

Н.В. МАЙСТРЕНКО, А.В. МАЙСТРЕНКО, И.Л. КОРОБОВА

МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ В САПР

Часть II

• ИЗДАТЕЛЬСТВО ТГТУ •

УДК 004(075)
ББК Ж2-5-05я73
М149

Рецензенты:

Доктор технических наук, профессор,
директор Тамбовского филиала Московского государственного
университета культуры и искусств
В.М. Тютюнник

Доктор технических наук, профессор,
заведующий кафедрой "Информационные системы и
защита информации" ТГТУ
Ю.Ю. Громов

М149 Майстренко, Н.В.
Мультимедийные технологии в САПР : учебное пособие /
Н.В. Майстренко, А.В. Майстренко, И.Л. Коробова. – Тамбов : Изд-во Тамб. гос. техн. ун-та, 2009. – Ч. II.
– 80 с. – 100 экз.
ISBN 978-5-8265-0803-9.

Излагаются начальные теоретические и практические понятия и сведения о представлении и обработке информации различных типов: текст, графика, звук, видео, анимация (в том числе принципах компрессии).

Предназначено для студентов 5 курса специальности 230104 "Системы автоматизированного проектирования", изучающих дисциплину "Мультимедийные технологии", но может быть полезно и для студентов, бакалавров и магистров других специальностей и направлений, аспирантов и преподавателей, знакомящихся с мультимедийными системами и принципами их разработки.

УДК 004(075)
ББК Ж2-5-05я73

ISBN 978-5-8265-0803-9 © ГОУ ВПО "Тамбовский государственный
технический университет" (ТГТУ), 2009
Министерство образования и науки Российской Федерации
ГОУ ВПО "Тамбовский государственный технический университет"

Н.В. Майстренко, А.В. Майстренко, И.Л. Коробова

МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ В САПР

Часть II

Утверждено Учёным советом университета
в качестве учебного пособия
для студентов 5 курса специальности 230104
"Системы автоматизированного проектирования"



Учебное издание

МАЙСТРЕНКО Наталья Владимировна,
МАЙСТРЕНКО Александр Владимирович,
КОРОБОВА Ирина Львовна

МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ В САПР

Часть II

Учебное пособие

Редактор З.Г. Чернова
Инженер по компьютерному макетированию М.Н. Рыжкова

Подписано в печать 23.03.2009.
Формат 60 × 84/16. 4,65 усл. печ. л. Тираж 100 экз. Заказ № 104

Издательско-полиграфический центр
Тамбовского государственного технического университета
392000, Тамбов, Советская, 106, к. 14

5. КОМПЬЮТЕРНЫЙ ЗВУК

Звук представляет собой колебания физической среды (обычно воздуха) частотой приблизительно 20...20 000 Гц, все современные системы обработки звука основаны на преобразовании этих колебаний в электрический сигнал, последующей его (аналоговой или цифровой) обработки и вывода вновь в виде колебаний физической среды. Эффект стереофонии достигается временной разницей колебаний, легко улавливаемой благодаря наличию приблизительно 20-сантиметровой базы между приёмниками аудиоинформации – ушами (разница порядка $7 \cdot 10^{-4}$ с).

В самом начале своей истории компьютер фирмы IBM был оснащён примитивным динамиком, позволявшим (посредством драйвера SPEAKER.DRV) воспроизводить звуки (одновременно) одного тона без регулировки уровня звука; именно в это время были разработаны основные принципы преобразования звука для бытовых компьютеров.

5.1. ИСТОРИЯ КОМПЬЮТЕРНОГО ЗВУКА

История развития компьютерного стереозвучания тесно связана с лидером в этой области – компанией Creative Labs. Первый опыт на рынке

PC-звука для этой тогда ещё мало кому известной азиатской фирмы был неудачным, хотя её изделие на то время (1988) представляло большой интерес и было достаточно оригинальным. Оно состояло из 12-голосого FM-синтезатора Creative Music System (CMS), аналого-цифрового и цифроаналогового преобразователей (АЦП и ЦАП); в комплект поставки входили программы собственной разработки для создания и редактирования музыки (CMS Composer, CMS Intelligent Organ и CMS Multimedia Presenter).

В то же самое время североамериканская фирма AdLib выпустила относительно дешёвую звуковую плату с FM (Frequency Modulation – частотная модуляция) синтезатором. В результате AdLib наравне с PC-спикером стал стандартом на звук для PC и получил поддержку у производителей игр.

В этой ситуации Creative выпустила недорогую звуковую карту на основе OPL2 от Yamaha, превосходящую AdLib по функциональным возможностям и совместимую с ней аппаратно. Так на свет появилась звуковая карта (а с ней и стандарт) Sound Blaster. В отличие от AdLib, который лишь воспроизводил MIDI-мелодии, звуковые карты Sound Blaster уже имели 8-битные АЦП и ЦАП. Затем выпускается несколько улучшенный вариант карты – Sound Blaster Pro – с полноценным стерео с частотой дискретизации до 22 кГц. В 1991 году Sound Blaster Pro фирмой Microsoft включается в спецификацию MPC (мультимедийный персональный компьютер).

Далее Creative выпускает 16-битную звуковую карту под названием Sound Blaster 16 с FM-синтезатором OPL3 и возможностью записи и воспроизведения цифрового звука с частотой дискретизации 44,1 кГц. В такт с расширением рынка растёт и множество вариаций на тему Sound Blaster 16 (звуковые карты SB 16, SB 16 Vibra, SB 16 ASP, SB 16 Value, SB 16 Pro, SB16 PnP, SB16 SCSI и т. п.).

В то же время начинают появляться WT (WaveTable) синтезаторы. Звуковые карты с таким синтезатором отличались более высоким качеством звука: если FM-синтезатор использовал модуляторы, то WT синтезировал звуки реальных музыкальных инструментов, образцы которых хранились в постоянном или оперативном запоминающем устройстве (ПЗУ, ОЗУ) платы или загружались в системное ОЗУ по мере необходимости с жёсткого диска; для некоторых FM-плат начали выпускаться дочерние платы с ПЗУ для инструментов (так называемые Wave Table daughter-board), которые подключались к основной через её специальный разъём Future или разъём MIDI-порта. Всё это заставляет Creative искать возможности технологического отрыва своих изделий от армии дешёвых азиатских подделок, в результате чего в 1993 году была лицензирована технология фирмы Emu – всемирно известного разработчика профессиональных музыкальных технологий. В результате слияния фирма Emu разработала аудио-процессор EMU8000 (EMU8K), который вывел на долгое время Creative в лидеры по качеству звучания MIDI-синтезатора и цифровой части. Новая карта на базе EMU8K получила название Sound Blaster AWE32 (Advanced Wave Effects – продвинутые волновые эффекты, 32 – количество голосов MIDI-синтезатора). При этом была обеспечена полная аппаратная совместимость с Sound Blaster 16 и даже улучшено звучание за счёт применения более совершенных АЦП и ЦАП.

Для противопоставления PCI звуковым картам в 1998 году Creative выпускает карту Sound Blaster Live! – нового семейства звуковых плат. Основой новой карты был звуковой процессор Emu 10K1 (Emu 10001), способный не только обрабатывать множество звуковых потоков, но и делать это по задаваемой извне программе. Два миллиона транзисторов и вычислительная мощность, эквивалентная 1000 миллионам операций обычного процессора в секунду, была вполне достаточной для обеспечения новой технологии 3D звука Emu Environmental Modeling. Новый процессор обеспечивал поддержку 2 – 8 звуковых колонок; обработку звука в соответствии со свойствами человеческого уха; трассировку звука (расчёт в реальном времени его взаимодействия с окружающими предметами в результате отражения, поглощения или искажения); эмуляцию АЗД (технология компании Aureal, главного конкурента Creative); одновременную обработку 128 независимых каналов; аппаратный 64-канальный WaveTable синтезатор с 8-точечной интерполяцией образца звучания инструмента (обычно

используется 2-точечная, у Aureal Vortex I-4-, в профессиональной аппаратуре – 6-точечная); обработку звука в 32-разрядном представлении; поддержку постраничной адресации памяти; банки инструментов в стандартном для всех изделий Emu Sound Font формате; 6 цифровых и 8 аналоговых входов; 8 цифровых выходов и др.

Одновременно появилась и технология EAX (Environmental Audio extensions) – API Creative, расширяющий возможности DS3D за счёт создания виртуальной звуковой среды окружения путём моделирования отражения звуков и реверберации, исходящих со всех сторон от слушателя. Волны отражённых звуков и реверберация, достигая слушателя, дают ему возможность составить представление о природе окружающей его среды: размерах помещения, отражающих свойств стен, расстояния до различных источников звука и многое другое. Разработчики могут использовать EAX при выборе установок различных свойств акустики для разных объектов; например, играя в игру, поддерживающую EAX, игрок может слышать, как изменяется звук при входе в пещеру.

В более совершенной версии EAX 2.0 предусмотрено расширенное управление акустикой окружающей среды: программист может изменять размеры помещения и манипулировать параметрами ранних отражённых звуков отдельно от затухающей реверберации с запаздыванием, что позволяет создавать реалистичные и полные модели широкого диапазона акустики окружающей среды, начиная от полуоткрытых пространств (например, городской двор, улица и т.д.) и заканчивая узким коридором или маленьким тесным кабинетом. В этой версии добавлены также эффекты окклюзии (occlusions – звуки, проходящие через препятствия), обструкции (obstruction – звук задерживается препятствием) и управления за ранними отражёнными звуками для каждого источника звука. Окклюзии применяются для моделирования источников звука, расположенных в другом помещении или в пространстве с другой стороны стены: например, если слушатель находится внутри дома, а источник звука – снаружи, то в игре или другом приложении свойство окклюзии используется для воспроизведения реалистичного звучания голоса или шума, раздающегося за дверью или вне дома. Обструкции позволяют моделировать дифракцию звуковых волн для создания ощущения, что источник звука находится в той же окружающей среде, что и слушатель, но закрыт от него преградой, например, большой колонной в том же помещении (доме).

Используемая в EAX модель распространения звука ray tracing (распространение лучей) аналогична модели распространения светового луча. Для реализации геометрической акустики используется компьютерная модель физического пространства с учётом расположения объектов, их звукоотражающих и звукопроводящих свойств. Расчёт слышимых пользователем звуков производится с учётом отражения и ослабления звука при прохождении через разнообразные преграды. При этом EAX использует метод статистического моделирования (вместо геометрического), что позволяет более эффективно использовать ресурсы центрального процессора и моделировать ранние отражённые звуки и реверберацию с запаздыванием, обеспечивая реалистичное воспроизведение глубины акустической сцены; статистическая модель EAX автоматически вычисляет параметры реверберации и отражённых звуков в зависимости от расположения слушателя относительно источников звука, размеров помещения, направленности источников звука и от дополнительного набора параметров, которые может изменять программист.

Следует заметить, что ещё за шесть лет до появления Emu 10K1 аэрокосмическая фирма NASA использовала в своих тренажёрах технологию небольшой исследовательской фирмы Aureal, которая вышла затем на рынок PC с технологией A3D создания трёхмерного звука с помощью двух или более колонок. Основой этой технологии являются так называемые HRTF (Head Related Transfer Functions – функции перемещения звука относительно его приёмника (головы слушателя)). Количественно HRTF определяются обратным интегральным Фурье преобразованием коэффициентов под названием HRIR (Head Related Impulse Response), которые в первом приближении определяются отношением давлений на барабанную перепонку уха звуковой волны в свободном пространстве (free field) и в реальном пространстве с учётом головы человека, ушных раковин, его корпуса и других препятствий.

HRTF представляет собой необычайно сложную функцию с четырьмя переменными: три пространственных координаты и частота. При использовании сферических координат для определения расстояния до источников звука больших, чем один метр, принимается, что источники звука находятся в дальнем поле (far field), значение HRTF уменьшается обратно пропорционально расстоянию. Большинство измерений HRTF производится именно в дальнем поле, при этом количество переменных уменьшается до трёх: азимут (azimuth), высота (elevation) и частота (frequency).

Одним из наиболее эффективных способов определения параметров HRTF является метод моделирования с использованием манекена под названием KEMAR (Knowless Electronics Manikin for Auditory Research) и специального "цифрового уха" (digital ear), разработанного компанией Sensaura и располагаемого на голове манекена. В ушах манекена размещаются микрофоны, а вокруг манекена – акустические колонки с записываемыми звуками, в результате происходит запись того, что слышит каждое "ухо". Получаемые при таком моделировании результаты используются для пополнения базы данных по HRTF, которые затем могут быть использованы для интерактивного выбора параметров при воспроизведении позиционируемого 3D звука (например, в базе данных компании Sensaura накоплено более 1000 HRTF). Необходимость в такой базе данных объясняется, во-первых, различием размеров и формы головы и ушных раковин манекена и потенциального слушателя и, во-вторых, определяемых этими параметрами так называемой зоны sweet spot (как у сканера), в которой корректно воссоздаётся эффект звучания в вертикальной плоскости и гарантируется правильное определе-

ние местоположения источников звука в пространстве. Заметим, что известная компания QSound при реализации технологий с HRTF опирается не только на математические методы, но и на прослушивание конкретными людьми (таких прослушиваний было проведено более полумиллиона).

Практической реализацией технологии А3D стал чип Aureal Vortex I (8820). Этот аудиопроцессор обеспечивает около 300 миллионов операций в секунду; обработку 48 независимых потоков с отдельными эффектами; плавную подстройку скорости воспроизведения; интерполяцию музыкальных образцов (набора семплов) инструментов по четырём точкам, два генератора огибающих, хорус, реверберацию и частотный фильтр на каждый канал; адресацию в ОЗУ ПК непрерывной области памяти объёмом 10 Мбайт; 32 канала для MIDI-синтезатора и 16 – для ускорения DirectSound; эмуляцию SoundBlaster в DOS'e (например, для DOS-игр).

Аппаратная поддержка более совершенной технологии А3D 2.0 осуществляется с помощью процессора Aureal Vortex 2 (8830) – прямого конкурента Emu 10K1, – обеспечивающего 800 миллионов операций в секунду; 96 аппаратных потоков одновременно; поддержку более сложных HRTF; трассировку звука; аппаратный десятиполосный стереоэквалайзер с отношением сигнал/шум 96 дБ; работу с 1, 2 или 4 кодеками (от 2 до 8 колонок); поддержку до 64 потоков для DirectSound и простых приложений Windows; 64-канальный MIDI-синтезатор; цифровой выход и др.

В процессе создания А3D 2.0 компанией Aureal была разработана новая технология Wavetracing, основанная на расчёте распространения отражённых и прошедших через препятствия звуковых волн на основе геометрии среды. Технология состоит из трёх компонент: интерфейс А3D (используется для расчёта прямых путей распространения звука (direct path)); геометрический движок (geometry engine), определяющий геометрию объектов в пространстве и производящий расчёт отражённых и прошедших сквозь препятствия звуковых волн; менеджер сцены (scene manager), который используется как геометрическим движком, так и интерфейсом А3D для управления сложными звуковыми сценами. Примером эффективного использования технологии Wavetracing может служить звуковое сопровождение полёта из открытого пространства в туннель и обратно на открытое пространство с несколькими высотными зданиями вокруг: отражённые звуки будут впечатляюще изменяться от полного их отсутствия на открытом пространстве до коротких эхо в туннеле и затем до нескольких отчётливых звуков, отражённых при пролёте мимо зданий. Для реализации приведённого примера необходимо составить базу данных графической геометрии пространства, конвертировать её в звуковую базу данных с учётом толщины стен и материала, заранее подготовить уровни эха (reverb) для всего набора акустических препятствий и др.

Как говорилось выше, большинство измерений HRTF производится в так называемом дальнем поле (far field), что существенным образом упрощает вычисления. Если же источники звука располагаются на расстоянии до одного метра от слушателя (в так называемом ближнем поле – near field), то рассчитанные ранее функции HRTF становятся недостаточно эффективными. Именно для воспроизведения звука от источников в ближнем поле с помощью HRTF-функций и создана технология MacroFX, которая позволяет создавать массу новых эффектов.

Компания EAR (Extreme Audio Reality, Inc.) в результате сотрудничества с разработчиками и производителями аппаратного обеспечения разработала технологию Interactive Around-Sound (IAS) для воспроизведения 3D звука на всех доступных платформах. Технология IAS получила образное определение write once, run anywhere (написав один раз, запускай везде), обеспечивая трёхмерный звук при любом аппаратном обеспечении и минимально возможной загрузке центрального процессора.

Поскольку, как видно из предыдущего материала, проблемами цифрового 3D звука занимаются достаточно много фирм, то для координации работ было создано объединение IASIG (Interactive Audio Special Interest Group – группа особых интересов по интерактивному звуку). IASIG является открытым объединением создателей технологий, производителей и разработчиков, в числе которых такие компании, как QSound, Creative, Aureal и многие другие. В настоящее время IASIG разрабатывает на основе EAX новую спецификацию. Выбор EAX вызван его сравнительной простотой и потенциальными возможностями более широкой аппаратной поддержки, чем, например, технология Wavetracing от Aureal.

5.2. МЕТОДЫ СИНТЕЗА КОМПЬЮТЕРНОГО ЗВУКА

Запись произвольного звука осуществляется путём прямой оцифровки аналогового сигнала, представляющего собой электрическую копию звукового давления (преобразователем является датчик звукового давления – микрофон). Частота оцифровки (частота преобразования) называется частотой выборки сигнала и по известной теореме Котельникова-Найквиста должна быть не ниже удвоенного значения максимальной частоты преобразуемого сигнала (например, если спецификация MPC Level 1 определяет частоту преобразования 11 кГц, то верхний предел записываемой частоты составляет около 5 кГц).

Преобразование аналогового сигнала в цифровую форму выполняет аналого-цифровой преобразователь (АЦП), служащий для дискретизации сигнала по времени (частота оцифровки) и квантования по уровню (собственно цифровое представление сигнала). Обычно в АЦП применяется технология преобразования с импульсно-кодовой модуляцией (PCM, *Pulse Code Modulation*). Временные промежутки между моментами преобразования сигнала называют интервалами выборки (*Sampling Interval*); эта величина обратно пропорциональна час-

тоте выборки, или сэмпингом (*Sampling Rate*). Амплитуда аналогового сигнала (*Sample Value*) при каждом преобразовании делится (квантуется) по уровню и кодируется в соответствующий параллельный цифровой код (*Digital Sample*), время преобразования аналогового сигнала в цифровой код именуется временем выборки (*Sampling Time*), рис. 5.1.

Разрешающей способностью АЦП называется наименьшее значение аналогового сигнала, которое приводит к изменению цифрового кода. Например, если АЦП выдает 8-разрядный код, разрешающая способность равна $1/(2^8) = 1/256$ от максимальной амплитуды аналогового сигнала (около 0,4 % в относительных единицах), 16-разрядный АЦП имеет точность представления сигнала не хуже $1/(2^{16}) = 1/65\,536$ (0,0015 %).

С увеличением разрядности АЦП растёт его динамический диапазон (каждый дополнительный бит соответствует увеличению приблизительно на 6 дБ). 8-разрядное преобразование обеспечивает динамический диапазон 48 дБ (качество кассетного магнитофона), 12-разрядное – 72 дБ (качественный катушечный магнитофон), 16-разрядное – 96 дБ (качество аудио компакт-диска).

Полученный с АЦП параллельный код разрядностью 8...16 последовательно (побитно) записывается с частотой сэмпинга в аудиофайл (при необходимости используется буферизация), при этом поток несжатых цифровых аудиоданных велик.

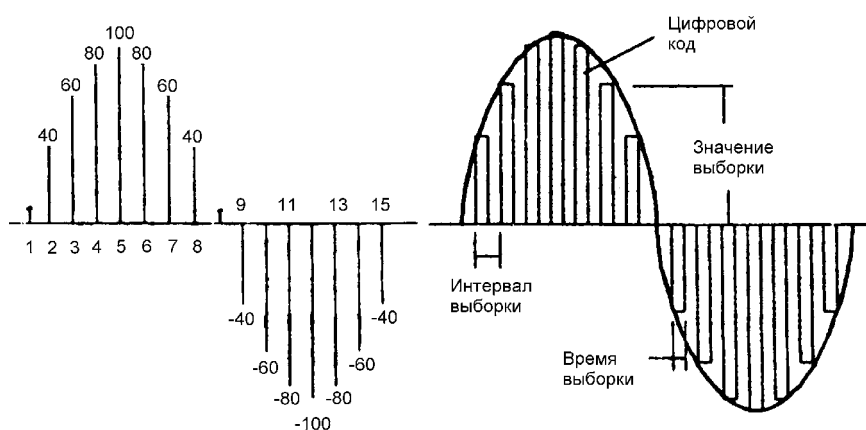


Рис. 5.1. Характеристики процесса преобразования между аналоговым и цифровым сигналами

В студийной работе происходит переход на стандарт 96 кГц/24 бита, который по теоретически достижимому качеству пока заметно перекрывает возможности существующих звуковых систем.

Заметим, что АЦП (также как и ЦАП – цифро-аналоговые преобразователи) выполняют свои функции аппаратно, не загружая ЦП (центральный процессор); последний управляет только режимами работы АЦП и ЦАП.

Для синтеза звука используются следующие методы:

Аддитивный (additive) – суммирование синусоидальных колебаний с различными частотами и амплитудами, для чего используется набор из нескольких синусоидальных генераторов с независимым управлением, выходные сигналы которых суммируются для получения результирующего сигнала. На этом методе основан принцип создания звука в духовом органе. Основным недостатком метода является необходимость в большом количестве независимых генераторов.

Разностный (subtractive) – генерация звукового сигнала сложной формы (со множеством гармоник) с последующей фильтрацией (по этому принципу работает речевой аппарат человека). В качестве исходных сигналов обычно используются меандр – прямоугольный (square) сигнал с переменной скважностью, пилообразный (saw) и треугольный (triangle), а также различные виды шумов (случайные непериодические колебания). Основным устройством при реализации этого метода являются управляемые фильтры: резонансный или полосовой с изменяемым положением и шириной полосы пропускания и фильтр нижних частот с регулируемой частотой среза. Для каждого фильтра также регулируется добротность (крутизна подъёма или спада резонансной кривой). Достоинства метода – сравнительно простая аппаратная реализация и достаточно широкий диапазон синтезируемых звуков. На этом методе построено множество студийных и концертных синтезаторов. Недостаток – необходимость в большом количестве управляемых фильтров.

Для компьютерного синтеза звука наиболее часто применяют *цифровой FM-синтез*, основы которого заложены в конце 70-х годов XX века студентом Стенфордского университета Джоном Чоунингом (*John Chowning*). Несколько десятилетий ранее Роберт Муг (*Robert Moog*) реализовал в серии своих всемирно известных синтезаторов аналоговый вариант

FM-синтеза путём использования генераторов огибающей, управляющих амплитудой отдельных VCO-генераторов (*Voltage-Controlled Oscillator*).

В цифровом FM-синтезе каждый из описанных управляемых генераторов называется *оператором*. В операторе выявляются два базовых элемента: *фазовый модулятор* и *генератор огибающей*. Фазовый модулятор задаёт частоту (высоту) звука, а генератор огибающей – его амплитуду (громкость); см. общую схему рис. 5.2.

