Высоконадёжная система дактилоскопирования звука

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.2175&rep=rep1&type=pdf&ei=29LKTJH8GYadOony5KYB&usg=AFQjCNE28fHasUbb3lmY93txMMfK5r-4AA

A Highly Robust Audio Fingerprinting System

Jaap Haitsma
Philips Research
Prof. Holstlaan 4, 5616 BA,
Eindhoven, The Netherlands
++31-40-2742648
Jaap.Haitsma@philips.com

Ton Kalker
Philips Research
Prof. Holstlaan 4, 5616 BA,
Eindhoven, The Netherlands
++31-40-2743839
Ton.Kalker@ieee.org

КРАТКИЙ ОБЗОР

Представьте себе следующую ситуацию. Вы находитесь в автомобиле, слушаете радио и вдруг вы слышите песню, которая захватывает ваше внимание. Это лучшая новая песня, которую вы слышали за долгое время, но вы пропустили анонс и не знаете исполнителя. Тем не менее, вы хотели бы узнать побольше об этой музыке. Что делать? Вы могли бы позвонить на радиостанцию, но это слишком обременительно. Было бы неплохо, если бы Вы могли нажать несколько кнопок на своем мобильном телефоне и через несколько секунд телефон бы ответил именем исполнителя и названием музыку, которую вы слушаете? Может быть, даже отправив сообщение на ваш адрес электронной почты с какой-то дополнительной информацией. В этой статье мы представляем систему дактилоскопии звука, которая делает вышеописанный сценарий возможным. С помощью отпечатка неизвестного аудио клипа в качестве запроса к базе данных отпечатков, которая содержит отпечатки большой библиотеки песен, аудио клип может быть идентифицирован. Основой представленной системы является очень надёжный

метод получения отпечатков и очень эффективная стратегия поиска отпечатка, которая позволяет искать в большой базе данных отпечатков с помощью лишь очень ограниченных вычислительных ресурсов.

1. ВВЕДЕНИЕ

Системам дактилоскопирования более ста лет. В 1893 году сэр Фрэнсис Гальтон был первым, кто "доказал", что нет людей с двумя одинаковыми отпечатками пальцев. Примерно 10 лет спустя Скотланд-Ярд принял систему, разработанную сэром Эдвардом Генри для идентификации отпечатков пальцев людей. Эта система опирается на шаблон кожных гребней на кончиках пальцев и по-прежнему лежит сегодня в основе всех методов "человеческих" отпечатков пальцев. Этот тип судебной системы "человеческой" дактилоскопии, однако, существовал больше века, так как 2000 лет назад китайские императоры уже использовали отпечатки большого пальца для подписи важных документов. Видимо, что уже эти императоры (или, по крайней мере, их административные служащие) поняли, что каждый отпечаток пальца был уникальным. Концептуально отпечаток пальца можно рассматривать как "человеческое" резюме или подпись, которая является уникальной для каждого человека. Важно отметить, что отпечаток пальца человека отличается от текстового резюме в том, что он не допускает реконструкции других аспектов оригинала. Например, отпечаток пальца человека не сообщает никакой информации о цвете волос или глаз человека.

В последние годы наблюдается растущий научный и производственный интерес к вычисляемым отпечаткам мультимедийных объектов [1][2][3][4][5][6]. Растущий промышленный интерес показывает среди прочего большое количество (запускающихся) компаний [7][8][9][10] [11][12][13] и недавний запрос для получения информации о технологиях аудио дактилоскопии от Международной федерации фонографической промышленности (International Federation of the Phonographic Industry, IFPI) и Ассоциации звукозаписывающей индустрии Америки (Recording Industry Association of America, RIAA) [14].

Главной целью мультимедийных отпечатков является эффективный механизм для установления равенства восприятия двух мультимедийных объектов: не путём сравнения (обычно больших) самих объектов, а путём сравнения соответствующих отпечатков (разработанными быть небольшими). В большинстве систем с использованием технологии снятия отпечатков, отпечатки большого числа мультимедийных объектов, а также связанные с ними мета-данные (например, имя исполнителя, название песни и альбома) хранятся в базе данных. Отпечатки служат указателем на мета-данные. Затем извлекаются мета-данные неопознанного мультимедийного контента путём вычисления отпечатка и используя его в качестве запроса в базу данных отпечатков/мета-данных. Преимуществ использования отпечатков вместо мультимедийного контента три:

1. Уменьшение требований к памяти/хранилищу, так как отпечатки относительно невелики;
2. Эффективное сравнение, так как невоспринимаемые различия уже были удалены из отпечатков;
3. Эффективный поиск, так как набор данных для поиска меньше.

Как можно заключить из вышесказанного, дактилоскопическая система обычно состоит из двух компонентов: метода получения отпечатков и метода эффективного поиска для сравнения отпечатков в базе данных отпечатков.

Эта статья описывает систему получения отпечатков звука, которая подходит для большого числа приложений. После определения концепции аудио отпечатков в Разделе 2 3 и

разработки возможных применений в Разделе 3 [5], мы фокусируем внимание на технических аспектах предлагаемой системы дактилоскопии звука. Получение отпечатка описано в Разделе 4 [6], а поиск отпечатка - в Разделе 5 [14].

2. КОНЦЕПЦИИ ДАКТИЛОСКОПИРОВАНИЯ ЗВУКА

2.1 Определение звукового отпечатка

Напомним, что звуковой отпечаток можно рассматривать как краткое описание аудио объекта. Поэтому функция отпечатков ***F*** должна связывать аудио объект ***X***, состоящий из большого числа битов, с отпечатком, содержащем лишь ограниченное число битов.

Здесь можно провести аналогию с так называемыми хэш-функциями (в дактилоскопической литературе иногда также называется как устойчивое или перцептуальное хэширование [5]), которые хорошо известны в криптографии. Криптографическая хэш-функции ***H*** связана с (обычно большим) объектом ***X*** с (обычно небольшим) хэш-значением (так называемый дайджест сообщения). Криптографическая хэш-функция позволяет сравнение двух крупных объектов ***X*** и ***Y***, простым сравнением их соответствующих хэш-значений ***H(X)*** и ***H(Y)***. Строгое математическое равенство последней пары подразумевает равенство первой лишь с очень малой вероятностью ошибки. Для правильно разработанной криптографической хэш-функции эта вероятность равна 2^{-N}, где ***n*** равно числу бит в этом хэш-значении. При использовании криптографических хэш-функций существует эффективный метод, чтобы проверить, содержится или нет элемент данных ***X*** в заданном и большом наборе данных $Y=\{Y_i\}$. Вместо хранения и сравнения со всеми данными в ***Y***, достаточно сохранить набор хэш-значений $\{h_i = H(Y_i)\}$, и сравнить ***H(X)*** с таким набором хэш-значений.

Сначала можно подумать, что криптографические хэш-функции являются хорошим кандидатом для функций отпечатков. Однако, вспомните из введения, что вместо строгого математического равенства мы заинтересованы в сходстве восприятия. Например, оригинальная версия с качеством компакт-диск 'Rolling Stones - Angie' и звук MP3 версии на 128 Кб/сек в человеческой слуховой системе одинаковы, но форма их сигналов может быть совершенно разной. Хотя две версии воспринимаются одинаково, математически они совершенно отличны. Поэтому криптографические хэш-функции не могут принимать решение о воспринимаемом равенстве этих двух версий. Ещё хуже, криптографические хэш-функции, как правило, чувствительны к битам: разница в один бит в оригинальном объекте приводит к совершенно другому значению хэша.

Другой законный вопрос, который может задать читатель: "Разве невозможно разработать функцию дактилоскопирования, которая создаёт математически равные отпечатки для объектов, воспринимаемых одинаково"? Вопрос правильный, но ответ в том, что такое моделирование сходства восприятия принципиально не представляется возможным. Чтобы быть более точными: известный факт, что сходство восприятия не является транзитивным. Воспринимаемое сходство пары объектов ***X*** и ***Y*** и другой пары объектов ***Y*** и ***Z*** не обязательно подразумевает сходство восприятия объектов ***X*** и ***Z***. Однако, моделирование сходства восприятия путём математического равенства отпечатков могло бы привести к таким отношениям.

Учитывая приведённые выше аргументы, мы предлагаем построить функцию дактилоскопирования таким образом, чтобы воспринимаемые одинаково звуковые объекты

приводили к одинаковым отпечаткам. Кроме того, для того, чтобы быть в состоянии различать различные звуковые объекты, должна быть очень высокая вероятность того, что непохожие звуковые объекты создают непохожие отпечатки. Подробнее математически, для правильно разработанной функции дактилоскопирования F должен быть такой порог T, чтобы с очень высокой вероятностью $||F(X)-F(Y)||\leq T$, если объекты X и Y являются похожими, и $||F(X)-F(Y)||>T$, когда они непохожи.

