



Audiophile's Software

digital audio encoding and playback

Логин: Пароль:

Запомнить скрытый

[Забыл пароль](#) | [Регистрация \(убрать всю рекламу\)](#)

[О сайте](#) | [Ликбез](#) | [Словарь](#) | [Audiophile's Testroom](#) | [Поддержать](#) | [Контакты](#)

Разделы

- » Главная
- » Новости
- » Софт
- » Статьи
- » Видео
- » Блог
- » Форум

Поиск по сайту

Популярное

- » [foobar2000 + плагины](#)
- » [Настройка звука](#)
- » [Драйвера Creative X-Fi](#)
- » [foobar2000 by Audiophile](#)
- » [Кодеры и утилиты](#)
- » [Настройка конвертера](#)
- » [Audiophile's Software](#)
- » [LAME + настройка](#)
- » [Драйвера SB Audigy](#)
- » [Настройка видео](#)

Полезный софт



Случайный опрос

Какой формат аудио Вы предпочитаете для прослушивания на портативном плеере/телефоне?

- MP3
- Lossless
- Ogg Vorbis
- AAC (MP4, M4A)
- другой

[[Результаты](#) | [Все опросы](#)]
Всего ответов: **1567**

Похожие проекты

- [digitalaudio.me](#)
- [AudioCoding.ru](#)
- [foobar2000.at.ua](#)
- [plastinka.org](#)
- [Cornerstone.ucoz.ru](#)
- [iLAB.me](#)

Follow me



Сейчас на сайте

Онлайн всего: **20**
Гостей: **18**
Пользователей: **2**
[ninnorio](#), [AntonDorn](#)

[Главная](#) » [Статьи](#) » [Обучающие видеоролики](#)

Xiph.org: A Digital Media Primer For Geeks (русские субтитры)

25 Марта 2013, 15:33



Краткий, но очень наглядный ликбез по основам цифрового представления и кодирования видео.

Текст видео

Рабочие станции и топовые персональные компьютеры справляются с обработкой цифрового звука довольно легко последние пятнадцать лет. Только около пяти лет приличные рабочие станции способны справиться с несжатым видео без кучи дорогого специализированного оборудования. Но сегодня даже самый дешевый домашний компьютер имеет процессор и жесткий диск, достаточные, чтобы реально раскидывать несжатое видео, по крайней мере, без излишнего напряжения.

Итак, теперь, когда у каждого есть все это необходимое недорогое оборудование, все больше людей, что неудивительно, хотят заниматься интересными вещами с цифровым контентом, особенно потоковым воспроизведением.

YouTube явился первым огромным успехом, и теперь каждый хочет поучаствовать.

Ну хорошо! Потому что это очень весело! Совсем не проблема найти потребителя цифрового контента. Но здесь я бы хотел обратиться к инженерам, к математикам, хакерам, людям, заинтересованным в открытиях, создании вещей и создании самих технологий. Людям, близким мне по духу.

Цифровой контент, особенно сжатие, воспринимается как что-то сверх-элитное, почему-то невероятно более сложное, чем что-либо другое в компьютерной науке. Крупные индустриальные игроки в этой области вообще не против такого восприятия; это помогает оправдывать ошеломляющее количество фундаментальных патентов, которые они держат. Им нравится образ, представляющий их исследователей лучшими из лучших, которые настолько умнее всех остальных, что их блестящие идеи не могут быть даже поняты простыми смертными.

Это чужь. Цифровое аудио, видео, потоковая передача и сжатие представляют бесконечно глубокие и стимулирующие умственные испытания, как и любая другая дисциплина. Это выглядит элитным, потому что слишком мало людей вовлечены. Так мало людей были вовлечены, возможно, потому, что так мало людей могли позволить себе необходимое дорогое специализированное оборудование.

Но сегодня почти любой смотрящий это видео, имеет дешёвый универсальный компьютер, достаточно мощный, чтобы играть с большими парнями.

Сегодня идут сражения вокруг HTML5, браузеров, видео и открытого против закрытого. Поэтому, сейчас самое подходящее время принять участие. Легче всего начать, возможно, с понимания у нас технологии.

