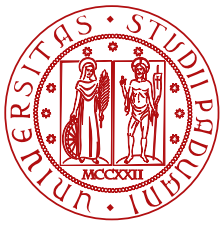

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

SOUNDRISE 2.0: Sviluppo di un sistema web-based di analisi vocale per supportare persone con disabilità uditive

Relatore
Prof. Sergio Canazza Targon

Laureando
Andrea Zanetti

Correlatore
Dott. Alessandro Fiordelmondo

ANNO ACCADEMICO 2023-2024

Data di laurea 24/09/2024

Sommario

La presente tesi si concentra sullo sviluppo di SoundRise 2.0, una rivisitazione moderna e web-based dell'applicazione SoundRise, originariamente sviluppata nel 2012 da Stefano Giusto e Marco Randon come progetto di tesi magistrale presso l'Università di Padova. SoundRise nasce come strumento didattico-terapeutico interattivo, pensato per supportare bambini con disabilità uditive nell'apprendimento del linguaggio e nell'allenamento vocale.

Il progetto originale del 2012 utilizzava Pure Data per l'elaborazione audio in tempo reale e offriva un'interfaccia grafica minimale. SoundRise 2.0 si propone di modernizzare e ampliare questa base, trasformando l'applicazione in una piattaforma web accessibile e visivamente coinvolgente.

Il presente lavoro si focalizza principalmente sull'implementazione degli algoritmi di analisi audio in tempo reale, con particolare attenzione all'estrazione del pitch e dell'intensità vocale. Questo lavoro include lo sviluppo della classe Microphone per la gestione dell'input audio, l'implementazione di algoritmi di pitch detection basati sull'autocorrelazione e il calcolo del Root Mean Square (RMS) per la determinazione dell'intensità vocale.

Il documento esplora inizialmente le sfide nell'educazione dei bambini sordomuti e l'importanza delle tecnologie assistive. Successivamente, presenta il progetto SoundRise, confrontando la versione originale con SoundRise 2.0 e delineando gli obiettivi e le innovazioni introdotte.

Una sezione significativa è dedicata alle tecnologie utilizzate, con particolare enfasi su Web Audio API e Three.js, fondamentali per l'analisi audio e la visualizzazione 3D rispettivamente. Il cuore della tesi risiede nella descrizione dettagliata dell'implementazione dell'analisi audio in tempo reale, includendo i fondamenti teorici e gli algoritmi sviluppati.

Questo lavoro si inserisce in un progetto collaborativo più ampio, che include lo sviluppo dell'interfaccia grafica 3D e l'implementazione del riconoscimento del timbro vocale, realizzati dai colleghi Gabriele Turetta e Riccardo Fila.

SoundRise 2.0 rappresenta così un'evoluzione significativa del progetto originale, offrendo uno strumento più moderno, accessibile e visivamente coinvolgente per supportare l'educazione e la terapia vocale dei bambini con disabilità uditive.

Indice

Elenco delle figure

Capitolo 1

L'educazione dei bambini sordomuti

1.1 Sfide nell'apprendimento e importanza dell'educazione multimodale

L'educazione dei bambini con disabilità uditive rappresenta una sfida significativa nel campo della pedagogia speciale. Questi bambini affrontano ostacoli unici nel loro percorso di apprendimento, in particolare per quanto riguarda lo sviluppo del linguaggio e delle abilità comunicative. La sordità o l'ipoacusia grave possono infatti interferire con l'acquisizione naturale del linguaggio parlato, che tipicamente avviene attraverso l'esposizione e l'imitazione dei suoni dell'ambiente circostante.

Una delle principali difficoltà risiede nella limitata capacità di percepire e discriminare i suoni del parlato, fondamentale per lo sviluppo fonologico e l'articolazione corretta delle parole. Ciò può portare a ritardi significativi nell'acquisizione del vocabolario e nella comprensione della struttura grammaticale della lingua. Inoltre, la mancanza di feedback uditivo rende arduo per questi bambini monitorare e modulare la propria produzione vocale, influenzando negativamente aspetti come l'intonazione, il ritmo e il controllo del volume della voce.

In questo contesto, l'approccio dell'educazione multimodale emerge come una strategia pedagogica cruciale. Questo metodo si basa sul principio di fornire input attraverso molteplici canali sensoriali, compensando così le limitazioni uditive e sfruttando le capacità residue del bambino. L'educazione multimodale integra stimoli visivi, tattili e propriocettivi per supportare l'apprendimento del linguaggio e lo sviluppo delle abilità comunicative [1].

L'importanza di questo approccio risiede nella sua capacità di adattarsi alle esigenze individuali di ogni bambino, offrendo diverse modalità di accesso all'informazione. Ad esempio, l'uso combinato del linguaggio dei segni, della lettura labiale e di supporti visivi può fornire un quadro più completo e accessibile del linguaggio parlato. Inoltre, l'integrazione di feed-

back tattili e propriocettivi può aiutare i bambini a sviluppare una maggiore consapevolezza dei movimenti articolatori necessari per la produzione del linguaggio.

L'educazione multimodale non solo facilita l'apprendimento del linguaggio, ma contribuisce anche allo sviluppo cognitivo generale e all'inclusione sociale dei bambini con disabilità uditive. Offrendo diverse modalità di espressione e comprensione, questo approccio promuove la flessibilità cognitiva e la capacità di problem-solving, competenze essenziali per il successo accademico e sociale.

1.2 Tecnologie assistive e sistemi interattivi per lo sviluppo delle abilità vocali

Negli ultimi decenni, l'avanzamento delle tecnologie digitali ha aperto nuove frontiere nel campo dell'educazione speciale, in particolare per quanto riguarda il supporto ai bambini con disabilità uditive. Le tecnologie assistive e i sistemi interattivi stanno emergendo come strumenti preziosi per integrare e potenziare i metodi tradizionali di insegnamento, offrendo nuove opportunità per lo sviluppo delle abilità vocali e linguistiche.

Le tecnologie assistive comprendono una vasta gamma di dispositivi e software progettati per migliorare le capacità funzionali delle persone con disabilità. Nel contesto dell'educazione dei bambini sordi o ipoacusici, queste tecnologie includono apparecchi acustici avanzati, impianti cocleari, sistemi di sottotitolazione in tempo reale e dispositivi di amplificazione sonora per l'ambiente scolastico.

Parallelamente, i sistemi interattivi stanno rivoluzionando il modo in cui i bambini con disabilità uditive possono esercitare e sviluppare le loro abilità vocali. Questi sistemi, spesso basati su software e applicazioni, offrono un ambiente di apprendimento coinvolgente e personalizzato. Utilizzando tecnologie come il riconoscimento vocale, l'analisi del suono in tempo reale e la visualizzazione grafica dei parametri vocali, questi sistemi possono fornire un feedback immediato e intuitivo sulla produzione vocale del bambino.

