ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Нижегородский государственный университет им. Н.И. Лобачевского

**Факультет вычислительной математики и кибернетики**

**Кафедра математической логики и высшей алгебры**

**РЕФЕРАТ**

для сдачи кандидатского экзамена по английскому языку

(Перевод)

Выполнил:

Аспирант 2-го года обучения

Мартынов Игорь Михайлович

Специальность 01.01.09 – Дискретная математика и математическая кибернетика

Н.Новгород

2015

май

Реферат выполнен по журналу:

Digital Technical Journal Vol. 5 No. 2, Spring 1993. Davis Yen Pan – Digital Audio Compression

# Сжатие цифрового звука

Дэвис Йен Пан (Davis Yen Pan)

## Аннотация

По сравнению с большинством типов цифровых данных, за исключением цифрового видео, объём данных, связанный с несжатым цифровым звуком, значителен. Сжатие цифрового звука позволяет более эффективно хранить и передавать аудио данные. Множество видов техник сжатия звука предлагают широкий диапазон сложности кодера и декодера, качества сжатого звука и коэффициентов сжатия. µ-преобразование и ADPCM-кодер – простые подходы с низкой сложностью, низким коэффициентом сжатия и алгоритмами среднего качества звука. Стандарт MPEG – алгоритм высокой сложности, высоким коэффициентом сжатия и высоким качеством звука. Эти техники применимы к звуковым сигналам общего вида, и не приспособлены специальным образом для сжатия речи.

## Введение

Сжатие цифрового звука позволяет рационально хранить и передавать звуковые данные. Разнообразные методы сжатия звука предлагают различные уровни сложности, качества сжатого звука и величины сжатия.

Эта статья – исследование методов, используемых для сжатия цифровых звуковых сигналов. Её цель – предоставить полезную информацию для читателей любого уровня опыта в обработке цифрового звука. Статья начинается с обзора начального процесса оцифровки звука. Два следующих раздела представляют детальное описание двух относительно простых подходов к сжатию звука: µ-закон и адаптивной дифференциальной импульсно-кодовой модуляции (adaptive differential pulse code modulation, ADPCM). В следующем разделе статьи рассматривается третий, гораздо более изощрённый, алгоритм сжатия звука от Motion Picture Experts Group (MPEG). Вопросы, охваченные в этом разделе, довольно сложны, и рассчитаны на читателя, знакомого с цифровой обработкой сигналов. Статья завершается обсуждением программных реализаций реального времени.

## Цифровые аудио данные

Цифровое представление звуковых данных предлагает много преимуществ: высокую устойчивость к шуму, стабильность и воспроизводимость. Звук в цифровом виде также позволяет эффективно реализовывать многие процедуры обработки звука (например, смешивание, фильтрацию и стабилизацию) с помощью цифровой ЭВМ.

Преобразование из аналоговой в цифровую область начинается с замеров входного звука через регулярные, дискретные промежутки времени и отображения замеренных значений на конечное число равномерно распределённых уровней. Цифровые звуковые данные состоят из последовательности двоичных значений, представляющих число уровней квантизации для каждого звукового сэмпла. Способ представления каждого сэмпла с помощью независимого машинного слова называется импульсно-кодовой модуляцией (pulse code modulation, PCM). Схема 1 демонстрирует процесс оцифровки звука.

Согласно теории Найквиста (*прим: имеется в виду теорема Котельникова; в англоязычной литературе – теорема Найквиста-Шеннона*), сэмплированный сигнал может верно представлять сигналы частотой вплоть до половины частоты сэпмлирования. Типичные частоты сэмплирования лежат в диапазоне от 8 килогерц (кГц) до 48 кГц. Частота в 8 кГц покрывает диапазон частот вплоть до 4 кГц, и поэтому покрывает большую часть частот, производимых человеческим голосом. Частота в 48 кГц покрывает диапазон частот вплоть до 24 кГц, что более чем достаточно для покрытия всего диапазона слышимых частот, который, для людей, обычно простирается только до 20 кГц. На практике, диапазон частот несколько меньше, чем половина частоты сэмплирования из-за практических ограничений системы.

