

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет МИЭМ

Головин Роман Дмитриевич, группа БИТ181

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОИСКА ПОХОЖИХ МУЗЫКАЛЬНЫХ ФАЙЛОВ

Междисциплинарная курсовая работа

По направлению подготовки Инфокоммуникационные технологии и системы связи
образовательная программа «Инфокоммуникационные технологии и системы связи»

Научный руководитель

Доцент

С. А. Хриткин

Москва 2020

Оглавление

Введение	3
Теоретическая информация.....	4
Представление звуковой информации в памяти компьютера	4
Преобразование Фурье	5
Выводы по главе 1	6
Подготовительный этап разработки программы	7
Постановка задачи.....	7
Обоснование выбора языка программирования и средств	7
Разработка программы.....	8
Спецификация программы	8
Соглашение о терминах.....	8
Общее описание.....	8
Описание интерфейса	8
Описание алгоритма.....	10
Описание тестов	12
Примеры работы с программой	15
Выводы по главе 3	19
Заключение.....	20
Список литературы.....	21
Приложение	22
Приложение 1	22
Приложение 2	24
Приложение 3	26
Приложение 4	28

Введение

В современном мире компьютеры нужны не только для проведения каких-либо расчётов, как это было на заре их появления. Сейчас многие люди используют их для игр, хранения информации, просмотра фильмов или прослушивания музыки. И часто бывает, что на компьютере хранится столько фильмов или музыки, что при сохранении новых человек не может быть уверен, что он не сохранял их ранее. При этом поиск по названию или автору может не быть точным на 100%. В этой работе будет предпринята попытка решить задачу поиска похожих файлов в её частном случае – поиске копий (дубликатов) музыкальных файлов или иных аудиофайлов.

Теоретическая информация

Представление звуковой информации в памяти компьютера

В реальном мире звук представляет из себя волну с меняющейся амплитудой и частотой, чем выше амплитуда – тем громче он воспринимается человеком; чем выше частота – тем выше тон звука.

Поскольку память компьютера состоит из битов, то есть она дискретная, то в таком же виде следует хранить и звук. Процедура перевода непрерывного (аналогового) звукового сигнала в дискретный называется дискретизацией. Параметром такого перевода является частота дискретизации – количество измерений сигнала за 1 секунду. Стандартное значение – 44100 Гц, то есть измерения проводятся 44100 раз в секунду. Такой выбор частоты связан с ограничениями человеческого восприятия звука и теоремой Котельникова (диапазон звуков, которые может распознать человек, - до 22050 Гц, а согласно теореме Котельникова, для записи аналогового сигнала с конечным спектром без потерь, требуется частота дискретизации в два раза больше, чем максимальная частота в сигнале; считая спектр ограниченным доступными для человека частотами, получаем минимальную частоту дискретизации в 44100 Гц). Ещё один параметр – глубина кодирования – отвечает за количество бит, отводимых на запись результата одного измерения. Чем выше значение этих параметров, тем лучше качество звучания полученного аудиофайла, но тем выше его объём (зависимость прямая линейная).

Недостаточно просто провести дискретизацию аналогового сигнала, следует кодировать его определённым образом, чтобы появилась возможность записи в виде двоичного кода. Среди методов кодирования стоит выделить два:

1. Наиболее распространённый метод – метод FM (frequency modulation – частотная модуляция) – состоит в том, что любой звук можно разложить на последовательность синусоид с разными частотами и амплитудами, что и позволит их кодировать. Устройства, которые выполняют такую операцию, называются АЦП (аналогово-цифровые преобразователи), а обратную операцию – ЦАП (цифро-аналоговые преобразователи).

2. Менее популярный метод – таблично-волновой – состоит в записи наиболее часто встречающихся звуков (звуки природы, человеческий голос и т.д.) в специальные таблицы, которые затем используются для кодирования исходного сигнала.

Поскольку первый способ значительно более широко используется, остановимся на нём. Его можно реализовать различными способами, что привело к появлению различных форматов, например, WAV или MP3.

WAV представляет звуковую информацию при помощи линейно-импульсной модуляции (LPCM), не сжимая её, из-за этого качество записи получается очень хорошим, но и объёмы занимаемой памяти тоже велики.

В формате MP3 используется алгоритм сжатия с потерями, что позволяет существенно уменьшить объём файла, но ухудшает качество звука.

Т.к. в современных компьютерах объём памяти весьма большой, а устройства для воспроизведения аудио вышли на высокий уровень качества, предпочтительнее выглядит формат WAV.

Преобразование Фурье

Один из важнейших видов информации, который может быть полезно узнать о сигнале – какие у него есть составляющие, то есть получить спектр этого сигнала. Эту задачу в организме человека выполняет часть уха под названием улитка, а математически способ был найден Жаном Батистом Жозефом Фурье ещё в 1807 году и позднее был назван в его честь преобразованием Фурье.