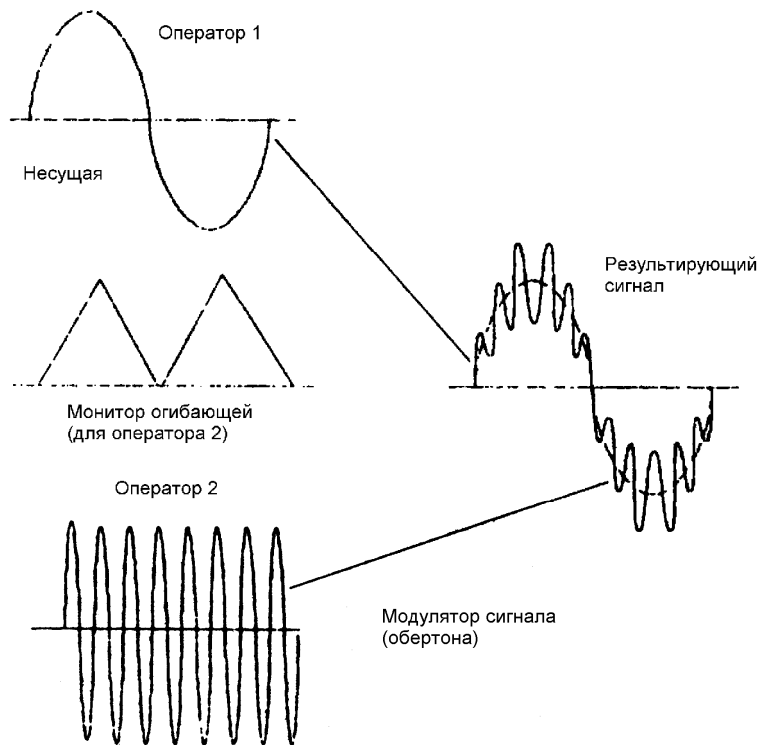


Рис. 5.2. Генерация сигналов с заданной огибающей при получении звука посредством FM-синтеза

В большинстве случаев для синтеза одного инструмента достаточно двух операторов – оператора несущей (основной тон) и оператора модулирующей частоты (*обертон*). Обычно пара операторов определяет *голос*; современные наборы микросхем для FM-синтеза звука содержат до 36 – 40 голосов, осуществляя различные режимы (алгоритмы) FM-синтеза (в том числе и самые сложные, предполагающие использовать 18 и более операторов для синтеза речи). В звуковых картах обычно присутствует специальный *генератор шума*, обрабатываемый одним оператором (оператором огибающей).

Сэмплерный (sample – выборка). При этом методе записывается реальное звучание инструмента (сэмпл), которое затем воспроизводится с ускорением или замедлением; при неизменной скорости выборки применяется расчёт промежуточных значений отсчётов путём интерполяции. Чтобы тембр звука при сдвиге частоты не менялся слишком сильно, используется несколько записей звучания через определённые интервалы (обычно через одну-две октавы). В ранних сэмплерных синтезаторах звуки записывались на магнитофон, в современных применяется цифровая запись. Метод позволяет получить сколь угодно точное подобие звучания реального инструмента, однако для этого требуются большие объёмы памяти. С другой стороны, запись звучит естественно только при тех же параметрах, при которых она была сделана; при попытке, например, придать ей другую амплитудную огибающую естественность резко ухудшается. Для уменьшения требуемого объёма памяти применяется закичивание сэмпла (looping). В этом случае записывается звучание инструмента за короткие промежутки времени, затем в нём выделяется средняя фаза с установившимся (sustained) звуком, которая при воспроизведении повторяется до тех пор, пока включена нота (нажата клавиша), а после отпущения воспроизводится концевая фаза. На самом деле этот метод нельзя с полным правом называть синтезом; это, скорее, метод записи-воспроизведения. Однако в современных синтезаторах на его основе воспроизводимый звук можно подвергать различной обработке: модуляции, фильтрации, добавлению новых гармоник, звуковых эффектов, в результате чего звук может приобретать совершенно новый тембр, иногда совсем непохожий на первоначальный. По сути, получается комбинация трёх основных методов синтеза, где в качестве основного сигнала используется исходное звучание. Типичным представителем этого класса синтезаторов является Emu Proteus.

Таблично-волновой (WT – wave table) – разновидность сэмплерного метода, при котором записывается не всё звучание целиком, а его отдельные фазы: атака, начальное затухание, средняя фаза и концевое затухание, что позволяет резко уменьшить объём памяти, требуемый для хранения сэмплов. Эти фазы записываются на

различных частотах и при различных условиях (мягкий или резкий удар по клавише рояля, различное положение губ и языка при игре на саксофоне и т.п.), в результате чего получается семейство звучаний одного инструмента. При воспроизведении эти фазы нужным образом комбинируются, что даёт возможность при относительно небольшом объёме сэмплов получить достаточно широкий спектр различных звучаний инструмента, а главное – заметно усилить выразительность звучания, выбирая, например, в зависимости от силы удара по клавише синтезатора не только нужную амплитудную огибающую, как делает любой синтезатор, но и нужную фазу атаки. Основная проблема этого метода – в сложности сопряжения различных фаз друг с другом, чтобы переходы не воспринимались на слух и звучание было цельным и непрерывным. Поэтому синтезаторы этого класса достаточно редки и дороги. Этот метод также используется в синтезаторах звуковых карт ПК, однако его возможность там сильно урезаны. В частности, почти не применяется составление звука из нескольких фаз и WT сводится к простому сэмплерному методу, хотя почти везде есть возможность параллельного воспроизведения более одного сэмпла внутри одной ноты. Большинство звуковых плат содержит встроенный набор инструментов в ПЗУ, некоторые платы позволяют дополнительно загружать собственные инструменты в ОЗУ платы, а платы семейства GUS (кроме GUS РпР) содержат только ОЗУ и набор стандартных инструментов на диске. Некоторые модели РСІ-плат позволяют использовать для загрузки инструментов общее ОЗУ компьютера (UMA – Unified Memory Architecture, унифицированная архитектура памяти).

Метод физического моделирования (physical modelling) состоит в моделировании физических процессов, определяющих звучание реального инструмента на основе заданных параметров (например, для скрипки: порода дерева, состав лака, геометрические размеры, материал струн, смычка и т.п.). В связи с крайней сложностью точного моделирования даже простых инструментов и огромным объёмом вычислений метод находит применение на уровне студийных и экспериментальных образцов синтезаторов.

Линейный арифметический синтез (linear arithmetic synthesis) работает следующим образом – берётся короткая сэмплированная волновая форма, называемая РСМ или Pulse Code Modulation, и комбинируется с синтезированным звуком, который формирует основу нового звука. Путём наложения и комбинирования этого звука с синтезированными отрывками и получаете новый звук, который затем, естественно, проходит через цепочку фильтров, генераторов огибающих и т.д. Это один из самых общих принципов синтеза использовался в 90-х XX столетия и по сей день.

Позволяет синтезировать довольно простым способом большое количество различных тембров.

Гранулярный синтез (granular synthesis) оперирует с небольшими кусочками волноформ (гранулами), чтобы сформировать новый звук. Используя вариации частот, фазовых сдвигов и амплитуд составных компонентов и составляя меняющиеся последовательности гранул и их продолжительность можно в итоге получить что-либо подходящее.

Улучшенный векторный синтез (advanced vector synthesis – AVS) заключается в комбинировании и обработке цифровых волновых форм. Используя РСМ (Pulse Code Modulation) сэмплы, эффекты и фильтрацию этот алгоритм может создать потрясающие звуки, начиная от меняющихся во времени пэдов (Pads) и до странных прерывающихся секвенций.

5.3. МЕТОДЫ СЖАТИЯ АУДИОДАНЫХ И ФОРМАТЫ АУДИОФАЙЛОВ

Различают следующие форматы звуковых файлов, подразумевающие соответствующие методы сжатия информации:

- форматы аудиоданных без потерь;
- форматы аудиоданных с потерями:
 - использующие простые технологии;
 - использующие психоакустику и сжатие спектра;
 - использующие технологии кодирования голоса (vocoder);
- форматы аудиоданных нотной записи;
- форматы, использующие ноты и образцы инструментов;
- форматы-контейнеры.

Далее рассмотрим некоторые примеры форматов звуковых файлов.

5.3.1. Форматы аудиоданных без потерь

"Без потерь" означает, что звуковые данные/качество не будут потеряны при сжатии и после распаковки не будут отличаться от оригинальных.

Monkey's Audio – популярный формат кодирования цифрового звука без потерь. Распространяется бесплатно вместе с открытым исходным кодом и набором программного обеспечения для кодирования и воспроизведения, а также плагинами к популярным плеерам. Файлы Monkey's Audio используют следующие расширения: .ape для хранения аудио и .ar1 для хранения метаданных. Несмотря на открытый исходный код, Monkey's Audio не является свободным, так как его лицензия накладывает значительные ограничения на использование.

Официально кодек Monkey's Audio выпускается только для платформы Microsoft Windows, хотя существует ряд неофициальных кодеков для GNU/Linux, BeOS и Mac OS X, которые в большинстве случаев позволяют лишь пережимать файлы из этого формата в какой-либо другой.

FLAC (англ. Free Lossless Audio Codec – свободный аудио-кодек без потерь) – популярный свободный кодек для сжатия аудио. В отличие от кодеков с потерями Ogg Vorbis, MP3 и AAC, не удаляет никакой информации из аудиопотока и подходит как для прослушивания музыки на высококачественной звуковоспроизводящей аппаратуре, так и для архивирования аудиокolleкции. На сегодня формат FLAC поддерживается многими аудиоприложениями.

Первые четыре байта аудиофайла идентифицируют поток FLAC. Следующие за ними метаданные содержат информацию о потоке, затем идут сжатые аудиоданные.

FLAC определяет несколько типов блоков метаданных. Блоки метаданных могут быть любого размера, новые блоки могут быть легко добавлены. Декодер имеет возможность пропускать неизвестные ему блоки метаданных. Обязателен только блок STREAMINFO. В нём содержится частота дискретизации, количество каналов и т.п., а также данные, позволяющие декодеру настроить буферы. Сюда также записывается подпись MD5 несжатых аудиоданных. Это полезно для проверки всего потока после его передачи.

Другие блоки предназначены для резервирования места, хранения таблиц точек поиска, тегов, список разметки аудиодисков, а также данных для конкретных приложений. Опции для добавления блоков PADDING или точек поиска приведены ниже. FLAC не нуждается в точках поиска, однако они позволяют значительно увеличить скорость доступа, а также могут быть использованы для расстановки меток в аудио редакторах.

За метаданными следуют сжатые аудиоданные. Метаданные и аудиоданные не чередуются. Как и большинство кодеков FLAC делит входной поток на блоки и кодирует их независимо друг от друга. Блок упаковывается во фрейм и добавляется к потоку. Базовый кодер использует блоки постоянного размера для всего потока, однако формат предусматривает наличие блоков разной длины в потоке.

Размер блока – очень важный параметр для кодирования. Если он очень мал, то в потоке будет слишком много заголовков фреймов, что уменьшит уровень сжатия. Если размер большой, то кодер не сможет подобрать эффективную модель сжатия. Понимание процесса моделирования поможет увеличить уровень сжатия для некоторых типов входных данных. Обычно при использовании линейного прогнозирования на аудиоданных с частотой дискретизации 44,1 кГц оптимальный размер блока лежит в диапазоне 2 – 6 тысяч сэмплов.

На следующем этапе кодер пытается аппроксимировать сигнал такой функцией, чтобы полученный после её вычитания из оригинала результат (называемый разностью, остатком, ошибкой) можно было закодировать минимальным количеством битов. Параметры функций тоже должны записываться, поэтому они не должны занимать много места. FLAC использует два метода формирования аппроксимаций:

- подгонка простого полинома к сигналу;
- общее кодирование с линейными предикторами (LPC).

Во-первых, постоянное полиномиальное предсказание работает значительно быстрее, но менее точно, чем LPC. Чем выше порядок LPC, тем медленнее, но лучше будет модель. Однако с увеличением порядка выигрыш будет всё менее значительным. В некоторой точке (обычно около 9) процедура кодера, определяющая наилучший порядок, начинает ошибаться и размер получаемых фреймов возрастает. Чтобы преодолеть это, можно использовать полный перебор, что приведёт к значительному увеличению времени кодирования.

Во-вторых, параметры для постоянных предикторов могут быть описаны тремя битами, а параметры для модели LPC зависят от количества бит на сэмпл и порядка LPC. Это значит, что размер заголовка фрейма зависит от выбранного метода и порядка и может повлиять на оптимальный размер блока.

Когда модель подобрана, кодер вычитает приближение из оригинала, чтобы получить остаточный (ошибочный) сигнал, который затем кодируется без потерь. Для этого используется то обстоятельство, что разностный сигнал обычно имеет распределение Лапласа и есть набор специальных кодов Хаффмана, называемый кодами Райса, позволяющий эффективно и быстро кодировать эти сигналы без использования словаря.

Кодирование Райса состоит из нахождения одного параметра, отвечающего распределению сигнала, а затем использования его для составления кодов. При изменении распределения меняется и оптимальный параметр, поэтому имеется метод, позволяющий пересчитывать его по необходимости. Остаток может быть разбит на контексты или разделы, у каждого из которых будет свой параметр Райса. flac позволяет указать, как нужно производить разбиение. Остаток может быть разбит на 2^n раздела.

True Audio (сокращённо **TAA**) – это свободный и бесплатный, аудио- кодек, осуществляющий сжатие аудиофайлов без потерь, способный работать в режиме реального времени. Кодек основан на адаптивных предсказывающих фильтрах и обладает такими же или лучшими характеристиками, как и большинство современных кодер/декодеров без потерь.

TAK – (T)om's verlustfreier (A)udio(k)ompressor – аудиокодек, формат сжатия цифрового звука без потерь. Отличается высокой степенью сжатия и скоростью кодирования и декодирования. Распространяется бесплатно вместе с набором программного обеспечения для кодирования и воспроизведения, а также плагинами к популярным плеерам: Winamp, foobar2000 и др.

Относительно новый кодек. Первая финальная версия 1.0 была опубликована 26 января 2007 года.

Формат продолжает активно развиваться (последняя версия 1.0.4) и в настоящее время, согласно проводимому опросу на форуме hydrogenaudio.org, входит в число трёх наиболее популярных форматов аудиосжатия без потерь (после FLAC и WavPack).

К особенностям данного кодека следует отнести:

- высокая скорость кодирования (TAK упаковывает аудиоданные с такой же скоростью, как FLAC-8x, в режиме максимальной компрессии ("Insane") и в несколько раз быстрее в режиме "Turbo");
- высокая скорость декодирования (сопоставимая с FLAC/WavPack);
- очень высокая степень сжатия (на уровне Monkey's Audio в режиме "High");
- защита от ошибок кодирования;
- быстрый поиск;
- поддержка потокового вещания.

WavPack – бесплатный аудиокодек с открытым исходным кодом для сжатия аудио без потери качества.

WavPack-формат (расширение .WV) позволяет сжимать (и восстанавливать) 8-, 16-, 24- и 32-битные аудиофайлы в .WAV формате. Он также поддерживает потоки звук вокруг и высокие частоты дискретизации (sampling rate). Как у других способов компрессии без потери качества, эффективность сжатия зависит от исходных данных, но обычно она лежит в диапазоне между 30 и 70 % для обычной популярной музыки, немного выше для классической музыки и других источников с более широким динамическим диапазоном.

WavPack также включает уникальный "гибридный" режим, который предоставляет все преимущества сжатия без потерь с дополнительным бонусом: вместо создания одного файла, в этом режиме создается относительно небольшой файл высокого (точнее, указанного при кодировании) качества с потерей (.WV), который может проигрываться сам по себе, а также файл "коррекции" (.WVC), который (в комбинации с предыдущим .WV) позволяет полностью восстановить оригинал. Для некоторых пользователей это означает, что им никогда не придётся выбирать между сжатием без потерь и с потерей качества.

5.3.2. Сжатие аудиоданных с потерями

Все распространенные потоковые форматы сжатия (MP3, AC3, WMA, OGG) основаны на схожем принципе работы, состоящем из трёх основных этапов.

Первый этап – быстрое преобразование Фурье (FFT) исходного сигнала (фрейма, так как форматы потоковые). Кратко FFT – это процесс, представляющий исходный сигнал в виде суммы синусоид:

$$F(t) = A_1 \sin(\lambda_1 t) + \dots + A_n \sin(\lambda_n t) + \dots$$

Теперь, вместо того чтобы хранить информацию о величине амплитуды волны в каждом сэмпле, остаётся запомнить только значения амплитуд (A_i) и длин волн (λ_i). Обратное преобразование Фурье для реальных звуков без потери качества невозможно.

Второй этап – психоакустическая обработка, призванная вычистить из звукового потока информацию, не воспринимаемую человеческим ухом.

И, наконец, третий этап – применение математических алгоритмов сжатия. Во время этой операции происходят только численные преобразования, позволяющие представить информацию в более компактном виде. В MP3, например, используется чуть-чуть доработанный алгоритм Хаффмана.

Алгоритм FFT известен сравнительно давно, поэтому разработчики совершенствуют методики сжатия за счёт оптимизации математического и психоакустических алгоритмов кодирования. Если математический алгоритм в каждом формате свой, то основные принципы действия психоакустического алгоритма сжатия схожи и заимствуют общие идеи у неизвестного формата MPEG-1 Layer II, разработанного в 1992 году Moving Picture Experts Group.

Опишем основные положения психоакустики, применяющейся в алгоритмах сжатия звука с потерями.

5.3.2.1. Психоакустика

Психоакустика – наука, изучающая психологические и физиологические особенности восприятия звука человеком.

Во многих приложениях акустики и обработки звуковых сигналов необходимо знать, что люди слышат. Звук, который образуют волны давления воздуха, может быть точно измерен современным оборудованием. Однако понять, как эти волны принимаются и отображаются в нашем головном мозге, – задача не такая простая. Звук – это непрерывный аналоговый сигнал, который (в предположении, что молекулы воздуха бесконечно малы) может теоретически переносить бесконечное количество информации (может быть бесконечное число частот, содержащих информацию об амплитуде и фазе).

Понимание процессов восприятия позволит учёным и инженерам сосредоточиться на возможностях слуха и не учитывать менее важные возможности других систем. Важно также отметить, что вопрос "что человек слышит" – не только вопрос о физиологических возможностях уха, но во многом также вопрос психологии, чёткости восприятия.

Человеческое ухо номинально слышит звуки в диапазоне 16...20 000 Гц. Верхний предел имеет тенденцию снижаться с возрастом. Большинство взрослых людей не могут слышать выше 16 кГц. Ухо само по себе не

реагирует на частоты ниже 20 Гц, но они могут ощущаться через органы осязания. Частотное разрешение звука находится в середине диапазона около 2 Гц, т.е. изменение частоты более чем на 2 Гц ощущается. Однако есть возможность слышать ещё меньшую разницу. Например, в случае, если оба тона приходят одновременно, в результате сложения двух колебаний возникает модуляция амплитуды сигнала с частотой, равной разности исходных частот. Этот эффект известен также как биение.

Диапазон громкости воспринимаемых звуков огромен. Наша барабанная перепонка в ухе чувствительна только к изменению давления.

Громкость звука принято измерять в децибелах (дБ). Нижний порог слышимости определён как 0 дБ, а определение верхнего предела слышимости относится скорее к вопросу, при какой громкости начнётся разрушение уха. Ухо способно переносить кратковременное повышение громкости до 120 дБ без последствий, но долговременное восприятие звуков громкостью более 80 дБ может вызвать потерю слуха.

Более тщательные исследования нижней границы слуха показали, что минимальный порог, при котором звук остаётся слышен, зависит от частоты. Этот график получил название абсолютного порога слышимости. В среднем, он имеет участок наибольшей чувствительности в диапазоне 1...5 кГц, хотя с возрастом чувствительность понижается.

Кривая абсолютного порога слышимости является частным случаем более общих – кривых одинаковой громкости. Кривые одинаковой громкости – это линии, на которых человек ощущает звук разных частот одинаково громкими. Кривые были впервые получены Флетчером и Мэнсоном (H Fletcher and W A Munson), и опубликованы в труде "Loudness, its definition, measurement and calculation" в J. Acoust. Soc Am.5, 82-108 (1933). Позже более точные измерения выполнили Робинсон и Датсон. Полученные кривые значительно различаются, но это не ошибка, а разные условия проведения измерений. Флетчер и Мэнсон в качестве источника звуковых волн использовали наушники, а Робинсон и Датсон – фронтально расположенный динамик в безэховой комнате.

Ещё одно важное свойство человеческого слуха – неравномерность распределения границы слышимости звука по частотам. Наилучшим образом мы слышим частоты в районе 2...4 кГц (не случайно речевой диапазон находится примерно в этой же области), к низким и высоким частотам чувствительность уха снижается. Таким образом, чем дальше частота слышимого звука от 2...4 кГц, тем выше граница слышимого звука (рис. 5.3), тем больше информации можно вырезать без заметных потерь в качестве.

Измерения Робинсона и Датсона легли в основу стандарта ISO 226 в 1986 году. В 2003 году стандарт ISO 226 был обновлён с учётом данных, собранных из 12 международных студий.

Существует также способ восприятия звука без участия барабанной перепонки – так называемый микро-волновый слуховой эффект, когда модулированное излучение в микроволновом диапазоне (1...300 ГГц) воздействует на ткани вокруг улитки, заставляя человека воспринимать различные звуки.

Человеческий слух во многом подобен спектральному анализатору, т.е. ухо распознаёт спектральный состав звуковых волн без анализа фазы волны. В реальности фазовая информация распознаётся и очень важна для направленного восприятия звука, но эту функцию выполняют ответственные за обработку звука отделы головного мозга. Разница между фазами звуковых волн, приходящих на правое и левое ухо, позволяет определять направление на источник звука, причём информация о разности фаз имеет первостепенное значение, в отличие от изменения громкости звука воспринимаемого разными ушами. Эффект фильтрации передаточных функций головы также играет в этом важную роль.

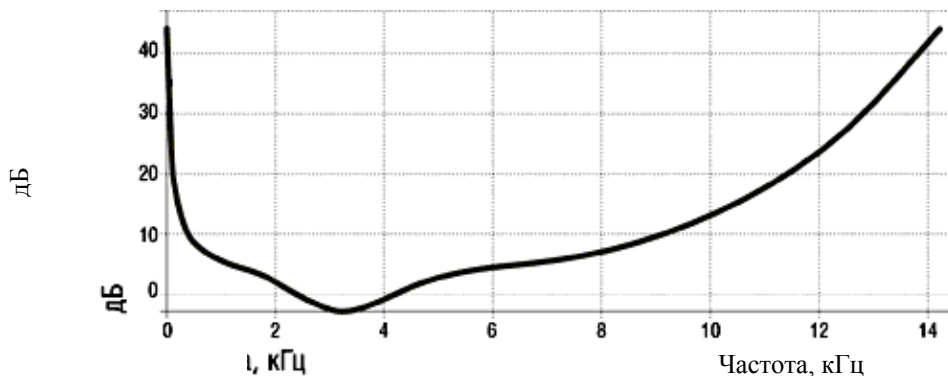


Рис. 5.3. Граница слышимости в тишине

Эффект маскировки. В определённых случаях один звук может быть скрыт другим звуком. Например, разговор на автобусной остановке может быть совершенно невозможен, если подъезжает шумный автобус. Этот эффект называется маскировкой. Говорят, что слабый звук маскируется, если он становится неразличим в присутствии более громкого звука.