2.2 Параметры системы дактилоскопирования звука

Имея соответствующее определения аудио отпечатка, сосредоточимся теперь на различных параметрах системы дактилоскопирования звука. Основными параметрами являются:

- **Надёжность**: могут ли аудио клипы быть все ещё идентифицированы после серьёзного искажения сигнала? Для достижения высокой надёжности отпечаток должен быть основан на воспринимаемых особенностях, которые неизменны (по крайней мере в определённой степени) по отношению к искажениям сигнала. Предпочтительно, чтобы сильно искажённый звук по-прежнему приводил к очень похожим отпечаткам. Для выражения надёжности как правило используется частота ложноотрицательных сравнений. Ложноотрицательные сравнения происходят, когда отпечатки воспринимаемых похожими звуковых отрывков слишком разные, чтобы привести к положительному сравнению.

- **Достоверность**: как часто песня идентифицируется неправильно? Например "Rolling Stones - Angie" идентифицируется как "Beatles - Yesterday". Частота, с которой это происходит, обычно называется частотой ложных срабатываний.

- **Размер отпечатка**: сколько памяти необходимо для отпечатка? Чтобы сделать возможным быстрый поиск, отпечатки обычно хранятся в оперативной памяти. Поэтому размер отпечатка, обычно выражаемый в битах в секунду или битах на песню, в значительной степени определяет ресурсы памяти, которые необходимы для сервера базы данных отпечатков.

- **Степень детализации**: сколько секунд звука необходимо для идентификации аудио клипа? Степень детализации - параметр, который может зависеть от приложения. В одних приложениях для идентификации может быть использована вся песня, в других предпочтительно идентифицировать песню только по короткому кусочку звука.

- **Скорость поиска и масштабируемость**: Сколько необходимо времени, чтобы найти отпечаток в базе данных отпечатков? Что делать, если база данных содержит тысячи и тысячи песен? Для коммерческого развёртывания систем дактилоскопирования звука скорость поиска и масштабируемость является ключевым параметром. Скорость поиска должна быть порядка миллисекунд для базы данных, содержащей более 100,000 песен, используя лишь ограниченные вычислительные ресурсы (например, несколько ПК высокого класса).

Эти пять основных параметров имеют большое влияние друг на друга. Например, если захотеть малую детализацию, необходимо получить больший отпечаток для получения той же надежности. Это связано с тем, что процент ложных срабатываний обратно пропорционален размеру отпечатка. Другой пример: скорость поиска обычно увеличивается, когда разработан более надёжный отпечаток. Это связано с тем, что поиск отпечатка - это поиск сходства. То есть должен быть найден похожий (или наиболее похожий) отпечаток. Если признаки являются более надёжными, сходство меньше. Поэтому скорость поиска может возрасти.

3. ПРИМЕНЕНИЯ

В этом разделе мы остановимся на разных применениях дактилоскопии звука.

3.1 Мониторинг вещания

Мониторинг вещания, вероятно, наиболее известное применение для звуковой дактилоскопии [2][3][4][5][12][13]. Это относится к автоматической генерации плейлиста радио, телевидения или веб-трансляций, среди других целей: сбор вознаграждений, верификация программ, проверка рекламы и измерение людей. В настоящее время мониторинг вещания по-прежнему ручной процесс: то есть организации, заинтересованные в плейлистах, такие как организации производителей и правозащитные организации, в настоящее время имеют "настоящих" людей, слушающих передачи и заполняющих таблицы.

Крупномасштабные системы мониторинга вещания, базирующиеся на снятии отпечатков, состоят из нескольких сайтов мониторинга и центрального сайта, где расположен сервер отпечатков. На сайтах мониторинга извлекаются отпечатки из всех (местных) каналов вещания. Центральный сайт собирает отпечатки с сайтов мониторинга. Впоследствии сервер отпечатков, содержащий огромную базу данных отпечатков, создаёт плейлисты всех каналов вещания.

3.2 Связанное со звуком

Связанное со звуком (Connected Audio) - это общий термин для потребительских приложений, где музыка как-то связана с дополнительной и вспомогательной информацией. Пример, приведённый в кратком обзоре: использование мобильного телефона, чтобы идентифицировать песню, является одним из таких примеров. Этот бизнес на самом деле ведётся рядом компаний [10][13]. Звуковой сигнал в это приложении серьёзно искажён из-за обработки, сделанной радиостанциями, FM/AM передатчика, акустического пути между громкоговорителем и микрофоном мобильного телефона, кодирования речи и, наконец, передачей по мобильной сети. Таким образом, с технической точки зрения это очень сложные приложения.

Другими примерами, связанными со звуком, являются (автомобильные) радиостанции с кнопкой идентификации или дактилоскопические приложения "прослушивающие" аудио-потоки, выходящие или входящие в звуковую карту на ПК. При нажатии кнопки "Информация" на таком дактилоскопическом приложении пользователь может быть направлен на страницу в сети Интернет, содержащую информацию об исполнителе. Или, нажав кнопку "Купить", пользователь мог бы купить альбом в Интернете. Иными словами, дактилоскопия звука может обеспечить универсальную систему связи для звукового контента.

3.3 Технология фильтрации для общего доступа к файлам

Фильтрация имеет отношение к активному вмешательству в распространении контента. Ярким примером технологии фильтрации для обмена файлами был Napster [15]. Начиная с июня 1999 года пользователи, которые скачали клиент Napster, могли делиться и загружать большую коллекцию музыки бесплатно. Позже, из-за судебного дела с музыкальной индустрией пользователям Napster-а было запрещено загружать песни, защищённые авторским правом. Поэтому в марте 2001 Napster установил звуковой фильтр, основанный на именах файлов, чтобы блокировать загрузку песен, защищённых авторскими правами. Фильтр был не

очень эффективным, так как пользователи начали намеренно допускать ошибки в именах файлов. В мае 2001 года Napster представил систему дактилоскопии звука от Relatable [8], которая направлена на фильтрацию защищённого авторским правом материала, даже если в названии были ошибки. Вследствие закрытия Napster только два месяца спустя, эффективность данной системы дактилоскопии, насколько известно автору, публично неизвестна.

В легальной службе обмена файлами могли бы применяться более изысканные схемы, чем просто фильтрация защищённого авторским правом материала. Можно было бы подумать о схеме с бесплатной музыкой, различных видах первосортной музыки (доступной для людей с надлежащей подпиской) и запрещенной музыкой.

Хотя с точки зрения потребителя фильтрацию звука можно рассматривать как отрицательную технологию, для потребителей есть также ряд потенциальных выгод. Во-первых, можно организовать названия песен в результатах поиска на постоянной основе с помощью надёжных мета-данных из базы данных отпечатков. Во-вторых, снятие отпечатков может гарантировать, что то, что загружено, на самом деле то, о чём был сделан запрос.

3.4 Автоматическая организация музыкальной библиотеки

В настоящее время многие пользователи ПК имеют музыкальную библиотеку, содержащую несколько сотен, иногда даже тысячи песен. Музыка, как правило, хранится в сжатом формате (обычно MP3) на жёстких дисках. Так как эти песни получены из разных источников, таких как копирование с компакт-дисков или загрузки из сетей обмена файлами, такие библиотеки зачастую не очень хорошо организованы. Мета-данные зачастую противоречивы, неполны и иногда даже неправильны. Предполагая, что база данных отпечатков содержит правильные мета-данные, дактилоскопия звука сможет поместить совместимые мета-данные из песни в библиотеку, позволяя легко организовать её на основе, например, альбома или исполнителя. Например, ID3Man [16], инструмент дактилоскопической технологии, разработанный Auditude [7], уже доступен для пометки непомеченных или ошибочно помеченных файлов MP3. Аналогичный инструмент от Moodlogic [11] доступен как плагин Winamp [17].

4. ПОЛУЧЕНИЕ ОТПЕЧАТКОВ ЗВУКА

4.1 Основные принципы

Звуковые отпечатки собираются уловить особенности звука, имеющие отношение к восприятию. В то же время получение и поиск отпечатков должно быть быстрым и простым, желательно с небольшой детализацией, чтобы позволить использование в очень востребованных приложениях (например, распознавание в мобильном телефоне). До начала разработки и реализации такой схемы снятия отпечатков должны быть рассмотрены несколько фундаментальных вопросов. Наиболее очевидный вопрос для обсуждения: какие особенности являются наиболее подходящими. Просмотр существующей литературы показывает, что набор соответствующих особенностей можно разделить на два класса: класс семантических особенностей и класс не-семантических особенностей. Типичными элементами первого класса являются ***жанр***, ***число ударов в минуту*** и ***настроение***. Эти типы особенностей обычно имеют прямое толкование и на самом деле используются для классификации музыки, создания плей-листов и многого другого. Второй класс состоит из особенностей, которые имеют более математический характер и трудны для "чтения" человеком непосредственно из музыки. Типичным элементом в этом классе является ***AudioFlatness***, который предлагается в MPEG-7

в качестве инструмента описания звука [2]. Для работы, описанной в этой статье, мы по ряду причин явно выбрали работу с не-семантическими особенностями:

1. Семантические особенности не всегда имеют чёткий и однозначный смысл. То есть личное мнение сильно отличается для таких классификаций. Кроме того, семантика может на самом деле изменяться с течением времени. Например, музыку, которая была классифицирована как ***хард-рок*** 25 лет назад, можно рассматривать сегодня как ***приятную музыку***. Это делает математический анализ затруднительным.

