Курсовой проект на оценку 3 по курсу дискретного анализа: Быстрое преобразование Фурье

Выполнил студент группы М8О-307Б-20 МАИ *Чекменев Вячеслав.*

Условие

1. Разработать программу на языке C или C++, реализующую указанный алгоритм. Формат входных и выходных данных описан в задании.

2. Реализуйте алгоритм быстрое преобразование Фурье для действительного сигнала. В качестве первого аргумента вашей программе передаётся название mp3 файла который необходимо обработать. Для каждых 4096 отсчётов с шагом задаваемым вторым аргументом. Перед преобразованием Фурье необходимо подействовать на отсчёты окном Ханна.

   В качестве результата для каждого набора отсчётов выведите наибольшее заначение по абсолютной величине полученное после преобразования Фурье.

Метод решения

1. Преобразовать полученный файл mp3 в вектор pcm, то есть значений амплитуд в каждом отсчете с частотой дискретизации 44,1 kHz

2. Из полученного вектора извлечь значения амплитуд идущие через значение, заданное вторым аргументом

3. Далее разбить данные значения на группы по 4096 штук и для каждой применить окно Ханна:
   $$w[n] = \frac{1}{2} * (1 - \cos\frac{2\pi n}{N-1}); 0 \leq n \leq N-1$$
   в моем случае два окна на каждую группу, то есть N = 1024.

4. Затем для каждой группы применить FFT, отнормировать (то есть взять натуральный логарифм) и найти максимум

5. объединить в вектор максимумы в каждой группе

Описание программы

Проект состоит из файла генератора mp3 сэмплов и трех директорий:

- data: хранит файлы .mp3 и .pcm
- pytests: хранит файл с программой вычисляющей то же, что и дано в задании. Также строит график dB(Hz) после FFT

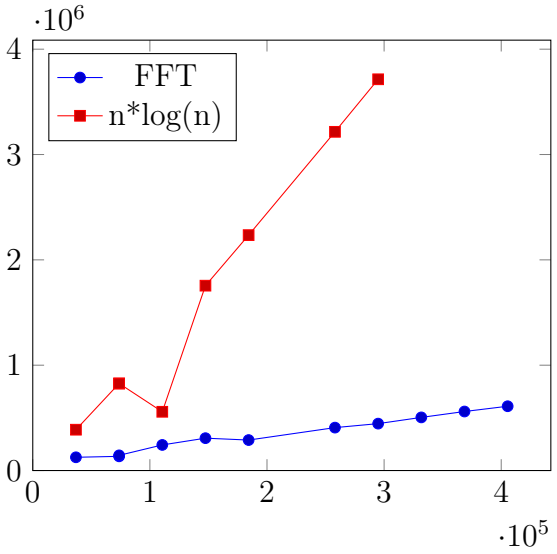
- src: хранит 5 файлов:
  - decoder.hpp: хранит заголовок для функции декодировщика mp3 файла в pcm
  - decoder.cpp: хранит реализацию функции декодировщика mp3 файла в pcm
  - fft.hpp: хранит заголовки для функций FFT и Hann_Window
  - fft.cpp: хранит реализации функций FFT и Hann_Window
  - main.cpp: драйвер

Дневник отладки

1. 20.12.22 Изучена теория
2. 22.12.22 Написаны тесты
3. 25.12.22 Написан код
4. 26.12.22 Протестирован код

Тест производительности

Кол-во элементов в pcm	Время (в мкс)
36864	125777
73728	134815
73728	144147
110592	243023
147456	307250
184320	289367
258048	407468
294912	445275
331776	504473
368640	560305
405504	610382
2174976	3122081
3207168	4875536



Как видно график nlogn ограничивает наш график севрху, поэтому сложность O(n*log(n))

Недочёты

Можно ускорить программу алгоритм с битовыми операциями.

Выводы

Проделав данный курсовой проект я узнал много нового про обработку звука. Понял как работает mp3 и как его преобразовывать в вектор амплитуд. Также разобрался и написал код быстрого преобразования фурье. Протестировал программу на реальных данных и сравнил результаты с работой программы, написанной на питоне с применением функции fft из модуля scipy. Данный проект будет хорошей основой для написания приложения для аудиопоиска.