Число уровней квантизации обычно является степенью двойки, чтобы полностью использовать фиксированное число бит на звуковой сэмпл для представления квантизованных уровней. С равномерным распределением шагов квантизации, каждый дополнительный бит может увеличить отношение сигнал-шум, или динамический диапазон, квантизованной амплитуды примерно на 6 децибел (6 дБ). Типичное число бит на сэмпл, используемое для цифрового звука, лежит в диапазоне от 8 до 16. Таким образом, возможности динамического диапазона лежит от 48 до 96 дБ соответственно. Если взглянуть на эти диапазоны со стороны, если 0 дБ соответствует уровню звукового давления самого слабого слышимого шума, то 25 дБ – минимальный уровень шума в обычной звукозаписывающей студии, 25 дБ – уровень шума внутри тихого дома, а 120 дБ – самый громкий звук до появления ощущения неудобства. С точки зрения восприятия звука, 1 дБ – минимальное слышимое изменение звукового давления в идеальных условиях, и удвоение звукового давления соответствует одному шагу восприятия громкости.

По сравнению с большинством типов цифровых данных (включая цифровое видео), объём данных, связанный с несжатым цифровым звуком, существенен. Например, звуковые данные на компакт-диске (2 звуковых канала, сэмплированные с частотой 44.1 кГц с 16 битами на сэмпл) требует потока данных примерно в 1.4 мегабит в секунду. Ясна необходимость какой-то формы сжатия, чтобы эффективно хранить и передавать эти данные.

Многие виды техник сжатия звука отличаются компромиссом между сложностью кодера и декодера, качеством сжатого звука и величиной сжатия. Техники, представленные в следующих разделах этой статьи описывают полный диапазон, от µ-закона – алгоритма низкой сложности, низкой степени сжатия и средним качеством, до MPEG – алгоритма высокой сложности, высокой степени сжатия и высокого качества звука. Эти техники применимы к звуковым сигналам общего вида, и не заточены специально для речевых сигналов. Эта статья не описывает алгоритмы сжатия звука, разработанные специально для речевых сигналов. Эти алгоритмы обычно основаны на моделировании речевого аппарата, и не работают хорошо для неречевых звуковых сигналов. Федеральные стандарты 1015 LPC (linear predictive coding, кодирование с линейным предиктором) и 1016 CELP (coded excited linear prediction, линейное предсказание с мультикодовым управлением) попадают в эту категорию.

## Сжатие звука через µ-преобразование

µ-закон – базовый метод сжатия звука, определённый рекомендацией G.711 Международного Консультативного Комитета по Телефонии и Телеграфии (Comité Consultatif International Téléphonique et Télégraphique, CCITT). Преобразование в сущности является логарифмическим по характеру, и позволяет с помощью 8 бит на сэмпл покрывать динамический диапазон, эквивалентный 14 битам линейно квантизованных значений. Это преобразование предоставляет коэффициент сжатия, равный (числу бит на исходный сэмпл)/8 к 1. В отличие от линейной квантизации, логарифмической распределение шагов представляет низкоамплитудные звуковые сэмплы с большей точностью, чем высокоамплитудные значения. Таким образом, отношение сигнал-шум преобразованного выхода более равномерно на диапазоне амплитуд входного сигнала. µ-закон есть:

где m = 255, и x – значение входного сигнала, нормализованного, чтобы иметь максимальное значение 1. Рекомендация G.711 CCITT также определяет подобный A-закон. µ-закон имеен широкое применение в Северной Америке и Японии для сэмплированного на 8 кГц голосового цифрового телефонного сервиса ISDN (Integrated Services Digital Network), а A-закон применяется в других местах для ISDN-телефонии.

Адаптивная дифференциальная импульсно-кодовая модуляция (Adaptive Differential Pulse Code Modulation, ADPCM)

Схема 2 демонстрирует упрощённую блок-схему кодера адаптивной дифференциальной импульсно-кодовой модуляции (ADPCM). Для ясности, схема опускает детали, такие как форматирование битового потока, возможное использование сторонней информации, и адаптивных блоков. ADPCM-кодер пользуется тем фактом, что соседние звуковые сэмплы обычно похожи друг на друга. Вместо представления каждого звукового сэмпла независимо, как в PCM, ADPCM-кодер вычисляет разницу между каждым звуковым сэмплом и его предсказанным значением, и выводит PCM-значение этой разницы. Заметьте, что ADPCM-кодер (Схема 2a) использует большую часть компонент ADPCM-декодера (Схема 2b) для вычисления предсказанных значений.