2. Семантические особенности являются в целом более сложно вычислимыми, чем не-семантические особенности.

3. Семантические особенности не универсальны. Например, ***число ударов в минуту*** обычно не применяется к классической музыке.

Вторым вопросом, который предстоит рассмотреть, является представление отпечатков. Очевидным кандидатом является представление в виде вектора вещественных чисел, где каждый компонент выражает вес некоторой основной особенности восприятия. Второй вариант - оставаться ближе по духу к криптографическим хэш-функциям и представить цифровые отпечатки как битовые строки. По причинам сокращения сложности поиска мы решили в этой работе выбрать второй вариант. Первый вариант предполагал бы измерение сходства с помощью суммирований/вычитаний реальных чисел и, в зависимости от меры сходства, возможно, даже умножения реальных чисел. Отпечатки, которые основаны на битовых представлениях, можно сравнивать просто считая биты. С учётом ожидаемых сценариев применения мы не ожидаем высокой надёжности для каждого бита в таких бинарных отпечатках. Поэтому в отличие от криптографических хэшей, которые обычно имеют по крайней мере несколько сотен бит, мы позволим отпечатки, которые имеют несколько тысяч бит. Отпечатки, содержащие большое количество битов, позволяют надёжную идентификацию, даже если процент несовпадающих битов является относительно высоким.

Последний вопрос относится к степени детализации отпечатков. В приложениях, которые мы предусматриваем, нет никакой гарантии, что аудио файлы, которые должны быть определены, являются полными. Например, при мониторинге эфир ***любой*** интервал от 5 секунд является единицей музыки, которая имеет коммерческую ценность, и поэтому, возможно, должен быть идентифицирован и опознан. Кроме того, в приложениях безопасности, таких как фильтрация файлов в обменной сети, никто не хотел бы, чтобы исключение первых нескольких секунд звукового файла препятствовало идентификации. В данной работе мы поэтому адаптировали контроль ***потоков отпечатков*** путём создания ***суб-отпечатков*** для достаточно малых базовых интервалов (названных ***кадрами***). Эти суб-отпечатки не могут быть достаточно большими, чтобы непосредственно идентифицировать кадры, но более длительный интервал, содержащий достаточно много кадров, позволит надёжную и достоверную идентификацию.

4.2 Алгоритм получения

Большинство алгоритмов получения отпечатков основаны на следующем подходе. Сначала звуковой сигнал разделяется на кадры. Для каждого кадра вычисляется набор характеристик. Предпочтительно характеристики выбраны так, что они инвариантны (по крайней мере в определённой степени) к искажениям сигнала. Предложенными характеристиками являются хорошо известные характеристики звука, такие как коэффициенты Фурье [4], коэффициенты косинусного преобразования Фурье (Mel Frequency Cepstral Coefficients, MFFC) [18],

неравномерность спектра (spectral flatness) [2], чёткость звучания (sharpness) [2], коэффициенты линейное кодирования с предсказанием (Linear Predictive Coding, LPC) [2] и другие. Также используются производные величины параметров звука, такие как производные, средний уровень и дисперсия. Вообще, полученные характеристики переходят в более компактное представление, используя алгоритмы классификации, такие как скрытые Марковские модели [3] или квантование [5]. Компактное представление одного кадра будем называть ***суб-отпечатком***. Полная процедура получения отпечатков преобразует поток звука в поток суб-отпечатков. Один суб-отпечаток обычно не содержит достаточных данных для идентификации аудио клипа. Основная единица, которая содержит достаточно данных для идентификации аудио клипа (и, следовательно, определяет детализацию), будет называться ***блок отпечатков***.

Предложенная схема получения отпечатков базируется на этом подходе обработки потока. Она создаёт 32-х разрядной суб-отпечаток для каждого интервала в 11.6 миллисекунд. Блок отпечатков состоит из 256 последовательных суб-отпечатков, что соответствует детализации всего лишь в 3 секунды. Обзор схемы показан на Рисунке 1. Аудио сигнал сначала разделяется на ***перекрывающиеся*** кадры. Перекрывающиеся кадры имеют длительность 0.37 секунды и взвешиваются окном Ханна с фактором перекрытия 31/32 (так в оригинале, правильнее, видимо, Перекрывающиеся кадры имеют длительность 0.37 секунды с фактором перекрытия 31/32 и взвешиваются окном Ханна). Эта стратегия приводит к получению одного суб-отпечатка каждые 11.6 мс. В худшем случае границы кадра, используемые во время идентификации, смещены на 5.8 миллисекунд относительно границ, использованных в базе данных предварительно вычисленных отпечатков. Большое перекрытия гарантирует, что даже в этом худшем случае суб-отпечатки аудиоклипа, который должен быть идентифицирован, всё ещё очень похожи на суб-отпечатки того же клипа в базе данных. Из-за большого перекрытия последовательные суб-отпечатки имеют большое сходство и медленно меняются во времени. На Рисунке 2а показан пример получения блока отпечатков и их медленно меняющийся характер вдоль оси времени.

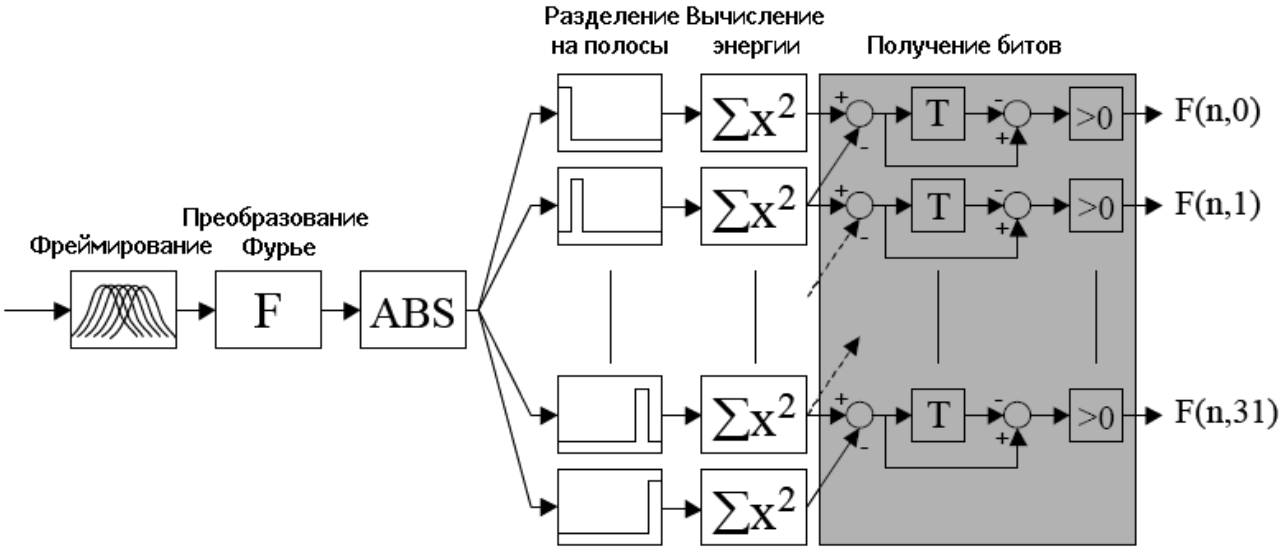

Рисунок 1. Обзор схемы получения отпечатка.

Наиболее важные воспринимаемые характеристики звука живут в частотной области. Поэтому спектральное представление вычисляется путём выполнения преобразования Фурье на каждом кадре. Из-за чувствительности преобразования Фурье к фазе для разных границ кадра и того факта, что человеческая слуховая система (Human Auditory System, HAS) является

относительно нечувствительной к фазе, сохраняется только абсолютная величина спектра, то есть спектральная плотность мощности.

Для того чтобы получить 32-х разрядное значение суб-отпечатка для каждого кадра, выбраны 33 неперекрывающихся диапазона частот. Эти полосы лежат в диапазоне от 300 Гц до 2000 Гц (наиболее важные области спектра для HAS) и имеют логарифмический интервал. Логарифмической интервал выбран потому, что известно, что HAS работает примерно на логарифмических полосах (так называемая шкала ***Барка***, Bark scale). Экспериментально было проверено, что знак разности энергий (одновременно по осям времени и частоты) является свойством, которое очень устойчиво ко многим видам обработки. Если мы обозначим энергию диапазона ***m*** кадра ***n*** как ***E(n,m)*** и ***m***-ый бит суб-отпечатка фрейма ***n*** как ***F(n,m)***, то биты суб-отпечатка формально определяются как (смотрите также серый блок на Рисунке 1, где T - элемент задержки):

$$F(n,m) = \begin{cases} 1, \text{если } E(n,m) - E(n,m+1) - \big(E(n-1,m) - E(n-1,m+1)\big) > 0 \\ 1, \text{если } E(n,m) - E(n,m+1) - \big(E(n-1,m) - E(n-1,m+1)\big) \leq 0 \end{cases} \quad (1)$$

Рисунок 2 показывает пример из 256 последовательных 32-х разрядных суб-отпечатков (то есть блок отпечатков), полученный с помощью вышеописанной схемы из небольшого фрагмента из ***"O Fortuna"*** Carl Orff. Бит '1' соответствует белому пикселю, а бит '0' - чёрному пикселю. Рисунки 2a и 2b показывают блок отпечатков из оригинального компакт-диска и сжатой в MP3 (32Kbps) версии того же отрывка, соответственно. В идеале эти две картинки должны быть идентичны, но из-за сжатия некоторые из битов получены неправильно. Эти ошибочные биты, которые используются в качестве меры сходства для нашей схемы отпечатков, показаны чёрным на Рисунке 2c.