Un esempio di sistema interattivo è rappresentato da applicazioni come Speech Viewer, che trasformano i parametri vocali in elementi visivi animati. Queste applicazioni permettono ai bambini di "vedere" la loro voce, visualizzando aspetti come il tono, l'intensità e la durata del suono attraverso animazioni colorate o movimenti di oggetti sullo schermo. Questo approccio non solo rende l'esercizio vocale più coinvolgente, ma aiuta anche i bambini a comprendere meglio la relazione tra i movimenti articolatori e il suono prodotto [2].

Inoltre, l'uso di tecnologie di realtà aumentata (AR) e realtà virtuale (VR) sta aprendo nuove possibilità nel campo dell'educazione speciale. Questi strumenti possono creare ambienti immersivi che simulano situazioni di comunicazione reali, permettendo ai bambini di praticare le

loro abilità linguistiche in contesti sicuri e controllati. Un esempio in questo campo è ARTUR (articulation tutor), che utilizza la realtà aumentata per mostrare una rappresentazione 3D del tratto vocale, aiutando i bambini a visualizzare i movimenti articolatori necessari per produrre diversi suoni.

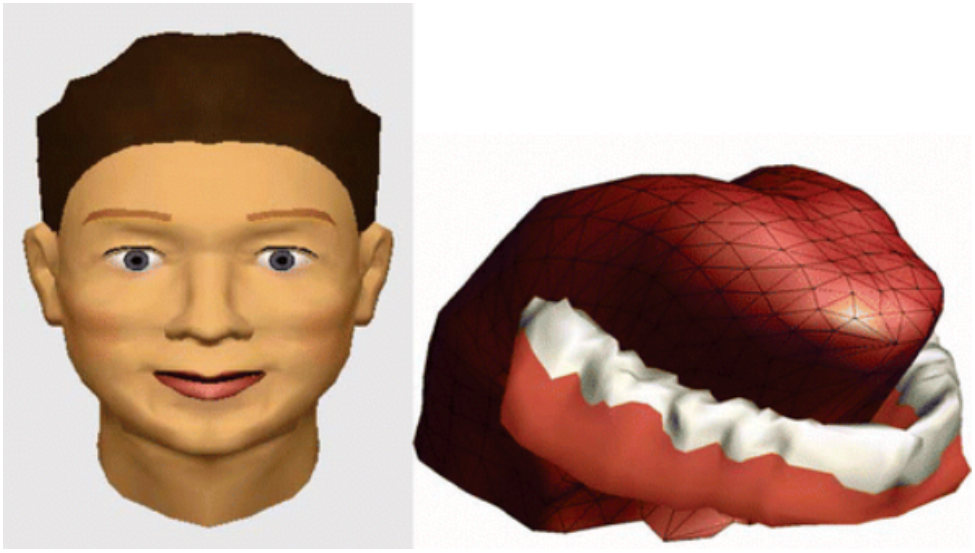

Figura 1.1: Schermata presa da ARTUR.

È importante sottolineare che l'efficacia di queste tecnologie dipende fortemente dalla loro integrazione in un programma educativo completo e dalla guida di educatori qualificati. Un approccio integrato, come quello offerto dal programma LENA (Language ENvironment Analysis), combina dispositivi di registrazione indossabili con software di analisi per monitorare l'ambiente linguistico del bambino e fornire feedback ai genitori e agli educatori.

In conclusione, l'evoluzione delle tecnologie assistive e dei sistemi interattivi sta aprendo nuove strade per supportare lo sviluppo delle abilità vocali nei bambini con disabilità uditive. Questi strumenti, quando utilizzati in modo appropriato e integrati in un approccio educativo olistico, hanno il potenziale di trasformare significativamente l'esperienza di apprendimento, promuovendo una maggiore autonomia e fiducia nelle proprie capacità comunicative.

Capitolo 2

Il progetto SoundRise

2.1 SoundRise: dalla versione originale a SoundRise 2.0

SoundRise, nato nel 2012 come progetto di tesi magistrale presso l'Università di Padova, è il risultato del lavoro congiunto di Stefano Giusto e Marco Randon [3]. Questa applicazione innovativa è stata concepita come uno strumento didattico-terapeutico, principalmente destinato a bambini con disabilità uditive, con l'obiettivo di fornire una rappresentazione visiva delle caratteristiche della voce.

Il funzionamento di SoundRise si basa sull'analisi in tempo reale di quattro parametri vocali fondamentali. Il primo è il pitch, o altezza, che rappresenta la frequenza fondamentale della voce. Il secondo parametro è l'intensità, che indica il volume o la forza della produzione vocale. La durata, terzo elemento analizzato, misura il tempo di emissione del suono. Infine, il timbro, nel contesto di SoundRise, viene utilizzato principalmente per il riconoscimento delle vocali. Questi quattro elementi costituiscono la base su cui l'applicazione costruisce la sua rappresentazione visiva del suono vocale.

L'originalità di SoundRise risiede nella sua capacità di tradurre questi parametri acustici in una rappresentazione visiva immediata e intuitiva. Il fulcro di questa visualizzazione è un sole animato, le cui caratteristiche variano in base all'input vocale dell'utente:

- L'altezza del sole sull'orizzonte corrisponde al pitch della voce.
- Le dimensioni del sole riflettono l'intensità del suono prodotto.
- L'apertura degli occhi e della bocca del sole indica la durata della produzione vocale.
- Il colore del sole cambia in base alla vocale riconosciuta, seguendo uno schema cromatico specifico.

Questa prima versione di SoundRise è stata sviluppata utilizzando Pure Data, un ambiente di programmazione grafica per l'elaborazione audio e video in tempo reale. Sebbene questa scelta offrisse potenti capacità di elaborazione del segnale, comportava anche alcune limitazioni in termini di portabilità e accessibilità dell'applicazione.

L'interfaccia utente di SoundRise, pur essendo funzionale, era relativamente semplice. Presentava una console di comando che permetteva di regolare vari parametri, come le scale di altezza e intensità, e di attivare o disattivare il filtro equalizzatore in ingresso. Gli utenti potevano anche scegliere tra quattro diversi sfondi per il paesaggio su cui veniva visualizzato il sole animato.