## Схема 2. Сжатие и разжатие в ADPCM

Выход квантизатора – обычно только (знаковое) представление числа уровней квантизации. Реквантизатор восстанавливает значения квантизованного сэмпла, умножая число уровней квантизации на размер шага квантизации, и, возможно, добавляя отступ в половину шага. В зависимости от реализации квантизатора, этот отступ может быть необходим для центрирования восстановленного значения между порогами квантизации.

ADPCM-кодер может адаптироваться к характеристикам звукового сигнала, изменяя размер шага квантизатора, предсказателя, или их обоих. Метод вычисления предсказанного значения и способ, которым предсказатель и квантизатор адаптируются к звуковому сигналу, различаются среди разных систем кодирования ADPCM.

Некоторые ADPCM-системы требуют, чтобы кодер предоставлял стороннюю информацию вместе с разностными PCM-значениями. Эта сторонняя информация может служить двум целям. Во-первых, в некоторых ADPCM-схемах декодеру необходима дополнительная информация, чтобы установить размер шага предиктора или квантизатора, или их обоих. Во-вторых, эти данные могут предоставлять декодеру избыточную контекстную информацию, что позволяет восстанавливаться после ошибок в битовом потоке или предоставляет прямой доступ к закодированному потоку битов.

Следующий раздел описывает алгоритм ADPCM, предложенный Ассоциацией интерактивного мультимедиа (The Interactive Multimedia Association, IMA). Этот алгоритм обеспечивает коэффициент сжатия, равный (числу бит на исходный сэмпл)/4 к 1. Среди других ADPCM-схем сжатия звука – рекомендация G.721 CCITT (поток сжатых данных 32 килобита в секунду), рекомендация G.723 (поток сжатых данных 24 килобит в секунду) и алгоритм сжатия интерактивного аудио на компакт-диске.

*Алгоритм ADPCM IMA.* IMA – это консорциум производителей компьютерного оборудования и программного обеспечения, сотрудничающих с целью разработать стандарт де-факто для компьютерных мультимедиа-данных. Цель IMA для их предложения по сжатию звуковых данных была выбрать публично доступный алгоритм сжатия звука, способный обеспечить хорошее качество сжатого звука с хорошей производительностью сжатия данных. Кроме того, алгоритм должен был быть достаточно простым для программного расжатия, в реальном времени, звукового сигнала 44.1 кГц на 20-мегагерцевом (МГц) компьютере 386-класса. Выбранный алгоритм ADPCM не только отвечает этим целям, но и достаточно прост для программного сжатия в реальном времени на том же компьютере.

Простота предложения IMA ADPCM обусловлена грубостью его предиктора. Предсказываемое значение звукового сэмпла – просто декодированное значение следующего прямо перед ним сэмпла. Таким образом, блок предиктора на Схеме 2 – всего лишь элемент временной задержки, чей выход – это вход, отложенный на один интервал сэмплирования. Поскольку предиктор не адаптивен, сторонняя информация не нужна для восстановления предиктора.

Схема 3 демонстрирует блок-схему процесса квантизации, используемого алгоритмом IMA. Квантизатор выдаёт четыре бита, представляющих знаковую величину числа уровней квантизации для каждого входного сэмпла.

Адаптация к звуковому сигналу происходит только в блоке квантизации. Квантизатор подстраивает размер шага в зависимости от текущего размера шага и вывода квантизатора с предыдущего входа. Это подстройка может быть выполнена как два последовательных поиска в таблице. Три бита, представляющие число уровней квантизации, служат индексом для первого поиска, чей выход – смещение для второго поиска. Это смещение прибавляется к сохранённому значению индекса, и ограниченный по диапазону результат используется для второго поиска в таблице. Суммарное значение индекса сохраняется для использования на следующей итерации подстройки размера шага. Результат второго поиска в таблице – это новый размер шага квантизации. Заметьте, что имея начальное значение индекса для второго поиска в таблице, данные для подстройки полностью выводятся из выхода квантизатора; сторонняя информация не нужна для подстройки квантизатора. Схема 4 изображает блок-схему процесса подстройки размера шага, а таблицы 1 и 2 демонстрируют содержимое таблиц для поиска.

