**Применения Фурье-анализа в software. Shazam**

**(Паша)**

**—**

**Всем привет! Рады видеть вас у нас на проекте-лекции, в которой мы расскажем вам о том, как применяется Фурье-анализ в обыденных вещами, которыми вы пользуетесь каждый день — в изображениях и музыке. И всё это прямо в вашем смартфоне! Начнём с самого интересного, с музыки и Шазама**

**—**

**Shazam** — это, как все знают, приложение для распознавания песен.

Но прежде чем стать приложением, Shazam был телефонным номером (2580). Чтобы определить песню, нужно было позвонить по этому номеру и поднести микрофон телефона к источнику музыки. Через 30 секунд Shazam завершал звонок и присылал СМС с информацией о прослушанной песне.

**Алгоритм**

Поговорим о том, как всё работает под капотом. После нажатия на кнопку распознавания Shazam запускает два потока: для сохранения и для распознавания. Поток сохранения сохраняет песню в базу данных для ее распознавания в будущем, так как вполне возможно, что фрагмент, который прозвучал в песне, пока ещё отсутствует в базе — таким образом Shazam сможет связать неизвестный отрывок с распознанной песней. Поток распознавания в свою очередь идентифицирует песню по фрагменту в реал-тайме, нетрудно догадаться.

Оба потока включают следующие шаги:

* **Расчет спектрограммы аудиофайла** – график частоты относительно времени с применением преобразования Фурье.
* **Поиск пиков на спектрограмме**. Они отражают самые громкие частоты в аудиозаписи и помогут нам создать отпечаток. Здесь используется фильтр максимумов, с которым мы познакомимся дальше
* **Хеширование пиков**. Если кратко, за этим скрывается составление пиков в пары для создания отпечатка.

После расчёта хешей поток сохранения запишет их в базе данных. Поток распознавания будет сравнивать рассчитанные хеши с хешами, уже находящимися в базе данных, чтобы определить, какая песня играет сейчас. Как только поток распознавания отыщет песню, которая сейчас играет, поток сохранения свяжет записанные в базе данных хеши с найденной песней.

**Построение спектрограммы**

Первый шаг для обоих потоков — получить спектрограмму аудио, для сохранения или распознавания.В этом нам поможет дискретное преобразование Фурье (ДПФ, DFT, Discrete Fourier Transform).

Кто смотрел 6 лекцию молодец, а для тех, кто не смотрел, ссылку на неё мы оставим в описании, чтобы вы могли ознакомиться с ДПФ подробнее. Если коротко, то ДПФ – это математический метод анализа Фурье для дискретных сигналов. С его помощью можно преобразовать конечный набор образцов сигнала, взятых с равными промежутками времени, в список коэффициентов конечной комбинации комплексных синусоид, упорядоченных по частоте, принимая во внимание, что эти синусоиды были дискретизированы с одной и той же частотой.

В ходе вычислений используется рекурсивное разложение исходного ДПФ на мелкие части. Прямое вычисление ДПФ для некоторого набора данных размерностью n требует O(n^2) операций, а использование алгоритмов ДПФ позволяет решить ту же задачу за O(n log n) операций.

Но если применить преобразование Фурье к целой песне, мы увидим амплитуды присутствующих в ней частот на всём её протяжении. Однако, на разных участках песни набор частот может различаться. Чтобы лучше представить изменение частот по времени, лучше сначала разбить песню на небольшие секции, а потом выполнить преобразование Фурье. Это и называется созданием спектрограммы.

Фрагмент видео, как это выглядит: <https://leonardo.osnova.io/5ac6deac-1326-5bde-b93f-d1e5d6c9afaa/-/format/mp4#t=0.1>

Спектрограмма содержит три измерения:

* Время (ось X)
* Частота (ось Y)
* Амплитуда (ось Z/цвет)

Ось Z представлена цветом на приведённой выше спектрограмме. Светло-зелёный означает большую амплитуду для определенного частотного компонента, а тёмно-синий — меньшую.