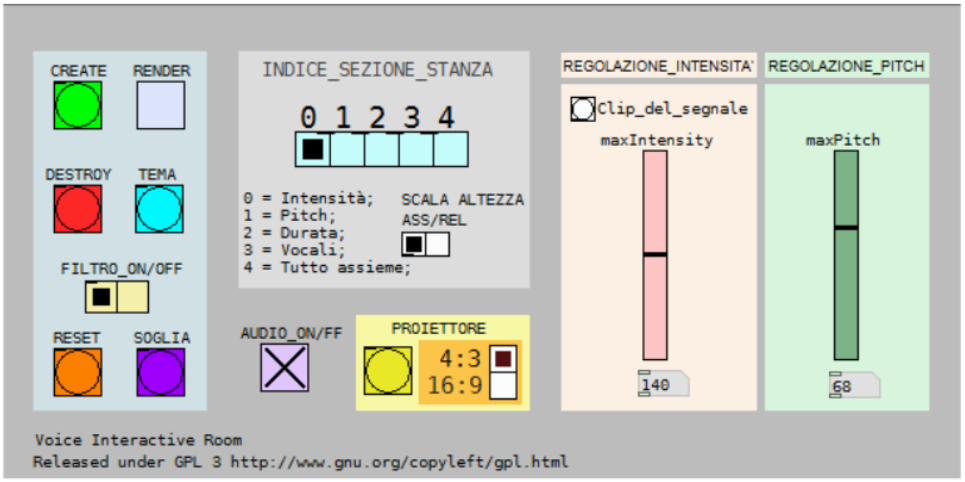

Figura 2.1: Console di comando del prototipo SoundRise.

Un aspetto particolarmente innovativo di SoundRise era la sua integrazione con la "Stanza Logo-Motoria", un ambiente sensorizzato che permetteva di analizzare il movimento del corpo e la mimica dell'utente, selezionando la caratteristica vocale da estrarre e rappresentare in base alla posizione fisica dell'utente nella stanza.

Nel 2022, dieci anni dopo la sua concezione iniziale, è emersa l'esigenza di sviluppare SoundRise 2.0. Questa nuova versione si propone di mantenere l'essenza e gli obiettivi originali di SoundRise, aggiornandone al contempo l'implementazione tecnologica per sfruttare le moderne possibilità offerte dallo sviluppo web e dalla grafica 3D. L'obiettivo rimane quello di fornire uno strumento efficace per l'educazione vocale e l'allenamento al linguaggio per bambini con disabilità uditive, ma con un'esperienza utente potenzialmente più coinvolgente e accessibile.

Questo percorso evolutivo da SoundRise a SoundRise 2.0 illustra come gli strumenti educativi specializzati possano beneficiare dell'innovazione tecnologica, mantenendo al contempo intatta la loro visione pedagogica originale.

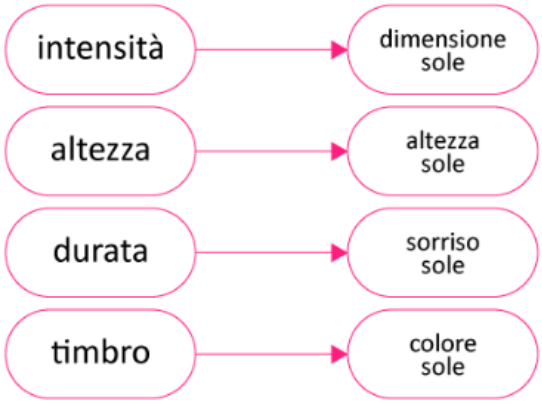

Figura 2.2: Mappatura delle features vocali nell'aspetto del sole.

2.2 Obiettivi, caratteristiche e innovazioni di SoundRise 2.0

Lo sviluppo di SoundRise 2.0 è stato guidato da una serie di obiettivi ambiziosi, miranti a superare le limitazioni della versione originale e a sfruttare appieno le potenzialità offerte dalle moderne tecnologie web. Il progetto si è posto come scopo principale quello di creare un'applicazione più accessibile, coinvolgente e versatile, mantenendo al contempo l'essenza educativa che caratterizzava il concept originale.

Un obiettivo fondamentale è stato quello di rendere l'applicazione indipendente dalla piattaforma, consentendone l'utilizzo su una vasta gamma di dispositivi. Questa scelta riflette la crescente importanza del "mobile learning" nell'educazione contemporanea, come evidenziato da Crompton e Burke (2018), che sottolineano come l'apprendimento attraverso dispositivi mobili possa aumentare il coinvolgimento degli studenti e facilitare l'accesso a risorse educative in qualsiasi momento e luogo [4].

SoundRise 2.0 si distingue per una serie di caratteristiche innovative che ne ampliano significativamente le potenzialità rispetto alla versione precedente:

1. Interfaccia grafica 3D: L'adozione di un ambiente tridimensionale rappresenta un salto qualitativo notevole. Questa scelta non solo rende l'applicazione visivamente più attraente per i giovani utenti, ma offre anche maggiori possibilità di rappresentazione delle caratteristiche sonore. L'ambiente 3D permette una visualizzazione più intuitiva e immersiva dei parametri vocali, facilitando la comprensione dei concetti acustici attraverso metafore visive più ricche e dettagliate.

2. Analisi audio in tempo reale: Sfruttando le moderne API web audio, SoundRise 2.0 è in grado di analizzare il segnale vocale con maggiore precisione e reattività. Questo permette una rappresentazione più fluida e accurata delle caratteristiche vocali, migliorando l'esperienza di feedback per l'utente.

3. Modalità di interazione multipla: L'applicazione può supportare diverse modalità di interazione, adattandosi sia agli input touch dei dispositivi mobili che ai tradizionali input di mouse e tastiera. Questa flessibilità rende SoundRise 2.0 utilizzabile in una varietà di contesti, dalla classe al setting terapeutico individuale.

4. Integrazione di elementi ludici: Riconoscendo l'importanza del gioco nell'apprendimento, specialmente per i più giovani, SoundRise 2.0 incorpora elementi di gamification, come modelli che ricordano la plastilina. Questi aspetti ludici non solo aumentano il coinvolgimento degli utenti, ma possono anche migliorare la ritenzione delle informazioni e la motivazione all'esercizio vocale, in linea con i principi dell'apprendimento basato sul gioco [5].

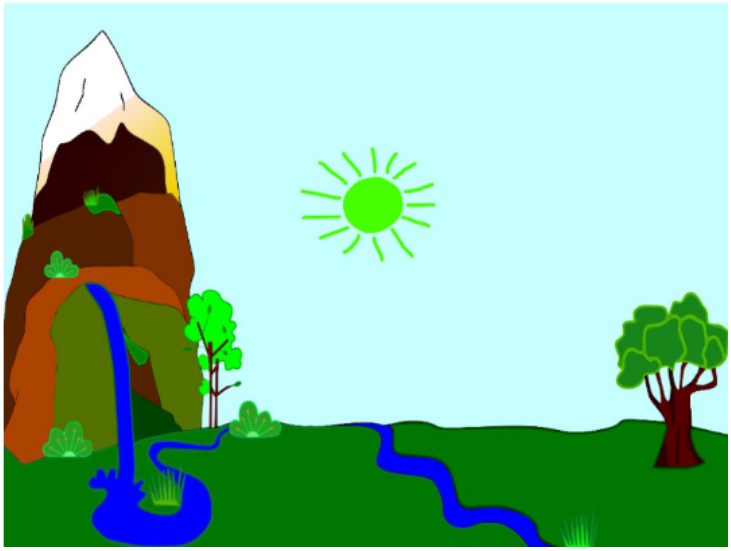

Figura 2.3: L'interfaccia originale di SoundRise.

