una risposta visiva real-time per il suono in ingresso, registra ogni sessione e colleziona dati sul tipo e sulla durata delle vocalizzazioni prodotte.

Per l'insegnamento e il training di produzioni vocali, i sistemi computer-based speech training (CBST), sono oggi ampiamente usati per bambini con disturbi dell'udito o del linguaggio ma possono anche essere utilizzati a fini didattici per l'apprendimento di una lingua. Nell'ambito di un progetto fondato nel 1999 dall'unione europea, denominato *SPECO*, è stato sviluppato un sistema di *audio-visual pronunciation teaching and training* per bambini di 5-10 anni. Con tale sistema una correzione degli aspetti disordinati del linguaggio è eseguita in real-time adottando anche una presentazione visiva dei parametri del linguaggio. Vengono così sincronizzate informazioni uditive con quelle visive in modo che il tutto risulti comprensibile e interessante per i bambini [7]. Questo sistema multimodale è stato sviluppato per tutti i linguaggi dei partner *SPECO*, ovvero per l'inglese, lo svedese, lo sloveno e l'ungherese. Due database sono stati raccolti e inseriti nel sistema per tutte le lingue trattate, uno per l'insegnamento e il training delle vocali per bambini con disturbi uditivi e l'altro per la correzione dei suoni fricativi e affricati disarticolati. Lo scopo dell'utente durante la terapia è la produzione, tramite la propria voce, di pattern visivi simili a quelli degli esempi di riferimento memorizzati nel sistema. Durante il training il paziente vede i pattern visivi del parlato di riferimento e allo stesso tempo ne ascolta il suono. Può inoltre ottenere un confronto visivo tra la propria pronuncia e i riferimenti memorizzati. Il feedback

automatico per il confronto è basato su un calcolo della distanza tra le componenti spettrali di spettrogrammi del parlato tipico e del parlato effettivo dell'utente. Questo feedback fornisce una buona opportunità per la pratica senza insegnante, in cui l'utente è solo con il sistema.

SIM (Speech Illumina Mentor – Soleymani, McCutcheon e Southwood 1997) [8] è un programma CBST sviluppato principalmente per bambini di età scolare con problemi di udito. Il programma presenta due componenti principali: un modulo di training che offre al bambino esempi memorizzati di articolazioni e un modulo di pratica che presenta un sistema di riconoscimento vocale per valutare le produzioni in un format di gioco. Il modulo di training usa un display che mostra i movimenti sia delle articolazioni interne che di quelle esterne. Per le parti interne il display anima le strutture normalmente nascoste come la lingua, il palato, la faringe e il velo palatino. L'animazione è realizzata facendo scorrere una sequenza di immagini del tratto vocale. La parte esterna è una vista frontale di un'immagine video modificata delle labbra e della mandibola di una persona che parla. Durante la sessione di pratica il sistema valuta i tentativi dell'utente di produrre le sillabe previste. Produzioni corrette provocano la presentazione di feedback visivi significativi, in relazione al successo o al fallimento di ogni tentativo. Un format di gioco e personaggi animati vengono usati per instaurare una comunicazione con il bambino e per motivarlo. *SIM* può essere visto come un insieme di moduli che si concentrano sul training di fonemi con livelli di difficoltà crescenti. Ogni modulo si concentra sull'istruzione di un fonema ed è responsabile dell'animazione dei tratti vocali articolatori interni ed esterni. L'animazione di ogni frame può essere ripetuta un numero qualsiasi di volte fino a quando l'utente non ottiene le abilità necessarie per ripetere il suono previsto. Una volta che l'utente è pronto, può passare alla fase di pratica dove testare le abilità acquisite. Viene infatti messo in interazione con il modulo di pratica dove il sistema di riconoscimento vocale monitora le produzioni e le confronta con i modelli interni memorizzati. Un algoritmo di pattern matching permette di mostrare a display i successi dei tentativi nel riprodurre i suoni previsti. I modelli possono essere cambiati da genitori o insegnanti.

Un ulteriore strumento per mostrare, in ambito educativo, i movimenti articolatori e la posizione della lingua durante le pronuncia è la *testa parlante*, animata al computer, denominata *Baldi* (Massaro e Light, 2004) sviluppata al Perceptual Science Laboratory di Santa Cruz (University of California) [9]. L'utilità di *Baldi* unita ad un particolare toolkit sviluppato per il riconoscimento e la sintesi vocale risulta evidente per coloro che vogliono imparare una seconda lingua e per bambini con difficoltà uditive. È stato sviluppato con l'intento di poter essere utilizzato per applicazioni didattiche, al fine di permetterne un impiego agli insegnanti durante le lezioni.

L'integrazione del toolkit nelle scuole deve passare attraverso un corso intensivo di una settimana rivolto ai docenti, in cui vengono date istruzioni su come utilizzare lo stesso toolkit.

Baldi viene usato per apprendere nuovi vocaboli ed esercitarsi su singoli suoni del parlato aiutando gli studenti a pronunciare correttamente parole, definizioni e frasi.

In ambito commerciale, *Fast ForWord* [10] costituisce una famiglia di programmi educativi, distribuiti dalla Scientific Learning Corporation, con lo scopo di migliorare le capacità cognitive e di lettura dei bambini, focalizzandosi in particolare sullo sviluppo della consapevolezza fonologica. *Fast ForWord* è un programma di intervento ideato per istituti di istruzione i cui studenti presentano difficoltà nella lettura. Identifica le carenze nell'apprendimento del singolo studente andando ad analizzare le risposte agli esercizi proposti, regolando inoltre il grado di difficoltà per tenere lo studente sempre in una situazione di attenzione verso i compiti da eseguire.

Il software *Reading Assistant* del programma *Fast ForWord* agisce come un tutor personale in fase di lettura fornendo un sostegno per lo studente che legge ad alta voce. Sfruttando così una tecnologia brevettata, è fornito un feedback correttivo in tempo reale, permettendo allo studente di correggersi quando sta leggendo a voce alta. Alla tecnologia per l'analisi del parlato vengono affiancati interventi, di base scientifica, per aiutare lo studente a rafforzare la fluidità nella lettura, il proprio vocabolario e la comprensione. Lo studente ha infatti la possibilità di ascoltare un esempio di lettura scorrevole del testo, rivedere la definizione di parole problematiche non comprese e riascoltare la propria lettura.

In quest'ambito della didattica attraverso tecnologie multimediali, l'applicazione *SoundRise*, progettata in questo lavoro di tesi, si propone come un sistema, che sfrutta un feedback grafico, per l'apprendimento di quelle che invece costituiscono le caratteristiche base della voce, estratte ed elaborate in tempo reale. *SoundRise* è rivolta principalmente ai bambini delle scuole elementari e propone un'interattività legata ad una risposta immediata e comprensibile all'azione dell'utente durante l'apprendimento. Il feedback grafico in risposta agli input vocali è in grado di assistere il bambino per promuovere la comprensione in modo da essere un aiuto nel processo cognitivo. Spinge inoltre a riflettere sul cosa, nella sua voce, ha portato ad ottenere tale riscontro grafico. In *SoundRise* è il bambino a controllare con la propria voce l'evoluzione del feedback grafico. È riunita quindi l'idea dell'apprendimento come gioco, in modo che il bambino sia stimolato ad esercitarsi il più possibile, ad un'analisi real-time del suono vocale come strumento per favorire l'apprendimento. Il tipo di feedback mira ad incrementare lo sforzo che il bambino mette nell'apprendere e la grande motivazione fornita dal sistema può incoraggiare a dedicare più tempo all'apprendimento. L'esercitazione è infatti alla base dell'apprendimento,

di conseguenza il sistema è sviluppato per favorirne un continuo impiego. Al contrario di altri sistemi interattivi per la didattica, *SoundRise* permette un utilizzo che non limita il bambino a rimanere seduto con lo sguardo fisso verso uno schermo ma che invece aumenta la multimodalità nell'interazione lasciandogli la possibilità di esplorare l'ambiente in cui l'applicazione è usata. In base alle zone in cui si muove, infatti, può decidere il tipo di feedback da ricevere. Questo può risultare più stimolante per il bambino perché, oltre ad offrire un ruolo attivo, richiede anche un lavoro di riflessione e memorizzazione delle zone che hanno generato quel particolare feedback visivo, piuttosto che un altro, a seguito dell'emissione di un suono con la propria voce. Un'altra caratteristica che distingue *SoundRise* consiste nel fatto di non essere limitata al funzionamento in un solo sistema operativo, cosa che in molti altri sistemi rappresenta un limite, ma può essere invece eseguita in sistemi Windows, Mac OS X e Linux.

Capitolo 2

Progettazione e sviluppo dell'applicazione SoundRise in Pure Data

2.1 Idea di fondo

Lo sviluppo di nuovi strumenti per l'apprendimento [11] è particolarmente impegnativo perché devono essere applicate ed integrate conoscenze da diverse aree. Gli aspetti dello sviluppo da considerare includono:

- principi di insegnamento adeguati;
- modi di rappresentazione della conoscenza attraverso contenuti multimediali. La scelta può essere fatta in accordo alle caratteristiche dei contenuti, a modelli cognitivi o in accordo a una classificazione dei contenuti didattici;
- tecnologie per l'apprendimento abbinate alle strategie di insegnamento;
- interazione con l'utente: diversi modi di interagire con un computer significa anche diversi modi di influenzare i processi cognitivi strettamente legati all'apprendimento;
- accettazione da parte dell'utente: è una questione legata ai benefici e agli svantaggi dello strumento didattico proposto.

Tenendo in considerazione questi aspetti, l'applicazione *SoundRise* è stata sviluppata, in collaborazione con la dott.ssa Serena Zanolla dell'Università degli Studi di Udine e dei proff. Federico Avanzini, Antonio Rodà e Sergio Canazza del Dipartimento di Ingegneria dell'Informazione dell'Università di Padova, partendo dall'idea di poter rappresentare tramite dei tratti grafici ben

distinti quelle che sono le caratteristiche (feature) più significative della voce con il fine di farle comprendere ai bambini. Il progetto è stato inizialmente pensato per scopi logopedistici, considerando le difficoltà che alcuni bambini hanno nel pronunciare certi tipi di suoni e nel conoscere i tratti distintivi della voce. Il lavoro fin qui svolto permette di valutare un utilizzo dell'applicazione a fini didattici, come l'impiego nelle scuole, non avendo avuto modo e tempo di sviluppare e sperimentare quanto realizzato con la collaborazione di specialisti che avrebbero potuto permetterne la rielaborazione in ambito logopedistico. Stabilita l'idea alla base, si sono poi definiti i dettagli. Le feature del segnale vocale considerate sono le seguenti:

- intensità: caratteristica in base alla quale i suoni possono essere ordinati su una scala che si estende dal piano al forte [12];
- altezza (pitch): consente di ordinare un suono su una scala che si estenda dal basso verso l'alto [12]. Indica la maggiore o minore acutezza o gravità del suono;
- durata: è la proprietà del suono che ne limita la maggiore o minore lunghezza o brevità;
- timbro: attributo che consente di giudicare che due suoni non identici, presentati allo stesso modo e aventi la stessa intensità e altezza, sono diversi [12];

Il riconoscimento dell'ultima proprietà, il timbro, è relativo all'identificazione delle cinque vocali della lingua italiana. Il sistema di riconoscimento delle feature è realizzato sulle caratteristiche vocali dei bambini.

Il feedback grafico con cui rappresentare a video le quattro feature è stato pensato facendo molto affidamento sull'esperienza della dott.ssa Zanolla nel campo dell'insegnamento ai bambini delle scuole elementari. Quello che si è voluto definire è un ambiente grafico in grado di attirare l'attenzione dell'utente e di stimolare un pieno coinvolgimento dello stesso, che permetta inoltre di comprendere e distinguere le informazioni contenute.

Per quanto riguarda l'interazione tra il bambino che deve utilizzare l'applicazione e l'applicazione stessa si è deciso di permettere la scelta di quale feature vocale si vuole analizzare, e di conseguenza visualizzare e conoscere, adottando due modalità. La prima riguarda una selezione tramite interfaccia grafica. La seconda prevede un'estensione di quella che è la multimodalità dell'applicazione permettendo una scelta tramite il movimento all'interno della stanza in cui l'utente si trova. Per questa modalità *SoundRise* viene integrata ad un altro sistema multimodale interattivo adottato per l'apprendimento, la *Stanza Logo-Motoria* [13]. L'integrazione tra i due sistemi viene descritta in dettaglio nel Paragrafo 2.9.1.

2.2 Specifiche base di SoundRise

I punti base imposti per lo sviluppo di *SoundRise* hanno previsto:

- estrazione delle feature in tempo reale: in base al parametro del segnale vocale scelto, il sistema deve essere in grado di rilevare il valore di quello specifico parametro in ogni istante;
- rappresentazione grafica del valore della feature analizzata in ogni momento: i valori delle feature individuate devono essere elaborati per offrire un feedback visivo all'utente in una finestra grafica. Un sole è preso come soggetto rappresentante le feature riconosciute da SoundRise. In corrispondenza di ognuna delle feature che si vuole il sistema analizzi, il sole viene animato in modo diverso. Nello specifico le quattro feature analizzate vengono mappate nelle caratteristiche grafiche mostrate in Figura 2.1
- interfacciamento: l'applicazione deve presentarsi all'utente come facile da utilizzare sfruttando un'interfaccia grafica ben definita.

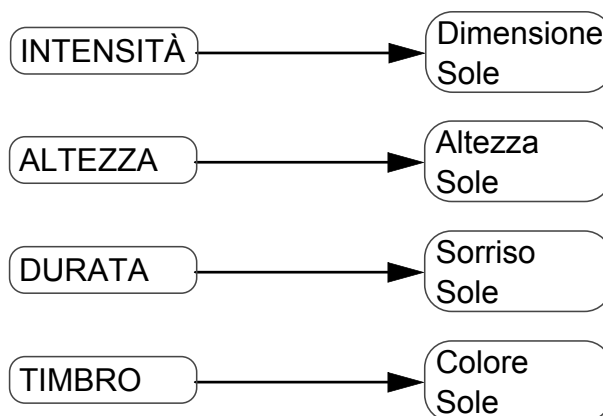


Figura 2.1: Mappatura delle caratteristiche vocali nelle caratteristiche grafiche.

In Figura 2.2 è riportato uno schema a blocchi del sistema di elaborazione delle feature. Il sistema è costituito da un microfono il cui segnale è elaborato in tempo reale. Il segnale in ingresso viene convertito in forma digitale. Dal segnale processato vengono estratte le feature vocali, che vengono poi mappate in caratteristiche grafiche dando un feedback visivo temporalmente correlato con l'evento sonoro. Il rendering grafico è visualizzato su uno schermo.

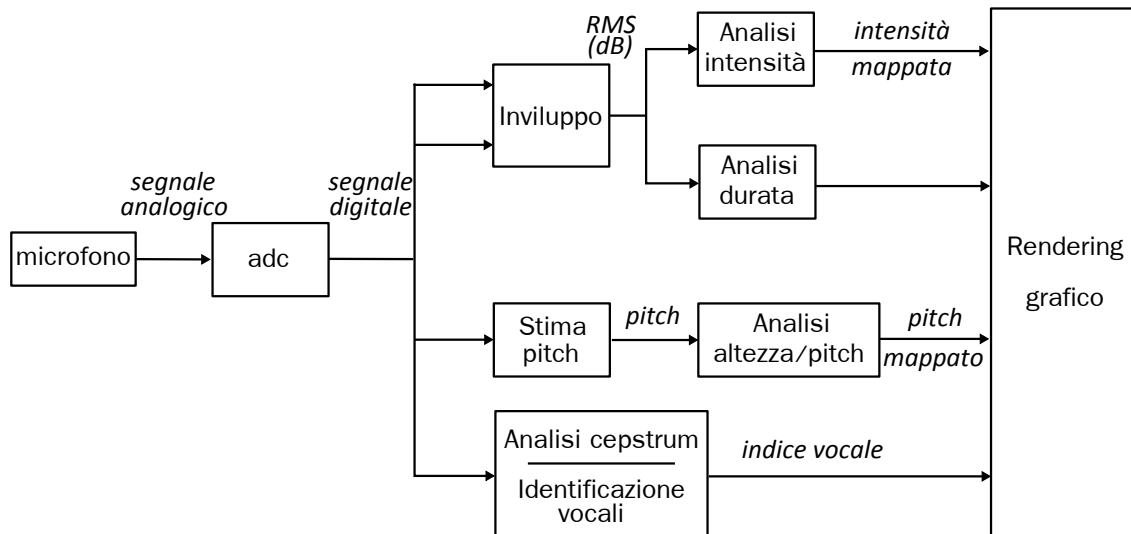


Figura 2.2: Schema a blocchi del sistema di estrazione delle feature vocali.

2.3 Strumenti di sviluppo utilizzati

Pure Data (Pd) [A.1] è l'ambiente di programmazione grafica real-time adottato per la creazione dell'applicazione. Pd è scritto per essere multiplatforma. Ciò conferisce a *SoundRise* una certa portabilità consentendone l'utilizzo su computer caratterizzati da sistemi operativi diversi ed anche su dispositivi mobili [14]. Una serie di patch Pd relazionate tra loro rappresenta così l'intera applicazione. EysWeb [A.1] è la piattaforma software utilizzata per estendere la multimodalità del sistema, integrandolo con la *Stanza Logo-Motoria*.

2.4 Estrazione e analisi della durata e dell'intensità della voce

Il segnale vocale in ingresso all'applicazione viene elaborato attraverso oggetti messi a disposizione da Pd ed external creati per necessità di sviluppo [14].

2.4.1 L'oggetto [env~]

L'oggetto [env~] [15] di Pd prende un segnale audio in ingresso e manda in uscita il corrispondente valore efficace dell'ampiezza in dB. L'uscita è limitata inferiormente al valore 0. La

finestra di analisi adottata dall'oggetto è quella di Hanning (coseno rialzato). Il numero di campioni analizzati all'interno della finestra, che ne definisce la dimensione, è fissato a 1024 mentre il periodo di analisi, sempre in campioni, è pari a metà della dimensione della finestra. La dimensione della finestra e il periodo sono argomenti della funzione svolta da `[env~]` che possono essere variati.

L'uscita di tale oggetto viene così presa in analisi per la durata e l'intensità del segnale. Per quanto riguarda l'analisi della durata viene considerato un valore di *soglia di rumore* (di fondoscala) oltre il quale si può considerare il segnale che si riceve in ingresso come segnale da analizzare, e non rumore di fondo. Tale soglia è calcolata attraverso una patch creata specificamente per il calcolo della media della potenza del segnale di ingresso su 3 secondi di campioni all'avvio dell'applicazione. La durata del segnale è così definita dall'intervallo che ha come inizio l'istante in cui viene rilevato il superamento della soglia calcolata e come fine l'istante in cui la potenza del segnale torna al di sotto di tale soglia.

Per l'analisi dell'intensità il valore in dB della potenza del segnale in ingresso è mappato su una scala di valori reali che va da 0 a 4. Questo range di valori permette di gestire le animazioni visive nel feedback grafico.

2.5 Estrazione e analisi dell'altezza della voce

L'altezza del segnale vocale è analizzata secondo una scala assoluta e relativa dove nel secondo caso i valori dell'altezza variano rispetto a quello che è il tono medio della voce dell'utente.

2.5.1 L'oggetto `[fiddle~]`

L'oggetto `[fiddle~]` di Pd stima il pitch e l'ampiezza di un segnale che giunge al suo ingresso. La dimensione della finestra di analisi è impostata a 2048 campioni. Un'analisi dell'ingresso viene fatta ogni 1024 campioni (metà della dimensione della finestra) il che significa ogni 23.2 ms (ricordando che Pure Data opera ad una frequenza di campionamento di 44100 Hz). L'oggetto è basato sull'algoritmo per il calcolo della FFT (Fast Fourier Transform) che esegue un'analisi sia del volume che del pitch.

I valori continui del pitch estratti vengono utilizzati per rappresentare graficamente il parametro altezza. Per la scala assoluta una soglia minima è stata fissata in modo che al di sotto del va-

lore corrispondente il pitch sia considerato nullo. Il valore estratto viene mappato su un range di valori reali che va da 0 a 4 dove il valore 0 è associato ad un valore di pitch inferiore alla soglia fissata ed il valore 4 ad un valore massimo accettato. In termini di frequenze viene presa in considerazione una scala che va da circa 100 Hz ad oltre 400 Hz.

2.6 Analisi del timbro vocale

2.6.1 Studio del centroide spettrale e delle formanti

La ricerca delle soluzioni per ottenere un buon sistema di riconoscimento del timbro vocale ha visto come passo iniziale un piano di lavoro a partire da un altro sistema, denominato *voice painter* [16], che permette di dipingere su una tela virtuale attraverso l'uso della propria voce e del movimento del corpo ed anch'esso utilizzante Pure Data per l'elaborazione audio in tempo reale. In voice painter alcuni parametri caratterizzanti la voce umana vengono controllati facilmente dall'utente e stimati in tempo reale. Tra questi è incluso il centroide, definito come il centro di gravità dell'ampiezza dello spettro calcolata su un frame audio. Il centroide spettrale [17] corrisponde ad una feature timbrica che descrive la brillantezza di un suono ed è quindi usato in audio digitale e in elaborazione musicale come misura automatica del timbro musicale. Durante il tempo di vita di un suono il centroide non rimane statico ma varia e, per di più, varia tipicamente con l'intensità. Si può definire dunque il centroide spettrale come nell'equazione 2.1:

$$SC = \frac{\sum_{k=1}^{N-1} (k * X[k])}{\sum_{k=1}^{N-1} (X[k])} \quad (2.1)$$

$X[k]$ è l'ampiezza corrispondente al valore discreto di frequenza k , N è la lunghezza della DFT (Discrete Fourier Transform) e SC è il centroide spettrale in Hertz.

Il centroide normalizzato rispetto alla frequenza fondamentale è utile nel contesto dei suoni acuti poiché suoni inarmonici o puramente percussivi mancano di una frequenza fondamentale.

In [18] è analizzata la dipendenza della brillantezza timbrica percepita rispetto al centroide spettrale, per note singole e coppie di note simultanee. Viene considerata nell'analisi la dipendenza del timbro, in un tono costante, dallo spettro di potenza, ovvero dalla distribuzione di potenza in

Vocali	Voce femminile		
	F1	F2	F3
/a/	1002,90	1549,95	2959,70
/e/	437,03	2429,76	3087,09
/i/	361,90	2583,89	3378,14
/o/	444,89	914,26	2899,80
/u/	461,82	763,41	2902,55
/ɛ/	672,45	2242,93	3018,60
/ɔ/	715,34	1073,27	2981,69

Tabella 2.1: *Frequenza delle prime tre formanti delle vocali italiane; le ultime due righe si riferiscono alla versione aperta di /e/ e /o/*

funzione della frequenza. Una semplice quantificazione della distribuzione dello spettro di potenza è proprio il centroide spettrale. Dai risultati ottenuti il semplice centroide spettrale risulta un buon predittore di brillantezza percepita, migliore rispetto al centroide normalizzato.

