Progettazione e realizzazione algoritmo di Voice Activity Detection

Alessandra Pastore

14 Giugno 2021

ABSTRACT

Il termine Voice Activity Detection (VAD) si riferisce all'elaborazione di segnali vocali per determinare se essi contengano o meno il parlato. Un algoritmo VAD usa normalmente regole di decisione basate sulle caratteristiche del segnale stesso per migliorare la propria performance e rendere il risultato finale indipendente dal segnale processato. Propongo un algoritmo implementato in MATLAB che utilizza caratteristiche a breve termine come la Spectral Flatness Measure (SF) e la Short-term Energy.

1. INTRODUZIONE

I segnali audio presi in considerazione sono segnali digitali audio modo, in formato PCM, che si assumono generati in tempo reale. La pacchettizzazione del segnale eseguita dal trasmettitore divide il segnale in pacchetti da 160 campioni l'uno, ognuno corrispondente a 20ms di audio. L'algoritmo proposto prende in considerazione 2 campioni precedenti all'attuale per eseguire un'elaborazione che permetta una scelta esaustiva riguardo al contenuto di quest'ultimo e poter generare un file di testo in output che contenga 1 o 0 a seconda che il pacchetto debba venire trasmesso (contiene audio voce) o possa venire eliminato (non contiene audio voce).
Per la scelta finale, l'algoritmo utilizza la Short-term Energy, la Most Dominant Frequency e la Spectral Flatness Measure. Se almeno due delle precedenti ritorneranno un risultato positivo, il pacchetto verrà trasmetto, altrimenti eliminato.

2. SHORT TERM ENERGY

Un indicatore per la presenza del parlato in un segnale può semplicemente essere l'energia del segnale stesso. Regioni con maggiore energia indicano con alta probabilità la presenza del parlato nel segnale.
È possibile impostare una soglia minima tale per cui quando l'energia vi è al di sopra si possa affermare di trovarsi in una regione di parlato.

$$EN(X) := \begin{cases} 0, energy < threshold \\ 1, energy > threshold \end{cases}$$

La definizione di Short-term Energy è la seguente:

$$E(x) = \sum_{k=0}^{N-1} x_k^2$$

Dove x per noi è la finestra di segnale scelto.

Spesso, i segnali audio sono corrotti da del rumore di fondo che però rendono la Short-term Energy molto inaffidabile e perciò si ha bisogno di altri metodi per il riconoscimento del parlato nell'audio.

3.ANALISI SPETTRALE

Per ottenere ottimi risultati ed aumentare l'efficacia di un algoritmo VAD, spesso è utile analizzare il segnale nel dominio delle frequenze. Per ottenere il nuovo segnale viene usata la Discrete Fourier Transformation (DFT) su intervalli ridotti di tre pacchetti, ottenendo perciò un risultato ottimale per la seguente analisi.
Una volta ottenuto il segnale nel dominio delle frequenze è possibile calcolare la frequenza dominante (Most Dominant Frequency Component- MDFC) e applicare la Spectral Flatness Measure (SF).

3.1 MOST DOMINANT FREQUENCY

$$MDFC(x) := argmax_k(s(k))$$

Come per l'energia, è utile impostare una soglia, oltre la quale la frequenza verrà segnalata come parlato e perciò indicherà con un 1 il pacchetto analizzato.

3.2 SPECTRAL FLATNESS MEASURE

$$SFM(x) := \frac{G_m}{A_m} = \frac{\exp\left(\frac{1}{N}\sum_{n=0}^{N-1} \ln x(n)\right)}{\frac{1}{N}\sum_{n=0}^{N-1} x(n)}$$

Dove Gm rappresenta la media geometrica, Am quella aritmetica e x(n) la magnitudine dello spettro. La Spectral Flatness ritorna un risultato che quantifica quanto un suono è più vicino all'essere un rumore o un parlato. Una SFM alta indica che il segnale è più simile al rumore bianco e perciò verrà segnato uno 0 in corrispondenza al pacchetto analizzato, 1 altrimenti.

4. ALGORITMO SCELTO

La parte più delicata di un algoritmo VAD è la scelta delle soglie sulle quali verranno fatte le scelte. Per ottenere delle soglie efficaci è possibile prendere in considerazione i primi 3 pacchetti (60ms) e considerarli come silenzio o rumore, in quanto la reattività umana all'inizio di una registrazione è sempre maggiore di 100ms.

La procedura completa è descritta sotto:

Algoritmo VAD proposto

1- Dividere il segnale in pacchetti da 20ms corrispondendi a 160 campioni audio e calcolare il numero totale dei pacchetti (N_Packs)

2- Definire le soglie per le scelte del'algoritmo. (Solo le soglie per la MDFC e SFM sono scelte esternamente dopo una serie di prove sperimentali)

- EN_Thresh
- F_Thresh
- Flat_Thresh

3- for i from 3 to N_pack

a. Calcolo dell'energia EN

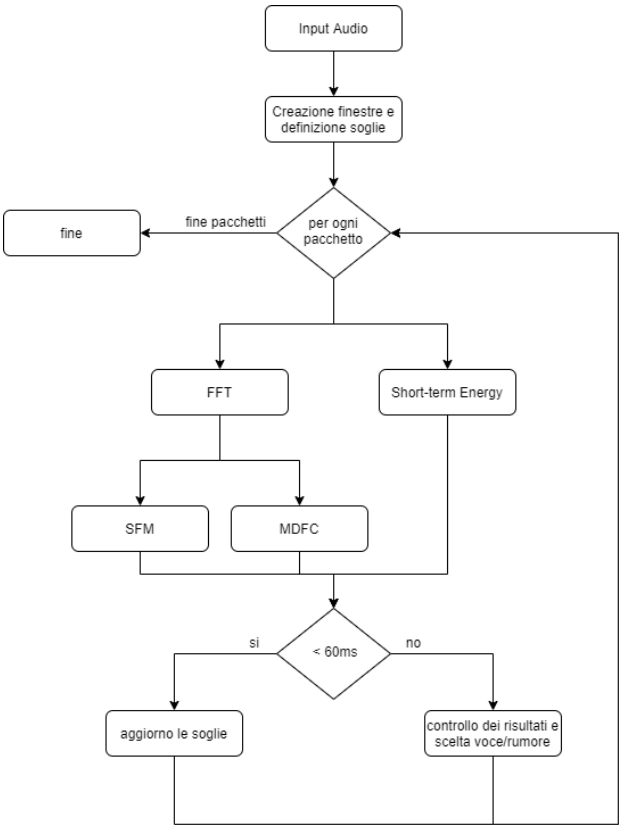
b. Applico la FFT
   - Calcolo la SFM
   - Calcolo la MDFC

c. Supponendo i primi 60ms di silenzio, calcolo i minimi raggiungi min_EN, min_F e max_Flat
   - Aggiorno EN_Thresh come massimo delle energie calcolate fino ad allora moltiplicato log(min_EN)

d. Dopo i 60ms inizia il controllo dei risultati ottenuti
   - EN - min_EN > EN_Thresh → count++
   - MDFC - min_F > F_Thresh → count++
   - SFM + max_Flat < Flat_Thresh → count++

e. Se count > 1 allora il pacchetto è voce

f. Se è voce, aggiorno EN_Thresh e min_EN



5. CONCLUSIONI

Aggiungo in seguito i risultati di alcune delle tracce audio fornite,come dimostrazione del corretto funzionamento dell'algoritmo descritto in precedenza.