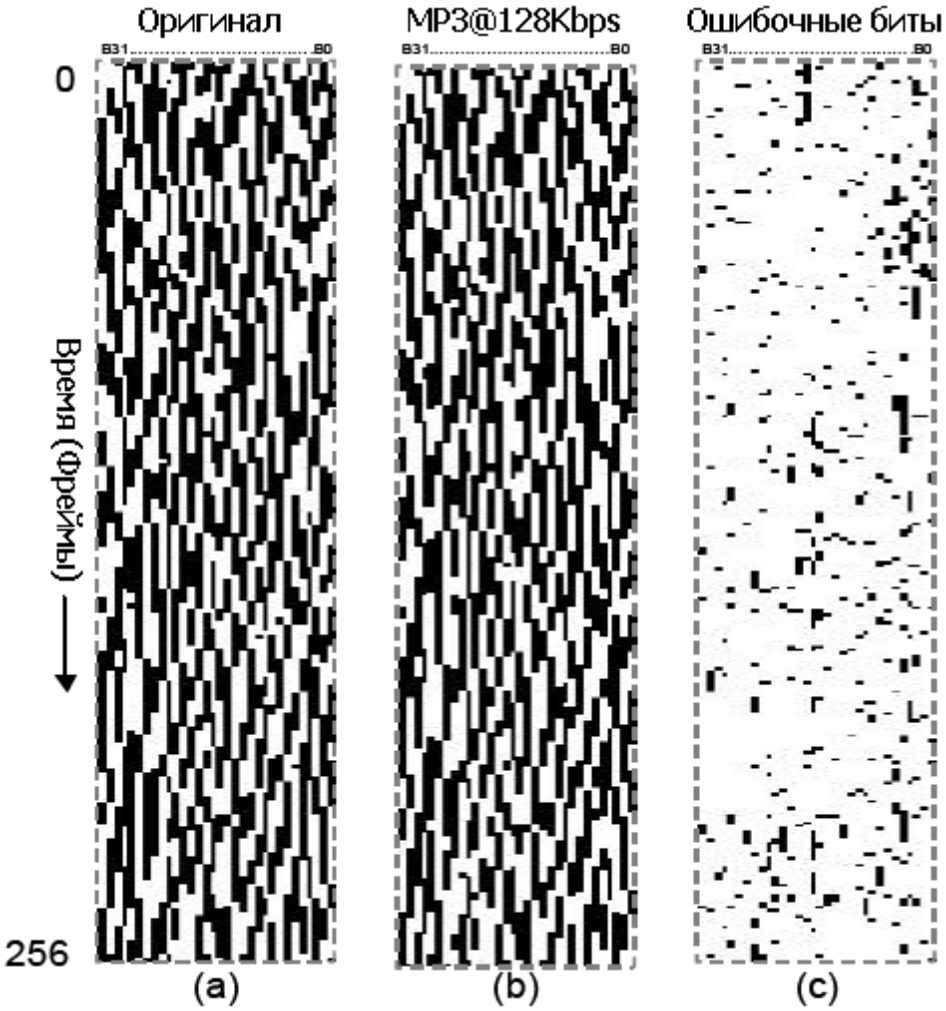

Рисунок 2. (a) Блок отпечатка оригинального музыкального клипа, (b) блок отпечатка скомпрессированной версии, (c) различие между a и b, показывающее ошибочные биты чёрным (BER=0.078).

Вычислительные ресурсы, необходимые для предложенного алгоритма являются ограниченными. Так как алгоритм учитывает только частоты ниже 2 кГц, получаемый звук сначала ресэмплируется до более низкой частоты для получения монофонического аудио потока с частотой дискретизации 5 кГц. Суб-отпечатки разработаны таким образом, что они являются стойкими к искажениям сигнала. Поэтому для понижающего частоту дискретизации ресэмплирования могут быть использованы очень простые фильтры, не приводящие к какому-либо снижению производительности. В настоящее время используются КИХ фильтры 16-го порядка. Наиболее вычислительно ресурсоёмкой операцией является преобразование Фурье каждого аудио кадра. В звуковом сигнале с пониженной частотой дискретизации фрейм имеет длину в 2048 сэмплов. Если преобразование Фурье реализовано в виде БПФ для реальных числе с фиксированной точкой, алгоритм получения отпечатков показывает эффективную работу на портативных устройствах, таких как КПК или мобильный телефон.

4.3 Анализ числа ложных срабатываний

Два 3-х секундных звуковых сигнала объявляются похожими, если расстояние Хэмминга (то есть количество ошибочных битов) между двумя полученными от них блоками отпечатков ниже определённого порога ***T***. Это пороговое значение ***T*** непосредственно определяет процент

ложных срабатываний P_f, то есть как часто звуковые сигналы объявляются похожими: чем меньше T, тем будет меньше вероятность P_f. С другой стороны, малое значение T будет негативно влиять на ложноотрицательную вероятностей P_n, то есть вероятность того, что два сигнала "равны ", но не идентифицированы как таковые.

Для анализа выбора этого порога T, мы считаем, что процесс получения отпечатков даёт случайные i.i.d (независимо и одинаково распределённые, independent and identically distributed) биты. Число ошибочных битов будет иметь биномиальное распределение (n,p), где n равно числу полученных битов, а $p (= 0.5)$ является вероятностью получения бита '0' или '1'. Так как $n (= 8192 = 32 * 256)$ велико в нашем приложении, биномиальное распределение можно аппроксимировать нормальным распределением со средним $\mu = np$ и стандартным отклонением $\sigma = \sqrt{(np(1-p))}$. Имея блок отпечатков F_1, вероятность того, что случайно выбранный блок отпечатков F_2 имеет по отношению к F1 n ошибок меньшее, чем $T = \alpha$, определяется по формуле:

$$P_f(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{(1-2\alpha)\sqrt{n}}^{\infty} e^{-x^2/2} dx = \frac{1}{2} erfc\left(\frac{(1-2\alpha)}{\sqrt{2}}\sqrt{n}\right) \quad (2)$$

где α обозначает частоту появления ошибочных битов (Bit Error Rate, BER).

Однако, на практике суб-отпечатки имеют высокую корреляцию по оси времени. Эта корреляция связана не только со свойственной звуку корреляции во времени, но и большим перекрытием кадров, используемым при получении отпечатков. Более высокая корреляция означает большее стандартное отклонение, как показывает следующее рассуждение.

Пусть есть симметричный бинарный источник с алфавитом {-1,1}, такой, что вероятность того, что символ x_i и символ x_{i+1} равны, является равной q. Тогда можно легко показать, что

$$E[x_i x_{i+k}] = a^{|k|} \quad (3)$$

где $a= 2*q-1$. Если источник Z является исключающим ИЛИ двух таких последовательностей X и Y, то Z является симметричной и

$$E[z_i z_{i+k}] = a^{2|k|} \quad (4)$$

Для больших N стандартное отклонение среднего $\overline{Z_N}$ за N последовательных сэмплов Z может приближённо описываться нормальным распределением со средним значением 0 и стандартным отклонением, равным

$$\sqrt{\frac{1+a^2}{N(1-a^2)}} \quad (5)$$

Переходя обратно к вышеописанному случаю битовых отпечатков, коэффициент корреляции между последовательными битами отпечатков предполагает увеличение стандартного

отклонения для BER на коэффициент

$$\sqrt{\frac{1+a^2}{1-a^2}} \quad (6)$$

Для определения распределения BER с помощью реальных блоков отпечатков была создана база данных отпечатков из 10,000 песен. После этого был определён BER для 100,000 случайно выбранных пар блоков отпечатков. Измеренное стандартное отклонение полученного в результате распределения BER было 0.0148, примерно в 3 раза выше, чем 0.0055, что можно ожидать от случайных независимо распределённых битов.

Рисунок 3 показывает логарифм функции плотности вероятности (Probability Density Function, PDF) измеренного распределения BER и нормальное распределение со средним 0.5 и стандартным отклонением 0.0148. PDF от BER является хорошо приближенным к нормальному распределению. Для BER-ов ниже 0.45 наблюдаются некоторые выбросы из-за недостаточной статистики. Чтобы учесть большее стандартное отклонение распределения BER, в Формулу (2) внесены изменения путём включения множителя 3.

