### Informatyka, studia dzienne, mgr II st. semestr I

## Przetwarzanie obrazu i dźwięku 2020/2021

### Prowadzący: dr inż. Bartłomiej Stasiak poniedziałek, 12:15

### Data oddania: Ocena:

Aleksander Dominiak 234049

Dominik Andrzejczak 234038

# Zadanie 3:

# Analiza częstotliwości podstawowej dźwięku.

## 1. Cel

Celem niniejszego zadania było zaprojektowanie aplikacji pozwalającej na wczytanie dźwięku z pliku \*.wav oraz wykrycie częstotliwości podstawowej w kolejnych chwilach czasowych. W tym celu należało zaimplementować jedną metodę w dziedzinie czasu oraz jedną w dziedzinie częstotliwości.

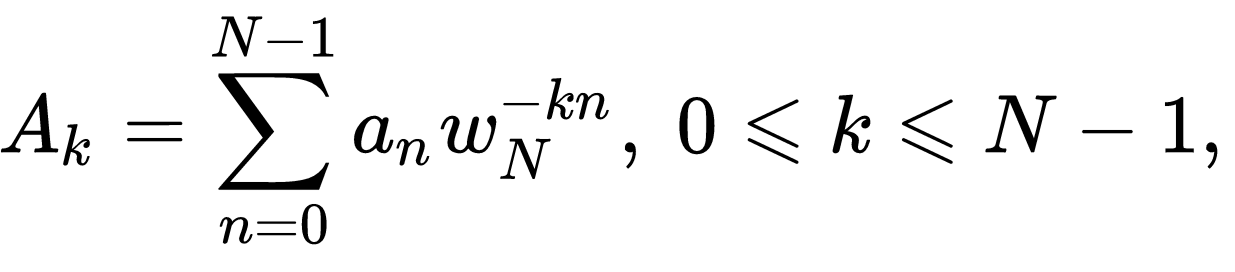
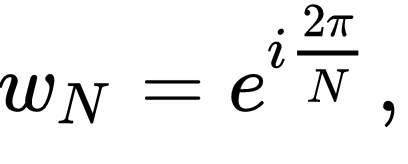
## 2. Wprowadzenie

Wykonanie zadania wiązało się z implementacją operacji wczytania dźwięku, które oparte zostało na bibliotece nAudio dostępnej w pakiecie nuget w środowisku Visual Studio. Do wykonania części badania częstotliwości podstawowej potrzebna była implementacja DFT. Zaimplementowano jej szybszą wersję FFT:

**FFT**

Szybkie przekształcenie Fouriera (FFT) to bardzo wydajny algorytm implementujący dyskretne przekształcenie Fouriera (DFT).

DFT to transformata Fouriera wyznaczona dla sygnału próbkowanego, a więc dyskretnego. Przekształca ona skończony ciąg próbek sygnału (*a0, a1, ..., aN-1*), *ai*∈R w ciąg harmoniczny (*A0, A1, ..., AN-1*), *Ai*∈C, zgodnie ze wzorem:

gdzie:

*i* – jednostka urojona,

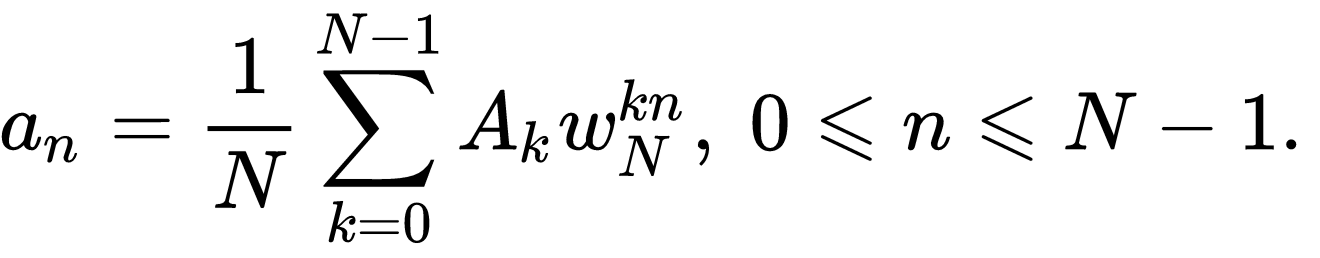
*k* – numer harmoniczny,

*n* – numer próbki sygnału,

*an* – wartość próbki sygnału,

*N* – liczba próbek

Przekształcenie odwrotne wyraża się wzorem:



FFT pozwala na szybsze obliczenie DFT, najpopularniejszym wariantem FFT jest ten o podstawie 2, wymagający, by rozmiar DFT był całkowitą potęgą liczby dwa.

