

Оглавление

Аннотация	3
Текст, гипертекст.	4
Структуры гипертекста.	5
Текст.....	6
Технические правила набора.	11
Векторная графика.	15
Типы компьютерных изображений.....	15
Элементы компьютерной графики.....	18
Растровая графика.....	28
Монохромная графика.	28
Цвет и модели цвета.	29
Форматы графических файлов.	37
Алгоритмы сжатия растровой графики.	45
Улучшение визуального качества изображений путем поэлементного преобразования.	60
Анимация.....	71
Звук.....	73
MIDI.....	74
Цифровой звук.	74
Методы шумоочистки.....	76
Кодирование звука.....	84
Основные группы звуковых плат.....	93
Технологии создания позиционируемого 3D - звука.....	95
Видео.	102
Аналоговый метод передачи видеосигналов.....	103
Цифровое видео.....	103
Сжатие видео.....	105
Обзор технологий и алгоритмов сжатия видео.	108
Алгоритм компрессии MPEG - 1.....	111
Области применения DVD.....	118
Авторские системы.	Ошибка! Закладка не определена.
Контрольные вопросы.	134
Литература.	135

МГТУ им. Н.Э. Баумана
Факультет «Информатика и Системы Управления»
Кафедра ИУ-5 «Системы обработки информации и управления»

БЕЛОНОГОВ ИГОРЬ БОРИСОВИЧ
КОРОТАЕВ АНАТОЛИЙ ИВАНОВИЧ
РЕВУНКОВ ГЕОРГИЙ ИВАНОВИЧ

Технология мультимедиа

Учебное пособие

МОСКВА

2010 год МГТУ им. Н. Э. Баумана

Аннотация.

Учебное пособие ориентировано на студентов 3 - 4 курса специальностей, связанных с компьютерными технологиями. Пособие имеет стандартную для работ по мультимедиа структуру:

- обработка текста (шрифты, набор, форматирование, понятие о вёрстке, особенности формирования текстовой информации на мониторе);
- векторная и растровая графика (математические основы, фильтры, эффекты, программы формирования и обработки);
- анимация;
- видео (формирование, сжатие, стандарты, программы обработки);
- авторские системы.

Во всех разделах большое внимание уделяется методам сжатия информации. Упор сделан на теоретические основы мультимедиа, необходимые навыки практической работы студенты получают на практических работах.

Текст, гипертекст.

Мультимедиа (многосредность) включает в себя обработку и формирование текста (гипертекста), графики, звука, видео, анимации. Для создания мультимедиа - систем разрабатываются специальные инструментальные средства - авторские системы, позволяющие организовать структурирование огромных объемов разнородной мультимедиа - информации и интерактивный доступ к ее элементам.

Тесно с понятием мультимедиа связано гипермедиа. Всякая система гипермедиа, т.е. система построенная на основании технологии гипертекста, но при этом обеспечивающая работу с нетекстовой информацией видеоизображениями, звуком, речью - одновременно является системой мультимедиа.

С другой стороны, такие классы систем мультимедиа, как, например, инфостудии (системы записи живого видео, звука в цифровом видео на магнитный носитель – видео - магнитофон или компакт диски) никак с гипертекст - технологией не пересекаются, и следовательно, не могут быть отнесены к системам гипермедиа.

Пожалуй, наиболее естественным будет обозначать "гипермедиа" как класс систем мультимедиа, для которых существенно структурирование информации с помощью гиперсвязей. Правомерно также употребление "гипермедиа" для обозначения самой теории (или технологии) гипертекста, расширенной на нетекстовые виды информации. Наконец, более или менее общепринятым стало обозначение новых продуктов информационной индустрии, электронных изданий на CD - ROM - как "изданий гипермедиа".

Мультимедиа, комбинация текста, графики и аудиоэлементов, объединенные в одном проекте, становятся интерактивным мультимедиа, когда пользователю предоставляется возможность управлять выводом информации на экран монитора. Последнее является важным, т.к. позволяет пользователю согласовывать скорость передачи информации со скоростью переработки её в своём сознании. Интерактивное мультимедиа становится гипермедиа, когда разработчик создает структуру связанных элементов, через которые пользователь может проходить и взаимодействовать с этими элементами.

Когда проект гипермедиа включает большое количество текста или символического содержания, можно выполнять индексирование содержания и связывать элементы для быстрого доступа к логически связанной информации. Если слова связаны с другими словами, то мы имеем систему гипертекста. Если текст представлен в цифровой форме, функциональные возможности компьютера позволяют сделать этот текст более доступным и наполнить его смыслом. Тогда текст называется гипертекстом; слова, разделы и предложения связаны вме-

сте, пользователь может быстро проходить через текст, выполняя различные переходы. С помощью систем гипертекста мы можем выполнять сложный поиск в тексте книги, которая находится в компьютере в цифровом виде, находит ссылки на определенные слова, после чего сразу выводит на экран изображения страниц, на которых находится это слово.

Некоторые инструментальные системы имеют встроенные средства работы с гипертекстом, например, HyperCard и ToolBook, которые позволяют нам выделять слова жирным шрифтом или цветом, затем устанавливать связь с другими словами, страницами или выполнением каких-либо операций, например, звуковым сопровождением или показом видеоклипа, которые относятся к выделенному слову.

Структуры гипертекста.

При описании систем гипертекста часто используются 2 термина: связь (link) и узел (node). Связи представляют собой логические связи между концептуальными элементами, т.е. узлами, которые содержат текст, графические изображения, звуковые элементы и другую информацию в базе данных. Связи представляют собой путеводительные переходы и меню; узлы являются доступными темами, документами, сообщениями и элементами содержания. Маркер связи (link anchor) – это место, с которого мы начали свой путь. Конец связи (link end) – это узел, связанный с маркером. Некоторые системы гипертекста предоставляют косвенный метод навигации и не имеют обратных путей; другие системы имеют возможность прямого и обратного пути.

Самый простой способ продвижения по структуре гипермедиа состоит в использовании кнопок, которые позволяют нам получать доступ к связанной информации (текст, графические изображения и звук), которая содержится в узлах. После окончания просмотра информации мы возвращаемся в начальную точку. Структура навигации усложняется, если мы добавляем ассоциативные связи, которые косвенно соединяют элементы в иерархии или последовательности. Существуют пути, на которых пользователи могут потеряться, если не установить маркеры.

Средства гипертекста. Системы обработки гипертекста имеют две общие функции, и иногда они поставляются в виде отдельных программ: создание и чтение. Построитель создает связи, определяет узлы и создает индексный указатель слов. Методология указателя слов и алгоритмы поиска, используемые для нахождения и группировки слов в соответствии с критерием поиска, разрабатываются обычно фирмами и представляют область, для которой применяются быстродействующие компьютеры.

Текст.

Слова и символы в любой форме являются общепризнанным способом общения. С помощью слов передаётся смысл идей огромному количеству людей. Поэтому слова являются важным элементом меню, навигационных систем и содержания. Звуковое сопровождение и голос комментатора помогут пользователям, но это быстро станет утомительным. Восприятие речи и звуков требует больше сил, чем чтение текста. При разработке экранов, меню и кнопок следует использовать слова, которые в наибольшей степени выражают вашу идею. Например, ВЕРНИСЬ НАЗАД более точно и сильно выражает мысль, чем ПРЕДЫДУЩИЙ. При работе необходимо экспериментировать со словами, проверять их воздействие на окружающих, может быть, следует создать специальную группу, для разработки текста.

Шрифты и гарнитуры. Шрифт - группа символов какого - либо алфавита со всеми знаками и цифрами. Гарнитура - это группа графических символов, которая включает различные размеры и стили шрифтов. Размер шрифта измеряется в пунктах: один пункт равен 0,0138 дюйма (0,3мм). Размер шрифта измеряется от верхней границы заглавных букв до нижней границы таких букв как у « у » и « д ». Helvetica, Times и Courier являются гарнитурами; Times, 12 пунктов, наклонный – это шрифт. В компьютерном мире слово “шрифт” обычно используется, когда наиболее точным названием было бы “гарнитура”.

Размер шрифта не совсем точно указывает на высоту или ширину символов. Высота символа «х» для двух разных шрифтов может иметь разные величины, в то время как высота заглавных букв в этих шрифтах может иметь одну величину. Компьютерные шрифты автоматически устанавливают расстояние между строками, которое называется интерлиньяж. Интерлиньяж можно изменять в большинстве программ на платформах Macintosh и Windows. С помощью программы – редактора можно также изменять горизонтальные параметры строк: метрики символа и кернинг пар символов. Метрики символа – это общие величины измерения для каждого символа. Кернинг – это расстояние между соседними символами, которое устанавливается, чтобы ликвидировать слишком большие промежутки между ними. Например, можно установить регулярную, сжатую или широкую ширину каждого символа. При изменении расстояния между символами и кернинга между парами символов можно получить сжатую или растянутую строку. Гарнитура имеет засечки или не имеет. Засечка представляет собой небольшой штрих на конце буквенного начертания. Например, следующие шрифты : Times, Boolman , Palatino , являются шрифтами с засечкой. Helvetica, Arial, Optima, Amant Garde – шрифты без засечки.

Классификация шрифтов.

По целевому назначению.

- Текстовые (для формирования больших объёмов текстовой информации).
- Выделенные (с целью обратить внимание читателя на какие – моменты).
- Титульные (для формирования заголовков).
- Акцидентные (для выполнения случайных работ – визитки, реклама и т.д.).
- Афишно – плакатные (высокие и узкие).

Классификация по ГОСТ по группе – совокупность шрифтов, отличающихся контрастностью и формой засечек. Контрастность – это разница в толщине основных и вспомогательных штрихов, составляющих символ. Засечки (необязательные элементы, служат для украшения) бывают круглыми, треугольными или квадратными.

- Рубленые - без засечек (Букварная, Древняя, Плакатная).
- Шрифты с едва наметившимися засечками (Октябрьская).
- Медиевальные шрифты, применяются при наборе основного текста. Засечки плавно сопрягаются с основными штрихами и, как правило, строятся как дуги окружностей (Литературная, Банниковская, Таймс).
- Группа обыкновенных шрифтов. Имеют ярко выраженный контраст и длинные тонкие засечки (Обыкновенная, Бодони, Елизаветинская).
- Брусковые шрифты. Контраст отсутствует или незаметен, засечки прямые и утолщённые (Балтика, Брусковая).
- Малоконтрастные. Имеют длинные закруглённые засечки, изысканные формы (Школьная, Журнальная, Академическая).
- Дополнительная. Чаще всего – это различные рукописные гарнитуры (Жихаревская).

Классификация Microsoft Windows.

- Roman. Определяет шрифты с засечками, например, Times New Roman, Bodoni.
- Swiss. Определяет рубленые шрифты, имеющие переменную толщину штрихов, например, Futura.
- Modern. Эти шрифты имеют постоянную толщину штрихов. Сюда же входят все моноширинные шрифты (Courier).
- Script. Определяет рукописные шрифты (Script).
- Decorative. Декоративные шрифты (Gotic).
- Dont know. Общий тип шрифтов.

По положению очка (в полиграфии очко - площадка с символом): прямые, курсивные, наклонные. По ширине очка: узкий, нормальный, широкий. По насыщенности: полужирный, жирный, светлый.

Системы шрифтов. На данный момент Windows поддерживает два формата шрифтов – немасштабируемый растровый (FON), и TrueType (TTF), повсеместно используемый формат, который постепенно заменит растровый.

Windows по умолчанию размещает шрифты в папке \Windows\Fonts, что значительно упрощает работу со шрифтами, их установку и удаление. Просмотреть данную папку можно как с помощью Explorer, так и выбрав соответствующий элемент в панели управления.

Adobe Type 1 Font Format

Этот формат был создан фирмой Adobe Systems Inc. около 1985 года, а в 1990--публично раскрыт и документирован. Этот формат полностью совместим с языком PostScript, выпущенным в том же 1985 году, и поддерживается всеми PostScript устройствами.

PostScript - это язык для описания страницы документа, он используется для вывода страницы с текстом и графикой на экран и на принтер (который поддерживает этот язык). Основой языка является возможность свободно манипулировать текстом и графикой, применяя к ним одни и те же преобразования. Основные особенности языка:

1. Работа с объектами (строки, числа, процедуры, словари, массивы). Объектом является любое слово программы, отделённое разделителем.
2. Все объекты (кроме команд и имён функции) заносятся в один из стеков языка.
3. Использование обратной (или польской) записи операндов.

При работе с графическими объектами обеспечивается полная независимость от графических устройств. Одним из основных понятий языка является графическое состояние – совокупность всех параметров состояния графической системы. Изменение графического состояния производится специальными командами. Все рисующие команды получают информацию из буфера графического состояния.

Шрифты Adobe Type основаны на данном языке. Язык содержит команды управления, описания переменных, массивов, набор геометрических фигур. Среди них могут быть любые виды кривых, базовые геометрические фигуры, которые могут быть использованы для формирования символов. Допускается многократное наложение фигур. Существуют две основные разновидности - Type 1 PostScript font и Type 3 PostScript font. Стандартом является Type 1. Type 3 font (также известный как "шрифт, определенный пользователем") используется довольно редко. Шрифты Type 1 применяются в различных графических и издательских программах, например Adobe PageMaker, но они не являются стандартными в Microsoft Windows и поэтому их использование ограничивается программами, специально разработанными для поддержки данных шрифтов.

В основе этого формата лежит т.н. "декларативный" метод разметки. Это означает, что общие особенности построения символа описываются отдельно от самого контура символа.

Type 1 эффектно использует эту особенность: при разработке шрифта дизайнеру даны такие мощные инструменты разметки, как хинты, "голубые зоны" и разметка слабо изогнутых кривых. Эти инструменты позволяют контролировать процесс растеризации символов и тем самым гарантируют ее (растеризации) высокое качество при любой разрешающей способности выводного устройства.

Главная проблема при создании цифрового векторного шрифта состоит в том, что на выводном устройстве (будь то фотонаборный аппарат, принтер или монитор) этот векторный шрифт будет растеризован, т.е. преобразован в набор точек. Все просто, когда контур достаточно большой. Но при выводе на устройства с небольшим разрешением или при мелком кегле возникает ситуация, когда в результате "механического" округления координат точек толщины одинаковых штрихов в символе принимают разные значения, высота символов сильно колеблется, нарушается симметричность и т.д.

Голубые зоны или зоны выравнивания осуществляют глобальную разметку шрифта. Они фиксируют положение и ширину всех областей, в которых имеются т.н. оптические наплывы (напр., эффект уменьшения высоты округлых символов – надо увеличить на 2-3%). Для упрощения локальной разметки определяется наиболее распространённые толщины вертикальных и горизонтальных штрихов.

Локальная разметка. Хинты представляют собой пары направляющих, дополнительно определяющие положения и толщины основных штрихов/элементов символа. Они реализуются в виде набора пар вертикальных и горизонтальных линий, которые фиксируют все штрихи символов и штрих - подобные элементы.



Рис. 1. Разметка Type 1.

При растеризации первым делом происходит расчет толщины и положения хинтов, а потом, уже на эти рассчитанные величины накладывается контур. Размер и положение данного хинта остаются неизменными от символа к символу. Даже в рамках такого низкого разрешения нам удастся сохранить постоянную толщину основных штрихов и соблюсти одинаковый рост знаков.

Разметка слабо изогнутых кривых. Контуров некоторых символов имеют слабо изогнутые кривые, при помощи которых подчёркивается форма вертикальных или горизонтальных эле

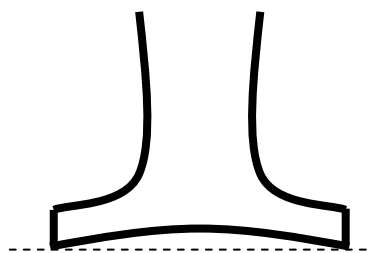


Рис. 2. Разметка слабо изогнутых кривых.

ментов. Применяется метод описания, при котором определяется размер символа, при котором пара кривых заменяется на вектор.

TrueType. Этот формат был разработан фирмами Microsoft и Apple и, прежде всего, отличается от Type 1 своим "программируемым" методом разметки.

Такой метод в теории делает возможной более точную и правильную разметку. Но на практике он затрудняет процесс разработки настолько, что непосредственно в TrueType шрифты вообще не разрабатываются, а те, что стоят у вас на компьютере, наверняка получены как преобразование шрифтов из других форматов. В результате, шрифты TrueType обычно сильно уступают по качеству аналогичным шрифтам в PostScript формате.

Конечно, нельзя не отметить, что с точки зрения пользователя использование TrueType несколько проще (например, не требует установки специальных программ), а сами шрифты более распространены и стоят дешевле Type 1.



Рис. 3. Формирование шрифтов TrueType.

Шрифты и фонты. С развитием графических операционных систем шрифты, или наборы графических изображений, которые служат для представления букв на мониторе или бумаге, превратились в их неотъемлемую и очень важную часть. Шрифты нужны для оформления текстов, Web - страниц и во множестве других случаев.

Для разработки шрифтов созданы специальные программы. В качестве примера рассмотрим графический редактор Fontographer фирмы Macromedia, который является специализированным редактором для платформ Macintosh и Windows. Этот редактор можно использовать для разработки шрифтов PostScript для компьютеров Macintosh, IBM – совместимых компьютеров PC и рабочих станций NeXT, и также для разработки шрифтов TrueType для

Macintosh и PC. Разработчики могут также изменять существующие гарнитуры, включать шрифты PostScript, автоматически трассировать сканированные изображения и создавать свои шрифты. Fontographer позволяет создавать рисованные изображения каллиграфических и скрипт - символов, используя мышь или другие способы ввода, например, сенсорные системы фирмы Wacom, цифровые планшеты фирмы Kurta и графический планшет CalComp DrawingBoard II. На основе двух гарнитур можно создавать различные виды шрифтов, изменять насыщенность всей гарнитуры шрифта. Fontographer для Windows открывает любой шрифт PostScript Type1 или True Type для компьютера PC и позволяет нам создавать уплотненный, расширенный и наклонный шрифты или модифицировать любой из этих шрифтов. Можно масштабировать, вращать или изгибать один или несколько символов или весь шрифт для создания уникальных гарнитур. Также позволяет устанавливать ширину символов, расстояние между символами, кернинг и офсет.

Однако, Fontographer не позволяет устанавливать ряд параметров для символьного набора (с засечкой или без засечки, сжатый или расширенный), т.е. имеет некоторые ограничения в написании надписи из определенного материала.

Технические правила набора.

Набор текста в редакторах типа Word осуществляется сплошной строкой. Слова разделяются пробелом (и только одним), клавиша ENTER нажимается только в конце абзаца. Использование пробела, табуляции и ENTER для форматирования недопустимо. Для форматирования необходимо использовать команды Формат – Абзац, Формат – Шрифт и т.д.

При оформлении текста для компьютера необходимо учитывать специфику чтения с монитора.

Для небольших размеров нужно применять легко читаемые шрифты. Декоративные шрифты трудно читаемые.

В одной работе рекомендуется использовать как можно меньше различных гарнитур. Однако можно изменять стиль шрифта.

Необходимо обращать внимание на расстояние между строками (интерлиньяж). Плотные расположенные строки трудно читаются.

Рекомендуется изменять кегль шрифта в зависимости от важности сообщения.

В крупных заголовках надо изменять расстояние между символами (кернинг) с целью избежать слишком разряженного заголовка.

Для выделения заголовка можно использовать цвет для символов и фона (например, обратный вид шрифта, белый на черном фоне).

Для того, чтобы заголовок выглядел более “мягко” можно использовать символы с плавным переходом границ.

Не рекомендуется, чтобы выровненный по центру заголовок содержал более 4 строк.

Рекомендуется использовать тени символов.

Вокруг заголовка необходимо оставлять достаточно свободного места.

Надо выбирать гарнитуры, которые соответствуют смыслу фразы.

Требования к простому набору могут быть сформулированы следующим образом.

Строка набора, требования к выключке. Оптимальным межсловным пробелом является полукегельная. Изменение ширины пробелов для выключки разрешается в пределах от $\frac{1}{4}$ - $\frac{3}{4}$ - кегельной. В одной строке пробелы должны быть одинаковыми. В двух смежных строках ширина пробелов не должна быть резко различной. Не следует допускать совпадения пробелов по вертикали или диагонали в трех или более смежных строках (коридоры). В строках, не заполненных текстом (концевые, выключенные в красную и т. д.) пробелы должны быть равны полукегельной. Строки без межсловных пробелов и строки малого формата могут быть выключены разрядкой с увеличением межбуквенных пробелов до 1 п. или уменьшением их до половины п. При малых форматах пробелы могут увеличиваться до кегельной. В неответственных изданиях возможны пробелы до $1\frac{1}{4}$ кегельной, коридоры и разные пробелы в смежных строках

Все абзацные отступы в тексте издания должны быть одинаковыми независимо от кегля. При обратном абзацном отступе одинаковыми должны быть втяжки. Для текстов разных кеглей допустима разница 1 п.

В начале прямой речи тире отбивают полукегельной. Кавычки и скобки от заключенных в них слов не отбивают, от смежных слов отбивают межсловным пробелом.

Знаки № и § отбивают от относящихся к ним чисел на полукегельную, знаки % и ‰ от относящихся к ним чисел не отбивают. Указанные знаки могут применяться только с числами.

Знаки *, \ " от предшествующих чисел не отбивают, от последующих чисел или обозначений шкалы отбивают на 2 п.

Многочисленные числа, набранные арабскими цифрами, разбивают на классы щпациями 2 п. Разбивку не делают для четырехзначных чисел, десятичных дробей и обозначений номеров и стандартов. От относящихся к ним знаков и наименований числа отбивают полукегельной. Нарращения к числительным присоединяют дефисом без отбивок. Простые дроби от целой части числа не отбивают. В обозначениях числа арабские и римские и буквы с точками и без точек между собой не разбивают. В неответственных изданиях числа от наименований допускается отбивать межсловным пробелом.