Figura 2.4: L'interfaccia di SoundRise 2.0.

Un'innovazione significativa di SoundRise 2.0 è l'espansione del suo ambito di applicazione. Mentre la versione originale era principalmente focalizzata sull'assistenza a bambini con disabilità uditive, la nuova versione si propone come uno strumento versatile per l'educazione musicale e l'allenamento vocale in generale. Questa evoluzione riflette una comprensione più ampia del potenziale dell'applicazione, che può essere utilizzata non solo in contesti terapeutici, ma anche nell'educazione musicale mainstream e nello sviluppo delle competenze linguistiche.

SoundRise 2.0 rappresenta dunque un significativo passo avanti nel campo delle applicazioni educative interattive. Combinando tecnologie web all'avanguardia con principi pedagogici solidi, l'applicazione offre un ambiente di apprendimento stimolante e accessibile. Le sue caratteristiche innovative non solo migliorano l'esperienza utente, ma aprono anche nuove possibilità per l'insegnamento e l'apprendimento delle competenze vocali e musicali in un'ampia gamma di contesti educativi.

Capitolo 3

Tecnologie e architettura di SoundRise 2.0

3.1 Web Audio API per l'analisi audio in tempo reale

Web Audio API rappresenta un significativo avanzamento nell'elaborazione audio all'interno dei browser web moderni. Questa interfaccia di programmazione, standardizzata dal World Wide Web Consortium (W3C), offre un sistema modulare e ad alte prestazioni per la manipolazione e la sintesi di contenuti audio in tempo reale [6].

L'architettura di Web Audio API si basa su un grafo di nodi audio interconnessi, ciascuno dei quali rappresenta una specifica funzionalità di elaborazione o routing del segnale. Questo paradigma di progettazione modulare consente la costruzione di catene di elaborazione audio complesse e flessibili, adattabili a una vasta gamma di applicazioni, dalla semplice riproduzione audio alla sintesi e analisi avanzata del suono.

Il concetto fondamentale nell'utilizzo del Web Audio API è l'AudioContext, che funge da ambiente di esecuzione per tutte le operazioni audio. All'interno di questo contesto, gli sviluppatori possono creare e interconnettere vari tipi di nodi audio, tra cui:

1. AudioSourceNode per la generazione o l'acquisizione di segnali audio;
2. AudioDestinationNode per l'output audio finale;
3. AudioParam per il controllo parametrico in tempo reale;
4. AnalyserNode per l'estrazione di informazioni nel dominio delle frequenze e del tempo.

Un aspetto particolarmente rilevante del Web Audio API è la sua capacità di gestire l'audio in modo asincrono e in tempo reale. Questo è reso possibile attraverso l'uso di buffer audio di dimensioni ridotte e tecniche di scheduling precise, che consentono una latenza minima e una risposta reattiva anche in applicazioni audio complesse [7].

L'API offre inoltre funzionalità avanzate per l'analisi spettrale del segnale audio, cruciali per applicazioni di riconoscimento vocale e analisi musicale. L'AnalyserNode, in particolare, permette l'accesso diretto ai dati della Fast Fourier Transform (FFT) del segnale audio, consentendo implementazioni efficienti di visualizzazioni dello spettro e algoritmi di estrazione delle caratteristiche audio.

Dal punto di vista dell'ingegneria del software, il Web Audio API presenta vantaggi significativi in termini di portabilità e scalabilità. La sua natura cross-platform consente lo sviluppo di applicazioni audio avanzate che possono essere eseguite su una vasta gamma di dispositivi e browser, senza la necessità di plugin o software aggiuntivi. Questo aspetto è particolarmente rilevante nel contesto dello sviluppo web moderno, dove la compatibilità multipiattaforma è spesso un requisito fondamentale.

3.2 Three.js per la rappresentazione grafica 3D

Nel contesto di SoundRise 2.0, Gabriele Turetta si è concentrato sullo sviluppo dell'interfaccia grafica tridimensionale, un elemento fondamentale per offrire un'esperienza immersiva e coinvolgente all'utente. Per realizzare questo obiettivo, Turetta ha fatto ampio uso di Three.js, una potente libreria JavaScript che semplifica la creazione e la manipolazione di grafica 3D nel browser.

Turetta ha sfruttato la versatilità di Three.js e la sua capacità di utilizzare l'accelerazione hardware attraverso WebGL per creare un ambiente tridimensionale dinamico. L'architettura del suo progetto si basa sui tre componenti principali di Three.js: scene, camera e renderer, che fungono rispettivamente da contenitore per gli oggetti 3D, punto di vista dell'osservatore e motore di rendering.

Per l'inizializzazione della scena 3D, Turetta ha implementato la funzione `init()`, responsabile di configurare gli elementi fondamentali dell'ambiente tridimensionale. Questa funzione si occupa di creare la scena, impostare il renderer con parametri ottimizzati per la qualità visiva e le prestazioni, definire la camera prospettica e gestire l'illuminazione ambientale attraverso una mappa HDR (High Dynamic Range).

L'importazione dei modelli 3D è stata gestita attraverso l'utilizzo del `GLTFLoader` di Three.js. Turetta ha integrato nella scena modelli tridimensionali del sole, delle nuvole e delle colline, precedentemente creati con il software di modellazione Blender. Questi elementi sono stati opportunamente posizionati e scalati all'interno dell'ambiente virtuale per creare un paesaggio coerente e visivamente accattivante.

Il cuore dell'interattività dell'interfaccia risiede nel loop di animazione, implementato nella funzione `animate()`. In questa fase, Turetta ha predisposto il sistema per ricevere dati di

intensità (RMS) e pitch, che nel progetto completo sarebbero derivati dall'input vocale. Ha implementato la logica per mappare questi dati rispettivamente sulle dimensioni e sull'altezza del sole. Inoltre, ha creato un sistema di regolazione dinamica della luminosità della scena basato sulla posizione verticale del sole, simulando in modo realistico la transizione tra il giorno e la notte.

Per aumentare l'immersività dell'esperienza, Turetta ha introdotto effetti visivi avanzati, come un cielo stellato che appare gradualmente quando il sole scende sotto una determinata soglia. Questa caratteristica contribuisce a creare un ambiente più coinvolgente e reattivo.

Per ottimizzare le prestazioni, nonostante la complessità della scena, Turetta ha fatto ricorso a tecniche avanzate come il texture baking e una gestione efficiente delle sorgenti luminose. Queste strategie hanno permesso di mantenere un frame rate elevato, essenziale per un'esperienza utente fluida e reattiva.