Различают несколько видов маскировки:

- По времени прихода маскирующего и маскируемого звука:
 - одновременное (моноуральное) маскирование;
 - временное (неодновременное) маскирование.
- По типу маскирующего и маскируемого звуков:
 - чистого тона чистым тоном различной частоты (рис. 5.4);
 - чистого тона шумом;
 - речи чистыми тонами;
 - речи монотонным шумом;
 - речи импульсными звуками и т.п.

Частотная маскировка. Любой слышимый тон изменяет восприятие остальной звуковой картины. При воспроизведении какого бы то ни было тона граница слышимости соседних с ним по частотам звуков изменяется.

В этом случае воспроизводимый тон называется маскирующим, а граница слышимости окружающих его тонов поднимается тем выше, чем ближе их частота к частоте маскирующего сигнала.

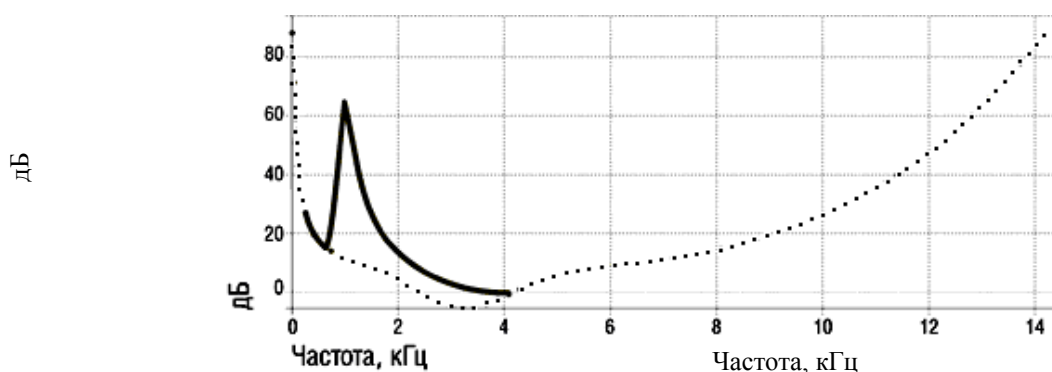


Рис. 5.4. Граница слышимости под воздействием тона частотой 1 кГц и интенсивностью 60 дБ

Одновременная маскировка. Любые два звука при одновременном прослушивании оказывают влияние на восприятие относительной громкости между ними. Более громкий звук снижает восприятие более слабого, вплоть до исчезновения его слышимости. Чем ближе частота маскируемого звука к частоте маскирующего, тем сильнее он будет скрываться. Эффект маскировки не одинаков при смещении маскируемого звука ниже или выше по частоте относительно маскирующего. Более низкочастотный звук сильнее маскирует высокочастотный.

Временная маскировка. Это явление похоже на частотную маскировку, но здесь происходит маскировка во времени. При прекращении подачи маскирующего звука маскируемый некоторое время продолжает быть не слышимым (рис. 5.5). В обычных условиях эффект от временной маскировки длится значительно меньше. Время маскировки зависит от частоты и амплитуды сигнала и может достигать 100 мс. В случае, когда маскирующий тон появляется по времени раньше маскируемого, эффект называют пост-маскировкой. Когда маскирующий тон появляется позже маскируемого (возможен и такой случай), эффект называют пре-маскировкой.

Постстимульное утомление. Нередко после воздействия громких звуков высокой интенсивности у человека резко снижается слуховая чувствительность. Восстановление обычных порогов может продолжаться до 16 ч. Этот процесс называется "временный сдвиг порога слуховой чувствительности" или "постстимульное утомление". Сдвиг порога начинает появляться при уровне звукового давления выше 75 дБ и соответственно увеличивается при повышении уровня сигнала. Причём наибольшее влияние на сдвиг порога чувствительности оказывают высокочастотные составляющие сигнала.

Фантомы. Иногда человек может слышать звуки в низкочастотной области, хотя в реальности звуков такой частоты не было. Так происходит из-за того, что колебания базилярной мембраны в ухе не являются линейными и в ней могут возникать колебания с разностной частотой между

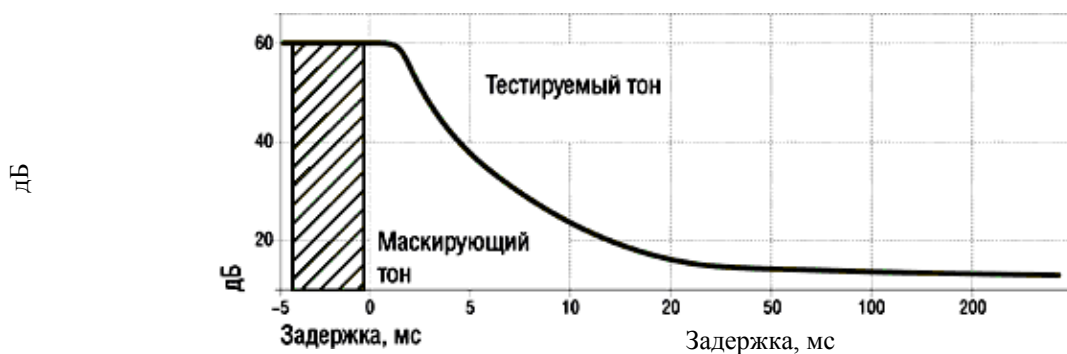


Рис. 5.5. Временная маскировка

двумя более высокочастотными. Этот эффект используется в некоторых коммерческих звуковых системах, чтобы расширить область воспроизводимых низких частот, если невозможно адекватно воспроизвести такие частоты напрямую.

Психоакустика в программном обеспечении. Психоакустические модели слуха позволяют с высоким качеством производить компрессию сигнала с потерей информации (когда восстановленный сигнал не совпадает с исходным), за счет того, что позволяют точно описать, что можно безопасно удалить из исходного сигнала, т.е. без значительного ухудшения качества звука. На первый взгляд может показаться, что вряд ли это позволит обеспечить сильное сжатие сигнала, но программы, использующие психоакустические модели позволяют добиться уменьшения объемов файлов с музыкой в 10 – 12 раз меньше, чем несжатые с очень незначительной разницей в качестве.

К таким видам компрессии относятся все современные форматы компрессии звука: MP3; Ogg Vorbis; Musicam (используется для цифрового аудиовещания в некоторых странах); ATRAC используется в формате MiniDisc; WMA.

5.3.2.2. Форматы звуковых файлов с потерей качества

Windows Media Audio – лицензируемый формат файла, разработанный компанией Microsoft для хранения и трансляции аудио-информации.

Номинально формат WMA характеризуется хорошей способностью сжатия, что позволяет ему "обходить" формат MP3 и конкурировать по параметрам с форматами Ogg Vorbis и AAC. Но как было показано независимыми тестами, а также при субъективной оценке качество форматов всё таки не является однозначно эквивалентным, а преимущество даже перед MP3 однозначным, как это утверждает компания Microsoft. Особенно стоит отметить что ранние версии формата (или его реализации) имели проблемы на низких скоростях потока. Также многие меломаны и владельцы цифровых плееров недолюбливают формат WMA за низкую стойкость к ошибкам. Если при кодировании/передаче файла WMA некоторая часть его повреждается, то воспроизведение файла становится невозможным как после места повреждения, так и за несколько десятков секунд до него. (Для сравнения: при повреждении файла формата MP3, его всё ещё можно воспроизвести от начала до самого места повреждения, затем пропустить несколько секунд и воспроизвести дальше до конца; иногда же ошибки в несколько байт в файле MP3 бывают на слух малозаметны или не заметны вообще.) Однако данный формат постоянно развивается, так что можно предполагать, качество будет оптимизироваться.

Microsoft включила в WMA поддержку цифровой системы управления авторскими правами (DRM) (система защиты). Основным следствием её является невозможность прослушивать защищённые композиции на других компьютерах, кроме того, на котором композиция была загружена из музыкального магазина.

AAC (англ. Advanced Audio Coding) – собственный (патентованный) формат аудиофайла с меньшей потерей качества при кодировании, чем MP3 при одинаковых размерах. Формат также позволяет сжимать без потери качества исходника (профиль ALAC AAC).

Также AAC – это широкополосный алгоритм кодирования аудио, который использует два основных принципа кодирования для сильного уменьшения количества данных, требуемых для передачи высококачественного цифрового аудио. Данный формат является наиболее качественным сжатием с потерями, который поддерживает большинство современного оборудования, в том числе портативного.

На 2008 год распространён несколько меньше, чем MP3 и другие альтернативные решения.

AAC (Advanced Audio Coding) изначально создавался как преемник MP3 с улучшенным качеством кодирования. Формат AAC, официально известный как ISO/IEC 13818-7, вышел в свет в 1997 как новая, седьмая, часть семьи MPEG-2. Существует также формат AAC, известный как MPEG-4 Часть 3.

Аудио стандарт MPEG-4 не требует единственного или малого набора высокоэффективных схем компрессии, а скорее сложный набор для выполнения широкого круга операций от кодирования низкогокачественной речи до высококачественного аудио и

синтезирования музыки. Семейство алгоритмов аудио кодирования MPEG-4 охватывает диапазон от кодирования низкокачественной речи (до 2 кбит/с) до высококачественного аудио (от 64 кбит/с на канал и выше).

ААС имеет частоту сэмплов от 8 Гц до 96 кГц и количество каналов от 1 до 48. В отличие от гибридного набора фильтров MP3, ААС использует Модифицированное Дискретное Косинусное Преобразование (MDCT) вместе с увеличенным размером "окна" в 2048 пунктов.

MP3 (более точно, англ. MPEG-1/2/2.5 Layer 3 (но не MPEG-3) – третий формат кодирования звуковой дорожки MPEG) – лицензируемый формат файла для хранения аудио-информации.

На данный момент MP3 является самым известным и популярным из распространённых форматов цифрового кодирования звуковой информации с потерями. Он широко используется в файлообменных сетях для точечной передачи музыкальных произведений. Формат может проигрываться практически в любой популярной операционной системе, на практически любом портативном аудио-плеере, а также поддерживается всеми современными моделями музыкальных центров и DVD-плееров.

MP3 разработан рабочей группой института Фраунгофера (англ. Fraunhofer Society) MPEG (англ. Moving Picture Expert Group – группа экспертов в области динамического изображения), состоящей из Джонсона, Штолла, Деери и Карлхайнца Бранденбурга. Основой разработки MP3 послужил экспериментальный кодек ASPEC (Adaptive Spectral Perceptual Entropy Coding). Первым кодировщиком в формат MP3 стала программа L3Enc, выпущенная летом 1994 года. Спустя один год появился первый программный MP3-плеер – Winplay3.

Высокая степень сжатия в MP3 достигается за счёт достаточно сложного алгоритма кодирования. Используются как математические методы компрессии, так и особенности человеческого слуха (психоакустическая модель): эффект маскировки слабого звука одной частоты более громким звуком такой же или соседней частоты, понижение чувствительности уха к тихому звуку сразу после громкого, невосприимчивость к звукам ниже определённого уровня громкости, удалением частот, которые неслышимы человеческим ухом (инфразвук и ультразвук).

Поток звука при кодировании разбивается на равные по длине участки (фреймы). Каждый из фреймов кодируется отдельно со своими параметрами и содержит заголовок, в котором эти параметры указаны. Сжатие может быть выполнено с разным качеством и соответственно размером конечного файла.

Степень сжатия характеризуется битрейтом (bitrate) – количество передаваемой за единицу времени информации. Файлы MP3 обычно закодированы с битрейтом 64...320 килобит в секунду (kbps или kb/s), а также с переменным битрейтом (VBR) – когда для каждого фрейма используется свой, оптимальный для данного участка, битрейт.

Исходный сигнал с помощью фильтров разделяется на несколько частотных диапазонов, для каждого диапазона определяется величина маскирующего эффекта от соседних диапазонов и предыдущего фрейма, несущественные сигналы игнорируются. Программа кодирования выделяет самые громкие звуки в каждой полосе и использует эту информацию для определения приемлемого уровня шума для этой полосы. Очень громкий звук в одной полосе может повлиять на маскирующий эффект и на близлежащие полосы.

Также производится удаление заведомо неслышимых частот с более тщательным сохранением звуков, хорошо различаемых человеческим ухом. Ещё одним приёмом сжатия является использование так называемого совмещённого стерео. Известно, что слуховой аппарат человека может определить направление лишь средних частот – высокие и низкие звучат как бы отдельно от источника. Значит, эти фоновые частоты можно кодировать в моно сигнал. Кроме всего этого для сжатия используется различие в сложности потоков в каналах. Например, если в правом канале какое-то время полная тишина, это "зарезервированное" место используется для повышения качества левого канала или туда помещаются необходимые биты, не уместившиеся в потоке чуть раньше.

Для оставшихся данных для каждого диапазона определяется, сколькими битами можно пожертвовать, чтобы потери были ниже величины маскирующего эффекта. На этом работа психоакустической модели завершается, а итоговый поток дополнительно сжимается по алгоритму Хаффмана.

Существуют три версии MP3 формата для различных нужд: MPEG-1, MPEG-2 и MPEG-2.5. Отличаются они возможными диапазонами битрейта и частоты дискретизации:

32...320 кбит/с – при частотах дискретизации 32 000 Гц, 44 100 Гц и 48 000 Гц для MPEG-1 Layer 3;

16...160 кбит/с – при частотах дискретизации 16 000 Гц, 22 050 Гц и 24 000 Гц для MPEG-2 Layer 3;

8...160 кбит/с – при частотах дискретизации 8 000 Гц и 11 025 Гц для MPEG-2.5 Layer 3.

Режимы управления кодированием звуковых каналов. Так как формат MP3 поддерживает двухканальное кодирование (стерео), существуют три режима:

1. Стерео – двухканальное кодирование, при котором каналы кодируются независимо друг от друга. Таким образом, заданный битрейт делится на два канала. Например, если заданный битрейт 192 кбит/с, то для каждого канала он будет равен только 96 кбит/с.

2. Моно – одноканальное кодирование. Если закодировать двухканальный материал этим способом, различия между каналами будут полностью стёрты, так как два канала смешиваются в один, он кодируется, и он же воспроизводится в обоих каналах стереосистемы. Единственным плюсом данного режима может являться только выходное качество по срав-

нению с режимом стерео при одинаковом битрейте, так как на один канал приходится вдвое большее количество бит, чем в режиме стерео. Но различия между каналами не слышны, так как канал здесь только один.

3. Объединённое стерео (Joint Stereo) – оптимальный способ двухканального кодирования, при котором левый и правый каналы преобразуются в их сумму и разность. Для большинства звуковых файлов канал с разницей получается намного тише канала с суммой, поэтому на сумму отводится большая часть битрейта. Таким образом, качество выходного файла разительно отличается в лучшую сторону от режима стерео при одинаковом битрейте, особенно при низком. Бытует мнение, что данный режим не подходит для звукового стереоматериала, в котором в двух каналах воспроизводится субъективно абсолютно различный материал, так как он стирает различия между каналами. Это ошибочное мнение, так как в действительности MP3-кодек оперирует частотами, а определённые частоты в большинстве случаев пересекаются в обоих каналах, т.е. идентичная информация всё же присутствует, а различная – кодируется отдельно. Особенно эффективен этот способ двухканального кодирования при использовании переменного битрейта, речь о котором пойдет ниже.

Как уже упоминалось ранее, в MP3 могут использоваться битрейты как переменной, так и постоянной величины:

1. *CBR* (Constant Bit Rate) – постоянный битрейт, который задаётся пользователем и не изменяется при кодировании произведения; таким образом каждой секунде произведения соответствует одинаковое количество закодированных бит данных (даже при кодировании тишины). Данный режим кодирования не является оптимальным, так как он не годится для большинства динамичных музыкальных произведений при битрейте ниже 256 кбит/с.

2. *VBR* (Variable Bit Rate) – варьирующийся битрейт или переменный битрейт, который динамически изменяется программой-кодером при кодировании, в зависимости от насыщенности кодируемого аудиоматериала и установленного пользователем качества кодирования (например, тишина кодируется с минимальным битрейтом). Этот метод MP3-кодирования является самым прогрессивным и до сих пор развивается и улучшается, так как аудиоматериал разной насыщенности может быть закодирован с определённым качеством, которое обычно выше, чем при установке среднего значения в методе CBR. Плюс к тому, размер файла уменьшается за счёт фрагментов, не требующих высокого битрейта. Минусом данного метода кодирования является полная невозможность предсказать размер выходного файла. Но этот недостаток VBR-кодирования незначителен в сравнении с его достоинствами. Также минусом является то, что VBR считает "незначительной" звуковой информацией более тихие фрагменты; таким образом получается, что если слушать очень громко, то эти фрагменты будут некачественными, в то время как CBR делает с одинаковым битрейтом и тихие, и громкие фрагменты. Формат VBR постоянно улучшается, благодаря постоянному совершенствованию математической модели кодеков, в частности после выхода обновленной версии свободного mp3-кодека lame (версия 3.97), кодирование с переменным битрейтом, по заявлению самих разработчиков, качественно лучше CBR и тем более ABR.

3. *ABR* (Average Bit Rate) – усреднённый битрейт, который является гибридом VBR и CBR: битрейт в кбит/с задается пользователем, а программа варьирует его, постоянно подгоняя под заданный битрейт. Таким образом, кодер будет с осторожностью использовать максимально и минимально возможные значения битрейта, так как рискует не вписаться в заданный пользователем битрейт. Это является явным минусом данного метода, так как сказывается на качестве выходного файла, которое будет немного лучше, чем при использовании CBR, но намного хуже, чем при использовании VBR. С другой стороны, этот метод позволяет наиболее гибко задавать битрейт (может быть любым числом между 8 и 320, против исключительно кратных 16 чисел метода CBR) и вычислять размер выходного файла.

MP3 является лидером по распространённости, но при этом не является лучшим по техническим параметрам. Также в формате MP3 отсутствует режим кодирования без потерь вполне подходит (с профессиональной точки зрения) для распространения демонстрационных композиций или иных способов распространения своей музыки из-за повсеместной распространённости проигрывателей.

5.3.3. Форматы аудиоданных нотной записи

Стандартный MIDI файл – это специально разработанный формат файлов, предназначенный для хранения данных, записываемых и/или исполняемых секвенсером. Секвенсер может быть как программой для компьютера (рис. 5.6), так и аппаратно выполненным модулем.

В этом формате хранятся стандартные MIDI сообщения (т.е. статус-байты и соответствующие им байты данных), а также временные метки или маркеры для каждого сообщения (т.е. последовательности байтов, указывающие, какое количество условных единиц времени (импульсов, тиков) необходимо подождать перед тем, как исполнить следующее событие MIDI). Этот формат позволяет сохранять информацию о темпе, временном разрешении, выраженном в количестве тиков на одну четвертную длительность, обозначения размера, информацию о музыкальных ключах, а также хранить названия трэков и паттернов. Формат предусматривает возможность сохранения в одном файле нескольких паттернов и трэков таким образом, что программы-приложения могут выбирать из всего набора хранимой информации ту, которая будет понятна данному приложению.

Как правило, трэк представляет собой аналог музыкальной партии, например партии трубы. Аналогом паттерна может служить весь набор партий, взятых вместе, например совокупность партий трубы, ударных, фортепиано и т.д., которые используются в данном произведении или его части и исполняются одновременно

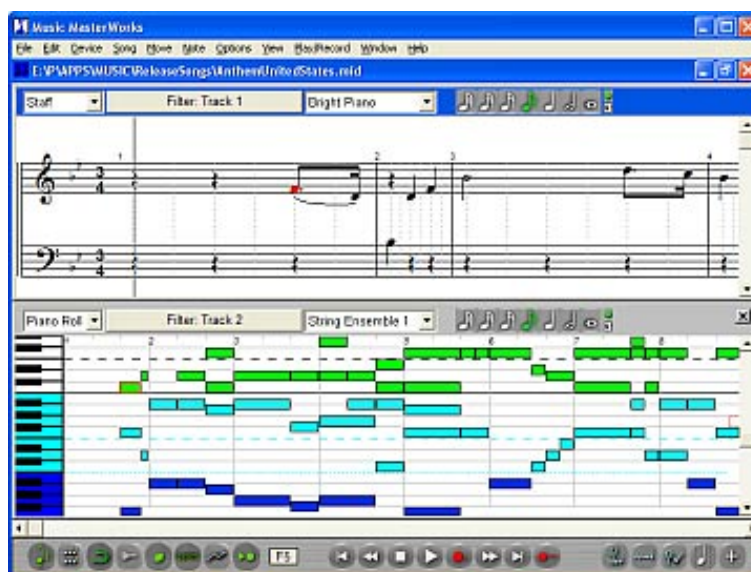


Рис. 5.6 Программа-секвенсор

Формат разработан таким образом, чтобы любой секвенсор мог читать и записывать такой файл таким образом, чтобы не потерялись его данные, и так, чтобы формат был достаточно гибким, чтобы приложения могли сохранять в файлах свою специфическую информацию, понятную только этим приложениям, но не понятную другим программам-приложениям, причем при загрузке файлов MIDI непонятная другим программам-приложениям информация не приводит к недоразумениям, а просто игнорируется.

В этом смысле формат файлов MIDI можно сравнить с файлами, хранящими текстовую информацию. Различные программы-секвенсеры способны читать MIDI-файлы, подобно тому, как различные текстовые редакторы читают ASCII-файлы, которые могут содержать вспомогательную информацию, понятную лишь данному редактору. Но в отличие от ASCII-файлов MIDI-файлы содержат цифровую информацию, и к тому же эта информация сохранена в виде записей, т.е. групп байтов, которые содержат свой заголовок, состоящий из идентификатора записи и длины записи. Эти записи могут форматироваться, загружаться, игнорироваться и т.д. независимо друг от друга. Для осуществления работы с записями программы-приложения используют дополнительную информацию, записываемую в MIDI-файл. Например, возможно, программа "захочет" сохранить флаг, указывающий на то, что пользователь установил включенным звук метронома. Программа может вставить этот флаг в MIDI-файл таким образом, что другая программа-приложение сможет пропустить этот флаг без внимания. В будущем, возможно, существующий формат MIDI будет расширен и появятся новые типы записей. Новые программы для работы с MIDI-файлами будут распознавать и новые типы записей. Однако старые MIDI-файлы могут быть воспроизведены в своём исходном виде. Формат MIDI задуман таким образом, что с его расширениями будут совместимы более ранние его версии.

Данные всегда хранятся в виде записей. В одном MIDI-файле могут сосуществовать несколько различных записей. Каждая запись может иметь свой собственный размер, т.е. количество байтов в различных записях может быть различно. Данные, хранящиеся в одной записи, связаны друг с другом определённым образом. Запись – это по своей сути набор взаимосвязанных байтов. Каждая запись начинается с указания её идентификатора, который состоит из четырёх букв, т.е. из четырёх ASCII байтов. Этот идентификатор указывает, какой тип записи представлен в содержащихся в записи байтах данных. Последующие за идентификатором четыре байта (каждый из которых состоит из 8 бит) образуют 32-битное значение, указывающее длину (или размер) данной записи.

5.3.3. Форматы аудиоданных, использующие ноты и образцы инструментов

MOD – формат файлов, разработанный для создания, хранения и воспроизведения музыкальных композиций на ПК Amiga. Своё название получил от того, что стал первым форматом, хранящим свои фрагменты (например, сэмплы) в других файлах (принцип модульности). Файлы этого формата имеют, как правило, расширения *.mod.