Это преобразование имеет вид

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx), \text{ где}$$

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x) dx,$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos nx \, dx,$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin nx \, dx.$$

Здесь коэффициенты обозначают амплитуды соответствующих им частот, а сами частоты задаются бесконечным рядом синусоид и косинусоид. Существуют также другие формы записи этого ряда, например, комплексная:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}, \text{ где}$$

$$c_0 = \frac{a_0}{2},$$

$$c_n = \frac{a_n - ib_n}{2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx,$$

$$c_{-n} = \frac{a_n + ib_n}{2}.$$

Т.к. в нём используются интегралы, то его использование невозможно при дискретном хранении информации (т.к. интегрируемые функции должны быть непрерывны), поэтому в компьютерной обработке используется аналог – дискретное преобразование Фурье. Однако оно работает сравнительно медленно (сложность $O(N^2)$), что непригодно для обработки файлов больших размеров, поэтому чаще используется быстрое преобразование Фурье, в котором применяется, например, прореживание по частоте и сложность которого уже $O(N \log(N))$.

Выводы по главе 1

1. В нашей программе будет использоваться метод кодирования FM ввиду его большей популярности.
2. Программа будет работать с файлами формата WAV.

Подготовительный этап разработки программы

Постановка задачи

Задача была сформулирована так:

Написать программу, которая будет искать дубликаты входного аудиофайла среди библиотеки уже имеющихся аудиофайлов.

Обоснование выбора языка программирования и средств

В ходе изучения возможностей различных языков в вопросе обработки аудиофайлов было принято решение выбрать язык Python и библиотеку librosa как предлагающие наиболее простой и быстрый способ выполнить поставленную задачу.

Разработка программы

Спецификация программы

Соглашение о терминах

Аудиофайл – файл, содержащий аудиоинформацию в формате WAV.

Дубликат – файл, для которого существует другой файл в том же формате на одном и том же компьютере, причём они оба содержат информацию об идентичных звуковых сигналах, но могут иметь разные названия и другие метаданные.

Входной аудиофайл – аудиофайл, которые получают на вход модули программы и для которого требуется найти дубликат и/или осуществить визуализацию некоторых его характеристик.

Спектрограмма – график, где по оси абсцисс отложено время, по оси ординат – частоты, а цветом изображены амплитуды частот в данные моменты времени.

Акустические отпечатки (отпечатки) – представление аудиофайла в виде совокупности значений, характеризующей его свойства.

Максимумы – одна из разновидностей акустического отпечатка.

Центроиды – одна из разновидностей акустического отпечатка.

NumPy – библиотека языка программирования Python, предназначенная для работы с массивами.

Общее описание

Программа состоит из четырёх модулей, выполненных в виде отдельных программ – «Создание отпечатков», «Поиск дубликатов – максимумы», «Поиск дубликатов – центроиды» и «Визуализация данных».

Описание интерфейса

Входные данные

Модуль «Создание отпечатков» получает на вход строки, содержащие директории, в которых хранится музыка и в которые будут сохранены служебные файлы и акустические отпечатки.

Модули «Поиск дубликатов – максимумы» и «Поиск дубликатов – центроиды» и «Визуализация данных» получают на вход строки, которые содержат пути к

директориям, в которых хранятся служебные файлы, а также входной аудиофайл и его имя.

Модуль «Визуализация данных» получает на вход строку, которая содержит путь ко входному аудиофайлу и его имя.

Выходные данные

Модуль «Создание отпечатков» создаёт в отдельных папках на жёстком диске служебные текстовые файлы, в которых хранятся директории, в которых хранятся акустические отпечатки аудиофайлов, а также файлы, в которых содержатся акустические отпечатки аудиофайлов, с которыми нужно будет сравнивать входной файл.

Модули «Поиск дубликатов – максимумы» и «Поиск дубликатов – центроиды» либо выводят на экран название аудиофайла, дубликатом которого является входной аудиофайл, либо создают в отдельных папках на жёстком диске файлы, в которыми содержатся акустические отпечатки входного аудиофайла.

Модуль «Визуализация данных» выводит на экран графики, а именно:

- график, отображающий массив временных отсчётов входного аудиофайла
- график – спектрограмму входного аудиофайла
- график максимумов входного аудиофайла
- график центроид аудиофайла

Архитектура программы

Как уже было сказано выше, программа состоит из четырёх модулей.

Модуль «Создание отпечатков» предназначен для создания акустических отпечатков вида максимум и центроиды. Максимум является NumPy-массивом, в котором элемент с индексом i равен максимальной амплитуде какой-либо частоты среди тех, которые встречаются во временной отсчёте с индексом i . Центроида является NumPy-массивом, в котором элемент с индексом i считается по формуле:

$$\frac{\sum_i S(i) * f(i)}{\sum_i S(i)}$$

где $S(i)$ — спектральная величина элемента разрешения i , а $f(i)$ — частота элемента i .