Это — вступление. Поскольку это вступление, оно опускает массу деталей, так что большую картину несколько легче понять. Порядочное число смотрящих уже знакомы со всем, о чем я буду говорить, по крайней мере пока. С другой стороны, возможно, я буду рассказывать слишком быстро для тех, кто совсем не знаком со всем этим, поэтому, если все это ново для вас, расслабьтесь. Важно выделить все идеи, которые действительно захватывают ваше воображение.

Особенно уделите внимание терминологии, связанной с этими идеями, потому что с ними, а так же с Гуглом и Википедией, вы можете копать настолько глубоко, насколько вы заинтересованы.

Итак, без лишнего церемоний, добро пожаловать в чертовски хорошее хобби.

Звук — это распространение волн давления по воздуху, расходящихся от источника подобно кругам на воде от брошенного в пруд камня. Микрофон, или человеческое ухо в его роли, преобразовывает эту распространяющуюся пульсацию в электрический сигнал.

Верно, это курс средней школы, это помнит каждый. Двигаемся дальше. Этот аудиосигнал является одномерной функцией, единственным значением, изменяющимся во времени. Если мы немного замедлим осциллограф... это будет несколько легче увидеть.

Также важны несколько других аспектов сигнала. Он непрерывен и во времени и в значениях; то есть, в любой момент времени у него есть действительное значение, а также его значение плавно меняется от одной точки во времени к другой. Не важно, насколько мы увеличиваем масштаб, не будет никаких разрывов, сингулярностей, мгновенных изменений значений или точек, в которых сигнал прекращает существовать. Он определён везде. Классическая непрерывная математика хорошо применима к этим сигналам.

Цифровой сигнал, с другой стороны, является дискретным по значению и по времени. В самой простой и наиболее общепринятой системе, которая называется "импульсно-кодовая модуляция", каждое из фиксированного числа значений напрямую представляет амплитуду непрерывного сигнала в точках времени, разделённых равными промежутками. Конечным результатом является поток чисел.

Теперь это выглядит ужасно похоже на это. Кажется интуитивным, что должен быть способ как-то строго преобразовывать одно в другое, и, хорошие новости, теорема отсчетов гласит, что мы можем, и рассказывает, как.

Опубликованная в своей самой известной форме Клодом Шенноном в 1949 и основанная на работах Найквиста и Хартли, и многих других, теорема отсчетов не только утверждает, что мы можем переходить туда-обратно между аналоговым и цифровым представлением, но также устанавливает ряд условий, при которых конверсия происходит без потерь и два представления становятся эквивалентными и взаимозаменяемыми. Когда эти условия не выполняются, теорема отсчетов говорит нам, как и сколько информации потеряно или испорчено. До самого последнего времени, аналоговая технология была основой для практически всей работы со звуком, и не только потому, что большинство аудиоданных имеют аналоговое происхождение. Вы могли подумать, что раз компьютеры появились относительно недавно, технология аналогового сигнала, должно быть, появилась первой. Нет. Цифровая, на самом деле, старше. Телеграф предшествовал телефону на полвека и был почти полностью механически автоматизирован к 1860м, посылая закодированные, мультиплексированные цифровые сигналы на длинные дистанции. Вы знаете... Телеграфная лента. Гарри Найквист из Bell Labs исследовал импульсную телеграфическую передачу, когда опубликовал своё описание того, что потом стало известно как частота Найквиста, ключевая концепция теоремы отсчетов. Теперь правда в том, что телеграф передавал символическую информацию, текст, а не оцифрованный аналоговый сигнал, но с пришествием телефона и радио технологии аналогового и цифрового сигнала развивались быстро бок о бок. Аудио всегда обрабатывалось как аналоговый сигнал потому что... ну, это настолько легче! Фильтр низких частот второго порядка, например, требует двух пассивных компонентов. Полностью аналогичное кратковременное преобразование Фурье — нескольких сотен.

