Filip Kuś | NTwK gr. 9:45 | 17.04.2024

Fonoskopia

Sprawozdanie z laboratorium 6

Analiza cech sygnału mowy pod kątem fonoskopii. F0

W ramach zadania należało dla nagrań samogłosek wyznaczyć ton podstawowy mówcy. W tym celu wykorzystałem funkcję autokorelacji, a wyniki zweryfikowałem za pomocą funkcji pitch(). Ostatnim zadaniem było określenie rodzaju głosu autora wypowiedzi na podstawie tonu krtaniowego.

Rozpocząłem od wczytania plików audio:

[ya, fs] = audioread('17\_aaa\_norm.wav');

[ye] = audioread('17\_eee\_norm.wav');

[yi] = audioread('17\_iii\_norm.wav');

[yu] = audioread('17\_uuu\_norm.wav');

Następnie stworzyłem własną funkcję obliczającą F0 dla każdego podanego sygnału mowy i EGG, ponieważ analizie należało poddać de facto aż 8 sygnałów (mowa i EGG dla 4 samogłosek). Zredukowałem w ten sposób powtarzający się kod.

function F0\_value = F0(vowel, signal\_type, fs)

% F0 przyjmuje argumenty: sygnał ya/ye/yi/y; typ sygnału - mowa (wartość 1) lub EGG (wartość 2), częstotliwość próbkowania sygnałów wejściowych

R = xcorr(vowel(:, signal\_type), 'unbiased'); % do obliczenia autokorelacji wykorzystana funkcja wbudowana xcorr

R = R(10000:80000); % zawężenie zakresu wyniku, by znaleźć prawidłowe maksimum (i tak znajduje się ono na środku przebiegu)

[~, xmax1] = max(R); % znalezienie maksimum funkcji autokorelacji

R(xmax1 - 20 : xmax1 + 20) = 0; % wyzerowanie wartości maksimum globalnego i jego otoczenia w celu prawidłowego znalezienia sąsiedniego maksimum

[~, xmax2] = max(R); % znalezienie sąsiedniego maksimum

F0\_value = fs/abs(xmax1 - xmax2); % obliczenie F0

end

Dzięki temu w głównym skrypcie wystarczy tylko wywołać funkcje z odpowiednimi parametrami:

F0\_a\_y = F0(ya, 1, fs);

F0\_a\_EGG = F0(ya, 2, fs);

F0\_e\_y = F0(ye, 1, fs);

F0\_e\_EGG = F0(ye, 2, fs);

F0\_i\_y = F0(yi, 1, fs);

F0\_i\_EGG = F0(yi, 2, fs);

F0\_u\_y = F0(yu, 1, fs);

F0\_u\_EGG = F0(yu, 2, fs);

Do weryfikacji wyników w ramach podpunktu b) również utworzyłem pomocniczą funkcję:

function pitch\_F0\_value = pitch\_F0(vowel, signal\_type, fs)

pitch\_F0\_value = mode(pitch(vowel(:,signal\_type), fs, Method = "SRH", WindowLength = round(0.03\*fs), OverlapLength = round(0.01\*fs)));

% aby ujednoznacznić wyniki, postanowiłem przyjąć wartość F0 jako modę wektora pitch - pozwala to osiągnąć zadowalające uśrednienie

end

Wywołania w głównym skrypcie:

pitch\_F0\_a\_y = pitch\_F0(ya, 1, fs);

pitch\_F0\_a\_EGG = pitch\_F0(ya, 2, fs);

pitch\_F0\_e\_y = pitch\_F0(ye, 1, fs);

pitch\_F0\_e\_EGG = pitch\_F0(ye, 2, fs);

pitch\_F0\_i\_y = pitch\_F0(yi, 1, fs);

pitch\_F0\_i\_EGG = pitch\_F0(yi, 2, fs);

pitch\_F0\_u\_y = pitch\_F0(yu, 1, fs);

pitch\_F0\_u\_EGG = pitch\_F0(yu, 2, fs);

Obraz zawierający tekst, Wykres, linia, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

**Rys 1a.**Wykres funkcji autokorelacji dla samogłoski „a” z zaznaczonymi na czerwono maksimami wykorzystywanymi w obliczaniu F0

**Rys 1b.**Wykres F0 dla samogłoski „a” narysowany za pomocą funkcji pitch()

Porównanie wartości odpowiednich zmiennych z przestrzeni roboczych (różnice między nimi są jedynie minimalne i wynikają z uśrednienia dla pitch()) pozwala stwierdzić, że obliczenia przebiegły poprawnie:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Rys 2a.**Wartości F0 obliczone za pomocą funkcji autokorelacji

**Rys 2b.**Wartości F0 obliczone za pomocą funkcji pitch()

Na tej podstawie mogę próbować określić swój rodzaj głosu jako baryton, gdyż częstotliwość podstawowa mojego głosu wynosi ok. 111 Hz. Modelową dolną granicą barytonu jest zaś 100 Hz.

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznie