Analiza i Przetwarzanie Dźwięku - Sprawozdanie z projektu 1.

Szymon Tomulewicz

9 IV 2020

Spis treści

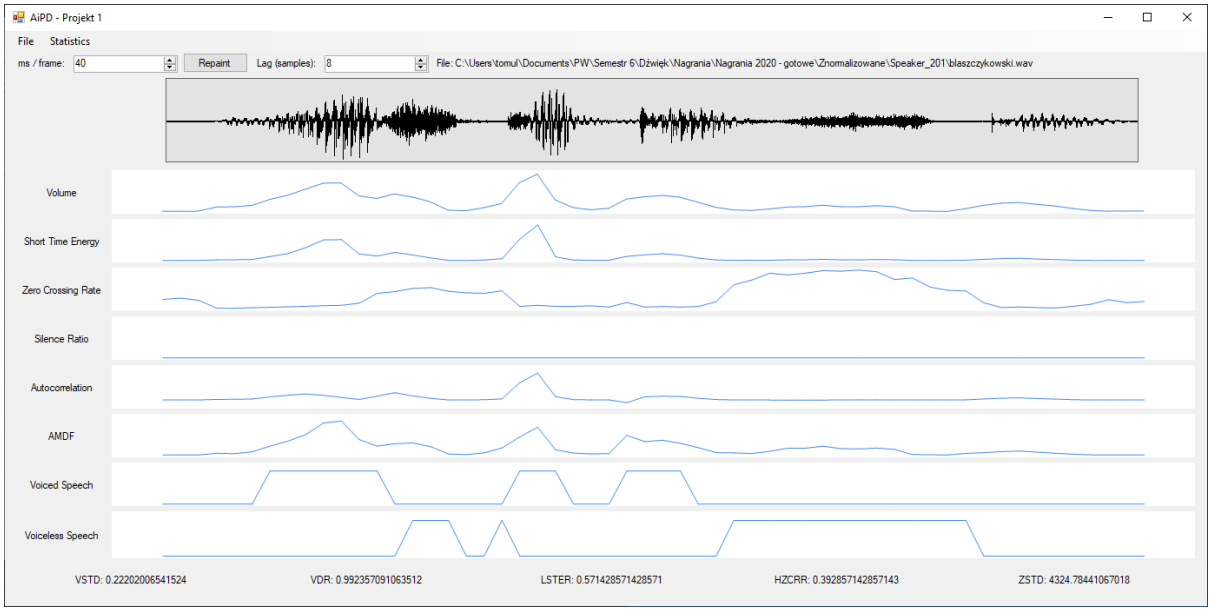
1 Wprowadzenie

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 dźwiękowego 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

Wszystkie 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. **??**). N występujące 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

$$Z(n) = \frac{1}{N2}\left(\sum_{i=1}^{N-1} |sign(s_n(i)) - sign(s_n(i-1))|\right) \tag{3}$$

Gdzie *sign* to funkcja signum. Zrezygnowałem z mnożenia wyniku przez częstotliwość próbkowania, ponieważ niepotrzebnie uzależniało to ZCR od kolejnej wartości, a co za tym idzie utrudniało określanie progów (np. 3.1).

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 użytkownika.

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. **??**). Liczba ramek, na które został podzielony klip oznaczana jest jako F.

2.2.1 VSTD

Odchylenie standardowe znormalizowane przez maksymalną wartość głośności w całym klipie.

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

Gdzie , $v(s_n)$ - 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.

$$ZSTD = \sqrt{\sum_{n=0}^{F-1} \Big(z(s_n) - avg(z)\Big)^2} \tag{7}$$

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

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

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

2.2.5 HZCRR - High Zero Crossing Rate Ratio

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

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

3 Metody

Po obliczeniu poprzednich wartości, aplikacja przechodzi do wykrywania ciszy oraz kategoryzowania mowy.