В случае наличия больших объёмов текста (обучающие системы) необходимо выполнять требования полиграфии к оформлению текстового материала, т.е. осуществлять вёрстку. Вёрстка – это макетирование публикации для тиражирования, компоновка страниц из подготовленных заранее текстов и иллюстраций. Современным инструментом для вёрстки являются настольные издательские системы. Вёрстка может быть проведена в программах Ventura, QuarkXpress, PageMaker и т.д. Традиции полиграфического оформления нашли свое отражение в технологических инструкциях по выполнению отдельных процессов.

Вёрстка веб - страниц.

По принципам использования средств разметки HTML различают логическую разметку и физическую. Например, *курсивный текст* можно получить как с помощью тэга <I>, так и с помощью тэга . В первом случае курсив задаётся явным образом, а во втором на текст производится логическое ударение, которое обычно отображается курсивом. Иными словами при первом подходе ориентируются на внешний вид, а во втором - на логическое предназначение. Преимуществом второго подхода является независимость вёрстки от используемого типа устройств и дизайна веб - страниц. Если придерживаться логической разметки, то можно использовать один и тот же вариант вёрстки для экрана и печати.

До недавнего времени в качестве основных инструментов верстки выступали фреймы и таблицы. Фреймы, ввиду их некоторого несоответствия принципам веб - дизайна и навигации ушли в прошлое. В настоящий момент не приветствуется любое использование фреймов. Таблицы применяются часто: с их помощью создают рамки, модульные сетки, цветной фон и т. д.

С выходом новых версий браузеров стали использовать средства по работе со слоями.

Свойства слоя удобно задавать и настраивать через стили. Слой можно перемещать, прятать и показывать без перезагрузки всей страницы. С помощью всего нескольких инструкций можно создавать разные эффекты, вроде выпадающих меню, всплывающих подсказок, движущихся элементов и другое. Добавление подобных трюков хотя и увеличивает объем кода, но не требует повторной загрузки и обновления документа и происходит без лишних задержек со стороны браузера. Кроме того, выразительность и привлекательность сайта во многом повышается благодаря использованию подобных приемов со слоями. Слои можно накладывать друг на друга, что упрощает размещение элементов на веб - странице и предоставляет больше возможностей при верстке. Слои можно размещать в окне браузера с точностью до пикселя. Слои по сравнению с таблицами отображаются быстрее. Более высокая скорость достигается за счет компактного кода и того, что отображение содержимого слоя происходит по мере его загрузки. Правда это может привести к «скачкам» элементов страни-

цы по мере их подгрузки. Управлять любыми объектами веб - страницы с помощью стилей проще, легче и эффективней.

Векторная графика.

Типы компьютерных изображений.

Имеются два основных класса компьютерных изображений: векторные и растровые.

Векторные изображения, известные также как объектно - ориентированные изображения, определяются узловым множеством точек, связанных математическими соотношениями.

Растровые изображения задаются матрицей, описывающей точки экрана и их цвет. Чем выше качество растрового изображения (больше пикселей и больше глубина цвета), тем больше потребуется памяти для хранения информации для каждого пикселя. Суть принципа точечной графики: если надо закодировать какой-то объект, то на него "накладываем" сетку и создаем матрицу (таблицу) той же размерности, заполняя единицами ячейки, наложенные на объект, и нулями вне объекта. Если границы оригинал - объекта параллельны границам ячеек сетки, получается идеальная матрица (bitmap) из нулевых и единичных битов, которая представляет закодированное изображение объекта. Если эту матрицу вывести на экран или принтер или на диск для хранения, то получим оттиск объекта.

Растровые изображения ближе к фотографии, поскольку позволяют более точно воспроизводить основные характеристики фотографии: освещенность, прозрачность, плавность переходов, тональность и пр. В отличие от векторных изображений растровые изображения плохо масштабируются, нужные масштабы и разрешение задаются при создании изображения. Один из способов получения растровых изображений - сканирование фотографий или слайдов. Существуют также разнообразные пакеты компьютерной графики, создающие растровые изображения. Более профессиональные пакеты предлагают пользователям разнообразные графические эффекты, которые позволяют в некоторых случаях достигать изображений, сравнимых с "ручным" рисованием или фотографией.

Если посмотреть на фрагмент растрового изображения в увеличенном виде, то можно заметить, что плавная линия представляется в виде ступени, но при высокой разрешающей способности и большом числе цветов эти ступеньки не заметны для глаза.

Векторное изображение может отличаться очень высокой точностью передачи линий и сложных геометрических форм. Векторная графика не дает особых преимуществ в передаче оттенков или текстуры по сравнению с растровыми изображениями, но всё же имеет ряд преимуществ:

- несложные векторные изображения обладают меньшими размерами по сравнению с растровыми изображениями;
- векторные изображения являются масштабируемыми, что означает возможность их увеличения или уменьшения без каких - либо искажений;

- в векторных изображениях часто отдельно представлены координаты векторов (точек) и характеристики их визуализации, что позволяет легко получать разнообразные визуальные образы для одной математической (геометрической) формы.



Рис. 4. Пример растрового и векторного изображения.

Векторные изображения не могут быть получены путем сканирования или перехвата экранных изображений. Они создаются с помощью специальных пакетов векторной графики или путем специальных математических преобразований растровых изображений. В векторной графике невозможно использование библиотеки эффектов (фильтров), используемых в растровой графике. Векторная графика ограничена в живописных средствах и не позволяет получить фотореалистичные изображения с тем же успехом что и растровая. Причиной этого является тот факт, что векторная графика оперирует с более громоздкими объектами, чем пиксель. Структуру любой векторной иллюстрации можно представить себе в виде иерархи-

ческого дерева. В таком дереве сама иллюстрация занимает верхний уровень, а её составные части – более низкие уровни.

Основным элементом векторной графики является линия. Узлы и отрезки находятся на самом нижнем уровне. Существует несколько типов линий и узлов. Узел задаётся парой чисел (x,y).

Сегментом называется отрезок, соединяющий два узла.

Заливка – это узор или цвет, выводимый в замкнутой области, ограниченной кривой.

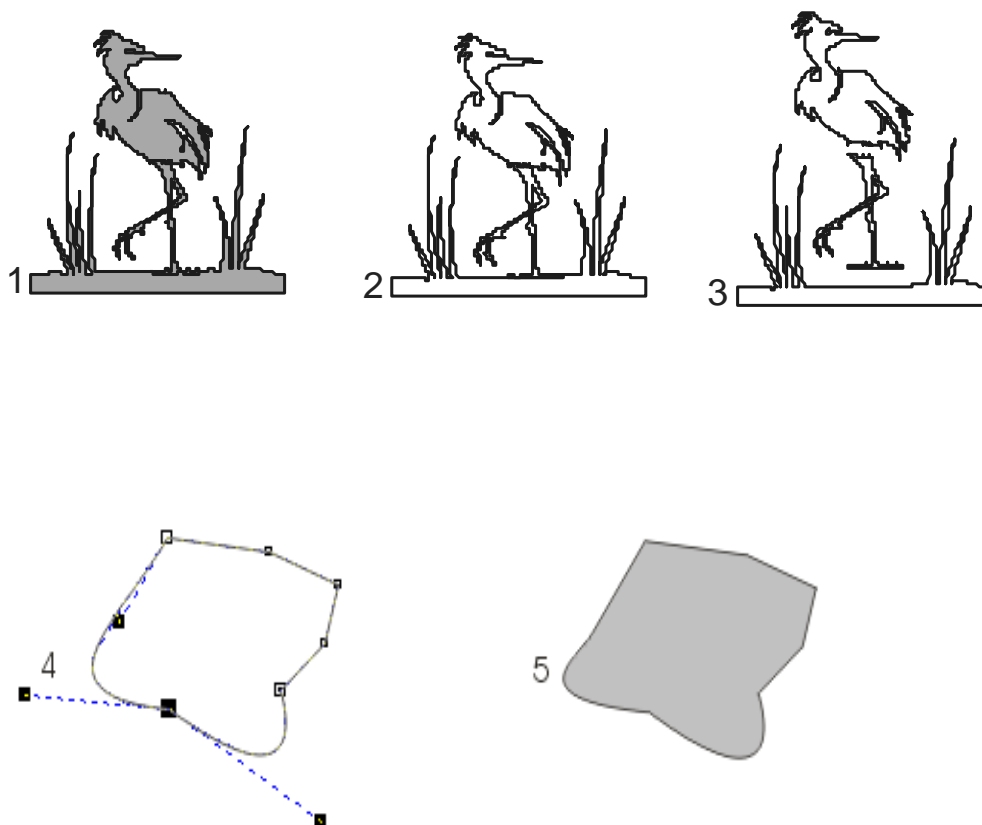


Рис. 5. 1 – исходное изображение, 2 – рисунок (как совокупность векторов), 3 – разгруппированный рисунок в виде набора контуров, 4 - представление рисунка в виде сегментов, состоящих из узлов и отрезков, модификация рисунка путём изменения положения узлов и управляющих точек, 5 – однотонная заливка.

Традиционно векторная графика широко использовалась в геоинформационных системах (ГИС) и картографии. Векторные форматы более компактны и с ними легче было работать. Современные ГИСы чаще используют теперь векторно - растровые технологии. Некоторые задачи, например, интерполяцию точечных изображений на поверхность, проще решать на растровых изображениях. Кроме того, здесь часто приходится иметь дело с фотографиями поверхности Земли, полученными со спутников. Такие снимки преобразуются в проекции карты, где начинают применяться методы математического моделирования.

Элементы компьютерной графики.

Заполнение многоугольника. Способы:

Определением принадлежности пикселя экрана многоугольнику.

Штриховкой.

Заливка области с затравкой.

Определение принадлежности пикселя многоугольнику. Простейший способ заполнения многоугольника – определять, принадлежит ли пиксель многоугольнику или нет.

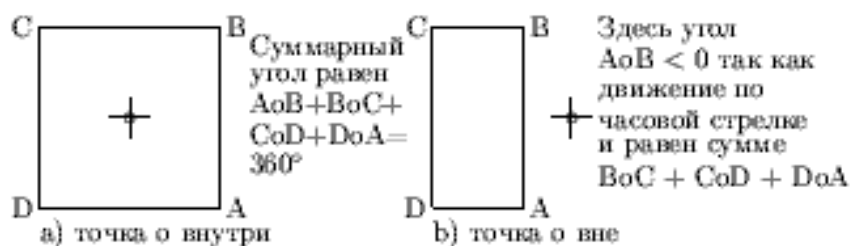


Рис. 6. Определением принадлежности пикселя экрана многоугольнику

Если пиксель внутри – суммарный угол, составленный отрезками, соединяющих пиксель и вершины = 360° . Пиксель снаружи – суммарный угол = 0°

Штриховка. Просматриваются пиксели, лежащие на одной строке и закрашиваются те, которые принадлежат многоугольнику.

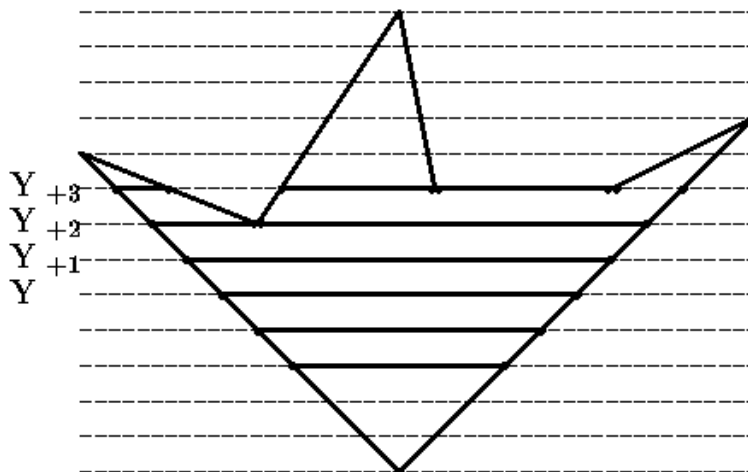


Рис. 7. Заполнение штриховкой.

Для каждой строки сканирования вычисляются X - координаты пересечений со всеми ребрами, X - координаты пересечений сортируются, закрашка ведется между парами отсортированных координат.

Заливка области с затравкой. Задаются: заливаемая (перекрашиваемая) область, код пикселя, которым будет выполняться заливка, начальная точка в области, начиная с которой

начнется заливка. Закрашиваются те пиксели, которые являются соседними с затравочным. Потом происходит переопределение затравочного пикселя.

Расширением двумерной графики является трёхмерная векторная графика, нашедшая своё широкое применение в программах анимации.

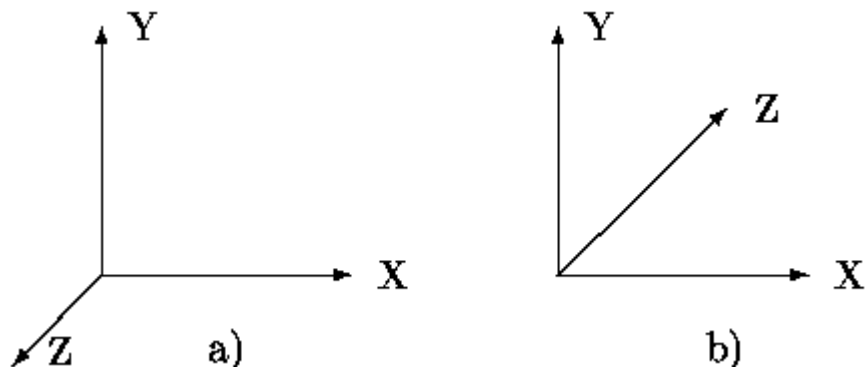


Рис. 8. Правая и левая системы координат.

Система координат называется правой, если для совмещения с положительной полуосью Y положительную полуось X требуется повернуть на $+90^\circ$ при этом направление движения расположенного вдоль оси Z и поворачивающегося против часовой стрелки правого винта и положительной полуоси Z совпадают.

В однородных координатах точка представляется четырехмерным вектором $[x \ y \ z \ w]$, где $w=1$, а матрицы преобразований имеют размер 4×4 . Точка в пространстве определяется вектором a с учетом однородных координат - (X, Y, Z, H) или $(X^*, Y^*, Z^*, 1)$.

Преобразование в однородных координатах:

$$\begin{bmatrix} x_n & y_n & z_n & w_n \end{bmatrix} = \begin{bmatrix} x & y & z & w \end{bmatrix} \cdot T.$$

Обобщенная матрица преобразования для трехмерного случая имеет вид:

$$T = \begin{pmatrix} A & B & C & P \\ D & E & F & G \\ H & I & J & R \\ L & M & N & S \end{pmatrix}$$

Эта матрица состоит из четырех частей:

Матрица 3×3 меняет масштаб, осуществляет сдвиг и вращение изображения, 1×3 – перенос, 3×1 – преобразование в перспективе, 1×1 – общее изменение масштаба.

$$\begin{pmatrix} 3*3 & 3*1 \\ 1*3 & 1*1 \end{pmatrix}$$

Изменение масштаба:

$$(X, Y, Z, 1)^* = (AX, EY, JZ, 1).$$

$$T = \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & J & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Сдвиг:

$$(X, Y, Z, 1)^* = (X+YD+HZ, BX+Y+IZ, CX+FY+Z, 1).$$

$$T = \begin{pmatrix} 1 & B & C & 0 \\ D & 1 & F & 0 \\ H & I & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Вращение:

$$T_x = \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T_y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos(\gamma) & \sin(\gamma) & 0 \\ 0 & -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_z = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

где T – вращение относительно указанной оси.

Пространственный перенос:

$$(X, Y, Z, 1)^* = (X+L, Y+M, Z+N, 1).$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L & M & N & 1 \end{pmatrix}$$

Проецирование:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Комбинированные вращения, которые следуют за проецированием из центра, лежащего в бесконечности, являются основой для получения аксонометрических проекций всех типов. Для наиболее распространенных типов аксонометрических проекций - изометрии и диметрии - углы поворота имеют следующие значения: $\alpha_z = -450$, $\beta_x = 350$; $\gamma_y = -200$; $\delta_x = 200$.

Для построения косоугольных проекций пользуются различными вариантами сдвига.

Для получения псевдореального изображения в центральной проекции оси координат следует расположить следующим образом. Начало координат расположить в левом нижнем углу. Ось X направить направо, ось Y наверх, ось Z вглубь экрана, от наблюдателя.

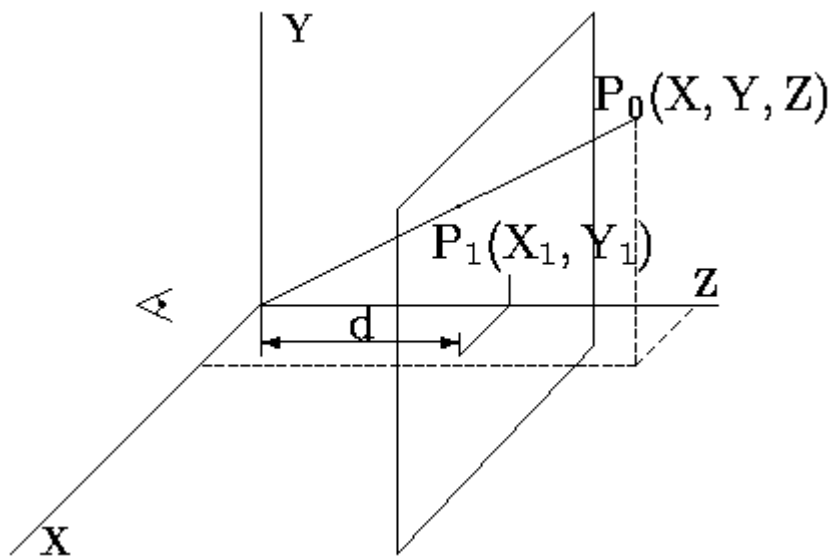


Рис. 9 . Система координат для получения псевдореального изображения в центральной проекции.

Расстояние до экрана обозначить через $z_0 < 0$, (x, y, z) – координаты объекта в пространстве, (X, Y) – на экране. Тогда матрица преобразования имеет вид:

$$(x, y, z, 1)^* = (X^*, Y^*, 0, H)$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Т.к. величина H должна быть равна 1, то последний вектор надо разделить на H.

Реалистичное представление сцен.

Включает:

- Модели освещения.
- Механизмы отражения света.
- Модели закраски.

- Прозрачность.
- Тени.
- Фактура.
- Трассировка лучей.
- Излучательность.

Источники света подразделяются на излучающие и отражающие, источники точечные и распределенные, источники рассеянного света.

Поверхности: отражающие, поглощающие, полупрозрачные, рассеивающие.

Диффузное отражение.

Закон Ламберта - падающий свет рассеивается во все стороны с одинаковой интенсивностью. Освещенность точки пропорциональна доле ее площади, видимой от источника.



Рис. 10. Отражение света. $I_r = I_p \cdot P_d \cdot \cos(Q)$,

I_r - интенсивность отраженного света,

I_p - интенсивность точечного источника,

P_d - коэффициент диффузного отражения, зависящий от материала поверхности и длины волны,

Q - угол между направлением света и нормалью к поверхности.

Учет рассеянного света

$$I = I_r \cdot P_r + I_p \cdot P_d \cdot \cos(Q),$$

I_r - интенсивность рассеянного света,

P_r - коэффициент отражения рассеянного света.

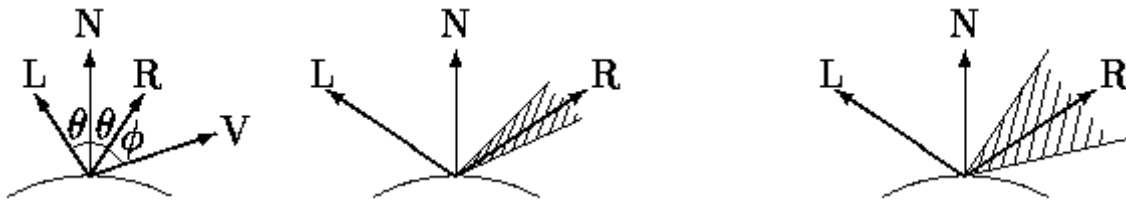
Учет расстояния.

$$I = I_r P_r + \frac{I_p P_d}{d + K}$$

d - расстояние от центра проекции до объекта, при параллельной проекции d - расстояние от объекта, ближайшего к наблюдателю,

K - произвольная константа.

Зеркальное отражение.



Зеркальное отражение б) Отражение от блестящей поверхности в) Отражение от тусклой поверхности

Рис. 11. Варианты моделирования отражения.

$[L\vec{v}]$ - ед. вектор направления на источник света. $[N\vec{v}]$ - нормаль к поверхности. $[R\vec{v}]$ - ед. вектор направления идеального отражения. $[V\vec{v}]$ - ед. вектор направления к наблюдателю.

Эмпирическая модель Фонга:

$$I_s = I_p \cdot W(\phi, \theta) \cdot \cos^n(\theta),$$

$W(\phi, \theta)$ - кривая отражения,

$$-\pi/2 \leq \phi \leq \pi/2,$$

$$1 \leq n \leq 200,$$

Для идеального отражателя $n = \infty$.

Для тусклых, негладких поверхностей типа мела или сажи $n \approx 1$.

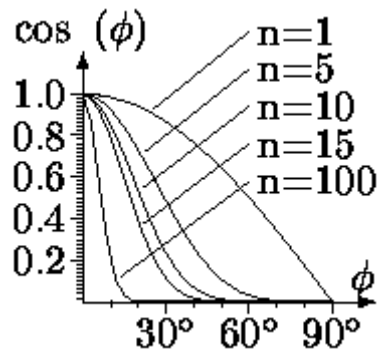


Рис. 12. Зависимость $\cos^n(\theta)$ от значения параметра отражения n .

Часто $W(l, q)$ заменяется константой K_s , такой, чтобы полученная картина была субъективно приемлема.

