Analiza i Przetwarzanie Dźwięku - Sprawozdanie z projektu 1.

Szymon Tomulewicz

9 IV 2020

Spis treści

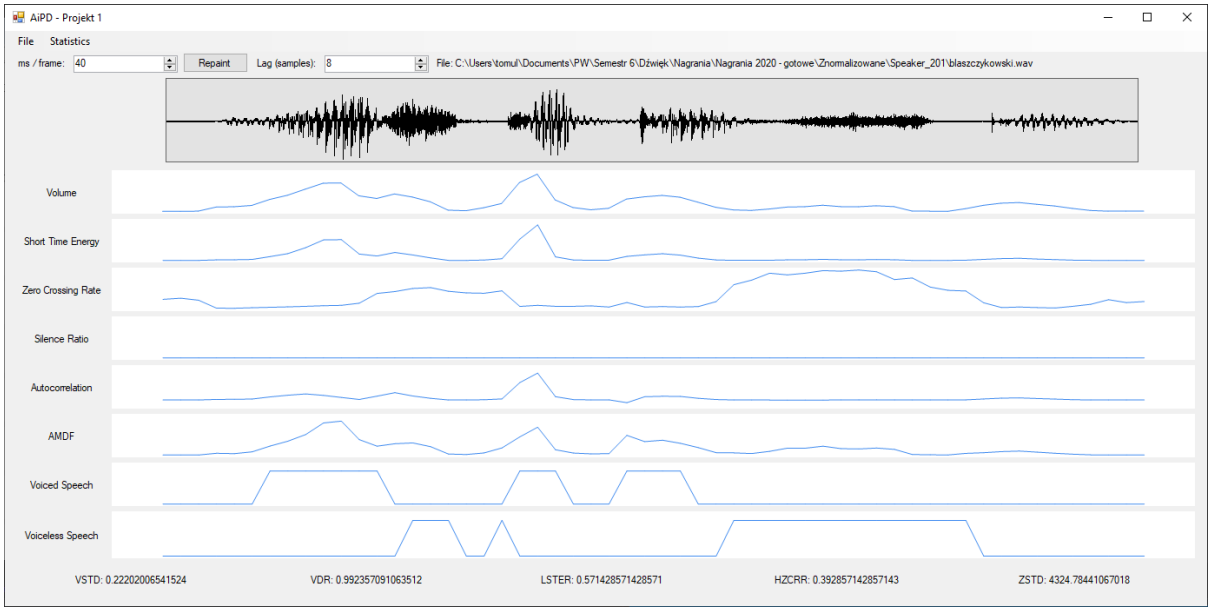
1 Wprowadzenie

Celem projektu było stworzenie aplikacji okienkowej, umożliwiającej wczytywanie plików wav, a następnie analizowanie ich pod kątem wybranych parametrów i wartości.

1.1 Aplikacja

Aplikacja została wykonana w języku C#, przy użyciu biblioteki NAudio do przetwarzania i analizy plików dźwiękowych, oraz biblioteki WinForms do stworzenia interfejsu oraz wyświetlania parametrów. Wybrałem NAudio ze względu na popularność tej biblioteki, ilość dokumentacji oraz integrację z biblioteką WinForms.

Aplikacja składa się z jednego widoku (Rys. 1), na którym po załadowaniu pliku dzwiękowego rysują się wykresy parametrów opisanych w dalszej części sprawozdania.



Rysunek 1: Przykładowy stan aplikacji

Użytkownik może podać w interfejsie dwie wartości: długość ramki w milisekundach oraz parametr (tzw. *lag*) używany w funkcji autokorelacji oraz AMDF (Average Magnitude Difference Function). Jako jednostkę parametru *lag* aplikacja przyjmuje liczbę próbek.