Схема 3. Квантизация IMA ADPCM

Таблица 1. Первый поиск в таблице для адаптации квантизатора IMA ADPCM

Таблица 2. Второй поиск в таблице для подстройки квантизатора IMA ADPCM

*IMA ADPCM: Восстановление от ошибок.* Как удачный побочный эффект архитектуры этой ADPCM-схемы, ошибки декодера, вызванные изолированными ошибками в кодовых словах или правками, склейками или случайным доступом к сжатому потоку бит, обычно не имеют катастрофического эффекта на выход декодера. Обычно это неверно для схем сжатия, использующих предсказание. Поскольку предсказание опирается на корректное декодирование предыдущего звукового сэмпла, ошибки в декодере склонны распространяться. Следующий раздел объясняет, почему распространение ошибок обычно ограниченно, и не гибельно для алгоритма IMA. Декодер восстанавливает звуковой сэмпл, , прибавляя предыдущий декодированный сэмпл, , к результату знаковой величины произведения кодового слова, , и размера шага квантизации, плюс смещение в половину размера шага:

где одной второй, плюс подходящее численное представление .

Анализ второго поиска в таблице размера шага открывает, что каждая следующая запись примерно в 1.1 раз больше предыдущей. Поскольку диапазон, ограничивающий индекс второй таблицы, не имеет места, значение примерно равно произведению предыдущего значения, , и функции от кодового слова, :

Два уравнения выше можно преобразовать, чтобы выразить декодированный сэмпл, , как функцию от размера шага и декодированного значения сэмпла в момент времени , и набор кодовых слов в диапазоне времени от до .

Заметьте, что элементы суммы – функция только от кодовых слов времени и далее. Ошибка в кодовом слове, , или случайная запись в поток битов во время , может привести к ошибке в разжатом выводе, , и размере шага квантизации, . Уравнение выше показывает, что ошибка в равняется константному отступу в будущих значениях . Это смещение неслышимо, если разжатое значение не выходит за рамки допустимого диапазона и не обрезается. Такое отсечение приводит к кратковременному слышимому искажению, но также служит для частичной или полной корректировки смещения. Уравнения выше также показывает, что ошибка в равняется нежелательному приросту или ослаблению будущих значений декодированного результата . Форма волны выходного сигнала не меняется, если индекс для второй таблицы поиска размера шага не выходит за рамки диапазона. Ограничение по диапазону приводит к частичной или полной коррекции значения размера шага.

Природа адаптации размера шага ограничивает влияние ошибки в размере шага. Заметьте, что ошибка в , вызванная ошибкой в одном кодовом слове, может изменить максимум или 7.45 дБ в значении размера шага. Заметьте также, что любая последовательность из 88 кодовых слов, имеющих значение 3 или меньше (см. Таблицу 1) полностью корректирует размер шага к его минимальному значению. Даже на самой низкой обычно используемой частоте сэмплирования, 8 кГц, 88 сэмплов соответствуют 11 миллисекундам звука. Таким образом, точки случайного доступа или редактирования существуют везде, где есть 11 миллисекунд низкоуровневого сигнала в звуковом потоке.

Сжатие MPEG/Audio

Алгоритм сжатия звука Экспертной группы по движущемуся изображению (The Motion Picture Experts Group, MPEG) – это стандарт ISO (International Organization for Standardization) для высокоточного сжатия звука. Это одна из трёх частей стандарта сжатия. С двумя другими частями, видео и систем, составной стандарт направлен на сжатие синхронизированных видео и звука при общем битрейте примерно 1.5 мегабит в секунду.

Как и µ-закон и ADPCM, сжатие MPEG/audio – с потерями, однако алгоритм MPEG может достигать ясного сжатия, практически без потерь (*по ощущениям*). Комитет MPEG/audio провёл обширные субъективные тесты на прослушивание во время разработки стандарта. Тесты показали, что даже при коэффициенте сжатия 6 к 1 (стерео, 16 бит на сэмпл на 48 кГц, сжатых до 256 килобит в секунду) и в идеальных для прослушивания условиях, эксперты не смогли различить сжатые и оригинальные аудиоклипы со статистической значимостью. Более того, эти клипы были специально выбраны, потому что их трудно сжимать. Гревин и Райден (Grewin, Ryden) описывают детали настройки, процедуры и результатов этих тестов.