Il calcolo del centroide nel sistema voice painter è realizzato da un oggetto Pd denominato [CENTROID]. Da questo si è cercato di riscrivere un external per l'identificazione delle cinque vocali passando prima per uno studio sulle formanti. Le formanti [19] sono esattamente le frequenze di risonanza di un tratto vocale quando viene pronunciata una vocale. Queste frequenze sono le frequenze in cui la concentrazione di energia acustica è maggiore. Lo spettro di fonemi può essere costituito da diverse formanti ma le prime tre sono le più importanti per il riconoscimento poiché contengono sufficienti informazioni affinché il nostro sistema di audio-codifica identifichi e classifichi un timbro vocalico. La frequenza della prima formante F1 è in relazione con l'ampiezza dell'apertura della bocca, la frequenza della seconda formante F2 è determinata dalle diverse posizioni della lingua ed infine la frequenza della terza formante F3 è importante per la qualità e la chiarezza del fonema pronunciato.

Per ogni vocale la posizione delle formanti non cambia pur variando la frequenza che si sta emettendo. Ciò che fa la differenza è la diversa quantità di informazioni contenute in ciascuna formante. Emettendo la stessa vocale su frequenze diverse si determinino identiche formanti ciascuna con una quantità diversa di componenti armoniche [20].

In Tabella 2.1 vengono riportate le frequenze in Hz delle prime tre formanti delle vocali, riferendosi ad una voce femminile.

Vengono prese in analisi le frequenze della voce femminile in quanto più simili a quelle della voce di un bambino.

2.6.2 L'external [sFormCENTROID]

Considerando le formanti caratterizzanti ogni vocale e prendendo come base l'oggetto [CENTROID], si è implementato in linguaggio C l'external *sFormCENTROID* per l'utilizzo nella patch Pd. Visual Studio è stato impiegato come ambiente di sviluppo.

sFormCENTROID è basato sul calcolo delle distanze tra la frequenza teorica di ogni formante e il valore del centroide calcolato considerando le componenti armoniche fino a quella formante. Nel codice è utilizzato il calcolo della Trasformata Discreta di Fourier (DFT) che analizza una finestra di 1024 campioni per calcolare l'ampiezza delle componenti sinusoidali del segnale in ingresso. La banda utile del segnale è stata divisa in 3 porzioni considerando i valori della Tabella 2.1. Le ampiezze corrispondenti all'intervallo di frequenze fino a 1020 Hz (prima formante) vengono utilizzate per il calcolo di un primo centroide (centroide1), quelle fino alle componenti sinusoidali fino a 2600 Hz (seconda formante) per ottenere un secondo valore di centroide (centroide2) e infine quelle fino alle componenti sinusoidali fino a 3400 Hz (terza formante) per il terzo valore di centroide (centroide3). Con il centroide si osserva dove risiede il picco di energia in ogni banda e dove quindi è la formante. Per ogni vocale è calcolata la distanza del centroide1, centroide2 e centroide3 dal valore teorico della prima, seconda e terza formante rispettivamente. Viene calcolata in seguito una distanza euclidea pesata tra i tre valori di distanza ottenuti, in cui è dato un peso maggiore alla distanza dal centroide1. Tra le cinque distanze euclidee calcolate è individuata quella minima e controllato a quale vocale è legata. Tale vocale corrisponde così con una certa probabilità alla vocale in ingresso all'external.

sFormCENTROID è stato testato allo scopo di valutarne la possibilità di utilizzo per l'applicazione finale e quindi l'affidabilità. A tal fine si è creata una patch di sperimentazione con la quale 10 campioni audio per ognuna delle 5 vocali (non facendo distinzione tra vocali aperte e chiuse), della durata ciascuno di poco più di 1 secondo, vengono dati in ingresso all'oggetto [sformCENTROID]. In uscita vengono considerati i valori di 45 campioni restituiti dall'external ed indicanti l'indice corrispondente alla vocale riconosciuta, poiché si considera una finestra d'analisi per la DFT di 1024 campioni (mantenendo il sample rate di default di Pd a 44100 Hz). Tra questi 45 valori si è osservato quanti indicano l'indice corrispondente alla vocale effettivamente analizzata e ricavato una percentuale. Da tutti i file audio relativi ad una vocale si è calcolata una percentuale media che esprime la correttezza dell'identificazione, ottenendo i risultati riportati in Tabella 2.2.

I risultati ottenuti non si sono dimostrati accettabili. Non c'è alcun riconoscimento per le

Vocale	Percentuale correttezza
a	40,01
e	2,66
i	0
o	11,5
u	0

Tabella 2.2: Correttezza *sFormCENTROID*

vocali *u* e *i* e per le altre vocali i risultati non possono di certo far ritenere la soluzione come impiegabile.

In [19] si descrive un approccio per la classificazione di vocali nel parlato continuo basata sulle prime tre formanti: F1, F2 e F3. Nel paper viene impiegata una tecnica di analisi LPC (Linear Predictive Coding) per determinare le ampiezze e le frequenze delle formanti nel parlato. Le frequenze delle prime tre formanti delle vocali Serbe sono state analizzate simultaneamente. La discriminazione delle vocali basate sulle prime due formanti non è possibile a causa di grandi sovrapposizioni nel piano F1-F2, ma se la terza formante è unita al processo decisionale la sovrapposizione diminuisce. A causa della sovrapposizione delle aree che le vocali occupano nello spazio F1-F2-F3 sono state introdotte delle linee di demarcazione nei piani F1-F2 e F2-F3 per qualche vocale, al fine di evitare sovrapposizioni e avere un errore minore nel riconoscimento delle vocali, ottenendo d'altra parte una riduzione del tasso di riconoscimento corretto. La discriminazione delle vocali basata sulle formanti necessita quindi di compromessi nella soluzione al problema e di altri approcci di analisi.

L'utilizzo del centroide, così come analizzato, non si presta bene all'identificazione del timbro vocale come invece avviene per il timbro musicale.

A seguito di questi risultati lo sviluppo è stato indirizzato verso una nuova soluzione, che prevede l'utilizzo della libreria di external denominata *timbreID*.

2.6.3 Soluzione adottata: *timbreID*

timbreID [21] rappresenta una collezione di external per Pd sviluppata da William Brent. E' composta da un gruppo di oggetti per l'estrazione di feature timbriche e da un oggetto per la classificazione che gestisce il database di informazioni risultante. Ogni oggetto riporta i propri risultati o come un numero singolo o come una lista che può ulteriormente essere manipolata in Pd. Liste di feature di qualsiasi dimensione possono essere raccolte assieme in modo che l'utente

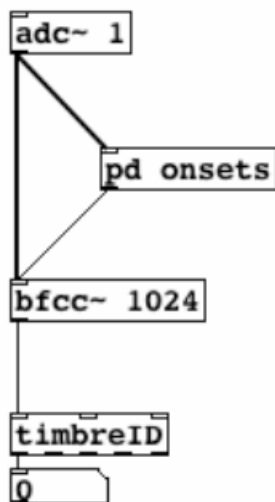


Figura 2.3: *timbreID in una configurazione di training.*

possa definire un approccio personalizzato.

Le feature generate attraverso gli oggetti `external` possono essere usate direttamente come informazioni di controllo in performance real-time.

L'oggetto stesso `[timbreID]` funziona come un meccanismo di storage e routing in grado di clusterizzare ed ordinare le feature che memorizza e classificare nuove feature in relazione al suo database. Questo oggetto rappresenta una delle componenti fondamentali nell'analisi del timbro vocale in SoundRise.

In Figura 2.3 è riportato la rete di base richiesta per fornire a `[timbreID]` le feature di training.

Nella patch `pd onsets` il segnale audio che giunge dall'oggetto `[adc~]` viene trattato da un oggetto specifico (`[bonk~]`) che individua gli “attacchi”, definiti come bruschi cambiamenti nell'involuppo spettrale del suono entrante. Tale oggetto invierà un messaggio di attivazione di eventi (bang) all'oggetto `[bfcc~]`, che fa parte della collezione di `external` e che sarà meglio descritto in seguito. Le feature di training in uscita da `[bfcc~]` vanno in ingresso al primo inlet di `[timbreID]`. Una volta che un database di training è stato accumulato in questa maniera, l'uscita del `[bfcc~]` può essere instradata verso il secondo inlet di `[timbreID]` così che ogni nuova insorgenza di uno strumento genererà una relazione di massima corrispondenza (match) dal primo outlet. Un risultato di match è dato come indice della più vicina istanza di matching nel database, sulla base di quanto ottenuto durante il training. Per ogni match, il secondo outlet riporta la distanza tra la feature in ingresso e il suo match più vicino, e il terzo outlet produce una misura di confidenza basata sul rapporto tra le distanze del primo e del secondo miglior match.

Un numero di cluster desiderato può essere settato attraverso un *messaggio di cluster*. Un algoritmo di clustering raggrupperà le istanze in base alle impostazioni metriche di somiglianza correnti. `[timbreID]` riporterà l'indice di cluster del match più vicino in risposta alle richieste di classificazione.

2.6.3.1 Analisi del cepstrum - Gli oggetti `[mfcc~]` e `[bfcc~]`

In [22] viene valutato l'insieme degli external di *timbreID* per l'analisi del cepstrum in un problema di classificazione di strumenti a percussione, destinato ad un'identificazione del timbro real-time. Questo paper è stato preso in considerazione per valutare quali feature risultano sufficientemente robuste in applicazioni real-time. Oltre infatti al centroide, diverse altre misure vengono utilizzate come feature timbriche nell'area dell'information retrieval in campo musicale, come i coefficienti mel-cepstrali (MFCCs) e i coefficienti bark-cepstrali (BFCCs).

L'identificazione delle vocali è ottenibile tramite l'utilizzo del cepstrum, come proposto dallo stesso autore di *timbreID* in [21]. Il cepstrum di un segnale [23] è definito come la trasformata di Fourier del logaritmo (con fase tra 0 e 2π o tra $-\pi$ e π) della trasformata di Fourier (FT) del segnale. Il cepstrum di un segnale $x(t)$ è dunque:

$$X(T) = FT(\log(FT(x(t)) + j2\pi m)) \quad (2.2)$$

È comune anche relazionare il cepstrum del segnale non alla trasformata di Fourier ma all'inversa della trasformata di Fourier (IFT) del $\log(FT(x(t)))$.

L'analisi cepstrale pura [24] si basa interamente su misurazioni oggettive del suono e non fa uso di scale legate alla percezione umana costruite sulla base di giudizi soggettivi.

Basandosi su una scala di frequenza percettiva, una forma diffusa di analisi del cepstrum per la rilevazione di feature è la misura del cepstrum sulla scala di mel. Una trasformazione non lineare della scala della frequenza è usata per ottenere il corrispondente valore in mel [22] :

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700}) \quad (2.3)$$

dove f è la frequenza in Hz. Per applicare la scala mel al cepstrum, viene utilizzato un banco di filtri triangolari passabanda con frequenza centrale in K valori equispaziati in mel. La larghezza di banda di ciascun filtro è la distanza dalla frequenza centrale del filtro precedente, moltiplicata per due. I coefficienti mel-cepstrali MFCC si ottengono mediante l'equazione:

$$MFCC_1 = \sum_{k=0}^{N-1} X_k \cos\left[i\left(k + \frac{1}{2}\right)\frac{\pi}{N}\right]; \quad i = 0, 1, \dots, M - 1 \quad (2.4)$$

dove M è il numero di coefficienti cepstrali, N il numero di filtri e X_k è il logaritmo della potenza in uscita dal k -esimo filtro. Il cepstrum sulla scala mel, a differenza del cepstrum puro, enfatizza i contenuti a bassa frequenza e fa uso della trasformata del coseno discreta (DCT) piuttosto che di una FT.

Nella scala di bark le frequenze usate per la costruzione del banco di filtri sono generate basandosi su una diversa equazione:

$$Bark = \left[26.81 \times \frac{f}{(1960 + f)} \right] - 0.53 \quad (2.5)$$

Anche il cepstrum sulla scala bark si differenzia dal cepstrum puro per le stesse caratteristiche definite per il cepstrum sulla scala mel. L'oggetto `[bfcc~]` della libreria *timbreID* da in uscita proprio i coefficienti BFCCs, relativi alla finestra di analisi più recente, come lista. Lo stesso compito è svolto da un altro oggetto della libreria, `[mfcc~]`, in merito ai coefficienti MFCCs. La lista può essere inviata all'external `[timbreID]` per l'identificazione del timbro real-time.

2.6.3.2 Identificazione delle vocali

Una patch Pd per l'identificazione delle sole vocali *a*, *e* ed *o*, che impiega l'oggetto `[cepstrum~]` e `[timbreID]`, e fornita assieme agli external di *timbreID*, è stata presa come una buona base per il sistema di riconoscimento del timbro di *SoundRise*. La patch è mostrata in Figura 2.4.

Dal momento che `[bfcc~]` risulta essere più accurato rispetto a `[cepstrum~]` e `[mfcc~]`, come descritto dallo stesso autore della libreria *timbreID* in [22], si è deciso di utilizzare tale external come estrattore delle feature variando così la patch di Figura 2.4. La classificazione è fatta

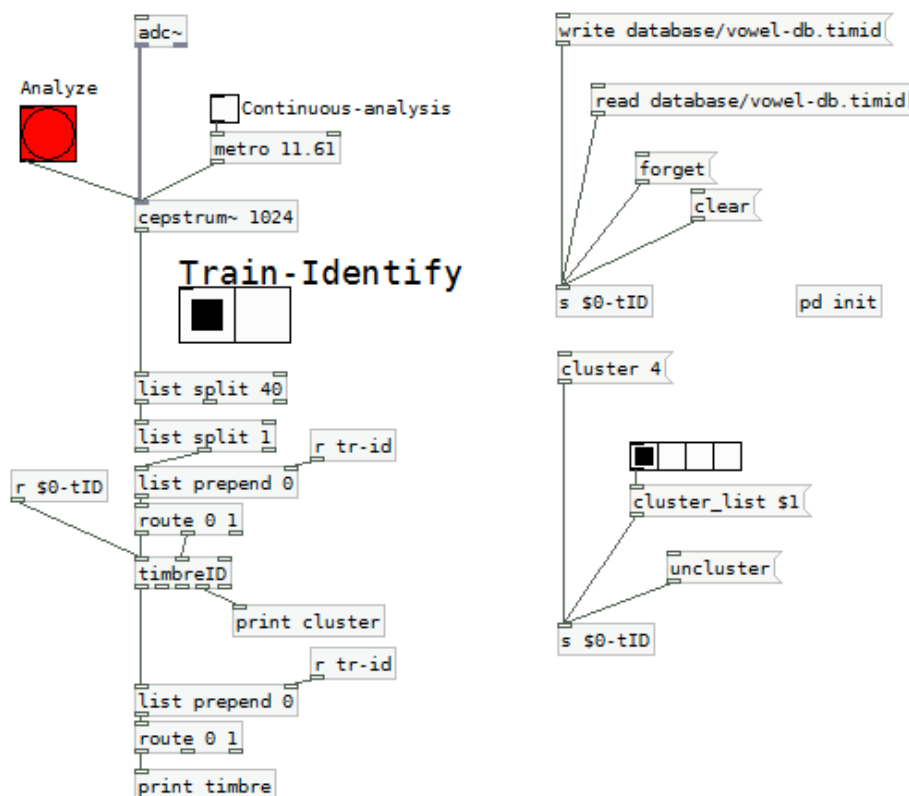


Figura 2.4: Patch di identificazione delle vocali *a*, *e* ed *o*.

interfacciando l'uscita dell'oggetto `[bfcc~]` con l'oggetto `timbreID` che accumula i vettori di feature di training come template e confronta i vettori in arrivo con i dati memorizzati. Una distanza Euclidea è calcolata tra i vettori in arrivo e tutti i template di training. Il template più vicino nello spazio delle feature è preso come il match migliore e il suo indice è inviato in uscita da `timbreID`.

Il database di vettori di feature è stato creato utilizzando per il training delle registrazioni di ogni vocale, fatte a dei bambini delle scuole elementari. Il database è formato così da sei cluster, cinque relativi alle vocali e uno associato al silenzio, ognuno contenente 60 istanze di feature di training. Il funzionamento della patch per l'identificazione delle vocali in *SoundRise* è spiegato in dettaglio nel paragrafo 2.7.

2.7 Overview della patch Pd SoundRise

2.7.1 Dettagli della patch

La patch Pure Data *SoundRise.pd*, è la patch che avviata permette l'utilizzo dell'applicazione. Si compone di diverse subpatch le quali rappresentano uno strumento per annidare delle porzioni di programma in apposite box. Le subpatch appaiono come delle normali box oggetto (*objective box*) formate dal nome pd seguito da un nome identificativo per la subpatch. Possono essere utilizzate solo all'interno della patch dove sono state create o all'interno di altre subpatch della stessa patch. Questa soluzione permette di gestire meglio una patch complessa e articolata come *SoundRise.pd*. Sfruttando una caratteristica molto potente di Pure Data, definita *graph-on-parent*, è stato possibile separare gli oggetti GUI rappresentanti l'interfaccia grafica per l'utilizzo della patch dalle varie subpatch contenenti gli oggetti per l'elaborazione audio e il rendering grafico. L'utente che avvia l'applicazione ha di fronte solamente gli strumenti di cui ha bisogno, come pulsanti e slider, senza essere confuso dai numerosi oggetti impiegati per l'elaborazione e dai collegamenti vari tra gli stessi oggetti. La patch *SounRise.pd* si presenta dunque come un'interfaccia di controllo e solamente se aperta è possibile osservarne il contenuto, come mostrato in Figura 2.5, caratterizzato principalmente da:

- una subpatch *pd settings* contenente la parte di gestione della rappresentazione grafica delle feature che sfrutta la libreria GEM;
- il *core* dell'applicazione, rappresentato invece non da una subpatch ma da quella che è l'unica *abstraction* di *SounRise.pd*, ovvero una normale object box, con il nome identificativo di *SoundRiseCore*, salvata indipendentemente dalla patch *SounRise.pd* che la contiene e perciò riutilizzabile in altre patch e nello sviluppo dell'applicazione per dispositivi mobili [14]. In *SoundRiseCore* si concentrano gli algoritmi di estrazione delle feature della voce.

SounRise.pd contiene a sua volta subpatch che realizzano funzioni più o meno complesse per la gestione delle feature. Tra queste la patch *pd vowel* che costituisce il sistema di identificazione delle vocali.

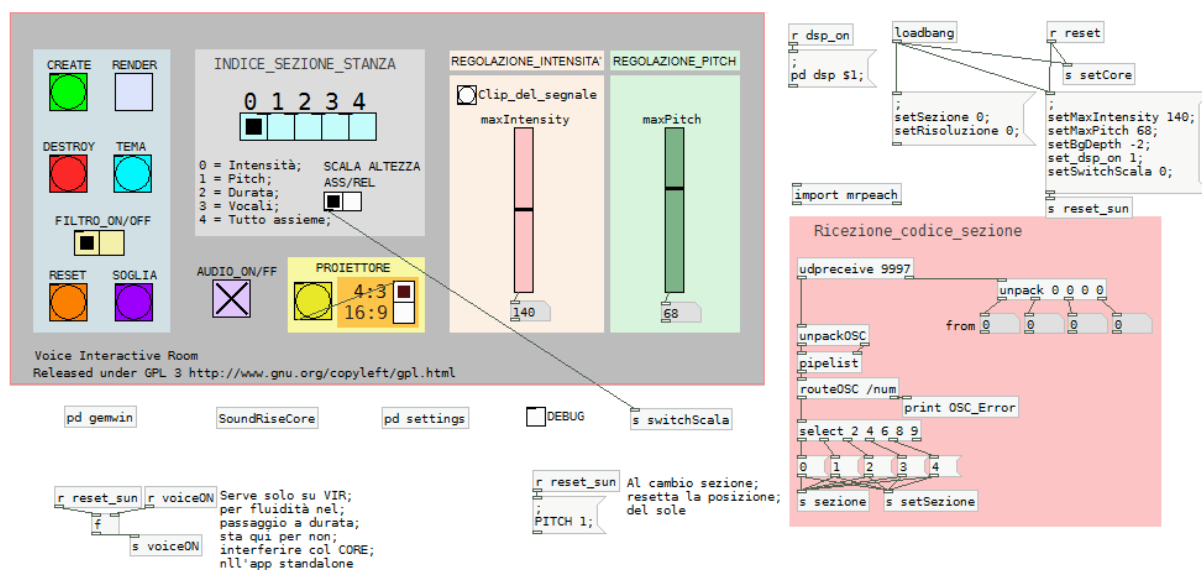


Figura 2.5: Contenuto della patch SoundRise.pd.

2.7.2 SoundRiseCore

Ingresso audio

In Figura 2.6 è mostrata il blocco per la ricezione del segnale vocale. La subpatch *pd audio.in* fornisce l'ingresso audio real-time all'applicazione che è poi inviato tramite l'oggetto `[s~ audio]` ai vari blocchi per l'identificazione delle feature.

Il segnale di ingresso viene filtrato tramite un filtro passa-banda, Figura 2.7 che permette il passaggio di frequenze all'interno dell'intervallo che va da 300 Hz a 5000 Hz poiché in questo range è contenuta l'informazione utile. Il filtraggio rendendo inoltre possibile ottenere un comportamento adeguato anche in ambienti rumorosi. Per tale scopo si è impiegato un banco di quattro filtri Butterworth del terzo ordine per ottenere una risposta piatta nella banda passante, composto da due filtri passa-basso con frequenza di taglio di 300 Hz e due passa-alto con frequenza di taglio di 5000 Hz. L'oggetto Pd `[butterworth3~]` che realizza un filtro Butterworth, ha permesso tale realizzazione.

Patch per il riconoscimento delle vocali: *pd vowel*

La patch per l'identificazione delle vocali si presenta come in Figura 2.8. L'oggetto `[timbreID]` riceve in ingresso i coefficienti dell'external `[bfcc~]`. Basandosi sulla patch di identificazione delle vocali fornita con la libreria *timbreID*, solamente i primi 40 coefficienti, dei 47 disponibili, sono considerati selezionandoli tramite l'oggetto `[list split]`. Il primo coefficiente è

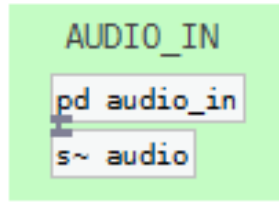


Figura 2.6: *Ingresso audio.*

Vocale	Indice
a	0
e	1
i	2
o	3
u	4

Tabella 2.3: *Associazione vocale-indice identificativo*

scartato in quanto risulta avere sempre valore 1 e di conseguenza non porta informazione utile. Messaggi di cluster vengono utilizzati per la creazione dei cinque cluster, uno per ogni vocale, nel database. Un ulteriore cluster è impiegato per classificare feature diverse da quelle associate alle vocali. Ogni cluster si compone di 60 istanze. Un indice di valore 1, inviato dall'oggetto GUI *Hradio* con etichetta *Train-Identify* all'oggetto *receive* [r tr-id], setta la modalità di identificazione. Tale indice è inserito in testa alla lista di coefficienti BFCC e riconosciuto in ingresso da [timbreID] che avvierà l'identificazione.

Dall'outlet della patch ([outlet]) esce l'indice della vocale identificata. L'associazione tra indice e vocale è riportata in Tabella 2.3.

L'assenza di riconoscimento della vocale è identificata dall'indice 5, l'indice del *silenzio*.

Analisi dell'ampiezza e della durata

Nei blocchi per l'analisi dell'ampiezza e della durata del segnale vocale, Figura 2.9, il valore efficace in dB del segnale in ingresso viene utilizzato per il calcolo della *soglia di rumore*. Il valore di questo limite, contenuto nel parametro dall'oggetto [s setSoglia], va in ingresso all'external *isteresi* settando il limite oltre al quale il segnale è da considerarsi analizzabile e sotto il quale viene invece considerato rumore di fondo. I parametri di default di questo oggetto sono rappresentati dal valore oltre il quale inizia la durata del segnale, pari a 65 dB, e il valore, pari a 60 dB, che definisce la fine della durata una volta che il segnale in ingresso ne risulta inferiore.

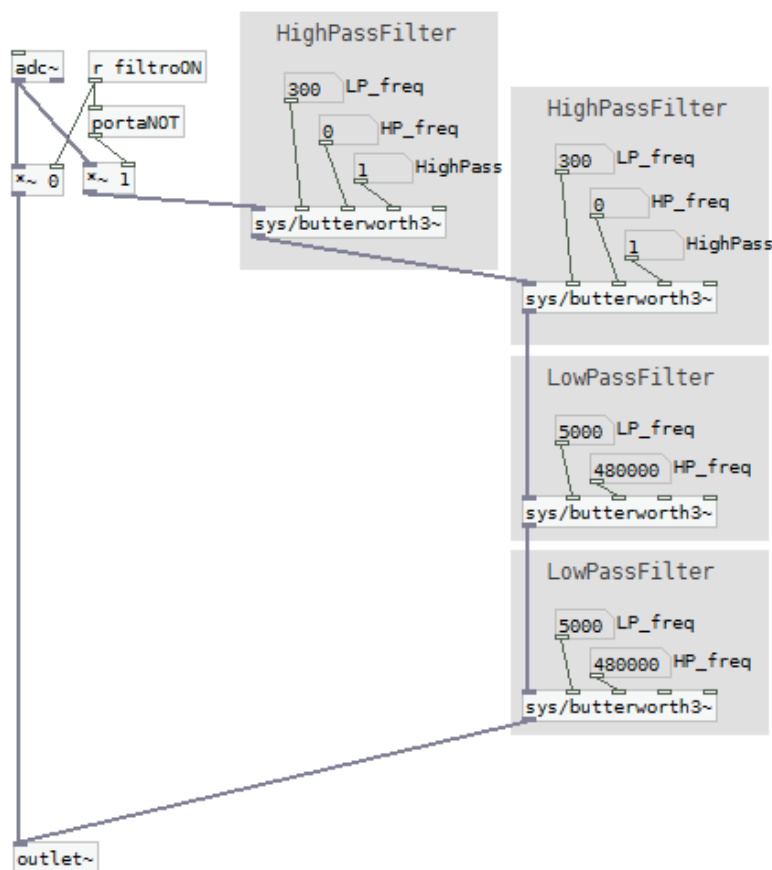


Figura 2.7: Filtraggio del segnale audio di ingresso.

Per l'analisi dell'ampiezza, la subpatch *pd if(in1<>0 in1 in2)* definisce se esiste, dato un ingresso, un valore valido da mappare nel range in cui il sole può muoversi. Il parametro *POWER* dell'oggetto *[s POWER]* contiene il valore della potenza del segnale da mandare alla subpatch per il rendering grafico, mappato nel range in cui il sole può variare la propria dimensione per mezzo della funzione implementata nella patch *pd (in1-in3)*scala/in2*. L'oggetto *[line]* è un generatore di rampa utilizzato per modificare gradualmente la dimensione del sole, impedendo variazioni troppo rapide.

Analisi del pitch

Per l'analisi del pitch la Figura 2.11 mostra l'ingresso del segnale audio entrare nella subpatch *[pd pitcher]* la quale stima il valore del pitch sfruttando l'oggetto Pd *[fiddle~]*, Figura 2.10.

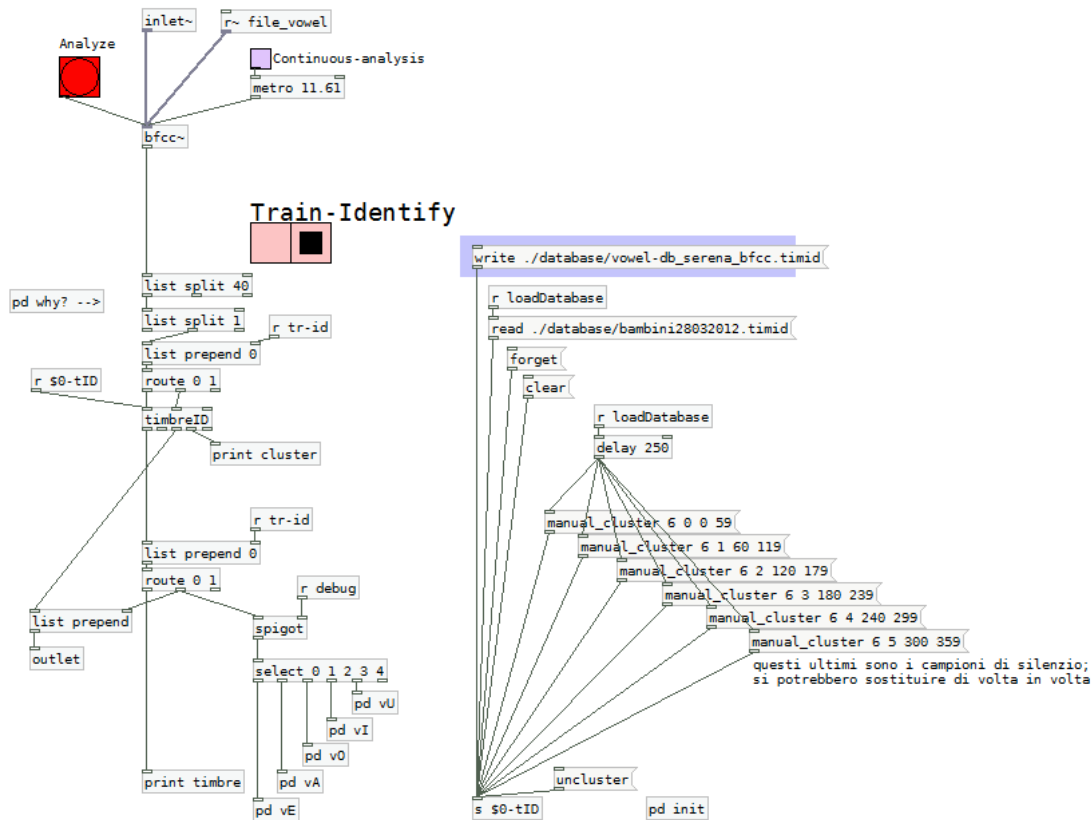


Figura 2.8: Patch per il riconoscimento delle vocali: *pd vowel*.

[pd pitchAssoluto] e [pd pitchRelativo] sono le subpatch per far variare l'altezza del sole rispetto alla scala assoluta e relativa. Anche in questo blocco viene impiegato l'oggetto [line] per dare al sole un movimento fluido. *PITCH* è il valore che è inviato alla subpatch per il rendering grafico.

2.7.3 La patch *pd settings*

pd settings costituisce la subpatch destinata alla rappresentazione grafica delle feature estratte in *SoundRiseCore*. In Figura 2.12 è mostrata una parte della patch, dove si notano i vari oggetti Pd, e il modo in cui sono connessi, per i blocchi associati alla rappresentazione delle vocali, dell'intensità e del pitch.

Per la colorazione del sole in base all'indice di vocale, ricevuto dall'oggetto [r vocale], viene sfruttato l'oggetto *select* per associare a tale indice un valore nello spazio dei colori HSV

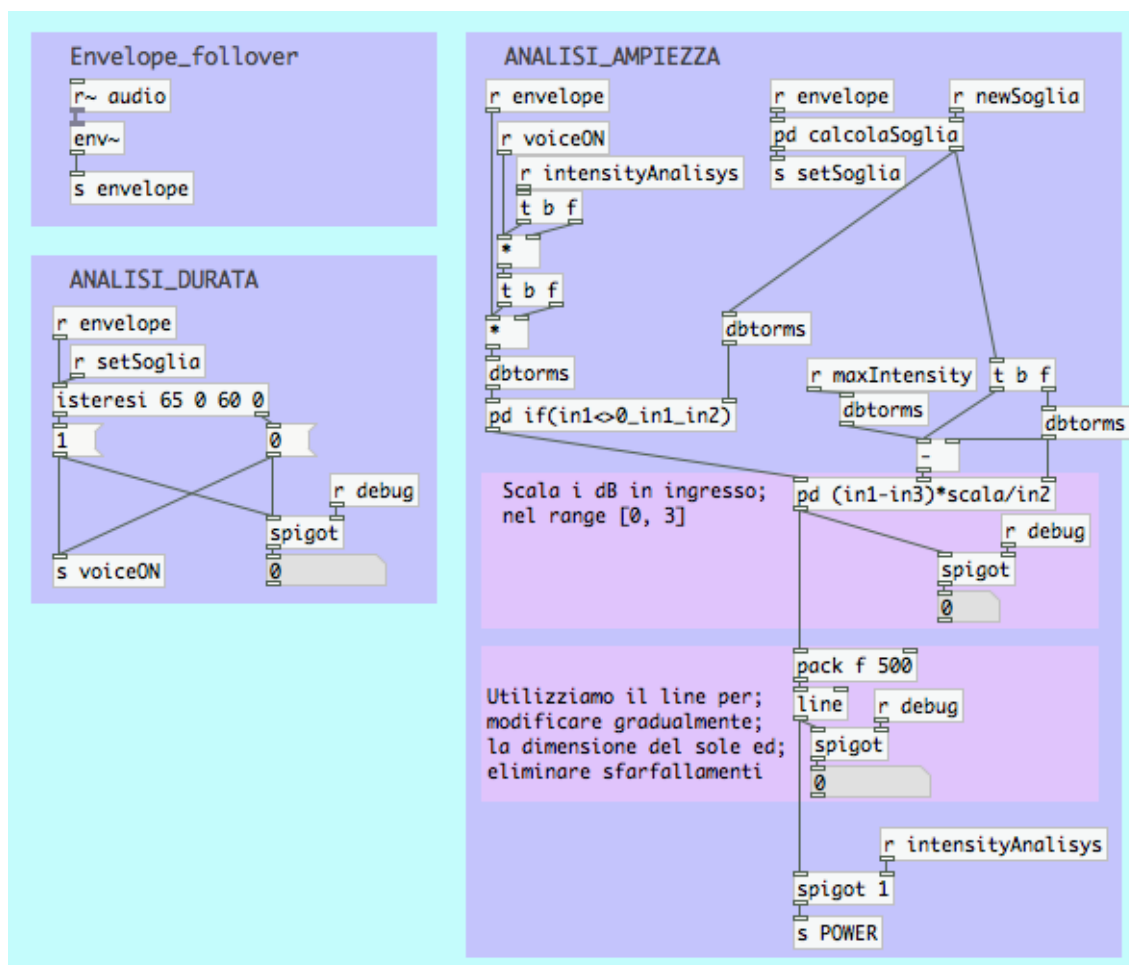


Figura 2.9: Blocco relativo all'analisi dell'ampiezza e della durata del segnale vocale.

(Hue Saturation Value). Il cambio di colore dovuto ad un cambio di vocale viene reso gradualmente tramite un'apposita subpatch creata per tale scopo e denominata *slow_output-IO*. Questa fa sì che non ci sia un cambiamento immediato nel colore del sole e che quindi non si crei un effetto lampeggiante in caso di risultato dell'identificazione delle vocali molto variabile, che risulterebbe molto fastidioso. D'altra parte, con la soluzione realizzata, si ha che un passaggio da un valore HSV, V1, ad un altro, V2, include la visualizzazione, della durata massima di circa 2 secondi, dei colori associati a tutti i valori interi intermedi inclusi nel range [V1, V2]. L'effetto passaggio tra un colore e l'altro è stato di gran lunga migliorato nella versione per dispositivi che sfrutta la versione embedded di Pure Data (*libpd*) [14].

Il valore HSV è convertito dall'oggetto `[hsv2rgb]` nel corrispondente valore RGB (Red Green

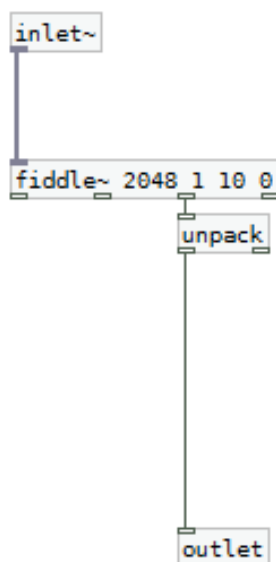


Figura 2.10: Estrazione del pitch del segnale audio in ingresso

Blue) che, associato ad un valore *alpha* di trasparenza settato ad 1, va al secondo inlet di un oggetto [color]. Quest'ultimo riceve come altro ingresso, al primo inlet, una gemlist ottenuta da [pix_image].

[gemhead] abilita il rendering dell'oggetto da disegnare connettendolo al windows manager. [pix_image] permette di caricare l'immagine del sole in modo che possa essere utilizzata come texture da applicare all'oggetto GEM [circle]. Il texture mapping è permesso da [pix_texture]. La parte relativa al rendering grafico del pitch prevede un oggetto [traslateXYZ] che consente la manipolazione degli oggetti GEM. Permette quindi di spostare il sole lungo l'asse Y della finestra grafica. L'intensità è visualizzata graficamente fornendo il valore POWER al secondo inlet di [circle] tramite cui si cambia la dimensione del cerchio.

La parte di rendering grafico associata alla durata è mostrata in Figura 2.13. [r_voiceON] ed [r_dutationAnalysis] permettono l'attivazione o disattivazione del rendering grafico riguardante l'immagine del volto del sole con gli occhi e la bocca chiusi o aperti.

Le immagini dei paesaggi da visualizzare, come ambiente in cui il sole è inserito, vengono caricate e visualizzate nella finestra grafica sfruttando gli stessi oggetti Pd impiegati per l'immagine del sole, come si vede in Figura 2.14. L'oggetto GEM utilizzato è un rettangolo, [rectangle], sul quale è applicata una texture dell'immagine del paesaggio. Per [rectangle] è impostata una larghezza ed un'ampiezza e la posizione lungo l'asse Z in modo che il paesaggio stia dietro

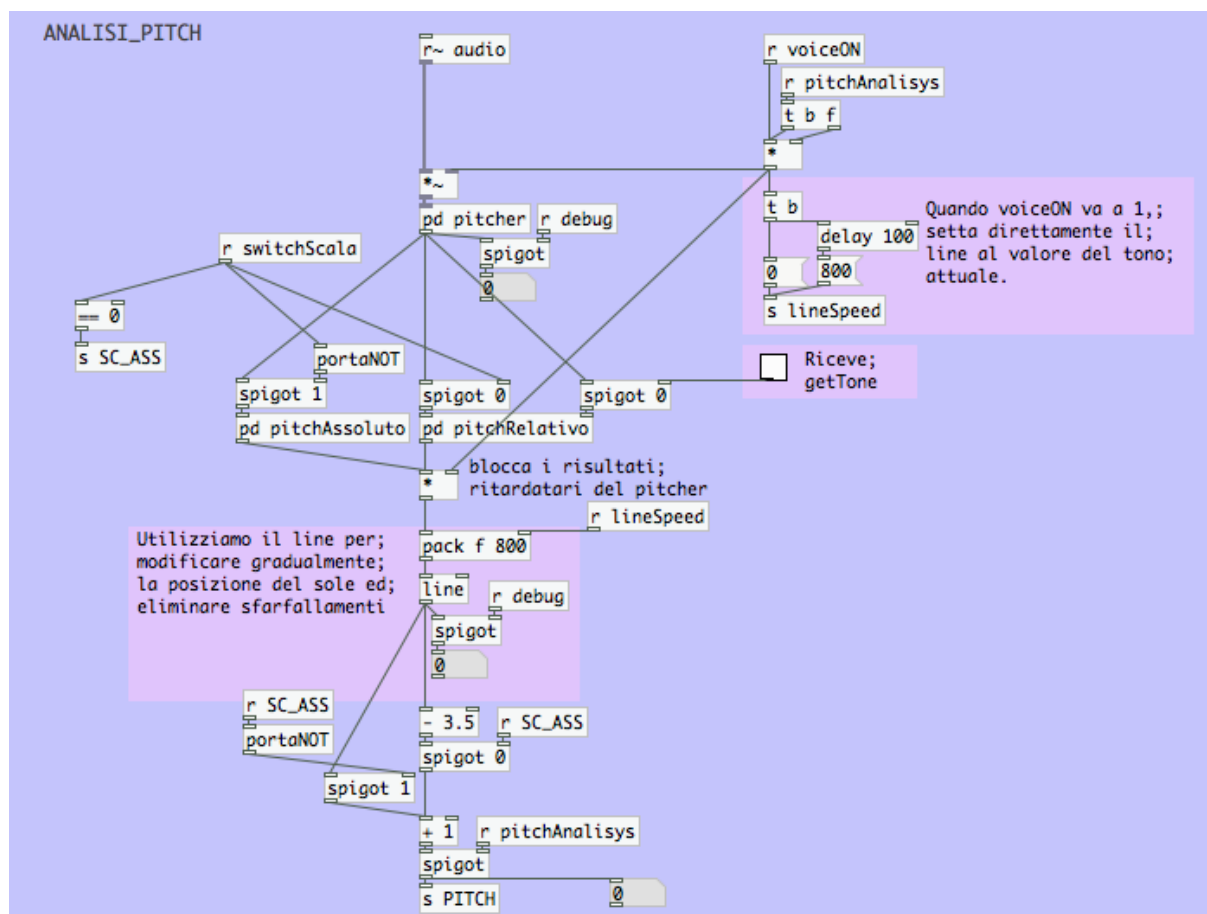


Figura 2.11: Blocco relativo all'analisi dell'altezza del segnale vocale.

il sole.

2.8 Interazione con l'utente

L'utente che utilizza *SoundRise* dal punto di vista hardware interagisce con l'applicazione attraverso l'utilizzo di un mouse e di un microfono, che può essere esterno o integrato al computer in cui l'applicazione viene eseguita. L'interfaccia grafica che rappresenta la console di controllo costituisce il punto di accesso alle varie funzionalità. Attraverso un monitor, un proiettore o una lavagna interattiva si visualizza la finestra grafica in cui i contenuti, animati e non, vengono rappresentati per dare un feedback alle azioni dell'utente. I pulsanti sono realizzati sfruttando oggetti GUI di Pure Data come bang, radio, slider e toggle i quali offrono una risposta visiva

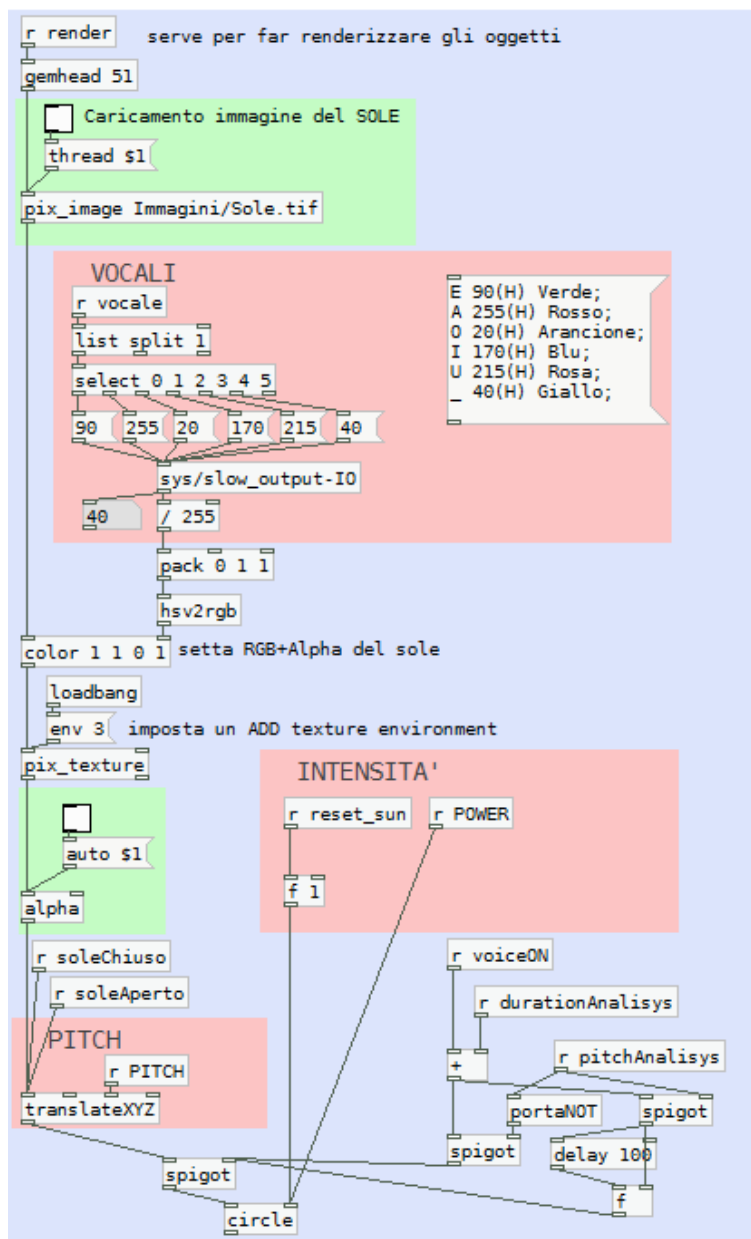


Figura 2.12: Blocco relativo alla rappresentazione grafica delle vocali, dell'intensità e del pitch.

al loro azionamento e pressione. Alla pressione di un pulsante della console, oltre alla risposta visiva data dallo stesso, si verifica un'azione nella finestra grafica aperta.

