### 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.

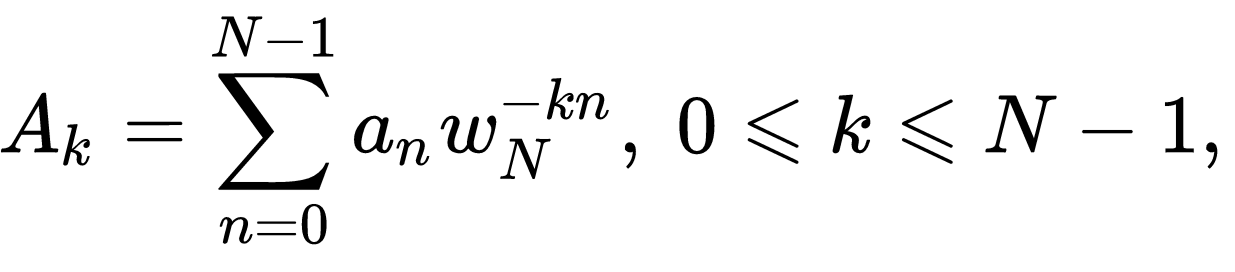
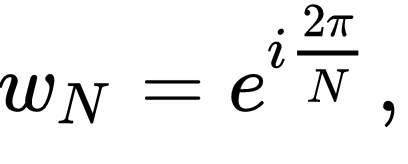
## 2. Wprowadzenie

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,

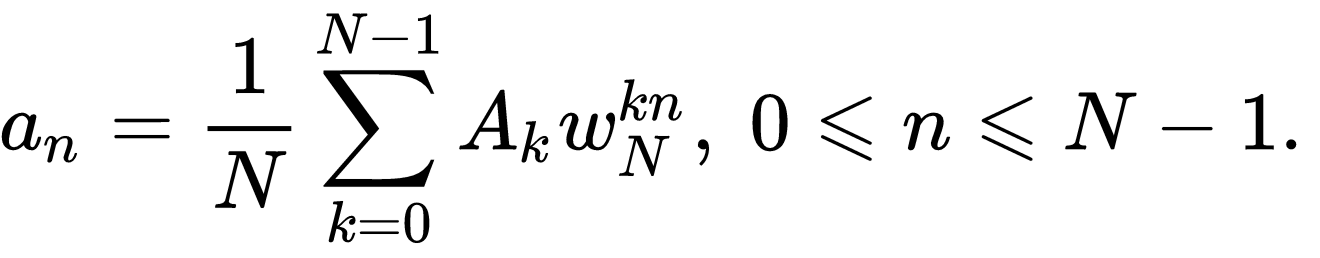
*k* – numer harmoniczny,

*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:**

TO DO

**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 Forma. Rozdzielono działanie aplikacji na dwie oddzielne metody: w dziedzinie częstotliwości i czasu.

To do (screeny z klas)

## 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 |  |  |
| Sztuczny, łatwy | 225Hz |  |  |
| Sztuczny, łatwy | 506Hz |  |  |
| Sztuczny, łatwy | 1139Hz |  |  |
| Sztuczny, średni | 90Hz |  |  |
| Sztuczny, średni | 135Hz |  |  |
| Sztuczny, średni | 303Hz |  |  |
| Sztuczny, średni | 683Hz |  |  |
| Sztuczny, średni | 1537Hz |  |  |
| Sztuczny, trudny | 80Hz |  |  |
| Sztuczny, trudny | 180Hz |  |  |
| Sztuczny, trudny | 405Hz |  |  |
| Sztuczny, trudny | 911Hz |  |  |
| Naturalny, flet | 276Hz |  |  |
| Naturalny, flet | 443Hz |  |  |
| Naturalny, flet | 591Hz |  |  |
| Naturalny, flet | 887Hz |  |  |
| Naturalny, flet | 1265Hz |  |  |
| Naturalny, flet | 1779Hz |  |  |
| Naturalny, viola | 130Hz |  |  |
| Naturalny, viola | 196Hz |  |  |
| Naturalny, viola | 247Hz |  |  |
| Naturalny, viola | 294Hz |  |  |
| Naturalny, viola | 369Hz |  |  |
| Naturalny, viola | 440Hz |  |  |
| Sekwencja | DWK\_violin.wav |  |  |
| Sekwencja | KDF\_piano.wav |  |  |
|  |  |  |  |

## 6. Dyskusja

## 7. Wnioski