Хорошо, возможно, тысячи, если вы хотите собрать что-то действительно хорошее. Обработка сигналов цифровым способом требует

Авторские статьи

- » [Настройка вывода звука](#)
- » [Настройка X-Fi @ Windows 7](#)
- » [Настройка конвертера](#)
- » [Сравнение lossless](#)
- » [Сравнение lossy](#)
- » [Сравнение ресемплеров](#)
- » [Как проводится ABX тест](#)
- » [24bit/96kHz->16bit/44.1kHz](#)
- » [Декодирование в foobar2000](#)
- » [Вывод стерео на 5.1](#)
- » [foobar2000 и Last.fm](#)

Сообщество

Последнее на форуме

- » [LAME MP3 Encoder \(150\)](#)
- » [Настоящий ли 24bit'ный трек или нет?.. \(3\)](#)
- » [\[flood\] Ваше первое сообщение \(5\)](#)
- » [Сборка foobar2000 by Audiophile \(273\)](#)
- » [Интернет-радио \(28\)](#)
- » [Выбор акустической системы \(238\)](#)
- » [Asus Xonar D1 vs Asus Xonar DX \(2\)](#)

Кодеки



Теги

Полезные ссылки

- » [WebSound.Ru](#)
- » [HydrogenAudio.Org](#)
- » [Rarewares.Org](#)
- » [iXBT.com](#)
- » [Virtual Audio Cable](#)

от миллионов до миллиардов транзисторов, работающих на микроволновых частотах, аппаратную поддержку по меньшей мере оцифровки и реконструкции аналоговых сигналов, законченную программную экосистему для программирования и управления этим миллиардно-транзисторным Джамгернаутом, и цифровые хранилища, на тот случай, если вы хотите сохранить результаты на будущее... Таким образом, мы приходим к выводу, что только аналоговый подход практичен для аудио... ну, по крайней мере, если у вас не завалены миллиарды транзисторов и все остальное. А поскольку у нас завалился, цифровая обработка сигнала становится очень заманчивой. Во-первых, аналоговые комплекты просто не имеют гибкости компьютера общего назначения. Добавление новой функции в этого зверя... да, вероятно, этого не произойдет. Хотя на цифровом процессоре... просто напишите новую программу. Программирование не тривиально, но оно намного легче.

Возможно более важно, что каждый аналоговый компонент является приближением. Не существует такой вещи как идеальный транзистор или идеальный проводник, или идеальный конденсатор. В аналоговой обработке каждый компонент добавляет шума и искажений, обычно не очень много, но они суммируются. Одна только передача аналогового сигнала, особенно на длинные дистанции, постепенно, умеренно, нообитом портит его. Кроме того, все эти одноцелевые аналоговые компоненты занимают много места. Две строки кода на миллиарде транзисторов здесь могут реализовать фильтр, который бы потребовал индуктора размером с холодильник. Цифровые системы не имеют этих недостатков. Цифровые сигналы могут храниться, копироваться, обрабатываться и передаваться без наложения шумов и искажений. Да, мы используем время алгоритмы с потерей, но единственно неизбежными неидеальными шагами являются оцифровка и восстановление, когда цифровое должно взаимодействовать со всем этим неряшливым аналоговым.

Неряшливые или нет, современные этапы преобразования очень, очень хороши. По меркам наших ушей, мы можем также считать их без потерь. С небольшим дополнительным оборудованием, большинство из которого компактно и недорого благодаря нашей современной промышленной инфраструктуре, цифровое аудио одерживает чистую победу над аналоговым. Давайте тогда перейдем к хранению, копированию, обработке и передаче.

Импульсно-кодовая модуляция — самое распространённое представление необработанного аудио. Существуют и другие представления, например, сигма-дельта кодирование, используемое в SACD, которое является разновидностью импульсно-кодовой модуляции. Что говорит, что импульсно-кодовая модуляция значительно доминирует, главным образом потому, что она так математически удобна. Аудио инженер может провести целую карьеру, не работая ни с чем другим. Кодирование ИКМ можно охарактеризовать тремя параметрами, что позволяет посчитать каждый возможный вариант ИКМ без больших хлопот. Первый параметр — это частота дискретизации. Навысшая частота, которую кодирование может передать, называется частотой Найквиста. Частота Найквиста ИКМ равняется ровно половине частоты дискретизации. Следовательно, частота дискретизации непосредственно определяет наивысшую возможную частоту в оцифрованном сигнале.