3.1 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 $Z(n) \leqslant 0.5$, 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] <= 0.5) ? 1.0 : 0.0;
    // Dodaj do wykresu
    dt.Rows.Add(n, value);
}
```

Metoda ta nie jest doskonała, co zostanie pokazane w sekcji 4.

3.2 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: $STE(n) < 0.03 \wedge z(n) \geqslant 0.13$
- Mowa dźwięczna: $STE(n) \geqslant 0.03 \wedge z(n) < 0.13$

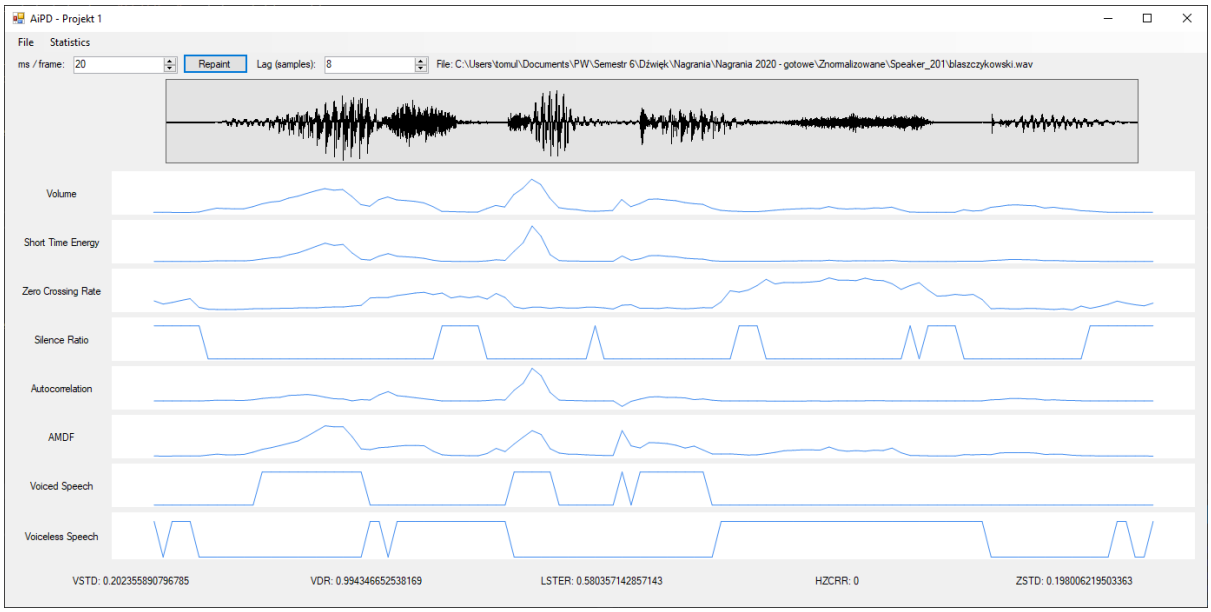
Dokładność tej metody zostanie pokazana w sekcji 4.

4 Prezentacja wyników działania

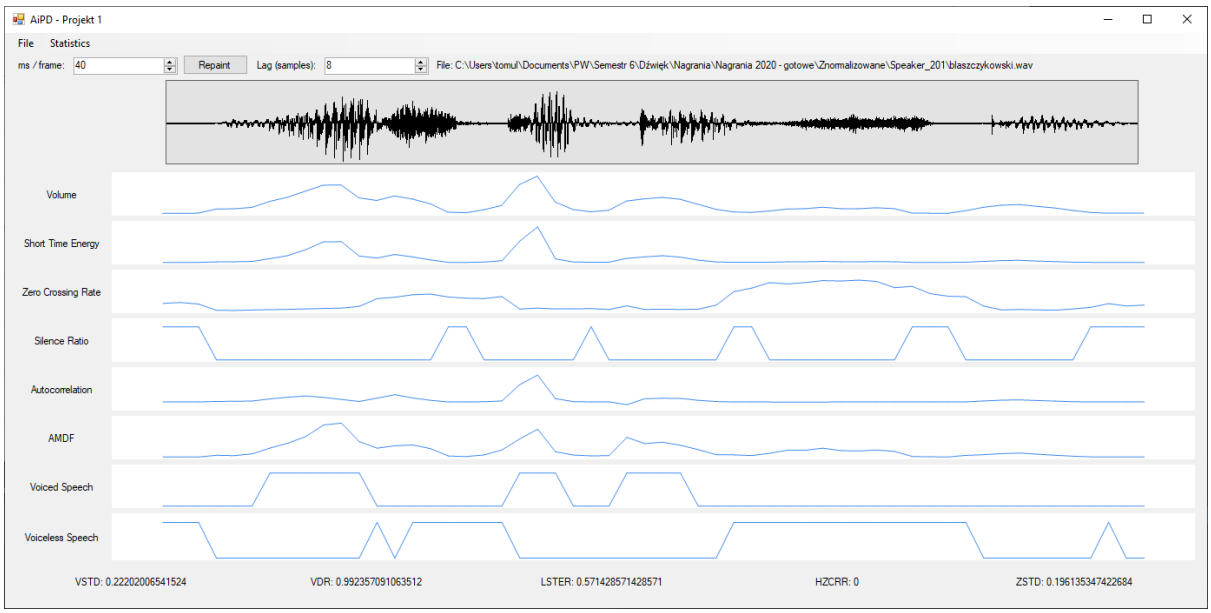
Jako pierwszy przykład wyników działania wybrałem słowo “Błaszczykowski”. Ma znacznie więcej fragmentów mowy bezdźwięcznej niż większość nagranych słów, a także

dość dobrze reprezentuje różnicę między nagraniami głosu męskiego, żeńskiego i nagraniami z duża ilością hałasu w tle.