Каждый файл формата MOD содержит в себе оцифрованные записи реального звучания инструментов, так называемые сэмплы. Композитор, пишущий в формате MOD, использует программу, называемую трекером, в которой указывает, какой именно инструмент, в какое время, какой нотой и какой из октав должен прозвучать. Последовательность нот записывается в список – так называемый трек, а несколько параллельно звучащих треков образуют блок, называемый паттерном. Создаваемые композитором паттерны получают номера, после чего композитор может в свободной форме указывать какой паттерн и когда должен прозвучать. Совокупность паттернов и образует модуль – файл в формате MOD.

Общая структура MOD-файла:

- заголовок, его длина может быть либо 600, либо 1084 байта, в зависимости от количества сэмплов;
- паттерны, хранятся по порядку номеров сразу после заголовка, начиная с паттерна 0; так как размер паттерна для конкретного модуля – константа, то зная номер паттерна легко рассчитать его смещение в файле;
- сэмплы, идут по порядку номеров сразу за последним паттерном, каждый сэмпл представляет собой сырую волновую форму из знаковых 8-битовых выборок; первые два байта в каждом сэмпле зарезервированы и не предназначены для хранения выборок, но, на практике, они там всегда есть.

5.3.4. Форматы-контейнеры аудиоданных

RIFF-WAVE. Сам по себе WAVE файл является не более чем контейнером для находящихся в его теле аудиоданных. При этом формат аудиоданных, находящихся в WAVE файле, может быть произвольным: PCM, MP3, WMA и прочее. Ошибочно полагать, что WAVE файл может содержать данные только в формате PCM.

Приблизительно WAVE файл можно представить в виде заголовка, описывающего формат хранимых аудиоданных, самих данных, и произвольного комментария.

Рассмотрим структуру WAVE файла более подробно.

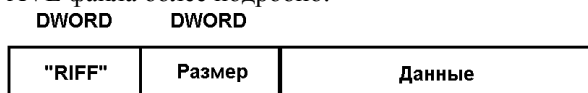


Рис. 5.7. Внешний фрагмент RIFF

Сам файл имеет формат RIFF (Resource Interchange File Format). Такой формат файлов широко распространён, так как удобен для хранения различных данных, например AVI файл также является файлом формата RIFF. Данные в таком файле хранятся в виде блоков. Каждый блок имеет три раздела: идентификатор блока (4 символа, CHAR), размер хранимых данных (двойное слово, DWORD) и непосредственно сами данные, до 4 Гбайт. Файл формата RIFF содержит вложенные фрагменты; внешний фрагмент состоит из заголовка и области данных (рис. 5.7).

Первое двойное слово заголовка содержит четырёхбуквенный код FOURCC, идентифицирующий хранящиеся во фрагменте данные. Второе двойное слово заголовка представляет собой размер области данных в байтах (без учёта размера самого заголовка). Область данных имеет переменную длину, однако она должна быть выравнена на границу слова (при необходимости дополняется в конце нулевым байтом до целого числа слов).

Важно понять, что формат RIFF не описывает конкретный формат данных; практически файл в RIFF-формате может содержать любые мультимедиа-данные, причем формат конкретных данных зависит от типа этих данных.

Обозначенная на рис. 5.7 как "Данные" область может содержать внутри себя другие фрагменты. Для содержащего звуковые данные файла (WAV-файл) эта область содержит идентификатор данных "WAVE", фрагмент формата звуковых данных "fmt" (три символа "fmt" и пробел в конце), а также фрагмент звуковых данных (рис. 5.8).

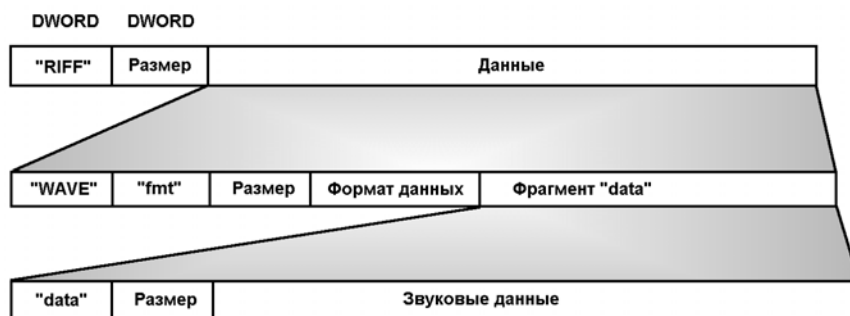


Рис. 5.8. Формат WAV-файла (в структуре RIFF)

Vorbis – свободный формат сжатия звука с потерями, официально появившийся летом 2002 года. Психоакустическая модель, используемая в Vorbis, по принципам действия близка к MPEG Audio Layer III и подобным, однако математическая обработка и практическая реализация этой модели существенно отличаются, что позволило авторам объявить свой формат совершенно независимым от всех предшественников.

Стоит заметить, что Vorbis является всего лишь небольшой частью мультимедиа-проекта Ogg, в который также входят свободные кодировщики: Speex – для сжатия голоса; FLAC – для сжатия звука без потерь; Theora – для сжатия видео.

Для хранения аудиоданных в формате Vorbis чаще всего применяется медиаконтейнер Ogg, такой файл обычно имеет расширение .ogg и называется двойным именем "Ogg/Vorbis"[xiph 1] или "Ogg Vorbis".[xiph 2]. Однако "Ogg Vorbis" называют и сам кодек без контейнера, так как он является частью проекта Ogg.[xiph 1]. По данным 2007 года распространён существенно меньше, чем MP3. По всевозможным оценкам является вторым по популярности форматом компрессии звука с потерями. Широко используется в компьютерных играх и в файлообменных сетях для передачи музыкальных произведений.

По заявлению разработчиков, Vorbis применяет более качественную психоакустическую модель, чем его конкуренты, дающую лучшую чёткость воспроизведения при равной плотности потока. В результате сжатый музыкальный файл окажется в два раза меньше (или в два раза лучше), чем у MP3.

Формат не ограничивает пользователя только двумя аудиоканалами (стерео – левый и правый). Он поддерживает до 255 отдельных каналов с частотой дискретизации до 192 кГц и разрядностью до 32 бит (чего не позволяет ни один другой формат сжатия с потерями), поэтому Vorbis великолепно подходит для кодирования 6-канального звука DVD-Audio.

К тому же, формат Vorbis – "sample accurate". Это гарантирует, что звуковые данные перед кодированием и после декодирования не будут иметь смещений, дополнительных или потерянных сэмплов. Это легко оценить, когда вы кодируете non-stop музыку (когда один трек постепенно переходит в другой) – в итоге сохранится целостность звука.

Ogg Vorbis по умолчанию использует переменный битрейт, при этом значения последнего не ограничены какими-то жёсткими значениями, и он может варьироваться даже на 1 kbps. При этом стоит заметить, что форматом жёстко не ограничен максимальный битрейт, и при максимальных настройках кодирования он может варьировать от 400 до 700 kbps. Такой же гибкостью обладает частота дискретизации – пользователям предоставляется любой выбор в пределах 2...192 кГц.

И, в заключение раздела, приведём список некоторых форматов файлов (табл. 5.1).

5.1. Сравнительная таблица форматов аудиоданных

Название формата	Сжатый	Аудиоформат без потерь	Использование психоакустической модели
AIFF	–	+	–
AU	–	+	–
RAW	–	+	–
WAV	–	+	–
FLAC	+	+	–
Lossless Audio (.la)	+	+	–
Monkey's Audio APE)	+	+	–
OptimFROG (.ofr)	+	+	–
TTA	+	+	–
WavPack (.wv)	+	+	–
Windows Media Audio 9 Lossless (WMA)	+	+	–
MP3	+	–	+
Vorbis	+	–	+
Windows Media Audio (WMA)	+	–	+
AAC (.m4a, .mp4, .m4p, .aac)	+	–	
RealAudio (RA, RM)	+	–	

Контрольные вопросы

1. Перечислите методы синтеза компьютерного звука.
2. Приведите общую схему сжатия аудиоданных с потерями качества.
3. Какие особенности человеческого слуха учитывает психоакустическая модель?
4. Какие форматы аудиофайлов без потери качества и с потерей качества вы знаете?
5. Каким образом в популярном формате mp3 кодируется стереосигнал?

6. ВИДЕО

6.1. ОСНОВНЫЕ ПОНЯТИЯ

Видео (от лат. video – смотрю, вижу) – под этим термином понимают широкий спектр технологий записи, обработки, передачи, хранения и воспроизведения визуального и аудиовизуального материала на мониторах. Когда в быту говорят "видео", то обычно имеют в виду видеоматериал, телесигнал или кинофильм, записанный на физическом носителе (видеокассете, видеодиске и т.п.).

Характеристики видеосигнала.

1. *Количество (частота) кадров в секунду* – это число неподвижных изображений, сменяющих друг друга при показе 1 секунды видеоматериала и создающих эффект движения объектов на экране. Чем больше частота кадров в секунду, тем более плавным и естественным будет казаться движение. Минимальный показатель, при котором движение будет восприниматься однородным – примерно 10 кадров в секунду (это значение индивидуально для каждого человека). В традиционном плёночном кинематографе используется частота 24 кадра в секунду. Системы телевидения PAL и SECAM используют 25 кадров в секунду (англ. 25 fps или 25 Гц), а система NTSC использует 29,97 кадров в секунду. Компьютерные оцифрованные видеоматериалы хорошего качества, как правило, используют частоту 30 кадров в секунду. Верхняя пороговая частота мелькания, воспринимаемая человеческим мозгом, в среднем составляет 39...42 Гц и индивидуальна для каждого человека. Некоторые современные профессиональные камеры могут снимать с частотой до 120 кадров в секунду.

2. *Развёртка видеоматериала* может быть прогрессивной (построчной) или *чересстрочной*. При прогрессивной развёртке все горизонтальные линии (строки) изображения отображаются поочередно одна за другой. А вот при чересстрочной развёртке показываются попеременно то все чётные, то все нечётные строки (называемые также полями кадра). Чересстрочную развёртку часто называют на английский манер интерлейс (англ. interlace) или интерлейсинг. Чересстрочная развёртка была изобретена для показа изображения на кинескопах и используется сейчас для передачи видео по "узким" каналам, не позволяющим передавать изображение во всём качестве. Системы PAL, SECAM и NTSC – это всё системы с чересстрочной развёрткой. Новые цифровые стандарты телевидения, например, HDTV предусматривают прогрессивную развёртку. Для подавления неприятных эффектов, возникающих при просмотре чересстрочного видео на построчном экране, применяются специальные математические методы, именуемые деинтерлейсингом.

3. По аналогии с разрешением компьютерных мониторов, любой видеосигнал также имеет *разрешение* (англ. resolution), горизонтальное и вертикальное, измеряемое в пикселях. Обычное аналоговое телевизионное разрешение составляет 720×576 пикселей для стандартов PAL и SECAM, при частоте кадров 50 Гц (одно поле, 2×25); и 720×480 пикселей для NTSC, при частоте 60 Гц (одно поле, $2 \times 29,97$). Новый стандарт высокоотчётливого (англ. high-definition) цифрового телевидения HDTV предполагает разрешения до 1920×1080 при частоте мелькания 60 Гц с прогрессивной развёрткой.

Разрешение в случае трёхмерного видео измеряется в вокселях – элементах изображения, представляющих точки (кубики) в трёхмерном пространстве. Например, для простого трёхмерного видео сейчас используется в основном разрешение $512 \times 512 \times 512$, демонстрационные примеры такого видео доступны сегодня даже на PDA.

4. *Соотношение ширины и высоты кадра* (англ. aspect ratio) – важнейший параметр в любом видеоматериале. Ещё с 1910 года кинофильмы имели соотношение сторон экрана 4:3 (иногда ещё записывается как 1,33:1 или просто 1,33). Считалось что, с учётом наличия у человека двух глаз, зрителю удобнее смотреть фильм на экране такой формы. Когда появилось телевидение, то оно переняло это соотношение и почти все аналоговые телесистемы (и, следовательно, телевизоры) имели соотношение сторон экрана 4:3. Компьютерные мониторы также унаследовали телевизионный стандарт сторон. Хотя ещё в 1950-х годах это представление о 4:3 в корне изменилось. Дело в том, что поле зрения человека имеет соотношение отнюдь не 4:3. Ведь у человека два глаза, расположенных на одной горизонтальной линии, следовательно, поле зрения человека приближается к соотношению 2:1. Чтобы приблизить форму кадра к естественному полю зрения человека (и, следовательно, усилить восприятие фильма), был введён стандарт 16:9 (1,78), почти соответствующий так называемому "Золотому сечению". Цифровое телевидение в основном тоже ориентируется на соотношение 16:9.

5. *Количество цветов и цветовое разрешение* видеосигнала описывается цветовыми моделями. Для стандарта PAL применяется цветовая модель YUV, для SECAM модель YDbDr, для NTSC модель YIQ, в компьютерной технике применяется в основном RGB, реже HSV, а в печатной технике CMYK. Количество цветов, которое может отобразить монитор или проектор зависит от качества монитора или проектора. Челове-

ский глаз может воспринять, по разным подсчётам, от 5 до 10 миллионов оттенков цветов. Количество цветов в видеоматериале определяется числом бит, отведённым для кодирования цвета каждого пикселя (англ. bits per pixel, bpp).

6. *Битрейт или ширина видеопотока* (для цифрового видео). Ширина (иначе говорят скорость) видеопотока или битрейт (англ. bit rate) – это количество обрабатываемых бит видеoinформации за секунду времени (обозначается бит/с – бит в секунду, или чаще Мбит/с – мегабит в секунду; в английском обозначении bit/s и Mbit/s, соответственно). Чем выше ширина видеопотока, тем лучше качество видео.

Различают два вида управления шириной потока в видеокодеке – постоянный битрейт (англ. constant bit rate, CBR) и переменный битрейт (англ. variable bit rate, VBR). Концепция VBR призвана максимально сохранить качество видео, уменьшая при этом суммарный объём передаваемого видеопотока. При этом на быстрых сценах движения ширина видеопотока возрастает, а на медленных сценах, где картинка меняется медленно, ширина потока падает. Это очень удобно для буферизованных видеотрансляций и передачи сохранённого видеоматериала по компьютерным сетям. Но для безбуферных систем реального времени и для прямого эфира (например, для телеконференций) это не подходит – в этих случаях необходимо использовать постоянную скорость видеопотока.

7. *Качество видео* измеряется с помощью формальных метрик таких, как PSNR или SSIM, или с использованием субъективного сравнения с привлечением экспертов.

Субъективное качество видео измеряется по следующей методике:

- выбираются видеопоследовательности для использования в тесте;
- выбираются параметры системы измерения;
- выбирается метод показа видео и подсчета результатов измерения;
- приглашается необходимое число экспертов (обычно не меньше 15);
- проводится сам тест;
- подсчитывается средняя оценка на основе оценок экспертов.

8. *Стереоскопическое видео* или просто стереовидео (англ. Stereoscopic video или 3D video) было очень популярно в конце XX века, и сейчас регулярно возникают волны интереса к нему. По всему миру есть кинотеатры, которые при помощи той или иной технологии воспроизводят стереоскопическое видео. Для стереовидео нужно два видеоканала, часто называемых слоями: один для левого глаза, другой для правого. Таким образом у зрителя возникает чувство объёмности, трёхмерности видеоматериала, повышается реалистичность ощущения просмотра. Примерно такой же по качеству, но более слабый эффект даёт просмотр видео в пластиковых очках, где одна линза красная, а другая голубая или зелёная. Новые технологии, представленные в 2006 году, в частности HD DVD и диски Blu-Ray, позволяют переносить больше стереовидеоматериала и призваны сделать и домашнее стереоскопическое видео более доступным.

6.2. КИНО И ВИДЕОФОРМАТЫ

6.2.1. Аналоговые форматы

SÉCAM или **SECAM** (от фр. Séquentiel couleur avec mémoire, позднее Séquentiel couleur à mémoire – последовательный цвет с памятью; произносится [секам]) – система аналогового цветного телевидения, впервые применённая во Франции. Исторически она является первым европейским стандартом цветного телевидения.

Как и все аналоговые телевизионные стандарты, SECAM является адаптированным и совместимым с более старым монохромным (черно-белым) телевидением. В адаптированных аналоговых стандартах цветного телевидения дополнительный сигнал цветности передаётся в конце спектра монохромного телесигнала.

Как известно из природы зрения человека, ощущение цвета складывается из трёх составляющих: красного (R), зеленого (G) и синего (B) цветов. Такую цветовую модель обозначают аббревиатурой RGB. Из-за преобладания в среднестатистической телевизионной картинке зелёной составляющей цвета и для избежания избыточного кодирования, в качестве дополнительного сигнала цветности используют цветоразностные сигналы R-Y и B-Y (Y – общая яркость монохромного телесигнала). В системе SECAM используют цветовую модель YDbDr (разновидность YUV).

Сигнал цветности в стандарте SECAM передаётся в частотной модуляции (ЧМ), по одной цветовой составляющей в одной телевизионной строке, поочередно. В качестве недостающих строк используют предыдущий сигнал R-Y или B-Y соответственно, получая его из памяти (в аналоговых телевизионных приёмниках для этого используется линия задержки). Таким образом, объективно, цветное телевизионное изображение в стандарте SECAM имеет в два раза меньшее разрешение по вертикали, чем монохромное изображение. Субъективно, в силу большей чувствительности глаза к яркостной составляющей, на среднестатистических картинках такое ухудшение почти не заметно. Применение цифровой обработки сигнала еще больше сглаживает этот недостаток.

Применение частотной модуляции, поочередной передачи цветового сигнала и цветовой модели YDbDr является отличительной особенностью SECAM от других телевизионных аналоговых стандартов (PAL и NTSC).

В мире используются несколько модификаций стандарта SECAM. Стандарт MESECAM используется только при записи на магнитную ленту. В этом стандарте для уменьшения влияния непостоянства скорости лентопротяжного механизма на качество цвета, поднесущие цветоразностных сигналов перенесены на более низкие частоты (примерно 1.1 МГц).

PAL (от англ. phase-alternating line) – система аналогового цветного телевидения, разработана инженером немецкой компании "Telefunken" Вальтером Брухом и представленная как стандарт телевизионного вещания в 1967 году.

Как и все аналоговые телевизионные стандарты, PAL является адаптированным и совместимым с более старым монохромным (чёрно-белым) телевидением. В адаптированных аналоговых стандартах цветного телевидения дополнительный сигнал цветности передаётся в конце спектра монохромного телесигнала.

Для передачи цвета в системе PAL используют цветовую модель YUV. Оба дополнительных сигнала цветности в стандарте PAL передаются одновременно в квадратурной модуляции (разновидность АМ), типичная частота поднесущего сигнала – 4 433 618,75 Гц (4,43 МГц). При этом каждый цветоразностный сигнал повторяют в следующей строке с поворотом фазы с частотой 15,625 кГц на 180 градусов, благодаря чему декодер PAL полностью устраняет фазовые ошибки (типичные для системы NTSC). Для устранения фазовой ошибки декодер складывает текущую строку и предыдущую из памяти (в аналоговых телевизионных приёмниках используется линия задержки). Таким образом, объективно, цветное телевизионное изображение в стандарте PAL имеет в два раза меньшее разрешение по вертикали, чем монохромное изображение. Субъективно, в силу большей чувствительности глаза к яркостной составляющей, на среднестатистических картинках такое ухудшение почти не заметно. Применение цифровой обработки сигнала ещё больше сглаживает этот недостаток.

Применение квадратурной модуляции является отличительной особенностью PAL от стандарта SECAM, повтор цветоразностных сигналов в противофазе отличает его от NTSC, цветовая модель YUV отличает от всех аналоговых систем.

NTSC (от англ. National Television Standards Committee, Национальный комитет по телевизионным стандартам) – система аналогового цветного телевидения, разработанная в США. 18 декабря 1953 года впервые в мире было начато цветное телевизионное вещание с применением именно этой системы.

NTSC принята в качестве стандартной системы цветного телевидения также в Канаде, Японии и ряде стран Америки.

Будем рассматривать систему NTSC, применяемую в США, так называемый "базовый" NTSC M:

- частота смены полей (полукадров) – 60 Гц (точнее 59,94005994 Гц);
- количество строк (разрешение) – 525;
- частота поднесущей – 3 579 545,5 Гц;
- количество кадров в секунду – 30;
- развёртка луча чересстрочная (интерлейсинг).

Передача цветоразностных сигналов в системе NTSC осуществляется в спектре яркостного сигнала на одной поднесущей. Два цветоразностных сигнала E_R -Y и E_B -Y передаются с помощью квадратурной модуляции.

Цветоразностные сигналы подаются на балансный модулятор, на котором они модулируются по амплитуде с подавлением поднесущей. Модулированные цветоразностные сигналы красного E_R -Y и синего E_B -Y сдвинуты относительно друг друга по фазе на 90°. При суммировании они образуют новый сигнал – сигнал цветности.

Применение амплитудной модуляции с подавленной поднесущей порождает трудности при приёме. При детектировании важно чтобы совпадали фазы и частоты гетеродина и поднесущей. Для этого после каждого синхроимпульса строки следует особый импульс-вспышка – он содержит 8 – 10 периодов колебаний опорного генератора. Частота поднесущей выбрана таким образом, чтобы как можно меньше влиять на приёмники чёрно-белого телевидения. При этом, в интервале строки размещается нечётное число полупериодов поднесущей (точно – 455), поэтому рисунок от помехи имеет вид шахматного поля. Такая структура менее заметная, чем вертикальные полосы. Полярность поднесущей в смежных кадрах изменяется на противоположную; таким образом, тёмные участки чередуются со светлыми. За счёт временной взаимной компенсации, помеха становится ещё менее заметной.

6.2.2. Цифровые форматы

HDTV (High-Definition Television) – телевидение высокой чёткости – набор стандартов телевизионного вещания повышенного качества посредством цифровых каналов связи (кабельные, спутниковые сети, цифровые носители).

Разработка телевидения высокой чёткости ведётся, начиная с 1940-х годов. В середине 1950-х годов были созданы первые прототипы. Однако для того, чтобы высокая чёткость телевидения стала заметна невооружённым глазом, необходим дисплей с большой диагональю экрана. Высокая стоимость таких дисплеев тормозила

развитие HDTV на протяжении десятилетий. Стремительное развитие HDTV началось в середине 2000-х годов, одновременно с широким распространением плазменных и жидкокристаллических дисплеев. Для просмотра сигнала HDTV были разработаны специальные HDTV-тюнеры, дисплеи с разрешением HDTV, цифровые интерфейсы HDMI и DVI-D, а также носители HD-DVD и Blu-Ray. Вещание фильмов и телепередач в стандарте HDTV в США, Европе, Японии ведётся уже несколько лет, по платным кабельным и спутниковым каналам.

Передача видеосигнала HDTV на дальние расстояния (от вещательной станции до приёмника конечного пользователя) осуществляется, как правило, в сжатом цифровом виде. Сжатие видео на порядок снижает требования к ширине канала передачи (с 250 до 15...25 Мбит/с), при этом качество изображения остаётся приемлемым. Общепринятый стандарт кодирования видео в HDTV-формате – кодек H.264.