Суммарная модель освещения:

$$I = I_r P_r + \frac{I_r}{d + K} (P_a \cos(q) + W(l, q) \cos n(\phi))$$

Или при замене $W(l, q)$ на константу K_s :

$$I = I_r P_r + \frac{I_r}{d + K} (P_d \cos(q) + K_s \cos n(\varphi))$$

Для поверхностей, представленных, например, в виде бикубических кусков, каждое произведение меняется в пределах куска.

Фонг предложил алгоритм пошагового вычисления по рассмотренной модели, существенно снижающий затраты.

Модели с микрогранями.

Отражающая поверхность представлена в виде плоских микрограней.

Ориентации нормалей к граням относительно нормали к средней линии поверхности задаются некоторым распределением, например, Гаусса.

Модели закраски.

Однотонная (и источник и наблюдатель в бесконечности).

Метод Гуро.

Метод Фонга.

Прозрачность.

Без учета преломления.

С учетом преломления.

Суммарная закраска:

$$I = k \cdot I_{\bar{o}} + (1-k) \cdot I_o,$$

k - характеризует прозрачность ближнего многоугольника. Если $k = 1$, то он непрозрачен. Если $k = 0$, то ближний многоугольник полностью прозрачен. $I_{\bar{o}}$ - интенсивность для пикселя ближнего многоугольника. I_o - дальнего.

Тени. Объект невидимый из источника света находится в тени.

Шаги алгоритма:

Определяются все многоугольники, видимые из точки освещения.

Удаление поверхностей невидимых из точки зрения.

Закраска многоугольников. Если видим из источника освещения, то учитываются диффузное и зеркальное отражения и рассеянный свет. Если невидим, то многоугольник в тени и надо учитывать рассеянное освещение.

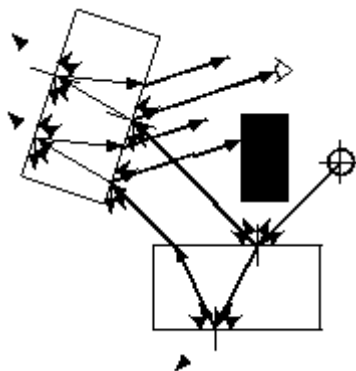


Рис. 13. Трассировка лучей.

Трассировка лучей: прямая трассировка лучей, обратная трассировка лучей.

Прямая трассировка лучей.

Расчет освещения сцены:

- от всех источников света испускаются лучи во всех направлениях;
- рассчитывается преломление и отражение каждого луча, в том числе и отраженного, т.е. каждая точка сцены может освещаться либо напрямую источником, либо отраженным светом;
- часть лучей, попавшая в глаз наблюдателя, сформирует в нем изображение сцены.

Изображение формирует только малая часть лучей.

Обратная трассировка лучей.

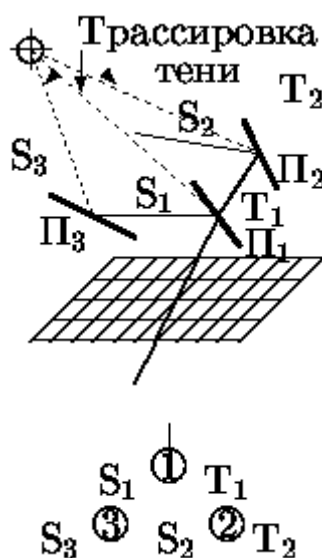


Рис. 14 . Обратная трассировка лучей.

Расчет освещения сцены:

- отслеживаются лучи, проходящие из глаза наблюдателя через каждый пиксель экрана в сцену;

- на каждой поверхности сцены, на которую попадает луч, формируются отраженный и преломленный лучи;
- каждый такой луч рекурсивно отслеживается, чтобы определить пересекаемые поверхности;

Удаление скрытых линий и поверхностей.

Классификация методов удаления невидимых частей

Методы удаления невидимых частей сцены можно классифицировать:

- По выбору удаляемых частей: удаление невидимых линий, ребер, поверхностей, объемов.
- По порядку обработки элементов сцены: удаление в произвольном порядке и в порядке, определяемом процессом визуализации.

По системе координат:

- алгоритмы работающие в пространстве объектов, когда каждая из N граней объекта сравнивается с остальными $N-1$ гранями (объем вычислений растет как N^2),
- алгоритмы работающие в пространстве изображения, когда для каждого пикселя изображения определяется какая из N граней объекта видна (при разрешении экрана $M \times M$ объем вычислений растет как $M^2 \times N$).

Алгоритмы удаления линий.

Применение - представление каркасных моделей. При этом не используется основное ценное качество растрового дисплея - возможность закраски поверхностей. В этой связи основная область применения - векторные устройства, но могут применяться и в растровых для ускорения процесса визуализации.

Наиболее известный ранний алгоритм - алгоритм Робертса (1963 г.). Работает с только выпуклыми телами в пространстве объектов. Каждый объект сцены представляется многогранным телом, полученным в результате пересечения плоскостей. Т.е. тело описывается списком граней, состоящих из ребер, которые в свою очередь образованы вершинами.

Вначале из описания каждого тела удаляются нелицевые плоскости, экранированные самим телом. Затем каждое из ребер сравнивается с каждым телом для определения видимости или невидимости. Т.е. объем вычислений растет как квадрат числа объектов в сцене. Наконец вычисляются новые ребра, полученные при протыкании телами друг друга.

Алгоритм удаления поверхностей с Z - буфером

Обычный буфер кадра хранит коды цвета для каждого пикселя в пространстве изображения. Идея алгоритма состоит в том, чтобы для каждого пикселя дополнительно хранить еще и координату Z или глубину. При занесении очередного пикселя в буфер кадра значение его Z - координаты сравнивается с Z - координатой пикселя, который уже находится в буфере. Если Z - координата нового пикселя больше, чем координата старого, т.е. он ближе к наблю-

дателю, то атрибуты нового пикселя и его Z - координата заносятся в буфер, если нет, то ничего не делается.

Этот алгоритм наиболее простой из всех алгоритмов удаления невидимых поверхностей, но требует большого объема памяти. Время работы алгоритма не зависит от сложности сцены. Многоугольники, составляющие сцену, могут обрабатываться в произвольном порядке. Для сокращения затрат времени нелицевые многоугольники могут быть удалены. Основным недостатком алгоритма с Z - буфером - дополнительные затраты памяти. Для их уменьшения можно разбивать изображение на несколько прямоугольников или полос. В пределе можно использовать Z - буфер в виде одной строки. Понятно, что это приведет к увеличению времени, так как каждый прямоугольник будет обрабатываться столько раз, на сколько областей разбито пространство изображения. Уменьшение затрат времени в этом случае может быть обеспечено предварительной сортировкой многоугольников на плоскости.

Другие недостатки алгоритма с Z - буфером заключаются в том, что пиксели в буфер заносятся в произвольном порядке и возникают трудности с реализацией эффектов прозрачности или просвечивания и устранением лестничного эффекта с использованием предфильтрации. Здесь каждый пиксель экрана трактуется как точка конечного размера и его атрибуты устанавливаются в зависимости от того какая часть пикселя изображения попадает в пиксель экрана. Но другой подход к устранению лестничного эффекта, основанный на постфильтрации - усреднении значений пикселя с использованием изображения с большим разрешением реализуется сравнительно просто за счет увеличения расхода памяти (и времени). В этом случае используются два метода. Первый состоит в том, что увеличивается разрешение только кадрового буфера, хранящего атрибуты пикселей, а разрешение Z - буфера делается совпадающим с размерами пространства изображения. Глубина изображения вычисляется только для центра группы усредняемых пикселей. Это метод неприменим, когда расстояние до наблюдателя имитируется изменением интенсивности пикселей. Во втором методе и кадровый и Z буфера имеют увеличенное разрешение и усредняются атрибуты пикселя, так и его глубина.

Растровая графика.

Основные понятия.

Статические растровые изображения, представляющие собой двумерный массив чисел - пикселей. Все изображения можно подразделить на две группы: с палитрой и без нее. У изображений с палитрой в пикселе хранится число - индекс в некотором одномерном векторе цветов, называемом палитрой. Чаще всего встречаются палитры из 16 и 256 цветов.

Изображения без палитры бывают в какой - либо системе цветопредставления и в градациях серого. Для последних значение каждого пикселя интерпретируется как яркость соответствующей точки. Встречаются изображения с 2, 16 и 256 уровнями серого.

Большинство программ по созданию или обработке растровых изображений предлагают следующие типы изображений, которые отличающиеся количеством выделяемых бит на один пиксель.

Монохромная графика.

Этот режим иногда называют черно - белой графикой, или графикой с однобитовым разрешением. Это означает, что каждый пиксель может быть окрашен только в белый или черный цвет. Такие изображения можно получить с помощью графических пакетов или с помощью сканирования картинок в режиме "Монохромная графика". В этом режиме можно создавать изображения, которые по стилю относятся к художественной черно-белой графике.

Оттенки серого. Изображения этого типа содержат 8 бит на пиксель и позволяют в каждой точке получить 256 оттенков серого цвета. Этот тип изображения можно получить, например, при сканировании черно-белых фотографий в режиме "Оттенки серого". Разумеется, любая цветная фотография также может быть отсканирована в этом режиме, но качество изображения может быть не вполне удовлетворительным, поскольку некоторые цвета из-за недостаточной контрастности могут сливаться.

Индексированный 16 - цветный.

Каждый пиксель здесь представлен 4 битами, в которых записывается номер цвета в палитре из 16-ти цветов. Палитра строится таким образом, что каждый цвет для всех трех составляющих цветов (красного, зеленого и синего) имеет конкретное значение в диапазоне от 0 до 256. Индексированные изображения небольшие по объему и содержат палитру внутри себя или в виде отдельного файла.

Индексированный 256 - цветный.

Аналогично предыдущему типу здесь каждый пиксель представлен номером цвета в палитре, состоящей из 256 цветов. Для хранения информации о пикселе требуется 8 бит.

Истинный цвет RGB.

Этот тип изображений содержит 24 бита на пиксель, и эти 24 бита представляют информацию о трех цветах (красном, зеленом и синем) по 8 бит на каждый цветной компонент. Этот тип изображения передает 16,7 млн. цветов.

Цвет и модели цвета.

Человеческий глаз — тонкий инструмент, но, к сожалению, восприятие цвета субъективно. Очень трудно пересказать другому человеку свое ощущение цвета. Цвет надо видеть.

Вместе с тем для многих отраслей производства, в том числе для полиграфии и компьютерных технологий, необходимы более объективные способы описания и обработки цвета.

Цвет является не физической величиной, а физиологической. Световые лучи, строго говоря, цвета не имеют.

Человек является трихроматом - сетчатка глаза имеет 3 вида рецепторов света, ответственных за цветное зрение. Каждый вид колбочек реагирует на определенный диапазон видимого спектра. Отклик, вызываемый в колбочках светом определенного спектра, называется цветовым стимулом, при этом свет с раз-

ными спектрами может иметь один и тот же цветовой стимул, и таким образом восприниматься одинаково человеком. Это явление называется метамерией - два излучения с разными спектрами, но одинаковыми цветовыми стимулами будут неразличимы человеком.

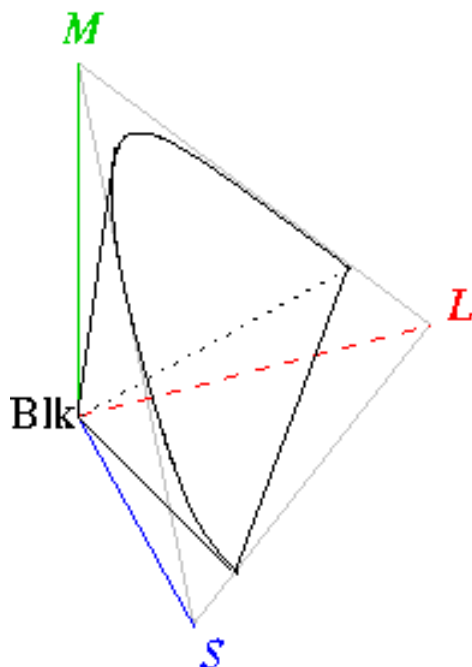


Рис. 15 . Трехмерное представление цветового пространства человека

Можно определить цветовое пространство стимулов как евклидово пространство, если задать координаты x , y , z в качестве значений стимулов, соответствующих отклику колбочек длинно - волнового (L), средне - волнового (M) и коротко - волнового (S) диапазона оптического спектра. Начало координат (S , M , L) = $(0, 0, 0)$ будет представлять черный цвет. Белый цвет не будет иметь четкой позиции в данном определении диаграммы всевозможных цветов, а будет определяться, например, через цветовую температуру, или через определенный баланс белого, или каким либо иным способом. Полное цветовое пространство человека имеет вид конуса в форме подковы (как показано на рисунке). Принципиально данное представление позволяет задавать цвета любой интенсивности - начиная с нуля (черного цвета) до бесконечности. Однако на практике человеческие рецепторы могут перенасытиться или даже быть поврежденными излучением с экстремальной интенсивностью. Поэтому данная мо-

дель не применима для описания цвета в условиях чрезвычайно высоких интенсивностей излучений, и так же не рассматривает вопросы цветовоспроизведения в условиях очень низких интенсивностей (поскольку у человека задействуется иной механизм восприятия через палочки).

Цветовое пространство стимулов имеет свойство аддитивного смешивания - сумма двух цветовых векторов будет соответствовать цвету, равному получаемому смешением этих двух цветов. Таким образом можно описывать любые цвета (вектора цветового пространства), через комбинацию красного, зеленого и синего излучателей основных цветов. На этом принципе основана работа экранов телевизоров и компьютеров. Но важно понимать, что эти устройства не воспроизводят оригинальное излучение (полный спектр), а лишь имитируют изображение, в идеале неотличимое человеком от оригинального.

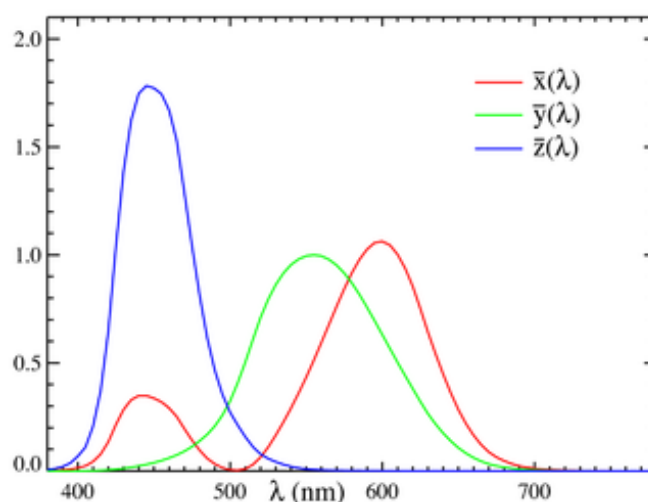


Рис. 16. Функции цветового соответствия Стандартного Колориметрического Наблюдателя.

Функции цветового соответствия Стандартного Колориметрического Наблюдателя, определенные комитетом CIE в 1931 году на диапазоне длин волн от 380 нм до 780 нм (с 5 нм интервалом). Цветовое пространство XYZ - это эталонная цветовая модель, заданная в строгом математическом смысле организацией CIE (International Commission on Illumination — Международная комиссия по освещению) в 1931 году. Модель XYZ является мастер - моделью практически всех остальных цветовых моделей, используемых в технических областях.

Эксперименты, проведенные в конце 20-х и начале 30-х годов, послужили основой для определения функций цветового соответствия (color matching functions). Изначально функции цветового соответствия были выяснены для 20 - го поля зрения (использовался соответствующий колориметр). В 1964 году комитет CIE опубликовал дополнительные данные для 100 - го поля зрения.

Упрощенная схема испытаний выглядит так. Три монохроматических источника света направляются на белый экран таким образом, что их цвет перемешивается. Испытуемый с пультом располагается перед экраном. В его распоряжении три ручки, управляющие яркостью этих источников. На другой стороне экрана воспроизводится точка некоторого эталонного цвета. Задача испытуемого состоит в том, чтобы меняя яркости управляемых источников света, сделать цвет контрольной точки совпадающим с эталонным. Интенсивность источников света меняется от -1 до 1. Лампа при интенсивности 1 работает на полную мощность, при 0 – лампа выключена, при -1 – свет источника вычитается из результирующего. Это достигается путём увеличения соответствующего компонента яркости эталонной точки.

По результатам проведённых испытаний были синтезированы три искусственные функции реакции глаза, зависящие от длины волны света. Они построены таким образом, чтобы упростить расчёты. На основе этих функций строится трёхмерная поверхность, включающая в себя все видимые человеком цвета.

Заметим, что в определении кривых модели XYZ заложен фактор своевольности - форма каждой кривой может быть измерена с достаточной точностью, однако кривая суммарной интенсивности (или сумма всех трех кривых) включает в своем определении субъективный момент. Реципиента просят определить, имеют ли два источника света одинаковую яркость, даже если эти источники абсолютно разного цвета. Также стоит заметить, что относительная нормировка кривых X, Y и Z основана на произвольном выборе, поскольку можно предложить альтернативную работающую модель, в которой кривая чувствительности X имеет двукратно усиленную амплитуду. При этом цветовое пространство будет иметь иную форму. Кривые X, Y, Z в модели CIE XYZ 1931 и 1964 выбра-

ны таким образом, чтобы площади поверхности под каждой кривой были равны между собой.

Работать с трёхмерным изображением неудобно, поэтому поверхность проецируют на плоскость xu .

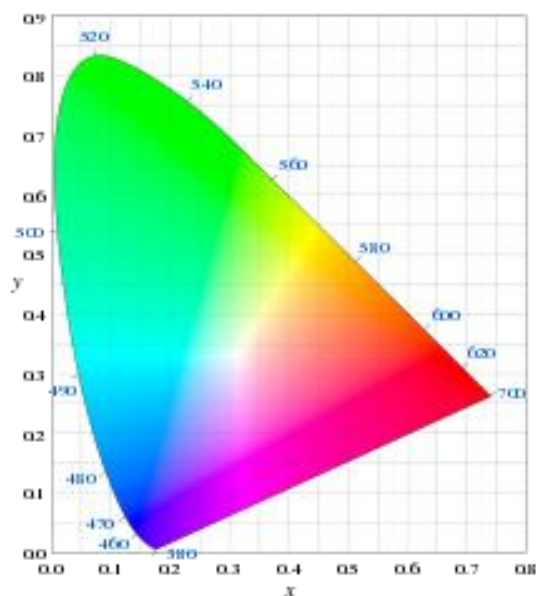


Рис. 17 . Хроматическая диаграмма с длинами волн цветов.

На рисунке справа представлена классическая хроматическая диаграмма модели XYZ с длинами волн цветов. Значения x и y в ней соответствуют X , Y и Z согласно следующим формулам:

$$x = X/(X + Y + Z),$$

$$y = Y/(X + Y + Z).$$

В математическом смысле на данной хроматической диаграмме x и y это координаты на плоскости проекции. Данное представление позволяет задавать значение цвета через светлоту Y (англ. luminance) и две координаты x , y . Заметим что светлота Y в модели XYZ и Y_{xy} это не то же самое что яркость Y в модели YUV или YCbCr.

Основные свойства этой диаграммы следующие:

- любая смесь выбранных компонент лежит на прямой, соединяющей эти цвета;
- геометрическим местом чистых хроматических тонов называется линия, называемая спектральной кривой, все видимые тона располагаются внутри фигуры, ограниченной спектральными кривыми;
- точка, лежащая в центре диаграммы описывает эталонный белый цвет.

Модель RGB.

Множество цветов видны оттого, что светятся (излучаются). К излучаемым цветам можно отнести, например, белый свет, цвета на экране телевизора, монитора, кино, слайд - проектора и так далее. Цветов огромное количество, но из них выделено только три, которые считаются основными (первичными): это — красный, зеленый, синий.

При смешении двух основных цветов результат осветляется: из смешения красного и зеленого получается желтый, из смешения зеленого и синего получается голубой, синий и красный дают пурпурный. Если смешиваются все три цвета, в результате образуется белый. Поэтому такие цвета называются аддитивными.

Модель, в основе которой лежат указанные цвета, носит название модели RGB по первым буквам английских слов Red (Красный), Green (Зеленый), Blue (Синий).

Эта модель представляется в виде трехмерной системы координат. Каждая координата отражает вклад каждой составляющей в результирующий цвет в диапазоне от нуля до максимального значения (его численное в данный момент не играет роли, обычно, это 100% или число 255). В результате получается некий куб, внутри которого и "находятся" все цвета, образуя цветовое пространство.

Важно отметить особенные точки и линии этой модели.

Начало координат: в этой точке все составляющие равны нулю, излучение отсутствует, а это равносильно темноте, т. е. это точка черного цвета.

Точка, ближайшая к зрителю: в этой точке все составляющие имеют максимальное значение, что, как вы уже поняли, дает белый цвет.

На линии, соединяющей эти точки (по диагонали), располагаются серые оттенки: от черного до белого. Это происходит потому, что все три составляющих одинаковы и располагаются в диапазоне от нуля до максимального значения. Этот диапазон иначе называют серой шкалой. В компьютерных технологиях сейчас чаще всего используются 256 градаций (оттенков) серого. Хотя некоторые сканеры имеют возможность кодировать и 1024 оттенка серого,

Три вершины куба дают чистые исходные цвета, остальные три отражают двойные смешения исходных цветов.

Модель CMYK

К отражаемым относятся цвета, которые сами не излучают, а используют белый свет, вычитая из него определенные цвета. Такие цвета называются субтрактивными ("вычитательными"), поскольку они остаются после вычитания основных аддитивных. Понятно, что в таком случае и основных субтрактивных цветов будет три, тем более, что они уже упоминались: голубой, пурпурный, желтый.

Эти цвета составляют так называемую полиграфическую триаду. При печати этими цветами они поглощают красную, зеленую и синюю составляющие белого света таким образом,

что большая часть видимого цветового спектра может быть репродуцирована на бумаге. Каждому пикселю в СМҮК - изображении присваиваются значения, определяющие процентное содержание триадных красок.

При смешениях двух субтрактивных составляющих результирующий цвет затемняется, а при смешении всех трех должен получиться черный цвет. При полном отсутствии краски остается белый цвет (белая бумага).