Porównując rys. 3 i rys. 2, możemy łatwo zaobserwować jaką różnicę w wynikach daje zastosowanie krótszej ramki (20 ms dla rys. 2, oraz 40 ms dla rys 3). Kształty wykresów są zdecydowanie dokładniejsze i bardziej zarysowane. Łatwiej natomiast o zakłócenia w wynikach np. wykrywania mowy dźwięcznej (Wykres "Voiced Speech" w aplikacji).



Rysunek 2: "Błaszczykowski" - głos męski, 20 ms na ramkę

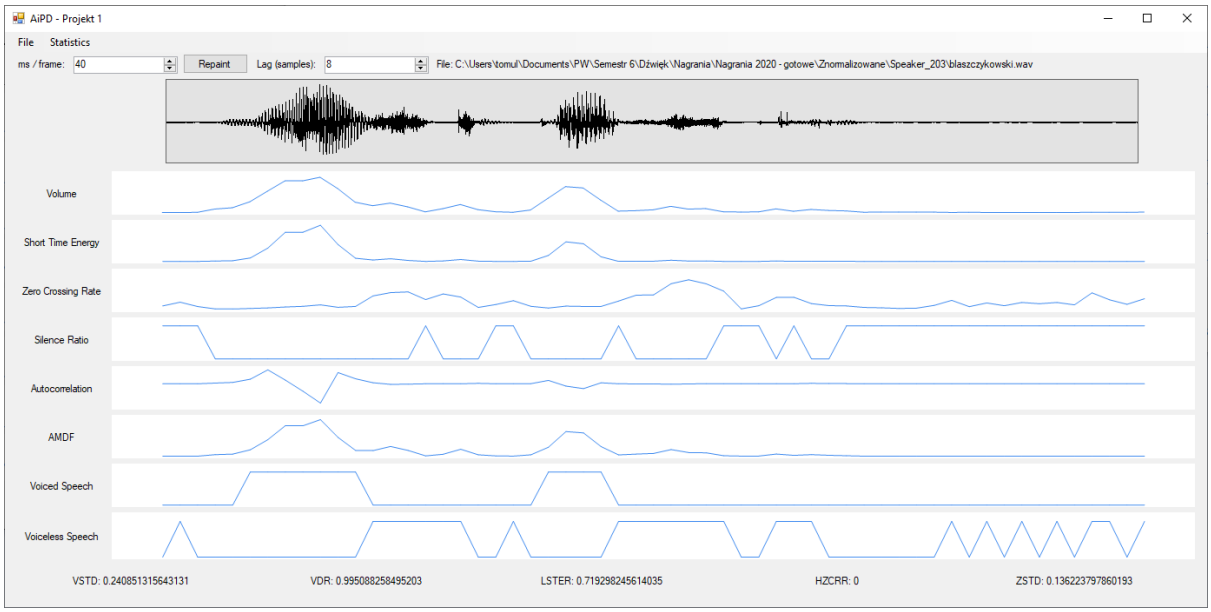


Rysunek 3: "Błaszczykowski" - głos męski, 40 ms na ramkę

W kolejnych przykładach będzie utrzymana stała wartość 40 ms na ramkę.

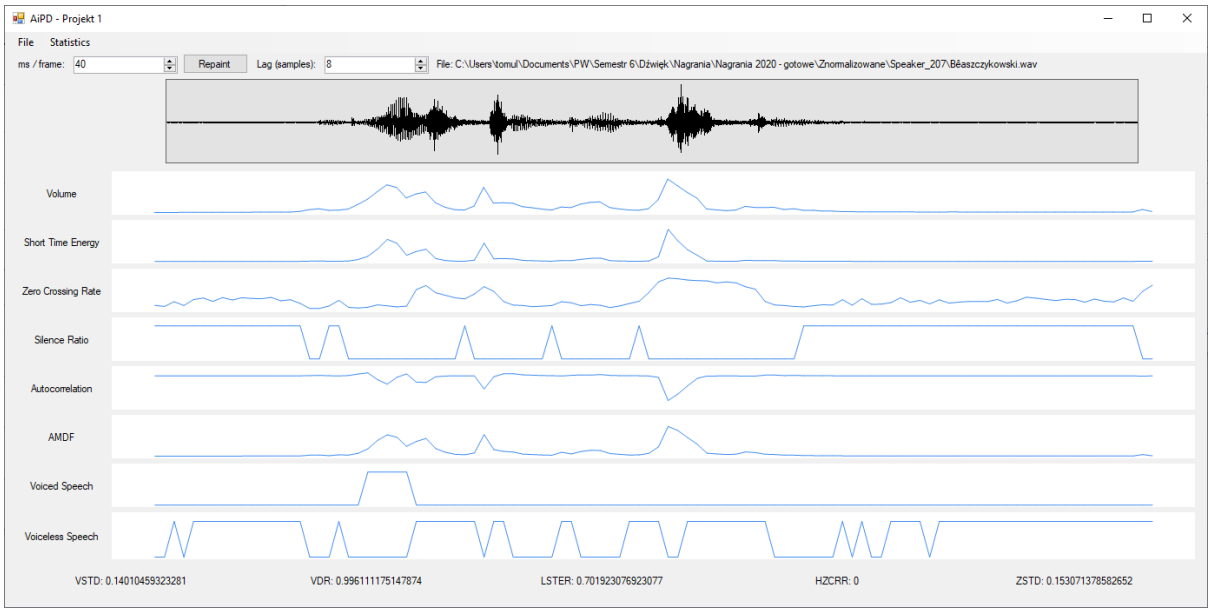
Rysunek 4 przedstawia analizę nagrania damskiego głosu, wypowiadającego słowo "Błaszczykowski". Na tym nagraniu znacznie bardziej uwydatnione są przerwy między sylabami, co widać na wykresie opisanym "Silence Ratio". Widać także, że litera "y"