Так как вещание HDTV осуществляется в цифровом виде, то для передачи контента годится практически любой цифровой канал с достаточным уровнем качества (QoS), т.е. достаточной ширины (15...25 Мбит/с, в зависимости от степени сжатия) и гарантирующий определённый приемлемый уровень задержки сигнала (1...10 секунд, в зависимости от размера буфера приёмного устройства и требований к задержке сигнала).

Передача HDTV-сигнала на короткие расстояния (от приёмника пользователя к дисплею) осуществляется в несжатом виде через цифровые интерфейсы (кабели) HDMI и DVI-D. Использование цифровых интерфейсов позволяет полностью избавиться от цифро-аналоговых преобразований на всём пути прохождения сигнала. Однако допускается подключение и по компонентным аналоговым интерфейсам (RGBHV и YPbPr).

Наиболее популярные форматы стандартов HDTV:

– 720p[2]: 1280 × 720 px, прогрессивная развёртка, отношение сторон 16:9, частота – 24, 25, 30, 50 или 60 кадров в секунду;

– 1080i: 1920 × 1080 px, чересстрочная развёртка, отношение сторон 16:9, частота – 50 или 60 полей в секунду;

– 1080p: 1920 × 1080 px, прогрессивная развёртка, отношение сторон 16:9, частота – 24, 25 или 30 кадров в секунду.

6.3. СЖАТИЕ ВИДЕОИНФОРМАЦИИ

Основной сложностью при работе с видео являются большие объёмы дискового пространства, необходимого для хранения даже небольших фрагментов. Причём даже применение современных алгоритмов сжатия не изменяет ситуацию кардинально. При записи на один компакт-диск "в бытовом качестве" на него можно поместить несколько тысяч фотографий, примерно 10 часов музыки и всего полчаса видео. Видео "телевизионного" формата 720 × 576 пикселей 25 кадров в секунду в системе RGB требует потока данных примерно в 240 Мбит/с (т.е. 1,8 Гб/мин). При этом традиционные алгоритмы сжатия изображений, ориентированные на отдельные кадры, не спасают ситуации, поскольку даже при уменьшении потока в 10 раз он составляет достаточно большие величины.

В результате подавляющее большинство сегодняшних алгоритмов сжатия видео являются алгоритмами с потерей данных. При сжатии используется несколько типов избыточности:

1) *когерентность областей изображения* – малое изменение цвета изображения в соседних пикселях (свойство, которое эксплуатируют все алгоритмы сжатия изображений с потерями);

2) *избыточность в цветовых плоскостях* – используется большая важность яркости изображения для восприятия;

3) *подобие между кадрами* – использование того факта, что на скорости 25 кадров в секунду, как правило, соседние кадры изменяются незначительно.

Первые два пункта знакомы вам по алгоритмам сжатия графики. Использование подобия между кадрами в самом простом и наиболее часто используемом случае означает кодирование не самого нового кадра, а его разности с предыдущим кадром. Для некоторых типов видео (передача новостей, видеотелефоны) большая часть кадра остаётся неизменной и даже такой простой метод позволяет значительно уменьшить поток данных. Более сложный метод заключается в нахождении для каждого блока в сжимаемом кадре наименее отличающегося от него блока в кадре, используемом в качестве базового. Далее кодируется разница между этими блоками. Этот метод существенно более ресурсоёмкий.

6.3.1. Требования приложений к алгоритму

Для алгоритмов сжатия видео характерны большинство тех же требований приложений, которые предъявляются к алгоритмам сжатия графики, однако есть и определённая специфика.

Произвольный доступ – подразумевает возможность найти и показать любой кадр за ограниченное время. Обеспечивается наличием в потоке данных так называемых *точек входа* – кадров, сжатых независимо (т.е. как обычное статическое изображение). Приемлемым временем поиска произвольного кадра считается 1/2 с.

Быстрый поиск вперед/назад – подразумевает быстрый показ кадров, не следующих друг за другом в исходном потоке. Требуется наличия дополнительной информации в потоке. Эта возможность активно используется всевозможными проигрывателями.

Показ кадров фильма в обратном направлении. Редко требуется в приложениях. При жёстких ограничениях на время показа очередного кадра выполнение этого требования может резко уменьшить степень сжатия.

Аудиовизуальная синхронизация – самое серьёзное требование. Данные, необходимые для того, чтобы добиться синхронности аудио и видео дорожек, существенно увеличивают размер фильма.

Устойчивость к ошибкам – требование, обусловленное тем, что большинство каналов связи ненадёжны. Испорченное помехой изображение должно быстро восстанавливаться. Требование достаточно легко удовлетворяется необходимым числом независимых кадров в потоке. При этом также уменьшается степень сжатия, так как на экране 2-3 с (50 – 75 кадров) может быть одно и то же изображение, которое нагружает поток независимыми кадрами.

Время кодирования/декодирования. Во многих системах (например, видеотелефонах) общая задержка на кодирование-передачу-декодирование должна составлять не более 150 мс. Кроме того, в приложениях, где необходимо редактирование, нормальная интерактивная работа невозможна, если время реакции системы составляет более 1 с.

Под *редактируемостью* понимается возможность изменять все кадры так же легко, как если бы они были записаны независимо.

Масштабируемость – простота реализации концепции "видео в окне". Масштабирование способно породить неприятные эффекты в алгоритмах, основанных на ДКП (дискретном косинусном преобразовании). Корректно реализовать эту возможность для MPEG на данный момент можно, пожалуй, лишь при достаточно сложных аппаратных реализациях, только тогда алгоритмы масштабирования не будут существенно увеличивать время декодирования. Масштабирование достаточно легко осуществляется в так называемых фрактальных алгоритмах. В них, даже при увеличении изображения в несколько раз, оно не распадается на квадраты, т.е. отсутствует эффект "зернистости". Если необходимо уменьшать изображение (что хоть и редко, но бывает нужно), то с такой задачей хорошо справляются алгоритмы, основанные на wavelet-преобразовании.

Небольшая стоимость аппаратной реализации. При разработке хотя бы приблизительно должна оцениваться и учитываться конечная стоимость. На практике это требование означает, что алгоритм должен реализовываться небольшим набором микросхем.

Описанные требования к алгоритму противоречивы. Очевидно, что высокая степень сжатия подразумевает архивацию каждого последующего кадра с использованием предыдущего. В то же время требования на аудиовизуальную синхронизацию и произвольный доступ к любому кадру за ограниченное время не дают возможности вытянуть все кадры в цепочку. Сбалансированная реализация, учитывающая систему противоречивых требований, может достигаться на практике за счёт настроек компрессора при сжатии конкретного фильма.

6.3.2. Технология сжатия видеоданных [2]

В 1988 году в рамках Международной организации по стандартизации (ISO) начала работу группа MPEG (Moving Pictures Experts Group) – группа экспертов в области цифрового видео (ISO-IEC/JTC1/SC2/WG11/MPEG). Группа работала в направлениях, которые можно условно назвать MPEGVideo-сжатие видеосигнала в поток со скоростью до 1,5 Мбит/с, MPEGAudio – сжатие звука до 64, 128 или 192 Кбит/с на канал и MPEG-System – синхронизация видео- и аудиопотоков.

Как *алгоритм* MPEG имеет несколько предшественников. Это, прежде всего, универсальный алгоритм JPEG. Его универсальность означает, что JPEG показывает неплохие результаты на широком классе изображений.

Если быть более точным, то *стандарт* MPEG, как и другие стандарты на сжатие, описывает лишь выходной битовый поток, неявно задавая алгоритмы кодирования и декодирования. При этом их реализация перекладывается на программистов-разработчиков. Такой подход открывает широкие горизонты для тех, кто желает оптимально реализовать алгоритм для конкретного вычислительного устройства (контроллера, ПК, распределённой вычислительной системы), операционной системы, видеокарты и т.п.

В сентябре 1990 года был представлен предварительный стандарт кодирования MPEG-1. В январе 1992 года работа над MPEG-1 была завершена и начата работа над MPEG-2, в задачу которого входило описание потока данных со скоростью 3...10 Мбит/с [3]. Практически в то же время была начата работа над MPEG-3, который был предназначен для описания потоков 20...40 Мбит/с. Однако вскоре выяснилось, что алгоритмические решения для MPEG-2 и MPEG-3 принципиально близки и можно безболезненно расширить рамки MPEG-2 до потоков в 40 Мбит/с. В результате работа над MPEG-3 была прекращена. MPEG-2 был окончательно доработан к 1995 году.

В 1991 году группой экспертов по видеотелефонам (EGVT) при Международном консультативном комитете по телефонии и телеграфии (CCITT) предложен стандарт видеотелефонов рх64 Kbit/s [4,9]. Запись рх64 означает, что алгоритм ориентирован на параллельную передачу оцифрованного видеоизображения по р-каналам с пропускной способностью 64 Кбита/с. Таким образом, захватывая несколько телефонных линий, можно получать изображение вполне приемлемого качества. Одним из главных ограничений при создании алгоритма являлось время задержки, которое должно было составлять не более ISO мс. Кроме того, уровень помех в телефонных каналах достаточно высок, и это, естественно, нашло отражение в алгоритме. Можно считать, что рх64 Kbits – предшественник MPEG для потоков данных менее 1,5 Мбит/с и специфического класса видео.

В группе при СМТТ (совместный комитет при СС1ТТ и СС1RInternational Consultative Committee on bRoadcasting) работы были направлены на передачу оцифрованного видео по выделенным каналам с высокой пропускной способностью и радиолинией. Соответствующие стандарты H21 и H22 ориентированы на 34 и 45 Мбит/с, и сигнал передаётся с очень высоким качеством.

MPEG-4 изначально был задуман как стандарт для работы со сверхнизкими потоками. Однако в процессе довольно долгой подготовки стандарт претерпел совершенно революционные изменения, и сейчас собственно сжатие с низким потоком входит в него как одна составная часть, причём достаточно небольшая по размеру. Например, сам формат сегодня включает в себя такие вещи, как синтез речи, рендеринг изображений и описания параметров визуализации лица на стороне программы просмотра.

Разработка MPEG-7 была начата в 1996 году. Собственно к алгоритмам сжатия видео этот стандарт имеет ещё меньшее отношение, чем MPEG-4, поскольку его основная задача заключается в описании контента и управлении им. Описание MPEG-7 выходит за рамки этой книги.

Параллельно всё это время существовали форматы Motion-JPEG и недавно появившийся Motion-JPEG 2000, предназначенные в основном для удобства обработки сжатого видео. Рассмотрим основные стандарты и лежащие в их основе алгоритмы.

6.3.2.1. Описание алгоритма компрессии видеоданных

Технология сжатия видео в MPEG распадается на две части: уменьшение избыточности видеoinформации во временном измерении, основанное на том, что соседние кадры, как правило, отличаются несильно, и сжатие отдельных изображений. Для того чтобы удовлетворить противоречивым требованиям и увеличить гибкость алгоритма, рассматриваются

четыре типа кадров:

- I-кадры, сжатые независимо от других кадров (I-Intra pictures);
- P-кадры, сжатые с использованием ссылки на одно изображение (P-Predicted);
- B-кадры, сжатые с использованием ссылки на два изображения (B-Bidirection);
- DC-кадры, независимо сжатые с большой потерей качества (используются только при быстром поиске).

I-кадры обеспечивают возможность произвольного доступа к любому кадру, являясь своеобразными входными точками в поток данных для декодера. P-кадры используют при архивации ссылку на один I- или P-кадр, повышая тем самым степень сжатия фильма в целом. B-кадры, используя ссылки на два кадра, находящиеся впереди и позади, обеспечивают наивысшую степень сжатия. Сами в качестве ссылки использоваться не могут.

Частота I-кадров выбирается в зависимости от требований на время произвольного доступа и надёжности потока при передаче через канал с ошибками. Соотношение P- и B-кадров подбирается, исходя из требований к величине компрессии и ограничений декодеру. Как правило, декодирование B-кадров требует больше вычислительных мощностей, однако позволяет повысить степень сжатия. Именно варьирование частоты кадров разных типов обеспечивает алгоритму необходимую гибкость и возможность расширения. Понятно, что для того, чтобы распаковать B-кадр, мы должны уже распаковать те кадры, на которые он ссылается.

Одним из основных понятий при сжатии нескольких изображений является понятие *макроблока*. При сжатии кадр из цветового пространства RGB переводится в цветовое пространство YUV. Каждая из плоскостей сжимаемого изображения (Y, U, V) разделяется на блоки 8×8 , с которыми работает ДКП. Причем плоскости U и V, соответствующие компоненте цветности, берутся с разрешением в 2 раза меньшим (по вертикали и горизонтали), чем исходное изображение. Таким образом, сразу получаем сжатие в 2 раза, пользуясь тем, что глаз человека хуже различает цвет отдельной точки изображения, чем её яркость. Блоки 8×8 группируются в макроблоки. Макроблок – это группа из четырёх соседних блоков в плоскости яркостной компоненты Y (матрица пикселей 16×16 элементов) и два соответствующих им по расположению блока из плоскостей цветности U и V. Таким образом, кадр разбивается на независимые единицы, несущие полную информацию о части изображения. При этом размер изображения должен быть кратен 16. Отдельные макроблоки сжимаются независимо, т.е. в B-кадрах можно сжать макроблок конкретный как I-блок, P-блок со ссылкой на предыдущий кадр, P-блок со ссылкой на последующий кадр и, наконец, как B-блок.

Алгоритм сжатия отдельных кадров в MPEG похож на соответствующий алгоритм для статических изображений – JPEG. В целом весь конвейер преобразований можно представить так:

1. Подготовка макроблоков. Для каждого макроблока определяется, каким образом он будет сжат. В I-кадрах все макроблоки сжимаются независимо. В P-кадрах блок либо сжимается независимо, либо представляет собой разность с одним из макроблоков в предыдущем опорном кадре, на который ссылается P-кадр.
2. Перевод макроблока в цветовое пространство YUV. Получение нужного количества матриц 8×8 .
3. Для P- и B-блоков производится вычисление разности с соответствующим макроблоком в опорном кадре.
4. ДКП.
5. Квантование.
6. Зигзаг-сканирование.

7. Групповое кодирование.

8. Кодирование Хаффмана.

При декодировании весь конвейер повторяется для обратных преобразований, начиная с конца.

6.3.2.2. Улучшение схемы алгоритма сжатия видеоданных

Существует несколько путей увеличения степени сжатия при применении общей схемы компрессии видеoinформации.

Использование векторов смещений блоков. Простейший способ учитывать подобие соседних кадров – это вычитать каждый блок сжимаемого кадра из соответствующего блока предыдущего.

Однако более гибким является алгоритм поиска векторов, на которые сдвинулись блоки текущего кадра по отношению к предыдущему. Для каждого блока в изображении находится блок, близкий по некоторой метрике (например, по сумме квадратов разности пикселей), в предыдущем кадре в некоторой окрестности текущего положения блока (рис. 6.1). Если минимальное расстояние по выбранной метрике с блоками в предыдущем кадре больше выбранного порога, блок сжимается независимо [3].

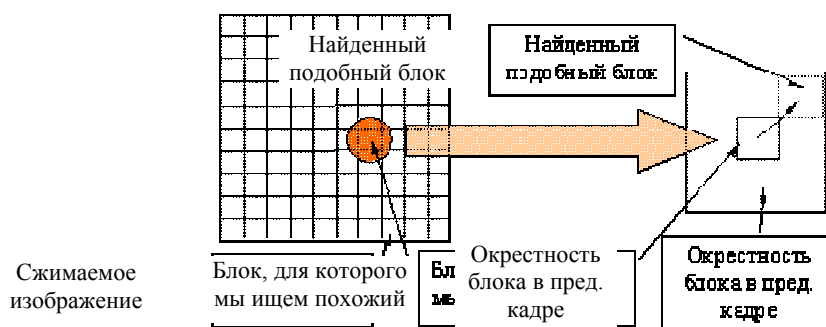


Рис. 6.1. Схема использования векторов смещений блоков

Таким образом, вместе с каждым блоком в поток теперь сохраняются координаты смещения максимально похожего блока в предыдущем I- или P-кадре, либо признак того, что данные сжаты независимо. Эти координаты задают вектор смещения блока. В ситуациях, когда камера наезжает на объект или даёт панораму, использование векторов смещений блоков позволяет значительно уменьшить амплитуду разности кадров и, как следствие, значительно поднять степень сжатия.

Анализ реальных фильмов показывает, что часто блок сдвигается не на кратное число пикселей, а, например, на 10,4 пикселя (камера быстро движется вправо, план съёмки сдвигается равномерно и проходит полный кадр размером 352×240 за 1,35 с). При этом оказывается, что для повышения степени сжатия выгодно строить четыре области поиска векторов смещений: исходную, сдвинутую на полпикселя по горизонтали, сдвинутую на полпикселя по вертикали и сдвинутую на полпикселя по горизонтали и по вертикали (по диагонали), которые строятся с помощью достаточно быстрых алгоритмов билинейной или кусочно-линейной аппроксимации. Этот приём также позволяет уменьшить разность между блоками и повысить степень сжатия при минимальной дополнительной информации, которую надо сохранять в файл (плюс 2 бита на каждый блок). Правда, строить аппроксимированные блоки придётся и при декомпрессии, однако это сравнительно дешёвая по времени операция, которая весьма незначительно увеличивает общее время декомпрессии.

Возможности по распараллеливанию работы алгоритма. Анализ обобщённой схемы алгоритма сжатия видеоданных позволяет заметить, что он сравнительно легко распараллеливается. Изображение 320×288 содержит 330 макроблоков, которые можно кодировать и декодировать независимо. Каждый макроблок, в свою очередь, содержит 6 блоков данных для ДКП. Распараллелить ДКП очень важно, так как, не считая поиска векторов смещения, это самая медленная операция. Заметим также, что остальные преобразования легко конвейеризуются. В результате получается параллельно-конвейерная схема обработки потока видеоданных.

Возможно распараллелить обработку различных кадров, но данный шаг сопряжен со сложностями. Как правило, компрессор строится таким образом, чтобы после сжатия изображение подвергалось обратным преобразованиям. Таким образом, формируется кадр с потерями, а остальные кадры архивируются отгалкиваясь от него. Это позволяет не накапливать ошибки, получаемые ещё при квантовании. Таким образом, если на экране между кадрами наблюдались большие изменения и качество изображения пришлось понизить, то при стабилизации изображения качество быстро повышается практически до качества исходного видеоряда. Неприятный эффект, порождаемый этим приёмом, заключается в том, что появляется мерцание отдельных точек (или областей) изображения, значение цвета в которых округляется то в большую, то в меньшую сторону.

При распаковке возможности по параллельной обработке различных кадров достаточно ограничены, поскольку велика зависимость между кадрами в потоке (велик процент P- и B-кадров).

Описанный выше алгоритм в целом крайне близок к большинству применяемых сейчас на практике алгоритмам сжатия видео. Однако новые идеи появляются ежегодно. Если для алгоритмов сжатия без потерь можно говорить о росте степени сжатия на 1 % в год (относительно предыдущего года) для достаточно большого тестового массива данных, то для алгоритмов сжатия видео речь обычно идет о 3...5 % прибавки степени сжатия для достаточно большого видеофрагмента при том же визуальном качестве.

Если, с одной стороны, повышается степень сжатия, то, с другой стороны, растёт сложность программы и падает скорость работы как при компрессии так и при декомпрессии.

Перечислим основные пути повышения степени сжатия.

1. Изменение алгоритма сжатия I-кадров. Выше приведён алгоритм, основанный на ДКП. Сегодня всё чаще используются алгоритмы, основанные на вейвлетах.

2. Изменение алгоритма сжатия без потерь. Выше приведён алгоритм, использующий сжатие по алгоритму Хаффмана. Однако недавно закончился основной патент на арифметическое сжатие (дающее преимущество 2...15 %), и, соответственно, всё чаще используется именно оно.

3. Изменение алгоритма работы с векторами смещения блоков. Подбор векторов смещений блоков по наименьшему среднеквадратичному смещению не является оптимальным. Существуют алгоритмы, дающие лучший результат при некоторых дополнительных затратах времени при сжатии.

4. Учёт компенсации движения альтернативными способами (без использования векторов смещения блоков). Идеальным алгоритмом сжатия движения объекта является его выделение в кадре и компактное описание его движений. Самым удобным способом описания движения является модель движения (parametric motion model), однако её использование предполагает большой объём вычислений и весьма сложные алгоритмы.

5. Применение обработки коэффициентов. Можно пытаться получить больше информации об изображении из сохранённых коэффициентов. Например, возможно быстрое сравнение коэффициентов после ДКП в соседних блоках и их усреднение по достаточно сложным алгоритмам. Этот приём заметно снижает количество артефактов, вносимых в изображение ДКП, при этом допуская реализацию, работающую в реальном времени.

6. Улучшение алгоритмов масштабирования изображений. Как правило, видео на компьютере просматривают во весь экран. При этом даже применение очень простого и быстрого кусочно-линейного масштабирования способом кардинально снизить скорость проигрывания ролика, т.е. примитивная операция масштабирования изображения будет работать заметно дольше, чем сложный алгоритм декодера. Однако растущие скорости современных компьютеров позволяют использовать более сложные и качественные алгоритмы масштабирования на весь экран, получая более высокое качество, чем для использовавшейся ранее билинейной интерполяции (в большинстве видеокарт реализованной аппаратно).

6.4. СТАНДАРТЫ СЖАТИЯ ВИДЕОДАНЫХ

Motion-JPEG (или M-JPEG) является наиболее простым алгоритмом сжатия видеоданных. В нём каждый кадр сжимается независимо алгоритмом JPEG. Этот приём даёт высокую скорость доступа к произвольным кадрам как в прямом, так и в обратном порядке следования. Соответственно легко реализуются плавные "перемотки" в обоих направлениях, аудиовизуальная синхронизация и, что самое главное, редактирование. Типичные операции JPEG сейчас поддерживаются на аппаратном уровне большинством видеокарт, и данный формат позволяет легко оперировать большими объёмами данных при монтаже фильмов. Независимое сжатие отдельных кадров позволяет накладывать различные эффекты, не опасаясь, что взаимное влияние соседних кадров внесёт дополнительные искажения в фильм.

К недостаткам можно отнести сравнительно низкую степень сжатия.

MPEG-1. Алгоритм MPEG-1 в целом соответствует описанной выше общей схеме построения алгоритмов сжатия.

К достоинствам стандарта можно отнести простую аппаратную реализацию, а к недостаткам – невысокую степень сжатия и малую гибкость формата.

H.261. Стандарт H.261 специфицирует кодирование и декодирование видеопотока для передачи по каналу рх64 Кбит, где $p = 1...30$. В качестве канала может выступать, например, несколько телефонных линий.

Входной формат изображения – разрешения CIF или QCIF в формате YUV (CCIR 601), частота кадров от 30 fps и ниже. Используется уменьшение разрешения в 2 раза для компонент цветности.

В выходной поток записываются два типа кадров: INTRA – сжатые независимо (соответствуют I-кадрам) и INTER – сжатые со ссылкой на предыдущий кадр (соответствуют P-кадрам). В передаваемом кадре не обязательно присутствуют все макроблоки изображения; если блок изменился незначительно передавать его обычно нет смысла. Сжатие в INTRA-кадрах осуществляется по схеме сжатия отдельного изображения. В INTER-кадрах производится аналогичное сжатие разности каждого передаваемого макроблока с "наиболее похожим" макроблоком из предыдущего кадра (компенсация движения). Для сглаживания артефактов ДКП предусмотрена возможность применения размытия внутри каждого блока 8×8 пикселей. Стандарт требует, чтобы INTRA-кадры встречались в потоке не реже чем через каждые 132 INTER-кадра (чтобы не накапливалась погрешность кодирования и была возможность восстановиться в случае ошибки в потоке).