Figura 3.1: Immagine presa dal lavoro di Gabriele Turetta: sopra il sole all'altezza minima, con il cielo di stelle attivato, sotto il sole all'estremo superiore, con l'esposizione della scena al massimo

3.3 Algoritmi per il riconoscimento vocale

Il riconoscimento vocale costituisce un elemento cruciale nello sviluppo di SoundRise 2.0, consentendo l'analisi in tempo reale delle caratteristiche della voce dell'utente. Questa funzionalità si basa su algoritmi avanzati di elaborazione del segnale audio, implementati dal collega Riccardo Fila specificamente per questa applicazione.

Il fulcro del sistema di riconoscimento è rappresentato dall'implementazione della Codifica Predittiva Lineare (LPC), una tecnica ben consolidata nell'ambito dell'analisi del parlato. Fila ha adattato e ottimizzato questo metodo per l'ambiente web di SoundRise 2.0, permettendo l'estrazione delle principali caratteristiche spettrali della voce, con particolare attenzione alle frequenze formanti, fondamentali per l'identificazione dei diversi fonemi.

L'algoritmo LPC implementato in SoundRise 2.0 opera attraverso una serie di fasi sequenziali. Inizialmente, il segnale audio viene sottoposto a un processo di sotto-campionamento, riducendo la frequenza di campionamento a 10 kHz. Questa operazione ottimizza il carico computazionale mantenendo al contempo le informazioni spettrali essenziali per il riconoscimento delle vocali. Successivamente, al segnale viene applicata una finestra di Hamming, migliorando l'analisi spettrale attraverso l'attenuazione degli effetti di discontinuità ai bordi del segmento audio analizzato.

Una volta preparato il segnale, l'algoritmo procede con il calcolo dell'autocorrelazione a breve termine, fase cruciale per la determinazione dei coefficienti LPC. Questi coefficienti vengono calcolati utilizzando il metodo di Durbin, un approccio ricorsivo che garantisce stabilità e efficienza computazionale. Fila ha scelto un ordine di predizione di 15 per l'analisi LPC, bilanciando accuratezza nella rappresentazione spettrale e costo computazionale.

La fase successiva dell'algoritmo prevede l'estrazione delle frequenze formanti dai coefficienti LPC. Questo processo viene realizzato attraverso la fattorizzazione del polinomio A(z) nelle sue radici complesse, utilizzando il metodo di Durand-Kerner. Le radici così ottenute vengono poi convertite in frequenze e larghezze di banda, permettendo l'identificazione delle formanti vocali.

Un aspetto innovativo dell'implementazione di Fila risiede nell'ottimizzazione di questi algoritmi per l'esecuzione in tempo reale in un ambiente web, sfruttando le potenzialità della Web Audio API. Questo ha richiesto un'attenta calibrazione dei parametri e l'implementazione di strategie per ridurre il carico computazionale, come il riutilizzo dei risultati intermedi del sotto-campionamento.

Il sistema si concentra principalmente sul riconoscimento delle vocali, essendo queste particolarmente rilevanti per l'educazione alla pronuncia. Le frequenze formanti estratte vengono confrontate con un set di valori di riferimento per le vocali italiane, permettendo così l'identificazione del fonema pronunciato. Questo confronto viene effettuato considerando le prime

due formanti (F1 e F2), sufficienti nella maggior parte dei casi per una discriminazione affidabile delle vocali, come evidenziato da Fant nel suo lavoro seminale sulla teoria acustica della produzione del parlato [8].

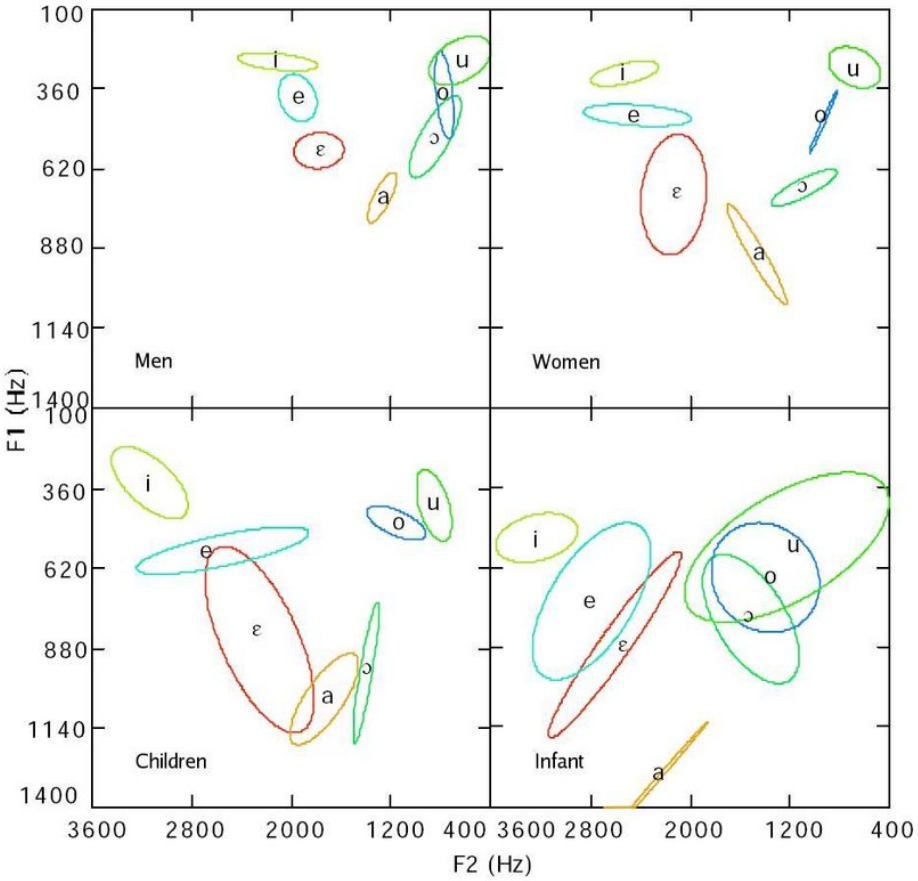

Figura 3.2: Le regioni di esistenza delle vocali, in funzione delle prime due formanti.

Capitolo 4

Analisi audio in tempo reale

4.1 Fondamenti teorici dell'analisi del segnale vocale

L'analisi del segnale vocale è un campo complesso che coinvolge diversi aspetti della fisica acustica e dell'elaborazione del segnale. Per comprendere appieno il funzionamento di SoundRise 2.0, è essenziale esaminare i principi teorici alla base dell'analisi del pitch e dell'intensità vocale.

4.1.1 Pitch detection

Il pitch, o altezza tonale, è una caratteristica percettiva del suono correlata alla frequenza fondamentale dell'onda sonora. Nel contesto della voce umana, il pitch è determinato principalmente dalla frequenza di vibrazione delle corde vocali.