Высокая производительность этого алгоритма сжатия объясняется использованием слухового маскирования. Это маскирование – слуховая слабость уха, которая проявляется, когда наличие сильного звукового сигнала делает спектрально близкие, более слабые звуковые сигналы невозможными для восприятия. Этот явление маскирования шума наблюдалось и было подтверждено множеством психоакустических экспериментов.

Эмпирические результаты также показывают, что ухо имеет ограниченную частотную избирательность, которая варьируется по остроте от менее чем 100 Гц для самых низких слышимых частот до более чем 4 кГц для самых высоких. Таким образом, слышимый диапазон может быть разбит на критические полосы, отражающие разрешающую способность уха как функцию от частоты. Схема 3 даёт список критических частот.

Из-за ограниченных разрешающих возможностей уха, порог для маскирования шума в любой заданной частоте зависит только от уровня сигнала внутри критической полосы этой частоты. Схема 5 иллюстрирует это свойство. Для сжатия звука, это свойство можно использовать, преобразуя звуковой сигнал в частотное представление, затем разделяя получившийся спектр на подполосы, аппроксимирующие критические полосы, и наконец квантизируя каждую подполосу в соответствие с слышимостью шума квантизации внутри этой полосы. Для оптимального сжатия, каждая полоса должна быть разбита на число уровней не больше, чем это необходимо, чтобы сделать шум дискретизации неслышимым. Следующие разделы представляют более детальное описание алгоритма MPEG/audio.

## Кодирование и декодирование MPEG/audio

На Схеме 6 изображены блок-схемы кодера и декодера MPEG/audio. В таком высокоуровневом представлении кодирование близко соответствует процессу, описанному выше. Входной аудио поток проходит через набор фильтров, который разделяет его на несколько полос. Одновременно с этим, входной аудио поток проходит через психоакустическую модель, определяющую отношение сигнал-маска для каждой из полос. Блок выделения битов или шума использует отношения сигнал-маска, чтобы решить, как распределить общее число кодовых битов, доступных для дискретизации сигналов полосы, чтобы минимизировать слышимость шума дискретизации. Наконец, последний блок принимает представление дискретизованных сэмплов, и форматирует данные в расжимаемый поток битов. Декодер просто обращает форматирование, затем реконструирует дискретизованные значения полос, и наконец отображает набор значений полосы в звуковой сигнал во времени. Как описано в требованиях MPEG, добавочные данные, не обязательно относящиеся к аудио потоку, могут быть добавлены в кодовый поток битов.

Стандарт MPEG/audio имеет три раздельных слоя для сжатия. Слой I составляет самый простой алгоритм, а слои II и III являются надстройками, использующими некоторые элементы из слоя I. Каждый следующий слой улучшает эффективность сжатия, но ценой большей сложности кодера и декодера.

*Слой I.* Алгоритм слоя I использует базовый набор фильтров, который присутствует во всех слоях. Этот набор фильтров разделяет аудио сигнал на 32 частотные полосы константной ширины. Фильтры относительно просты, и предоставляют хорошее разрешение по времени с приемлемым разрешением по частоте, по отношению к особенностям восприятия человеческого уха. Такая архитектура – компромисс, с тремя существенными уступками. Во-первых, 32 полосы константной ширины не точно отражают критические полосы уха. Схема 7 отражает это несоответствие. Диапазон слишком широк для низких частот, поэтому число бит дискретизации не может быть специально заточено под чувствительность к шуму в каждой конкретной критической полосе. Вместо этого, включённая критическая полоса с наибольшей чувствительностью к шуму определяет число бит дискретизации, необходимое для целой полосы фильтров. Во-вторых, набор фильтров и обратные им не являются преобразованиями без потерь. Даже без дискретизации обратное преобразование не восстановит идеально исходный сигнал. К счастью, ошибка, привносимая набором фильтров, мала и неслышима. Наконец, соседние полосы фильтров существенно перекрываются по частоте. Сигнал на единственной частоте может повлиять на выход двух соседних полос фильров.