Степень сжатия зависит в основном от метода нахождения "похожих" макроблоков в предыдущем кадре, алгоритма решения, передавать ли конкретный макроблок, выбора способа кодирования каждого макроблока (INTER/INTRA) и выбора коэффициентов квантования результатов ДКП.

Ни один из перечисленных вопросы стандартом не регламентируются, оставляя свободу для построения собственных оптимальных алгоритмов.

H.263. Данный стандарт является расширением, дополнением и значительным усложнением H.261. Он содержит "базовый" стандарт кодирования, практически не отличающийся по алгоритмам сжатия от H.261, плюс множество опциональных его расширений.

Перечислим наиболее важные отличия.

1. Использование арифметического кодирования вместо кодов Хаффмана. Даёт возможность на 5...10 % повысить степень сжатия.

2. Возможность задания векторов смещения, указывающих за границы изображения. При этом граничные пиксели используются для предсказания пикселей вне изображения. Данный приём усложняет алгоритм декодирования, но позволяет значительно улучшить изображение при резкой смене плана сцены.

3. Возможность задания вектора смещения для каждого блока 8×8 в макроблоке, что в ряде случаев существенно увеличивает сжатие и снижает блочность изображения.

4. Появление В-кадров, которые позволяют увеличить степень сжатия за счёт усложнения и увеличения времени работы декодера.

5. Поддержка большого числа форматов входных видеоданных: sub-QCIF, QCIF, CIF, 4CIF, 16CIF и отдельно настраиваемых. Основное отличие от более универсальных форматов заключается в адаптации для нескольких фиксированных разрешений, что позволяет делать менее универсальные, но более быстрые процедуры обработки кадров. Построенный таким образом декодер работает несколько быстрее.

6. Компенсация движения с субпиксельной точностью. Возможность сдвинуть блок на полпиксела также увеличивает степень сжатия, но увеличивает время работы декодера. Особый режим сжатия INTRA макроблоков со ссылкой на соседние макроблоки в обрабатываемом кадре, особый режим квантования и специальная таблица Хаффмана для улучшения сжатия I-кадров в ряде случаев.

7. Сглаживание границ блоков декодированного изображения для уменьшения эффекта "блочности". Зачастую при резком движении в кадре при сжатии алгоритм оказывается вынужден повысить степень квантования блоков после ДКП, чтобы уложиться в отведённый на передачу битовый поток. При этом в кадре возникают хорошо знакомые по JPEG блоки размером 8×8 . Как показала практика, "сращивание" границ, когда крайние пиксели блоков сдвигают по яркости так, чтобы уменьшить разницу, позволяет зачастую заметно повысить визуальное качество фильма. Изменение разрешения и деформирование базового кадра, использующегося в качестве базового при сжатии.

8. Различные режимы квантования и кодирования по Хаффману.

MPEG-2 сжимает оцифрованное видео при потоке данных от 3 до 10 Мбит/с. Многие в нём заимствовано из формата CCIR-601. CCIR-601 представляет собой стандарт цифрового видео с размером передаваемого изображения 720×486 при 60 полукадрах в секунду. Строки изображения передаются с чередованием, и два полукадра составляют кадр. Этот приём нередко применяют для уменьшения мерцания.

Хроматические каналы (U и V в YUV) передаются размером 360×243 60 раз в секунду и также чередуются, уже между собой. Подобное деление называется 4:2:2. Перевод из CCIR-601 в MPEG-I прост: делится в 2 раза яркостная компонента по горизонтали, делится поток в 2 раза во временном измерении (убрав чередование), добавляется вторая хроматическая компонента и удаляются "лишние" строки, чтобы размер по вертикали делился на 16. Получается поток YUV кадров размером 352×240 с частотой 30 кадров в секунду.

Если же необходимо довести качество изображения до CCIR-601, то, чтобы исключить размывание объекта и артефакты, применяют деинтерлейсинг и архивацию чётных и нечётных кадров в потоке CCIR-601 независимо. Конечно, потери степени сжатия здесь неизбежны.

MPEG-4 кардинально отличается от принимаемых ранее стандартов.

Рассмотрим наиболее интересные и полезные нововведения.

1. Расчёт трёхмерных сцен и работа с синтетическими объектами.

В состав декодера MPEG-4 как составная часть входит блок визуализации трёхмерных объектов (Animation Framework extension (AFX) – то, что в просторечии называют данными для трёхмерного движка). Те, кто кодировал видео, знают, сколько проблем доставляют титры и вообще любые накладываемые поверх фильма объекты (логотипы, заставки и т.п.).

В MPEG-4 предлагается накладываемые объекты рассчитывать отдельно и затем накладывать. Кроме того, можно использовать видеопоток даже как текстуру, накладываемую на поверхности рассчитываемых объектов.

Такая гибкая работа с трёхмерными объектами позволяет существенно поднять степень сжатия при заметно лучшем качестве изображения. Более того, никто не мешает делать видеоролики вообще без живого видео, а состоящие только из рассчитанных (синтетических) объектов. Размер их описания будет в разы меньше, чем размер аналогичных фильмов, сжатых просто как поток кадров. Кстати, отдельно в стандарте предусмотрена работа со "спрайтами" – статическими изображениями, накладываемыми на кадр. При этом размер спрайта может быть как совсем маленький (логотип канала в уголке экрана), так и превышать размер кадра и "прокручиваться". Это даёт значительную гибкость при создании MPEG-4-фильмов и позволяет заметно уменьшить объём кодируемой информации.

2. Объектно-ориентированная работа с потоком данных. Теперь работа с потоком данных становится объектно-ориентированной. При этом данные могут быть живым видео, звуковыми данными, синтетическими объектами и т.д. Из них создаются сцены, этими сценами можно управлять.

3. Помещение в поток двоичного кода "C++ подобного" языка BIFS.

С помощью BIFS в поток добавляются описания объектов, классов объектов и сцен. Также на нём можно менять координаты, размеры, свойства, поведение и реакцию объектов на действия пользователя. В своё время Flash был назван революцией 2D графики в Интернете. Аналогичный прорыв в области видео совершает MPEG-4.

4. Активная зрительская позиция. Как было замечено выше, BIFS позволяет задавать реакцию объектов сцены на действия пользователя. Потенциально возможно удаление, добавление или перемещение объектов, ввод команд с клавиатуры. Событийная модель заимствована из развивавшегося уже долгое время языка моделирования виртуальной реальности VRML.

5. Синтезатор лиц и фигур. В стандарт заложен интерфейс к модулю синтеза лиц и фигур. Например, в файле сохраняются ключевые данные о профиле лица и текстуры лица, а при записи фильма сохраняются только коэффициенты изменения формы. Для передач типа новостей этот приём позволяет в десятки раз сократить размер файла при замечательном качестве.

6. Синтезатор звуков и речи. Помимо синтеза лиц в стандарт MPEG-4 также заложены алгоритмы синтеза звуков и даже речи.

7. Улучшенные алгоритмы сжатия видео. В стандарте предусмотрены блоки, отвечающие за потоки 4.8-65 Кбит/с с прогрессивной разверткой и большие потоки с поддержкой чересстрочной развертки. Для передачи по ненадежным каналам возможно использование помехоустойчивых методов кодирования (за счёт незначительного увеличения объёма передаваемых данных резко снижается вероятность искажения изображения). При передаче видео с одновременным просмотром заложена возможность огрубить изображение, если декодер из-за ограничений канала связи не успевает получить всю информацию. Всего в стандарт заложено три уровня детализации.

Эта возможность позволит легко адаптировать алгоритм для трансляций видео по сети.

8. Поддержка профилей на уровне стандарта. Конечно, что реализация всех возможностей стандарта превращает декодер в весьма сложную и большую конструкцию. При этом далеко не для всех приложений необходимы какие-то сложные специфические функции (например, синтез речи).

Создатели стандарта оговорили наборы профилей, каждый из которых включает в себя набор обязательных функций. Если в фильме записано, что ему для проигрывания необходим такой-то профиль и декодер этот профиль поддерживает, то стандарт гарантирует, что фильм будет проигран правильно.

6.5. КОМПЬЮТЕРНЫЙ МОНТАЖ ВИДЕО

Монтаж – это процесс "сборки" фильма из отдельных элементов – кадров. Но вырезать неудачные места и склеить оставшиеся ещё не значит "смонтировать". Грамотный монтаж, даже в самых простых фильмах, состоящих из одного эпизода, предполагает соблюдение целого набора правил. Правила эти основаны на некоторых физиологических законах восприятия зрительной и звуковой информации. Они выработаны чисто эмпирическим путем на протяжении первых двух-трех десятилетий существования кино и с тех пор не претерпели существенных изменений.

Когда речь идёт о монтаже одной сцены, задача автора добиться того, чтобы зрителю было понятно, что происходит на экране и где разворачивается действие. При этом в большинстве случаев весьма желательно, чтобы при просмотре зритель не замечал того, что эта сцена состоит не из одного, а из нескольких склеенных между собой кадров.

6.5.1. Терминология

1. *Кадр*. Съёмочный кадр или план – любой участок исходной видеоленты, с записью от нажатия кнопки RECORD до паузы, следующее нажатие – начинается следующий съёмочный кадр.

Монтажный кадр или план – элемент смонтированного видеофильма – то, что осталось от съёмочного кадра после того как его "подрезали" и вставили в нужное место.

Монтажный лист – описание исходного материала или готового фильма с последовательным указанием содержания каждого кадра и его координат на плёнке (по счётчику).

2. *План*. Термин "план" в кинематографии означает размер и положение объекта в кадре.

Определим планы в порядке уменьшения масштаба:

1) сверхкрупный план или Деталь – кадр, в котором помещается снятый элемент на объекте (например, только глаза).

2) крупный план – кадр, в котором голова человека занимает почти все место;

3) первый средний план – человек по пояс;

4) второй средний план – человек по колени;

- 5) общий (театральный) – человек в полный рост;
- 6) дальний план – человек занимает не более 1/10 высоты кадра.

3. *Ракурсы*. Ракурс обозначает положение камеры по отношению к объекту. Важно помнить, что ракурс – это, в первую очередь, позиция, с которой зритель будет смотреть на сюжет.

Съёмка с нижней точки позволяет подчеркнуть масштаб снимаемого объекта, с верхней точки – напротив, визуально уменьшает объект.

Героя, снятого камерой, которая расположена на уровне его глаз, зритель воспринимает как равного. Такую точку съёмки в большинстве случаев используют при съёмке людей.

4. *Освещение*. Важнейший инструмент, который создаёт изображение – это, конечно же, свет.

Всего различают пять видов освещения.

Рисующее освещение – направленный световой поток, подчеркивающий рельеф и объём деталей, но формирующий резкие, жёсткие тени. Такое освещение создают, к примеру, пучок солнечных лучей или искусственный источник, дающий узкий луч.

Заполняющее освещение создаётся протяжённым источником и поэтому практически не образует теней. Пример – естественное освещение в облачную погоду, когда источником света становится всё небо.

Моделирующее освещение даёт прямой световой луч, направленный на объект. Моделирующее освещение чрезвычайно полезно, если необходимо смягчить жесткие тени или создать эффектные блики на блестящих поверхностях.

Контровое освещение даёт свет, падающий на объект съёмки сзади. Этот вид освещения обычно применяется, когда необходимо отделить объект съёмки от фона. Он чрезвычайно эффектен на крупных планах, когда дополнительный источник света создаёт "ореол" вокруг головы героя.

Фоновое освещение. В этом случае освещён фон за главным объектом съёмки. Этот вид освещения бывает полезен в излишне контрастных сценах, а также там, где необходимо создать настроение, придав фону, например, красный оттенок для создания ощущения жара.

6.5.2. Сценарий

Первоосновой видеопленки является творческий замысел автора, его идея, изложенная в тщательно разработанном плане будущего фильма – *сценарии*, а затем выраженная в кинематографических образах.

6.1. Таблица сценария

№ кадра	Содержание кадра	План	Время, с	Операторская экспликация	Звуковая экспликация
1	Открытие торгового центра (начало)	Общий	8	Панорама	Торжественная музыка
...

Фильм состоит из эпизодов, представляющих собой завершённый круг событий, имеющий отчетливые начало, продолжение и конец. Поэтому, разбивая на эпизоды предварительно написанный сценарий и учитывая, что впоследствии фильм нужно будет монтировать, уже во время съёмки нужно стремиться снимать законченные эпизоды.

Сценарий можно представлять в виде таблицы, содержащей следующую информацию, помещённую в табл. 6.1.

6.5.3. Монтаж

Монтировать фильм можно по-разному. Иногда сначала подбирается звукоряд, при любительском видео – фоновая музыка. Когда же логическую последовательность диктует видеоматериал, сначала собирается по смыслу история, а потом под неё уже пишется текст и подбирается музыкальное сопровождение.

Технически монтаж представляет собой расстановку кадров в определенном порядке и организацию переходов между ними. Считается, что оптимальная длительность каждого смонтированного кадра – не более 5...6 с.

Существует несколько основных принципов монтажа.

1. *Монтаж по крупности*. Особенности человеческого зрения таковы, что он легко и без напряжения воспринимает следующие друг за другом кадры, которые достаточно сильно различаются по крупности. При этом соседние кадры не должны кардинально отличаться по этому параметру, иначе зритель будет чувствовать себя некомфортно.

Многолетний опытом было установлено, что наиболее гладко воспринимается стык между планами, находящимися на приведённой выше шкале через один, т.е. общий план монтируется с первым средним и

наоборот, второй средний с крупным, и т.д. Исключения: крупный план монтируется с деталью, общий план с дальним.

2. *Монтаж по ориентации в пространстве.* Иначе этот принцип называют "монтажом по взгляду". Простейший случай монтажа по положению объектов в пространстве – монтаж диалога двух персонажей (так называемая "восьмёрка"). Если взгляды этих людей будут направлены навстречу, кадры монтируются, если в одну сторону – нет. В общем случае правило формулируется так: съёмка двух взаимодействующих объектов должна производиться строго по одну сторону от линии их взаимодействия. Линия взаимодействия – это воображаемая линия, проходящая через оба объекта.

3. *Монтаж по направлению движения.* В ситуациях, когда нужно монтировать кадры с движущимися объектами, нужно помнить следующее: изменение направления движения объекта на стыке кадров не должно быть больше чем на 90 градусов. При этом не должна пересекаться вертикальная ось, т.е. если на одном кадре объект движется налево, то в следующем кадре он не должен двигаться направо. Один из важных выводов, который должен сделать для себя начинающий видеолюбитель – нельзя склеивать горизонтальные панорамы, снятые в разных направлениях. Профессиональные операторы всегда начинают и заканчивают съёмку панорам короткими статичными кусками. Это даёт возможность монтировать панорамы через небольшую паузу в движении.

4. *Монтаж по фазе движения.* Фазу приходится учитывать при монтаже циклически повторяющихся положений объекта. Это может быть идущий человек, велосипедист, вращающий педали, жонглер в цирке и т.п. В игровом кино многие эпизоды снимают несколько раз не только для того, чтобы иметь несколько дублей одного кадра. Одно и то же действие, как правило, снимается несколько раз планами разной крупности. Монтажер, таким образом, имеет возможность на *монтажном столе* подгонять каждое движение по фазе с точностью до кадра и иногда при монтаже нужно прибегать к помощи промежуточных кадров, чтобы избежать "сдвига по фазе".

5. *Монтаж по композиции.* Начинающие видеолюбители предпочитают центральную композицию кадра, т.е. располагают главный объект по центру. Такая компоновка кадра очень распространена, и без нее не обойтись ни в кино, ни на телевидении. Однако, часто для большей выразительности, а иногда и по необходимости главный объект съёмки располагается не по центру кадра. И тогда при монтаже возникает необходимость согласования соседних кадров по композиции. В качестве примера можно рассмотреть ситуацию, когда нужно смонтировать общий и средний план человека. Если на общем плане человек заметно смещён в одну сторону кадра, а на среднем в другую, при просмотре в месте склейки зритель на время потеряет из вида объект из-за резкого смещения центра внимания. Чтобы этого не произошло, во время съёмки и при монтаже нужно помнить ещё одно правило: смещение центра внимания по горизонтали при переходе от кадра к кадру не должно превышать одной трети ширины экрана. Ситуация, когда может возникнуть резкое смещение центра внимания по вертикали, встречается значительно реже, но и тут действует аналогичное правило.

6. *Монтаж по свету.* Принцип такого монтажа прост – очень тёмные и очень светлые кадры не должны соседствовать: человеческий глаз должен привыкнуть к уровню освещённости кадра. Это относится и к ситуации, когда, например, дальний план снят при солнечном освещении, а общий или средний при пасмурном.

"Склеивать" контрастное по свету видео следует через промежуточные кадры. Если такие планы заблаговременно не сняты, можно попробовать слегка осветлить тёмные или затемнить светлые кадры.

7. *Монтаж по цвету.* Рассмотренный выше монтаж по свету является частным случаем *монтажа по цвету*. Обобщая вывод, сделанный по отношению к тону и характеру освещения объекта, можно легко перейти к следующему правилу: соседние кадры в месте стыка не должны резко отличаться по цвету. Если в новом кадре возникают новые цвета, то они должны занимать не более одной трети площади кадра.

8. *"Перебивка"*. Рассмотрены основные, но далеко не все правила монтажа двух соседних кадров. Но даже при соблюдении этих правил могут возникать и возникают проблемы нестыковки. На этот случай есть одно универсальное средство, которое называется *перебивкой*.

Перебивка – это кадр, который вклеивается между двумя другими кадрами, связанными между собой единством объектов и места действия.

Содержание перебивки всегда резко отличается от предыдущего и следующего за ней кадров, но оно должно быть прямо или косвенно связанным с основным содержанием. Наиболее часто перебивки применяются на телевидении при монтаже длинных монологов, например, если нужно сократить часть выступления человека, снятого длинным статичным средним планом. Для вставки перебивки выбирается место примерно за одну-две секунды до того как говорящий делает небольшую паузу. К среднему плану клеится перебивка, во время которой звучит конец фразы. Следующий кадр – опять средний план героя, который начинает говорить с нужного редактору места.

Перебивкой в этом случае может служить кадр, на котором снята картина, висящая на стене кабинета, часы или книга на письменном столе и т.п. Однако, если в монологе речь идёт именно о той картине или о том, что на ней изображено, то кадр с картиной уже нельзя назвать перебивкой. Он становится вполне самостоятельным смысловым кадром.

Контрольные вопросы

1. Приведите общую схему сжатия видеоданных.
2. Кадры каких типов участвуют в процессе сжатия видеоданных?
3. Приведите форматы сжатия видеоданных.
4. Что такое видеомонтаж?
5. Какие виды монтажа вы знаете?

7. АНИМАЦИЯ

Термин "анимировать" дословно означает "оживить" изображение. Теория анимации базируется на положении о способности человеческого глаза сохранять на сетчатой оболочке след увиденного и соединять быстро меняющиеся изображения в единый зрительный ряд. Это создаёт иллюзию непрерывного движения. С точки зрения физиологии человека, минимальная частота смены изображений, при которой зритель воспринимает изменения объектов как плавные и эластичные, называется нижней границей непрерывного восприятия зрительного ряда. Верхняя граница при этом определяется реакцией мозга человека на происходящие изменения, способностью при данной частоте смены изображений понимать смысл воспроизводимого события.

Анимация – это демонстрирующаяся в быстром темпе последовательность рисованных изображений-кадров, каждый из которых несколько отличается от предшествующего ему и следующего за ним. В этих кадрах последовательно изменяются позиции объекта анимации, что создаёт иллюзию движения.

Эти обстоятельства учитываются при визуальном воспроизведении динамических процессов с помощью различных технических средств.

Частота смены кадров за секунду экранного времени составляет:

- 12–16 – для компьютерной анимации, в зависимости от использования различных пакетов программного обеспечения;
- 24 – для кинематографа;
- 25 – для системы PAL телевидения;
- 30 – для системы NTSC телевидения.

7.1. МЕТОДЫ АНИМАЦИИ

Прежде всего, необходимо разделить анимацию на группы (рис. 7.1).

1. Классическая анимация – метод, представляющий собой очередную смену рисунков, каждый из которых нарисован отдельно (принцип мультфильма), очень трудоёмкий из-за необходимости создания каждого рисунка.

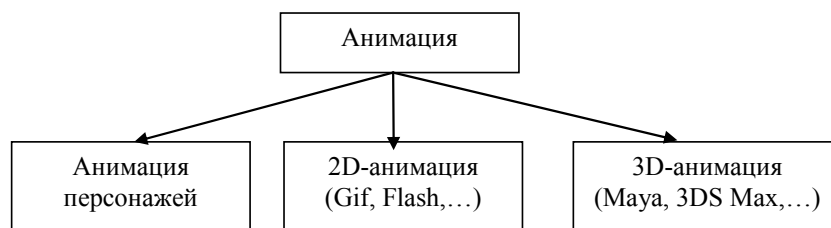


Рис. 7.1. Виды анимации

2. Кукольная анимация. В пространстве размещаются объекты – кадр фиксирует их положение, положение объектов меняется – и опять фиксируется следующим кадром.

3. Спрайтовая анимация – анимация, реализуемая при помощи языка программирования или специального инструментального средства. В спрайтовой анимации отсутствует понятие кадра (принцип подвижных игр). Почти всегда базируется на работе с "прозрачным" цветом.

4. Морфинг – преобразование одного графического образа в другой. Часто выполняется программно. Программное обеспечение морфинга генерирует заданное число промежуточных кадров, которое обеспечивает плавный переход начального образа в конечный.

5. Анимация цветом – положение объектов не изменяется, меняется лишь цвет. Часто выполняется программно.

6. 3D-анимация – создаётся с помощью специальных программ (3D Studio MAX, PovRay, LightWave, Maya, ...).

Картинка получается путём визуализации сцены. Каждая сцена представляет собой набор:

- объектов;
- источников света;
- текстур;

– камер (хотя обычно используется одна).

7. Метод ключевых или опорных кадров (keyframing) – является наиболее распространённым способом создания анимации. Ключевым событием может являться не только изменение параметров одного из возможных преобразований объекта (положения, поворота или масштаба), но также изменение любого из допускающих анимацию параметров (свойства источников света, материалов и др.). После определения всех ключевых кадров система компьютерной анимации выполняет автоматический расчёт событий анимации для всех остальных кадров, занимающих промежуточное положение между ключевыми – промежуточных кадров (рис. 7.2).

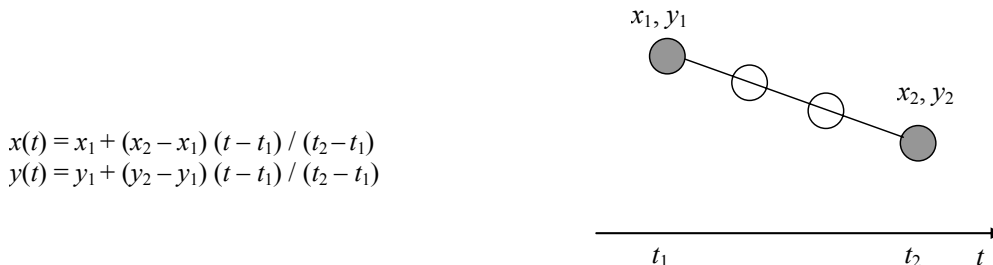


Рис. 7.2. Метод опорных кадров

8. Процедурная анимация – используется для моделирования движений, или эффектов, которые трудно воспроизвести с помощью ключевых кадров. В процедурной анимации рассчитывают текущие значения параметров анимации, основываясь на начальных значениях, заданных пользователем, и на математических выражениях, описывающих изменение параметров во времени. Этот метод позволяет выполнять качественные анимации. Часто процедурная анимация используется для разнообразных физических эффектов.