В итоге получается, что нулевые значения составляющих дают белый цвет, максимальные значения должны давать черный, их равные значения — оттенки серого, кроме того, имеются чистые субтрактивные цвета и их двойные сочетания. Это означает, что модель, в которой они описываются, похожа на модель RGB.

Но проблема заключается в том, что данная модель описывает реальные полиграфические краски, которые не так идеальны, как цветной луч. Они имеют примеси, поэтому не могут полностью перекрыть весь цветовой диапазон, а это приводит, в частности, к тому, что смешение трех основных красок, которое должно давать черный цвет, дает какой-то неопределенный ("грязный") темный цвет, но это скорее темно - коричневый, чем истинно черный цвет.

Для компенсации этого недостатка в число основных полиграфических красок была внесена черная краска. Именно она добавила последнюю букву в название модели СМҮК, хотя и не совсем обычно: С - это Cyan (Голубой), М - это Magenta (Пурпурный), Y - Yellow (Желтый), а К - это black (Черный), т. е. от слова взята не первая, а последняя буква.

Таким образом, модели RGB и СМҮК, хотя и связаны друг с другом, однако их взаимные переходы друг в друга (конвертирование) не происходят без потерь. Тем более, что цветовой охват у них разный. Яркие красные и оранжевые цвета, искрящиеся зелёные и голубые, ускокаивающие фиолетовые – выходят за пределы СМҮК. И речь идет лишь о том, чтобы уменьшить потери до приемлемого уровня. Это вызывает необходимость очень сложных калибровок всех аппаратных частей, составляющих работу с цветом, — сканера (он осуществляет ввод изображения), монитора (по нему судят о цвете и корректируют его), выводного устройства (оно создает оригиналы для печати), печатного станка (выполняющего конечную стадию).

Модель HSB

Если две вышеописанные модели представить в виде единой модели, то получится усеченный вариант цветового круга, в котором цвета располагаются в известном еще со школы порядке: красный (R), желтый (Y), зеленый (G), голубой (C), синий (B). Модель HSB представляется в виде круга, по краю которого расположены спектральные цвета; в треугольнике: по вертикальному катету - насыщенность, а по гипотенузе - яркость. На цветовом круге основ-

ные цвета моделей RGB и CMYK находятся в такой зависимости: каждый цвет расположен напротив дополняющего его (комплементарного) цвета; при этом он находится между цветами, с помощью которых он получен. Например, сложение зеленого и красного цветов дает желтый. Чтобы усилить какой-либо цвет, нужно ослабить дополняющий его цвет (расположенный напротив него на цветовом круге). Например, чтобы изменить общее цветовое решение в сторону голубых тонов, следует снизить в нем содержание красного цвета.

По краю этого цветового круга располагаются так называемые спектральные цвета или цветовые тона (Hue), которые определяются длиной световой волны, отраженной от непрозрачного объекта или прошедшей через прозрачный объект. Цветовой тон характеризуется положением на цветовом круге и определяется величиной угла в диапазоне от 0 до 360 градусов. Эти цвета обладают максимальной насыщенностью, т. е. синий цвет синее быть уже не может.

Насыщенность (Saturation) — это параметр цвета, определяющий его чистоту. Уменьшение насыщенности цвета означает его разбеливание. Цвет с уменьшением насыщенности становится пастельным, блеклым, размытым. На модели все одинаково насыщенные цвета располагаются на концентрических окружностях, т. е. можно говорить об одинаковой насыщенности, например, зеленого и пурпурного цветов, и чем ближе к центру круга, тем все более получаются разбеленные цвета. В самом центре любой цвет максимально разбеливается, проще говоря, становится белым цветом.

Работу с насыщенностью можно характеризовать как добавление в спектральный цвет определенного процента белой краски.

Яркость (Brightness) — это параметр цвета, определяющий освещенность или затемненность цвета. Уменьшение яркости цвета означает его зачернение.

Работу с яркостью можно характеризовать как добавление в спектральный цвет определенного процента черной краски.

В общем случае, любой цвет получается из спектрального цвета добавлением определенного процента белой и черной красок, т. е. фактически серой краски.

Эта модель уже гораздо ближе к традиционному пониманию работы с цветом. Можно определять сначала цветовой тон (Hue), а затем насыщенность (Saturation) и яркость (Brightness). Такая модель получила название по первым буквам приведенных выше английских слов - HSB.

Модель HSB неплохо согласуется с восприятием человека: цветовой тон является эквивалентом длины волны света, насыщенность - интенсивности волны, а яркость - количества света.

Недостатком этой модели является необходимость преобразовывать ее в модель RGB для отображения на экране монитора или в модель CMYK для получения полиграфического оттиска.

Плашечные цвета.

Плашечные – независимые цвета, которые можно либо добавлять к четырём основным триадным цветам, либо использовать вместо любого из них. Если не удаётся, используя триадные цвета, выполнить работу, четырёхкрасочную печать можно сделать плашечными цветами. При этом стоимость работы возрастает. Для каждого дополнительного плашечного цвета требуются дополнительные расходы на краску и печатную форму, на дополнительный прогон бумаги. Даже очень состоятельные компании не позволяют себе использовать более шести цветов на странице (4 CMYK и два дополнительных).

Некоторые производители выделяют отдельные семейства "фотографических" струйных принтеров. Это принтеры, в которых во главу угла ставится именно качество печати, пусть даже в ущерб скорости и экономичности. Здесь также используется печать не в 4, а в 6 цветов. Для того, чтобы напечатать светло - голубой цвет принтер разбрасывает голубые точки настолько редко так, что они "разбавляются" белым цветом бумаги до нужного оттенка. При передаче светлых тонов зерно особенно заметно - точки расположены редко и не сливаются, как в участках с интенсивными цветами. Чтобы улучшить передачу светлых участков, в большинстве новых струйных принтеров введены два дополнительных цвета – светло - голубой и светло - пурпурный. Светлые точки гораздо менее заметны на белой бумаге и с их использованием переходы цветов стали более плавными. Менее распространенный тип принтера - термосублимационный принтер. Совсем недавно цены на сублимационные принтеры измерялись десятками тысяч долларов и позволить их себе могли только полиграфические фирмы для изготовления цветопроб. Сегодня цены на них резко упали, как и на другие компоненты цифровой фотолаборатории. В отличие от струйных принтеров, где краска наносится на лист отдельными цветными точками, в сублимационных принтерах краски действительно смешиваются и дают натуральные фотографические переходы цветов. Процесс печати основан на переходе красителей из твердого состояния в газообразное, минуя жидкую стадию (сублимация).

Форматы графических файлов.

Разные программные пакеты для работы с графикой ориентированы на создание или редактирование графических файлов нескольких типов. Из всего множества разнообразных фор-

матов можно выделить несколько наиболее популярных, поддерживаемых большим числом пакетов. К таким форматам относятся:

Цвета растрового массива преобразуются в формат, совместимый с адаптером дисплея, и передаются аппаратуре вывода видеоизображения.

Существует несколько форматов файлов растровой графики, и каждый формат предусматривает собственный способ кодирования информации о пикселях и другой присущей компьютерным изображениям информации.

Распространенные форматы файлов растровой графики

Формат	Макс. число бит/пиксель	Макс. число цветов	Макс. размер изображения, пиксель	Методы сжатия	Кодирование нескольких изображений
BMP	24	16'777'216	65535 x 65535	RLE*	-
GIF	8	256	65535 x 65535	LZW	+
JPEG	24	16'777'216	65535 x 65535	JPEG	-
PCX	24	16'777'216	65535 x 65535	RLE	-
PNG	48	281'474'976'710'656	2'147'483'647 x 2'147'483'647	Deflation (вариант LZ77)	-
TIFF	24	16'777'216	Всего 4'294'967'295	RLE, LZW и другие	+

Файлы BMP

Формат файла BMP (сокращенно от BitMaP) наиболее близко соответствует внутреннему формату Windows, в котором эта система хранит свои растровые массивы. Для имени файла, представленного в BMP - формате, чаще всего используется расширение BMP, хотя некоторые файлы имеют расширение RLE, означающее run length encoding (кодирование длины серий). Расширение RLE имени файла обычно указывает на то, что произведено сжатие растровой информации файла одним из двух способов сжатия RLE, которые допустимы для файлов BMP - формата.

В файлах BMP информация о цвете каждого пикселя кодируется 1, 4, 8, 16 или 24 бит (бит/пиксель). Числом бит/пиксель, называемым также глубиной представления цвета, определяется максимальное число цветов в изображении. Изображение при глубине 1 бит/пиксель может иметь всего два цвета, а при глубине 24 бит/пиксель - более 16 млн. различных цветов.

Структура файла BMP

Заголовок файла растровой
графики (14 байт)

Сигнатура файла BMP (2 байт) (exe-файлов первыми двумя байтами в файле являются "MZ", для zip-архивов "PK").

Размер файла (4 байт)

Не используется (2 байт)

Не используется (2 байт)

Местонахождение данных растрового
массива (4 байт)

Информационный заголовок
растрового массива (40 байт)

Длина этого заголовка (4 байт)

Ширина изображения (4 байт)

Высота изображения (4 байт)

Число битовых плоскостей (2 байт)

Бит/пиксель (2 байт)

Метод сжатия (4 байт)

Длина растрового массива (4 байт)

Горизонтальное разрешение (4 байт)

Вертикальное разрешение (4 байт)

Число цветов изображения (4 байт)

Число основных цветов (4 байт)

Таблица цветов (длина изменяется
от 8 до 1024 байт)

Собственно данные растрового
массива (длина переменная)

На приведенной схеме показана структура типичного BMP - файла, содержащего 256-цветное изображение (с глубиной 8 бит/пиксель). Файл разбит на четыре основные раздела: заголовок файла растровой графики, информационный заголовок растрового массива, таблица цветов и собственно данные растрового массива. Заголовок файла растровой графики содержит информацию о файле, в том числе адрес, с которого начинается область данных растрового массива. В информационном заголовке растрового массива содержатся сведения об изображении, хранящемся в файле, например, его высоте и ширине в пикселях. В таблице цветов представлены значения основных цветов RGB (красный, зеленый, синий) для используемых в изображении цветов. Программы, считывающие и отображающие BMP - файлы, в случае использования видеоадаптеров, которые не позволяют отображать более 256 цветов, для точной цветопередачи могут программно устанавливать такие значения RGB в цветовых палитрах адаптеров. Формат собственно данных растрового массива в файле BMP зависит от числа бит, используемых для кодирования данных о цвете каждого пикселя. При 256 - цветном изображении каждый пиксель в той части файла, где содержатся собственно данные растрового массива, описывается одним байтом (8 бит). Это описание пикселя не представляет значений цветов RGB, а служит указателем для входа в таблицу цветов файла. Таким

образом, если в качестве первого значения цвета RGB в таблице цветов файла BMP хранится R/G/B=255/0/0, то значению пикселя 0 в растровом массиве будет поставлен в соответствие ярко-красный цвет. Значения пикселей хранятся в порядке их расположения слева направо, начиная (как правило) с нижней строки изображения. Таким образом, в 256 - цветном BMP - файле первый байт данных растрового массива представляет собой индекс для цвета пикселя, находящегося в нижнем левом углу изображения; второй байт представляет индекс для цвета соседнего справа пикселя и т. д. Если число байт в каждой строке нечетно, то к каждой строке добавляется дополнительный байт, чтобы выровнять данные растрового массива по 16 – бит. границам. Не все файлы BMP имеют структуру, подобную показанной на схеме. Например, файлы BMP с глубиной 16 и 24 бит/пиксель не имеют таблиц цветов; в этих файлах значения пикселей растрового массива непосредственно характеризуют значения цветов RGB.

Файлы PCX.

PCX стал первым стандартным форматом графических файлов для хранения файлов растровой графики в компьютерах IBM PC. На этот формат, применявшийся в программе Paintbrush фирмы ZSoft, в начале 80-х гг. фирмой Microsoft была приобретена лицензия, и затем он распространялся вместе с изделиями Microsoft. В дальнейшем формат был преобразован в Windows Paintbrush и начал распространяться с Windows. Хотя область применения этого популярного формата сокращается, файлы формата PCX все еще широко распространены сегодня.

Файлы PCX разделены на следующие три части: заголовок PCX, данные растрового массива и факультативная таблица цветов. 128 - байт заголовок PCX содержит несколько полей, в том числе поля размера изображения и числа бит для кодирования информации о цвете каждого пикселя. Информация растрового массива сжимается с использованием простого метода сжатия RLE; факультативная таблица цветов в конце файла содержит 256 значений цветов RGB, определяющих цвета изображения. Формат PCX первоначально был разработан для адаптеров CGA - и EGA - дисплеев и в дальнейшем был модифицирован для использования в адаптерах VGA и адаптерах истинных цветов. Кодирование цвета каждого пикселя в современных изображениях PCX может производиться с глубиной 1, 4, 8 или 24 бит.

Файлы TIFF.

Если PCX - один из самых простых для декодирования форматов растровой графики, то TIFF (Tagged Image File Format, формат файлов изображения, снабженных тегами) - один из самых сложных. Файлы TIFF имеют расширение TIFF. Каждый файл начинается 8 - байт заголовком файла изображения (IFH), важнейший элемент которого - каталог файла изображения (Image File Directory, IFD) - служит указателем к структуре данных. IFD представляет

собой таблицу для идентификации одной или нескольких порций данных переменной длины, называемых тегами; теги хранят информацию об изображении. В спецификации формата файлов TIFF определено более 70 различных типов тегов. Например, тег одного типа хранит информацию о ширине изображения в пикселях, другого - информацию о его высоте. В теге третьего типа хранится таблица цветов (при необходимости), а тег четвертого типа содержит сами данные растрового массива. Изображение, закодированное в файле TIFF, полностью определяется его тегами, и этот формат файла легко расширяется, поскольку для придания файлу дополнительных свойств достаточно лишь определить дополнительные типы тегов. Так что же делает TIFF столь сложным? С одной стороны, составление программ, различающих все типы тегов, - это непростое дело. В большинстве программ для чтения файлов TIFF реализуется только подмножество тегов, именно поэтому созданный одной программой файл TIFF иногда не может быть прочитан другой. Кроме того, программы, создающие файлы TIFF, могут определять собственные типы тегов, имеющие смысл только для них. Программы чтения файлов TIFF могут пропускать непонятные для них теги, но всегда существует опасность, что это повлияет на внешний вид изображения. Еще одна сложность заключается в том, что файл TIFF может содержать несколько изображений, каждому из которых сопутствуют собственный IFD и набор тегов. Данные растрового массива в файле TIFF могут сжиматься с использованием любого из нескольких методов, поэтому в надежной программе для чтения файлов TIFF должны быть средства распаковки RLE, LZW (LempelZivWelch) и несколько других. Ситуацию еще больше ухудшает то обстоятельство, что пользование программами распаковки LZW должно осуществляться в соответствии с лицензионным соглашением с фирмой Unisys Corp. на право пользования алгоритмом LZW и часто за плату. В результате даже самые лучшие программы считывания TIFF нередко "сдаются", когда сталкиваются со сжатым по методу LZW изображением.

Несмотря на свою сложность, файловый формат TIFF остается одним из лучших для передачи растровых массивов с одной платформы на другую благодаря своей универсальности, позволяющей кодировать в двоичном виде практически любое изображение без потери его визуальных или каких-либо иных атрибутов.

Файлы GIF.

Большинство ведущих специалистов - графиков, имеющих дело с алгоритмом LZW, сталкиваются с аналогичными юридическими проблемами при использовании популярного межплатформенного формата файлов растровой графики GIF (Graphics Interchange Format - формат обмена графическими данными), разработанного компанией CompuServe. Обычно для имени файлов GIF используется расширение GIF. Структура файла GIF зависит от версии GIF - спецификации, которой соответствует файл. В настоящее время используются две вер-

сии, GIF87a и GIF89a. Первая из них проще. Независимо от номера версии, файл GIF начинается с 13 - байт заголовка, содержащего сигнатуру, которая идентифицирует этот файл в качестве GIF - файла, номер версии GIF и другую информацию. Если файл хранит всего одно изображение, вслед за заголовком обычно располагается общая таблица цветов, определяющая цвета изображения. Если в файле хранится несколько изображений (формат GIF, аналогично TIFF, позволяет в одном файле кодировать два и больше изображений), то вместо общей таблицы цветов каждое изображение сопровождается локальной таблицей цветов.

В файле GIF87a вслед за заголовком и общей таблицей цветов размещается изображение, которое может быть первым из нескольких располагаемых подряд изображений. Каждое изображение состоит из 10 - байт описателя изображения, расположенной вслед за ним локальной таблицы цветов и битов растрового массива. Для повышения эффективности использования памяти данные растрового массива сжимаются с помощью алгоритма LZW.

Файлы GIF89a имеют аналогичную структуру, но они могут содержать факультативные блоки расширения с дополнительной информацией о каждом изображении. В спецификации GIF89a определены четыре типа блоков расширения. Это блоки расширения для управления графикой, которые описывают, как изображение должно выводиться на экран (например, накладывается ли оно на предыдущее изображение подобно диапозитиву или просто заменяет его); блоки расширения с обычным текстом, содержащие текст, отображаемый вместе с графикой; блоки расширения для комментария, содержащие комментарии в коде ASCII; и блоки расширения прикладных программ, в которых хранится информация, принадлежащая только создавшей этот файл программе. Блоки расширения могут находиться практически в любом месте файла после общей таблицы цветов.

Основные достоинства GIF заключаются в широком распространении этого формата и его компактности. Но ему присущи два достаточно серьезных недостатка. Один из них состоит в том, что в изображениях, хранящихся в виде GIF - файла, не может быть использовано более 256 цветов. Второй, возможно, еще более серьезный, заключается в том, что разработчики программ, использующие в них форматы GIF, должны иметь лицензионное соглашение с CompuServe и вносить плату за каждый экземпляр программы; такая ценовая политика была принята CompuServe после того, как Unisys объявила, что начнет добиваться соблюдения своих прав собственности и потребовала от тех, кто пользуется алгоритмом сжатия LZW, вносить лицензионные платежи. Возникшее в результате этого запутанное юридическое положение тормозит внедрение программистами в свои графические программы средств для работы с файлами GIF.

Файлы PNG

Формат PNG (Portable Network Graphic - переносимый сетевой формат) был разработан для замены GIF, чтобы обойти юридические препятствия, стоящие на пути использования GIF-файлов. PNG унаследовал многие возможности GIF и, кроме того, он позволяет хранить изображения с истинными цветами. Еще более важно, что он сжимает информацию растрового массива в соответствии с вариантом пользующегося высокой репутацией алгоритма сжатия LZ77 (предшественника LZW), которым любой может пользоваться бесплатно.

Файлы JPEG

Формат файла JPEG (Joint Photographic Experts Group - Объединенная экспертная группа по фотографии) был разработан компанией C - Cube Microsystems как эффективный метод хранения изображений с большой глубиной цвета, например, получаемых при сканировании фотографий с многочисленными едва уловимыми (а иногда и неуловимыми) оттенками цвета. Самое большое отличие формата JPEG от других рассмотренных здесь форматов состоит в том, что в JPEG используется алгоритм сжатия с потерями (а не алгоритм без потерь) информации. Алгоритм сжатия без потерь так сохраняет информацию об изображении, что распакованное изображение в точности соответствует оригиналу. При сжатии с потерями приносится в жертву часть информации об изображении, чтобы достичь большего коэффициента сжатия. Распакованное изображение JPEG редко соответствует оригиналу абсолютно точно, но очень часто эти различия столь незначительны, что их едва можно (если вообще можно) обнаружить. Реальные фотографические изображения часто совсем невозможно сжать с помощью методов сжатия без потерь, поэтому 50% - ное сжатие следует признать достаточно хорошим. С другой стороны, применяя методы сжатия без потерь, можно сжимать некоторые изображения на 90%. Такие изображения плохо подходят для сжатия методом JPEG.

WMF - метафайл Windows, созданный фирмой Microsoft для хранения векторной и растровой графики. Формат может создавать и хранить изображения вплоть до 24 бит на пиксель, передавая высококачественные изображения. Изображения в этом формате хорошо масштабируются.

Битовые плоскости. Растровое устройство можно рассматривать как матрицу дискретных ячеек (точек), каждая из которых может быть подсвечена. Таким образом, оно является точно - рисующим устройством.

Чаще всего для графических устройств с растровой ЭЛТ используется буфер кадра. Буфер кадра представляет собой большой непрерывный участок памяти компьютера. Для каждой точки или пикселя, в растре отводится как минимум один бит памяти. Эта память называется битовой плоскостью. Для квадратного раstra размером $512 * 512$ требуется $218 = 5122 =$

262144 бита памяти в одной битовой плоскости. Изображение в буфере кадра строится по-битно. Из-за того, что бит памяти имеет только два состояния, имея одну битовую плоскость, можно получить лишь черно-белое изображение. Битовая плоскость является цифровым устройством, тогда как растровая ЭЛТ - аналоговое устройство, для работы которого требуется электрическое напряжение. Поэтому при считывании информации из буфера кадра и ее выводе на графическое устройство с растровой ЭЛТ должно происходить преобразование из цифрового представления в аналоговый сигнал. Такое преобразование выполняет цифро-аналоговый преобразователь (ЦАП). Каждый пиксель буфера кадра должен быть считан и преобразован, прежде чем он будет отображен на растровой ЭЛТ. На **рис.** приведена схема графического устройства с черно-белой растровой ЭЛТ, построенного на основе буфера кадра с одной битовой плоскостью.

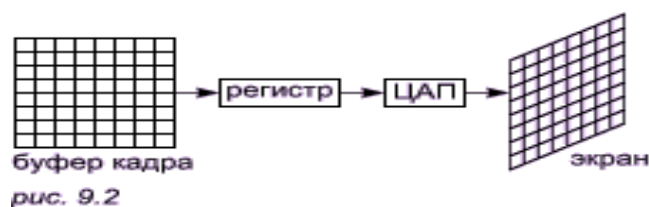


Рис. . Графическое устройство.