Dal punto di vista fisico, il segnale vocale può essere modellato come una somma di sinusoidi, dove la frequenza più bassa corrisponde alla frequenza fondamentale (F_0), mentre le altre sono i suoi multipli interi, chiamati armoniche. Matematicamente, questo può essere espresso come:

$$s(t) = \sum_{k=1}^{\infty} A_k \cdot \sin(2\pi k F_0 t + \varphi_k)$$

dove A_k è l'ampiezza della k-esima armonica, F_0 è la frequenza fondamentale, e φ_k è la fase.

La rilevazione del pitch è un processo di stima di questa frequenza fondamentale. Esistono diversi metodi per effettuare questa stima, ma uno dei più robusti e ampiamente utilizzati è l'autocorrelazione.

L'autocorrelazione $R(\tau)$ di un segnale discreto $x(n)$ è definita come:

$$R(\tau) = \sum_{n=0}^{N-1} x(n)x(n+\tau)$$

dove N è il numero di campioni e τ è il ritardo.

In SoundRise 2.0, l'implementazione dell'autocorrelazione nella funzione `getPitch()` segue questo principio, calcolando:

$$c[n] = \sum_{m=0}^{\text{SIZE}-n} \text{timebuf}[m] \cdot \text{timebuf}[m+n]$$

Questo metodo sfrutta la periodicità intrinseca dei segnali vocali. Per un segnale periodico, la funzione di autocorrelazione mostrerà dei picchi a intervalli corrispondenti al periodo fondamentale. Il primo picco significativo dopo lo zero corrisponde al periodo del pitch, e l'inverso di questo periodo fornisce la frequenza fondamentale.

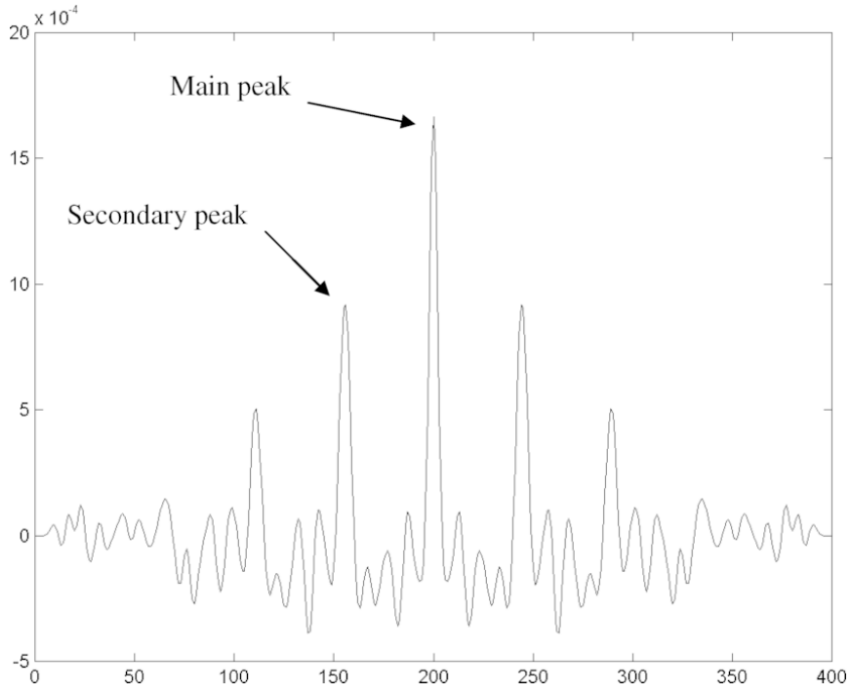

Figura 4.1: Funzione di autocorrelazione nel rilevatore di intonazione e pitch.

4.1.2 Calcolo dell'intensità (RMS)

L'intensità del segnale vocale è un'altra caratteristica fondamentale che fornisce informazioni sull'energia acustica prodotta dal parlato. In acustica, l'intensità è definita come la potenza per unità di area:

$$I = \frac{P}{A}$$

dove I è l'intensità, P è la potenza acustica, e A è l'area attraverso cui l'onda sonora si propaga.

Tuttavia, nella pratica dell'elaborazione digitale del segnale, l'intensità viene spesso approssimata utilizzando il valore RMS (Root Mean Square) dell'ampiezza del segnale. Il valore RMS fornisce una misura dell'energia media del segnale ed è definito come:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x[i]^2}$$

dove $x[i]$ rappresenta l'i-esimo campione del segnale e N è il numero totale di campioni.

In SoundRise 2.0, il calcolo RMS è implementato nella funzione `getRms()`:

$$\text{rms} = \sqrt{\frac{\sum_{i=0}^{\text{arr.length}-1}(\text{arr}[i]^2)}{\text{arr.length}}}$$

Il valore RMS è particolarmente utile nell'analisi del segnale vocale per diverse ragioni:

1. È direttamente correlato all'energia del segnale, fornendo una misura oggettiva dell'intensità vocale.

2. È meno sensibile alle fluttuazioni rapide dell'ampiezza rispetto ai valori di picco, offrendo una rappresentazione più stabile dell'intensità percepita.

3. È computazionalmente efficiente, rendendolo adatto per l'elaborazione in tempo reale.

L'implementazione di queste tecniche di analisi del segnale vocale in SoundRise 2.0 permette di estrarre informazioni cruciali sulla voce dell'utente in tempo reale. Queste informazioni vengono poi utilizzate per generare un feedback visivo immediato, creando un'esperienza interattiva che supporta efficacemente sia l'apprendimento vocale che le applicazioni terapeutiche nel campo della logopedia.

4.2 Implementazione della classe Microphone

La classe Microphone è il componente centrale per l'analisi audio in tempo reale in SoundRise 2.0. Questa classe gestisce l'acquisizione, la pre-elaborazione e l'analisi del segnale audio proveniente dal microfono dell'utente. Esamineremo in dettaglio le componenti principali di questa classe, le loro funzionalità e le motivazioni dietro le scelte implementative.

4.2.1 Acquisizione e pre-elaborazione del segnale audio

Il costruttore della classe Microphone inizializza l'ambiente audio:

```
constructor(){
    this.initialized = false;
    navigator.mediaDevices.getUserMedia({audio:true})
    .then(function(stream){
        this.audioContext = new AudioContext();
        this.microphone = this.audioContext.createMediaStreamSource(stream)
            ;
        this.analyser = this.audioContext.createAnalyser();
        this.analyser.fftSize = 4096;
        const bufferLength = this.analyser.frequencyBinCount;
        this.dataArray = new Uint8Array(bufferLength);
        this.microphone.connect(this.analyser);
        this.initialized = true;
    }.bind(this)).catch(function(err){
        alert(err);
    });
}
```

In questo codice, diversi aspetti meritano un'analisi approfondita:

1. Utilizzo di AudioContext: Questa API Web fornisce un ambiente flessibile per la manipolazione audio in tempo reale, essenziale per le nostre esigenze di analisi vocale.