Набор фильтров предоставляет 32 частотных сэмпла, один на полосу, для каждых 32 входных сэмплов. Алгоритм слоя I группирует вместе 12 сэмплов из каждой из 32 полос. Каждая группа из 12 сэмплов получает размещение битов и, если оно не нулевое, коэффициент масштабирования. Кодирование для сжатия стерео с избыточностью немного отличается, и обсуждается далее в этой статье. Размещение битов определяет число бит, используемых для представления каждого сэмпла. Коэффициент масштабирования – это множитель, который изменяет размер сэмплов, чтобы максимизировать разрешение дискретизации. Кодер слоя I преобразует 32 группы по 12 сэмплов (т.е., 384 сэмпла) во фрейм. Помимо аудио данных, каждый фрейм содержит заголовок, необязательное кодовое слово рефлексного избыточного кода и, возможно, добавочные данные.

*Слой II.* Алгоритм слоя II – это простое усовершенствование слоя I. Оно улучшает эффективность сжатия за счёт кодирования данных большими группами. Кодер слоя II создаёт фреймы 3 на 12 на 32 = 1,152 сэмпла на аудио канал. В то время как слой I кодирует данные отдельными группами по 12 сэмплов для каждой полосы, слой II кодирует данные в 3 группы, по 12 сэмплов на каждую полосу. Опять, не считая избыточного кодирования стерео, на каждое трио из 12 сэмплов выделяется один бит и до трёх коэффициентов масштабирования. Кодер сжимает с уникальным коэффициентов масштабирования для каждой группы из 12 сэмплов только если это необходимо, чтобы избежать слышимого искажения. Алгоритм слоя II также улучшает качество сжатия по сравнению со слоем I, представляя размещение битов, значения коэффициентов масштабирования и сэмплированные значения более эффективным кодом.

*Слой III.* Алгоритм слоя III демонстрирует гораздо более утончённый подход. Хоть он и основан на том же наборе фильтров что и в слоях I и II, слой III компенсирует некоторые недостатки этого набора фильтров с помощью подифицированного дискретного косинусного преобразования (modified discrete cosine transform, MDCT). На схеме 8 изображена блочная диаграмма этого процесса.

Далее MDCT разделяет выход набора фильтров по частоте, чтобы предоставить лучшее спектральное разрешение. Из-за неизбежного компромисса между разрешением по времени и по частоте, слой III определяет две различные длины блока MDCT: длинный блок в 36 сэмплов и короткий блок в 12. Короткая длина блока улучшает разрешение по времени, чтобы справиться с помехами. Заметьте, что короткий блок по длине составляет одну треть длинного. Когда он используется, три коротких блока заменяют один длинный. Переключение между длинными и короткими блоками не происходит моментально. Длинный блок со специализированным окном данных предоставляет механизм преобразования длинного блока в короткий. Слой III имеет три режима разбиения на блоки: два режима, когда выходы 32 наборов фильтров все могут пройти через MDCT с одинаковой длиной блока, и смешанный режим, когда 2 низкочастотные полосы используют длинные блоки, а 30 высокочастотных – короткие.

Другие основные улучшения по сравнению с алгоритмами слоёв I и II включают:

* Снижение помех дискретизации – Слой III определяет способ обработки значений MDCT, частично убирающий избыточность из-за перекрывающихся полос наборов фильтров в слоях I и II.
* Неравномерная дискретизация – Квантизатор слоя III возводит свой вход в степень 3/4 перед дискретизацией, чтобы обеспечить более постоянное отношение сигнал-шум на диапазоне квантования. Реквантизатор декодера MPEG/audio релинеаризует значения, возводя свой выход в степень 4/3.
* Энтропийное кодирование значений данных – Слой III использует коды Хаффмана для кодирования квантизованных сэмплов, для лучшего сжатия данных.
* Использование хранилища битов – Архитектура битового потока слоя III в больше степени соответствует переменной, по своей природе, длине сжатых данных. Как и слой II, слой III обрабатывает аудио данные фреймами по 1,152 сэмплам. В отличие от слоя II, сжатые данные, представляющие эти сэмплы, не обязательно размещаются во фрейме фиксированной длины в кодовом потоке битов. Кодер может добавлять или брать биты из хранилища, когда это приемлемо.
* Выделение шума вместо выделения битов – Процедура выделения битов в слоях I и II аппроксимирует количество шума от дискретизации только с точностью до заданного числа бит. Кодер слоя III использует цикл выделения шума. В этом цикле, квантизаторы переставляются в должном порядке, и получившийся шум дискретизации вычисляется и специально выделяется для каждой полосы.