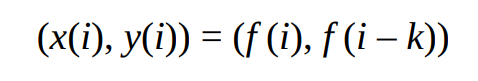
DFT do obliczenia wymaga wykonania N2 mnożeń. Dodatkowym problemem jest fakt, że są to mnożenia liczb zespolonych, znacznie zużywające moc obliczeniową procesora. W przypadku FFT liczba mnożeń spada do około N/2 \* log2N, przy czym część z nich wymaga jedynie zmiany znaku.

Algorytm FFT korzysta z podanych heurystyk budowania szybkich algorytmów:

* „dziel i rządź” – podział zadania na mniejsze fragmenty,
* rekurencja,
* programowanie dynamiczne.

**Wykrycie częstotliwości podstawowej w dziedzinie czasu:**

Do wyznaczenia częstotliwości podstawowej w dziedzinie czasu wykorzystano metodę analizy sygnału w przestrzeni fazowej. Polega ona na wykrywaniu cykli w reprezentacji sygnału dźwiękowego w odpowiednio zdefiniowanej przestrzeni wielowymiarowej. W pierwszym wymiarze jest to wartość tego sygnału, a w kolejnych chwilowe wartości jego następnych pochodnych. W celach uproszczenia obliczeń można także skorzystać z wzoru:



gdzie *f(i)* oznacza *i*-tą próbkę sygnału, a *k* to opóźnienie o pewną liczbę próbek.

Częstotliwość w tej metodzie można ustalić korzystając z przejścia przez konkretny punkt, w takim momencie czas dla tego punktu zostaje zapisany, a następnie porównany do czasu następnego przejścia przez dany punkt. Różnica w uzyskanych czasach to okres sygnału. Dla poprawienia dokładności ustaleń można dodać dodatkowy wymiar.

**Wykrycie częstotliwości podstawowej w dziedzinie częstotliwości:**

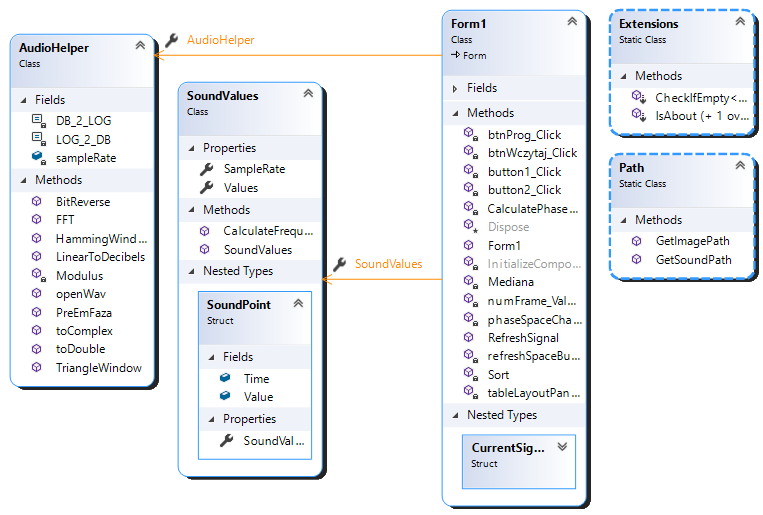
Do wyznaczenia częstotliwości podstawowej dźwięku w dziedzinie częstotliwości wykorzystano metodę Analizy widma Fouriera sygnału (F1). W tej metodzie potrzebnych było kilka kroków:

1. wczytanie sygnału z pliku \*.wav
2. podział sygnału na równej długości ramki
3. zastosowanie pre-emfazy
4. obliczenie widma amplitudowego sygnału (FFT)
5. ustalenie progu
6. znalezienie maksimów lokalnych powyżej progu
7. wyznaczenie odległości pomiędzy maksimami
8. wybór mediany odległości jako częstotliwości podstawowej danej ramki

Umożliwiono podgląd każdej ramki sygnału oraz ustalenie ogólnego progu dla wszystkich ramek.

## 3. Opis implementacji

Aplikacja została napisana w języku C# z użyciem frameworka Windows Forms. Rozdzielono działanie aplikacji na dwie oddzielne metody: w dziedzinie częstotliwości i czasu.



Klasa Form1 odpowiada za interfejs użytkownika.

Klasa AudioHelper odpowiada za wykrycie częstotliwości podstawowej w dziedzinie częstotliwości.

Klasa SoundValues przechowuje wartości dźwięku użyte w wykrywaniu częstotliwości w dziedzinie czasu. Jej zagnieżdżona struktura SoundPoint przechowuje wartości dźwięku wraz z odpowiadającym im czasem w ms.