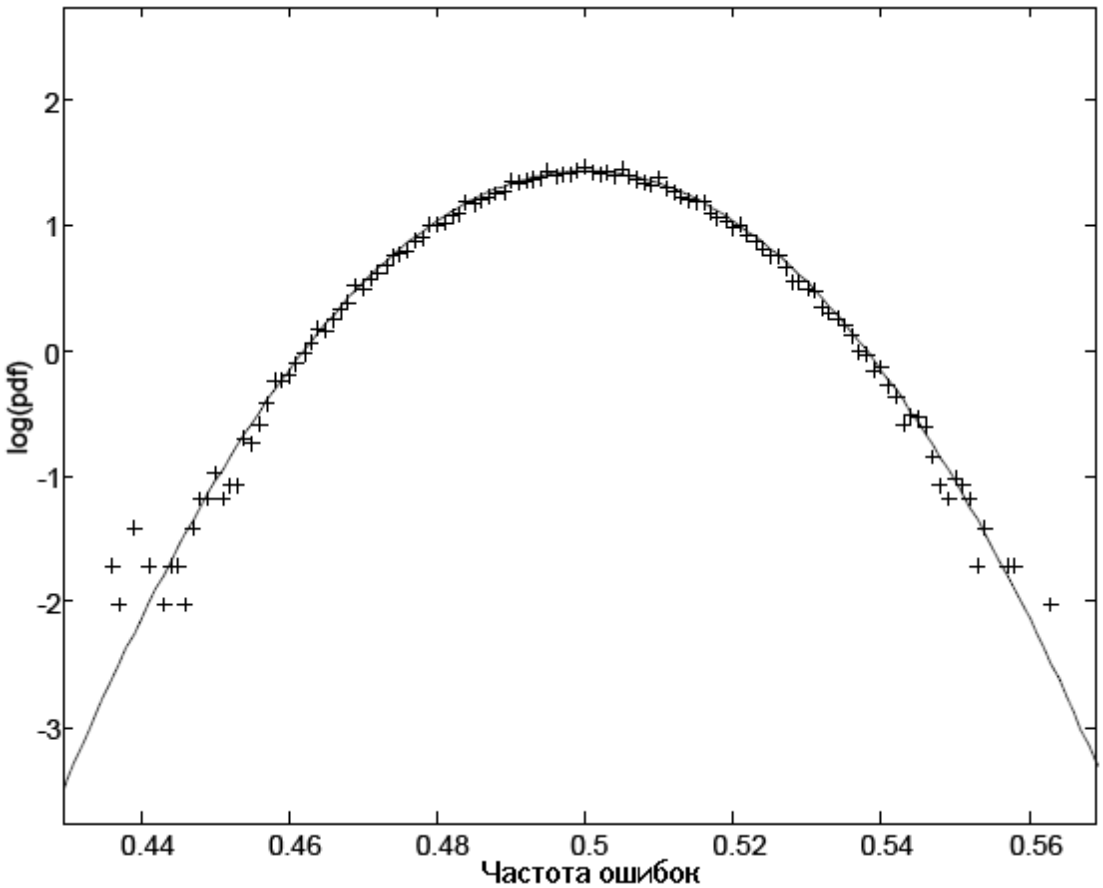

Рисунок 3. Сравнение функции плотности вероятности BER, изображённой знаком '+' и нормальное распределение.

Порог для BER, использованный в ходе экспериментов, был $\alpha = 0.35$. Это означает, что из 8192 бит должно быть менее 2867 ошибочных битов, чтобы решить, что блоки отпечатков

происходят из одной и той же песни. Используя формулу (7) мы приходим к очень низкому проценту ложных срабатываний **erfc**(6.4)/2=3.6*10^{-20}.

4.4 Экспериментальные результаты надёжности

В этом подразделе мы покажем экспериментальную надёжность предлагаемой схемы получения отпечатков звука. То есть мы попытаемся ответить на вопрос, действительно ли BER между блоком отпечатков оригинальной и искажённой версией аудио клипа остаётся ниже порога $\alpha(= 0.35)$.

Мы выбрали четыре коротких отрывка звука (стерео, 44.1 кГц, 16 бит) из песен, которые принадлежат к различным музыкальным жанрам: "O Fortuna" Carl Orff, "Success has made a failure of our home" Sinead o'Connor, "Say what you want" Texas и "A whole lot of Rosie" AC/DC. Все отрывки были подвергнуты следующим искажениям сигнала:

- **MP3 кодирование/декодирование** со скоростью 128 Кбит/с и 32 кбит/с.
- **Кодирование/декодирование в Real Media** при 20 Кбит/с.
- **GSM кодирование на полной скорости** на канале без ошибок и канале с интерференцией несущей (C/I) на уровне 4 дБ (сравнимо с приёмом GSM в туннеле).
- **Всепропускающий фильтр** с использованием системной функции: ***H(z)=(0.81z²-1.64z+1)/(z²-1.64z+0.81)***.
- **Амплитудная компрессия** со следующими степенями сжатия: 8.94:1 для |A| ≥ -28.6 дБ; 1.73:1 для -46.4 dB < |A| < -28.6 дБ; 1:1.61 для |A| ≤ -46.4 dB.
- **Эквализация**, типичный 10-ти полосный эквалайзер со следующими настройками:

Частота (Гц)	31	62	125	250	500	1k	2k	4k	8k	16k
Уровень (дБ)	-3	+3	-3	+3	-3	+3	-3	+3	-3	+3

- **Полосовой фильтр** с использованием фильтра Баттерворта второго порядка с частотами среза 100 Гц и 6000 Гц.
- **Изменение времени звучания** +4% и -4%. Изменялся только темп, тон не изменялся.
- **Линейное изменение скорости** +1%, -1%, +4% и -4%. Изменялись и тон, и темп.
- **Добавленный шум** с помощью равномерного белого шума с максимальной амплитудой из 512-ти шагов квантования.
- **Изменение частоты дискретизации**, последовательное преобразование вниз и вверх на частоте 22.05 кГц и 44.10 кГц, соответственно.
- **Цифроаналоговое и аналогоцифровое преобразование** с использованием коммерческого аналогового магнитофона.

После этого между блоками отпечатков оригинальной версии и всеми искажёнными

версиями для каждого аудио клипа были определены BER-ы. Полученные в результате BER-ы приведены в Таблице 1. Почти все результирующие частоты ошибочных битов значительно ниже порога в 0.35, даже для GSM кодирования (напомним, что кодек GSM оптимизирован для передачи речи, а не для любого звука.). Единственным искажением, приводящим к BER выше порога, является большое линейное изменение скорости. Изменение линейной скорости более, чем на +2.5% или -2.5% процента, обычно приводит к частоте ошибочных битов большей, чем 0.35. Это связано с неточным выравниванием кадров (временные смещения) и масштабированием спектра (смещениями частот). Соответствующее предварительное масштабирование (например, путём перебора) может решить эту проблему.

Таблица 1. BER для разного вида искажений сигнала.

Обработка	Orff	Sinead	Texas	ACDC
MP3@128Kbps	0.078	0.085	0.081	0.084
MP3@32Kbps	0.174	0.106	0.096	0.133
Real@20Kbps	0.161	0.138	0.159	0.210
GSM	0.160	0.144	0.168	0.181
GSM C/I = 4dB	0.286	0.247	0.316	0.324
Всепропускающий фильтр	0.019	0.015	0.018	0.027
Амплитудная компрессия	0.052	0.070	0.113	0.073
Эквализация	0.048	0.045	0.066	0.062
Добавление эха	0.157	0.148	0.139	0.145
Полосовой фильтр	0.028	0.025	0.024	0.038
Изменение времени +4%	0.202	0.183	0.200	0.206
Изменение времени –4%	0.207	0.174	0.190	0.203
Линейная скорость +1%	0.172	0.102	0.132	0.238
Линейная скорость -1%	0.243	0.142	0.260	0.196
Линейная скорость +4%	0.438	0.467	0.355	0.472
Линейная скорость -4%	0.464	0.438	0.470	0.431
Добавление шума	0.009	0.011	0.011	0.036
Изменение частоты дискретизации	0.000	0.000	0.000	0.000
Ц/А и А/Ц преобразование	0.088	0.061	0.111	0.076

5. ПОИСК В БАЗЕ ДАННЫХ

5.1 Алгоритм поиска

Поиск полученных отпечатков в базе данных отпечатков является нетривиальной задачей. Вместо поиска немногих точных отпечатков (что просто!), должны быть найдены наиболее похожие отпечатки. Проиллюстрируем это некоторыми числами на основе предложенной схемы отпечатков. Рассмотрим базу данных отпечатков умеренного размера, содержащую 10,000 песен со средней длиной в 5 минут. Это соответствует примерно 250 миллионам суб-отпечатков. Для выявления блока отпечатков, полученного от неизвестного куска звука, необходимо найти в базе данных наиболее похожий блок отпечатков. Другими словами, необходимо найти такую позицию в 250 миллионах суб-отпечатков, где число ошибочных битов минимально. Это, конечно, возможно простым прямым поиском. Однако это потребует 250 миллионов сравнений блоков отпечатков. При использовании современного ПК может быть достигнута скорость примерно в 200,000 сравнений блоков отпечатков в секунду. Поэтому общее время поиска для нашего примера будет порядка 20 минут! Это показывает, что для

практического применения поиск с помощью грубой силы не является жизнеспособным решением.