Психоакустическая модель

Психоакустическая модель – ключевой компонент кодера MPEG, обеспечивающий высокое качество сжатия. Работа психоакустической модели – проанализировать входной аудио сигнал и выяснить, где на спектре шум дискретизации будет маскирован, и в какой степени. Кодер использует эту информацию чтобы решить, как лучше представить входной аудио сигнал конечным числом кодовых битов. Стандарт MPEG/audio предоставляет для примера две реализации психоакустической модели. Ниже приведено общее описание основных шагов, входящих в психоакустические вычисления каждой из моделей.

* Аудио данные, выровненные по времени – Психоакустическая модель должна принять во внимание задержку аудио данных при проходе через набор фильтров и отступ по данным, так что относящиеся к делу данные выровнены по центру внутри окна анализа. Например, при использовании второй психоакустической модели для слоя I, задержка через набор фильтров – 256 сэмплов, а отступ, необходимый чтобы отцентровать 384 сэмпла во фрейме слоя I в 512-точечном окне анализа есть (512-384) = 64 точки. Суммарный отступ – 320 точек, чтобы выровнять данные психоакустической модели с выходом набора фильтров.
* Преобразование звука в частотный формат – Психоакустическая модель использует отображение времени на частоту, например 512- или 1024-точечное преобразование Фурье. Стандартное взвешение Ханна, будучи применённым к аудио данным перед преобразованием Фурье, уменьшает краевые эффекты окна преобразования. Модель использует это отдельное и независимое отображение вместо набора фильтров, так как для подсчёта границ маскирования требуется более высокое разрешение по частоте.
* Разделение спектральных значений на критические полосы – Чтобы облегчить психоакустические вычисления, модель группирует значения частот в кванты по восприятию.
* Встраивание порога для тишины – Модель включает эмпирически находимый порог для абсолютного маскирования. Этот порог является нижней границей для маскирования шума, и определяется отсутствием маскирующих сигналов.
* Разделение на тональные и нетональные компоненты – Модель должна идентифицировать и разделить тональные и шумовые компоненты звукового сигнала, так как характеристики шума этих двух типов сигналов различаются.
* Применение функции распространения – Модель определяет пороги маскирования шума, применяя эмпирически выбираемую функцию маскирования или распространения к компонентам сигнала.
* Нахождение минимального порога маскирования для каждой полосы – Психоакустическая модель вычисляет пороги маскирования с более высоким разрешением по частоте, чем та, что предоставляется набором фильтров. Если набор фильтров широк по сравнению с критической полосой (в нижней части спектра), модель выбирает минимальный из порогов маскирования, покрытых полосой фильтра. Если полоса фильтра узка по сравнению с критической полосой, модель использует среднее порогов маскирования, покрытых полосой фильтра.
* Вычисление отношения сигнал-маска – Психоакустическая модель берёт минимальный порог маскирования и вычисляет отношение сигнал-маска. Затем это значение передаётся в секцию выделения битов (или шума) кодера.

Избыточное кодирование стерео

Алгоритм сжатия MPEG/audio поддерживает два типа избыточного стерео кодирования: интенсивное стерео кодирование и middle/side (MS) стерео кодирование. Оба типа избыточного кодирования используют ещё одну слабость уха. Психоакустические результаты показывают, что внутри критических полос, покрывающих частоты свыше примерно 2 кГц, восприятие ухом стерео сигнала базируется скорее на временной поверхности сигнала, чем на его подробной временной структуре. Все слои поддерживают интенсивное стерео кодирование. Слой III также поддерживает MS стерео кодирование.