2 Obliczane parametry

Wszyskie obliczenia w aplikacji, które są wykonywane na próbkach, używają typów zmiennoprzecinkowych. NAudio umożliwia to przy użyciu metody `ToSampleProvider`:

```
using (WaveFileReader reader = new WaveFileReader(filePath))
{
    // Convert to 32-bit floating point samples:
    ISampleProvider sampleProvider = reader.ToSampleProvider();
    ...
    float[] buffer = new float[frameSizeFloats];
```

```
        ...
        sampleProvider.Read(buffer, 0, frameSizeFloats);
        ...
    }
```

2.1 Parametry w dziedzinie czasu na poziomie ramki (Frame-level)

Parametry opisane w kolejnych sekcjach są obliczane, a następnie zapisywane i wyświetlane w formie wykresów (Rys. 1). N występujace we wzorach oznacza zwykle długość ramki w próbkach.

2.1.1 STE - Short Time Energy

Dla n-tej ramki, STE wynosi:

$$STE(n) = \frac{1}{N}\sum_{i=0}^{N-1} s_n^2(i) \tag{1}$$

Gdzie $s_n(i)$ to amplituda i-tej próbki w n-tej ramce.

2.1.2 Volume - głośność

Dla n-tej ramki, głośność wynosi:

$$v(n) = \sqrt{STE(n)} \tag{2}$$

2.1.3 ZCR - Zero Crossing Rate

Dla n-tej ramki, ZCR wynosi

$$Z(n) = \frac{1}{2}\left(\sum_{i=1}^{N-1} |sign(s_n(i)) - sign(s_n(i-1))|\right)\frac{f_s}{N} \tag{3}$$

Gdzie $sign$ to funkcja signum, a f_s to częstotliwość próbkowania sygnału.

2.1.4 Częstotliwość tonu podstawowego

W nadziei na możliwość obliczenia tonu podstawowego zaimplementowałem obliczanie następujących funkcji:

- Funkcji autokorelacji:

$$R_n(l) = \sum_{i=0}^{N-l-1} s_n(i)s_n(i+l) \tag{4}$$

- Funkcji AMDF (Average Magnitude Difference Function):

$$A_n(l) = \sum_{i=0}^{N-l-1} |s_n(i+l) - s_n(i)| \tag{5}$$

Gdzie $s_n(i)$ oznacza amplitudę i-tej próbki w n-tej ramce, a l - parametr lag podany przez uzytkownika.

Niestety dla tych parametrów, operacja obliczania (a właściwie przybliżania) częstotliwości tonu podstawowego okazała się zbyt kosztowna (Zob. [2]) na potrzeby tego projektu.

2.2 Cechy sygnału audio na poziomie klipu (Clip-level)

Parametry opisane w kolejnych sekcjach są obliczane, a następnie zapisywane i wyświetlane na dole widoku (Rys. 1).

2.2.1 VSTD

Odchylenie standardowe znormalizowane przez maksymalną wartość głośności w całym klipie.

$$VSTD = \frac{1}{max(v)} \sqrt{\sum_{n=0}^{F} \Big(v(s_n) - avg(v)\Big)^2} \tag{6}$$

Gdzie F jest liczbą ramek, na które został podzielony klip, $v(s_n)$ - głośnością w n-tej ramce, $max(v)$ - maksymalną wartością głośności w całym klipie, a $avg(v)$ - średnią głośnością w całym klipie.

2.2.2 ZSTD

Analogicznie liczone jest odchylenie standardowe ZCR. W tym wypadku jednak aplikacja nie normalizuje otrzymanej wartości.

$$ZSTD = \sqrt{\sum_{n=0}^{F} \Big(z(s_n) - avg(z)\Big)^2} \tag{7}$$

Gdzie $z(s_n)$ jest wartością ZCR w n-tej ramce, a $avg(z)$ - średnią wartością ZCR w całym klipie.

2.2.3 VDR - Volume dynamic range

$$VDR = \frac{max(v) - min(v)}{max(v)} \tag{8}$$

Gdzie $max(v)$ to maksymalna głośność w klipie, a $min(v)$ - minimalna.