2. Dimensione FFT: La scelta di una dimensione FFT di 4096 campioni è un compromesso cruciale. Una FFT più grande offre una migliore risoluzione in frequenza, utile per un'analisi precisa del pitch, ma introduce una maggiore latenza. Con una frequenza di campionamento standard di 44.1 kHz, questa dimensione corrisponde a circa 93 ms di audio, un valore che bilancia la precisione dell'analisi con la reattività dell'interfaccia utente.

3. Buffer audio: Il bufferLength viene impostato a metà della dimensione FFT (2048 campioni). Questa scelta ottimizza l'uso della memoria e la velocità di elaborazione, mantenendo una quantità di dati sufficiente per l'analisi accurata sia del pitch che dell'intensità.

Il metodo `getSamples()` acquisisce e normalizza i campioni audio:

```
getSamples() {
    this.analyser.getByteTimeDomainData(this.dataArray);
    let normSamples = [...this.dataArray].map(e => e/128 - 1);
```

```
    return normSamples;
}
```

Questo metodo utilizza `getByteTimeDomainData()` per ottenere i dati nel dominio del tempo, normalizzandoli poi nell'intervallo [-1, 1].

4.2.2 Algoritmo di pitch detection

L'algoritmo di pitch detection è implementato nella funzione `getPitch()`:

```
getPitch(timebuf) {
    let SIZE = timebuf.length;
    let r1=0, r2=SIZE-1, thres=0.2;
    for (let i=0; i<SIZE/2; i++)
        if (Math.abs(timebuf[i])<thres) { r1=i; break; }
    for (let i=1; i<SIZE/2; i++)
        if (Math.abs(timebuf[SIZE-i])<thres) { r2=SIZE-i; break; }
    timebuf = timebuf.slice(r1, r2);
    SIZE = timebuf.length;

    const c = new Array(SIZE).fill(0);
    for (let i=0; i<SIZE; i++)
        for (let j=0; j<SIZE-i; j++)
            c[i] = c[i] + timebuf[j]*timebuf[j+i];

    let d=0; while (c[d]>c[d+1]) d++;
    let maxval=-1, maxpos=-1;
    for (let i=d; i<SIZE; i++) {
        if (c[i] > maxval) {
            maxval = c[i];
            maxpos = i;
        }
    }

    let T0 = maxpos;
    let x1=c[T0-1], x2=c[T0], x3=c[T0+1];
    let a = (x1 + x3 - 2*x2)/2;
    let b = (x3 - x1)/2;
    if (a) T0 = T0 - b/(2*a);

    return 44100/T0;
}
```

Questo algoritmo implementa il metodo dell'autocorrelazione discusso nella sezione teorica. Inizia con una fase di pre-elaborazione per identificare i punti di attraversamento dello zero, riducendo l'influenza del rumore di fondo. Successivamente, calcola la funzione di autocorrelazione e identifica il picco massimo, raffinando la stima attraverso un'interpolazione parabolica.

4.2.3 Calcolo dell'intensità RMS

L'intensità del segnale vocale in Soundrise 2.0 viene quantificata attraverso il calcolo del valore RMS (Root Mean Square), implementato nella funzione `getRMS()`. Questa funzione opera su un array di campioni audio, fornendo una misura statistica robusta dell'ampiezza del segnale. L'implementazione è la seguente:

```
getRMS(arr){
    let rms = 0;
    for(let i = 0; i < arr.length; i++){
        rms += (arr[i] * arr[i]) / arr.length;
    }
    return rms;
}
```

L'algoritmo itera attraverso l'array di input, accumulando la somma dei quadrati dei campioni, normalizzata per la lunghezza dell'array. Questa normalizzazione viene effettuata dividendo ogni termine quadratico per la lunghezza dell'array all'interno del ciclo, piuttosto che alla fine del calcolo. Questa scelta implementativa può offrire vantaggi in termini di precisione numerica, specialmente quando si lavora con array di grandi dimensioni, riducendo il rischio di overflow aritmetico.

È importante notare che questa implementazione assume che i campioni di ingresso siano già normalizzati nell'intervallo $[-1, 1]$, coerentemente con l'output della funzione `getSamples()`. Questa normalizzazione preliminare semplifica l'interpretazione del risultato RMS, che sarà anch'esso compreso tra 0 e 1.

4.3 Visualizzazione real-time delle caratteristiche vocali

Al fine di avere un banco di prova per la classe Microphone, ho implementato un sistema di visualizzazione in tempo reale delle caratteristiche vocali, fungendo da banco di prova per la classe Microphone. Questo modulo crea un'interfaccia grafica interattiva che permette di osservare visivamente le proprietà del segnale audio catturato, fornendo un feedback immediato sulle funzionalità di analisi implementate.

L'inizializzazione del visualizzatore avviene attraverso la funzione `main()`:

```
function main(){
    const canvas = document.getElementById('myCanvas');
    const ctx = canvas.getContext('2d');
    canvas.width = window.innerWidth;
    canvas.height = window.innerHeight;

    class Bar {
        constructor(x, y, width, height, color){
            this.x = x;
            this.y = y;
            this.width = width;
            this.height = height;
            this.color= color;
        }

        update(micInput){
            this.height = micInput * 1000;
        }

        draw(context){
            context.fillStyle = this.color;
            context.fillRect(this.x, this.y, this.width, this.height);
        }
    }

    const microphone = new Microphone();
    let bars = [];
    let barWidth = canvas.width / 256;
}
```

Questa funzione configura l'ambiente di visualizzazione, creando un canvas che occupa l'intera finestra del browser. Definisce inoltre una classe Bar per rappresentare graficamente le componenti spettrali del segnale audio.

La creazione delle barre spettrali avviene attraverso la funzione `createBars()`:

```
function createBars(){
    for(let i = 0; i < 256; i++){
        let color = 'hsl(' + i * 2 + ', 100%, 50%)';
        bars.push(new Bar(i * barWidth, canvas.height/2, 1, 20, color));
    }
}
createBars();
```

Questa funzione genera 256 istanze della classe Bar, ciascuna rappresentante una componente spettrale del segnale audio. Il colore di ogni barra è determinato dinamicamente utilizzando lo spazio colore HSL, creando un gradiente visivo che facilita l'interpretazione dello spettro.

Il cuore della visualizzazione in tempo reale è implementato nella funzione `animate()`:

```
function animate(){
    if(microphone.initialized){
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        const samples = microphone.getSamples();
        bars.forEach(function(bar, i){
            bar.update(samples[i]);
            bar.draw(ctx);
        });
    }
    requestAnimationFrame(animate);
}
animate();
```

Questo loop di animazione, eseguito attraverso requestAnimationFrame, aggiorna continuamente la visualizzazione. Ad ogni frame:

1. Il canvas viene pulito per preparare il nuovo rendering.
2. Vengono acquisiti i campioni audio più recenti attraverso `microphone.getSamples()`.
3. Ogni barra viene aggiornata e ridisegnata in base al valore corrispondente del campione audio.