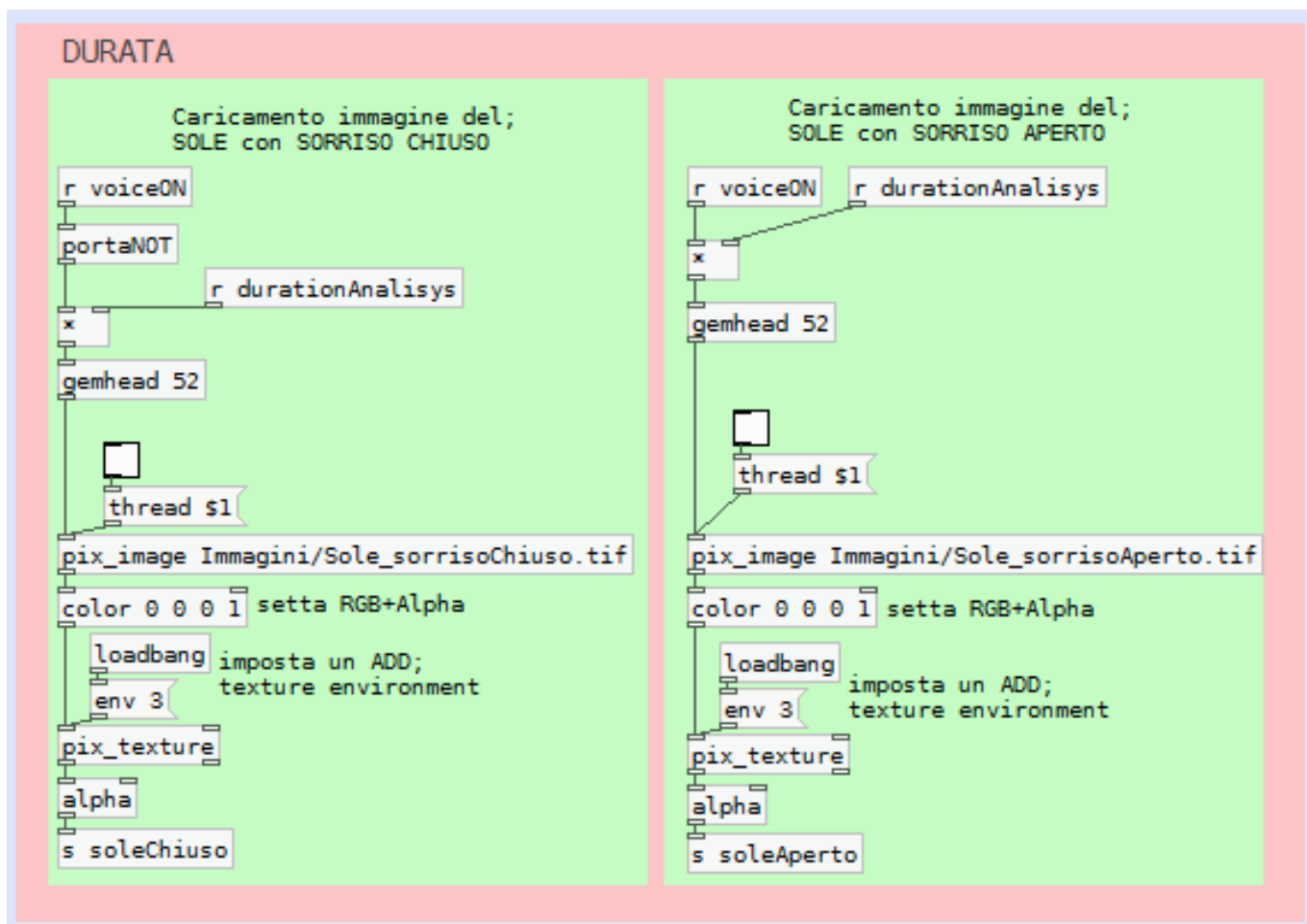


Figura 2.13: Blocco relativo alla rappresentazione grafica della durata.

2.8.1 Interfaccia grafica

L'interfaccia grafica che si presenta all'utente all'avvio di *SoundRise* è quella riportata in Figura 2.15. La console mostrata presenta i seguenti pulsanti:

- CREATE: per creare la finestra grafica in cui apparirà il sole, inserito in un paesaggio di default;
- DESTROY: chiude la finestra grafica;
- TEMA: permette di selezionare un immagine di sfondo per la finestra grafica. La scelta interessa quattro immagini di paesaggi che raffigurano la montagna, il mare, un deserto e una città;

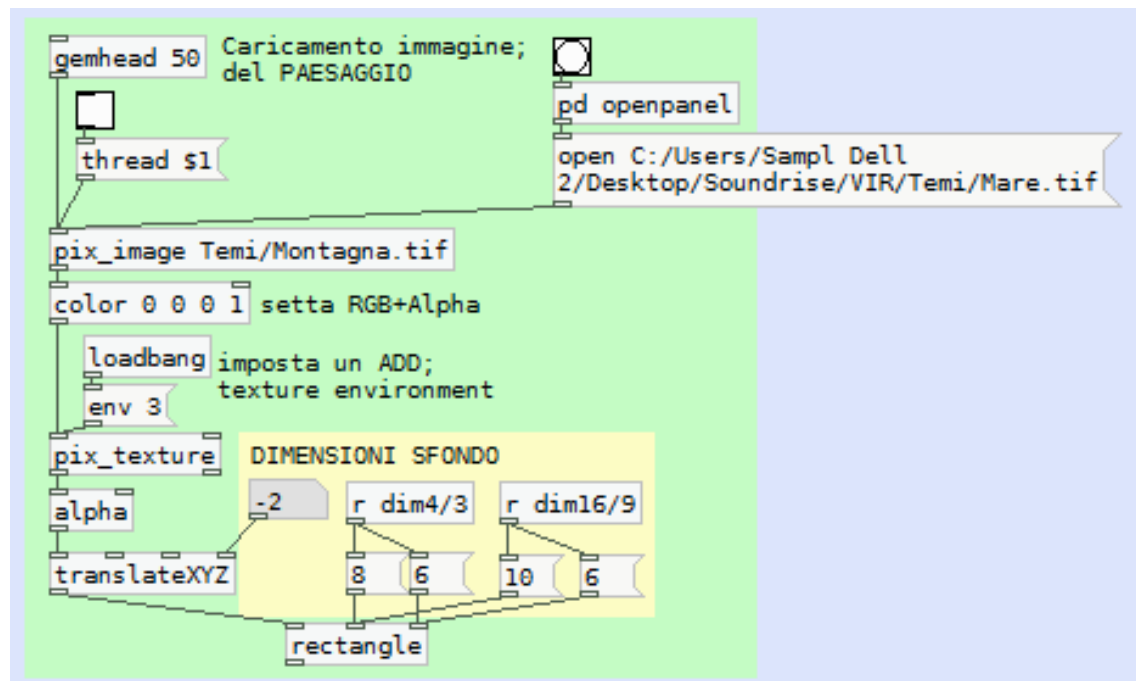


Figura 2.14: Sezione relativa al caricamento dell'immagine di sfondo della finestra grafica .

- **RESET:** consente di resettare al valore di default parametri variabili quali la massima dimensione e altezza che il sole può raggiungere ed impostare la scala del pitch al valore di default che è quello della scala assoluta;
- **SOGLIA:** ricalcola il valore della *soglia di rumore* (di fondoscala);
- **INDICE_SEZIONE_STANZA:** rappresenta cinque pulsanti, ognuno con un indice di riferimento corrispondente ad una modalità:
 - 0: attiva la modalità analisi dell'intensità
 - 1: attiva la modalità analisi del pitch (altezza)
 - 2: attiva la modalità della durata
 - 3: attiva la modalità analisi del timbro
 - 4: attiva tutte le modalità di analisi contemporaneamente
- **ASS/REL:** permette di scegliere tra scala assoluta e relativa per le modalità 1 e 4;
- **AUDIO_ON/OFF:** attiva e disattiva l'analisi audio;

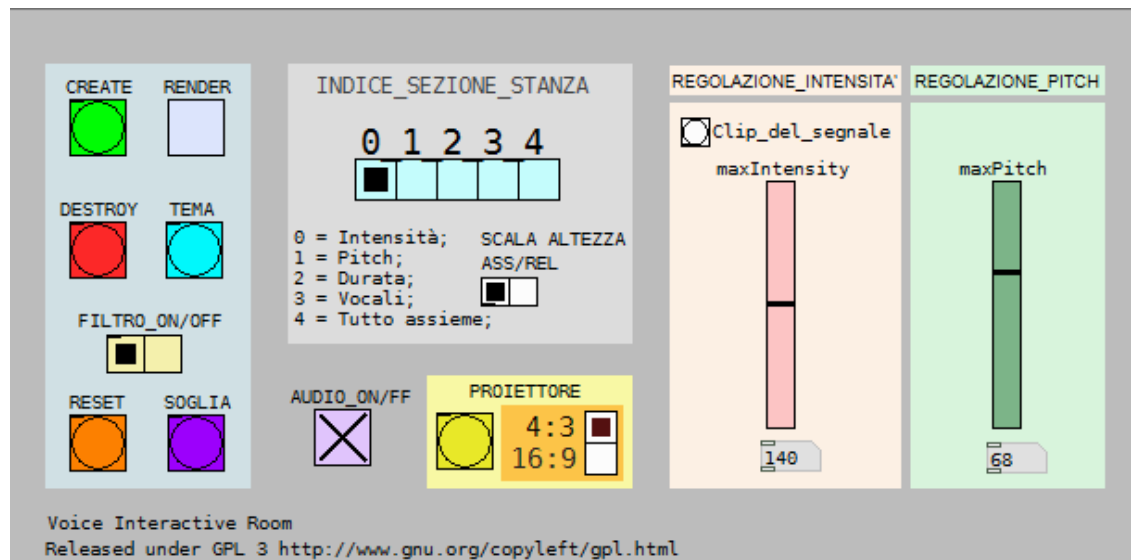


Figura 2.15: Console di controllo di SoundRise .

- **PROIETTORE:** in caso di utilizzo di un proiettore a cui il computer è connesso o lavagna interattiva, visualizza la finestra grafica in risoluzione 4:3 o 16:9;
- **REGOLAZIONE_INTENSITÀ:** consente all'utente di regolare l'ampiezza massima che il sole può raggiungere in modo da adattare il feedback visivo al tono medio della propria voce;
- **REGOLAZIONE_PITCH:** consente all'utente di regolare l'altezza massima che il sole può raggiungere in modo da adattare il feedback visivo al tono medio della propria voce.

2.8.2 Rappresentazione grafica

Il rendering grafico è stato realizzato utilizzando la libreria GEM (Graphical environment for Multimedia), che ben si integra nella piattaforma Pure Data. Sono state utilizzate caratteristiche grafiche ben riconoscibili come dimensione, colore e geometria con l'intento di ottenere un'efficace rappresentazione delle feature estratte.

Il feedback risultante dalle azioni dell'utente è strettamente legato alla rappresentazione dell'informazione in quanto il suo contenuto deve essere comunicato in una certa forma, ma la forma ha anche un valore informativo [11].

I contenuti di comunicazione dell'applicazione prevedono immagini, interessate anche da animazioni, realizzate con un disegno semplice e pulito in modo da non distrarre troppo l'attenzione del bambino da quello che è lo scopo didattico.

Il movimento [11] ha un valore informativo univoco in video, animazioni e simulazioni. Rende possibile dimostrare condizioni e relazioni complesse tra azioni ed eventi. I movimenti attirano l'attenzione. Una parte importante nella selezione dei contenuti multimediali consiste nell'analizzare i compiti di apprendimento in modo che ci sia una corrispondenza tra la forma di presentazione e i contenuti didattici. Quando la specifica dei contenuti è stata prodotta si procede alla selezione e definizione dei contenuti multimediali. In seguito vengono stabiliti modi per dirigere l'attenzione dell'utente verso l'estrazione delle informazioni richieste da una data presentazione e per indirizzare la concentrazione sul corretto livello di dettagli. La scelta dei contenuti di comunicazione deve accompagnarsi bene con il tipo di informazioni. I contenuti devono combinarsi in modo adeguato con gli aspetti statici o dinamici dell'informazione. Descrizioni di funzionalità dinamiche o processi necessitano di essere rappresentate dinamicamente, con animazioni. In modo simile, un contenuto informativo statico viene rappresentato con contenuti statici, come immagini e testo.

Il sistema *SoundRise* risponde alle emissioni vocali ed enunciati del bambino con animazioni che interessano un unico soggetto che è il sole, il quale risulta controllabile con la voce in tempo reale. Il sole è inserito in un paesaggio che può variare ed è animato nello spazio delimitato dalla porzione di cielo. Esempi di schermate sono mostrati in Figura 2.16. Per le modalità intensità, durata e vocali il sole si presenta al centro del cielo.

Le immagini dei paesaggi disponibili presentano ognuna, nella zona sinistra, un elemento caratterizzante che ha anche lo scopo di dare un riferimento all'utente, in particolare per la comprensione del movimento di salita e discesa del sole e di aumento e riduzione delle sue dimensioni. Tale elemento è rappresentato da un faro, da una piramide, da un grattacielo e da un monte rispettivamente nelle immagini dei paesaggi che raffigurano il mare, il deserto, la città e la montagna.

Per quanto riguarda la modalità di analisi dell'intensità, dell'ampiezza, del timbro e di tutte le quattro feature assieme, in presenza di silenzio il sole non compare a video e dunque il bambino può osservare solamente un paesaggio con un cielo privo della presenza del sole. All'emissione di un qualche enunciato da parte dell'utente il sole appare. Se l'intensità è scelta come parametro da analizzare la dimensione del sole varierà in relazione al valore di tale feature e dunque con un suono forte il sole risulterà molto grande mentre con un suono debole la sua dimensione risulterà

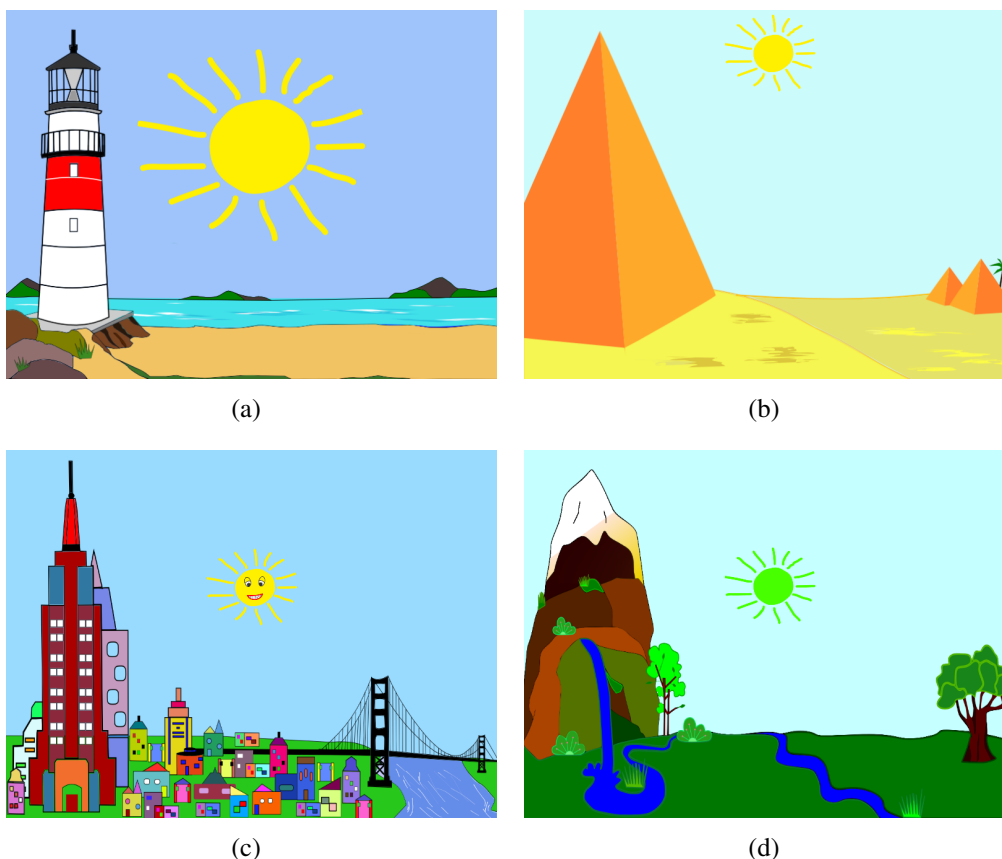


Figura 2.16: Esempio di schermate della finestra grafica: (a) modalità analisi intensità, (b) modalità analisi pitch, (c) modalità analisi durata, (d) modalità analisi timbro

invece piccola. In corrispondenza della selezione dell'altezza, il sole, se la scala scelta è assoluta, dopo essere apparso nella parte bassa del cielo a seguito dell'emissione di un segnale vocale, inizierà a salire raggiungendo la massima altezza in corrispondenza di un tono molto acuto. Se il suono è grave il sole rimarrà in un punto nella parte bassa del cielo. Se invece la scala scelta è quella relativa il sole apparirà al centro del cielo all'emissione di un suono e si sposterà verso l'alto o verso il basso in corrispondenza di toni più alti o più bassi rispettivamente, rispetto ad un tono medio di default o ad il tono medio dell'utente precedentemente acquisito attraverso una funzionalità non implementata però per essere a disposizione nella console di interfacciamento. In corrispondenza della scelta della durata come feature da visualizzare, il sole è sempre presente ed in presenza di silenzio mostrerà un volto con occhi e bocca chiusi. Non appena un suono è emesso, aprirà gli occhi e la bocca mostrando questo aspetto per tutta la durata del suono, per

Vocale	Colore
a	Rosso
e	Verde
i	Blu
o	Arancione
u	Grigio

Tabella 2.4: *Associazione vocale-colore*

poi richiudere occhi e bocca quando l'utente smette di produrre suono.

In merito all'identificazione delle vocali, il timbro suggerisce numerose analogie con il colore per quanto riguarda la percezione visiva. Considerando uno studio [25] in cui viene mostrato che le associazioni significative tra le lettere e i colori esistono, sia in soggetti sinestetici che non sinestetici, sono stati presi come riferimento i risultati della ricerca sulle associazioni tra lettere e colori proposti per rappresentare ognuna delle cinque vocali con un colore, nonostante i dati mostrino che il colore di grafemi può essere fortemente influenzato da pregiudizi ambientali includendo i nomi che usiamo per indicare le categorie di colore e nonostante il linguaggio influenzi l'assegnazione dei colori. L'associazione vocale colore scelta è quella riportata nella Tabella 2.4.

La rappresentazione grafica consiste dunque in un cambio di colore del sole, che di default ha colore giallo, a seconda della vocale pronunciata.

2.9 Impiego di SoundRise

Il dialogo tra utente e computer è costituito dall'interfaccia utente ed è una parte essenziale dell'interazione. L'interfaccia è la parte del sistema con cui l'utente viene percettivamente, cognitivamente e fisicamente in contatto. L'interfaccia utente in *SoundRise* include le caratteristiche software, la grafica e l'organizzazione strutturale del programma e dal punto di vista hardware mouse o touchscreen. Il successo dell'apprendimento di un certo task [11] è molto legato alla strategia di istruzione scelta, di conseguenza l'interfaccia utente può giocare un ruolo importante nel successo dell'apprendimento sostenendo o inibendo certe strategie. L'interazione con l'interfaccia grafica tramite manipolazione diretta implica un continuo feedback visivo sulla posizione e stato degli oggetti. Questo feedback contiene informazioni che generano continue riflessioni sull'azione in corso.

L'utente di *SoundRise* ha un pieno controllo sulla selezione delle feature tramite l'interfaccia grafica grazie anche a pulsanti disposti in modo ordinato ed etichettati da parole chiave che aiutano ad intuirne l'utilità. L'azionamento di un pulsante può avere un esito immediato, come nel caso di cambio del paesaggio in cui il sole si trova o di creazione della stessa finestra grafica, oppure un esito che dipende da un intervento dell'utente. Quest'ultimo è il caso dell'analisi delle feature vocali in cui sperimentando con la propria voce l'utente riesce a vedere ed in seguito a capire quello che succede nelle animazioni della finestra grafica.

2.9.1 Modalità di utilizzo

2.9.1.1 La Stanza Logo-Motoria

La *stanza Logo-Motoria* presenta un sistema real-time che analizza il movimento del corpo e la mimica del bambino all'interno di un ambiente sensorizzato e mappa il tutto in una manipolazione ed elaborazione real-time di contenuti audiovisivi. L'architettura del sistema consiste di tre componenti principali:

- Il *componente di input*, il quale riceve lo stream video catturato da una o più videocamere osservanti lo spazio e le informazioni raccolte da eventuali ulteriori sensori ambientali.
- Il *componente di estrazione delle feature*, che analizza i dati in ingresso allo scopo di ottenere informazioni riguardo a come l'utente occupa lo spazio (dove va, quanto a lungo rimane in una certa zona) e l'espressività dei gesti.
- Il *componente per l'elaborazione real-time di contenuti audiovisivi*, il quale è responsabile del controllo e dell'elaborazione real-time del materiale audio e video e dipende dalle feature estratte dal componente di estrazione delle feature.

La *Stanza Logo-Motoria* può essere sfruttata da parte degli insegnanti sia per trasmettere contenuti per mezzo di un metodo alternativo sia per verificare il livello di conoscenza nei bambini che esprimono al meglio le proprie capacità utilizzando l'intelligenza visiva, spaziale o del corpo.

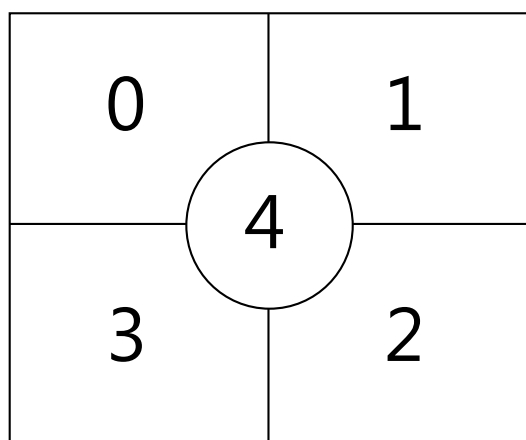


Figura 2.17: Indice delle zone in cui è divisa la stanza analizzata tramite il sistema Stanza Logo-Motoria.

2.9.1.2 Integrazione con la Stanza Logo-Motoria

Oltre all'utilizzo che vede l'utente interagire con l'applicazione rimanendo fermo di fronte ad un microfono, osservando gli effetti della propria produzione vocale a video, la modalità di utilizzo di *SoundRise* in accoppiata con il sistema *Stanza Logo-Motoria* permette di variare il modo di interagire con il sistema. L'utente infatti spostandosi all'interno di zone prefissate nella stanza in cui è installato il sistema è in grado di selezionare quale feature far estrarre, e quindi rappresentare graficamente, a *SoundRise*. La stanza è stata divisa in cinque zone come mostrato in Figura 2.17. L'indice all'interno della zona identifica l'indice del pulsante, nella console, di selezione delle modalità di estrazione che quella specifica zona permette di attivare semplicemente entrandovi.

La comunicazione tra la *Stanza Logo-Motoria* e *SoundRise* avviene tramite comunicazione con protocollo OSC (Open Sound Control) tra la patch EyesWeb e la patch Pure Data. L'invio dell'indice della zona occupata nella stanza da parte della patch EyesWeb della *Stanza Logo-Motoria* è realizzato tramite i blocchi mostrati in Figura 2.18.

Il pattern di indirizzo è una stringa che inizia con uno slash (/). L'argomento è l'effettivo contenuto del messaggio OSC, che è un intero tra compreso tra 0 e 4. La ricezione del messaggio contenente il numero della zona è gestita tramite la porzione della patch pd *SoundRise.pd*.

La gestione dei messaggi tramite OSC è consentito dall'utilizzo dell'estensione *OSC* di *MrPeach* in Pure Data.

2.9.1.3 Utilizzo dell'applicazione

La stanza logomotoria permette di estendere l'interattività multimodale del sistema *SoundRise*. L'utilizzo prevede la presenza della patch EyesWeb della *Stanza Logo-Motoria* e di quella Pd di *SoundRise* nello stesso computer. Le due patch devono essere entrambe in esecuzione. L'ambiente in cui il sistema è utilizzato, Figura 2.20, prevede una webcam, montata sul soffitto, connessa al computer e un tappeto verde che ricopre il pavimento utile a visualizzare la presenza dell'utente all'interno della stanza attraverso la webcam. Quando l'utente entra nel campo visivo della webcam entra anche nella zona formata dall'unione delle cinque sezioni (Figura 2.17). Trovandosi all'interno di una zona, attiva la modalità di analisi della feature corrispondente a quella zona e può usare la propria voce per visualizzare a video il feedback grafico di *SoundRise*. Il microfono utilizzato per catturare il suono è quello della webcam. Altri microfoni possono ovviamente essere utilizzati.