Аналоговые телефонные системы традиционно ограничивали полосу голосовых каналов до чуть менее, чем 4 кГц, поэтому цифровая телефония и самые классические голосовые приложения используют частоту дискретизации 8 кГц, минимальную частоту, необходимую для охвата целевой полосы в 4 кГц.

Вот как звучит частота дискретизации в 8 кГц — немного приглушённо, но совсем разборчиво для голоса. Это самая низкая частота дискретизации, когда-либо широко использованная на практике. С этого пор с увеличением производимости памяти и накопителей, компьютеры потребителей смогли обеспечить частоту 11, потом 16, потом 22, а потом — 32 кГц. Очевидно, что с каждым увеличением частоты дискретизации и частоты Найквиста, качество звучания топовых конфигураций становится немного чище и звучит более естественно. Compact-Disk использует частоту дискретизации 44,1 кГц, которая немного лучше, чем 32 кГц, но улучшения становятся все менее различимыми. 44,1 кГц — это несколько странноватый выбор, особенно при том, что он не используется до Compact-Disk, но огромный успех компакт-дисков сделал его общепринятой частотой. Наиболее распространённой частотой дискретизации высокого качества за пределами CD является 48 кГц. Фактически между ними нет ощутимой разницы в звучании. Это видео, по крайней мере его оригинальная версия, была снята и выпущена с аудио 48 кГц, что является первоначальным стандартом для высококачественного звука в видеозаписях.

Сверхвысококачественные частоты дискретизации в 88, 96 и 192кГц также имели место. Суть в дискретизации на частоте выше 48 кГц заключается не в дальнейшем расширении высокочастотного диапазона. Его суть в другом. Отвлекаясь на секунду, скажу, что французский математик Жан Батист Жозеф Фурье показал, что мы также можем рассматривать сигналы вроде аудио как набор составных частот. Это представление частотной области эквивалентно временному представлению, сигнал точно такой же, просто мы рассматриваем его по-другому. Здесь мы видим представление частотной области гипотетического аналогового сигнала, который мы собираемся оцифровать. Теорема Найквиста-Шеннона рассказывает о двух вещах, связанных с процессом дискретизации. Во-первых, цифровой сигнал не может передавать частоты выше частоты Найквиста. Во-вторых, и это новая часть, если мы не удалим их низкочастотным фильтром перед дискретизацией, этот процесс обрежет их до допустимого диапазона, создав искажения наложения. Наложение, в двух словах, звучит ужасно, поэтому естественно удалять все частоты ниже частоты Найквиста перед дискретизацией и после восстановления.

Считается, что восприятие звука человеком простирается примерно до 20 кГц. При частоте 44,1 или 48 кГц, фильтрация низких частот перед стадией дискретизации должна быть предельно точной, чтобы избежать удаления слышимых частот ниже 20 кГц, и в то же время не допуская попадания частот выше частоты Найквиста в процесс дискретизации. Этот тяжело реализуемый фильтр и ни один фильтр на практике не преуспел полностью. С другой стороны, если частота дискретизации 96 кГц или 192 кГц, низкочастотный фильтр имеет дополнительно октаву или две в диапазоне модуляции. Такой фильтр намного легче построить. Частоты дискретизации ниже 48 кГц являются на самом деле одним из неприятных компромиссов на аналоговом этапе.