2.2.4 LSTER - Low Short Time Energy Ratio

$$LSTER = \frac{1}{2N} \sum_{n=0}^{N-1} [sgn(0.5 * avg(STE) - STE(n) + 1] \tag{9}$$

Gdzie N to całkowita liczba ramek, $avg(STE)$ - średnia wartość STE w jednosekundowym oknie, a $STE(n)$ - wartość STE w n-tej ramce.

2.2.5 HZCRR - High Zero Crossing Rate Ratio

$$HZCRR = \frac{1}{2N} \sum_{n=0}^{N-1} [sgn(ZCR(n) - 1.5 * avg(ZCR)) + 1] \tag{10}$$

Gdzie N to całkowita liczba ramek, $avg(ZCR)$ - średnia wartość ZCR w jednosekundowym oknie, a $ZCR(n)$ - wartość ZCR w n-tej ramce.

3 Wykrywanie ciszy

Aplikacja wykrywa ciszę w klipie audio przy pomocy wcześniej obliczonych wartości ZCR oraz głośności. Dla każdej ramki, sprawdzane są poziomy tych wartości. Jeśli $v(n) < 0.02$ oraz $Z(n) < 50$, sygnał w danej ramce uznawany jest za ciszę.

```
// Dla każdej ramki
for (int n = 0; n < frameCount; n++)
{
    // 1.0 -> cisza
    double value =
        (volumeData[n] < 0.02 && zcrData[n] < 50.0) ? 1.0 : 0.0;
    // Dodaj do wykresu
    dt.Rows.Add(n, value);
}
```

Metoda ta nie jest doskonała, co zostanie pokazane w sekcji 5.

4 Wykrywanie mowy dźwięcznej i bezdźwięcznej

Określanie mowy dźwięcznej i bezdźwięcznej jest możliwe z dość dużym przybliżeniem przy użyciu STE oraz ZCR ([1]). Jako materiał pomocny w przybliżaniu progów użyłem znormalizowanych nagrań z poprzednich lat. Finalnie warunki jakie muszą być spełnione żeby dany sygnał audio w ramce został zakwalifikowany jako

- Mowa bezdźwięczna: $STE(n) < 0.03 \wedge z(n) \geqslant 6000$
- Mowa dźwięczna: $STE(n) \geqslant 0.03 \wedge z(n) < 6000$

Dokładność tej metody zostanie pokazana w sekcji 5.

5 Prezentacja wyników działania

Może dla kilku różnych parametrów, na różnych sygnałach żeby można było porównać

6 Wnioski

Na podstawie wyników (i implementacji) można napisać wnioski. Np.:

6.1 Napotkane problemy

Domyślne wykresy, jakie oferuje biblioteka WinForms okazały się dalekie od oczekiwań jakie możnaby mieć w projekcie takim jak ten. Nie udało mi się niestety dostosować ich na tyle, aby czytelne były zarówno wykresy, jak i wartości je opisujące. W kolejnych projektach planuję wypróbować inne biblioteki do rysowania wykresów.

Kolejną trudnością, jaką napotkałem i której nie udało mi się rozwiązać, było wczytywanie plików mp3. Brak płynności w nomenklaturze związanej zarówno z NAudio, jak i formatami plików audio okazał się dość dużą barierą.

6.2 Czy metoda zawsze działa dobrze? Dla wszystkich przypadków?

Czy metoda zawsze działa dobrze? Dla wszystkich przypadków?

6.3 Czy parametry można zmieniać dowolnie? Może są poprawne tylko w jakimś zakresie.

Czy parametry można zmieniać dowolnie? Może są poprawne tylko w jakimś zakresie.

6.4 Dlaczego są różne wersje? Czy któraś jest lepsza?

Dlaczego są różne wersje? Czy któraś jest lepsza?

7 Zakończenie

Zakończenie.

Literatura

[1] Adapa B. Barkana B.D. Bachu R.G., Kopparthi S. Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. 2008.

[2] Maciej Leszczyna. Metody wykrywania częstotliwości podstawowej. `https://sound.eti.pg.gda.pl/student/eim/synteza/leszczyna/index.htm`.