obecna w drugiej sylabie (a więc po pierwszej przerwie) jest wypowiedziana znacznie ciszej. Wpływa to na brak kwalifikacji tego fragmentu zarówno do mowy dźwięcznej, jak i bezdźwięcznej.



Rysunek 4: “Błaszczykowski” - głos żeński

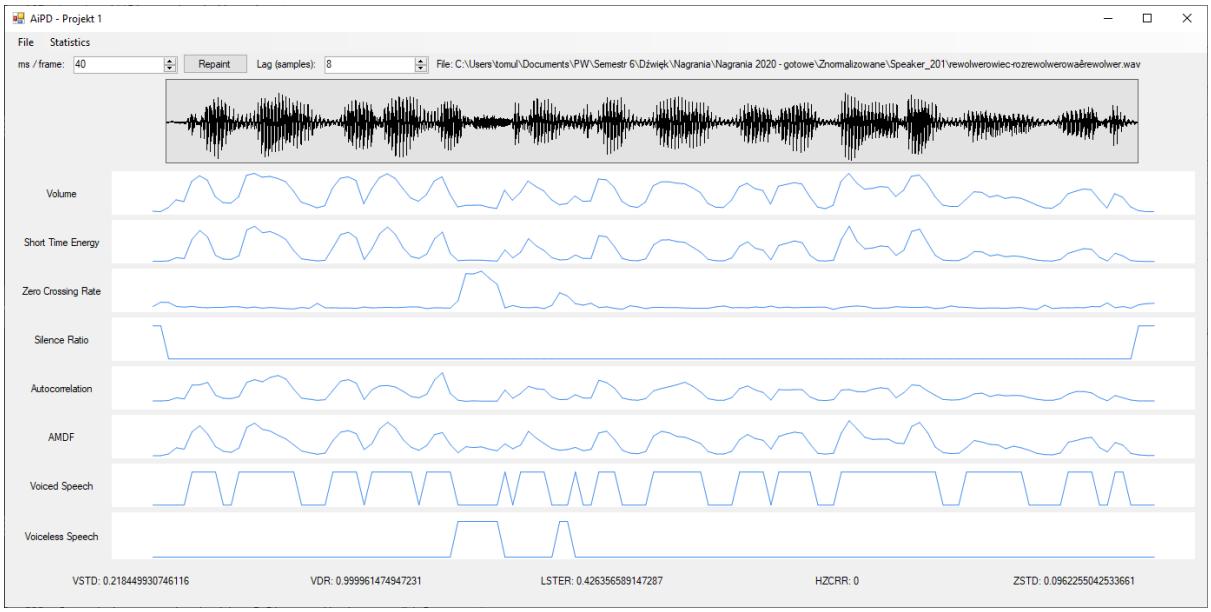
Na rys. 5 przedstawiona została analiza nagrania damskiego głosu wypowiadającego słowo “Błaszczykowski” z obecnym słyszalnym hałasem w tle. Hałas, charakteryzujący się wysokimi wartościami ZCR wpływa bezpośrednio na klasyfikację mowy. Jako mowa dźwięczna zakwalifikowana jest jedynie pierwsza część pierwszej sylaby. Hałas ten sprawia także, że jako mowa bezdźwięczna kwalifikowany jest końcowy fragment, uznawany równocześnie za ciszę.



Rysunek 5: “Błaszczykowski” - hałas w tle

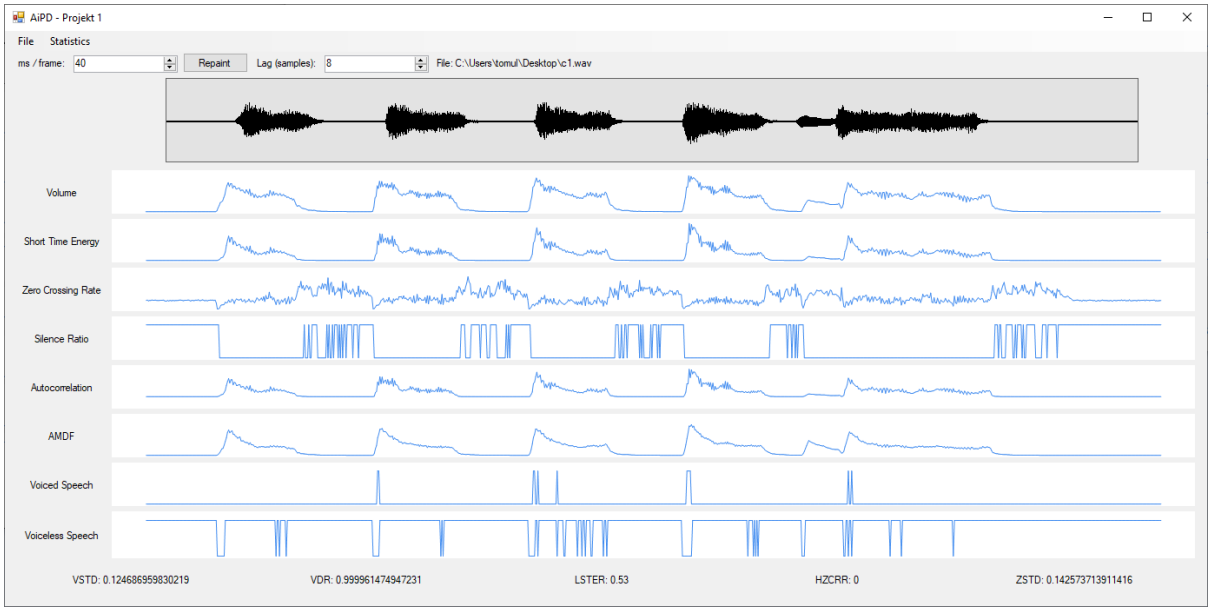
Kolejnym przykładem jest trudne zdanie o rewolwerowcu, wypowiadane przez głos

męski. To zdanie charakteryzuje się jednym wyraźnym bezdźwięcznym “c”. Doskonale pokazuje to wykres przebiegu ZCR.