Мы предлагаем использовать более эффективный алгоритм поиска. Вместо вычисления BER для всех возможных позиций в базе данных, как в методе прямого поиска, посчитаем его только для нескольких позиций-кандидатов. Эти кандидаты находятся с очень высокой вероятностью на наиболее подходящей позиции в базе данных.

В простой версии улучшенного алгоритма поиска позиции-кандидаты генерируются на основе предположения, что очень вероятно, что по крайней мере один суб-отпечаток имеет точное соответствие в оптимальной позиции в базе данных [3][5]. Если это предположение справедливо, должны быть проверены только те позиции, где один из 256 суб-отпечатков блока отпечатков запроса имеет точное соответствие. Чтобы убедиться в справедливости предположения, график на Рисунке 4 показывает количество ошибочных битов на суб-отпечаток для отпечатков, изображённых на Рисунке 2. Это показывает, что действительно существует суб-отпечаток, который не содержит никаких ошибок. На самом деле не содержат ошибок 17 из 256 суб-отпечатков. Если предположить, что это "оригинальный" отпечаток Рисунка 2a, действительно загруженный в базу данных, эта позиция будет среди выбранных позиций-кандидатов для "отпечатка MP3@128Kbps" Рисунка 2b.

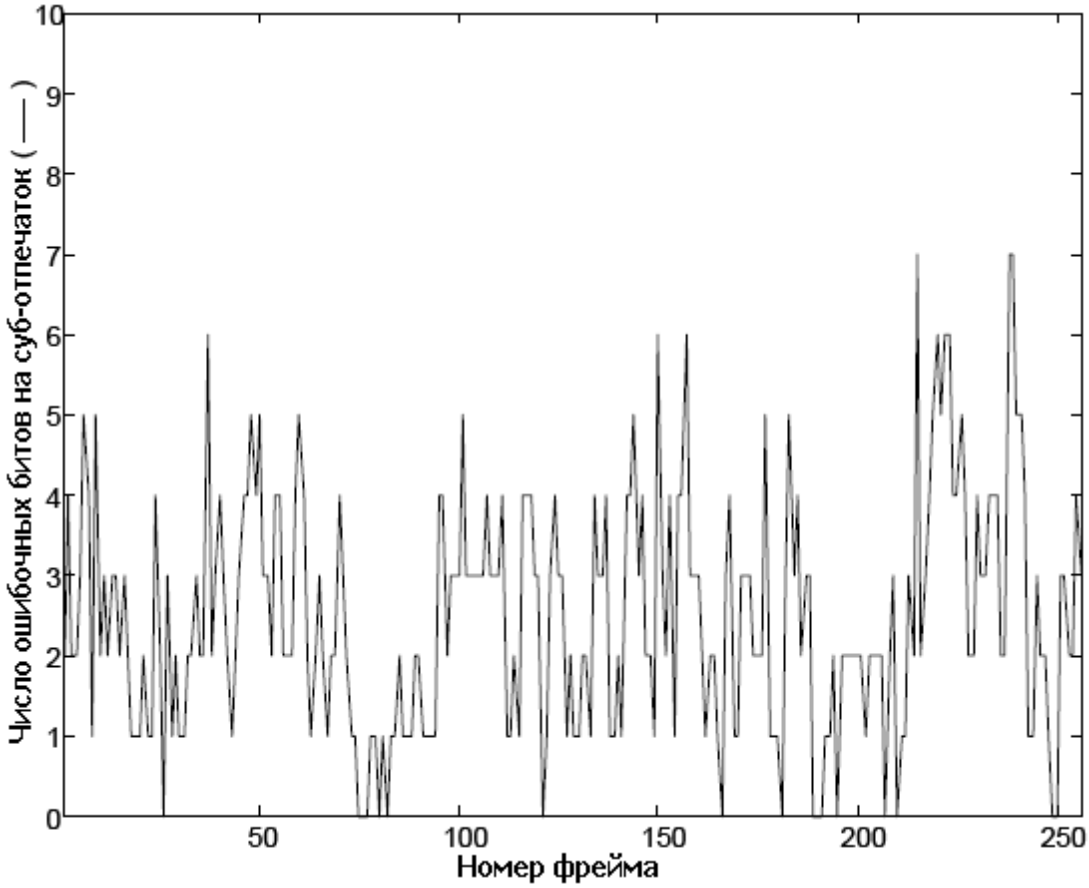

Рисунок 4. Число ошибочных битов на суб-отпечаток для “версии MP3@128Kbps” фрагмента ‘O Fortuna’ Carl Orff.

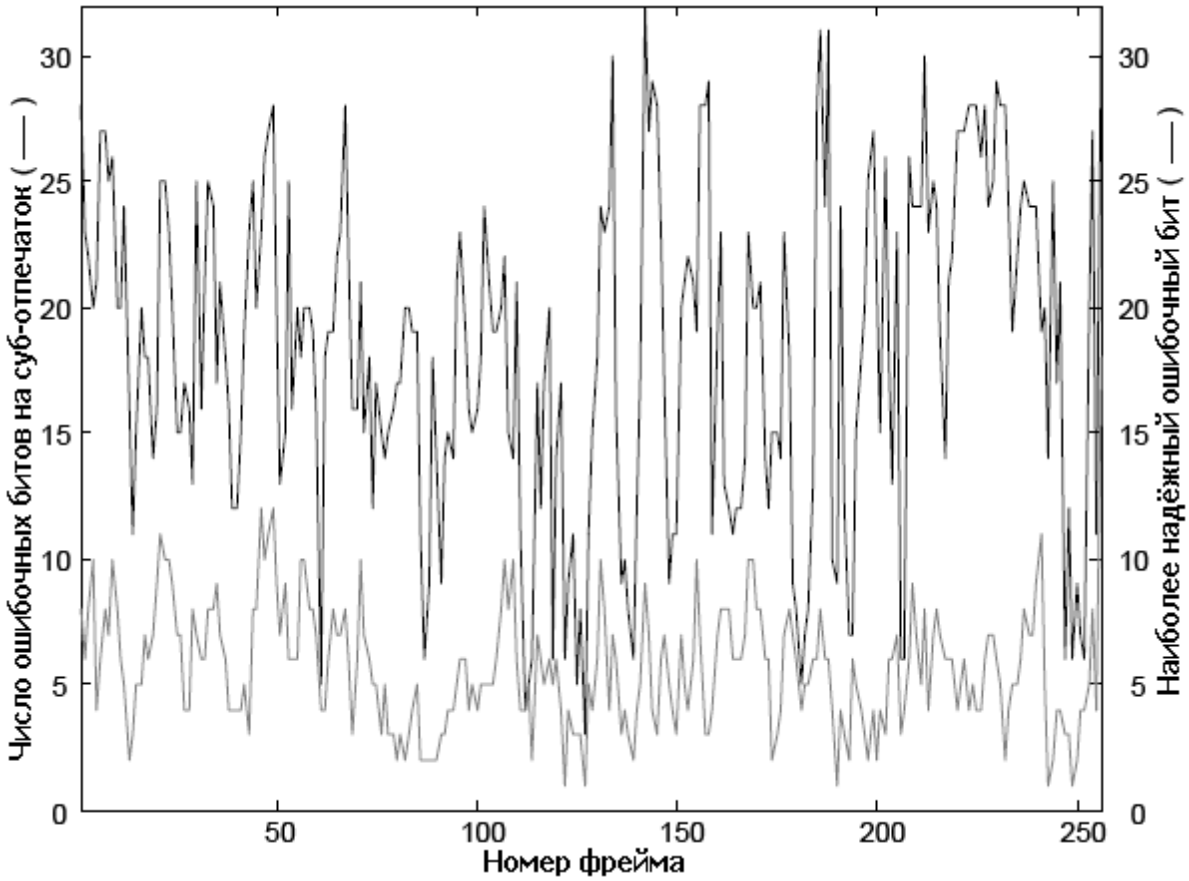

Рисунок 5. Число ошибочных битов на суб-отпечаток (серая линия) и надёжность наиболее надёжного ошибочного бита (чёрная линия) для "версии MP3@32Kbps" 'O Fortuna' Carl Orff.

Позиции в базе данных, где находится определённый 32-х разрядный отпечаток, извлекаются с помощью архитектуры базы данных Рисунка 6. База данных отпечатков содержит таблицу поиска (lookup table, LUT) со всеми возможными 32-х разрядными суб-отпечатками в качестве записи. Каждая запись указывает на список с указателями на позиции в реальных списках отпечатков, в которых расположены соответствующие 32-х разрядные суб-отпечатки. В практических системах с ограниченным объёмом памяти (например, ПК с 32-х разрядным процессором Intel имеет предел памяти в 4 ГБ) таблица поиска, содержащая 2^{32} записей, часто невозможна, или непрактична, или и то и другое. Кроме того, таблица поиска будет заполнена редко, потому что в памяти может находиться только ограниченное число песен. Таким образом, на практике вместо таблицы поиска используется хэш-таблица [19].