Второй фундаментальный параметр ИКМ — это формат выборки, то есть форма каждого цифрового числа. Число — это число, но оно может быть представлено несколькими разными путями. Ранняя ИКМ была линейной 8 битной, кодировалась беззнаковым байтом. Динамический диапазон ограничен примерно 50 дБ и шум квантования, как вы можете слышать, достаточно резок. Восемьбитное аудио сегодня исчезающе встречается. В цифровой телефонии обычно используется одна из двух связанных нелинейных восьмибитных кодировок, называемых А-закон и мю-закон. Эти форматы кодируют примерно 14 битный динамический диапазон в квантования битах, раздвигая более высокие значения амплитуды дальше друг от друга. А-закон и мю-закон, очевидно, улучшают шум квантования в сравнении с линейным 8-битным, а голосовые гармоники хорошо скрывают оставшийся шум квантования. Все три восьмибитные кодировки — линейная, А-закон и мю-закон, обычно используются вместо с частотой дискретизации 8 кГц, хотя я демонстрирую их на 48 кГц. Большинство современных ИКМ использует знаковые целые 16 или 24 бит с дополнением до двух для кодирования диапазона от минус бесконечности до нуля децибел с точностью 16 или 24 бита. Максимальное абсолютное значение соответствует нулю децибел. Как и во всех современных форматах дискретизации, сигналы выше нуля децибел и, таким образом, за пределами представимого диапазона, образуются. В кодировании и сведении не является редкостью использование чисел с плавающей точкой для ИКМ вместо целых чисел. 32-битное число с плавающей точкой, обыкновенная разновидность плавающей точки на современных компьютерах, представляет 24 бит для разрешения, а семь бит плавающей экспоненты увеличивает передаваемый диапазон. В числах с плавающей точкой, ноль децибел обычно представляется как +/-1.0, и, поскольку они могут, очевидно, представлять диапазон значительно выше этого, временно превышение нуля децибел в процессе смешивания не приводит к обрезанию. ИКМ с плавающей точкой занимает больше объема, поэтому она, как правило, используется только в качестве промежуточного производственного формата. Наконец, большинство компьютеров общего назначения по-прежнему читают и пишут данные восьмизрядными байтами, потому важно помнить, что семплы больше восьми бит могут быть в порядке big или little endian, и оба порядка распространены. Например, файлы Microsoft WAV — in little endian, а Apple AIFC — как правило в big-endian. Имейте это в виду.

Третий параметр ИКМ — это число каналов. Соглашением в сыром ИКМ является кодирование множества каналов, чередуя семплы каждого канала вместе в едином потоке. Просто и расширяемо. Вот и все. Это описывает все возможные представления ИКМ.

Готово. Цифровое аудио _настолько просто! Конечно, есть ещё много чего, но на данный момент у нас есть хороший полезный кусок аудиоданных, давайте добудем также немного видео.

Можно рассматривать видео как аудио, только с двумя пространственными измерениями, X и Y, в дополнение к измерению времени. Математически, это звук. Теорема отсчетов применима ко всем трем измерениям видео точно также, как и к одному измерению времени в аудио. Очевидно, на практике аудио и видео немного различны. С одной стороны, в сравнении с аудио, видео огромно. Несжатое CD-аудио — это примерно 1,4 гигабайта в секунду. Несжатое видео 1080i HD — более 700 мегабит в секунду. Приходится захватывать, обрабатывать и сохранять более, чем в 500 раз больше данных в одну секунду. По закону Мура... выходит... давайте посмотрим... примерно восемь удвоенный раз в два года, так что да, компьютеры требуют около пятнадцати дополнительных лет для обработки несжатого видео после укорочения несжатого аудио. Основы сырого видео также несколько сложнее, чем основы сырого аудио. Большой объем данных в настоящее время требуется представление более эффективного, чем линейная ИКМ, используемая в аудио. К тому же, электронное видео происходит практически полностью из одного телевизионного вещания, а компьютеры по стандартам, которые регулируют телевещание всегда были очень озабочены обратной совместимостью. До самых последних лет в США старый, шестидесятилетний черно-белый телевизор мог по-прежнему показывать обычные аналоговые телевизионные передачи. На самом деле это действительно искусный трюк. Недостатком обратной совместимости является то, что если когда-то что-то попадает в стандарт, вы не можете на самом деле больше никогда выкинуть его. Электронное видео никогда не переделывалось с нуля, как это неоднократно с аудио. Шестдесят лет стоят умных но устаревших хаков, необходимых в связи с уходом технологии данной эпохи, образующих на самом деле нагромождение, а поскольку цифровые стандарты также пришли с телевидения, все эти жуткие хаки были перенесены также и в цифровые стандарты. Вкратце, в цифровом видео замешано намного больше деталей, чем в аудио. Нет надежды охватить здесь их все полностью, поэтому мы широко охватим основы.