Цвета или полутона серого цвета могут быть введены в буфер кадра путем использования дополнительных битовых плоскостей. На **рис.** показаны схема буфера кадра с N битовыми плоскостями для градаций серого цвета. Интенсивность каждого пикселя на ЭЛТ управляется содержимым соответствующих пикселей в каждой из N битовых плоскостей. В соответствующую позицию регистра загружается бинарная величина (0 или 1) из каждой плоскости. Двоичное число, получившееся в результате, интерпретируется как уровень интенсивности между 0 и 2^N-1 . С помощью ЦАП это число преобразуется в напряжение между 0 (темный экран) и 2^N-1 (максимальная интенсивность свечения). Всего можно получить 2^N уровней интенсивности. **Рис.** иллюстрирует систему с тремя битовыми плоскостями для 8 (23) уровней интенсивности. Для каждой битовой плоскости требуется полный объем памяти при данном разрешении раstra: например, буфер кадра с тремя битовыми плоскостями для раstra $512 * 512$ занимает 786432 ($3 * 512 * 512$) битов памяти.

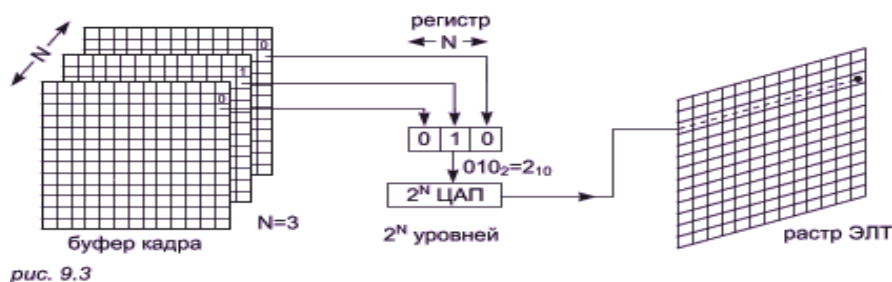


Рис. . Схема буфера кадра с N битовыми плоскостями.

Классы изображений.

Для того, чтобы корректнее оценивать степень сжатия, нужно ввести понятие класса изображений. Под классом будет пониматься некая совокупность изображений, применение к которым алгоритма архивации дает качественно одинаковые результаты. Например, для одного класса алгоритм дает очень высокую степень сжатия, для другого - почти не сжимает, для третьего - увеличивает файл в размере. Известно, что многие алгоритмы в худшем случае увеличивают файл. Рассмотрим следующие примеры неформального определения классов изображений:

изображения с небольшим количеством цветов (4-16) и большими областями, заполненными одним цветом (примеры: деловая графика - гистограммы, диаграммы, графики и т.п.);

изображения, с использованием плавных переходов, построенные на компьютере (примеры: графика презентаций, эскизные модели в САПР, изображения построенные по методу закрашки Гуро);

фотореалистичные изображения (пример: отсканированные фотографии);

фотореалистичные изображения с наложением деловой графики.

В качестве отдельных классов могут быть предложены некачественно отсканированные в 256 градаций серого цвета страницы журналов или растровые изображения топографических карт. Формально являясь 8 или 24-битными, они несут даже не растровую, а векторную информацию. Отдельные классы могут образовывать и совсем специфичные изображения: рентгеновские снимки или фотографии в профиль и фас из электронного досье.

Достаточно сложной и интересной задачей является поиск наилучшего алгоритма для конкретного класса изображений.

Алгоритмы сжатия растровой графики.

Для того, чтобы корректно оценивать направление изменения алгоритмов, недостаточно определить только классы изображений. Необходимо задать и определенные критерии:

Худший, средний и лучший коэффициенты сжатия. То есть доля, на которую возрастет размер изображения, если исходные данные будут наихудшими; некий среднестатистический коэффициент для того класса изображений, на который ориентирован алгоритм; и, наконец, лучший коэффициент. Последний необходим лишь теоретически, поскольку показывает степень сжатия наилучшего (как правило, абсолютно черного) изображения, не редко - фиксированного размера.

Класс изображений, на который ориентирован алгоритм. Иногда указано также, почему на других классах изображений получаются худшие результаты.

Симметричность. Характеризует ресурсоемкость процессов кодирования и декодирования. Для нас наиболее важными являются два коэффициента: отношение времени кодирования ко времени декодирования и требования на память.

Есть ли потери качества? И если есть, то за счет чего изменяется коэффициент архивации? Дело в том, что у большинства алгоритмов сжатия с потерей информации существует возможность изменения коэффициента сжатия.

Характерные особенности алгоритма и изображений, к которым его применяют.

Приложения, использующие статическую графику.

Ниже представлены приложения, использующие статическую графику, и требования, которые они выдвигают к алгоритмам сжатия.

Большое распространение сейчас получили издательские системы. Программы верстки типа PageMaker, QuarkXPress, Ventura есть на очень многих персональных компьютерах. В подобных системах приходится иметь дело с полноцветными изображениями самого разного размера (от 640*480 до 3000*2000) и с большими двуцветными изображениями. Поскольку иллюстрации занимают львиную долю от общего объема материала в документе, проблема хранения стоит очень остро. Изображение, соответствующее рекламной странице журнала, занимает до 20 Мб. А в номере их, естественно, несколько. Кроме того, немало могут занимать и иллюстрации к самим статьям. В результате средний журнал в 100 страниц может занимать больше 500 Мб. То же самое относится и к хорошо изданным книгам, буклетам, брошюрам. Проблемы также создает большая разнородность иллюстраций. Единственное, что можно сказать заранее, это то, что будут преобладать фотореалистичные изображения и деловая графика.

Другим примером являются справочники и энциклопедии на CD - ROM. Несмотря на то, что емкость одного диска довольно велика (примерно 650 Мб), ее, как правило, не хватает. При создании энциклопедий и игр большую часть диска занимают статические изображения и видео. Таким образом, для этого класса приложений актуальность приобретают существенно асимметричные алгоритмы.

Похожие требования к алгоритмам архивации выдвигает и быстро развивающаяся система Internet. В этой гипертекстовой системе достаточно активно используются иллюстрации. При оформлении информационных или рекламных страниц хочется сделать их более яркими и красочными. Больше всего при этом страдают пользователи, подключенные к сети с помощью медленных каналов связи. Если страница WWW перенасыщена графикой, то ожидание ее полного появления на экране может затянуться. Поскольку при этом нагрузка на процессор мала, то здесь могут найти применение эффективно сжимающие сложные алгоритмы со сравнительно большим временем разархивации.

Свое применение машинная графика находит и в различных информационных системах. Например, уже становится привычным исследовать ультразвуковые и рентгеновские снимки не на бумаге, а на экране монитора. В электронный вид переводят и истории болезней. Понятно, что хранить эти материалы логичнее в единой картотеке. При этом без использования специальных алгоритмов большую часть архивов займут фотографии. Поэтому при создании эффективных алгоритмов решения этой задачи нужно учесть специфику рентгеновских снимков - преобладание размытых участков.

В геоинформационных системах - при хранении аэрофотоснимков местности - специфическими проблемами являются большой размер изображения и необходимость выборки лишь части изображения по требованию. Кроме того, может потребоваться масштабирование. Это неизбежно накладывает свои ограничения на алгоритм компрессии.

В электронных картотеках и досье различных служб для изображений характерно подобие между фотографиями в профиль, и подобие между фотографиями в фас, которое также необходимо учитывать при создании алгоритма архивации. Подобие между фотографиями наблюдается и в любых других специализированных справочниках. В качестве примера можно привести энциклопедии птиц или цветов. Однако там различие между рисунками значительно больше и использовать подобие сложнее.

Особенности алгоритмов.

Первыми для архивации изображений стали применяться привычные алгоритмы. Те, что использовались и используются в системах резервного копирования, при создании дистрибутивов и т.п. Старые алгоритмы перестали удовлетворять требованиям, предъявляемым к архивации. Многие изображения практически не сжимались, хотя "на взгляд" обладали явной избыточностью. Это привело к созданию нового типа алгоритмов - сжимающих с потерей информации. Как правило, в них можно задавать коэффициент архивации и, следовательно, степень потерь качества. При этом достигается компромисс между размером и качеством изображений.

Одна из серьезных проблем машинной графики заключается в том, что до сих пор не найден адекватный критерий оценки потерь качества изображения. А теряется оно постоянно - при оцифровке, при переводе в ограниченную палитру цветов, при переводе в другую систему цветопредставления для печати, и, что для нас особенно важно, при архивации с потерями. Можно привести пример простого критерия: среднеквадратичное отклонение значений пикселей согласно которому изображение будет сильно испорчено при понижении яркости всего на 5% (глаз этого не заметит - у разных мониторов настройка яркости варьируется гораздо сильнее). В то же время изображения со "снегом" - резким изменением цвета отдельных точек, слабыми полосами или "муаром" будут признаны "почти не изменившимися".

Свои неприятные стороны есть и у других критериев. Таким образом, необходим критерий, учитывающий всевозможные пространственные регулярные эффекты, который, оказывается, не так просто построить.

Лучше всего потери качества изображений оценивают наши глаза. Отличной считается архивация, при которой невозможно на глаз различить первоначальное и раскодированное изображение. Хорошей - когда сказать, какое из изображений подвергалось архивации, можно только сравнивая две находящиеся рядом картинки.

При дальнейшем увеличении степени сжатия, как правило, становятся заметны побочные эффекты, характерные для данного алгоритма. На практике, даже при отличном сохранении качества, в изображение могут быть внесены специфические регулярные изменения. Поэтому алгоритмы архивации с потерями не рекомендуется использовать при сжатии изображений, которые в дальнейшем собираются либо печатать с высоким качеством, либо обрабатывать программами распознавания образов.

Следует сделать следующую оговорку. Один и тот же алгоритм часто можно реализовать разными способами. Многие известные алгоритмы, такие как RLE, LZW или JPEG, имеют десятки различающихся реализаций. Кроме того, у алгоритмов бывает несколько явных параметров, варьируя которые, можно изменять характеристики процессов архивации и разархивации. При конкретной реализации эти параметры фиксируются, исходя из наиболее вероятных характеристик входных изображений, требований на экономию памяти, требований на время архивации и т.д. Поэтому у алгоритмов одного семейства лучший и худший коэффициенты могут отличаться, но качественно картина не изменится.

Итак, методы сжатия растровой информации делятся на две большие группы: сжатие с потерями и сжатие без потерь. Методы сжатия без потерь дают более низкий коэффициент сжатия, но зато сохраняют точное значение пикселей исходного изображения. Методы с потерями дают более высокие коэффициенты сжатия, но не позволяют воспроизвести первоначальное изображение с точностью до пикселя. Человеческий глаз не воспринимает все тонкие оттенки цвета в обычном растровом изображении. Таким образом, некоторые детали могут быть опущены без видимого нарушения информационного содержания картинки.

Алгоритмы сжатия без потерь.

Групповое кодирование - Run Length Encoding (RLE) - один из самых старых и самых простых алгоритмов архивации графики. Изображение в нем (как и в нескольких алгоритмах, описанных далее) вытягивается в цепочку байт по строкам раstra. Само сжатие в RLE происходит за счет того, что в исходном изображении встречаются цепочки одинаковых байт. Замена их на пары "счетчик, значение" уменьшает избыточность данных. Лучший, средний и худший коэффициенты сжатия - $1/32$, $1/2$, $2/1$. Ситуация, когда файл увеличивается в два

раза, для этого простого алгоритма не так уж редка. Ее можно легко получить, применяя групповое кодирование к обработанным цветным фотографиям. Последовательность действий при групповом кодировании следующая:

Начиная с первой строки, программа группового кодирования просматривает значения пикселей слева направо и ищет отрезки повторяющихся пикселей. Всякий раз, когда встречаются три или более идущих подряд пикселей с одинаковым значением, программа заменяет их парой чисел: первое число указывает длину отрезка, второе - значение пикселей. Число, определяющее длину отрезка, будем называть меткой отрезка.

Графическая программа декодирует изображение, считывая сжатый файл и восстанавливая отрезки повторяющихся значений пикселей. Заметим, что восстановленное изображение полностью совпадает с оригиналом.

К положительным сторонам алгоритма, пожалуй, можно отнести только то, что он не требует дополнительной памяти при работе, и быстро выполняется. Ориентирован алгоритм на изображения с небольшим количеством цветов: деловую и научную графику. Применяется в форматах PCX, TIFF, BMP.

LZW

Собственно исходный Lempel/Ziv подход к сжатию данных был впервые обнародован в 1977г., а усовершенствованный (Terry Welch) вариант был опубликован в 1984г. LZW - код (Lempel-Ziv & Welch) является на сегодняшний день одним из самых распространенных кодов сжатия без потерь. Именно с помощью LZW-кода осуществляется сжатие в таких графических форматах, как TIFF и GIF, с помощью модификаций LZW осуществляют свои функции очень многие универсальные архиваторы. Существует довольно большое семейство LZ-подобных алгоритмов, различающихся, например, методом поиска повторяющихся цепочек.

Работа алгоритма основана на поиске во входном файле повторяющихся последовательностей символов, которые кодируются комбинациями длиной от 8 до 12 бит. Таким образом, наибольшую эффективность данный алгоритм имеет на текстовых файлах и на графических файлах, в которых имеются большие одноцветные участки или повторяющиеся последовательности пикселей. Реализация алгоритма LZW жестко зафиксирована международным стандартом и Американским Национальным институтом стандартов (ANSI), однако существуют достаточно интересные его модификации, которые дают больший коэффициент сжатия некоторых специфичных типах файлов - например, на исходных текстах программ.

Коэффициенты сжатия: 1/1000, 1/4, 7/5. Коэффициент 1/1000 достигается только на одноцветных изображениях размером больше 4 Мб. Ориентирован LZW на 8 - битные изображения, построенные на компьютере. Ситуация, когда алгоритм увеличивает изображение, встречается крайне редко.

Отсутствие потерь информации при LZW-кодировании обусловило широкое распространение основанного на нем формата TIFF. Этот формат не накладывает каких-либо ограничений на размер и глубину цвета изображения и широко распространен, например, в полиграфии. Другой основанный на LZW формат - GIF - более примитивен - он позволяет хранить изображения с глубиной цвета не более 8 бит/пиксель. В начале GIF - файла находится палитра. Это таблица, устанавливающая соответствие между индексом цвета - числом в диапазоне от 0 до 255 и истинным, 24 - битным значением цвета.

Таким образом, этот формат можно назвать форматом без потерь лишь в том смысле, что все потери информации происходят до LZW - кодирования - при преобразовании исходной картинки в 8 - битную с индексированной палитрой.

Несомненным достоинством этого формата является возможность хранить в одном файле последовательности изображений, образующих примитивную анимацию. Именно благодаря этой особенности он нашел широкое применение в Internet.

Алгоритм компрессии с кодом переменной длины LZW является разновидностью алгоритма компрессии Lempel - Ziv, в котором коды переменной длиной используются для замены сочетаний, обнаруженных в исходных данных. В данном алгоритме используется код, либо переводная таблица, составленная из комбинаций, встретившихся в исходных данных. Каждая новая комбинация заносится в эту таблицу, и затем в потоке данных, подлежащем компрессии, замещается определенным индексом.

Программа компрессии получает входные данные и строит код или таблицу перевода из новых ключевых комбинаций по мере того, как они встречаются в потоке данных. Каждая новая комбинация заносится в таблицу кодов, а ее индекс записывается в исходящий поток данных. Если программа компрессии сталкивается с комбинацией, которая уже встречалась, то вместо нее в исходящий поток данных она ставит соответствующий индекс из таблицы, таким образом достигается общая компрессия данных. Программа декомпрессии получает на вход сжатый поток данных и воссоздает по нему код, либо таблицу перевода. В процессе обработки потока сжатых данных эти коды используются в качестве индексов таблицы, а соответствующие им комбинации заносятся в поток восстанавливаемых данных. Отличительная черта алгоритма - код переменной длины - напрямую связана с исходным размером кода, сообщаящим, сколько бит изначально было использовано в процессе компрессии для записи кода. Если количество комбинаций, обнаруженных программой компрессии в исходном потоке данных, превышает количество образцов, которое можно было бы записать в таблицу при данном размере кода в соответствии с алгоритмом LZW, то количество бит, предоставляемых для записи кода LZW, автоматически увеличивается на единицу.

Алгоритм Хаффмана.

Кодирование Хаффмана, вероятно, самый известный и классический метод сжатия данных. Простота и элегантность способа сделали его на долгое время академическим фаворитом. Но коды Хаффмана имеют и практическое применение; например, статические коды Хаффмана используются на последнем этапе сжатия JPEG. Кодирование Шеннона - Фано (Shannon - Fano), довольно близкое к кодированию Хаффмана, используется как один из этапов в мощном "imploding" - алгоритме программы PKZIP.

Кодирование Хаффмана работает на предпосылке, что некоторые символы используются в представлении данных чаще, чем другие. Наиболее общее представление - алфавит ASCII - использует 8 бит для каждого символа. В английском языке буква e явно будет чаще встречаться, чем буква q, хотя мы используем для их представления одинаковое количество бит. Если мы используем только 4 бита для e и 12 бит для q, мы могли бы выиграть несколько бит, сохраняя английский текст.

Использует только частоту появления одинаковых байт в изображении. Сопоставляет символам входного потока, которые встречаются большее число раз, цепочку бит меньшей длины, и напротив - встречающимся редко - цепочку большей длины. Для сбора статистики требует двух проходов по изображению. Коэффициенты сжатия: 1/8, 2/3, 1. Требуется записи в файл таблицы соответствия кодируемых символов и кодирующих цепочек.

На практике используются его разновидности. Так, в некоторых случаях резонно либо использовать постоянную таблицу, либо строить ее "адаптивно", т.е. в процессе архивации/разархивации. Эти приемы избавляют от двух проходов по изображению и необходимости хранения таблицы вместе с файлом. Кодирование с фиксированной таблицей применяется в качестве последнего этапа архивации в JPEG.

Близкая модификация алгоритма используется при сжатии черно-белых изображений. Последовательности подряд идущих черных и белых точек заменяются числом, равным их количеству с признаком цвета, а этот ряд уже, в свою очередь, сжимается по Хаффману с фиксированной таблицей. Этот алгоритм реализован также и в формате TIFF.

Кодирование Хаффмана формализует идею связи длины символа с вероятностью появления символов. Статическое кодирование Хаффмана требует, чтобы у вас была таблица вероятностей, прежде чем вы начнете сжимать данные. Эта таблица может быть взята из результатов статистических исследований (такие таблицы были опубликованы для некоторых данных, например, для английского языка), или же система сжатия может просканировать входные данные для определения вероятностей символов, прежде чем начать сжимать данные.

С помощью этой информации о вероятностях компрессор и декомпрессор могут сконструировать кодирующее дерево.

Лист дерева - это узел, из которого не выходит ни одной дуги. Два листа, имеющие общий узел, называются родственными.

Двоичное дерево - дерево, у которого из каждого угла выходит только две дуги.

Н - дерево - двоичное дерево кодирования, у которого каждый узел имеет вес, причем вес родителя равен сумме весов детей.

Входной алфавит - совокупность всех символов.

Алгоритм Хаффмана: Кодированное дерево является двоичным деревом с одним листом для каждого символа. Чтобы построить дерево, компрессор начинает с двух символов наименьшей вероятности. Затем он объединяет их как два листа двумя ветвями в узел; этому узлу, в свою очередь, назначается сумма двух вероятностей. Компрессор затем рассматривает этот узел вместе с оставшимися символами в списке вероятностей, и снова выбирает два наименее вероятных элемента. Он продолжает строить и объединять узлы, пока не построит единое дерево с вероятностью корня, равной 1.

Полученное дерево имеет листья с различным расстоянием от корня. Листья, которые представляют символы с наивысшей вероятностью, самые близкие к корню, тогда как символы с наименьшей вероятностью вынесены подальше.

Для кодировки символа компрессор находит путь от корня дерева к листу символа. Предположим, что компрессор хочет закодировать букву *s*. Он начинает с листа, соответствующего букве *s*, и переходит в родительский узел, отмечая при этом, по какой ветви он прошел (по 0 или по 1). Он продолжает идти вверх по дереву, пока не достигнет корня. Список ветвей в обратном порядке описывает путь от корня до *s*: это и есть код Хаффмана для этого символа.

Символы с большой вероятностью находятся ближе к корню, так что их коды короткие. Символы с малой вероятностью дальше от корня, и их коды длиннее.

Чтобы декодировать, декомпрессор берет код и обрабатывает его в обратном порядке. Это значит, что он начинает с корня дерева. Если первый бит кода равен 1, то декомпрессор идет от корня в узел по ветви 1. Он продолжает чтение битов и проход по ветвям, пока не достигнет листа; символ в листе и есть декодированный символ.

Следует упомянуть еще об одном свойстве дерева Хаффмана. Поскольку символы всегда являются листьями, узлы символов никогда не имеют потомков. Когда декомпрессор получает лист, он знает, что он должен немедленно прекратить чтение, потому что он знает, что пришел в лист. Другими словами, один код Хаффмана никогда не будет префиксом другого. Это значит, что хотя длина кода меняется, декомпрессор всегда знает, когда кончается один код и начинается другой, и нет необходимости явно помещать разделители между кодами.

Динамическое кодирование Хаффмана.

Самой большой сложностью с кодами Хаффмана является необходимость иметь таблицы вероятностей для каждого типа сжимаемых данных. Это не представляет проблемы, если вы знаете, что всегда будете сжимать английский текст; вы просто предоставляете компрессору и декомпрессору подходящее для английского текста дерево. Протокол JPEG определяет дерево Хаффмана, используемое по умолчанию для сжатия данных JPEG. В общем же случае, когда не известны вероятности символов для ваших входных данных, статические коды Хаффмана не могут использоваться эффективно.

Динамическая версия сжатия Хаффмана может строить дерево Хаффмана "на лету" во время чтения и активного сжатия. Дерево постоянно обновляется, чтобы отражать изменения вероятностей входных данных.