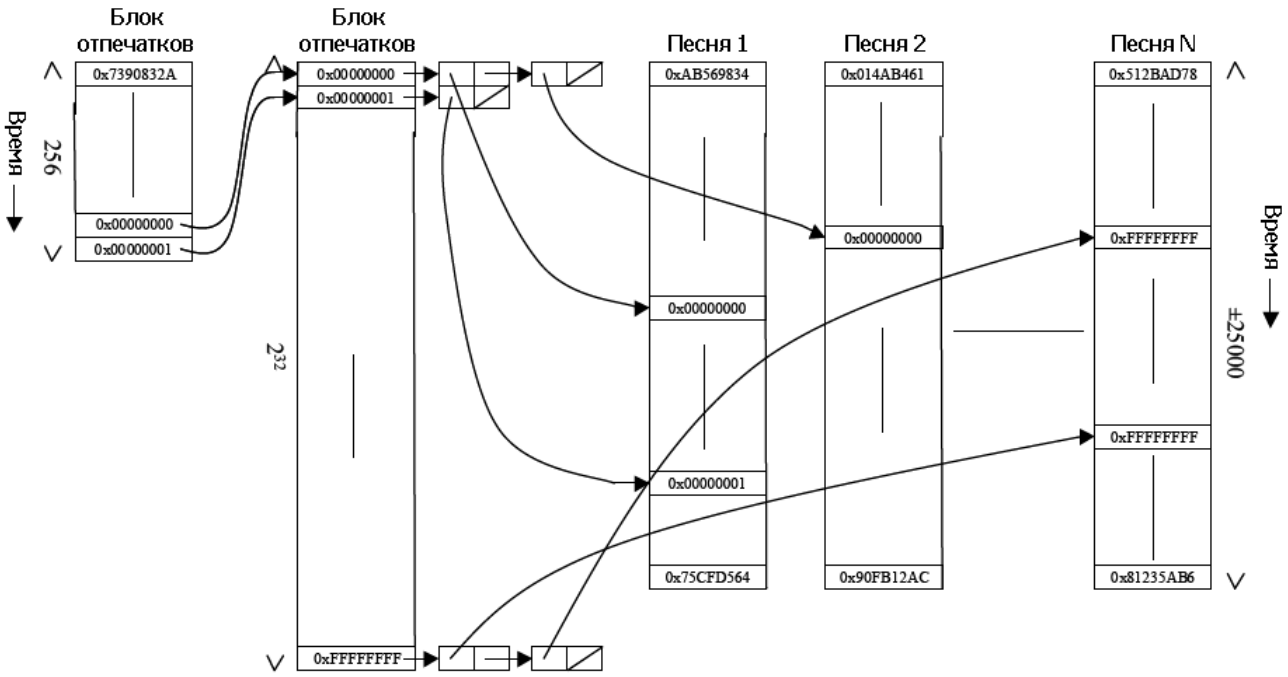

Рисунок 6. Строение базы данных отпечатков.

Давайте снова посчитаем среднее количество сравнений блоков отпечатков на идентификацию для базы данных из 10,000 песен. Поскольку база данных содержит около 250 миллионов суб-отпечатков, среднее число позиций в списке будет 0.058 (=$250*10^6/2^{32}$). Если предположить, что все возможные суб-отпечатки равновероятны, средним числом сравнений отпечатков на идентификацию является всего 15 (=0.058*256). Однако, на практике наблюдается, что из-за неравномерного распределения суб-отпечатков количество сравнений отпечатков увеличивается примерно в 20 раз. В среднем необходимо 300 сравнений, что даёт на современных ПК среднее время поиска 1.5 мс. Таким образом, таблица поиска может быть реализована, но это не влияет на время поиска. С дорогой таблицей поиска предложенный алгоритм поиска примерно в 800,000 раз быстрее, чем подход прямого поиска.

Наблюдательный читатель может спросить: "Но что, если ваши предположение, что один из суб-отпечатков не содержит ошибок, не выполняется"? Ответом является то, что данное предположение почти всегда выполняется для звуковых сигналов с "умеренными" искажениями звукового сигнала (смотрите также Раздел 5.2 19). Однако, для сильно искажённых сигналов предположение действительно не всегда справедливо. Пример графика числа ошибочных битов на суб-отпечаток для блока отпечатков, который не содержит ни одного безошибочного суб-отпечатка, показан на Рисунке 5. Однако, есть суб-отпечатки, которые содержат только одну ошибку. Таким образом, вместо проверки только тех позиций в базе данных, где встречается один из 256 суб-отпечатков, мы также можем проверить все позиции, в которых встречаются суб-отпечатки, находящиеся на расстоянии Хэмминга, равном единице (то есть переключен один бит), для всех 256 суб-отпечатков. Это приведёт к числу сравнений отпечатков в 33 раза большему, которое всё ещё приемлемо. Однако, если мы хотим справиться с ситуациями, в которых, например, минимальное количество ошибочных битов на суб-отпечаток равно трём (это может произойти в приложении для мобильного телефона), число сравнений отпечатков будет увеличено в 5489 раз, что приведёт к неприемлемым затратам времени для поиска. Обратите внимание, что наблюдаемый фактор неоднородности 20 снижается с увеличением числа переключенных битов. Если, например, для переключения используются все 32 бита суб-отпечатков, мы в конце концов снова придём к прямому перебору, получив коэффициент умножения 1.

Так как случайное переключение битов для создания большего числа позиций-кандидатов очень быстро приводит к неприемлемому времени поиска, мы предлагаем использовать другой подход, который использует программное декодирование информации. То есть мы предлагаем оценить и использовать вероятность того, что бит отпечатка получен правильно.

Суб-отпечатки получаются путём сравнения и оценки порога разности энергий (смотрите блок получения битов на Рисунке 1). Если разность энергий очень близка к порогу, достаточно вероятно, что бит был получен неправильно (ненадёжный бит). С другой стороны, если разность энергий гораздо больше порога, вероятность неправильности бита мала (надёжный бит). Получив информацию о надёжности для каждого бита суб-отпечатка, можно расширить данный отпечаток до списка вероятных суб-отпечатков. Если считать, что один из наиболее вероятных суб-отпечатков имеет точное соответствие в оптимальной позицией в базе данных, такой блок отпечатков может быть идентифицирован, как раньше. Битам назначается рейтинг надёжности от 1 до 32, где 1 обозначает самый ненадёжный, а 32 - самый надёжный бит.

Это приводит к простому способу создания списка наиболее вероятных суб-отпечатков переключением только самых ненадёжных битов. Точнее, список состоит из всех суб-отпечатков, которые имеют N фиксированных самых надёжных бит, а все другие изменяются. Если достоверность информации является безупречной, можно ожидать, что в случае, когда суб-отпечаток имеет 3 ошибочных бита, биты с надёжностью 1, 2 и 3 являются ошибочными. Если это так, блоки отпечатков, где минимальное количество ошибочных битов на суб-отпечаток равно 3, могут быть идентифицированы созданием позиций-кандидатов с помощью только 8 ($=2^3$) суб-отпечатков на суб-отпечаток. По сравнению с коэффициентом 5489, полученном при использовании всех суб-отпечатков с расстоянием Хэмминга, равным 3 для генерации позиций-кандидатов, это - улучшение с коэффициентом около 686.

На практике достоверность информации не является безупречной (например, бывает так, что бит с низкой надёжностью получен правильно и наоборот), и, следовательно, улучшение менее впечатляющее, но всё же значительное. Например, это можно увидеть на Рисунке 5. Минимальное количество ошибочных битов на суб-отпечаток равно единице. Как уже упоминалось ранее, такой блок отпечатков может быть идентифицирован созданием в 33 раза большего числа позиций-кандидатов. Рисунок 5 также содержит график надёжности для самого надёжного бита, который получен ошибочно. Надёжность получена из версии MP3@32Kbps с помощью предложенного метода. Мы видим, что первый суб-отпечаток содержит 8 ошибок. Эти 8 бит не являются самыми ненадёжными битами, поскольку один из ошибочных битов имеет назначенную надёжность 27. Таким образом, ***информация о надёжности*** не всегда ***надёжна***. Однако, если мы рассмотрим суб-отпечаток 130, который имеет только один ошибочный бит, то увидим, что назначенная надёжность ошибочного бита равна 3. Поэтому этот блок отпечатков указывал бы на правильное расположение в базе данных отпечатков при переключении только 3-х наиболее ненадёжных битов. Таким образом, эта песня была бы определена правильно.

Мы закончим этот подраздел вновь ссылаясь на Рисунок 6 и давая пример того, как работает предложенный алгоритм поиска. Последний извлечённый суб-отпечаток из блока отпечатков на Рисунке 6 это 0x00000001. Первый блок отпечатков сравнивается с теми позициями в базе данных, где находится суб-отпечаток 0x00000001. Для суб-отпечатка 0x00000001 LUT указывает только на одну позицию, относящуюся к позиции ***p*** в песне 1. Вычислим теперь BER между 256-ю полученными суб-отпечатками (блоком отпечатков) и значениями суб-отпечатков песни 1 от позиции ***p-255*** до позиции ***p***. Если BER ниже порога в 0.35, высока вероятность того, что полученный блок отпечатков происходит от песни 1.