Наиболее очевидными параметрами сырого видео являются ширина и высота изображения в пикселях. Так же просто, как это может показаться, разрешение в пикселях само по себе на самом деле не определяет абсолютные ширины и высоты изображения, поскольку большинство видео, происходящего из телевидения не использует квадратные пиксели. Число растровых строк в трансляционном изображении было фиксированным, но действительное количество горизонтальных пикселей зависело от ширины канала. Действительное горизонтальное разрешение могло приводить к тому, что пиксели были либо уже, либо шире, чем интервал между строками. Стандарты в основном указывают, что оцифрованное видео должно отражать реальное разрешение оригинального аналогового источника, поэтому большое количество цифрового видео также использует не квадратные пиксели. Например, обычное NTSC DVD с соотношением 4:3 обычно кодируется с разрешением экрана 704 на 408, соотношение, более широкое, чем 4:3. В данном случае сами пиксели получают соотношение 10:11, что делает их выше собственной ширины и сужает изображение горизонтально до правильного соотношения. Такое изображение должно проходить ресемплинг для корректного отображения на дисплеях с квадратными пикселями. Второй очевидный параметр видео — это частота смены кадров, количество полных кадров в секунду. Активно используются несколько стандартных частот кадров. Цифровое видео, в той или иной форме, может использовать их все. Или, другую частоту кадров. Или даже переменные частоты, когда частота кадров адаптивно меняется по ходу видео. Чем выше частота кадров, тем более плавны движения, и это приводит нас, к сожалению, к интерлейсингу. В самые ранние дни трансляции видео, инженеры искали наивысшую практическую частоту кадров для гладкого движения и с минимальным мерцанием на фосфорных ЭЛТ. Им было необходимо использовать минимальную ширину полосы для достижения максимального разрешения и наивысшей частоты кадров. Их решением было "сплеть" видео, когда четные линии отправляются в один проход и нечетные — в следующий. Каждый проход называется полем, а два поля как бы образуют законченный кадр. Как бы — потому что четные и нечетные поля на этом деле не принадлежат одному исходному кадру. В изображениях с 60 полями в секунду исходная частота кадров соответствует 60 полным кадрам в секунду, а половина каждого кадра, каждая лишняя линия, просто отображается. Вот почему мы не можем применить деинтерлейсинг к видео простым комбинированием двух полей в один кадр: на самом деле они не из одного и того же кадра. Электронно-лучевая трубка была единственной доступной технологией отображения на протяжении большей части истории электронного видео. Выходная яркость ЭЛТ нелинейна, приблизительно равняется входному управляющему напряжению, возведённому в степень 2,5. Эта степень, 2,5 — избранная гамма, и поэтому она часто обозначается как гамма дисплея. Камеры, напротив, являются линейными, и если вы передадите ЭЛТ линейный входной сигнал, это будет выглядеть следующим образом. Поскольку первоначально было очень мало камер, которые были фантастически дорогими, и предполагалось много, много телевизоров, которые должны быть как можно менее дорогими, инженеры решили добавить необходимую схему гамма-коррекции в камеры, а не телевизоры. Видео, переданное по радиоволнам, таким образом, должно иметь нелинейную интенсивность, используя обратную степень гаммы телевизора, поэтому, как только сигнал с камеры, наконец, отображен на ЭЛТ,

общий ответ системы камера-телевизор вновь становится линейным. Почти.

Также есть ещё две настройки. Телекамера на самом деле использует экспоненту гаммы, обратную 2,2, а не 2,5. Это просто коррекция для просмотра в темном окружении. Кроме того, экспоненциальная кривая переходит в линейный закон вблизи черного цвета. Это просто старый хак для подавления шума датчика в камере. Гамма-коррекция также имела удачную выгоду. По воле случая, человеческий глаз имеет гамму восприятия около 3. Это относительно близко к гамме ЭЛТ, 2,5. Изображение, использующее гамма-коррекцию, уделяет больше разрешения низким интенсивностям, где, как оказалось, у глаза наилучшая различимость, а следовательно, использует имеющееся разрешение более эффективно.