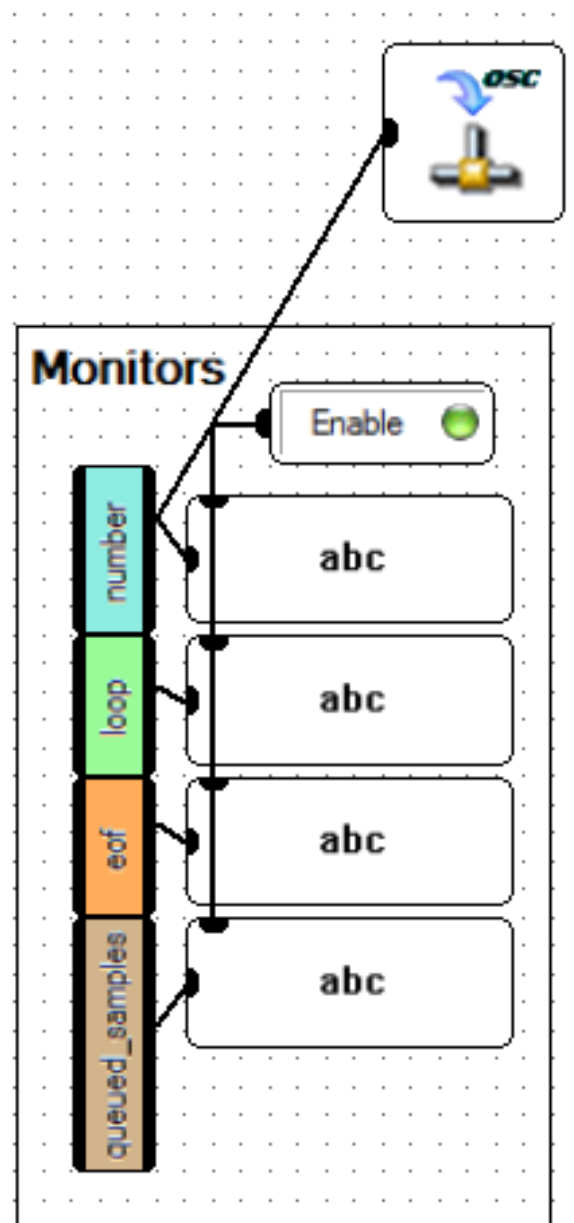


Figura 2.18: Blocchi della patch EyesWeb per l'invio dell'indice della zona della stanza.

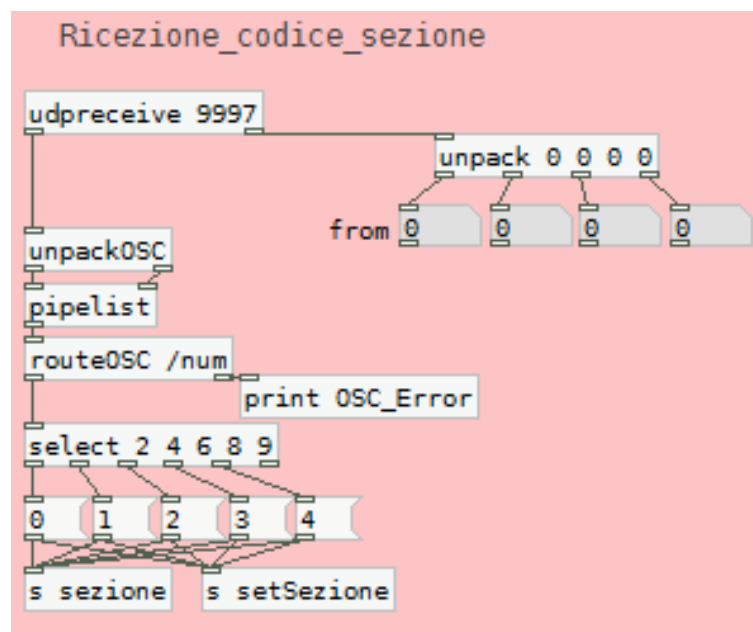


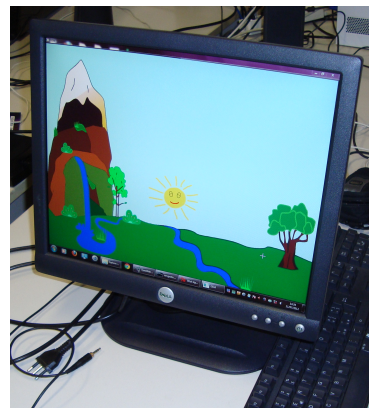
Figura 2.19: Oggetti della patch SoundRise.pd per la ricezione dell'indice della zona della stanza.



(a)



(b)



(c)

Figura 2.20: (a) Foto dell'ambiente in cui vengono integrate la Stanza Logo-Motoria e SoundRise (b) webcam per la cattura della posizione dell'utente (c) SoundRise in esecuzione su un PC.

Capitolo 3

Valutazioni

3.1 Analisi di *timbreID*

Come descritto in [21], dagli oggetti della libreria *timbreID* è stato ricavato un utile sistema di identificazione delle vocali utilizzato per SoundRise. L'utilizzo dell'external [bfcc~] nella patch *pd vowel* di *SoundRise* è giustificato dalla sua efficacia nell'identificazione del timbro e dalla robustezza nelle applicazioni real-time, come riportato in [22]. All'impiego dell'external nello sviluppo dell'applicazione è seguito tuttavia un test di analisi, che ha avuto come obiettivo quello di verificare comunque che quando cepstrum è ponderato sulla base di scale percettive, come la scala mel e la scala bark, funziona in modo efficace come descrittore di feature timbrica. In particolare che il cepstrum sulle frequenze di bar costituisce la feature più efficiente nell'identificazione del timbro. Questo test è stato effettuato dal momento che le gli oggetti della libreria *timbreID* sono valutati in [22] basandosi su un problema di classificazione di strumenti a percussione e di conseguenza i dati raccolti fanno riferimento a tale tipo di classificazione. Con il test si è cercato di verificare se risultati simili possono riscontrarsi anche nella classificazione delle vocali.

La patch di identificazione delle vocali è stata così impiegata per un confronto tra external. I dati osservati per il confronto sono i seguenti:

- indice del cluster che identifica la vocale riconosciuta, a cui è associato il match più vicino identificato (prima uscita di [timbreID]);
- distanza tra la feature in ingresso e il suo match più vicino (seconda uscita di [timbreID]).

Una prima parte del test ha coinvolto [bfcc~] e [mfcc~] ed in particolare si è considerato:

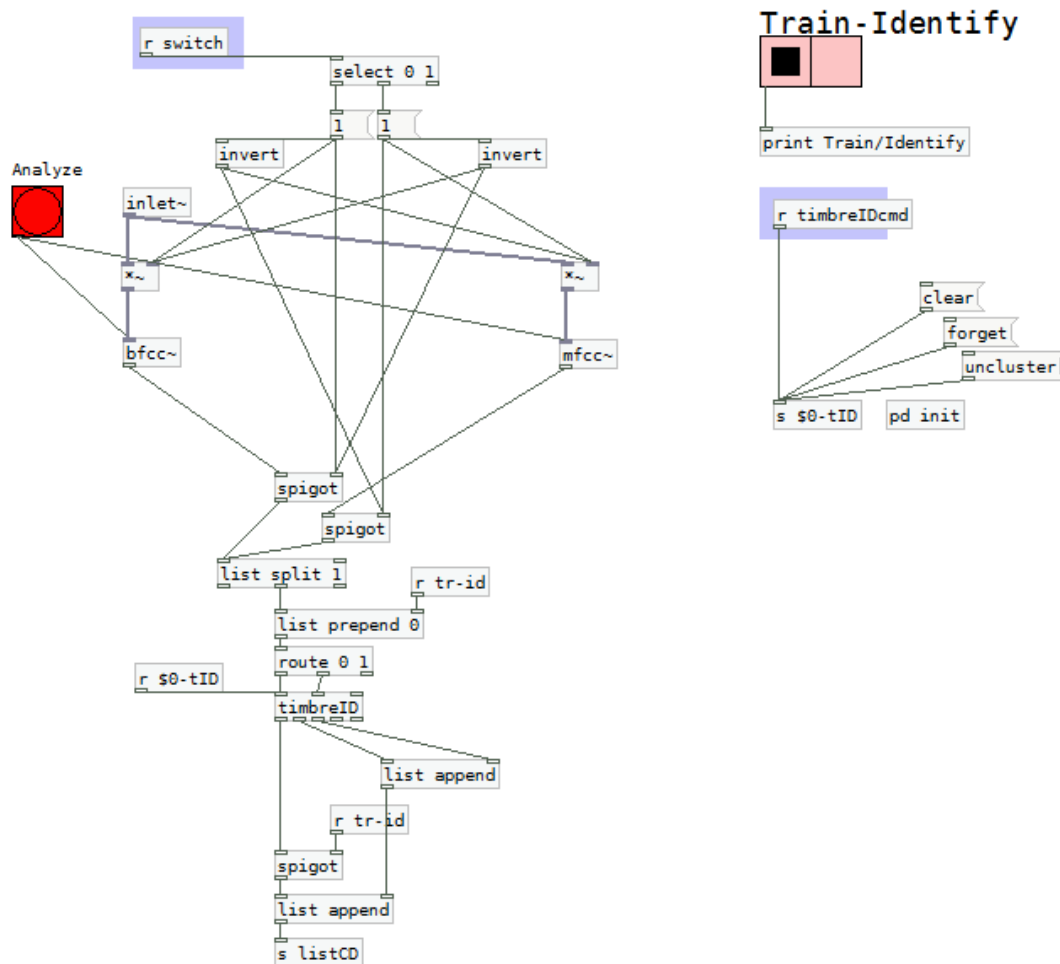


Figura 3.1: Patch per la valutazione degli external *[bfcc~]* e *[mfcc~]*

- un confronto tra i risultati ottenuti con gli external *[bfcc~]* e *[mfcc~]*
- un confronto dei risultati ottenuti prendendo i primi 40 coefficienti in uscita da *[bfcc~]*, come fatto nella patch *pd vowel* di *SoundRise*, con quelli ottenuti considerando tutti i coefficienti messi a disposizione dall'external.

La patch utilizzata per il confronto tra *[bfcc~]* e *[mfcc~]* è mostrata in Figura 3.1.

I risultati ottenuti da tale configurazione sono mostrati nelle Tabella 3.1 e 3.2.

La patch in cui viene considerato un diverso numero di coefficienti dell'external *[bfcc~]* per l'identificazione è quella di Figura 3.2.

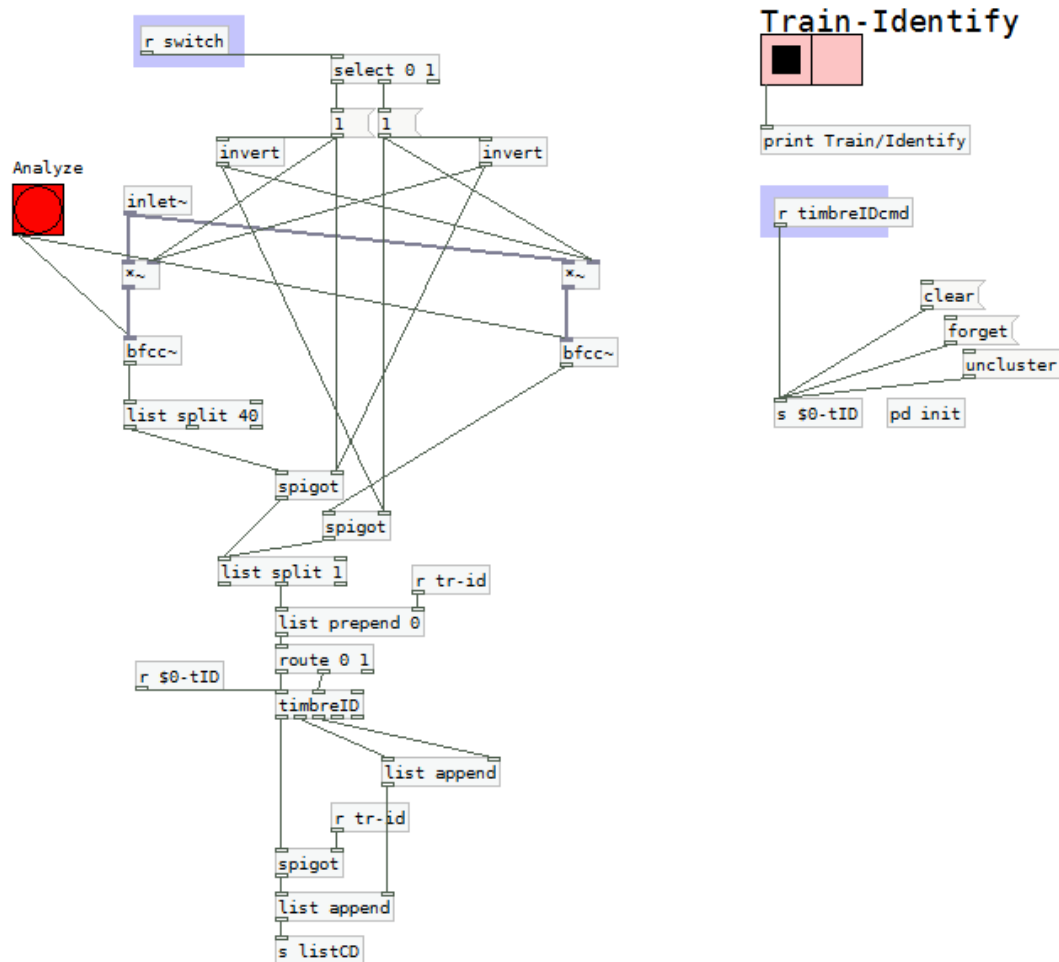


Figura 3.2: Patch per la valutazione degli external `[bfcc~]` e `[mfcc~]`.

I risultati forniti da quest'ultima patch sono invece riportati in Tabella 3.3.

La seconda parte del test ha invece visto l'utilizzo di altri external per l'estrazione di feature timbriche ed in particolare l'external `[specCentroid~]`, che riporta la frequenza associata al centro di massa dell'ampiezza spettrale, e l'external `[specRolloff~]` che considera la frequenza sotto la quale una certa quantità di energia spettrale è concentrata. Il valore di default della concentrazione di energia è pari a 0.85 (85 % dell'energia spettrale totale). Si sono considerati questi due external in quanto in [22] offrono individualmente dei risultati accettabili nella classificazione, anche se non equiparabili con quelli ottenuti con `[bfcc~]`. Ognuno dei due

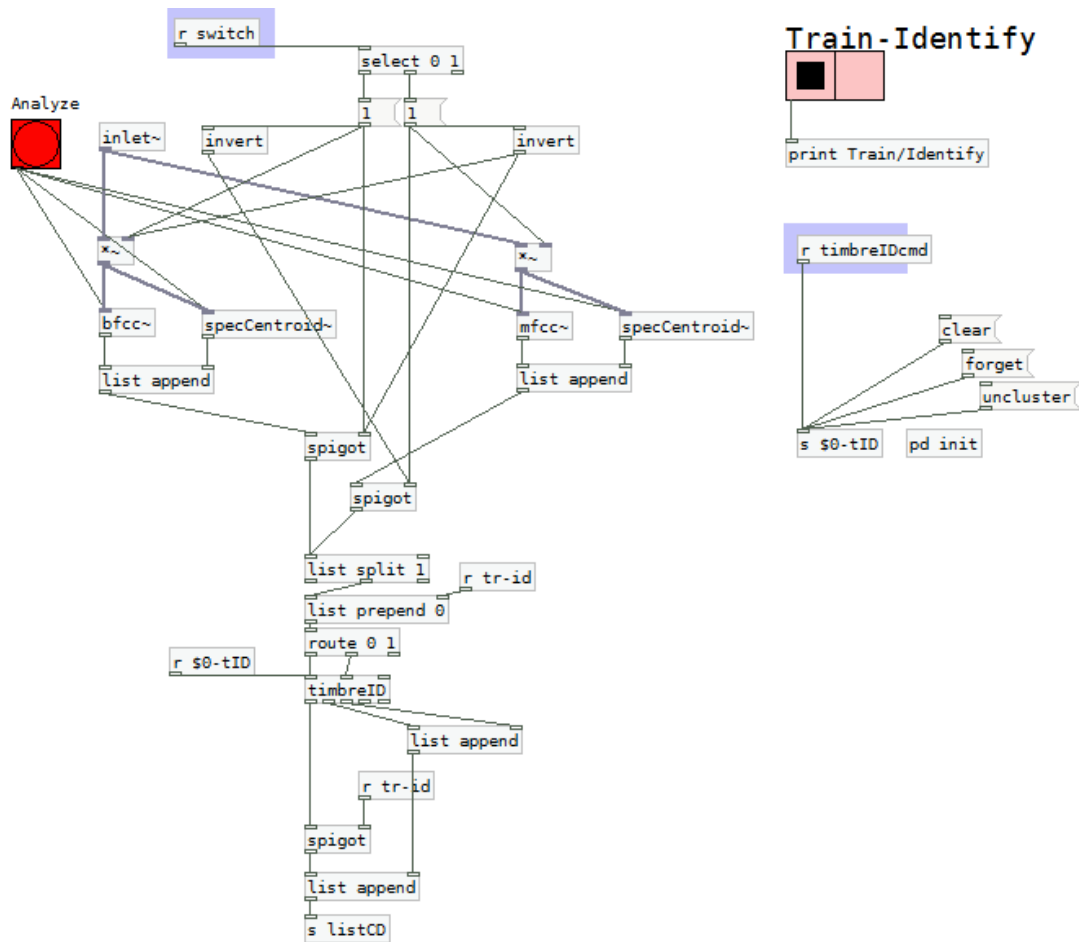


Figura 3.3: Patch che combina le misure di [bfcc~], [mfcc~] e [specCentroid~] in un unico vettore di feature.

external è stato impiegato nel test concatenandone l'uscita con i vettori generati da [bfcc~] e da [mfcc~]. La patch che combina i coefficienti BFCC e MFCC con il centroide in un singolo vettore di feature inviato a [timbreID] è quella di Figura 3.3.

I risultati della combinazione tra le uscite di [bfcc~] e [mfcc~] con quelle di [specCentroid~] sono presentati nelle Tabelle 3.4, 3.5, 3.6 e 3.7.

La combinazione dei coefficienti BFCC e MFCC con l'external [specRolloff~] è mostrata in Figura 3.4. Nelle Tabelle 3.8, 3.9, 3.10 e 3.11 sono riportati i risultati di quest'ultimo test.

Per il controllo e l'esecuzione delle patch è stato sviluppato un programma in linguaggio C, sfruttando le potenzialità del framework *openFrameworks*, per l'utilizzo della versione embedded di Pure Data (*libpd*), come fatto per la versione mobile di *SoundRise* [14]. Con questo

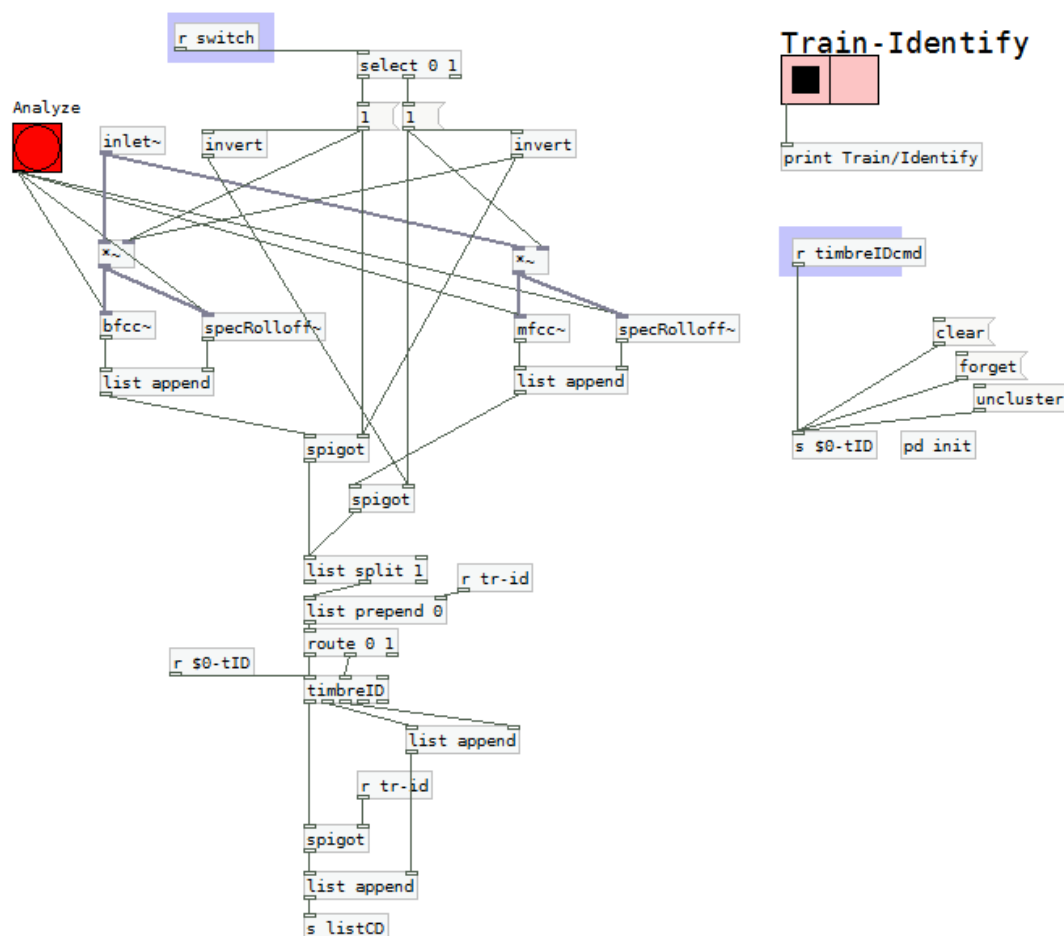


Figura 3.4: Patch che combina le misure di `[bfcc~]`, `[mfcc~]` e `[specRolloff~]` in un unico vettore di feature

programma è stato possibile gestire in modo efficiente gli oggetti nella patch, lo scambio di messaggi, i file audio di training e di test passati in ingresso e i risultati in uscita dalla patch. Gli external ricevono in ingresso da `[inlet~]` dei file audio, della durata di circa un secondo, contenenti ognuno una registrazione di una vocale. Per ogni vocale esistono sei registrazioni per la fase di train e quattro per la fase di test, fatte da bambini diversi. Per ogni file in esecuzione, tramite il codice C del programma di controllo, vengono fatti eseguire dagli external 10 frame di analisi in successione, in modo che rilascino i valori di uscita da inviare a `[timbreID]`. `[mfcc~]` produce un vettore di 38 componenti MFCC indipendentemente dalla finestra di analisi, che di default ha dimensione 1024 campioni. I componenti sono normalizzati tra i valori 1 e -1 e il primo avrà valore 1. `[bfcc~]` produce un vettore di 47 componenti BFCC indipendentemente dalla dimensione della finestra di analisi. Anche i componenti BFCC sono normalizzati tra 1 e

-1 di default e il primo componente avrà sempre valore 1.