L'utilizzo di requestAnimationFrame assicura una sincronizzazione ottimale con il refresh del display, garantendo una visualizzazione fluida e performante.

Questo sistema di visualizzazione fornisce un feedback visivo immediato sulle capacità di analisi della classe Microphone. La rappresentazione grafica delle componenti spettrali del segnale audio permette di osservare in tempo reale le variazioni nell'input vocale, offrendo un utile strumento di debug e verifica durante lo sviluppo di Soundrise 2.0.

La scelta di rappresentare 256 componenti spettrali offre un buon compromesso tra dettaglio e performance, permettendo di visualizzare una gamma sufficientemente ampia di frequenze mantenendo al contempo una risposta fluida dell'interfaccia.

Questo visualizzatore, sebbene non faccia parte dell'interfaccia utente finale di Soundrise 2.0, ha giocato un ruolo cruciale nel processo di sviluppo, permettendo di validare visivamente l'accuratezza e la reattività dell'analisi audio in tempo reale implementata nella classe Microphone.

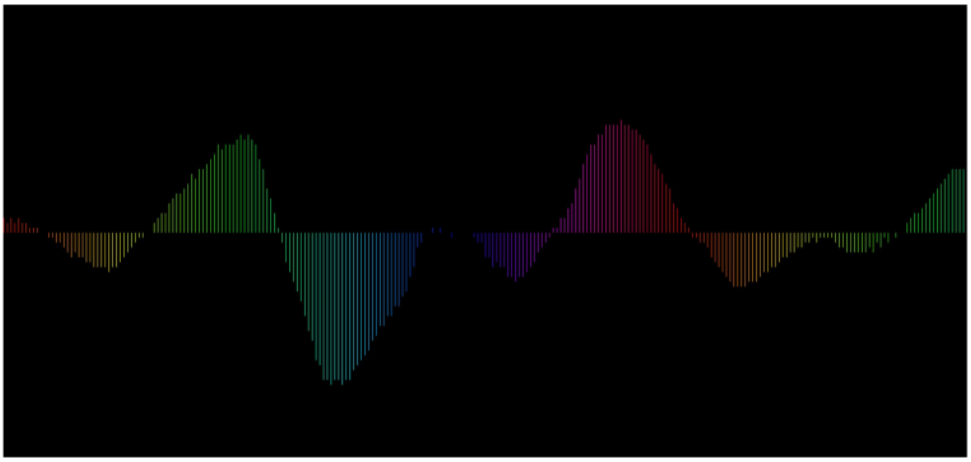

Figura 4.2: Visualizzazione in tempo reale delle variazioni nell'input vocale.

Capitolo 5

Conclusioni e sviluppi futuri

Il presente lavoro ha portato alla realizzazione di SoundRise 2.0, una versione completamente rinnovata dell'applicazione originale sviluppata nel 2012. Questo progetto, frutto della collaborazione tra me e i colleghi Riccardo Fila e Gabriele Turetta, ha permesso di trasformare SoundRise in un'applicazione web moderna e interattiva, mantenendo al contempo gli obiettivi didattici e riabilitativi originali. SoundRise 2.0 rappresenta un significativo passo avanti rispetto alla versione precedente. L'applicazione può essere accessibile via web, rendendola fruibile su una vasta gamma di dispositivi senza necessità di installazione. L'interfaccia grafica è stata completamente ridisegnata, passando da una visualizzazione 2D minimale a un ambiente 3D coinvolgente e accattivante, particolarmente adatto al pubblico giovane a cui l'applicazione si rivolge.

Le potenzialità di SoundRise 2.0 sono dunque notevoli e diverse direzioni di sviluppo futuro si profilano all'orizzonte. In primo luogo, sarà fondamentale integrare e ottimizzare le diverse componenti dell'applicazione, assicurando un funzionamento fluido e coerente su diverse piattaforme e dispositivi. Un'area di miglioramento riguarda l'analisi delle caratteristiche vocali. L'implementazione di tecniche di machine learning potrebbe aumentare la precisione nel riconoscimento del timbro e delle vocali, rendendo l'applicazione più accurata e personalizzabile per le esigenze di ciascun utente. Sul fronte dell'interfaccia utente, si potrebbe implementare un sistema di progressione e gamification, con livelli di difficoltà crescente e ricompense virtuali. Questo approccio potrebbe aumentare il coinvolgimento degli utenti più giovani, incentivandoli a praticare regolarmente gli esercizi vocali.

SoundRise 2.0 ha il potenziale per diventare uno strumento prezioso in diversi contesti. Nel campo dell'educazione musicale, potrebbe essere utilizzato come supporto per l'apprendimento del canto, aiutando gli studenti a visualizzare e comprendere le caratteristiche della propria voce. In ambito logopedico, l'applicazione potrebbe assistere nella riabilitazione di pazienti con disturbi del linguaggio o dell'udito. Infine, l'applicazione potrebbe trovare impiego anche in

contesti di educazione speciale, supportando l'inclusione di studenti con diverse abilità. Come evidenziato da una recente revisione della letteratura, "le tecnologie assistive basate su feedback multimodali possono significativamente migliorare l'accessibilità e l'efficacia dei percorsi educativi per studenti con bisogni speciali" [9]. SoundRise 2.0, con la sua interfaccia visiva intuitiva, si allinea perfettamente con questo principio.

Bibliografia

[1] H. Knoors e M. Marschark, *Teaching deaf learners: Psychological and developmental foundations*. Oxford University Press, 2014.

[2] A. M. Öster, *Computer-based speech therapy using visual feedback with focus on children with profound hearing impairments*. Stockholm: KTH, 2006.

[3] S. Giusto, «SoundRise: studio e progettazione di un'applicazione multimodale interattiva per la didattica basata sull'analisi di feature vocali,» Master's thesis, Università degli studi di Padova, 2012.

[4] H. Crompton e D. Burke, «The use of mobile learning in higher education: A systematic review,» *Computers & Education*, 2018.

[5] J. L. Plass, B. D. Homer e C. K. Kinzer, «Foundations of game-based learning,» *Educational Psychologist*, 2015.

[6] W3C. «Web Audio API.» (2022), indirizzo: `https://www.w3.org/TR/webaudio/`.

[7] C. Wilson. «A Tale of Two Clocks - Scheduling Web Audio with Precision.» (2013), indirizzo: `https://web.dev/audio-scheduling/`.

[8] G. Fant, *Acoustic Theory of Speech Production*. Mouton, 1960.

[9] F. Stasolla, A. O. Caffò, L. Picucci e A. Bosco, «Assistive technology for promoting choice behaviors in three children with cerebral palsy and severe communication impairments,» *Research in Developmental Disabilities*, 2013.