Модуль «Поиск дубликатов – максимум» предназначен для поиска дубликатов входного аудиофайла с помощью акустического отпечатка типа максимум.

Модуль «Поиск дубликатов – центроида» предназначен для поиска дубликатов входного аудиофайла с помощью акустического отпечатка типа центроида.

Модуль «Визуализация данных» предназначен для построения графиков, отображающий массив временных отсчётов входного аудиофайла, спектрограмму входного аудиофайла, а также графиков акустических отпечатков входного аудиофайла.

Функциональные требования

Требования к программе:

Программа должна выполнять следующие операции:

- создавать акустических отпечатков вида максимум и центроида
- сохранять их в отдельных папках на жёстком диске
- строить графики спектрограммы, массива временных отсчётов ко времени, а также графики акустических отпечатков

Операция запроса пути к файлу и имени файла:

В случае существования указанного адреса входного аудиофайла программа должна принять его и работать по алгоритму, в противном случае вывести сообщение с просьбой указать существующий адрес.

Требования к пользователю:

Пользователь должен предварительно создать папки для хранения акустических отпечатков и аудиофайлов и/или изменить указанные в программы адреса, для того чтобы они указывали на реально существующие папки, где хранятся аудиофайлы/будут храниться акустические отпечатки.

Описание алгоритма

Модуль «Создание отпечатков»

1. Перед запуском программы пользователь создаёт папку, в которой хранятся аудиофайлы, чьи акустические отпечатки требуется создать, а также папки, в которые следует сохранить акустические отпечатки и служебные файлы, после чего проверяет правильность их хранения в программе.

2. Пользователь вводит пути до этих папок в программу, после чего они записываются в текстовые файлы, которые сохраняются в папку для хранения служебных файлов.

3. В отдельный массив собираются названия всех файлов в формате WAV, которые хранятся в папке для аудиофайлов.

4. Создает акустические отпечатки типа максимум и центроиды.

5. Сохраняет полученные акустические отпечатки в указанные папки.

Модуль «Поиск дубликатов – максимум»

1. Получает на вход строку, содержащую путь к папке со служебными файлами.

2. Проверяет правильность введенного пути, в случае неправильного указания пути выводится предупредительная надпись и см. пункт 1.

3. Получает на вход строку, содержащую путь и имя входного аудиофайла.

4. Проверяет правильность введенного пути и имени, в случае отсутствия такого аудиофайла в компьютере выводится предупредительная надпись и см. пункт 3.

5. Создает акустический отпечаток типа максимум входного аудиофайла.

6. Сравнивает полученный акустический отпечаток с базой акустических отпечатков, уже созданных ранее.

7. При нахождении соответствия выводит на экран название аудиофайла, к которому относится найденный акустический отпечаток, иначе добавляет в базу акустический отпечаток входного аудиофайла и выводит сообщение об отсутствии соответствий.

Модуль «Поиск дубликатов – центроида»

1. Получает на вход строку, содержащую путь к папке со служебными файлами.

2. Проверяет правильность введенного пути, в случае неправильного указания пути выводится предупредительная надпись и см. пункт 1

3. Получает на вход строку, содержащую путь и имя входного аудиофайла.

4. Проверяет правильность введенного пути и имени, в случае отсутствия такого аудиофайла в компьютере выводится предупредительная надпись и см. пункт 3.

5. Создает акустический отпечаток типа центроида входного аудиофайла.

6. Сравнивает полученный акустический отпечаток с базой акустических отпечатков, уже созданных ранее.

7. При нахождении соответствия выводит на экран название аудиофайла, к которому относится найденный акустический отпечаток, иначе добавляет в базу акустический отпечаток входного аудиофайла и выводит сообщение об отсутствии соответствий.

Модуль «Визуализация данных»

1. Получает на вход строку, содержащую путь и имя входного аудиофайла.
2. Проверяет правильность введенного пути и имени, в случае отсутствия такого аудиофайла в компьютере выводится предупредительная надпись и см. пункт 1.
3. Создает акустические отпечатки типа максимум и центроида.
4. Строит все необходимые графики.

Описание тестов

Тесты для модуля «Создание отпечатков»:

Тест 1.

Входные данные: строка с адресом несуществующей директории для хранения служебных файлов.

Выходные данные: строка «Такой директории не существует», после чего осуществляется повторный ввод строки.

Тест 2.

Входные данные: строка с адресом несуществующей директории с аудиофайлами.

Выходные данные: строка «Такой директории не существует», после чего осуществляется повторный ввод строки.

Тест 3.

Входные данные: строка с адресом несуществующей директории для хранения акустических отпечатков.