3.1.1 Risultati

Dal confronto tra gli external [bfcc~] e [mfcc~], ottenuto con l'analisi della libreria *timbreID*, si osserva che la percentuale di vocali identificate in modo corretto risulta essere pressoché simile nei valori del timbro. Come mostrato in Figura 3.5 si ottengono valori oltre il 90% per le vocali *e*, *i* e *u* mentre l'unica differenza che rende [bfcc~] più accurato riguarda l'identificazione della vocale *a* in cui si osserva una percentuale pari al 77,5% contro un 45% data da [mfcc~]. A parità di indice corretto nell'identificazione, andando ad osservare il valore della distanza si nota che nel caso dell' [mfcc~] i valori risultano leggermente inferiori. Ciò nonostante bfcc nel complesso risulta avere una migliore affidabilità.

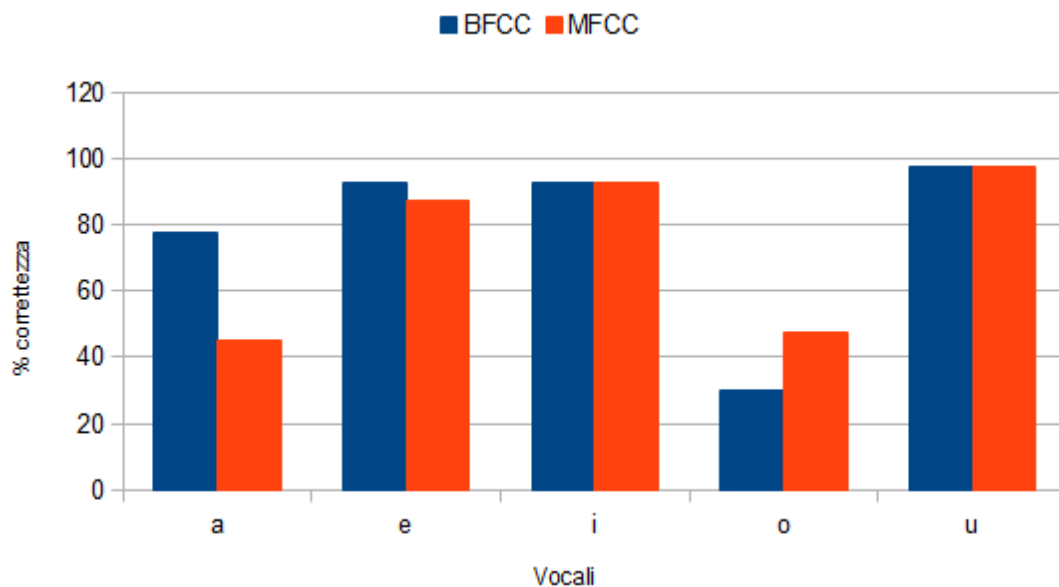


Figura 3.5: Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra [bfcc~] e [mfcc~].

Osservando i risultati del test in cui i coefficienti cepstrali sono stati combinati assieme a valori di frequenza come quelli dati dal centroide o dalla frequenza di roll-off l'identificazione

delle vocali non presenta risultati accettabili per nel primo caso e migliori, anche se non all'altezza di $[bfcc\sim]$ preso individualmente, nel secondo caso. Le percentuali di riconoscimento delle vocali risultano, nel caso di uso combinato con il centroide, al di sotto del 32,5% per le vocali a , e , i , o , mentre solo per la vocale u si ottiene una percentuale di correttezza pari al 72,5% come mostrato in Figura 3.6.

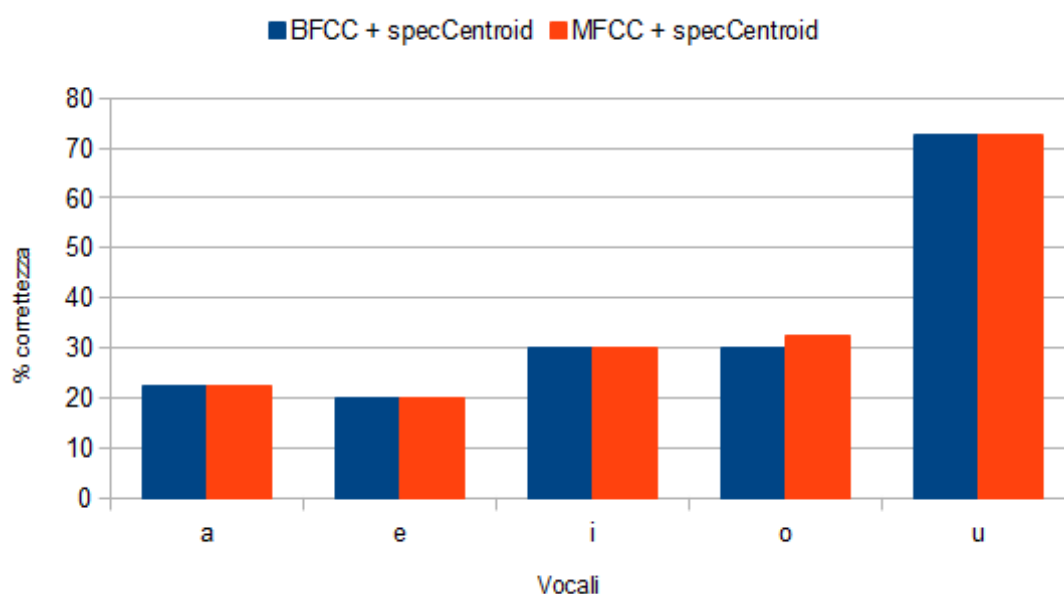


Figura 3.6: Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra $[bfcc\sim]$ e $[mfcc\sim]$ combinati con il centroide.

Per quanto riguarda l'uso combinato con la frequenza di roll-off la situazione migliora rispetto all'uso combinato col centroide, tuttavia nel complesso i valori ottenuti non eguagliano quelli del solo $[bfcc\sim]$. La percentuale massima di riconoscimento risulta essere pari al 75 % per la sola e , il resto delle vocali porta a risultati inferiori come si legge in Figura 3.7.

Il grafico in Figura 3.8 mostra una percentuale di correttezza complessiva dell'identificazione ottenuta considerando tutte e cinque le vocali, per quanto riguarda gli external che meglio si sono comportati nei tre test di confronto, ovvero $[bfcc\sim]$, $[mfcc\sim]$ con centroide e $[mfcc\sim]$ con frequenza di roll-off.

Nel caso delle combinazioni di $[bfcc\sim]$ e $[mfcc\sim]$ con $[specCentroid\sim]$ e $[specRolloff\sim]$

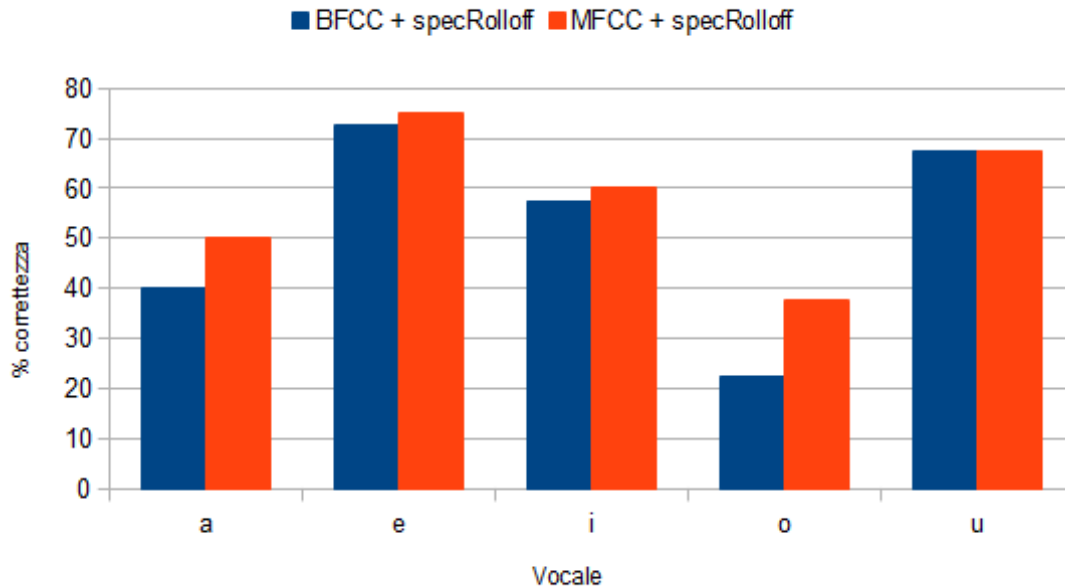


Figura 3.7: Valori di correttezza in percentuale per l'identificazione delle vocali ottenuti dal confronto tra $[bfcc\sim]$ e $[mfcc\sim]$ combinati con la frequenza di roll-off.

si osservano, per alcuni frame di analisi e su tutte le vocali, alcuni valori della distanza che si discostano di molto dalla media. Questo fa pensare che una serie di fattori come la qualità dei file audio utilizzati, le variazioni di tono negli stessi e la latenza, seppur minima, di Pure Data, possano alterare la frequenza di roll-off e causare tali scostamenti nella distanza Euclidea tra i vettori di feature contenuti nel database e il vettore analizzato. Tali alterazioni non si riscontrano sottoponendo ai test gli oggetti $[bfcc\sim]$ e $[mfcc\sim]$ singolarmente, poiché si esegue un calcolo del cepstrum su più bande pesato secondo la scale bark e mel rispettivamente. Inoltre, considerato che queste scale enfatizzano le basse frequenze, dove risiede l'informazione utile del segnale vocale, i valori ottenuti sono meno influenzati dalle alte frequenze.

I risultati derivanti dalla combinazione dei coefficienti mel-cepstrali e bark-cepstrali con il valore del centroide spettrale motiva ulteriormente la decisione di abbandonare l'uso del centroide in favore del cepstrum.

Ad ogni modo non è da escludere che combinazioni di più external per l'estrazione di feature possano portare a risultati migliori rispetto al solo uso di $bfcc$. Un'altra soluzione, ancora da esplorare, consisterebbe nell'analizzare frame multipli del segnale, monitorando quindi un'evo-

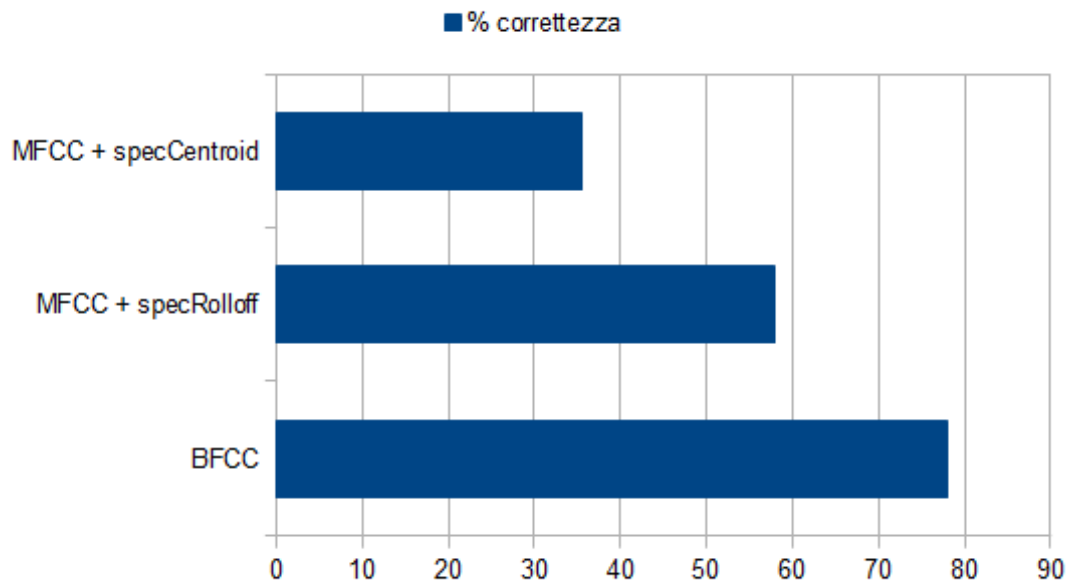


Figura 3.8: Valori complessivi di correttezza in percentuale, ottenuti considerando l'identificazione di tutte le vocali, dati da $[bfcc\sim]$ e da $[mfcc\sim]$ combinato con la frequenza di roll-off e con il centroide.

luzione temporale più ampia dell'evento sonoro, e prendere una media dei valori delle feature risultanti.

3.2 Test di Usabilità per misurare l'intuitività dell'interfaccia di *SoundRise*

Il test di usabilità dà indicazioni su possibili elementi dell'applicazione che possono ostacolare il corretto svolgimento dei compiti da parte dell'utente medio o di un target più preciso di utenti. Dal momento che lo scopo principale di *SoundRise* è quello di fornire al bambino una modalità di apprendimento delle caratteristiche della voce attraverso la loro resa grafica, è necessario che il sistema produca un chiaro e inequivocabile feedback grafico di ognuna di esse. Un obiettivo è dunque quello di misurare il grado di utilizzabilità della voce dei bambini come input per interagire con l'applicazione; testare quindi solo ed esclusivamente quanto consistente è la risposta del sistema all'input vocale dei bambini. Il test è stato svolto presso la Scuola Primaria Elisa Frinta di Gorizia (Go). Per condurre questo test sono stati necessari:

1. una demo dell'applicazione da testare;
2. una serie di compiti significativi da somministrare ai partecipanti;
3. una sede comoda, in cui non essere disturbati, con un computer dello stesso livello di quelli che usano di solito gli utenti; è stato possibile utilizzare un'aula della scuola elementare in cui è stato svolto il test;
4. un numero di utenti relativamente omogeneo, da convocare uno alla volta; si tratta di 39 alunni di III elementare.
5. un osservatore (dott.ssa Serena Zanolla) per condurre il test mettendo a proprio agio le persone senza influenzarne la prestazione, in grado di annotare errori e osservazioni in tempo reale, traendo il massimo dai soggetti coinvolti.

I compiti da far eseguire ai bambini sono stati i seguenti:

- Task: produci un suono con la tua voce e osserva cosa fa il sole. Questo task viene richiesto per ogni feature della voce.
- Seguono 2 domande fatte dall'esaminatore:
 - Domanda 1: cosa fa il sole?
 - Domanda 2: secondo te cosa fai con la tua voce per far salire/ingrandire/sorridere/cambiare il colore al sole?

Per la realizzazione del test si sono inoltre rispettati alcuni criteri:

- resistere alla tentazione di indagare quello che interessa ad ogni costo. È necessario lasciare libero l'utente di affrontare il compito con la strategia che preferisce e con la libertà di ragionamento che crede;
- approfondimenti su un certo aspetto dell'applicazione possono essere richiesti al termine della prestazione, quando è opportuno un piccolo colloquio chiarificatore con l'utente che si è prestato al test;
- i dati che si raccolgono, dato l'esiguo numero dei partecipanti, non hanno validità statistica. Possono comunque essere riassunti in grafici o tabelle per semplificare l'esposizione, con l'accortezza però di non farli passare per rappresentativi di una popolazione, ma come utili indicazioni di tendenza da confrontare con le prestazioni riscontrate.

Nel test è valutato, mediante le risposte dell'esaminato, se il sistema fornisce un feedback grafico coerente con l'informazione da trasmettere. La Tabella 3.12 riassume i risultati ottenuti. Un risultato pari a *si* per la Domanda 1 e 2 rappresenta una corretta ed esaustiva risposta. Un risultato *no* indica invece che tale comprensione non c'è stata. Nell'ultima colonna della tabella vengono riportate per ogni alunno le sigle delle feature riconosciute nella modalità in cui tutte sono allo stesso istante analizzate e rappresentate a livello grafico.

Quello che si è potuto osservare è che per quanto riguarda il movimento del sole (Domanda 1), questo è stato pienamente compreso dai bambini. È risultato comprensibile, nelle sue animazioni, per ogni feature della voce analizzata singolarmente, come riportato nel grafico di Figura 3.9.

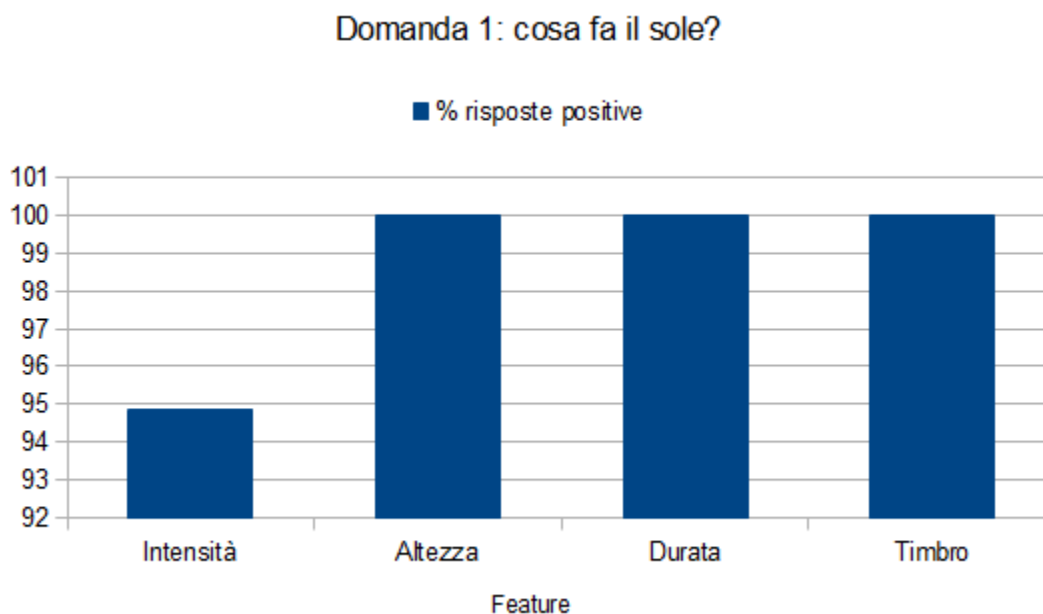


Figura 3.9: Istogramma che mostra la percentuale di risposte corrette date dai bambini alla Domanda 1 per tutte le feature analizzate individualmente.

Il riconoscimento di ciò che il sole fa è invece meno chiaro nel caso in cui le feature sono analizzate simultaneamente. In tal caso vengono riconosciuti solo alcune animazioni del sole ed il motivo potrebbe risiedere in un contenuto informativo eccessivo che non permette all'utente di distinguere con chiarezza ogni animazione. Nella maggior parte dei casi solamente due animazioni su quattro vengono individuate, in particolare le coppie intensità-timbro (IT) ed

ampiezza-intensità (AI). I risultati sono mostrati nel grafico di Figura 3.10.

Rappresentazione di tutte le feature assieme - Domanda 1: cosa fa il sole?

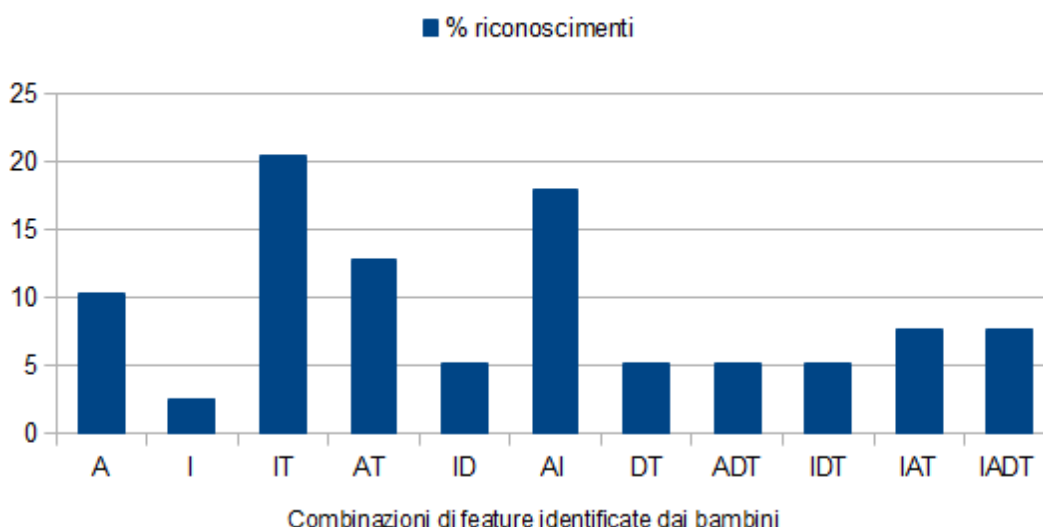


Figura 3.10: Istogramma che mostra la percentuale di combinazioni di feature identificate dai bambini come risposta alla Domanda 1 per tutte le feature analizzate contemporaneamente. *I* = Intensità, *A* = Ampiezza, *D* = Durata, *T* = Timbro.

Per quel che riguarda la comprensione del controllo del feedback grafico, e quindi del sole, tramite le caratteristiche della propria voce (Domanda 2), un buon riscontro si è avuto con tutte le feature, ad eccezione del timbro. Le percentuali di risposte positive alla Domanda 2 sono riportate in Figura 3.11.

Questo risultato lo si deve al fatto che non vi è una fedele identificazione delle vocali, il che non porta di conseguenza ad un'associazione vocale colore costante, che risulta perciò confusa all'utente. Il database non contiene infatti un numero di istanze che permetta una buona affidabilità da parte del sistema di identificazione del timbro di *SoundRise*. La mancata robustezza del database è stata dunque ritenuta causa dei risultati avuti dal test per la feature timbro.

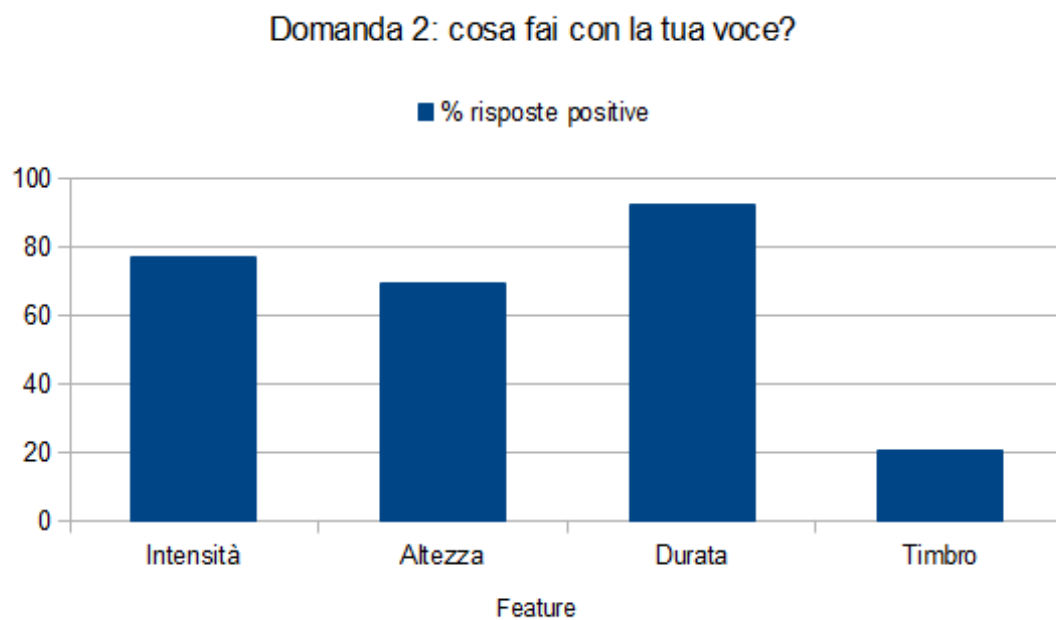


Figura 3.11: Istogramma che mostra la percentuale di risposte corrette date dai bambini alla Domanda 2 per tutte le feature analizzate individualmente.