Хотя в настоящее время ЭЛТ исчезают, стандартный дисплей компьютера sRGB по-прежнему использует нелинейную кривую интенсивности, схожую с телевизионной, с линейным законом в области черного, с экспоненциальной кривой со значением гаммы 2,4. Это кодирует шестнадцатититный линейный диапазон в восьми битах.

В человеческом глазу есть три отдельных цветовых канала — красный, зеленый и синий, и большинство дисплеев использует эти три цвета как компоненты смешивания, чтобы получить полный спектр цветов. В печати, по той же причине, основными пигментами являются голубой (Cyan), пурпурный (Magenta) и Желтый (Yellow); пигменты являются вычитаемыми, и каждый из этих пигментов вычитает один чистый цвет из отраженного света. Голубой вычитает красный, пурпурный вычитает зеленый, а желтый вычитает синий. Видео может, и иногда бывает, представлено красным, зеленым и синим каналами, но RGB-видео нетипично. Человеческий глаз намного более чувствителен к яркости, нежели к её цвету, а RGB стремится распределять энергию на изображение равномерно по всем трем каналам. Посмотрите, красная плоскость выглядит как красная версия оригинально изображения, зеленая плоскость выглядит как зеленая версия оригинального изображения, а синяя плоскость выглядит как синяя версия оригинально изображения. Черно-белый в тройном объеме. Неэффективно. По этой причине и потому, что телевидение изначально все равно было черно-белым, видео обычно представляется как канал яркости высокого разрешения, черно-белый, вместе с дополнительными, зачастую с меньшим разрешением, каналами цвета. Канал яркости, Y, получают взвешиванием, а затем сложением отдельных, красного, зеленого и синего сигналов. Каналы насыщенности (цвета) U и V потом получают вычитанием сигнала яркости из синего, и вычитанием сигнала яркости из красного. Когда YUV масштабируется, смещается и дискретизируется для цифрового видео, его на самом деле правильнее называть Y'CbCr, но более общий термин YUV широко распространен для описания всех аналоговых и цифровых вариантов данной цветовой модели. Цветовые каналы U и V могут быть того же разрешения, что и канал Y, но, поскольку человеческий глаз имеет куда худшее пространственное разрешение насыщенности цвета, нежели разрешение яркости, разрешение насыщенности обычно берется половиной или даже четвертью в горизонтальном направлении, вертикальном, или обоих, обычно без какого-либо существенного влияния на качество видимого изображения. Практически любой возможный вариант субдискретизации был использован в то или иное время, но сегодня распространенными являются 4:4:4, который, в действительности вообще не субдискретизируется, 4:2:2, в котором горизонтальное разрешение каналов U и V делится пополам, и наиболее общепринятый из всех, 4:2:0, при котором и горизонтальное и вертикальное разрешение каналов насыщенности делятся пополам, что приводит к тому, что каждая из плоскостей U и V в четыре раза меньше Y. Термины 4:2:2, 4:2:0, 4:1:1 и т.д., и т.п. не являются полными описаниями субдискретизации насыщенности. Существует несколько возможных способов позиционирования пиксельной насыщенности относительно пикселей яркости, и опять, есть несколько активно используемых для каждой субдискретизации. Например, motion JPEG, MPEG-1 video, MPEG-2 video, DV, Theora и WebM, все используют или могут использовать субдискретизацию 4:2:0, но они располагают цветовые пиксели тремя разными способами. Motion JPEG, MPEG1 video, Theora и WebM располагают пиксели насыщенности горизонтально и вертикально между пикселями яркости. MPEG2 располагает пиксели насыщенности между линий, но горизонтально выравнивает их с каждым пикселем яркости.

Режимы с интерлейсингом несколько все усложняют, размещая их немного странно. И наконец, PAL-DV, который всегда использует интерлейсинг, размещает пиксели насыщенности в горизонтальном направлении в том же положении, что и пиксели яркости, а вертикально чередует канал насыщенности в каждой линии. И это все — только видео 4:2:0.

Я оставлю остальные варианты субдискретизации как домашнее задание зрителю. У вас есть основная идея, двигайтесь дальше.