Однако, если это не так, либо в базе данных нет песни, либо этот суб-отпечаток содержит ошибку. Допустим напоследок и что бит 0 является наименее надёжным битом. Следующим наиболее вероятным кандидатом является суб-отпечаток 0x00000000. По-прежнему ссылаясь на Рисунок 6, суб-отпечаток 0x00000000 имеет две возможные позиции-кандидаты: одна в песне 1, а другая в песне 2. Если блок отпечатков имеет BER ниже порога по отношению к связанному блоку отпечатков в песне 1 или 2, то совпадение будет объявлено соответственно для песни 1 или 2. Если ни одна из двух позиций-кандидатов не даёт BER ниже порога, то либо можно использовать другие возможные суб-отпечатки для генерации большего числа позиций-кандидатов, либо переключиться на один из 254 оставшихся суб-отпечатков для повтора процесса. Если для создания позиций-кандидатов были использованы все 256 суб-отпечатков и их наиболее вероятные суб-отпечатки, и не было найдено соответствия ниже порога, алгоритм решает, что не может опознать песню.

5.2 Экспериментальные результаты

Таблица 2 показывает, как много из сгенерированных позиций-кандидатов указывают на соответствующий блок отпечатков в базе данных для того же набора искажённых сигналов, использованного в экспериментах по надёжности. Мы будем называть это число как число совпадений в базе данных. Число совпадений должно быть равно единице или больше, чтобы идентифицировать блок отпечатков и может быть максимально равно 256 в случае, когда все суб-отпечатки создают верную позицию-кандидат. Первое число в каждой ячейке является числом совпадений в случае, когда для создания позиций-кандидатов использовались только сами суб-отпечатки (то есть программная информация декодирования не используется). Заметим, что большинство блоков отпечатков могут быть идентифицированы, так как произошло одно или более совпадений. Однако, несколько блоков отпечатков, главным образом происходящие из более серьёзно искажённого звука, например, в GSM с C/I равным 4 дБ, не генерируют никаких совпадений. Такую настройку алгоритма поиска можно использовать в таких приложениях, как мониторинг вещания и автоматизированную маркировку MP3, где ожидается лишь незначительное искажение звука.

Второе число в каждой ячейке соответствует числу совпадений с настройкой, которая используется для идентификации сильно искажённого звука, как, например, в приложении для мобильного телефона. По сравнению с предыдущей настройкой для генерации кандидатов дополнительно используются 1024 наиболее вероятных суб-отпечатков для каждого суб-отпечатка. Другими словами, для генерации большего числа позиций-кандидатов переключались 10 наименее надёжных бит каждого суб-отпечатка. В результате число совпадений выше, и могут быть идентифицированы даже "блоки отпечатков GSM C/I = 4 дБ". Большинство блоков отпечатков с линейными изменениями скорости по-прежнему не имеют ни одного совпадения. BER этих блоков уже выше, чем порог, и по этой причине они не могут быть определены, даже если есть совпадения. Кроме того, надо иметь в виду, что при соответствующем предыскажении, как это предложено в Разделе 4.4, такие блоки отпечатков с большим изменением линейной скорости могут быть определены достаточно легко.

Таблица 2. Совпадения в базе данных для разных видов искажения сигнала. Первая цифра показывает совпадения для использования только 256 суб-отпечатков для получения позиций-кандидатов. Второе число указывает показывает совпадения, когда также используются 1024 наиболее вероятных кандидатов для каждого суб-отпечатка.

Обработка	Orff	Sinead	Texas	ACDC
MP3@128Kbps	17, 170	20, 196	23, 182	19, 144
MP3@32Kbps	0, 34	10, 153	13, 148	5, 61
Real@20Kbps	2, 7	7, 110	2, 67	1, 41
GSM	1, 57	2, 95	1, 60	0, 31
GSM C/I = 4dB	0, 3	0, 12	0, 1	0, 3
Всепропускающий фильтр	157, 240	158, 256	146, 256	106, 219
Амплитудная компрессия	55, 191	59, 183	16, 73	44, 146
Эквализация	55, 203	71, 227	34, 172	42, 148
Добавление эха	2, 36	12, 69	15, 69	4, 52
Полосовой фильтр	123, 225	118, 253	117, 255	80, 214
Изменение времени +4%	6, 55	7, 68	16, 70	6, 36
Изменение времени –4%	17, 60	22, 77	23, 62	16, 44
Линейная скорость +1%	3, 29	18, 170	3, 82	1, 16
Линейная скорость -1%	0, 7	5, 88	0, 7	0, 8
Линейная скорость +4%	0, 0	0, 0	0, 0	0, 1
Линейная скорость -4%	0, 0	0, 0	0, 0	0, 0
Добавление шума	190, 256	178, 255	179, 256	114, 225
Изменение частоты дискретизации	255, 256	255, 256	254, 256	254, 256
Ц/А и А/Ц преобразование	15,149	38, 229	13, 114	31,145

6. ВЫВОДЫ

В этой статье мы представили новый подход к дактилоскопии звука. Получение отпечатков основано на получении 32-х разрядного суб-отпечатка каждые 11.6 мс. Суб-отпечатки создаются, глядя на различия энергий по частотной и временной осям. Блок отпечатков, включающий 256 последовательных суб-отпечатков, является основной единицей, которая используется для определения песни.

База данных отпечатков содержит двухфазный алгоритм поиска, основанный на выполнении полных сравнений отпечатков только на предварительно выбранных с помощью поиска суб-отпечатков позициях-кандидатах. Что касается параметров, которые были введены в Разделе 2.2, предлагаемая система может быть обобщена следующим образом:

- **Надёжность**: полученные отпечатки являются очень надёжными. Они могут быть использованы даже для идентификации музыки, записанной и переданной по мобильному телефону.

- **Достоверность**: в Разделе 4.3 мы получили модель для частоты ложных срабатываний, что было подтверждено экспериментально. При установке порога в 0.35 достигается частота ложных срабатываний в $3.6*10^{-20}$.

- **Размер отпечатка**: 32-х разрядный отпечаток извлекается каждые 11.6 мс, что приводит к скорости передачи отпечатков 2.6 кбит/с.

- **Степень детализации**: для идентификации в качестве основной единицы используется блок отпечатков, состоящий из 256-ти суб-отпечатков и соответствующий 3 секундам звука.

- **Скорость поиска и масштабируемость**: на современном ПК при использовании двухфазного алгоритма поиска отпечатков может работать база данных отпечатков, содержащая 20,000 песен и обрабатывающая десятки запросов в секунду.

Дальнейшие исследования будут сосредоточены на других методах выделения признаков и оптимизации алгоритма поиска.

7. ЛИТЕРАТУРА

[1] Cheng Y., “Music Database Retrieval Based on Spectral Similarity”, International Symposium on Music Information Retrieval (ISMIR) 2001, Bloomington, USA, October 2001.

[2] Allamanche E., Herre J., Hellmuth O., Bernhard Frobach B. and Cremer M., “AudioID: Towards Content-Based Identification of Audio Material”, 100th AES Convention, Amsterdam, The Netherlands, May 2001.

[3] Neuschmied H., Mayer H. and Battle E., “Identification of Audio Titles on the Internet”, Proceedings of International Conference on Web Delivering of Music 2001, Florence, Italy, November 2001.

[4] Fragoulis D., Rousopoulos G., Panagopoulos T., Alexiou C. and Papaodysseus C., “On the Automated Recognition of Seriously Distorted Musical Recordings”, IEEE Transactions on Signal Processing, vol.49, no.4, p.898-908, April 2001.

[5] Haitsma J., Kalker T. and Oostveen J., “Robust Audio Hashing for Content Identification, Content Based Multimedia Indexing 2001, Brescia, Italy, September 2001.

[6] Oostveen J., Kalker T. and Haitsma J., “Feature Extraction and a Database Strategy for Video Fingerprinting”, 5th International Conference on Visual Information Systems, Taipei, Taiwan, March 2002. Published in: "Recent advances in Visual Information Systems", LNCS 2314, Springer, Berlin. pp. 117-128.

[7] Auditude website <http://www.auditude.com>

[8] Relatable website <http://www.relatable.com>

[9] Audible Magic website <http://www.audiblemagic.com>

[10] Shazam website <http://www.shazamentertainment.com>

[11] Moodlogic website <http://www.moodlogic.com>

[12] Yacast website <http://www.yacast.com>

[13] Philips (audio fingerprinting) website <http://www.research.philips.com/InformationCenter/Global/FArticleSummary.asp?lNodeId=927&channel=927&channelId=N927A2568>

[14] RIAA-IFPI. Request for information on audio fingerprinting technologies, 2001. <http://www.ifpi.org/site-content/press/20010615.html>

[15] Napster website <http://www.napster.com>

[16] ID3Man website <http://www.id3man.com>

[17] Winamp website <http://www.winamp.com>

[18] Logan B., “Mel Frequency Cepstral Coefficients for Music Modeling”, Proceeding of the International Symposium on Music Information Retrieval (ISMIR) 2000, Plymouth, USA, October 2000.

[19] Cormen T.H., Leiserson C.H., Rivest R.L., “Introduction To Algorithms”, MIT Press, ISBN 0-262-53091-0, 1998