9. Инверсная (обратная) и прямая кинематика. Инверсная кинематика – движение задаётся перемещением самого младшего объекта-потомка, что заставляет всю остальную цепочку перемещаться в соответствии с ограничениями на работу сочленений объектов. В частности, это могут быть ограничения на вращение и на скольжение. Можно ограничить диапазон действия этих сочленений любыми осями координат, размером углового сектора или расстоянием. Выполняя настройки параметров сочленений, таких как приоритетность, наличие и сила трения и т.п., можно добиться построения реалистичных движений для сложных многозвенных объектов. В отличие от метода прямой кинематики, метод обратной кинематики допускает получение нескольких решений при наличии множества сочленений объектов.

10. Захват движения (Motion Capture) – это новое направление в анимации, которое даёт возможность передавать естественные, реалистичные движения в реальном времени. Маленькие легкие датчики прикрепляются на живого актёра в тех местах, которые будут приведены в соответствие с контрольными точками компьютерной модели для ввода и оцифровки движения. Координаты актёра и его ориентация в пространстве передаются графической станции, и анимационные модели оживают.

7.2. ДВЕНАДЦАТЬ ПРИНЦИПОВ АНИМАЦИИ

Рассматривая классическую анимацию, нельзя не упомянуть об основных принципах анимации, сформированных У. Диснеем.

1. Принцип сжатия и растяжения. Самой главной проблемой была не просто имитация движения, а достижение максимальной реалистичности этого движения. Дисней, рисуя своих героев, старался делать их максимально пластичными, сравнивая с движениями живых существ. Живое тело во время движения постоянно сжимается и растягивается. Основная идея применения этого принципа – чтобы объём тела всегда оставался постоянным, т.е. если героя "растянули" по вертикали, по горизонтали его тело обязательно сжимается. Открытие этого принципа стало поистине революционным – теперь рисованные герои не выглядели как деревянные марионетки.

2. Обязательное "отказное" движение. В реальной жизни для произведения какого-либо действия человеку часто приходится делать подготовительные движения. Например, перед прыжком человеку необходимо присесть для того, чтобы бросить что-либо – завести руку назад. Такие действия называются отказными движениями, так как перед тем, как сделать что-то, персонаж как бы отказывается от действия. Такое движение подготавливает зрителя к последующему действию персонажа и придаёт инерцию движениям.

3. Сценичность. Для правильного восприятия персонажа зрителями все его движения, позы и выражения лица должны быть предельно просты и выразительны. Сутью этого принципа является воздействие на зрителя, вызов ответной эмоциональной реакции: персонаж должен быть ясно виден зрителю, все его движения должны быть открыты взгляду, а позы выразительны.

4. Сквозное движение и захлёт. Люди, животные, да и вообще все живые существа никогда не начинают движение всем телом одновременно, всегда присутствует последовательность движений, т.е. при взмахе рукой вначале движется предплечье, локоть, плечо, запястье, кисть. Так же обстоят дела с поворотом головы, ходьбой, прыжками и так далее. Суть принципа состоит в том, что движение никогда не должно прекращаться. Для графического выражения такой последовательности одного движения, аниматоры устанавливают что-то вро-

де жёсткой очерёдности, выбирая в качестве начальной точки какую-то отдельную часть тела, и дальше уже прописывают чёткую последовательность движения каждого элемента. Такая очерёдность называется "сквозным движением". Сквозное движение обеспечивает непрерывность движения и плавность перехода фаз, например, из бега в шаг и наоборот. Движение отдельных элементов объекта в то время, когда сам элемент уже перестал двигаться, называется захлестом.

5. От позы к позе. Ранее художники-аниматоры рисовали движения персонажей покадрово, следуя сюжету. Такой метод был весьма длительным, сложным и трудоемким. Темпы работы резко увеличились после внедрения метода "от позы к позе": когда аниматор вначале рисует основные моменты движения и располагает персонажа на сцене, а уже после основной компоновки прорисовывает все кадры движения.

6. Медленный вход и медленный выход. Чем больше кадров оформляют основную позу, тем выразительнее становится само движение. Рисованный персонаж как бы "скользит" от одного движения к другому, в результате внимание зрителей фиксируется именно на основном движении, при этом нет "разрыва" в перемещениях персонажа.

7. Движение по дугам. До этого, буквально революционного открытия Уолта Диснея, художники рисовали ходьбу или бег своих персонажей "по прямой", отчего последние смотрелись как роботы. Кажется, что при резких движениях этот принцип не соблюдается, т.е. движение идёт по прямой. Однако это только видимость, так как даже в самых резких движениях траектории имеют дугообразный характер, хотя и более приближенный к прямой.

8. Второстепенные действия – призваны придать рисованному персонажу больше жизненности и выразительности, а также часто помогают составить представления о его характере. Вторичные действия получили широкое распространение в мировой анимации. Благодаря их использованию персонажи становятся более живыми и эмоциональными.

9. Преувеличение – этот принцип Дисней также активно использовал, часто ударяясь в карикатурность при создании персонажей. Преувеличение используется, прежде всего, для передачи эмоций героя: резкие движения, не обусловленные сюжетной необходимостью, смех, крик и т.д.

10. Расчёт времени. Чтобы "вдохнуть" в нарисованного героя жизнь, сделать его близким зрителю, аниматоры используют такой приём, как "расчёт времени". Это позволяет сделать медленными или быстрыми движения героя, чтобы придать его рисованному телу вес и передать его настроение. Вес персонажа складывается из таких факторов как скорость перемещения и инертность. Для того чтобы персонаж двигался в соответствии со своим весом, художник рассчитывает время движения и захлёста. При расчёте времени учитываются вес, инертность, объём и эмоциональное состояние персонажа.

11. Привлекательность персонажа – путь к успеху всего фильма. Фактически, все вышеперечисленные принципы "работали" именно на этот: сделать интересным для зрителя даже самого неприятного персонажа, заставить любоваться героями, искать в них что-то.

12. Профессиональный рисунок – основа всего, аниматоры всего мира очень ревностно относятся к профессиональному подходу в рисовании анимационных лент.

7.3. ФОРМАТЫ АНИМАЦИОННЫХ ФАЙЛОВ

7.3.1. Graphic Interchange Format (GIF)

GIF (GIF89a) – наиболее распространённый формат веб-анимации, как правило, именно он используется для баннеров.

Формат **GIF** был разработан в 1987 году (GIF87a) фирмой CompuServe для упрощения обмена данными в локальных компьютерных сетях, при возможности отображения этих данных. В 1989-м формат был модифицирован (GIF89a), были добавлены поддержка прозрачности и анимации. Этот формат является одним из наиболее употребительных растровых форматов в электронных, в особенности сетевых, изданиях.

Основные достоинства формата:

- пригодность для различных платформ, т.е. формат является платформо-независимым;
- малый размер файлов благодаря использованию мощного алгоритма сжатия без потерь;
- поддерживается практически всеми версиями браузеров;
- не требует постоянной связи с сервером;
- не требуются специальных программ для просмотра (в отличие от Flash);
- не требуются больших ресурсов клиентской машины (в отличие от Java-апплетов);
- возможно сохранение с чередованием;
- позволяет применять эффект прозрачности.

Основные недостатки формата:

- палитра не превышает 256 цветов;
- сжатие фотографических изображений гораздо менее эффективно, чем в формате JPEG.

Благодаря основным своим достоинствам, таким как совместимость со всеми браузерами и небольшой информационный объём, анимационные файлы в формате GIF занимают почётное место на Web-страницах. В электронных изданиях любого типа они также используются достаточно широко.

Изображение записывается в этом формате с использованием RGB-цветовой модели и данных встроенной в файл палитры индексированных цветов. Серьёзным ограничением для этого формата является ограниченная глубина цвета, не превышающая 8 бит на пиксел. Существенный недостаток GIF-файлов связан с применением индексированных цветов, для чего в файле используется глобальная и локальные цветовые палитры. Глобальная цветовая палитра хранит до 256 различных цветовых оттенков, каждый из которых может быть использован в любом из изображений, которое хранится в данном файле. Локальные палитры относятся к каждому отдельному изображению, т.е. хранимые в них цветовые оттенки не могут использоваться в других (не своих) изображениях.

Каждое такое изображение формирует отдельный кадр, причём задержка следующего кадра и его линейное смещение относительно предыдущего по каждой координате может регулироваться. Разрешение для всех изображений, входящих в данный файл, или количество пикселей по каждой координате должно в каждом файле поддерживаться постоянным.

Общий формат файла приведён на рис. 7.3. Файл начинается с общего заголовка и дескриптора логического экрана. *Идентификатор GIF* – это специальная "подпись", указывающая, что последующие данные являются действительно потоком данных изображения в формате GIF. Эта "подпись" состоит из следующих шести символов: G I F 8 7 а. Три последних символа '87а' могут рассматриваться как номер версии для данного конкретного определения GIF.

Дескриптор экрана описывает общие параметры для всех последующих изображений в формате GIF. Он определяет размеры пространства изображения или требуемого логического экрана, существование информации о "глубине" экрана. Там же задаётся размер глобальной цветовой таблицы, которая может и отсутствовать. В этом случае обязательно используется для каждого отдельного изображения локальная палитра.

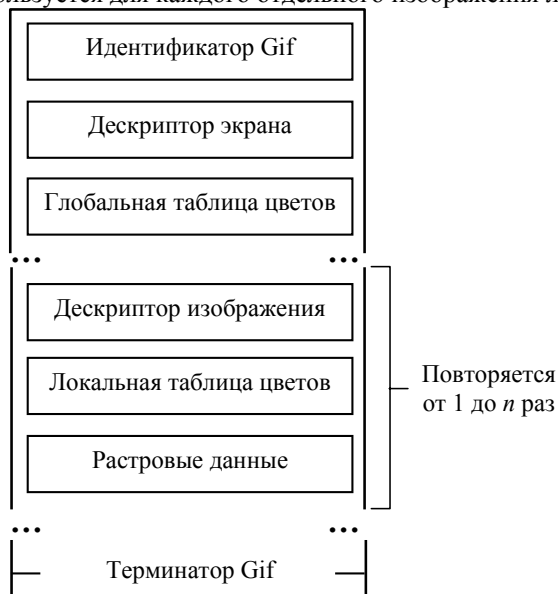


Рис. 7.3. Формат Gif-файла

В большинстве случаев рекомендуется пользоваться именно глобальной палитрой, что экономит общее информационное пространство, занимаемое файлом. Вся эта информация запоминается в виде серии из 8 байтов, как показано на рис. 7.4.

Ширина и высота логического экрана могут быть больше размеров физического экрана. Способ высвечивания изображений больших, чем размеры физического экрана, зависит от реализации и может использовать преимущества конкретного оборудования (например, окна скроллинга в Macintosh scrolling windows). В противном случае изображение будет усечено по краям экрана.

Значение "pixel" также определяет число цветов в изображении. Диапазон значений "pixel" составляет от 0 до 7, что соответствует от 1 до 8 битам. Это значение транслируется в диапазон от 2 (чёрно-белые изображения) до 256 цветов. Бит 3 в байте 5 зарезервирован для будущих определений и должен быть нулевым.

Глобальная таблица цветов является необязательной и рекомендуется для изображений, где требуется точная передача цветов. На существование этой таблицы указывает поле "M" в байте 5 дескриптора экрана. Цветовая таблица может быть также связана с каждым изображением в GIF-файле, что будет описано позже. Флаг "M" в дескрипторе конкретного изображения обычно равен 0. Если глобальная таблица цветов присутствует, её определение следует непосредственно за дескриптором экрана.



Рис. 7.4. Формат дескриптора экрана Gif-файла:

ширина и высота экрана – ширина и высота раstra в пикселях; M = 1, за дескриптором следует глобальная таблица цветов; cg+1 – число бит цветового разрешения; pixel+1 – число бит/пиксель в изображении; Background – цветовой индекс фона экрана

Число элементов цветовой таблицы, следующей за описателем экрана, равно $2^{*}(\text{число бит/пиксел})$, причем каждый элемент состоит из трёх байтов, значения которых описывают соответственно относительную интенсивность красного, зелёного и синего цветов.

Получаемое значение каждого пикселя при высвечивании изображения будет соответствовать ближайшему доступному цвету из цветовой таблицы дисплея. Цветовые компоненты представляют собой значение относительной интенсивности от нулевой (0) до полной (255). Белый цвет может быть представлен как (255,255,255), черный как (0,0,0) и жёлтый как (180,180,0). При высвечивании на дисплеях, которые поддерживают менее 8 бит на цветовую компоненту, используются старшие биты. При создании элементов цветовой таблицы GIF на аппаратуре, поддерживающей менее 8 бит на компоненту, значение аппаратной компоненты должно быть конвертировано в 8-битный формат по следующей формуле:

$$\langle \text{значение_в_таблице} \rangle = \langle \text{компонента} \rangle * 255 / (2^{*} \langle \text{число_бит} \rangle - 1).$$

Это обеспечивает точный перевод цветов для всех дисплеев. В случае создания изображения GIF на аппаратуре без возможности цветовой палитры, должна быть создана фиксированная палитра на основе доступных для данного оборудования цветов. Если указано отсутствие глобальной таблицы цветов, цветровая таблица по умолчанию генерируется внутренним образом так, что каждый цветовой индекс был равен аппаратному цветовому индексу.

Дескриптор изображения определяет действительное расположение и размеры последующего изображения внутри пространства, определённого в дескрипторе экрана. Также определяются флаги, указывающие на присутствие локальной таблицы для поиска цветов и определения последовательности высвечивания пикселей. Каждый дескриптор изображения начинается с символа-разделителя изображений (рис. 7.5). Роль разделителя изображений состоит просто в синхронизации при входе в дескриптор изображения. Это желательно, если GIF-файл состоит более чем из одного изображения. Этот символ определён как шестнадцатеричное 0x2C или ", " (запятая). Как только этот символ встречается между изображениями, непосредственно за ним следует дескриптор изображения.

Любой символ, встреченный между концом предыдущего изображения и символом-разделителем изображения, игнорируется. Это позволит при последующих модификациях GIF допускать присутствие нескольких форматов и правильно игнорировать их старыми декодерами.

Описание положения и размеров экрана должно находиться внутри матрицы, определённой в дескрипторе экрана. С другой стороны, нет необходимости, чтобы изображение полностью заполняло весь экран.

биты							Номер байта	
7	6	5	4	3	2	1		0
0	0	1	0	1	1	0	0	1
Левый край							2	3
Верхний край							4	5
Ширина							6	7
Высота							8	9
M	I	0	0	0	0	0	pixel	10

Рис. 7.5. Формат дескриптора изображения Gif-файла:

- 1 байт – " ; " (символ-разделитель изображения);
- 2, 3 байты – начало изображения в пикселях относительно левого края экрана;
- 4, 5 байты – начало изображения в пикселях относительно верхнего края экрана;
- 6, 7 байты – ширина изображения в пикселях;
- 8, 9 байты – высота изображения в пикселях;
- 10 байт – M (индекс использования таблицы цветов), I (индекс форматирования изображения), pixel+1 (число бит/пиксель в изображении)

Локальная таблица цветов необязательна. Если установлен бит "M" байта 10 в дескрипторе изображения, то вслед за дескриптором изображения следует локальная таблица цветов, которая относится только к последующему изображению. После обработки изображения цветовую таблицу следует привести к той, которая была определена после дескриптора экрана. Поле "pixel" байта 10 в дескрипторе изображения используется только в том случае, если указана локальная таблица цветов. Она определяет не только размер пикселя (число битов в нем), но и число элементов последующей цветовой таблицы. Число битов на пиксель также следует восстановить к тому значению, которое было определено в дескрипторе экрана, после того, как закончится обработка изображения.

Растровые данные. Формат самого изображения определен как серия значений номеров пикселей, которые образуют изображение. Пиксели запоминаются слева направо последовательно по строкам изображения. По умолчанию строки записываются последовательно, сверху вниз. В том случае, если установлен бит "I" в байте 10 дескриптора изображения, то порядок строк при записи изображения соответствует четырехпроходному процессу. При первом проходе записывается каждая 8-я строка, начиная с верхней строки окна изображения. При втором проходе записывается каждая 8-я строка, начиная с пятой строки сверху. На третьем проходе записывается каждая 4-я строка, начиная с третьей строки окна. Четвёртый проход завершает изображение, записывая каждую вторую строку, начиная со второй строки сверху.

Значения пикселей изображения обрабатываются как цветовые индексы, указывающие на существующую таблицу цветов. В результате получается цветное значение из таблицы, которое реально воспроизводится на экране. Эти серии цветовых индексов, число которых равно ширине изображения, умноженной на высоту изображения, пропускаются через поток данных изображения GIF по одному значению на пиксель, сжимаются и упаковываются в соответствии с версией алгоритма сжатия LZW.

Малый размер GIF-файлов связан с использованием поблочного LZW-сжатия изображения, причем большинство сжимаемых блоков имеют размер 255 байтов. Каждый пиксель декодированного изображения характеризуется размером в 1 байт и содержит значение индекса цвета, т.е. положение нужного цветового тона в глобальной или локальной цветовой палитре.

Терминатор GIF. Для того чтобы обеспечить синхронизацию с окончанием файла изображения GIF, декодер GIF должен обрабатывать окончание режима GIF по символу шестнадцатеричное 0x3B или ";", найденному после окончания обработки изображения. По соглашению декодирующие программы должны делать паузу и ждать действий, указывающих, что пользователь готов к продолжению. Это может быть возврат каретки, введенный с клавиатуры, или щелчок кнопкой мыши. Для интерактивных приложений эти действия пользователя должны быть переданы в ядро программы как перевод каретки для того, чтобы вычислительный процесс мог продолжаться. Обычно декодирующая программа покидает графический режим и возвращается к предыдущему процессу.

Возможности GIF-анимации связаны с тем, что этот формат позволяет хранить в одном файле несколько различных изображений. Современная версия GIF89a, выпущенная двумя годами позднее первоначальной, добавила несколько новых возможностей и решила проблему обработки изображений, размещенных в одном файле. Для этого в описание файла был добавлен специальный блок "Управляющие расширения", который размещен сразу после трёх общих для всего файла элементов и предшествует описанию отдельных изображений в составе файла. В состав управляющих расширений входят: расширение комментариев, расширение приложений и расширение управления графикой. В последнем указана, в частности, и величина задержки кадра в сотых долях секунды, а также значение индекса прозрачности цвета, который позволяет создавать новые анимационные эффекты с помощью дополнительно включенного в файл блока управления графикой. Этот блок позволяет программе просмотра организовать взаимодействие каждого последующего изображения с текущим, что и обеспечило создание широко распространенных анимационных GIF-файлов.

Большинство современных программ-аниматоров обеспечивает подготовку анимационных файлов именно в формате GIF. Однако существуют ограничения при использовании GIF-анимации. Формат GIF89a не допус-

кает включение любого рода аудиоизображений, отдельные элементы анимации нельзя сделать интерактивными. Поэтому, чтобы создать анимацию, включающую аудио и более высокие уровни интерактивности, необходимо использовать альтернативные форматы файлов и приложения (например, Macromedia Flash).

7.3.2. FLC-анимация

Весьма популярным форматом анимационных файлов являются **флики (flic)** – **FLI, FLC** и **CEL**. Это форматы файлов анимационного пакета Animator фирмы AutoDesk, разработанные Джимом Кентом, который поведал миру о своем изобретении в марте 1993 года в журнале "The Doctor Dobb's Journal". Они позволяют проигрывать на экране компьютера подобие кинофильмов. В них не содержится звука и обеспечивается передача всего 256 цветов. Но их простота и быстрота проигрывания сделали данный формат популярным среди разработчиков игр и художников-аниматоров.

FLI Animation используется для хранения анимационных последовательностей в графических приложениях, системах САПР и компьютерных играх на платформах фирмы Intel. Имена файлов используют расширения ***.fli** (старая версия формата), ***.flc** (более поздняя версия формата, поддерживаемая IBM Multimedia Tools Series, Microsoft Video for Windows и Autodesk Animator Pro и др.). Файлы **FLI** использовались первоначально в Animator. Файлы **FLC** затем стали использоваться в Animator Pro.

Все данные в этих файлах группируются во фреймы (frame). Фрейм – это один кадр анимации, состоящий в свою очередь из блоков (chunk). Блоки файла и содержат в себе всю информацию, необходимую для проигрывания анимации. В начале блока указывается его размер и его тип, также как и в начале каждого фрейма, поэтому если тип фрейма или блока неизвестен, то данный блок или фрейм можно просто пропустить.

Для сжатия данных в flc-файлах применяется дельта-сжатие, в основе которого лежит идея сохранять только отличия одного кадра от другого. Это позволяет проигрывать файлы даже на медленных видеоадаптерах, так как надо выводить только часть изображения. Сами данные сжимаются по схеме RLE (кодирование длин серий). Причем первый фрейм содержит изображение целиком, и относительно него и строятся отличия других фреймов. Последний фрейм фильма кольцевой и служит для плавного циклического проигрывания.

Схема flc-файла приведена на рис. 7.6.

Первый фрейм обычно сжимают, используя побайтное кодирование. Последующие фреймы содержат отличия от предыдущих фреймов. Иногда первый фрейм и/или последующие фреймы не сжимаются. Имеется один дополнительный фрейм в конце FLI, который содержит различие между первым и последним фреймом. Это позволяет плавно проигрывать файл по кругу, без паузы между первым и последним фреймом. Все файлы фильма содержат данный кольцевой фрейм, даже файлы с одним фреймом.

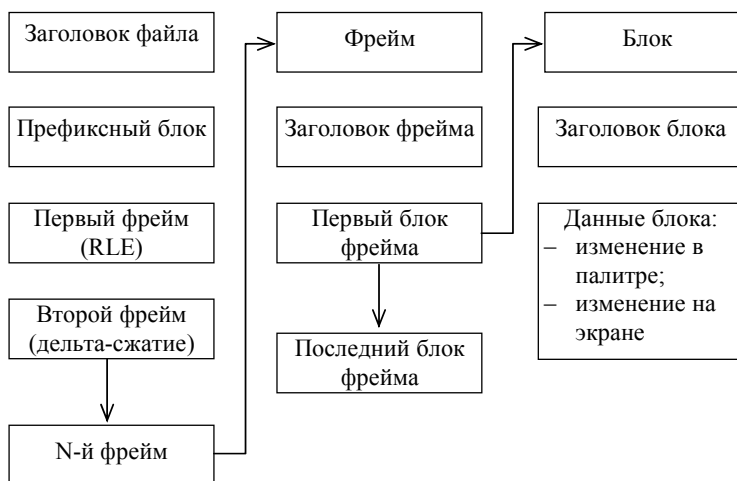


Рис. 7.6. Схема flc-файла

После заголовка идёт префиксный блок, который содержит дополнительную информацию для работы Animator Pro, непосредственно в анимации он не участвует и его можно пропустить.