Klasy Path i Extensions to klasy pomocnicze.

## 4. Materiały i metody

Do weryfikacji poprawności działania aplikacji wykorzystano następujące pliki audio:

*Dźwięki sztuczne:*

***łatwe****: 100Hz.wav, 225Hz.wav, 506Hz.wav, 1139Hz.wav*

***średnie****: 90Hz.wav, 135Hz.wav, 303Hz.wav, 683Hz.wav, 1537Hz.wav*

***trudne:*** 80Hz.wav, 180Hz.wav, 405Hz.wav, 911Hz.wav

*Dźwięki naturalne:*

***flet***: 276Hz.wav, 443Hz.wav, 591Hz.wav, 887Hz.wav, 1265Hz.wav, 1779Hz.wav. ***viola***: 130Hz.wav, 196Hz.wav, 247Hz.wav, 294Hz.wav, 369Hz.wav, 440Hz.wav, 698Hz.wav

*Sekwencje:*

*DWK\_violin.wav*

*KDF\_piano.wav*

## 5. Wyniki

|  |  |  |  |
| --- | --- | --- | --- |
| ***Rodzaj*** | ***Plik audio*** | ***Analiza widma*** | ***Analiza sygnału w przestrzeni fazowej*** |
| Sztuczny, łatwy | 100Hz | 102 Hz | 100 Hz |
| Sztuczny, łatwy | 225Hz | 226 Hz | 225 Hz |
| Sztuczny, łatwy | 506Hz | 506 Hz | 38 Hz |
| Sztuczny, łatwy | 1139Hz | 1138,5 Hz | 10 Hz |
| Sztuczny, średni | 90Hz | 88,5 Hz | 3000 Hz |
| Sztuczny, średni | 135Hz | 134 Hz | 45 Hz |
| Sztuczny, średni | 303Hz | 301 Hz | 35 Hz |
| Sztuczny, średni | 683Hz | 683,5 Hz | 33 Hz |
| Sztuczny, średni | 1537Hz | 1539 Hz | 103.5 Hz |
| Sztuczny, trudny | 80Hz | 118 Hz | 5250 Hz |
| Sztuczny, trudny | 180Hz | 180,5 Hz | 3000 Hz |
| Sztuczny, trudny | 405Hz | 452 Hz | 1920 Hz |
| Sztuczny, trudny | 911Hz | 912,5 Hz | 1000 Hz |
| Naturalny, flet | 276Hz | 275 Hz | 312 Hz |
| Naturalny, flet | 443Hz | 441 Hz | 480 Hz |
| Naturalny, flet | 591Hz | 589 Hz | 1341 Hz |
| Naturalny, flet | 887Hz | 885 Hz | 114 Hz |
| Naturalny, flet | 1265Hz | 1246,5 Hz | 83 Hz |
| Naturalny, flet | 1779Hz | 1756,5 Hz | 210 Hz |
| Naturalny, viola | 130Hz | 132 Hz | 90 Hz |
| Naturalny, viola | 196Hz | 196 Hz | 695 Hz |
| Naturalny, viola | 247Hz | 247,5 Hz | 74 Hz |
| Naturalny, viola | 294Hz | 291 Hz | 292 Hz |
| Naturalny, viola | 369Hz | 366 Hz | 124 Hz |
| Naturalny, viola | 440Hz | 441 Hz | 109 Hz |
| Sekwencja | DWK\_violin.wav | 264 Hz | 1100 Hz |
| Sekwencja | KDF\_piano.wav | 215 Hz | 2006 Hz |
|  |  |  |  |

## 6. Dyskusja

Metoda analizy sygnału w przestrzeni fazowej okazała się niewystarczająca w większości przypadków. Nie licząc najprostszych dźwięków uzyskane rezultaty znacząco odbiegały od prawdziwych wartości.

Metoda analizy widma z kolei okazała się działać skutecznie w przypadku większości testowanych dźwięków. W pozostałych przypadkach uzyskane wyniki pomimo rozbieżności, wciąż były zbliżone do prawdziwych częstotliwości.

## 7. Wnioski

Najprawdopodobniejszą przyczyną błędnych wyników uzyskanych przy użyciu metody analizy sygnału w przestrzeni fazowej był sposób testowania podobieństwa punktów, a także kwestia doboru stałej *k*, możliwe, że wyniki uległyby polepszeniu, gdyby zamiast stałej skorzystać z wartości obliczanej dynamicznie, np. z pomocą histogramu.

Metoda analizy widma wydaje się być skutecznym sposobem szacowania częstotliwości dźwięku, nawet jeżeli jedynie w przybliżeniu, jak w przypadku niektórych testowanych próbek.