Tabella 3.1: Risultati [bʃec~]: *F* = nome del file audio; *T* = indice del cluster; *D* = distanza dal match più vicino

	frame 1		frame 2		frame 3		frame 4		frame 5		frame 6		frame 7		frame 8		frame 9		frame 10	
F	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D
a03	0	0,10	0	0,37	0	0,55	0	0,62	0	0,40	0	0,50	0	0,34	0	0,65	0	0,61	0	0,58
a04	3	0,13	0	0,63	0	0,63	0	0,62	0	0,70	0	0,82	0	0,83	0	0,94	0	0,90	0	0,76
a07	3	0,20	3	0,49	3	0,42	3	0,56	0	0,48	3	0,54	3	0,51	0	0,47	3	0,43	3	0,39
a10	0	0,04	0	0,26	0	0,38	0	0,44	0	0,38	0	0,38	0	0,47	0	0,44	0	0,43	0	0,47
e03	1	0,16	1	0,86	1	0,76	1	0,73	1	0,75	1	0,63	1	0,72	3	0,70	1	0,78	1	0,98
e04	1	0,17	1	0,53	1	0,56	1	0,64	1	0,73	1	0,58	1	0,48	1	0,58	1	0,55	1	0,54
e07	1	0,14	1	0,30	1	0,26	1	0,28	1	0,31	1	0,27	1	0,35	1	0,32	1	0,32	1	0,29
e10	1	0,22	1	0,27	1	0,29	1	0,26	1	0,27	2	0,30	1	0,29	2	0,28	1	0,27	1	0,25
i03	2	0,20	2	0,86	2	0,98	4	1,09	4	1,14	4	1,12	2	0,94	2	0,56	2	0,92	2	1,13
i04	2	0,29	2	0,68	2	0,89	2	1,00	2	1,01	2	0,81	2	0,84	2	0,82	2	0,77	2	0,87
i07	2	0,21	2	0,22	2	0,29	2	0,32	2	0,32	2	0,24	2	0,26	2	0,36	2	0,32	2	0,27
i10	2	0,11	2	0,26	2	0,40	2	0,30	2	0,23	2	0,19	2	0,23	2	0,22	2	0,21	2	0,25
o03	4	0,11	1	1,36	1	1,29	1	1,27	1	1,48	1	1,53	1	1,50	1	1,35	1	1,41	1	1,58
o04	3	0,38	3	1,81	3	1,78	3	1,71	3	1,76	3	1,53	0	1,54	0	1,56	3	1,54	3	1,59
o07	0	0,15	0	0,79	0	0,68	0	0,60	0	0,64	0	0,72	3	0,75	1	1,22	3	1,12	3	1,39
o10	3	0,23	1	0,55	1	0,78	1	0,68	1	0,62	1	0,84	1	0,89	1	0,82	1	0,97	1	0,86
u03	4	0,17	4	1,39	4	1,18	4	1,49	4	1,43	4	1,45	4	1,32	4	1,43	4	1,21	4	1,41
u04	3	0,22	4	2,30	4	1,91	4	2,25	4	1,90	4	2,35	4	2,76	4	2,71	4	2,20	4	2,95
u07	4	0,07	4	0,68	4	0,78	4	0,95	4	1,29	4	1,22	4	0,64	4	0,80	4	0,97	4	1,31
u10	4	0,09	4	0,56	4	0,66	4	0,59	4	0,45	4	0,45	4	0,50	4	0,63	4	0,58	4	0,64

Tabella 3.2: Risultati [mfcc~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino

F	frame 1		frame 2		frame 3		frame 4		frame 5		frame 6		frame 7		frame 8		frame 9		frame 10	
	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D
a03	0	0,10	0	0,20	0	0,40	0	0,48	0	0,29	0	0,37	0	0,26	0	0,47	0	0,39	0	0,37
a04	3	0,07	3	0,31	3	0,31	3	0,28	3	0,33	3	0,30	3	0,29	3	0,31	3	0,34	3	0,30
a07	3	0,17	3	0,18	3	0,18	3	0,19	3	0,18	3	0,19	3	0,15	3	0,18	3	0,15	3	0,15
a10	0	0,04	0	0,12	0	0,19	0	0,21	3	0,19	0	0,22	0	0,26	0	0,27	0	0,19	3	0,26
e03	1	0,14	1	0,63	1	0,54	2	0,49	1	0,57	1	0,60	3	0,66	3	0,52	1	0,49	1	0,46
e04	1	0,16	1	0,39	1	0,38	1	0,41	1	0,47	1	0,35	1	0,33	1	0,32	1	0,38	1	0,40
e07	1	0,16	1	0,23	1	0,22	2	0,26	1	0,24	1	0,21	1	0,20	1	0,19	1	0,19	1	0,18
e10	4	0,19	1	0,12	1	0,12	1	0,11	1	0,12	1	0,13	1	0,14	1	0,20	1	0,10	1	0,08
i03	2	0,22	2	0,78	2	0,95	4	1,11	4	1,22	4	1,22	2	1,02	2	0,48	2	0,97	2	1,13
i04	2	0,28	2	0,67	2	0,92	2	1,05	2	1,06	2	0,82	2	0,79	2	0,81	2	0,69	2	0,76
i07	2	0,18	2	0,16	2	0,20	2	0,23	2	0,21	2	0,17	2	0,19	2	0,26	2	0,27	2	0,24
i10	2	0,12	2	0,20	2	0,33	2	0,25	2	0,16	2	0,16	2	0,19	2	0,17	2	0,16	2	0,19
o03	4	0,08	3	0,82	3	0,80	3	0,86	1	0,85	1	0,87	1	0,92	1	0,85	1	0,85	1	0,92
o04	3	0,35	0	1,52	3	1,27	0	1,26	0	1,28	0	1,09	0	1,25	0	1,25	0	1,10	0	1,18
o07	0	0,14	3	0,34	0	0,30	0	0,26	0	0,29	3	0,29	3	0,27	3	0,71	3	0,55	3	0,70
o10	3	0,16	3	0,42	3	0,72	3	0,52	3	0,50	3	0,73	1	0,91	3	0,74	1	0,98	3	0,79
u03	4	0,23	4	1,36	4	0,88	4	1,34	4	1,38	4	1,58	4	1,30	4	1,47	4	1,08	4	1,27
u04	3	0,22	4	2,15	4	1,78	4	2,16	4	1,63	4	2,27	4	2,78	4	2,44	4	1,88	4	3,27
u07	4	0,07	4	0,64	4	0,60	4	0,66	4	0,66	4	0,76	4	0,50	4	0,65	4	0,74	4	0,68
u10	4	0,08	4	0,40	4	0,47	4	0,44	4	0,31	4	0,28	4	0,31	4	0,43	4	0,39	4	0,46

Tabella 3.3: Risultati [bjɛc~] con 40 coefficienti: *F* = nome del file audio; *T* = indice del cluster; *D* = distanza dal match più vicino

	frame 1		frame 2		frame 3		frame 4		frame 5		frame 6		frame 7		frame 8		frame 9		frame 10	
F	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D
a03	0	0,10	0	0,35	0	0,53	0	0,60	0	0,38	0	0,49	0	0,33	0	0,61	0	0,58	0	0,54
a04	3	0,13	0	0,58	0	0,59	0	0,55	0	0,64	0	0,72	0	0,72	0	0,85	0	0,85	0	0,67
a07	3	0,20	3	0,43	3	0,36	3	0,47	0	0,43	3	0,45	3	0,44	0	0,44	3	0,38	3	0,33
a10	0	0,04	0	0,25	0	0,34	0	0,41	0	0,35	0	0,35	0	0,43	0	0,40	0	0,39	0	0,44
e03	1	0,15	1	0,79	1	0,70	1	0,70	1	0,72	1	0,61	1	0,70	3	0,68	1	0,75	1	0,88
e04	1	0,17	1	0,51	1	0,51	1	0,61	1	0,67	1	0,52	1	0,43	1	0,52	1	0,51	1	0,48
e07	1	0,14	1	0,28	1	0,26	1	0,28	1	0,30	1	0,27	1	0,33	1	0,31	1	0,30	1	0,27
e10	1	0,22	1	0,24	1	0,25	1	0,23	1	0,23	1	0,29	1	0,26	2	0,28	1	0,23	1	0,21
i03	2	0,20	2	0,84	2	0,96	4	1,06	4	1,11	4	1,09	2	0,91	2	0,55	2	0,90	2	1,08
i04	2	0,28	2	0,66	2	0,87	2	0,98	2	0,99	2	0,79	2	0,81	2	0,81	2	0,75	2	0,81
i07	2	0,21	2	0,19	2	0,26	2	0,28	2	0,27	2	0,22	2	0,23	2	0,33	2	0,30	2	0,26
i10	2	0,11	2	0,25	2	0,38	2	0,29	2	0,20	2	0,18	2	0,22	2	0,20	2	0,20	2	0,24
o03	4	0,11	1	1,26	1	1,22	1	1,19	1	1,38	1	1,42	1	1,39	1	1,26	1	1,32	1	1,47
o04	3	0,38	3	1,69	3	1,65	3	1,61	3	1,64	3	1,45	0	1,48	0	1,51	3	1,48	3	1,53
o07	0	0,15	0	0,73	0	0,62	0	0,55	0	0,58	0	0,67	3	0,71	1	1,17	3	1,09	3	1,34
o10	3	0,23	1	0,54	1	0,77	1	0,67	1	0,61	1	0,83	1	0,89	1	0,82	1	0,97	1	0,86
u03	4	0,17	4	1,36	4	1,16	4	1,45	4	1,40	4	1,42	4	1,29	4	1,41	4	1,19	4	1,38
u04	3	0,22	4	2,26	4	1,87	4	2,21	4	1,85	4	2,31	4	2,72	4	2,67	4	2,16	4	2,89
u07	4	0,07	4	0,65	4	0,75	4	0,93	4	1,28	4	1,19	4	0,61	4	0,77	4	0,94	4	1,30
u10	4	0,09	4	0,55	4	0,65	4	0,58	4	0,44	4	0,45	4	0,49	4	0,62	4	0,56	4	0,63

Tabella 3.4: Risultati [bfcc~] combinato con [specCentroid~]: $F =$ nome del file audio; $T =$ indice del cluster; $D =$ distanza dal match più vicino

F	frame 1		frame 2		frame 3		frame 4		frame 5	
	T	D	T	D	T	D	T	D	T	D
a03	2	1,50	3	89,94	3	9,92	3	1,98	4	11,81
a04	5	1781,10	2	68,14	0	6,04	0	21,47	1	30,13
a07	0	126312,00	3	1,61	0	16,29	0	1,26	4	245195,00
a10	2	449698,00	3	2,69	3	4,22	0	29,60	3	10,62
e03	1	1,67	1	1,97	4	6,32	1	4,47	1	2,70
e04	3	3,39	2	14,10	2	2,43	0	120,28	0	2,78
e07	1	1,39	4	14,59	2	70,07	2	5,53	4	37,96
e10	1	0,57	4	15,04	2	3,10	2	74,79	1	88,06
i03	3	1,64	0	4,06	3	1,65	0	4,40	4	806468,00
i04	5	1203,48	3	87,10	2	208377,00	0	4,71	2	135597,00
i07	3	2,59	2	0,71	2	1,56	1	1,35	2	5,01
i10	0	1177,19	2	0,38	2	55,59	1	20,58	0	118101,00
o03	0	1353,89	4	423412,00	4	6,30	4	26,68	4	16,36
o04	2	16,58	0	15,79	0	7,39	4	7,27	0	13,04
o07	2	89,01	4	4,25	3	56,71	4	7,21	3	2,37
o10	1	15,24	3	11,34	4	5,07	4	206157,00	4	54,21
u03	3	233398,00	4	7881,50	4	3497,66	4	9988,33	4	16327,40
u04	2	1,53	0	7,76	3	36,51	3	6,20	3	4,79
u07	3	20,98	4	3501,95	4	2015,92	4	5628,44	4	3788,45
u10	3	853023,00	4	4315,38	4	6489,68	4	1,69	4	4148,66

Tabella 3.5: Risultati [bjcc~] combinato con [specCentroid~]: $F =$ nome del file audio; $T =$ indice del cluster; $D =$ distanza dal match più vicino

F	frame 6		frame 7		frame 8		frame 9		frame 10	
	T	D	T	D	T	D	T	D	T	D
a03	1	1,77	3	8,17	1	24,01	3	1,78	1	11,27
a04	4	8,21	3	34,42	0	3,36	0	1,81	1	4,90
a07	3	2,03	4	11,97	4	907418,00	3	5,08	3	0,98
a10	3	1,39	0	8,90	3	76,72	3	8,82	3	4,53
e03	0	6,71	3	64,07	3	3,16	4	5,70	3	2,97
e04	3	1,89	2	3,10	0	210136,00	2	5,94	0	56,72
e07	4	10,64	0	2,14	0	0,98	1	1,95	3	3,49
e10	0	50,41	0	88,65	4	1,33	0	2,86	0	231501,00
i03	0	15,94	0	22,16	0	328,97	1	12,43	3	8,27
i04	2	9,24	2	208201,00	2	1,26	2	177,20	3	81447,00
i07	0	147,00	1	5,63	3	28,12	1	3,01	0	2,82
i10	2	4,37	1	3,85	0	4,11	0	169269,00	1	18,84
o03	4	9,74	4	320389,00	4	6,15	4	86,84	4	10,33
o04	3	4,30	4	9,05	0	10,56	3	8,56	3	9,23
o07	3	55,08	3	101433,00	3	31,99	4	7,95	3	3651,00
o10	4	18,80	4	452906,00	3	7,79	4	287,16	3	25,98
u03	4	15062,60	4	7477,97	4	1894,03	4	46,64	4	6,93
u04	4	20,78	3	6717,00	4	324792,00	3	55,85	3	7,94
u07	4	5143,27	4	7198,95	4	3593,42	4	3441,48	4	6766,09
u10	4	4090,59	4	1571,84	4	3,26	4	10,82	4	2,96

Tabella 3.6: Risultati [mfcc~] combinato con [specCentroid~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino

F	frame 1		frame 2		frame 3		frame 4		frame 5	
	T	D	T	D	T	D	T	D	T	D
a03	2	1,60	3	89,56	3	9,59	3	1,75	4	11,84
a04	5	1781,07	2	67,82	0	5,70	0	21,04	1	29,57
a07	0	126,25	3	1,07	0	15,96	0	0,87	4	244,93
a10	2	449772,00	3	2,36	3	3,69	0	29,07	3	10,43
e03	1	1,64	1	1,72	4	5,78	1	4,28	1	1,90
e04	3	3,46	2	13,96	2	2,37	0	119929,00	0	2,49
e07	1	1,34	4	14469,00	2	70,01	2	5,34	4	37977,00
e10	1	0,39	4	15,30	2	2,97	2	74,67	1	87,67
i03	3	1,61	0	4,11	3	1,93	0	4,61	4	806448,00
i04	5	1203,44	3	87,15	2	208578,00	0	5,08	2	135864,00
i07	3	2,80	2	0,73	2	1,21	1	1,46	2	4,76
i10	0	1177,24	2	0,36	2	55,42	1	20,61	0	118381,00
o03	0	1353,86	4	423078,00	4	5,53	4	26,24	4	16,13
o04	2	16,76	0	15,54	0	7,07	4	7,13	0	12,50
o07	2	89,18	3	3,35	3	55,86	4	6,80	3	1,54
o10	1	15,23	3	10,10	4	4,76	4	206013,00	4	53,75
u03	3	233,52	4	7880,49	4	3496,19	4	9987,14	4	16326,50
u04	2	1,67	0	7,63	3	36,70	3	6,21	3	4,51
u07	3	21,30	4	3499,73	4	2013,64	4	5626,11	4	3786,15
u10	3	853005,00	4	4315,33	4	6489,64	4	0,90	4	4148,59

Tabella 3.7: Risultati [mfec~] combinato con [specCentroid~]: $F =$ nome del file audio; $T =$ indice del cluster; $D =$ distanza dal match più vicino

F	frame 6		frame 7		frame 8		frame 9		frame 10	
	T	D	T	D	T	D	T	D	T	D
a03	1	1,59	3	7,84	1	23,81	3	1,00	1	11,08
a04	4	7,34	3	33,53	0	3,00	0	0,96	1	4,29
a07	3	1,61	4	11685,00	4	907164,00	3	5,01	3	0,31
a10	3	0,83	0	8434,00	3	76,23	3	8,50	3	4,26
e03	0	6,77	3	64,33	3	3,23	4	5,46	3	2,87
e04	3	1,76	2	3,25	0	209771,00	2	6,00	0	56,65
e07	4	10,70	0	1,72	0	0,64	1	1,72	3	3,15
e10	0	50301,00	0	88,52	4	1,60	0	2,87	0	231524,00
i03	0	16,65	0	22,50	0	328754,00	1	12,84	3	8,74
i04	2	9,32	2	208334,00	2	1,39	2	177168,00	3	81,64
i07	0	147131,00	1	5,74	3	28,31	1	2,34	0	2,41
i10	2	4,41	1	4,00	0	4,01	0	169322,00	1	18,85
o03	4	9,34	4	319432,00	4	5,31	4	85,98	4	9,35
o04	3	3,77	4	8,73	0	10,41	3	8,45	3	9,15
o07	3	54,08	3	100475,00	3	30,70	4	7,69	3	2,15
o10	4	18652,00	4	452794,00	3	6,22	4	287039,00	3	24,72
u03	4	15062,00	4	7477,03	4	1893,08	4	45,45	4	7,20
u04	4	19927,00	3	6,24	4	325251,00	3	56,10	3	8,12
u07	4	5140,87	4	7196,41	4	3591,05	4	3439,10	4	6763,65
u10	4	4090,51	4	1571,81	4	3,33	4	10,02	4	3,14

Tabella 3.8: Risultati [bfcc~] combinato con [specRolloff~]: F = nome del file audio; T = indice del cluster; D = distanza dal match più vicino

F	frame 1		frame 2		frame 3		frame 4		frame 5	
	T	D	T	D	T	D	T	D	T	D
a03	5	0,62	2,00	1856,37	1,00	0,84	1,00	1,25	1,00	0,99
a04	0	1,10	4,00	3,90	1,00	1,61	1,00	3,69	0,00	0,85
a07	1	0,76	0,00	1855,67	4,00	1856,68	3,00	1,04	0,00	0,91
a10	0	0,83	2,00	1856,41	0,00	0,82	3,00	0,66	0,00	1855,83
e03	0	1,08	1,00	16694,50	1,00	1,88	1,00	1,24	4,00	1,02
e04	1	0,52	2,00	1,29	1,00	1,23	1,00	0,76	1,00	0,87
e07	2	0,30	1,00	0,64	1,00	0,71	1,00	0,82	1,00	0,73
e10	1	0,52	1,00	0,35	2,00	0,53	1,00	0,39	1,00	0,54
i03	1	1,94	0,00	29678,20	2,00	2,26	1,00	1,54	1,00	2,58
i04	1	0,90	4,00	1858,84	2,00	118704,00	3,00	3,16	2,00	1,31
i07	2	1,63	2,00	0,93	2,00	0,41	2,00	1855,12	2,00	0,53
i10	2	1854,83	2,00	0,28	2,00	0,84	2,00	1855,69	2,00	1855,65
o03	2	1855,06	4,00	1859,31	5,00	3,54	5,00	3,56	4,00	4,39
o04	4	1855,81	2,00	3,48	3,00	2,01	3,00	1,95	1,00	4,98
o07	4	1,10	2,00	7421,69	4,00	1856,46	4,00	1856,33	3,00	1,28
o10	3	1,74	1,00	0,96	4,00	2,91	4,00	1857,17	4,00	1857,25
u03	3	0,88	4,00	4,47	4,00	7420,56	4,00	7420,96	4,00	7420,88
u04	4	7419,13	2,00	46371,90	1,00	3,48	3,00	2,83	1,00	4,13
u07	3	1856,29	3,00	4,95	4,00	1855,71	4,00	1855,94	4,00	1856,69
u10	4	2,26	2,00	7420,73	4,00	1,75	4,00	1,75	4,00	1856,14

Tabella 3.9: Risultati [bʃec~] combinato con [specRolloff~]: *F* = nome del file audio; *T* = indice del cluster; *D* = distanza dal match più vicino

F	frame 6		frame 7		frame 8		frame 9		frame 10	
	T	D	T	D	T	D	T	D	T	D
a03	0,00	0,94	0,00	1,09	1,00	1,24	1,00	1,27	0,00	1,06
a04	4,00	1,80	0,00	0,89	0,00	1,87	0,00	1,15	1,00	1,38
a07	3,00	0,66	0,00	0,53	3,00	0,67	3,00	0,83	3,00	0,85
a10	0,00	0,46	0,00	0,96	3,00	1856,07	3,00	1,07	3,00	1,04
e03	1,00	1,14	1,00	1,50	1,00	1855,53	1,00	1,34	1,00	2,47
e04	0,00	0,95	1,00	0,48	0,00	1,08	1,00	0,73	1,00	0,86
e07	1,00	2,02	1,00	0,90	1,00	0,70	1,00	0,70	2,00	1855,19
e10	2,00	0,45	1,00	0,29	2,00	1855,18	3,00	0,55	1,00	391,00
i03	0,00	3,56	2,00	1,27	2,00	0,73	1,00	1,10	1,00	2,48
i04	4,00	2,57	2,00	1855,74	2,00	2,36	2,00	1856,86	2,00	2,39
i07	2,00	0,48	2,00	0,55	1,00	1,25	1,00	0,51	3,00	0,70
i10	2,00	1,28	2,00	1,01	1,00	1,32	0,00	1,90	1,00	1,90
o03	4,00	1856,99	4,00	2,21	4,00	1857,35	4,00	1856,83	4,00	5,65
o04	0,00	2,91	1,00	2,39	1,00	2,45	0,00	2,92	0,00	1857,61
o07	4,00	1856,37	4,00	1856,44	3,00	2,60	4,00	1857,03	3,00	1857,21
o10	3,00	1,75	4,00	1857,98	3,00	1856,80	4,00	1858,19	3,00	1,61
u03	4,00	16694,40	4,00	16694,30	4,00	7420,80	4,00	16694,20	4,00	7420,84
u04	3,00	2,96	0,00	6,10	3,00	1859,42	0,00	1859,76	1,00	4,97
u07	4,00	1856,29	4,00	1855,64	4,00	1855,80	4,00	1856,02	4,00	1856,45
u10	4,00	1856,09	4,00	16694,00	4,00	16694,20	4,00	1855,97	4,00	1856,02

Tabella 3.10: Risultati [mfcc~] combinato con [specRolloff~]: $F =$ nome del file audio; $T =$ indice del cluster; $D =$ distanza dal match più vicino

F	frame 1		frame 2		frame 3		frame 4		frame 5	
	T	D	T	D	T	D	T	D	T	D
a03	5	0,60	2	1856,36	0	0,69	0	0,95	1	0,86
a04	0	0,86	4	3,74	1	1,22	1	3,31	0	0,52
a07	1	0,67	0	1855,34	4	1856,22	3	0,33	0	0,49
a10	0	0,67	2	1856,56	0	0,39	0	0,50	0	1855,62
e03	0	0,95	1	16694,30	1	0,82	1	0,96	1	0,73
e04	1	0,51	2	1,06	1	0,99	1	0,51	1	552,00
e07	2	0,28	1	0,52	1	0,48	1	0,68	1	0,56
e10	1	0,54	1	0,16	2	0,59	1	0,14	1	0,25
i03	1	1,61	0	29678,20	2	2,20	1	1,91	1	3,08
i04	1	0,83	4	1858,65	2	118704,00	3	3,47	2	1,58
i07	2	1,63	2	0,58	2	0,35	2	1855,12	2	0,46
i10	2	1854,83	2	0,26	2	0,64	2	1855,53	2	1855,56
o03	2	1855,02	4	1858,74	4	3,90	5	4,11	4	3,99
o04	4	1855,78	2	3,27	3	1,48	3	1,52	1	4,45
o07	4	0,85	2	7421,36	4	1855,62	4	1855,59	3	0,30
o10	3	1,31	1	0,83	3	1,64	3	1856,03	3	1856,04
u03	3	1,04	4	3,72	4	7420,58	4	7421,14	4	7421,17
u04	4	7419,15	2	46371,80	1	3,64	3	3,22	1	3,81
u07	3	1856,53	3	3,14	4	1855,49	4	1855,74	4	1856,38
u10	4	2,25	2	7420,45	4	0,89	4	0,90	4	1855,41

Tabella 3.11: Risultati [mfcc~] combinato con [specRolloff~]: $F =$ nome del file audio; $T =$ indice del cluster; $D =$ distanza dal match più vicino

F	frame 6		frame 7		frame 8		frame 9		frame 10	
	T	D	T	D	T	D	T	D	T	D
a03	0	0,60	0	0,98	0	0,94	1	0,97	0	0,92
a04	4	1,12	0	0,56	0	0,84	0	0,74	1	0,84
a07	3	0,45	0	0,42	3	0,50	3	0,15	3	0,23
a10	0	0,34	0	0,51	3	1856,00	3	0,50	3	0,45
e03	1	1,00	1	0,81	1	1855,50	1	1,20	1	2,56
e04	0	0,80	1	0,33	0	0,99	1	0,55	1	0,80
e07	1	1,86	1	0,79	1	0,52	1	0,54	2	1855,03
e10	2	0,46	1	0,14	2	1855,16	3	0,51	1	0,16
i03	0	4,24	2	1,44	2	0,72	2	1,39	1	2,80
i04	4	2,85	2	1855,83	2	2,13	2	1856,65	2	2,13
i07	2	0,41	2	0,47	1	1,26	1	0,59	3	0,89
i10	2	0,91	2	0,71	1	1,15	3	1,51	1	1,41
o03	4	1856,21	4	1,54	4	1857,51	4	1856,24	4	5,29
o04	0	2,34	1	2,40	1	2,39	0	2,60	0	1857,20
o07	4	1855,51	4	1855,49	3	1,14	3	1855,68	3	1855,59
o10	3	0,97	3	1856,61	3	1856,29	3	1856,77	3	1,13
u03	4	16694,80	4	16694,60	4	7421,10	4	16694,40	4	7420,98
u04	3	3,35	0	6,14	3	1859,28	0	1859,04	1	5,84
u07	4	1856,12	4	1855,36	4	1855,55	4	1855,83	4	1856,27
u10	4	1855,36	4	16693,10	4	16693,30	4	1855,63	4	1855,69

Tabella 3.12: Risultati del test di Usabilità di SoundRise

A cosa fa il sole? **B** cosa fai con la tua voce?