Ключом к началу инициализированного дерева является введение пустого листа. Пустой лист - это просто лист без присоединенного к нему символа; этот лист имеет нулевую вероятность. Начальное дерево как в компрессоре, так и в декомпрессоре, имеет только корень и единственный пустой лист.

Компрессор присоединяет символ к ветви 1 корня, оставляя пустой лист на ветви 0. Затем он посылает этот символ в декомпрессор как буквальный код ASCII, и декомпрессор выполняет точно такое же действие над своим деревом.

Для каждого символа, прочитанного после этого, компрессор выполняет следующие шаги. Вначале он проверяет, находится ли код в кодовом дереве. Если код есть, компрессор выдает его так же, как и в случае статического сжатия. Если же нет, он посылает код для пустого листа. Затем он посылает новый символ как буквальный код ASCII. Наконец, компрессор добавляет два кода, один для нового пустого листа на ветви 0, и один - для нового кода ветви 1. Когда дерево заполнится, т.е. когда в нем будут все символы, компрессор просто изменяет последний пустой лист на последний символ. Программа декомпрессии может делать уточнения в своем дереве, потому что у нее имеется в точности то же самое дерево, что и у компрессора. Когда она принимает код пустого листа, она читает следующий код из сжатых данных как литерал ASCII. Затем декомпрессор применяет для изменения дерева ту же программу, какую использовал и компрессор.

Однако пустой лист и инициализированное дерево не решают проблему отслеживания изменения вероятностей. Чтобы делать это, вам необходимо добавить веса к каждому узлу дерева и изменять эти веса во время обработки данных. Вы должны также поддерживать список обозначений узлов (и весов), сортированный по весу.

Каждый символ начинается с веса 1 (пустой лист начинается с 0). Всякий раз, когда компрессор передает символ, который находится в таблице, он инкрементирует вес узла, соот-

ветствующего этому символу. Если это изменение делает узел "тяжелее", чем узлы, которые выше него в списке, компрессор обменивает узел символа с самым тяжелым узлом, который, однако, легче, чем узел символа. При обмене в дело идут только определения родительских узлов и ветвей; это не касается потомков обмениваемых узлов, так что не существует опасности, что лист станет внутренним узлом, а внутренний узел - листом.

Компрессор затем проходит вверх по дереву к родителю символа, который мог быть изменен во время последнего обмена. Он продолжает процесс с родителем, и далее по дереву, но не достигает корня.

Алгоритмы сжатия с потерями.

Алгоритм	Цепочки, за счет которых происходит сжатие
Групповое кодирование	2 2 2 2 2 2 15 15 15 - Подряд идущие цвета
LZW	2 3 15 40 2 3 15 40 - Одинаковые подцепочки
Хаффмана	2 2 3 2 2 4 3 2 2 2 4 - Разная частота появления цвета
JPEG	Отсутствие резких границ
Фрактальный	Подобие между элементами изображения

JPEG - один из самых новых и достаточно мощных алгоритмов. Практически он становится стандартом де-факто для полноцветных изображений. Алгоритм JPEG был разработан группой фирм под названием Joint Photographic Experts Group. Целью проекта являлось создание высокоэффективного стандарта сжатия как черно-белых, так и цветных изображений, эта цель и была достигнута разработчиками. В настоящее время JPEG находит широчайшее применение там, где требуется высокая степень сжатия - например, в Internet.

В отличие от LZW - алгоритма JPEG - кодирование является кодированием с потерями. Сам алгоритм кодирования базируется на очень сложной математике, но в общих чертах его можно описать так:

Прежде всего, программа делит изображение на блоки - матрицы размером 8x8 пикселей. Поскольку при использовании метода JPEG время, затрачиваемое на сжатие изображения, пропорционально квадрату числа пикселей в блоке, обработка нескольких блоков меньшего размера делается значительно быстрее, чем обработка всего изображения целиком.

Схема YUV использует три компоненты, Y – яркость (может быть использована как чёрно – белое изображение), U – голубизна, V – краснота. Перевод из RGB осуществляется по схеме:

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = 0.1687R - 0.3313G + 0.5B$$

$$V = 0.5R - 0.1187G + 0.0813B$$

Это преобразование, в принципе, не имеет потерь, но на практике вводится ошибка округления, обусловленная необходимостью округления результата до целого. Так как глаз человека более чувствителен к первой компоненте, чем к двум другим, то в JPEG допускает дискретизацию различных компонент с различной частотой. Наиболее общий случай – использование одной выборки U и V для четырёх выборок Y . Это позволяет сохранять лишь 50% используемого объёма при практически неизменном качестве изображения. Технология дискретизации, при которой некоторые компоненты оцифровываются с меньшей частотой, чем другие, называется поддискретизацией.

К значениям пикселей применяется формула, названная дискретным косинусоидальным преобразованием (Discrete Cosine Transform - DCT). DCT переводит матрицу значений пикселей 8×8 в матрицу значений амплитуд такой же размерности, соответствующую определенным частотам синусоидальных колебаний. Левый верхний угол матрицы соответствует низким частотам, а правый нижний - высоким.

Дискретное косинусоидальное преобразование (DCT) превращает массив данных интенсивности в массив данных частоты, который содержит информацию о том, как быстро изменяется интенсивность. В JPEG применяется DCT для квадратов 8×8 данных о пикселях для каждого компонента цвета. Точки в каждом блоке нумеруются от $(0,0)$ в верхнем левом до $(7,7)$ в нижнем правом углах. $F(x,y)$ есть значение данных в точке (x,y) . Создаётся новый блок по формуле:

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

$$C(z) = \frac{1}{\sqrt{2}}, \quad z = 0,$$

$$C(z) = 1, \quad z \neq 0.$$

Обратное преобразование имеет вид:

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

По физическому смыслу преобразование сводится к представлению изображения в виде суммы (ко)синусоидальных гармоник (волн). Значения F определяют амплитуды гармоник, u, v – их частоты. $F(u,v)$ указывает на степень изменения величин для каждой из множества частот. Значение $F(0,0)$ указывает что уровень значения не изменился, $F(7,7)$ определяет наиболее быстрое изменение величины в обоих направлениях. Более высокие частоты “отвечают” за передачу более “тонких” деталей изображения.

Для блоков 8*8 выходные данные на 4 бита длиннее входных. Отличие интенсивностей от частот в том, что медленные изменения гораздо заметнее, чем быстрые, так что данные низкой частоты более важны, чем высокой для восстановления изображения.

Коэффициент качества, введенный пользователем, используется в простой формуле, которая генерирует значения элементов другой матрицы 8x8, названной матрицей квантования. Чем ниже коэффициент качества, тем большие значения будут иметь элементы матрицы.

Каждое значение в матрице, получившееся после DCT - преобразования, делится на соответствующее значение из матрицы квантования, затем округляется до ближайшего целого числа. Так как большие числа находятся в правой нижней половине матрицы квантования, то основная часть высокочастотной информации изображения будет отброшена. Поэтому нижняя правая часть матрицы пикселей будет состоять в основном из нулей. Конкретная матрица квантования задаётся кодером, например, для сжатия видео часто применяется матрица

$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \rightarrow u \quad \downarrow v$$

Далее программа, двигаясь по матрице зигзагообразно (13456 и т.д.), считывает элементы матрицы и кодирует их последовательно методами без потерь.

F(00)	F(01)	F(02)	F(03)				
1	2	6					
F(10)	F(11)	F(12)					
3	5						
F(21)	F(22)	F(23)					
4							
F(31)	F(32)	F(33)					

Заметим, что сжатие существенно зависит от нулей в правой нижней половине матрицы. Чем ниже коэффициент качества, тем больше нулей в матрице и, следовательно, тем выше степень сжатия.

Декодирование JPEG-изображения начинается с шага обратного кодированию без потерь, в результате чего восстанавливается матрица квантования пикселей.

Значения из матрицы пикселей умножаются на значения из матрицы квантования, чтобы восстановить, насколько это возможно, матрицу, которая была вычислена на шаге применения DCT. На этапе квантования была потеряна некоторая часть информации, поэтому числа в матрице будут близки к первоначальным, но не будет абсолютного совпадения.

Обратная к DCT формула (IDCT) применяется к матрице для восстановления значений пикселей исходного изображения.

Еще раз отметим, что полученные цвета не будут полностью соответствовать первоначальным из - за потери информации на шаге квантования. Восстановленное изображение, при сравнении с оригиналом, будет выглядеть несколько размытым и обесцвеченным.

Итак, теперь мы можем сделать вывод о достоинствах и недостатках формата JPEG. Безусловно, его сильной стороной является большой коэффициент сжатия при сохранении исходной цветовой глубины. Именно это свойство обусловило его широкое применение в Internet, где уменьшение размера файлов имеет первостепенное значение, в мультимедийных энциклопедиях, где требуется хранение возможно большего количества графики в ограниченном объеме.

Отрицательным свойством этого формата является неустранимое никакими средствами, внутренне ему присущее ухудшение качества изображения. Именно этот печальный факт не позволяет применять его в полиграфии, где качество ставится во главу угла.

Однако формат JPEG не является пределом совершенства в стремлении уменьшить размер конечного файла. В последнее время ведутся интенсивные исследования в области так называемого вейвлет - преобразования (или всплеск - преобразования). Основанные на сложнейших математических принципах вейвлет - кодеры позволяют получить большее сжатие, чем JPEG, при меньших потерях информации. Несмотря на сложность математики вейвлет - преобразования, в программной реализации оно проще, чем JPEG. Хотя алгоритмы вейвлет - сжатия пока находятся в начальной стадии развития, им уготовано большое будущее.

Фрактальное сжатие.

Эта группа алгоритмов, по-видимому, является самой перспективной и развивается сейчас наиболее бурно. Первые практические результаты были получены сравнительно недавно - в 1992 году - и произвели ошеломляющее впечатление. В декабре 1992 года компания Microsoft выпустила свой новый компакт - диск Microsoft Encarta. С тех пор эта мультимедиа - энциклопедия, содержащая информацию о животных, цветах, деревьях и живописных местах, не покидает списки наиболее популярных энциклопедий на компакт - дисках. Коэффициент сжатия у фрактальных алгоритмов варьируется в пределах 2-2000. Причем большие коэффициенты достигаются на реальных изображениях, что, вообще говоря, нетипично для предшествующих алгоритмов. Кроме того, при разархивации изображение можно масштабировать. Уникальная особенность этого алгоритма заключается в том, что увеличенное изображение не дробится на квадраты. Во фрактальном сжатии используется принципиально новая идея - не близость цветов в локальной области, а подобие разных по размеру областей изображения. Это, безусловно, наиболее прогрессивный подход на сегодняшний день. Алгоритм ориентирован на полноцветные изображения и изображения в градациях серого цвета.

Его особенностью является потребность в колоссальных вычислительных мощностях при архивации. При этом распаковка требует меньше вычислений, чем у JPEG. Причем, если у предыдущих алгоритмов коэффициент симметричности

(отношение времени архивации ко времени разархивации) не превышал 3, то у фрактального алгоритма он колеблется в пределах 1000-10000. Как следствие - основные работы сейчас ведутся по распараллеливанию и ускорению его работы. Фрактальное сжатие реализовано в формате GIF.

Идея фрактальная архивация основана на том, что с помощью коэффициентов системы функций изображение представляется в более компактной форме. Прежде чем рассматривать процесс архивации, разберем, как IFS (система итерируемых функций) строит изображение.

Строго говоря, IFS - это набор трехмерных аффинных преобразований, переводящих одно изображение в другое. Преобразованию подвергаются точки в трехмерном пространстве (x координата, y координата, яркость).

Наиболее наглядно этот процесс продемонстрировал Барнсли в своей книге "Фрактальное сжатие изображения". В ней введено понятие Фотокопировальной Машины, состоящей из экрана, на котором изображена исходная картинка, и системы линз, проецирующих изображение на другой экран. Каждая линза проецирует часть исходного изображения. Расставляя линзы и меняя их характеристики, можно управлять получаемым изображением. Линзы должны уменьшать в размерах проектируемую часть изображения. Кроме того, они могут менять яркость фрагмента и проецируют не круги, а области с произвольной границей.

Один шаг Машины состоит в построении с помощью проецирования по исходному изображению нового. Утверждается, что на некотором шаге изображение перестанет изменяться. Оно будет зависеть только от расположения и характеристик линз, и не будет зависеть от исходной картинки. Это изображение называется неподвижной точкой. Поскольку отображение линз является сжимающим, каждая линза в явном виде задает самоподобные области в нашем изображении. Благодаря самоподобию мы получаем сложную структуру изображения при любом увеличении.

Наиболее известны два изображения, полученных с помощью IFS треугольник Серпинского и папоротник Барнсли. Первое задается тремя, а второе - пятью аффинными преобразованиями (или, в нашей терминологии, линзами). Каждое преобразование задается буквально считанными байтами, в то время, как изображение, построенное с их помощью, может занимать и несколько мегабайт.

Становится понятно, как работает архиватор, и почему ему требуется так много времени. Фактически, фрактальная компрессия - это поиск самоподобных областей в изображении и определение для них параметров аффинных преобразований.

В худшем случае, если не будет применяться оптимизирующий алгоритм, потребуется перебор и сравнение всех возможных фрагментов изображения разного размера. Даже для небольших изображений при учете дискретности мы получим астрономическое число перебираемых вариантов. Даже резкое сужение классов преобразований, например, за счет масштабирования только в определенное число раз, не позволит добиться приемлемого времени. Кроме того, при этом теряется качество изображения. Подавляющее большинство исследований в области фрактальной компрессии сейчас направлены на уменьшение времени архивации, необходимого для получения качественного изображения.

Для фрактального алгоритма компрессии, как и для других алгоритмов сжатия с потерями, очень важны механизмы, с помощью которых можно будет регулировать степень сжатия и степень потерь. К настоящему времени разработан достаточно большой набор таких методов. Во - первых, можно ограничить количество преобразований, заведомо обеспечив степень сжатия не ниже фиксированной величины. Во - вторых, можно потребовать, чтобы в ситуации, когда разница между обрабатываемым фрагментом и наилучшим его приближением будет выше определенного порогового значения, этот фрагмент дробился обязательно (для него обязательно заводится несколько линз). В-третьих, можно запретить дробить фрагменты размером меньше, допустим, четырех точек. Изменяя пороговые значения и приоритет этих условий, можно очень гибко управлять коэффициентом компрессии изображения: от побитного соответствия, до любой степени сжатия.

Сравнение с JPEG .

Сегодня наиболее распространенным алгоритмом архивации графики является JPEG. Сравним его с фрактальной компрессией.

Во-первых, заметим, что и тот, и другой алгоритм оперируют 8-битными (в градациях серого) и 24-битными полноцветными изображениями. Оба являются алгоритмами сжатия с потерями и обеспечивают близкие коэффициенты архивации. И у фрактального алгоритма, и у JPEG существует возможность увеличить степень сжатия за счет увеличения потерь. Кроме того, оба алгоритма очень хорошо распараллеливаются.

Различия начинаются, если мы рассмотрим время, необходимое алгоритмам для архивации/разархивации. Так, фрактальный алгоритм сжимает в сотни и даже в тысячи раз дольше, чем JPEG. Распаковка изображения, наоборот, произойдет в 5-10 раз быстрее. Поэтому, если изображение будет сжато только один раз, а передано по сети и распаковано множество раз, то выгодней использовать фрактальный алгоритм.

JPEG использует разложение изображения по косинусоидальным функциям, поэтому потери в нем (даже при заданных минимальных потерях) проявляются в волнах и ореолах на границе резких переходов цветов. Именно за этот эффект его не любят использовать при сжатии изображений, которые готовят для качественной печати: там этот эффект может стать очень заметен.

Фрактальный алгоритм избавлен от этого недостатка. Более того, при печати изображения каждый раз приходится выполнять операцию масштабирования, поскольку растр (или линия) печатающего устройства не совпадает с растром изображения. При преобразовании также может возникнуть несколько неприятных эффектов, с которыми можно бороться либо масштабируя изображение программно (для дешевых устройств печати типа обычных лазерных и струйных принтеров), либо снабжая устройство печати своим процессором, винчестером и набором программ обработки изображений (для дорогих фотонаборных автоматов). Как можно догадаться, при использовании фрактального алгоритма таких проблем практически не возникает.

Улучшение визуального качества изображений путем поэлементного преобразования.

В большом числе информационных систем применяется представление результатов обработки данных в виде изображения, выводимого на экран для использования наблюдателем. Процедуру, обеспечивающую такое представление, называют визуализацией. Желательно при помощи обработки придать выводимому изображению такие качества, благодаря которым его восприятие человеком было бы по возможности комфортным. Часто бывает полез-

ным подчеркнуть, усилить какие-то черты, особенности, нюансы наблюдаемой картины с целью улучшения ее субъективного восприятия.

Последнее - субъективность восприятия - сильно усложняет применение формализованного подхода в достижении данных целей. Поэтому при обработке изображений для визуализации получили распространение методы, в которых часто отсутствуют строгие математические критерии оптимальности. Их заменяют качественные представления о целесообразности той или иной обработки, опирающиеся на субъективные оценки результатов.

Подавляющее большинство процедур обработки для получения результата в каждой точке кадра привлекает входные данные из некоторого множества точек исходного изображения, окружающих обрабатываемую точку. Однако имеется группа процедур, где осуществляется так называемая поэлементная обработка. Здесь результат обработки в любой точке кадра зависит только от значения входного изображения в этой же точке. Очевидным достоинством таких процедур является их предельная простота. Вместе с тем, многие из них приводят к очевидному субъективному улучшению визуального качества. Этим определяется внимание, которое уделяется поэлементным процедурам. Не преувеличивая их роли, отметим, что очень часто поэлементная обработка применяется как заключительный этап при решении более сложной задачи обработки изображения.

Сущность поэлементной обработки изображений сводится к следующему. Пусть $x(i, j) = x_{ij}$, $y(i, j) = y_{ij}$ - значения яркости исходного и получаемого после обработки изображений соответственно в точке кадра, имеющей декартовы координаты i (номер строки) и j (номер столбца).

Поэлементная обработка означает, что существует функциональная однозначная зависимость между этими яркостями $y_{i,j} = f_{i,j}(x_{i,j})$, позволяющая по значению исходного сигнала определить значение выходного продукта. В общем случае, как это учтено в данном выражении, вид или параметры функции f_{ij} , описывающей обработку, зависят от текущих координат. При этом обработка является неоднородной. Однако в большинстве практически применяемых процедур используется однородная поэлементная обработка. В этом случае индексы могут отсутствовать.

Линейное контрастирование изображения.

Задача контрастирования связана с улучшением согласования динамического диапазона изображения и экрана, на котором выполняется визуализация. Если для цифрового представления каждого отсчета изображения отводится 1 байт (8 бит) запоминающего устройства, то входной или выходной сигналы могут принимать одно из 256 значений. Обычно в качестве рабочего используется диапазон 0...255; при этом значение 0 соответствует при визуализации уровню черного, а значение 255 - уровню белого. Предположим, что минимальная и макси-

мальная яркости исходного изображения равны x_{\min} и x_{\max} соответственно. Если эти параметры или один из них существенно отличаются от граничных значений яркостного диапазона, то визуализированная картина выглядит как ненасыщенная, неудобная, утомляющая при наблюдении. При линейном контрастировании используется линейное поэлементное преобразование вида:

$$y = ax + b,$$

$$y_{\min} = ax_{\min} + b$$

$$y_{\max} = ax_{\max} + b$$

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}(y_{\max} - y_{\min}) + y_{\min}$$

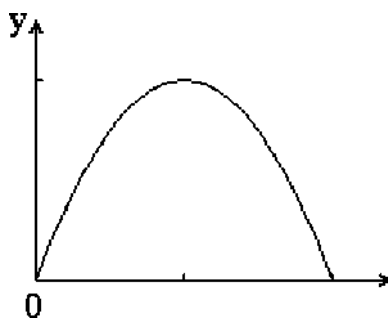


Рис. 18 .Линейное контрастирование изображения.

Соляризация изображения.

При данном виде обработки преобразование позволяет управлять динамическим диапазоном преобразованного изображения. Функция, описывающая данное преобразование, является квадратичной параболой, ее график приведен на рисунке.

$$y = kx(x_{\max} - x)$$



Рмс. 19. Соляризация.

Смысл соляризации заключается в том, что участки исходного изображения, имеющие уровень белого или близкий к нему уровень яркости, после обработки имеют уровень черного. При этом сохраняют уровень черного и участки, имеющие его на исходном изображении. Уровень же белого на выходе приобретают участки, имеющие на входе средний уровень яркости (уровень серого).



Рис. 20. Соляризация изображения.

Препарирование изображения.

Препарирование представляет собой целый класс поэлементных преобразований изображений. Характеристики применяемых на практике процедур препарирования приведены на рис.

Преобразование с пороговой характеристикой (рис.а) превращает полутоновое изображение, содержащее все уровни яркости, в бинарное, точки которого имеют яркости 0 или $u=umax$. Такая операция, называемая иногда бинаризацией или бинарным квантованием, может быть полезной, когда для наблюдателя важны очертания объектов, присутствующих на изображении,



Рис. 21. Бинаризация изображения.