Выходные данные: строка «Такой директории не существует», после чего осуществляется повторный ввод строки.

Тест 4.

Входные данные: строка с адресом существующей директории с аудиофайлами, в которой отсутствуют аудиофайлы.

Выходные данные: строка «В указанной директории аудиофайлов не обнаружено».

Тест 5.

Входные данные: строка с адресом существующей директории с аудиофайлами, в которой присутствуют аудиофайлы.

Выходные данные: строки «Создание отпечатков началось, ожидайте» и «Создание отпечатков завершено», файлы с акустическими отпечатками.

Тесты для модуля «Поиск дубликатов – центроида»:

Тест 1.

Входные данные: строка с адресом несуществующей директории для хранения служебных файлов.

Выходные данные: строка «Такой директории не существует или выбрана неверная директория» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 2.

Входные данные: строка с адресом существующей, но неверной (т.е. в ней отсутствуют служебные файлы), директории для хранения служебных файлов.

Выходные данные: строка «Такой директории не существует или выбрана неверная директория» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 3.

Входные данные: строка с адресом несуществующего файла.

Выходные данные: строки «Такого файла не существует» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 4.

Входные данные: строка с адресом существующего файла, но расширения, отличного от WAV.

Выходные данные: строки «Выбран неподходящий файл» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 5.

Входные данные: строка с адресом существующего файла формата WAV, для которого существует дубликат.

Выходные данные: строки «Поиск дубликата по методу центроид дал следующие результаты:» и строка с именем файла (или несколько строк с именами файлов), дубликатом которого (которых) является вводный аудиофайл.

Тест 6.

Входные данные: строка с адресом существующего файла формата WAV, для которого не существует дубликат.

Выходные данные: строка «Дубликаты не обнаружены» и файл, содержащий акустический отпечаток типа центроида.

Тесты для модуля «Поиск дубликатов – максимум»

Тест 1.

Входные данные: строка с адресом несуществующей директории для хранения служебных файлов.

Выходные данные: строка «Такой директории не существует или выбрана неверная директория» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 2.

Входные данные: строка с адресом существующей, но неверной (т.е. в ней отсутствуют служебные файлы), директории для хранения служебных файлов.

Выходные данные: строка «Такой директории не существует или выбрана неверная директория» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 3.

Входные данные: строка с адресом несуществующего файла.

Выходные данные: строки «Такого файла не существует» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 4.

Входные данные: строка с адресом существующего файла, но расширения, отличного от WAV.

Выходные данные: строки «Выбран неподходящий файл» и «Введите адрес файла», после чего осуществляется повторный ввод строки.

Тест 5.

Входные данные: строка с адресом существующего файла формата WAV, для которого существует дубликат.

Выходные данные: строки «Поиск дубликата по методу максимумов дал следующие результаты:» и строка с именем файла (или несколько строк с именами файлов), дубликатом которого (которых) является вводный аудиофайл.

Тест 6.

Входные данные: строка с адресом существующего файла формата WAV, для которого не существует дубликат.

Выходные данные: строка «Дубликаты не обнаружены» и файл, содержащий акустический отпечаток типа максимум.

Тесты для модуля «Визуализация данных»

Тест 1.

Входные данные: строка с адресом несуществующим файла.

Выходные данные: строки «Такого файла не существует» и «Введите адрес файла», после чего осуществляется повторный вводы строки.

Тест 2.

Входные данные: строка с адресом существующего файла, но расширения, отличного от WAV.

Выходные данные: строки «Выбран неподходящий файл» и «Введите адрес файла», после чего осуществляется повторный вводы строки.

Тест 3.

Входные данные: строка с адресом существующего файла формата WAV.

Выходные данные: графики спектрограммы, массива временных отсчётов ко времени, а также графики акустических отпечатков.

Примеры работы с программой

Пример работы с модулем «Создание отпечатков» (текст программы доступен в Приложении 1)

```
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Адрес  
Такой директории не существует  
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Адреса  
Введите директорию с аудиофайлами:  
C:\Курсовая работа\Музыка  
Такой директории не существует  
Введите директорию с аудиофайлами:  
C:\Курсовая работа\Адреса  
Введите директорию для хранения отпечатков:  
C:\Курсовая работа\Отпечатки  
В указанной директории аудиофайлов не обнаружено
```

Рис. 1. Тесты 1, 2 и 4.

```
Введите директорию для хранения отпечатков:  
C:\Курсовая работа\Отпечатки  
Такой директории не существует  
Введите директорию для хранения отпечатков:
```

Рис. 2. Тест 3

```
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Адреса  
Введите директорию с аудиофайлами:  
C:\Курсовая работа\Музыка  
Введите директорию для хранения отпечатков:  
C:\Курсовая работа\Отпечатки  
Создание отпечатков началось, ожидайте  
Создание отпечатков завершено
```

Рис. 3. Тест 5.