**(Маша)**

**Создание отпечатка или цифровой сигнатуры (А какая твоя сигнатура?)**

Принцип создания отпечатков строится на поиске частотных пиков на спектрограмме. Эти пики представляют собой самые громкие частоты в определённом временном сегменте песни. Поскольку они самые громкие, есть вероятность, что они останутся различимы при наличии в записи фонового шума или других искажений.

Еще одним преимуществом использования пиков спектрограммы для создания отпечатков является то, что они сокращают объем данных, которые придётся хранить. Хранение только самых громких частотных компонентов ускоряет поиск совпадений.

Также важно, чтобы пики были равномерно распределены по частотам, чтобы шумы и частотные искажения не были препятствием к распознаванию.

Чтобы найти пики спектрограммы, при этом сохраняя их распределение, мы можем воспользоваться техникой из области обработки изображений – фильтр максимумов. Процесс выглядит примерно следующим образом.

**Шаг 1: Фильтр максимумов**

Фильтр максимумов делает пики заметнее. Для этого он берет каждый пиксель изображения по очереди и находит максимум в области вокруг него. Затем пикселю присваивается значение, равное этому локальному максимуму. В следующей анимации показан максимальный фильтр, который рассматривает окно 3x3 вокруг каждого пикселя:

<https://leonardo.osnova.io/7ce5b518-ec84-5af2-9469-429344324597/-/format/mp4#t=0.1>

Меньший размер окна в результате будет давать большее количество пиков, и наоборот.

В результате получим спектрограмму с более низким разрешением, поскольку значения локальных максимумов распространилось на пиксели вокруг них.

<https://leonardo.osnova.io/6f8eae6a-4290-521f-9562-d2afcd1c1ad6/-/preview/1200x/-/format/webp>

**Шаг 2: Восстановление исходных пиков**

Чтобы найти местоположения пиков, нужно найти точки, значения которых в исходной и “отфильтрованной” спектрограмме совпадают. Если отметить все найденные таким образом пики на графике, убрав всё лишнее, мы получим визуализацию, называемую картой созвездий.

<https://leonardo.osnova.io/fbb84252-c53e-594c-a906-6037ca4b93ad/-/preview/1200x/-/format/webp>

**Шаг 3: Отбрасывание пиков**

Следующий шаг – при необходимости отбрасывание некоторых пиков. Размер отпечатка зависит от количества пиков, которые в нём используются. Однако, уменьшая количество используемых пиков для достижения большей скорости поиска, мы уменьшаем точность нашей системы, поэтому нам необходимо найти что-то среднее между точностью и скоростью.

**Хеширование**

Для упрощения поиска музыкальных композиций их сигнатуры используются как ключи в хэш-таблице. Ключам соответствуют значения времени, когда набор частот, для которых найдена сигнатура, появился в произведении, и идентификатор самого произведения (название песни и имя исполнителя, например). Вот вариант того, как подобные записи могут выглядеть в базе данных.

Если обработать таким способом некую библиотеку музыкальных записей, можно будет построить базу данных с полными сигнатурами каждого произведения.

**Сопоставление**

У многих фрагментов различных произведений хэш-тэги совпадают. В таком случае, если найдено несколько совпадений, можно проанализировать относительный тайминг совпадений, и, таким образом, повысить достоверность поиска.

Это работает так. Для каждой песни сопоставляется время от начала, когда хеш встречается в оригинальной песне, со временем, когда хеш встречается в анализируемом фрагменте.

Таким образом, мы можем сравнить аудиофрагмент со всеми песнями с совпадающими хешами в базе данных и присвоить оценку каждой из них. Песня с наивысшей оценкой, скорее всего, и будет искомым результатом.

**Источники:**

1. [Как работает Shazam? Простое объяснение алгоритма распознавания аудио — Разработка на vc.ru](https://vc.ru/dev/965228-kak-rabotaet-shazam-prostoe-obyasnenie-algoritma-raspoznavaniya-audio)
2. [Shazam: алгоритмы распознавания музыки, сигнатуры, обработка данных](https://habr.com/ru/companies/wunderfund/articles/275043/)