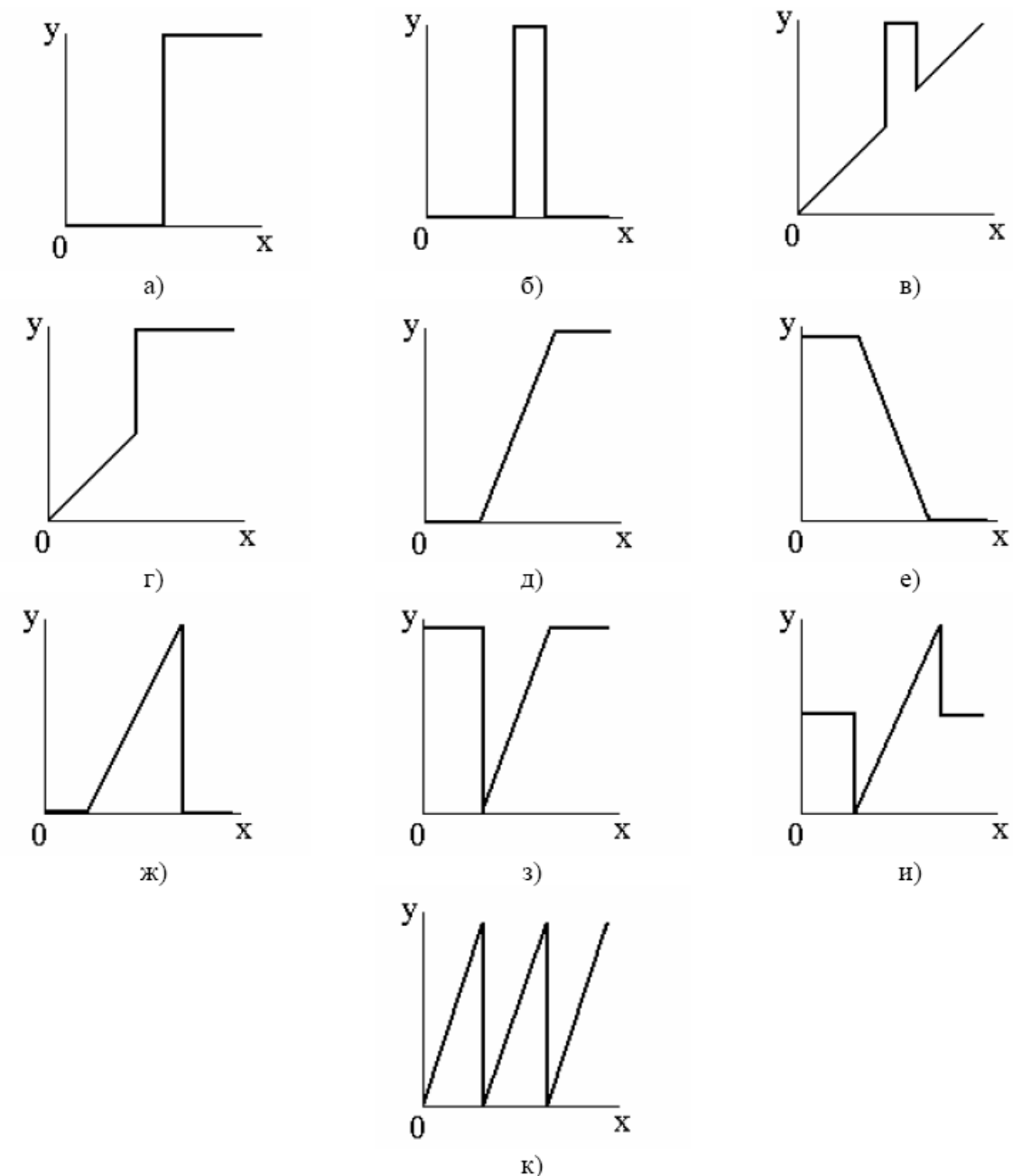


Рис. 22. Примеры функций для препарирования.

Фильтры (Plug - ins) и спецэффекты (Effects).

Большинство фильтров (filters или plug - ins) предназначено для создания специальных эффектов, например имитации мозаики или какого – либо живописного стиля. С помощью трехмерных спецэффектов двухмерные графические программы способны трансформировать плоское двухмерное изображение в объемное. Группа художественных эффектов позволяет превратить обычную фотографию в произведение живописи. При этом вы можете имитировать самые разные виды живописи (масло, акварель и т. п.) и стили любых художников.

Фильтры и спецэффекты представляют собой небольшие программы, выполняющие заранее установленную последовательность команд. Они автоматически вычисляют значения и

характеристики каждого пикселя изображения и затем модифицируют их в соответствии с новыми значениями. Например, при применении к изображению фильтра Размывка движением (Motion Blur) идет анализ значений всех входящих в изображение пикселей и сдвиг этих значений в определенном направлении для создания иллюзии движения.

Большинство современных графических программ поддерживает возможное применения фильтров, разработанных третьими фирмами. Эти модули называются подключаемыми (Plug - Ins). Их использование расширяет функциональные возможности программы.

Получение специальных эффектов не представляет особого труда. Секрет каждого из них кроется в матрице, которую называют ядром свертки. Матрица размером 3x3 содержит три строки по три числа в каждой.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Для преобразования каждого пикселя изображения необходимо выполнить следующие действия:

Шаг 1. Значение цвета пикселя умножается на число в центре ядра.

Шаг 2. На следующем шаге выполняется умножение восьми значений цветов пикселей, окружающих центральный пиксель, на соответствующие им коэффициенты ядра с последующим суммированием всех девяти значений. В результате получается новое значение цвета преобразуемого пикселя.

Шаг 3. Для каждого пикселя изображения повторяется процесс, включающий выполнение шагов 1 и 2. Данную процедуру принято называть фильтрацией изображения.

Алгоритм работы фильтра Размывание.

Ядро размывания размером 3*3 применяется к компонентам цвета каждого пикселя изображения. Значение цвета пикселя (который, собственно, находится в центре ядра) вычисляется умножением соответствующего весового коэффициента на значение цвета в изображении с последующим суммированием. Изображение получается размытым по сравнению с исходным, так как цвет каждого пикселя выравнивается (усредняется) благодаря влиянию соседей.

Степень размывания можно увеличить одним из трех способов:

- использованием большего размера ядра для распределения цвета среди большего числа соседей;
- подбором коэффициентов ядра и уменьшением влияния центрального коэффициента;
- повторной фильтрацией изображения с тем же ядром размывания.

Алгоритм работы фильтра увеличение резкости.

При обработке каждого пикселя в изображении также используется ядро резкости размером 3×3 . Красная, зелёная и синяя составляющая обрабатываются отдельно и позже объединяются. Отрицательные веса вокруг центра ядра увеличивают контраст между центральным пикселем и соседними.

Алгоритм работы фильтра Акварелизация.

На первом шаге производится сглаживание цветов. Часто используется медианное сглаживание, при котором значение цвета пикселя и его 24 соседей сортируется по возрастанию. Центральному пикселю присваивается тринадцатое значение. На втором шаге производится обработка пикселей ядром резкости.

Алгоритм работы эффекта Тиснение.

В изображении, к которому применен эффект Тиснение, контуры кажутся выдавленными над поверхностью. Процесс производится над обычным цветным изображением.

Каждый пиксель в изображении обрабатывается ядром тиснения размером 3×3 . Сумма весов в ядре тиснения равна 0. Это означает, что фоновым пикселям (пикселям, которые не находятся на границах перехода от одного цвета к другому) присваиваются нулевые значения, а не фоновым пикселям значения отличные от нуля. После того как пиксель обработан матрицей, к нему прибавляется значение 128.



a)



б)



в)



г)

Рис. 23. Исходное изображение (а), акварелизация (б), размывание (в), тиснение (г).

Фильтрация. Идеология фильтрации основывается на рациональном использовании данных как из рабочей точки, так и из её окрестности. В этом заключается её отличие от рассмотренных выше процедур поэлементного преобразования. Алгоритмы фильтрации подразделяются в зависимости от области, используемой в расчётах. Однако, при обработке растровых изображений на компьютере, обычно, известны все точки изображения, поэтому данная классификация опускается. Пусть x_{ij} – значение яркости изображения (полезного сигнала) на пересечении i -й строки и j -го столбца. Тогда наблюдаемое на входе фильтра изображение описывается моделью

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \quad i = \overline{0, I-1}, j = \overline{0, J-1}.$$

Здесь $n_{i,j}$ - значение помехи.

При линейной фильтрации выходной сигнал определяется линейной комбинацией входных данных.

$$x^*(i, j) = \sum_{(i_1, j_1) \in S} a(i_1, j_1) y(i - i_1, j - j_1).$$

В этом выражении $x^*(i, j) = x_{i,j}^*$ - результат фильтрации полезного сигнала $x(i, j) = x_{i,j}$,

S – множество координат точек, $a(i_1, j_1)$ - весовые коэффициенты, совокупность которых представляет собой двумерную импульсную характеристику. Здесь импульсная характеристика не зависит от координат точки, в которой определяется выходной эффект. Такие процедуры называются однородными.

Наиболее распространённым критерием оптимальности, применяемым для оценки качества изображения, является критерий минимума среднего квадрата ошибок.

$$E \left\{ \left[x(i, j) - \sum_{(i_1, j_1) \in S} a(i_1, j_1) y(i - i_1, j - j_1) \right]^2 \right\} = \min_{a(i)}.$$

Здесь $E\{\cdot\}$ - символ математического ожидания. Математическое ожидание вычисляется по всем случайным величинам, содержащимся в выражении. Оптимизационную задачу сводят к решению системы уравнений. Для этого вычисляют производную от левой части выражения по коэффициенту $a(k, l) = a_{k,l}$ и приравнивают её 0. Операции дифференцирования суммирования и математического ожидания линейны и поэтому перестановочны

$$E \{ x_{i,j} \cdot y_{i-k, j-l} \} = \sum_{i_1, j_1} a_{i_1, j_1} \cdot E \{ y_{i-i_1, j-j_1} \cdot y_{i-k, j-l} \}$$

Математические ожидания являются отсчётами корреляционных функций:

$$B_{xy}(k, l) = E \{ x_{i,j} \cdot y_{i-k, j-l} \}, \quad B_y(k - i_1, l - j_1) = E \{ y_{i-i_1, j-j_1} \cdot y_{i-k, j-l} \}.$$

Тогда:

$$B_{xy}(k, l) = \sum_{i_1, j_1 \in S} a_{i_1, j_1} \cdot B_y(k - i_1, l - j_1).$$

Корреляционная функция $B_y\{\cdot\}$ и взаимно корреляционная функция $B_{xy}\{\cdot\}$ считаются известными. Приведённое уравнение является линейным относительно коэффициентов a_{i_1, j_1} . Число неизвестных равно числу точек n_s в окрестности S . Если повторить дифференцирование по остальным $n_s - 1$ неизвестным, то получим ещё $n_s - 1$ уравнений. В результате имеем n_s алгебраических линейных уравнений с n_s неизвестными. В теории фильтрации они называются уравнением Винера – Хопфа.

$$B_{xy}(k, l) = \sum_{i_1, j_1 \in S} a_{i_1, j_1} \cdot B_y(k - i_1, l - j_1)$$

$$k, l \in S$$

Решив его, получим значения коэффициентов, определяющих импульсную характеристику линейного фильтра.

Часто при обработке изображений стремятся сохранить среднюю яркость изображения. Достигается это при

$$\sum_{i_1, j_1 \in S} a(i_1, j_1) = 1.$$

Анимация.

Анимация оживляет и привлекает внимание к изображению, однако, как и звук, анимацию следует использовать в меру, не загромождая остальные элементы проекта. Инструментальные системы мультимедиа часто предоставляют средства для разработки анимации. Эти системы имеют возможность для воспроизведения анимационных файлов. Одной из самых распространённых программ такого типа является Director Macromedia. Создание анимации на компьютере аналогично созданию анимации на прозрачной плёнке.

Техника анимации использует последовательность графических изображений, которые воспроизводятся с частотой 24 кадра в секунду. Создание анимации на прозрачной плёнке начинается с ключевых кадров, первого и последнего кадра движения. Когда фигура человека перемещается по экрану, он переносит вес тела сна одну ногу, а затем на другую как бы подставляя её при падении. На первом ключевом кадре может быть изображён человек, переносащий вес с левой ноги вперёд, ноги находятся на одной линии и кажется, что человек падает вперёд. На последнем ключевом кадре правая нога находится впереди и центр тяжести располагается впереди левой и позади правой ноги. Серия кадров между ключевыми создаётся в процессе, который называют попарным объединением кадров. Здесь рассчитывается число промежуточных кадров и траектория каждой части объекта. После этого легко сделать эскизы объекта на прозрачной плёнке. После просмотра нарисованных карандашом объектов они заполняются красками.

Создание анимации для Интернета.

Возможность создания анимированной графики, предоставляемая форматом GIF, позволяет разработчикам Web - сайтов оживить страницы и привлечь к ним внимание. Рассмотрим средства, которые предлагают производители графического программного обеспечения для создания анимации.

Одним из них служит пакет Paint Shop Pro 5 фирмы Jasc Software. Это не только графический редактор, предоставляющий широкий выбор кистей для рисования, средства ретуширования изображения, более 25 стандартных фильтров для его обработки, базовый набор стандартных эффектов и возможность подключения фильтров пакета Photoshop. В его состав также включен Animation Shop - программа для создания анимационных GIF-файлов, которые можно использовать в Интернете или в собственных мультимедиа - приложениях. В ней к одному или нескольким статическим изображениям можно применить различные эффекты и переходы для создания мультипликации. Встроенные мастера позволяют быстро создать анимацию, подобрать цвета и сохранить файл. Анимация сохраняется в GIF - файле (.gif) или собственном формате программы Animation files (.mng).

На предварительном этапе в графическом редакторе Paint Shop Pro создают изображения заданного размера в соответствии с числом объектов будущей анимации. Как правило, размер выбирается небольшим, так как от этого зависит скорость загрузки Web - страницы с анимацией. Количество кадров изображения также должно быть невелико и обычно не превышает трех - четырех. Объект, который будет анимирован, должен быть нарисован или уже сохранен в отдельном файле. Если требуется изобразить несколько фаз движения, то каждая из них должна сохраняться в отдельном файле. При наложении объектов друг на друга задается прозрачный фон изображения.

Звук.

Звук является одним из самых выразительных средств мультимедиа. Умение использовать звук определяет уровень разработки проекта. Это и смысловая речь, и музыка, и специальные звуковые эффекты.

Звуковая волна – это процесс распространения в атмосфере объёмных деформаций сжатия – разряжения.

Изучением звука занимается акустика. Звуковые волны распространяются со скоростью 330 метров в секунду (1 Мах) над уровнем моря. Уровень звукового давления измеряется в децибелах (дБ). Децибел – это логарифм отношения интенсивности (или звукового давления) данного звука к интенсивности звука на пороге слышимости:

$$L = 10 \lg \frac{I}{I_0}, \quad \text{дБ}, \quad L = 10 \lg \frac{P}{P_0}$$

Увеличение мощности в 100 раз приводит к увеличению звука на 20 децибел. Интенсивность показывает, какая энергия переносится звуковой волной через единицу площади.

дБ	Вт	Пример
170	100000	Турбореактивный двигатель
120	1	Молот
90	0.001	Электричка в метро
70	0.00001	Беседа
30	0.000000001	Шёпот

Исследования показали, что шум на уровне 45 дБ не беспокоит соседей, 45 - 55 дБ вызывает редкие жалобы, 50 - 60 дБ – частые жалобы соседей, 55 - 65 дБ – вызывают угрозы, а при более 65 дБ начинаются активные действия.

Частота колебаний измеряется в Герцах (Гц). Человек воспринимает звук в интервале 16 – 20000 Гц.

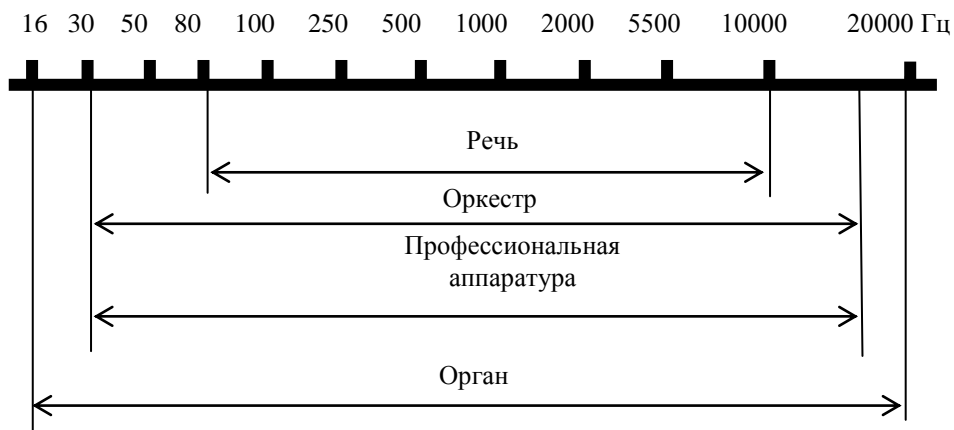


Рис. 24. Диапазон воспринимаемых частот.

MIDI

Musical Instrument Digital Interface - цифровой интерфейс музыкальных инструментов. Создан в 1982 году ведущими производителями электронных музыкальных инструментов - Yamaha, Roland, Korg, E-mu и др. Изначально был предназначен для замены принятого в то время управления музыкальными инструментами при помощи аналоговых сигналов управлением при помощи информационных сообщений, передаваемых по цифровому интерфейсу. Впоследствии стал стандартом де-факто в области электронных музыкальных инструментов и компьютерных модулей синтеза. MIDI представляет собой так называемый событийно-ориентированный протокол связи между инструментами. Всякий раз, когда исполнитель производит какое-либо воздействие на органы управления (нажатие/отпускание клавиш, педалей, изменение положений регуляторов и т.п.), инструмент формирует соответствующее MIDI-сообщение, в тот же момент посылаемое по интерфейсу. Другие инструменты, получая сообщения, обрабатывают их так же, как и при воздействии на их собственные органы управления. Таким образом, поток MIDI-сообщений представляет собой как бы слепок с действий исполнителя, сохраняя присущий ему стиль исполнения - динамику, технические приемы и т.п. При записи на устройства хранения информации MIDI-сообщения снабжаются временными метками, образуя своеобразный способ представления партитуры. При воспроизведении по этим меткам полностью и однозначно восстанавливается исходный MIDI-поток.

Спецификация MIDI состоит из аппаратной спецификации самого интерфейса и спецификации формата данных - описания системы передаваемых сообщений. Соответственно, различается аппаратный MIDI-интерфейс и формат MIDI-данных (так называемая MIDI-партитура); интерфейс используется для физического соединения источника и приемника сообщений, формат данных - для создания, хранения и передачи MIDI-сообщений. В настоящее время эти понятия стали самостоятельными и обычно используются отдельно друг от друга - по MIDI-интерфейсу могут передаваться данные любого другого формата, а MIDI-формат может использоваться только для обработки партитур, без вывода на устройство синтеза.

Цифровой звук.

Акустический звук представляет собой непрерывный во времени и по амплитуде процесс, то есть давление воздуха изменяется во времени плавно, а не перепрыгивает от одного зна-

чения к другому. Акустический звук может быть преобразован в электрический сигнал при помощи микрофона, который в зависимости от изменения давления воздуха изменяет создаваемое им на выходе электрическое напряжение. После перевода акустического звука в электрический сигнал непрерывность во времени и по амплитуде сохраняется: напряжение сигнала изменяется аналогично изменению давления воздуха, вот почему данный звук называют аналоговым. Мы можем записать электрический сигнал на магнитную ленту и превратить его вновь в звук при помощи динамика, который работает как "микрофон наоборот": перемещает воздух в соответствии с изменениями напряжения. Соответственно, сохраняется и упомянутая непрерывность сигнала.

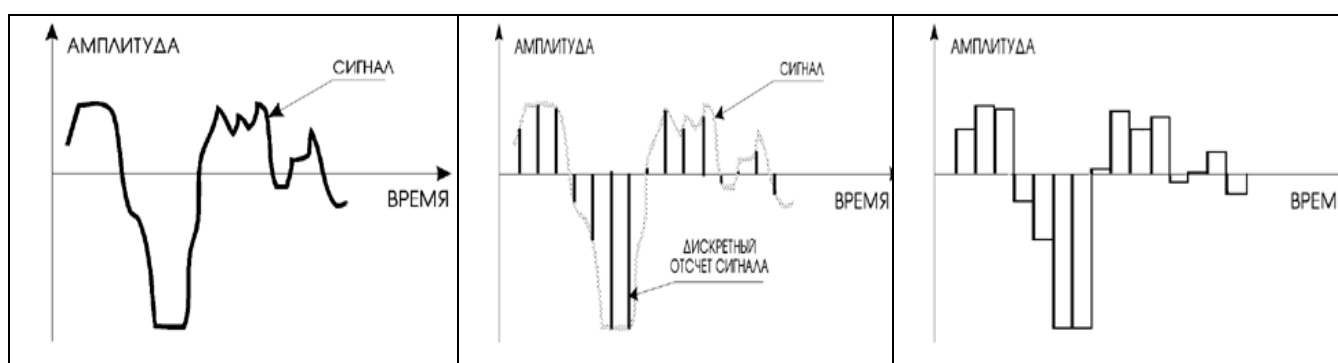


Рис. 25. Аналоговый сигнал, процесс оцифровки, сигнал подготовленный для воспроизведения (после ЦАП).

Частота дискретизации

Преобразование аналогового сигнала в цифровой состоит из двух этапов: дискретизации по времени и квантования по амплитуде. Дискретизация по времени означает, что сигнал представляется рядом своих отсчетов, взятых через равные промежутки времени. Например, когда мы говорим, что частота дискретизации 44,1 кГц, то это значит, что сигнал измеряется 44100 раз в течении секунды.

Основной вопрос на первом этапе преобразования аналогового сигнала в цифровой (оцифровки) состоит в выборе частоты дискретизации аналогового сигнала. Как уже было сказано, чем больше частота - тем точнее соответствует цифровой сигнал аналоговому. Однако, пропорционально увеличению частоты возрастают:

- а) интенсивность потока цифровых данных, а пропускные возможности интерфейсов не безграничны, особенно если записывается/воспроизводится одновременно несколько каналов;
- б) вычислительная нагрузка на цифровые процессоры эффектов, а их вычислительные возможности также ограничены;
- в) объем памяти, необходимой для хранения цифрового сигнала.