Пример работы с модулем «Поиск дубликатов – максимумы» (текст программы доступен в Приложении 2)

```
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Адреса\ppath.txt  
Такой директории не существует или выбрана неверная директория  
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Музыка  
Такой директории не существует или выбрана неверная директория
```

Рис. 4. Тесты 1 и 2.

```
Введите адрес файла:  
C:\Курсовая работа\music file 3.wav  
Такого файла не существует  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.mp3  
Выбран неподходящий файл  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.wav  
Дубликаты не обнаружены
```

Рис. 5. Тесты 3, 4 и 6.

```
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 33.wav  
Поиск дубликатов файла по методу максимумов дал следующие результаты:  
music file 3.wav
```

Рис. 6. Тест 6.

Пример работы с модулем «Поиск дубликатов – центроиды» (текст программы доступен в Приложении 3)

```
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Адреса\ppath.txt  
Такой директории не существует или выбрана неверная директория  
Введите директорию для хранения служебных файлов:  
C:\Курсовая работа\Музыка  
Такой директории не существует или выбрана неверная директория
```

Рис. 7. Тесты 1 и 2.

```
Введите адрес файла:  
C:\Курсовая работа\music file 3.wav  
Такого файла не существует  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.mp3  
Выбран неподходящий файл  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.wav  
Дубликаты не обнаружены
```

Рис. 8. Тесты 3, 4 и 6.


```
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.wav  
Поиск дубликатов файла по методу спектральных центроидов дал следующие результаты:  
music file 3.wav
```

Рис. 9. Тест 6.

Пример работы с модулем «Визуализация данных» (текст программы доступен в Приложении 4)

```
Введите адрес файла:  
C:\Курсовая работа\music file 3.wav  
Такого файла не существует  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.mp3  
Выбран неподходящий файл  
Введите адрес файла:  
C:\Курсовая работа\Музыка\music file 3.wav
```

Рис. 10. Тесты 1 – 3.

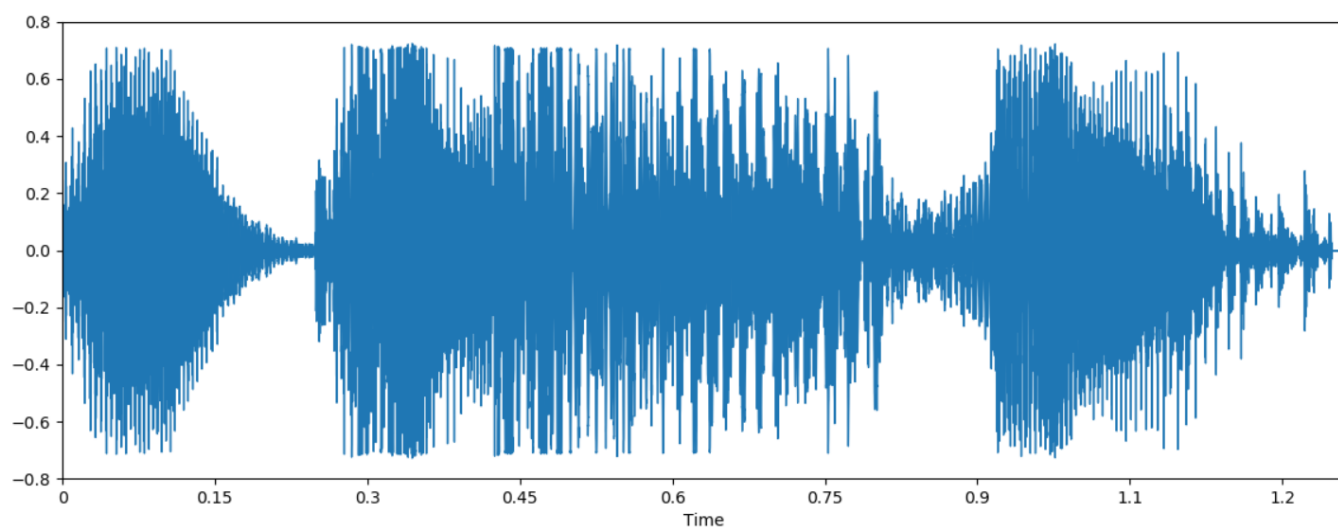


Рис. 11. График массива временных отсчётов ко времени¹.