Затем идут остальные фреймы. Каждый фрейм (в том числе и префиксный блок) имеет заголовок длиной 16 байт

Типов фреймов всего два. Для префиксного блока тип равен 0xF100, для фреймов, в которых содержатся данные по анимации равен 0xF1FA. Возможны другие типы фреймов, которые влияют на анимацию, но здесь они не описаны.

После заголовка фрейма идут его блоки. Вначале идёт настройка цвета, если цвета поменялись. Затем изменения на экране, если пиксели экрана поменялись. Если фрейм идентичен предыдущему, то он будет без кусков вообще и необходимо только сделать задержку.

Блок фрейма имеет свой заголовок размером 6 байт, за которым идут данные.

7.3.3. Flash-анимация (SWF)

Технология Flash основана на использовании векторной графики в формате Shockwave Flash (SWF). Хотя это далеко не первый векторный формат, создателям SWF удалось найти наиболее удачное сочетание между изобразительными возможностями графики, инструментальными средствами для работы с ней, и механизмом включения результата в Web-страницы. Дополнительным преимуществом SWF является его переносимость, т.е. этот формат может использоваться на любой аппаратно-программной платформе (в частности, на компьютерах Macintosh, работающих под управлением операционной системы MacOS, и на компьютерах IBM с ОС Windows). И ещё одна особенность SWF: созданные на его основе изображения не только могут быть анимированы, но также дополнены интерактивными элементами и звуковым сопровождением.

Переносимость и возможность создания интерактивных мультимедийных приложений обусловили быстрый рост популярности формата SWF среди Web-дизайнеров. Поэтому почти одновременно с появлением самого формата фирмой Macromedia были созданы встраиваемые компоненты (Plug-In) для двух основных браузеров Сети: Internet Explorer и Netscape Communicator. А это, в свою очередь, способствовало еще более широкому распространению SWF на просторах Всемирной Паутины. В результате разработчики этих браузеров объявили о намерении включить поддержку SWF непосредственно в ядро своих продуктов. Поддержали подобный подход и другие ведущие производители программного обеспечения (в частности, фирма Adobe).

Изначально формат SWF предназначался для создания небольшой векторной анимации и изображений для Интернета. Формат был разработкой компании Future Animation, которая в 1995 году создала небольшую (всего 3 Мбайт), но вполне революционную для своего времени программу FutureSplash Animator, предназначавшуюся для создания мультфильмов на домашнем компьютере. Эта разработка была приобретена компанией Macromedia, которая, расширив её рядом специфичных для сети возможностей, вскоре выпустила новый продукт под названием Flash (англ. "вспышка"). Идея заключалась в том, чтобы иметь формат, который будет способен работать в любой системе и в медленных сетях (например, при подключении к сети через модем).

В дальнейшем Macromedia создавала примерно по одной новой версии Flash в год, вплоть до версии 8. Затем, в 2005–2006 годах, Macromedia заключила сделку с компанией Adobe, которая хотела бы использовать формат SWF в своих PDF файлах.

По своему устройству swf-файлы похожи на "анимированный" GIF. Вот только реализованы они не на основе растровых изображений, а с использованием векторной графики, за счёт чего являются более компактными. Кроме того, Flash обеспечивает возможности интерактивности и распространён в он-лайнных приложениях – таких, как Интернет. SWF-формат является в настоящее время единственным векторным форматом, файлы которого могут использоваться при создании Web-страниц. На Web-странице можно разместить как отдельные элементы, так и разработать практически всю страницу целиком на основе технологии Flash. Чтобы это сделать, необходимо в HTML-код страницы вписать несколько строк. При этом, если окажется, что Web-браузер пользователя не имеет средств для просмотра SWF-файлов, то он предложит загрузить из сети нужный элемент управления ActiveX. Такую загрузку нужно произвести только один раз.

Промежуточные результаты, которые доступны для редактирования во Flash, сохраняются в файлах с расширением FLA. Эти файлы еще называют исходными. Примеры мультфильмов, распространяемые в Интернете, обычно предоставляются как fla-файлы. Окончательный вариант разработки, предназначенный для просмотра, сохраняется в виде SWF-файла. Для этого в Flash имеется команда File->Publish (Файл->Публиковать). В Flash можно создать и исполняемый exe-файл, который кроме собственно графики (мультфильма) содержит в себе Flash Player.

Заголовок файла находится в самом начале файла. Он используется для определения того, является ли файл SWF файл или нет. Кроме того, он содержит информацию о размере кадра, скорость, с которой должна проигрываться анимация, и версии (которая определяет, какие теги и действия можно использовать в файле).

Приведем структуру заголовка:

```
struct swf_header {
    unsigned char f_magic[3]; //FWS' or 'CWS'
    unsigned char f_version; //версия
    unsigned long f_file_length; //размер файла
}
struct swf_header_movie {
    swf_rect f_frame_size; //размер кадра
    unsigned short fixed f_frame_rate; //разрешение кадров
    unsigned short f_frame_count; //число кадров
};
```

F_magic [3] массив символов, содержащий символы "FWS". Фильм может быть сжат, если используется версия 6. В этом случае массив будет содержать символы "CWS".

F_version имеет значение от 1 до 6 (максимум на момент написания, максимум будет продолжать расти).

F_file_length – длина файла. Это информация полезна в случае тех сетевых подключений, которые не дают сведений о размере файла. В случае сжатых фильмов, эта величина обозначает общую протяжённость несжатого фильма.

F_frame_size представляет собой определение прямоугольника, используемого для установки размера кадра на экране. Как правило, нет необходимости определять значения *minx* и *miny* в этом прямоугольнике.

Параметр *F_frame_rate* имеет фиксированный размер 8,8 бит. Он представляет собой количество кадров в секунду, которые должны воспроизводиться в фильме. Начиная с версии 8 SWF, эта величина имеет тип *unsigned short fixed* вместо *unsigned short* и не должна быть установлена в ноль.

F_frame_count – счётчик числа кадров фильма. Большая часть программных средств вычисляет это количество автоматически, и оно, как правило, может быть неправильным, однако фильм по-прежнему будет воспроизводиться без видимых искажений.

Анимация в формате SWF может быть создана двумя принципиально разными способами. Первый – это прорисовка пользователем каждого кадра вручную, с использованием Flash только в качестве средства, позволяющего быстро пролистывать изображения. Второй способ – заставить Flash автоматически просчитывать промежуточные кадры. При этом используются промежуточные отображения (*tweening animation*). В этом случае требуется только задать ключевые кадры (*keyframes*): начальный и конечный, а промежуточные Flash просчитывает автоматически. По умолчанию Flash рассчитывает промежуточные кадры по линейному закону, но можно задать возрастающую или затухающую экспоненту. Это нужно, чтобы отразить какие-нибудь процессы, происходящие в реальном мире.

Покадровая анимация – это анимация, полностью составленная из ключевых кадров. Пользователь сам определяет как содержимое кадра, так и его "длительность" (т.е. сколько таких статических кадров будет занимать изображение).

К достоинствам покадровой анимации во Flash следует отнести следующее:

- покадровая анимация даёт, в некотором смысле, больший контроль над анимацией, и опытный аниматор может выгодно ею пользоваться;
- это единственный способ организовать смену абсолютно независимых изображений – слайд шоу (например, создавая обычный баннер средствами Flash).

Покадровую анимацию сложно модифицировать. Особенно, если это не дискретный набор изображений, а связанная анимация. Приходится модифицировать все кадры и покадровая анимация занимает достаточно большой объём, так как приходится хранить информацию о каждом кадре.

При создании анимации с построением промежуточных кадров (*tweened motion*), Flash автоматически строит промежуточные кадры между ключевыми, заданными пользователем. Это означает, что пользователь рисует объект, потом на другом кадре производит изменения и просит Flash рассчитать те кадры, которые лежат между этими двумя ключевыми кадрами. Flash-редактор выполняет эту работу, и получается плавная анимация.

Скорость и плавность анимации зависят от количества кадров, которое отводится под движение, и скорости Flash фильма (*movie*). Скорость фильма можно изменять. Для качественной анимации скорость должна быть не меньше 25 – 30 кадров в секунду.

Плавность и длительность задаётся количеством кадров, отведённых на анимацию (её фрагмент). Например, если скорость фильма – 30 кадров/с, и нужно совершить перемещение объекта из одного угла картинки в другой за 2,5 с, то на это движение нужно отвести 75 кадров.

Во Flash существуют два варианта построения промежуточных изображений – *motion tweening* (построение анимации на основе модификации символов) и *shape tweening* (построение анимации на основе изменения формы). Второй способ применяется в случаях, когда нужно плавное изменение формы.

При использовании анимации на основе изменения формы (*shape tweening*) могут модифицироваться следующие параметры фигуры:

- форма;
- расположение;
- размер (любые пропорции);
- цвет;
- угол поворота.

Наиболее часто используемая техника анимации во Flash – Motion Tweening. В этом случае анимация строится на основе модификации символов, т.е. объектом анимации является символ. При использовании Motion Tweening могут модифицироваться следующие параметры объекта:

- размер (как пропорционально, так и непропорционально – отдельно высоту и ширину);
- наклон;
- расположение;
- угол поворота;
- цветовые эффекты;
- можно использовать направляющие слои для задания траектории движения объекта.

Motion tweening позволяет использовать различные цветовые эффекты применительно ко всему символу. Эта возможность отсутствует в *shape tweening*.

На сегодняшний день у формата SWF нет заслуживающего внимания конкурента на Web-страницах, способного выступить в качестве анимационного стандарта. Компания Macromedia недавно представила

Shockwave Flash на рассмотрение консорциуму W3C (World Wide Web Consortium) в качестве рекомендованного векторного формата для использования в Сети, а компания Netscape обещает встроить поддержку Flash в Navigator 5.

7.3.4. Обзор новых анимационных форматов

В сети Интернет ещё остаётся несколько незанятых рыночных ниш и для других форматов. По крайней мере компания *Parable* надеется найти такую для своего формата *Thing*.

Things – это небольшие интерактивные растровые анимационные ролики, которые можно прокручивать в любом браузере при помощи специального plug-in-модуля. Анимации представляют собой комбинацию движений по заданному контуру и кадровых изменений, приводимых в действие нажатием кнопки мыши.

Пакет, используемый для создания роликов в формате *Thing* (*thing* по-английски – "вещь"), называется *ThingMaker* ("делатель вещей") – очень простой в использовании и недорогой, а целый ряд аксессуаров к нему, таких как viewer (программа просмотра), converter (конвертор для преобразования фильмов *Director* в формат *Thing*) и *screensaver* (программа для создания экранных заставок), распространяется вообще бесплатно. Несмотря на то, что все эти дополнения завоевывают всё большую популярность (всё большее число коммерческих сайтов предлагает посетителям перекачать какой-нибудь *Thing*), исходная программа не выглядит способной на что-то большее, чем то, чего может добиться более или менее грамотный дизайнер от нескольких анимированных файлов формата GIF и небольшой порции JavaScript. Единственное реальное преимущество формата *Thing* – это простота изготовления.

Формат *Metastream* был предложен фирмой *Metacreations* и уже получил некоторое распространение в Интернете. *Metastream* демонстрирует высочайшее фотореалистичное качество изображения для визуализации текстурированных 3D-моделей. Создаваемые в *Metastream* модели полностью управляемые, с текстурой высокого разрешения, тенями, отражениями и анимацией в реальном времени. При их просмотре можно изменять размер окна, масштабировать, вращать и перемещать объекты в любых направлениях практически без потери качества визуализации.

Этот формат можно использовать для разного рода презентаций, электронных витрин в Интернет-магазинах и т.д. Причём любой демонстрируемый товар можно сильно увеличить, осмотреть со всех сторон, а также поменять цвет, обивку мебели, перекомпоновать детали и части объекта, а также составить различные комбинации товаров. Таким образом, перспективы использования формата *Metastream* для интерактивного показа товаров в Интернет-магазинах выглядят очень привлекательно, однако для использования этого формата необходим специальный модуль (*plug-in*), который можно скачать с сайта *Metacreations*.

Анимация в реальном времени, "оживление" персонажей 3D-сцен, сквозное интерактивное управление, возможность интеграции с видео, синхронизация со звуком и другие возможности постоянно совершенствуются, и в самом ближайшем будущем формат *Metastream* станет полностью универсальным и, возможно, составит серьёзную конкуренцию VRML-формату.

Пользоваться форматом *Metastream* довольно удобно, и времени на его освоение практически не затрачивается. К основным возможностям формата относятся:

- тени на объектах;
- наложение карт отражения;
- моделирование преломлений на драгоценных камнях и стекле;
- текстурные карты высокого разрешения;
- высокая детализация моделей;
- маленький размер передаваемого файла.

В дискуссиях и спорах вокруг Web-технологий чаще всего предметом обсуждения становятся стандарты анимации. Несмотря на то, что наиболее популярным промышленным стандартом здесь считается Shockwave Flash, абсолютный лидер пока всё-таки не выявился. Самым широко используемым анимационным форматом является Animated GIF.

Контрольные вопросы

1. Перечислите методы анимации.
2. Какие методы анимации позволяют создавать анимации, не прорисовывая каждый кадр?
3. Перечислите принципы анимации.
4. Каким образом создается gif-анимация?
5. Каким образом создается flash-анимация?

8. КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ

Электронная презентация – это современный эффективный способ представления информации о товарах и услугах, о разрабатываемых программных продуктах и предлагаемых технологиях, в котором удачно сочетаются возможности справочника, буклета, каталога и проспекта вместе взятых. Как правило, в электронной пре-

зентации задействованы все современные мультимедийные возможности: она включает графику и анимацию, тексты и таблицы, фотографии, видео- и аудиоматериалы. Презентации бывают различных типов в зависимости от задач, которые на них возложены.

8.1. ТИПЫ ПРЕЗЕНТАЦИЙ

1. *Официальная презентация* – различного рода отчёты, доклады и т.д. перед вышестоящим руководством. В такого рода презентациях необходим строгий дизайн, выдержанность, единый шаблон оформления для всех слайдов, ограничено применение анимационных эффектов. Задача такой презентации (и докладчика) – расположить аудиторию к себе, используя чёткое структурирование материала, минимум вводных фраз, крупный текст. Обычно слайд должен демонстрироваться на экране от 10 до 60 с.

2. *Плакаты*. Такая презентация заменяет собой простейшие средства технического сопровождения. Компьютер используется как обычный слайд-проектор, причём на слайдах представлены только иллюстрации с минимальным набором подписей и вся работа по разъяснению содержимого лежит на докладчике. В презентациях такого рода желателен единый шаблон оформления, который не рекомендуется изменять без необходимости.

3. *Презентация "Двойное действие"*. На слайдах такой презентации помимо визуальных материалов приведена конкретная информация, которая может либо пояснять содержимое слайда, либо расширять его. В результате, при правильном распределении внимания слушателей, задействуются три механизма восприятия:

- зрительно-образное восприятие, связанное с фотографией;
- слуховое сознательное, связанное с пониманием речевого сопровождения презентации;
- дополнительное зрительное сознательное, связанное с одновременным чтением дополнительного материала.

В таких презентациях допускается смена текстового ряда при неизменном визуальном материале.

4. *Презентация "Интерактивный семинар (урок)"* предусматривает проведение какого-либо вида занятия в режиме диалога с аудиторией.

В таких презентациях допустимы различные анимационные эффекты и объекты навигации, не обязательно требование единого шаблона для всех слайдов.

5. *Материал для самостоятельной проработки*. Такая презентация фактически является электронным учебным материалом, при этом она создаётся именно как электронный материал с расчётом на его чтение с экрана, и поэтому не является простым переложением печатного документа в электронный вид. В такого типа презентациях как никогда важна структурированность материала и его нацеленность на аудиторию, возможность наглядного демонстрирования учебного материала с помощью дополнительных видов воздействий (видео и анимации), а также возможность самотестирования.

6. *Информационный ролик (реклама)*. В таких презентациях весь показ проходит в автоматическом режиме и предусматривает:

- крупные тексты информационно-рекламного характера;
- наглядные материалы, рассчитанные на быстрое восприятие;
- множество анимационных эффектов;
- звуковое сопровождение.

8.2. НАВИГАЦИЯ В ПРЕЗЕНТАЦИЯХ

Презентации по типу навигации могут быть:

- неинтерактивные – пользователь не может влиять на порядок просмотра презентации;
- интерактивные – обладают системой навигации, т.е. позволяют пользователю самому выбирать интересующие его разделы и просматривать их в произвольном порядке.

8.3. ТЕХНОЛОГИЯ СОЗДАНИЯ И ПРОВЕДЕНИЯ ЭФФЕКТИВНЫХ ОФИЦИАЛЬНЫХ ПРЕЗЕНТАЦИЙ

1. **Подготовка содержания презентации.** Прежде всего, необходимо структурировать материал презентации. Для этого стержневую тему нужно разбить на несколько блоков, названия которых составляют план презентации. Каждый блок разбивают на сцены – законченное высказывание, которое может включать ключевое положение, классификацию, объяснение, пример и т.д., представленное одним слайдом.

При формировании сцен нужно помнить "золотое сечение презентации" – чем больше сцен, тем они короче. Действительно, много коротких сцен легко презентовать и о них легко говорить, но при этом может не быть целостного восприятия от презентации и тяжело рассчитать время доклада.

2. Создание слайдов.

2.1. *Выбор шаблона оформления:*

- лучше использовать фон без фоновых изображений;
- фон и текст должны быть контрастными;

- цвет фона должен быть холодным;
- при использовании градиентной заливки фона, текст располагается на светлой части, а изображения – на тёмной;
- необходимо применять в качестве шаблона простые по структуре декорации, не занимающие много места на слайде;

- нежелательно пересечение декораций с элементами слайда;

- декорации должны быть более бледными на фоне основных элементов.

2.2. Выбор шрифта:

- для презентаций используется шрифт без засечек и без теней;
- необходимо использовать один и тот же стиль шрифта во всех слайдах;
- цвет шрифта должен контрастировать с фоном и другими элементами слайда;
- для заголовка лучше применять отдельный цвет шрифта;
- желательно использовать в презентациях как более крупный шрифт.

2.3. *Изображения на слайде.* Все изображения, включаемые в презентации можно разделить на содержательные и оживляющие. Первые должны быть чёткими и ясными, профессионально выполненными и без лишних элементов, при этом докладчик должен объяснить все важные части изображения. Оживляющие изображения должны располагаться сбоку от основной информации, быть небольшими по размеру и требовать к себе не слишком много внимания.

2.4. *Таблицы, диаграммы и графики на слайде.* Таблицы, размещённые на слайдах, должны быть как можно более компактными (лучше – не более 3 строк и не более 3 столбцов) с выделенными шапкой и наиболее важными ячейками.

На одном слайде желательно размещать только одну диаграмму или график, причем желательно простых типов с небольшим количеством узлов (за исключением графиков и динамических рядов) с подписанными осями и категориями.

2.5. *Анимация в презентациях.* Если говорить об анимации в официальных презентациях, то, прежде всего, она должна быть необходимой. Помимо этого, анимация должна быть достаточно короткой и хорошо воспринимаемой. Чаще всего хорошо смотрятся анимации, показывающие движения людей и объекты, меняющие форму, размеры и взаиморасположение.

2.6. *Звуковое сопровождение официальных презентаций.* Конечно, лучшее звуковое сопровождение официальных презентаций – это голос докладчика. Если всё же звук необходим, то речевое сопровождение должно быть орфографически выверенным и как можно более кратким.

2.7. Расположение объектов на слайде:

- на одном слайде должны присутствовать не более двух основных объектов;
- весь текст, расположенный на слайде, должен прочитываться вслух (без осмысления) максимум за 20 с – это не более 40 – 50 слов;
- заголовки должны быть выделены, изображения и текст на слайде – разделены.

3. Техника проведения официальной презентации. Можно много говорить о психологических аспектах восприятия аудиторией докладчика, но нужно помнить основное.

Факторы восприятия информации:

10 % – что говорит докладчик;

20 % – содержание слайдов;

30 % – внешний вид докладчика и язык жестов;

40 % – как говорит докладчик (громкость, скорость, интонации, акценты).

Контрольные вопросы

1. Что такое презентация?
2. Какими бывают презентации?
3. Навигация какого типа используется в презентациях? Зависит ли выбор типа навигации от типа презентации?
4. Перечислите основные элементы презентации.
5. Перечислите основные правила создания презентации.

ЗАКЛЮЧЕНИЕ

Мультимедиа является бурно развивающейся областью деятельности, фактически приближающей компьютерные технологии к человеку. Несмотря на большое количество известных технологий работы со звуком, изображениями и видео ежемесячно разрабатываются новые.

В данном учебном пособии рассмотрены основные понятия мультимедиа и сферы применения мультимедиа продуктов, стандартные носители мультимедиа информации, основополагающие знания о текстовой информации в ММП, а также основные понятия теории цвета, колориметрические модели и системы и различные алгоритмы сжатия информации, применяющиеся в графических форматах.

Список литературы

1. Петров, М.Н. Компьютерная графика : учебник / М.Н. Петров, В.П. Молочков. – СПб. : Питер, 2003. – 736 с.
2. Баканов, В.М. Программирование мультимедиа-систем : учебное пособие / В.М. Баканов. – М. : МГА-ПИ, 2004. – 121 с.
3. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин – М. : ДИАЛОГ-МИФИ, 2003. – 384 с.
4. Карлащук, В.И. Презентация на компакт-диске / В.И. Карлащук. – М. : СОЛОН-Р, 2002. – 160 с.
5. Кречман, Д.Л. Мультимедиа своими руками / Д.Л. Кречман, А.И. Пушков. – СПб. : БХВ–Санкт-Петербург, 1999. – 528 с.

оглавление

5. КОМПЬЮТЕРНЫЙ ЗВУК	3
5.1. История компьютерного звука	3
5.2. Методы синтеза компьютерного звука	8
5.3. Методы сжатия аудиоданных и форматы аудиофайлов	13
5.3.1. Форматы аудиоданных без потерь	14
5.3.2. Сжатие аудиоданных с потерями	17
5.3.3. Форматы аудиоданных нотной записи	27
5.3.3. Форматы аудиоданных, использующие ноты и образцы инструментов	29
5.3.4. Форматы-контейнеры аудиоданных	29
Контрольные вопросы	32
6. ВИДЕО	33
6.1. Основные понятия	33
6.2. Кино и видеоформаты	35
6.2.1. Аналоговые форматы	35
6.2.2. Цифровые форматы	38
6.3. Сжатие видеоинформации	39
6.3.1. Требования приложений к алгоритму	40
6.3.2. Технология сжатия видеоданных	41
6.4. Стандарты сжатия видеоданных	47
6.5. Компьютерный монтаж видео	51
6.5.1. Терминология	51
6.5.2. Сценарий	52
6.5.3. Монтаж	53
Контрольные вопросы	55
7. АНИМАЦИЯ	56
7.1. Методы анимации	56
7.2. Двенадцать принципов анимации	58
7.3. Форматы анимационных файлов	60
7.3.1. Graphic interchange format (GIF)	60
7.3.2. FLC-анимация	66
7.3.3. Flash-анимация (SWF)	68
7.3.4. Обзор новых анимационных форматов	72
Контрольные вопросы	73
8. КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ	74
8.1. Типы презентаций	74
8.2. Навигация в презентациях	75
8.3. Технология создания и проведения эффективных официальных презентаций	75
Контрольные вопросы	77
ЗАКЛЮЧЕНИЕ	77
СПИСОК ЛИТЕРАТУРЫ	78