При интенсивном стерео кодировании, кодер сжимает некоторые высокочастотные выходы наборов фильтров одним суммарным сигналом, а не посылает независимые коды для левого и правого канала на каждый из 32 выходов набора фильтров. Декодер восстанавливает левый и правый каналы на основе только независимых левого и правого коэффициентов масштабирования. При этом, спектральная форма обоих каналов совпадает, но величина сигнала различается.

MS кодирует сигналы для левого и правого каналов в некоторых диапазонах частот как средний (сумма левого и правого) и боковой (разность левого и правого) каналы. В этом режиме кодер использует специальные техники для дополнительного сжатия сигнала бокового канала.

Программные реализации реального времени

Программные реализации алгоритмов µ-закона и ADPCM легко могут работать в реальном времени. С помощью одного поиска в таблице можно произвести сжатие или разжатие по µ-закону. Программная реализация алгоритма IMA ADPCM может обрабатывать стерео звук с частотой сэмплирования 44.1-кГц в реальном времени на компьютере с частотой 20-МГц 386-класса. Испытание заключается в разработке программной реализации реального времени для алгоритма MPEG/audio. Стандарт MPEG в этом отношении не даёт подсказок: есть гораздо более рациональные способы проводить вычисления, необходимые для кодирования и декодирования, нежели те, что указаны в стандарте. Например, в следующем разделе показано, как число умножений и сложений, используемых в определённых вычислениях, можно сократить в 12 раз.

Схема 9 показывает график для анализа фильтра полос, используемого кодером MPEG/audio. Большая часть вычислительной нагрузки приходится на второй с конца блок. Этот блок содержит следующее матричное умножение:

для i=0..31

Если использовать уравнение выше, каждое из 31 значений S(i) требует 63 сложений и 64 умножений. Чтобы оптимизировать эти вычисления, заметим, что коэффициенты M(i,k) похожи на коэффициенты, используемые 32-точечным ненормализованным обратным дискретным косинусным преобразованием:

для i=0..31

В самом деле, S(i) идентично f(i), если F(k) вычисляется следующим образом:

Также возможно применение многих других оптимизаций для кодера и декодера MPEG/audio. Такие оптимизации позволяют программной реализации декодера MPEG/audio слоя I или II работать в реальном времени для декодирования высококачественного монофонического звука на DECstation 5000 Model 200. Эта машина использует 25-МГц R3000 MIPS CPU и имеет 128 килобайт кэша внешних инструкций и данных. С этим оптимизированным ПО, алгоритм MPEG/audio слоя II требует в среднем 13.7 секунд времени CPU (12.8 секунд времени пользователя и 0.9 секунд системного времени), чтобы декодировать 7.47 секунд стерео звука с частотой сэмплирования 48 кГц с 16 битами на сэмпл.

## Заключение

Техники сжатия звукового сигнала общего вида включают µ-закон и адаптивную дифференциальную импульсно-кодовую модуляцию. Эти простые подходы применяют алгоритмы низкой сложности, низкой степени сжатия и среднего качества сжатого звука к аудио сигналам. Третий метод, алгоритм сжатия MPEG/audio, это стандарт ISO для высокоточного сжатия звука. Стандарт MPEG/audio имеет три слоя, имеющие возрастающие сложность и качество сжатия.

КРАТКИЙ СЛОВАРЬ

|  |  |
| --- | --- |
| band | Полоса, диапазон |
| annealing | Нормализация, выравнивание |
| ban | Ограничение, запрет |
| bandwidth | Ширина полосы, диапаон частот |
| bet | Пари, ставка |
| bias | Смещение, сдвиг, расхождение |
| bind | Привязка, связь, поток данных |
| blur | Нерезкость, размытие |
| capacity | Ёмкость, вместительность |
| coalescence | Объединение, слияние, смешение |
|  |  |

СПИСОК ПРОЧИТАННОЙ ЛИТЕРАТУРЫ

|  |  |  |
| --- | --- | --- |
| Номер по порядку | Полное библиографическое название источника с указанием страниц оригинала | Количество страниц |
| 1 | David J.C. MacKay. Information Theory, Inference, and Learning Algorithms – Cambridge University Press, 2003 | 611 |

Список прочитанной литературы заверяю