	Alunni	Intensità (I)		Altezza (A)		Durata (D)		Timbro (T)		Tutte (IADT)
		A	B	A	B	A	B	A	B	A
1	B	sì	sì	sì	sì	sì	sì	sì	no	IT
2	M	sì	sì	sì	no	sì	sì	sì	sì	IT
3	G	sì	sì	sì	sì	sì	sì	sì	sì	IT
4	L	sì	sì	sì	no	sì	sì	sì	no	AT
5	R	sì	sì	sì	sì	sì	sì	sì	no	ADT
6	M	sì	no	sì	sì	sì	sì	sì	no	ID
7	D	sì	sì	sì	no	sì	no	sì	no	AT
8	F	sì	sì	sì	sì	sì	sì	sì	no	ADT
9	A	sì	sì	sì	sì	sì	sì	sì	no	IDT
10	E	sì	no	sì	no	sì	sì	sì	no	ID
11	M	sì	no	sì	sì	sì	sì	sì	no	AT
12	QF	sì	sì	sì	sì	sì	sì	sì	no	IDT
13	S	sì	sì	sì	no	sì	sì	sì	no	IADT
14	N	no	no	sì	sì	sì	sì	sì	sì	AI
15	J	sì	sì	sì	no	sì	sì	sì	sì	IAT
16	N	no	sì	sì	sì	sì	no	sì	no	IT
17	E	sì	sì	sì	sì	sì	sì	sì	no	AI
18	C	sì	no	sì	sì	sì	sì	sì	no	IAT
19	A	sì	sì	sì	sì	sì	sì	sì	sì	A
20	C	sì	sì	sì	no	sì	no	sì	no	AI
21	A	sì	sì	sì	sì	sì	sì	sì	no	AI
22	E	sì	sì	sì	sì	sì	sì	sì	no	A
23	A	sì	sì	sì	no	sì	sì	sì	no	A
24	S	sì	sì	sì	sì	sì	sì	sì	no	IT
25	B	sì	sì	sì	sì	sì	sì	sì	no	IT
26	A	sì	sì	sì	sì	sì	sì	sì	no	IT
27	P	sì	sì	sì	sì	sì	sì	sì	sì	IADT
28	E	sì	sì	sì	sì	sì	sì	sì	no	AI
29	E	sì	no	sì	sì	sì	sì	sì	sì	AI
30	G	sì	no	sì	no	sì	sì	sì	no	AT
31	L	sì	sì	sì	sì	sì	sì	sì	no	DT
32	F	sì	sì	sì	no	sì	sì	sì	no	AI
33	D	sì	sì	sì	sì	sì	sì	sì	no	IAT
34	A	sì	no	sì	no	sì	sì	sì	no	I
35	M	sì	no	sì	sì	sì	sì	sì	no	IADT
36	C	sì	sì	sì	sì	sì	sì	sì	sì	DT
37	N	sì	sì	sì	sì	sì	sì	sì	no	AT
38	L	sì	sì	sì	sì	sì	sì	sì	no	IT
39	V	sì	sì	sì	no	sì	sì	sì	no	A

Capitolo 4

Conclusioni

Il lavoro di studio e progettazione di *SoundRise* ha permesso di ottenere un'applicazione sicuramente di facile utilizzo in grado di attirare l'interesse dei bambini e di creare in loro stimoli per comprenderne il comportamento. La realizzazione del sistema ha previsto più fasi:

- una fase di studio in cui si è definito lo scopo da raggiungere e delineato un percorso da seguire;
- una fase di ricerca delle soluzioni da adottare;
- una fase di sviluppo accompagnata da sperimentazioni delle varie versioni realizzate fino all'ottenimento di quella finale;
- una valutazione finale per misurare l'intuitività dell'interfaccia.

Dai test svolti, risultati positivi sono stati ottenuti per quanto riguarda la comprensione di aspetti dell'interazione con l'utente come la rappresentazione grafica dell'intensità, dell'altezza e della durata. L'utente infatti, oltre che capire il movimento del sole, capisce anche a cosa questo è legato, ovvero quale caratteristica della sua voce viene da esso rappresentata. Risultati meno soddisfacenti si sono ottenuti per quanto riguarda l'identificazione delle cinque vocali della lingua italiana. Un'ipotesi che potrebbe spiegare tali esiti risiede nel fatto che il database su cui il sistema di identificazione si basa non risulta essere sufficientemente robusto.

L'usabilità dell'applicazione è anche legata al modo in cui questa viene presentata, di conseguenza se accompagnata da una spiegazione delle funzionalità e di come si deve interagire, la comprensione dell'utente verso ogni aspetto di *SoundRise* aumenta e risulta anche più immediata lasciando così subito modo di iniziare pienamente il lavoro di esercizio e di apprendimento.

Dal punto di vista dell'estrazione delle feature vocali, un lavoro di miglioramento nel sistema di identificazione delle vocali si presenta come un primo obiettivo da poter raggiungere. Per migliorare le prestazioni dell'identificazione è possibile intervenire sul database utilizzato per il confronto tra i vettori di feature tramite l'impiego di molti più file audio contenenti le registrazioni delle vocali dai quali estrarre i vettori per il training del sistema. Le registrazioni devono essere realizzate con molti più soggetti, maschili e femminili, facendo in modo che ciascun soggetto ripeta ogni vocale con toni diversi, includendo sia le vocali chiuse che quelle aperte. Per rafforzare il database anche la qualità delle registrazioni dovrebbe essere migliorata.

Considerando gli external per le estrazioni di feature timbriche proposti con la libreria *timbreID*, ulteriori test sull'impiego di tali oggetti possono essere fatti con il fine di avere un miglioramento in merito all'affidabilità data dall'identificazione della vocale. Miglioramenti possono essere raggiunti attraverso combinazioni di più external rispetto a quelli impiegati nei test di analisi per lo sviluppo di SoundRise.

Oltre a questi interventi, un più accurato studio può interessare il filtraggio del segnale audio in ingresso al sistema, in modo da non includere nell'analisi informazione non utile.

Un intervento più netto consiste nella ricerca di una soluzione che sostituisca la libreria *timbreID* qui utilizzata con ad esempio un'analisi LPC (Linear Predictive Coding).

Considerando l'aspetto del feedback grafico uno studio più accurato può essere svolto in particolare per la modalità in cui si rappresentano in contemporanea tutte le feature. È necessario infatti rendere più comprensibile il contenuto informativo dato in questo caso dal feedback, in modo che l'utente sia in grado di cogliere ogni informazione rappresentata e riesca così a controllare tramite la propria voce ogni animazione con la quale il sole risponde. In merito all'associazione vocale-colore, possono essere tenuti in considerazione risultati derivanti da altri progetti di ricerca come ad esempio il progetto *Vowel Colours* [26] organizzato dal dott. Rob Drummond della Manchester Metropolitan University, iniziato il 31 agosto 2011 ed ancora attivo.

Un ulteriore lavoro di sviluppo è quello che vede l'utilizzo di SoundRise in ambito logopedistico, mantenendo la sua natura di applicazione-gioco in grado di catturare l'attenzione e l'interesse di chi la utilizza. Per realizzare ciò è necessario un forte lavoro di ricerca che interessi non solo la parte di analisi ed elaborazione audio ma anche quella di interazione con l'utente. Attraverso l'interesse e la collaborazione di logopedisti l'applicazione potrebbe divenire un utile strumento di aiuto verso bambini con deficit vocali, utilizzabile con o senza la presenza di un tutor da affiancare al bambino.

Un aspetto da espandere a livello di interazione può riguardare degli obiettivi da fissare all'utente

per stimolare meglio le sue capacità e mantenere l'attenzione nell'esercizio.

L'integrazione con la *Stanza Logo-Motoria* risulta una possibile tematica da evolvere per migliorare l'applicazione. É possibile infatti inserire dei task da superare per poter passare da una zona all'altra della stanza, utilizzando ad esempio anche un feedback audio oltre che grafico, o definire altri compiti simili sempre a livello di gioco.

Appendici

Appendice A

Strumenti software

Vengono di seguito descritti i principali strumenti di sviluppo impiegati per la realizzazione del lavoro di tesi: Pure Data ed EyesWeb.

Pure Data è il linguaggio di programmazione visuale utilizzato per la creazione dell'intera applicazione, sia per la parte di elaborazione audio che video.

EyesWeb è la piattaforma usata per estendere la multimodalità del sistema real-time e dell'interfacciamento allo stesso.

I due software sono caratterizzati da una programmazione grafica simile e sono entrambi gratuiti.

EyesWeb comunica con Pure Data tramite il protocollo di comunicazione OSC (Open Sound Control). Legge l'ingresso proveniente da una webcam e manda in uscita il segnale di controllo per Pure Data.

A.1 Pure Data

Pure Data (Pd) è un ambiente di programmazione grafica real-time per l'elaborazione audio e video sviluppato in linguaggio C. È un software libero creato per essere multiplatforma e perciò portabile. Esistono versioni per Win32, IRIX, GNU/Linux, BSD e MacOS X [27].

Il core di Pure Data è stato scritto ed è mantenuto da Miller S. Puckette. Include inoltre il lavoro di molti altri sviluppatori ed ha come supporto una community molto attiva [28].

Pure Data è prima di tutto un linguaggio di programmazione perché consente di realizzare algoritmi più o meno complessi come tutti gli altri linguaggi [29].

Di seguito vengono esposti le principali caratteristiche di Pure Data.

Ambiente grafico

L'interfaccia grafica è costruita usando un linguaggio visuale di patching. Le linee di codice, che descrivono le funzioni di un programma e come queste agiscono, vengono rimpiazzate con elementi visivi di varia natura detti "box" [30]. Questi box sono di quattro tipi: oggetti, messaggi, GUI e commenti. Non c'è dunque la necessità di scrivere il codice in un editor di testo, ma si realizzano delle patch combinando fra loro vari tipi di oggetti, definendo graficamente l'ordine con cui questi sono collegati fra loro [29]. Le patch vengono salvate in formato testuale, ma editate attraverso GUI.

Sistema modulare

Pd è dunque un sistema modulare. I vari moduli inviano e ricevono attraverso le interconnessioni, inlet e outlet, due tipi principali di informazioni: segnali audio e segnali di controllo. E' possibile eseguire combinazioni di moduli in patch. Queste sono memorizzate in formato testuale e normalmente costruite graficamente [28].

Funzionamento real-time

Il vero vantaggio di Pure Data è che funziona in real-time. Ciò significa che cambiamenti possono essere fatti nel programma anche se è in esecuzione e l'utente può vedere o ascoltare il risultati immediatamente [30].

Gli algoritmi sono quindi interattivi e i parametri possono essere modificati durante l'esecuzione. É anche possibile cambiare la struttura stessa di tali algoritmi mentre sono attivi, aggiungendo o rimuovendo moduli in modo semplice e intuitivo [29]. Moduli possono essere scritti e compilati separatamente e caricati a runtime. Pd comunica con il mondo esterno attraverso moduli speciali come "adc", "dac", "midiin", "midiout", "print" e attraverso gli elementi GUI come slider [28].

Aggiunta di external

Numerose estensioni e librerie aggiuntive sono disponibili. In particolare Gem, per rendering grafico [28]. Esiste ad ogni modo la possibilità di estendere facilmente Pd tramite la scrittura di classi oggetto dette "external" o di patch [27].

Modi di funzionamento

Due modi di lavoro caratterizzano Pd: edit mode, rappresentante quella che è l'elaborazione della patch in cui si creano i box, le connessioni e il layout, e run mode, in cui gli ingressi giungono ai vari box definendo quella che è l'interazione con gli oggetti (si può gestire la patch quando questa è in esecuzione). Il flusso dei messaggi è "istantaneo": i messaggi vengono elaborati allo stesso tempo logico della loro generazione [28].

Nel caso di elaborazione di segnali audio, è necessario attivare il motore DSP (Digital Signal Processor) per ascoltare il risultato delle operazioni sui segnali. Il motore DSP si occupa di elaborare il segnale in tempi rapidissimi e di permettere la sua trasformazione da digitale ad analogico e viceversa [29].

Il *sample rate* di Pure Data è di default 44100Hz.

Il programma in sé è sempre in esecuzione, non vi è alcuna separazione tra la scrittura del programma e la sua esecuzione ed ogni azione ha effetto nel momento in cui è completata [30].

A.2 EyesWeb

EyesWeb è una piattaforma aperta per la progettazione e lo sviluppo di interfacce e sistemi real-time multimodali, concepito, progettato e sviluppato da InfoMus Lab. Supporta un ampio numero di dispositivi di ingresso includendo sistemi di motion capture, vari tipi di telecamere professionali e low cost, interfacce di gioco (ad esempio Kinect e Wii), ingressi audio multicanale, input analogici.

Le uscite supportate includono audio multicanale, video, dispositivi analogici, piattaforme robotiche. Diversi standard vengono tollerati includendo OSC, MIDI, ASIO, Motion Capture, Matlab.

Il software EyesWeb include un ambiente di sviluppo, un sistema run-time distribuito per Windows, Linux e piattaforme mobile per creare applicazioni real-time distribuite o networked, ed un insieme di librerie di componenti software.

L'ambiente di sviluppo supporta il processo di progettazione di sistemi interattivi multimodali, consentendo agli utenti di costruire sistemi per mezzo di un linguaggio di programmazione visivo [31].

Essendo un ambiente di programmazione grafica i concetti fondamentali della programmazione risultano essere blocchi, connessioni tra i vari blocchi e patch.

I principali componenti [32] del sistema EyesWeb sono:

- i blocchi o muduli: rappresentano un'operazione che può essere eseguita sui dati. Esiste un set di blocchi forniti con EyesWeb ma ne possono essere creati altri dall'utente in C++. La comunicazione avviene tramite input, output e parametri;
- i datatype: conservano il valore dei dati scambiati tra blocchi. Un insieme di base è fornito con EyesWeb (esempi sono Matrici, messaggi Midi e Immagini);
- il manager: contiene le informazioni relative alle connessioni tra blocchi. Verifica la correttezza delle connessioni. E' responsabile della visualizzazione dei blocchi (input/output, nome, parametri, etc) e crea lo scheduler;
- lo scheduler: attiva periodicamente i blocchi. Gestisce un certo numero di blocchi. Chiama la funzione di esecuzione del blocco quando questo è chiamato.

Bibliografia

- [1] P. Lindamood. *ICDL Clinical Practice Guidelines - Chapter 24: Technologies to Facilitate Language, Sensory Processing, and Motor-Planning Capacities*. The Interdisciplinary Council on Developmental and Learning Disorders (ICDL), 2000.
- [2] C.S. Wang e C.C. Liu. A game-based learning content design framework for the elementary school children education. *Nano, Information Technology and Reliability (NASNIT), 2011 15th North-East Asia Symposium on*, pp. 53–57, 2011.
- [3] W. F. W. Ahmad, E. A. P. Akhir e S. Azmee. *Game-based learning courseware for children with learning disabilities*, volume 1, p. 1–4. IEEE, 2010.
- [4] James C. Lester, Brian A. Stone e Gary D. Stelling. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and UserAdapted Interaction*, 9(1):1–44, 1999.
- [5] R. Moreno e R. Mayer. Interactive multimodal learning environments. *Educational Psychology Review*, 19(3):309–326, 2007.
- [6] H. Fell, C. J. Cress, J. MacAuslan e L. Ferrier. visibabble for reinforcement of early vocalization. *Proceedings of the ACM SIGACCESS conference on Computers and accessibility ASSETS 04, (77-78)*:161–168, 2003.
- [7] K. Vicsi, P. Roach, A. Öster, Z. Kacic, F. Csatári, A. Sfakianaki e R. Veronik. A multilingual, multimodal, speech training system speco. *Proceedings of Eurospeech*, pp. 2807–2810, 2001.
- [8] Ali J. A. Soleymani, M. J. Mccutcheon e M. H. Southwood. *Design Of Speech Illumina Mentor (SIM) For Teaching Speech To The Hearing Impaired*, pp. 425–428. 1997.
- [9] P. Connors, A. Davis, G. Fortier, K. Gilley, B. Rundle, C. Soland e A. Tarachow. Participatory design: Classroom applications and experiences. *Proceedings of the International Conference of Phonetic Sciences*, 1999.
- [10] Accelerate learning |scientific learning global. Disponibile all'indirizzo: <http://www.scilearnglobal.com> (Ultima visita: 09 Luglio 2012).

- [11] S. G. Schar e H. Krueger. Using new learning technologies with multimedia. *Ieee Multimedia*, 7(3):40–51, 2000.
- [12] M. Riess Jones, R. Fay Richard e Arthur N. Popper. *Music Perception*, volume 36 di *Springer Handbook of Auditory Research*. 1st edizione, 2010.
- [13] A. Camurri, G. Volpe, S. Canazza, C. Canepa, A. Rodà, S. Zanolla e G. L. Foresti. *THE “STANZA LOGO–MOTORIA”: AN INTERACTIVE ENVIRONMENT FOR LEARNING AND COMMUNICATION*. 2010.
- [14] M. Randon. Soundrise: sviluppo e validazione di un'applicazione multimodale interattiva per la didattica basata sull'analisi di feature vocali. Tesi per Master, Università degli studi di Padova - Facoltà di Ingegneria, 2012.
- [15] *Pure Data (PD) Object Reference Glossary*. Disponibile all'indirizzo: <http://www.flexatone.net/docs/pdg.pdf> (Ultima visita: 09 Luglio 2012).
- [16] A. de Götzen, R. Marogna e F. Avanzini. Voice painter: un'interfaccia multimodale per dipingere con la voce. *Proc. of the Enactive Conference*, pp. 29–33, 2008.
- [17] T. H. Park. *Introduction to Digital Signal Processing: Computer Musically Speaking*, pp. 399–400. World Scientific, 2010.
- [18] E. Schubert, J. Wolfe, A. Tarnopolsky e N. South. Spectral centroid and timbre in complex, multiple instrumental textures. *Most*, pp. 654–657, 2004.
- [19] B. Prica e S. Ilíc. Recognition of vowels in continuous speech by using formants. *Facta Univ. Ser.: Elec. Energ.*, 23:379–393, 2010.
- [20] Madde e i formanti vocali - tertium auris. Disponibile all'indirizzo: <http://marcotonini.wordpress.com/2007/07/11/madde-e-i-formanti-vocali/> (Ultima visita: 09 Luglio 2012).
- [21] W. Brent. A timbre analysis and classification toolkit for pure data. *Proceedings of the International Computer*, pp. 2–7, 2010.
- [22] W. Brent. Cepstral analysis tools for percussive timbre identification. *Proceedings of the 3rd International Pure Data Convention*, 2009.
- [23] Pedro S. Gir. *Data Modeling for Metrology and Testing in Measurement Science*, p. 489. Birkhäuser Boston, 2009.
- [24] W. Brent. *Perceptually Based Pitch Scales in Cepstral Techniques for Percussive Timbre Identification*, pp. 3–6. 2009.
- [25] J. Simner, J. Ward, M. Lanz, A. Jansari, K. Noonan, L. Glover e David A. Oakley. Non-random associations of graphemes to colours in synaesthetic and non-synaesthetic populations. *Cognitive Neuropsychology*, 22(8):1069–1085, 2005.

-
- [26] www.vowelcolours.org: a research project. Disponibile all'indirizzo: <http://http://www.vowelcolours.org> (Ultima visita: 11 Luglio 2012).
- [27] Pure data - pd community site. Disponibile all'indirizzo: <http://puredata.info> (Ultima visita: 09 Luglio 2012).
- [28] M. Rath. *PD overview*. Dipartimento di Informatica. Università degli Studi di Verona.
- [29] F. Bianchi. *Inventare il suono con PURE DATA. Manuale introduttivo di musica elettronica*, 2010.
- [30] Pure data. Disponibile all'indirizzo: <http://www.flossmanuals.net/pure-data/> (Ultima visita: 09 Luglio 2012).
- [31] Casa paganini - infomus. Disponibile all'indirizzo: <http://www.eyesweb.org> (Ultima visita: 09 Luglio 2012).
- [32] D. Cirotteau. Eyesweb: an open platform for multimodal creation, October 2004.