¹ Этот график и графики ниже будут идентичны для входного аудиофайла и его дубликата

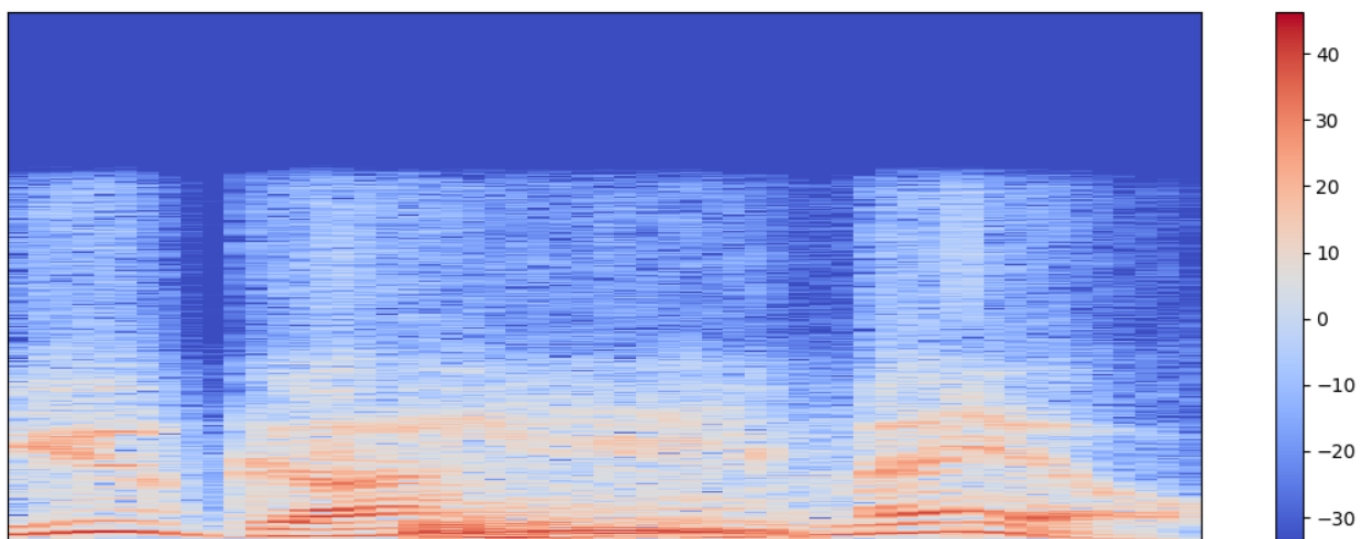


Рис. 12. Спектрограмма

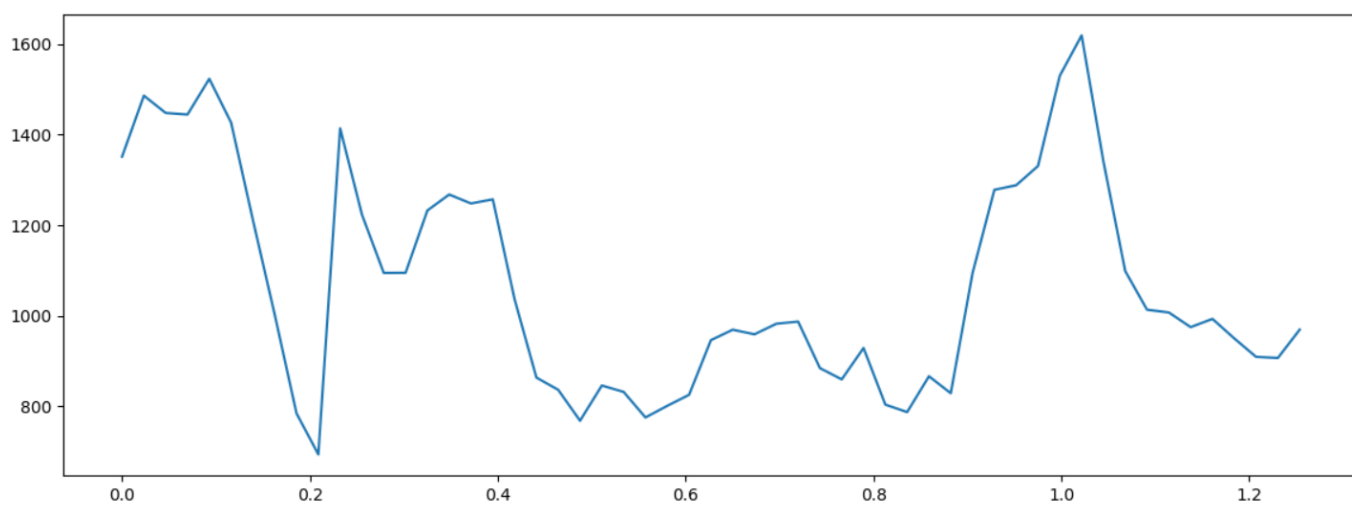


Рис. 13. График акустических отпечатков в виде центриды

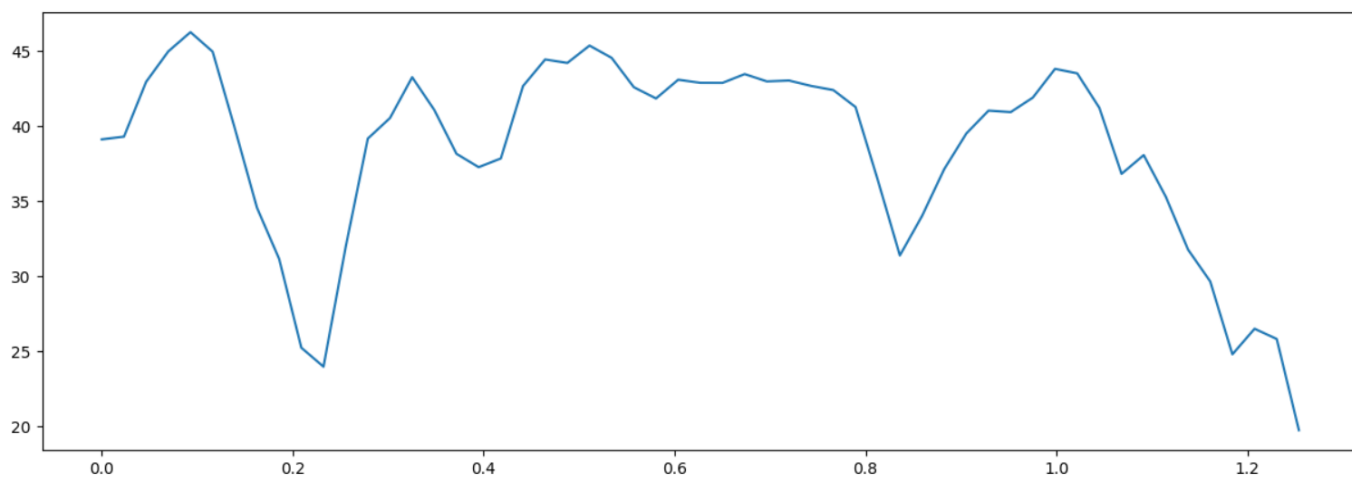


Рис. 14. График акустических отпечатков в виде максимумов

Выводы по главе 3

1. Была реализована программа по поиску похожих программ с помощью акустических отпечатков типа максимум и центроида.
2. Была реализована вспомогательная программа, создающая акустические отпечатки.
3. В качестве дополнения был создан модуль, визуализирующий данные о входном аудиофайле.

Заключение

Созданная программа весьма эффективно решает поставленную задачу по поиску похожих аудиофайлов, причём сразу двумя способами. Было обнаружено, что файлы, хранящие акустические отпечатки типа максимум, занимают примерно в 2 раза меньше места, чем аналогичные файлы с акустическими отпечатками типа центроида, но т.к. в современных ПК различия в несколько килобайт незначительны, а скорость работы обоих методов поиска сопоставима, было принято решение оставить оба метода. В дальнейшем планируется улучшить программу, чтобы избавиться от некоторых ограничений текущей версии:

1. Добавить возможность работы с другими форматами аудиофайлов, такими как MP3 и MIDI.
2. Соединить все модули в одну программу и дать возможность пользователю выбрать места хранения музыки и акустических отпечатков во время, а не до использования программы.

Список литературы

1. Документация к библиотеке NumPy // Сайт [numpy.org](https://numpy.org/doc/).
(<https://numpy.org/doc/>) Просмотрено: 20.04.2020
2. Документация к библиотеке librosa // Сайт librosa.github.io.
(<https://librosa.github.io/librosa/>) Просмотрено: 20.04.2020
3. Документация к библиотеке matplotlib //