Rysunek 6: Trudne zdanie o rewolwerowcu - głos męski

Ostatnim wybranym przeze mnie przykładem jest nagranie serii akordów zagranych na syntezatorze *Roland Juno-60*, a następnie przepuszczonym przez efekt typu delay. Mimo, że wykres energii wskazuje, że między poszczególnymi akordami jest cisza, w rzeczywistości słyszalny jest tam bardzo subtelny sygnał generowany przez efekt delay. To nagranie charakteryzuje się także najniższym współczynnikiem $VSTD$ oraz $LSTER$ ze wszystkich przedstawionych w tej sekcji przykładów.



Rysunek 7: Syntezator analogowy

5 Wnioski

Na podstawie wyników i implementacji projektu nasuwają się następujące wnioski:

5.1 Czy metody zawsze działają dobrze?

Przedstawione w tym sprawozdaniu metody działają zadowalająco w stosunku do niskiego nakładu obliczeń, jakiego wymagają (wszystkie są co najwyżej liniowe względem długości nagrania). Oczywistym przykładem wskazującym na niepełność metody klasyfikowania mowy jako dźwięcznej lub bezdźwięcznej jest przykład pokazany na rys. 7. Można wzbogacić tę metodę o rozróżnianie mowy i muzyki, jednak nie udało mi się tego zaimplementować.

5.2 Modyfikacja parametrów

Metody wykrywania ciszy oraz mowy dźwięcznej i bezdźwięcznej potrzebują w tej implementacji z góry narzuconych progów dla wartości ZCR, STE oraz głośności. Sprawia to, że jeśli progi te nie będą dobrane dokładnie pod dane nagrania, mogą pojawić się błędy. Bardziej zaawansowane metody rozwiązują ten problem.

5.3 Inne wersje metod

Prawdopodobnie metoda wykrywania mowy dźwięcznej i bezdźwięcznej wykorzystująca częstotliwość tonu podstawowego byłaby skuteczniejsza.

5.4 Napotkane problemy

Domyślne wykresy, jakie oferuje biblioteka WinForms okazały się dalekie od oczekiwań jakie można by 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.

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`.