В аудио мы обычно представляем несколько каналов в потоке ИКМ, чередуя по порядку семплы каждого канала. Видео использует как закопанные форматы, которые переплетают цветные каналы, так и плоские форматы, которые хранят пиксели из каждого канала вместе в отдельных плоскостях, составленных по порядку к кадре. Существуют как минимум 50 различных форматов в этих двух больших категориях, возможно, десять или пятнадцать из которых широко используются. Каждая субдискретизация насыщенности и разная битность требуют разного порядка упаковки, а потому и разного формата пикселей. Для каждой уникальной субдискретизации обычно также существует несколько эквивалентных форматов, состоящих из тривиальных перестановок или перепакетов порядка каналов, связанных либо с удобством на определенном железе в определенном случае либо, иногда, со старой доброй завистью. Пиксельные форматы описываются уникальным названием четырехсимвольного кода (FourCC). Их существует довольно много, и нет смысла сейчас приводить их все. Google ваш друг. Имейте в виду, что коды FourCC для сырого видео определяют порядок пикселей и субдискретизации насыщенности, но в общем случае не говорят ничего конкретного о размещении пикселей насыщенности или пространстве цветов. Для примера, YV12 video может использовать размещение JPEG, MPEG-2 или DV, а также любой из нескольких определений пространств цвета YUV. Это завершает наше не столь быстрое и ещё не совсем законченное путешествие в сырое видео.

Хорошие новости в том, что мы уже можем выполнить достаточно много настоящей работы, опираясь на данный обзор. Во множестве ситуаций, кадр видеоданных является кадром видеоданных. Детали приобретают значение, когда приходит время писать программы, а сейчас я удовлетворён тем, что уважаемый зритель широко осведомлен в соответствующих вопросах.

Итак. У нас есть аудиоданные. У нас есть видеоданные. Что осталось — так более знакомые не-сигнальные данные и непосредственное проектирование программного обеспечения, используемого разработчиками. И многое из всего этого!

Цепочки данных сырого аудио или видео не имеют внешней видимой структуры. Мы можем просто выстроить их вместе в жёстком предопределённом порядке для стриминга или хранения, и многие простые системы делают примерно то же самое. С другой стороны, сжатые кадры не обязательно имеют предсказуемый размер, и нам часто может потребоваться некоторая гибкость в использовании данных ряда различных типов в потоках. Если мы расположим произвольные бесформенные данные вместе, мы потеряем границы, разделяющие кадры и не сможем при необходимости узнать, какие данные к какому потоку относятся. Поток необходима некоторая обобщенная структура, чтобы он был полезен.

В дополнение к нашим сигнальным данным, у нас также есть наши параметры ИКМ и видео. Также существует множество других метаданных, с которыми мы также хотим работать, таких как аудио теги и главы видео, субтитры — все естественные компоненты rich media. Имеет смысл располагать эти метаданные, другими словами данные о данных, внутри самого носителя.

Хранение и структурирование бесформенных данных и разнородных метаданных — это работа контейнера. Контейнеры предоставляют структурирование blob'ов (бинарных блоков) данных, чередование и идентификацию множества потоков данных, предоставляют информационно синхронизацию и хранят метаданные, необходимые для разбора, навигации, управления и представления данных. В общем, любой контейнер может содержать любой вид данных. И данные могут размещаться в любом контейнере.

За последние тридцать минут мы охватили цифровое аудио и видео, немного истории и математики и немного проектирования. Мы только начали, но пришло время для заслуженного перерыва. Есть ещё так много, о чем стоит поговорить, поэтому я надеюсь, что вы присоединитесь ко мне снова в нашем следующем эпизоде. До тех пор — пока!

Источник: <http://www.xiph.org/video/vid1.shtml>

[\[ОБСУДИТЬ НА ФОРУМЕ \]](#)

Мне нравится   Tweet  Recommend

Категория: [Обучающие видеоролики](#) | Добавил: [Audiophile](#) (2013-03-25T15:33) 

Просмотров: **2026**

Всего комментариев: **0**

Добавлять комментарии могут только зарегистрированные пользователи.
[\[Регистрация \]](#) [\[Вход \]](#)

© 2009-2014 Taras Kovrijenko