(<https://matplotlib.org/contents.html>) Просмотрено: 20.04.2020

Приложение

Приложение 1

Модуль «Создание отпечатков»

```
import librosa
import numpy as np
import glob
import os

while True:
    print('Введите директорию для хранения служебных файлов:')
    data = input()
    if os.path.exists(data):
        break
    if not os.path.exists(data):
        print('Такой директории не существует')

while True:
    print('Введите директорию с аудиофайлами:')
    path = input()
    if os.path.exists(path):
        break
    if not os.path.exists(path):
        print('Такой директории не существует')

while True:
    print('Введите директорию для хранения отпечатков:')
    path_2 = input()
    if os.path.exists(path_2):
        break
    if not os.path.exists(path_2):
        print('Такой директории не существует')

first_path = path_2 + "\Максимумы\\"
second_path = path_2 + "\Центроиды\\"
os.chdir(path)
music_list = []

file = open(data+'\\first_path.txt', "w+")
file.write(first_path)
file.close()
```

```
file = open(data+'\\second_path.txt', "w+")
file.write(second_path)
file.close()

for file in glob.glob("*.wav"):
    music_list.append(file)

if len(music_list) > 0:
    print('Создание отпечатков началось, ожидайте')
else:
    print('В указанной директории аудиофайлов не обнаружено')

for i in range(len(music_list)):
    data = music_list[i]
    x, sr = librosa.load(data)
    X = librosa.stft(x)
    Xdb = librosa.amplitude_to_db(abs(X))

    mass_max = Xdb.max(axis=0)
    final_path = first_path + music_list[i][:len(music_list[i])-4]
    np.save(final_path, mass_max)

    centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
    final_path_2 = second_path + music_list[i][:len(music_list[i])-4]
    np.save(final_path_2, centroids)

if len(music_list) > 0:
    print('Создание отпечатков завершено')
```

Приложение 2

Модуль «Поиск дубликатов – максимумы»

```
import librosa
import numpy as np
import glob
import os

while True:
    print('Введите директорию для хранения служебных файлов:')
    data = input()
    if data[-1] != '\\':
        data += '\\'
    if os.path.exists(data + "first_path.txt"):
        break
    if not os.path.exists(data + "first_path.txt"):
        print('Такой директории не существует или выбрана неверная директория')

file = open(data + "first_path.txt")
path = file.read()
file.close()

while True:
    print('Введите адрес файла:')
    data = input()
    if (os.path.exists(data)) & (data[-4:] == '.wav'):
        break
    if not os.path.exists(data):
        print('Такого файла не существует')
    if data[-4:] != '.wav':
        print('Выбран неподходящий файл')

mass, sr = librosa.load(data)
mass_stft = librosa.stft(mass)
mass_db = librosa.amplitude_to_db(abs(mass_stft))
mass_max = mass_db.max(axis=0)

os.chdir(path)
music_list = []
for file in glob.glob("*.npy"):
    music_list.append(file)
```



```
flag = True
for i in range(len(music_list)):
    mass_2 = np.load(path+music_list[i])
    if np.array_equal(mass_max, mass_2):
        print('Поиск дубликатов файла по методу максимумов дал следующие результаты:')
        print(music_list[i][:3]+'wav')
        flag = False

    if not flag:
        break

if flag:
    final_path = path + data[data.rfind('\\')+1:-4]
    np.save(final_path, mass_max)
    print('Дубликаты не обнаружены')
```

Приложение 3

Модуль «Поиск дубликатов – центроиды»

```
import librosa
import numpy as np
import glob
import os

while True:
    print('Введите директорию для хранения служебных файлов:')
    data = input()
    if data[-1] != '\\':
        data += '\\'
    if os.path.exists(data + "second_path.txt"):
        break
    if not os.path.exists(data + "second_path.txt"):
        print('Такой директории не существует или выбрана неверная директория')

file = open(data + "second_path.txt")
path = file.read()
file.close()

while True:
    print('Введите адрес файла:')
    data = input()
    if (os.path.exists(data)) & (data[-4:] == '.wav'):
        break
    if not os.path.exists(data):
        print('Такого файла не существует')
    if data[-4:] != '.wav':
        print('Выбран неподходящий файл')

mass, sr = librosa.load(data)
new_centroids = librosa.feature.spectral_centroid(mass, sr=sr)[0]
```

```
os.chdir(path)
music_list = []
for file in glob.glob("*.npy"):
    music_list.append(file)

flag = True
print('Поиск дубликатов файла по методу спектральных центроидов дал следующие результаты:')
for i in range(len(music_list)):
    centroids_2 = np.load(path+music_list[i])
    if np.array_equal(new_centroids, centroids_2):
        print(music_list[i][::-3]+'wav')
        flag = False

if flag:
    final_path = path + data[data.rfind('\\')+1:-4]
    np.save(final_path, new_centroids)
    print('Дубликаты не обнаружены')
```

Приложение 4

Модуль «Визуализация данных»

```
import librosa
import matplotlib.pyplot as plt
import librosa.display
import numpy as np

while True:
    print('Введите адрес файла:')
    data = input()
    if (os.path.exists(data)) & (data[-4:] == '.wav'):
        break
    if not os.path.exists(data):
        print('Такого файла не существует')
    if data[-4:] != '.wav':
        print('Выбран неподходящий файл')

mass, sr = librosa.load(data)
mass_stft = librosa.stft(mass)
mass_db = librosa.amplitude_to_db(abs(mass_stft))
mass_max = mass_db.max(axis=0)
centroids = librosa.feature.spectral_centroid(mass, sr=sr)[0]

plt.figure(figsize=(15, 5))
librosa.display.waveplot(mass, sr=sr)

plt.figure(figsize=(15, 5))
librosa.display.specshow(mass_db, sr=sr)

plt.figure(figsize=(15, 5))
frames = range(len(centroids))
t = librosa.frames_to_time(frames)
plt.plot(t, centroids)

plt.figure(figsize=(15, 5))
plt.plot(t, mass_max)
plt.show()
```