Очевидно, что необходим компромисс. От выбора частоты дискретизации зависит частотный диапазон полученного цифрового звука или максимальная частота аналогового сигнала, правильно представленная в цифровом. Считается, что диапазон частот, которые слышит человек, составляет от 20 до 20000 Гц. Согласно известной теореме Котельникова, для того, чтобы аналоговый (непрерывный по времени) сигнал можно было точно восстановить по его отсчетам, частота дискретизации должна быть как минимум вдвое больше максимальной звуковой частоты. Таким образом, если реальный аналоговый сигнал, который мы собираемся преобразовать в цифровую форму, содержит частотные компоненты от 0 Гц до 20 кГц, то частота дискретизации такого сигнала должна быть не меньше, чем 40 кГц. Сегодня самыми распространенными частотами дискретизации являются 44,1 кГц и 48 кГц. Впрочем, в последнее время идет немало разговоров о том, что обертоны, расположенные свыше 20 кГц, вносят немалый вклад в звучание и в результате появляются преобразователи, использующие частоты дискретизации 96 кГц и 192 кГц.

Второй этап - это квантование амплитуды дискретных отсчетов, полученных на первом этапе.

Процесс квантования амплитуд отсчетов фактически заключается в измерении их величин по отношению к некоторому опорному источнику напряжения, обычно имеющемуся внутри корпуса микросхемы АЦП, и выражении этих величин в виде чисел, состоящих из конечного числа бит. Причем числа могут быть не только целые, например 16-, 18-, 20-, 24- битные, но и 24- или 32- битные с плавающей запятой или с другой кодировкой (например, в кодах с исправлением ошибок), зависящей от конкретной реализации устройства АЦП. Довольно часто используется все же кодирование результатов измерения амплитуд отсчетов в виде целых чисел.

Методы шумоочистки

Все без исключения методы очистки речи от шумов в той или иной степени разработаны с учетом либо особенностей образования речевого сигнала, либо параметров удаляемых помех, либо и того и другого вместе. Немаловажную роль при оценке эффективности методов играет критерий качества очистки, роль которого, в общем случае, играет разборчивость (то есть понимание слушателем того, что произносится). Однако часто для оценки качества очистки используются и формальные параметры, например, соотношение сигнал/помеха, комфортность прослушивания и т.д.

Для описания речевого сигнала, как правило, используется следующая модель. Речь образуется путем возбуждения акустического резонатора (речевого тракта) импульсами воздуха,

проходящими через голосовые связки для звонких и турбулентными потоками воздуха для глухих звуков. С математической точки зрения такая модель может быть описана в виде линейной системы, возбуждаемой периодической функцией для звонких звуков или широкополосным шумом для глухих. С учетом физиологических особенностей образования речи математически речевой сигнал можно представить в виде отклика линейной системы с медленно меняющимися параметрами. На коротком интервале времени ее передаточная функция характеризуется набором резонансных частот, называемых формантами. Для звонких звуков на таком интервале времени речевой сигнал имеет гармоническую структуру, (период сигнала в этом случае называется периодом основного тона), для глухих - шумоподобную. Подобная модель, хоть и достаточно упрощенно описывает реальные процедуры речеобразования, с успехом применяется в различных системах коррекции.

Более сложен вопрос описания модели шумов. При учете всех возможных типов реальных шумов математический аппарат описания процедур коррекции оказывается достаточно громоздким. По этой причине большинство методов шумоочистки разрабатываются в предположении, что шумы носят только аддитивный характер и никак не связаны с исходным речевым сигналом, то есть между шумами и речью отсутствует корреляция. Другими словами, подлежащий коррекции сигнал представляется в виде суммы двух компонентов: речевого сигнала и шума.

Важную роль играют особенности восприятия речи человеком. Например, известно, что согласные звуки хотя и маломощны, но весьма существенны для понимания речи. Неравноценный вклад в обеспечение разборчивости речи вносят и форманты. Первая форманта, обычно лежащая в диапазоне от 250 до 800 Гц, менее существенна для восприятия, чем вторая и последующие. Хорошее воспроизведение кратковременного амплитудного спектра речевого сигнала является важным фактором, тогда как фазовый спектр имеет существенно меньшее значение. Важная особенность слухового восприятия: маскировка речи. Так, например, после громких звуков тихие некоторое время не воспринимаются; рядом с сильными спектральными пиками на слух не различимы более слабые частотные компоненты и т.д.

Теперь, разобравшись в самых общих теоретических вопросах образования речи и особенностей ее восприятия, рассмотрим конкретные методы шумоочистки, часто применяемые на практике.

Простые методы.

Простейшим способом повышения разборчивости речи является ее полосовая фильтрация. Фильтр верхних частот удаляет из сигнала часто встречающиеся низкочастотные шумы. Уменьшение мощности первой форманты при этом, как мы уже говорили, не ведет к серьезному ухудшению разборчивости речи. Фильтр нижних частот с частотой среза порядка 4 кГц

удаляет высокочастотные шумовые составляющие. Данный метод эффективен для ослабления шумов, которые лежат вне диапазона частот, существенного для восприятия речи, и фактически непригоден, когда спектры полезного сигнала и шума находятся в одной частотной области. В случае когда помеха занимает сравнительно узкий участок или участки спектра полезного сигнала, применяют многополосную фильтрацию (эквалайзинг).

В случае воздействия на речь достаточно широкополосного шума разборчивость речи можно улучшить, дополняя полосовую фильтрацию сигнала операцией нормализации (клиппирования). В основу этого метода положен тот факт, что нормализация увеличивает относительную амплитуду важных для разборчивости, но слабых по мощности согласных звуков, тем самым уменьшая их маскирование более громкими гласными звуками или шумом.

Методы, основанные на вычитании спектров.

Суть данных методов состоит в следующем. При аддитивном шуме обрабатываемый сигнал можно представить в виде следующего выражения: $y(t)=s(t)+n(t)$, где $s(t)$ — исходный речевой сигнал, $n(t)$ — шум. В этом случае спектр зашумленного речевого сигнала также является суммой спектров сигнала и шума. Получить спектр исходного сигнала можно путем вычитания спектра шума из спектра наблюдаемого сигнала.

Следует отметить, что методы, основанные на вычитании спектра, в настоящее время наиболее широко используются на практике. Их можно применять для подавления практически любых (кроме импульсных) помех, находящихся в полосе сигнала.

Методы, основанные на винеровской фильтрации сигнала

В этих методах первоначально по зашумленному речевому сигналу $y(t)$ оценивается передаточная характеристика “оптимального” фильтра. Далее этот фильтр применяется во временной или частотной области. В результате фильтрации находится оценка неискаженного речевого сигнала.

Методы коррекции и сглаживания спектра речевых сигналов

Основная энергия речевого сигнала сосредоточена на определенных частотах, называемых формантами, а энергия шума, вообще говоря, может быть распределена по всему диапазону звуковых частот. Таким образом, используя соответствующий фильтр, реализованный во временной или спектральной области и точно настроенный на формантные частоты, можно уменьшать шум, сохраняя при этом полезный сигнал. Однако поскольку значения формант остаются постоянными только на коротких интервалах времени, то при обработке разных участков речи требуется постоянная адаптивная подстройка фильтра. Это не всегда легко реализуется на практике. Например, такая фильтрация совершенно неприемлема в случае воздействия на сигнал суммы гармонических помех.

Методы сглаживания спектра предназначены для удаления в нем нехарактерных для речи резких перепадов и выравнивания динамического диапазона среднего спектра с целью компенсации амплитудно-частотных искажений сигнала в канале связи или звукозаписи.

Методы адаптивного подавления помех

Группа методов, называемых методами адаптивного подавления (или фильтрации) помех, основана на совместной обработке искаженного сигнала $y(t)=s(t)+n(t)$ и опорного сигнала $r(t)$, некоррелированного с исходным речевым сигналом $s(t)$, но коррелированного с шумом $n(t)$ (либо, наоборот, коррелированного с речью, но некоррелированного с помехой). В процессе адаптивной фильтрации сигнала $r(t)$ формируется оценка коррелированного с $r(t)$ компонента, который вычитается из $y(t)$.

Существуют два типа систем, реализующих принцип адаптивного подавления помех. Различаются они по способу получения опорного сигнала. В первой из них опорный сигнал формируют из зашумленного путем тех или иных преобразований последнего. Иногда такие системы называют одноканальными.

Двухканальные системы основаны на использовании двух, как правило, слабо коррелированных между собой источников получения обрабатываемой смеси речевого сигнала и помех. Например, такой метод может быть реализован в режиме “стереозаписи” при использовании двух разнесенных в пространстве направленных микрофонов, которые по-разному ориентированы на источники полезного сигнала и помехи. Достоинством данного метода является способность в некоторых случаях восстанавливать разборчивость даже самых зашумленных сигналов. Недостаток очевиден — режим “стереозаписи” не всегда может быть реализован на практике.

Методы коррекции на основе моделей речевого сигнала.

Эти методы основаны, как правило, на предположении о линейности передаточной функции голосового тракта в модели образования речевого сигнала. В этом случае сигнал в некоторый момент времени может быть представлен в виде линейной комбинации своих значений в предыдущие моменты. Такие методы называют методами линейного предсказания. При их использовании основной задачей при коррекции является оценка коэффициентов линейного предсказания с последующей реконструкцией исходного речевого сигнала. Однако этим методам присущ серьезный недостаток: в случае обработки сильно зашумленных речевых сигналов не всегда возможно правильно вычислять коэффициенты линейного предсказания. Это, в свою очередь, может еще больше ухудшить разборчивость сигнала на выходе системы линейного предсказания.

Метод “псевдостерео”.

Сущность этого метода - использование двух звуковых сигналов, подаваемых на прослушивание отдельно. В этом качестве обычно используют исходный и обработанный каким-либо другим методом шумоочистки сигнал. Можно использовать и два сигнала, обработанные разными методами. Повышение разборчивости при использовании данного режима воспроизведения является чисто субъективным, “на слух”. Тем не менее, часто оказывается, что в связи с естественной привычкой человека получать звуковую информацию по двум каналам одновременно (то есть слушать обоими ушами) при подобном способе подачи звука суммарный продукт воспроизведения будет казаться более разборчивым, чем каждая из его составляющих в отдельности.

Программные средства шумоочистки.

На отечественном рынке специализированного программного обеспечения по шумоочистке речи представлены самые разнообразные программные средства, способные удовлетворить как подготовленного профессионала, так и новичка, пытающегося разобраться, как можно использовать в только что купленном компьютере комплект “Мультимедиа”.

Самые дорогие системы, используемые на профессиональных студиях звукозаписи, позволяющие не просто удалить из записи щелчки и шумы, но и поднять качество ее звучания до состояния, пригодного для переноса записи на мастер - диски.

В этом классе аппаратуры наиболее известны цифровые звукомонтажные станции “Sonic NoNoise” американской компании “Sonic Solutions”. Система “Sonic NoNoise” представляет собой комплект, состоящий из нескольких программных модулей и плат ввода-вывода звука и обработки сигналов. Стандартное программное обеспечение “NoNoise” включает следующий набор модулей шумоподавления и восстановления фонограмм:

“Broadband Denoising” - используется для подавления в фонограмме широкополосных фоновых шумов;

“Complex Filtering” - удаляет фон сети переменного тока, свист и другие помехи с устойчивым характером спектра;

“Manual Declicking” - предназначен для удаления из фонограмм импульсных помех различного происхождения в режиме диалога с оператором;

“Production Declicking and Decracking” - автоматически распознает и удаляет импульсные помехи.

Подобные системы реставрации фонограмм создаются и в России. В качестве примера можно привести разработанный на фирме “Русский компакт-диск” аппаратно-программный комплекс реставрации и цифрового монтажа звука “КАНОНЪ”. В его состав входят встраиваемая в компьютер плата цифровой обработки сигналов и комплект программных модулей,

обеспечивающих удаление из записи широкополосных шумов и импульсных помех, а также фазовую, временную и тональную коррекцию фонограмм.

К сожалению, всем профессиональным системам, предназначенным для очистки музыкальных аудиозаписей и подготовки компакт - дисков и кассет, присущ один 'недостаток' - их сравнительно высокая стоимость.

Программы шумоочистки речевых сигналов. Отечественные профессиональные продукты этого класса представлены в основном разработками двух фирм - московской НПЦ "Нелк" и Санкт-Петербургской "Центр Речевых Технологий" (ЦРТ). Интересно отметить, что, хотя обе эти компании занимаются разработкой подобных программ сравнительно давно, все их продукты находят свою профессиональную и стоимостную нишу на рынке данных программных средств, что уменьшает их конкуренцию между собой.

Наиболее долгая история у программных продуктов фирмы ЦРТ. Первая версия их программы обработки речи, известной сегодня как "Speech Interactive Software" (SIS), увидела свет в 1992 году. С тех пор эта программа неоднократно совершенствовалась, и последняя ее версия имеет обозначение "SIS 5.2".

В состав программы "SIS 5.2" включены следующие основные процедуры шумоочистки речи: нормализация; многополосная фильтрация; адаптивная одно- и двухканальная фильтрация; удаление импульсных помех; удаление помех методом спектрального вычитания; выравнивание спектральных характеристик; темпокоррекция воспроизведения речи с сохранением тембра; воспроизведение в режиме "псевдостерео".

Теперь расскажем о профессиональных продуктах второго основного разработчика программ шумоочистки речи - фирмы "Нелк". В 1995 году на рынок был представлен первый вариант программы "AUDIO", в которой был реализован ряд алгоритмов цифровой фильтрации в сочетании с несколькими алгоритмами нелинейной обработки речевого сигнала. Более поздние версии этой программы, последняя из которых увидела свет в 1997 году, имели существенно более развитый набор методов обработки. Отметим, что все они работали под управлением DOS и использовали для ввода-вывода аудиосигналов всего один тип звуковой платы, впрочем, широко распространенной в те годы, - SB 16 фирмы "Creative Labs". В 1998 году на смену "AUDIO" пришел Windows - вариант, названный "WinAudio". Последняя выпущенная в начале 1999 года версия этой программы - "WinAudio 1.2".

В состав программы вошло достаточно большое число различных процедур анализа и обработки аудиосигналов, в том числе следующие основные процедуры восстановления разборчивости зашумленной речи: нормализация; многополосная фильтрация; сглаживание и коррекция спектра; удаление помех на основе метода спектрального вычитания; "псевдостерео".

Обзор рынка возможных программных средств по обработке речи был бы неполным без упоминания о редакторах звуковых (чаще всего, конечно же, музыкальных) записей сигналов, хранящихся в файлах. Спектр этих программ достаточно широк - от профессиональных до любительских.

В качестве примера рассмотрим такой известный и заслуженно популярный редактор аудиосигналов, как “Cool Edit” американской компании “Syntrillium Software Corporation”. При всем многообразии различных режимов функционирования этой программы, наличии разнообразных методов анализа и обработки сигналов задачу восстановления разборчивости сильно зашумленной речи она решить не может. Существующий в ней инструментарий шумочистки, включающий нормализацию, многополосную фильтрацию и спектральное вычитание, реализован в достаточно общем виде и не учитывает специфику обработки человеческой речи. Это не позволит вам, к примеру, произвольно настроить верхний и нижний уровни нормализации или достаточно точно локализовать модель помехи во временных и частотных границах для использования процедуры спектрального вычитания. Все это вовсе не говорит, что эта программа плоха, это говорит лишь о том, что ее надо использовать по прямому назначению - для обработки обычных записей произвольных аудиосигналов.

Изменение высоты и темпа звука.

Регулировка высоты и темпа звука не вызвала бы особых трудностей, если бы не неотъемлемая физическая связь между этими свойствами звуковой волны. Как уже говорилось, при увеличении вдвое скорости воспроизведения, высота всех звуков возрастает на одну октаву. При уменьшении скорости наблюдается обратный эффект. Использование этой связи позволяет без изменения реальных звуковых данных динамически согласовано изменять высоту звука и скорость воспроизведения, создавая необычное звучание.

Скорость воспроизведения изменяется по времени в соответствии с графиком. Для точек, расположенных выше средней линии скорость увеличивается, для точек, расположенных внизу, - уменьшается. Частота звука изменяется соответственно. Общая продолжительность звучания записи также изменяется в соответствии с формой кривой.

Независимое изменение скорости или частоты требует серьезной коррекции записи и сопровождается внесением в нее искажений, которые могут быть заметны на слух. Тем не менее, в звуковых редакторах алгоритмы независимого изменения частоты звука и времени воспроизведения имеются. Они дают достаточно хорошие результаты, если не злоупотреблять их возможностями (изменения до 20% не вызывают особых нареканий по качеству).

Эффекты с задержкой звука.

Наиболее богатые возможности внесения качественных изменений в звуковой файл предоставляют эффекты, основанные на добавлении в файл тех же самых звуковых данных, но в

измененном виде. Это богатейшая группа различных звуковых эффектов, простейшим из которых является эхо, то есть воспроизведение того же самого звука несколько раз с ослаблением задержкой. В программе Cool Edit Pro имеется несколько возможностей создания эха и близких эффектов.

Однократное эхо представляет собой звук, который повторяется один раз с заданной временной задержкой и заданным ослаблением. Кроме использования этого эффекта по прямому назначению (то есть, действительно в качестве эха), с его помощью можно моделировать стереоэффект на основе монозаписей. Для этого надо во втором канале воспроизвести тот же самый сигнал с минимальной временной задержкой (в тысячные доли секунды). Разность во времени между сигналами, приходящими справа и слева, и воспринимается человеком как стереоэффект. Эхо-сигнал можно инвертировать, что позволяет создавать неожиданные эффекты, основанные на гашении звука.

Многократное эхо позволяет придать звуку окраску, имитирующую его естественное происхождение. Создать такой эффект можно путем формального описания его параметров: величины временной задержки и степени затухания (с учетом естественной фильтрации частот).

Эффект Chorus обеспечивает богатство звучания, создавая впечатление наличия нескольких инструментов или нескольких голосов. Этот эффект достигается за счет добавления копий исходных данных с небольшим смещением по времени и вариациями амплитуды и частоты звука. Эффект Chorus можно также использовать для создания на основе монозаписи (или, точнее говоря, стереозаписи с абсолютно идентичными каналами) стереозаписи, создающей впечатление пространственного звучания.

Своеобразный эффект Flange достигается за счет использования переменной задержки и фазового сдвига. В результате сигнал неравномерно усиливается и ослабляется, создавая специфическое звучание. Первоначально (в 60-70 годы) этого эффекта добивались путем одновременной записи сигнала на два магнитофона, причем на одном из них ленту слегка притормаживали руками. В результате звуковая волна накладывается сама на себя с переменной задержкой, что обеспечивает неоднородное усиление и ослабление звука.

Эффект Reverb моделирует акустическое пространство («помещение, наполненное звуком»). Он состоит из первичных эхо, дающих представление о размерах помещения, и последующих отражений, которые сливаются друг с другом, образуя единый «рассыпающийся» звук. Этот эффект отличается от обычного эха отсутствием регулярных интервалов между повторениями звука.

Другие эффекты.

В программе Cool Edit Pro имеются также такие эффекты, как Distortion, Convolution, а также имеется возможность генерации различных шумов и музыкальных звуков.

Из перечисленных выше эффектов остановимся на Distortion. Этот эффект также имеет названия Gain, Overdrive, Fuzz. Данный эффект получают с помощью перегрузки динамика (или усилителя) по входу. Иными словами, на вход усилителя поступает сигнал с амплитудой, превышающей максимально допустимую. Если сделать это грамотно, то можно получить довольно интересные результаты. В зависимости от видов перегрузки звук можно сделать более певучим, можно – более резким, а можно сделать и «грязным».

Вот так выглядят некоторые переходные характеристики данного эффекта:

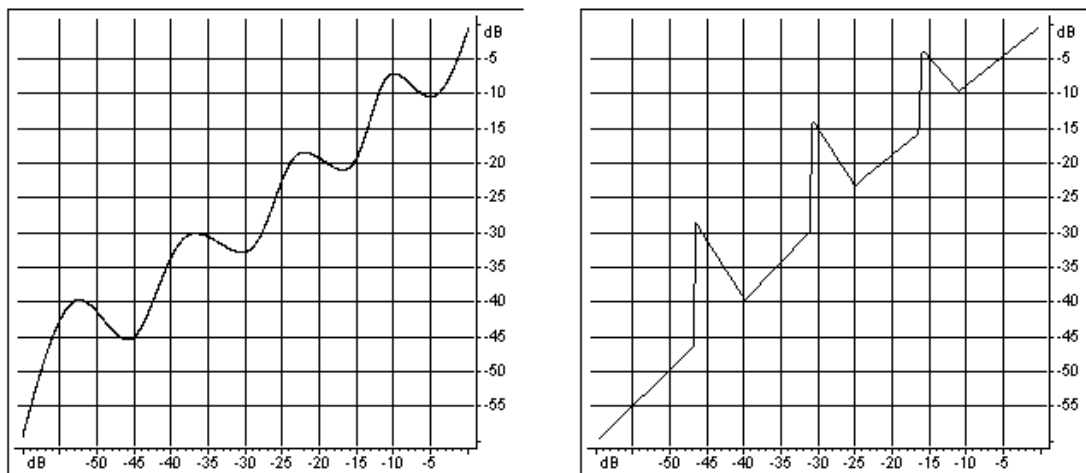


Рис. 26. Эффект Distortion.

В заключение хочется сказать, что эффекты хороши лишь в том случае, если их использовать аккуратно, соблюдая меру. Только тогда можно получить действительно хорошую качественную запись. Неумеренное использование эффектов не приведет ни к чему хорошему, разве только испортит исходную запись.

Кодирование звука.

Некоторые факты о восприятии звука.

Частотный спектр воспринимаемый человеком (примерно) от 20 Hz до 20 kHz, наибольшая чувствительность в диапазоне от 2 до 4 KHz.

Динамический диапазон (от самых тихих воспринимаемых звуков до самых громких) около 96 dB (более чем 1 к 30000 по линейной шкале).

Общеизвестно, что человек в состоянии различить изменение частоты на 0.3% на частоте порядка 1kHz.

Если два сигнала различаются менее чем на 1дБ по амплитуде - они трудноразличимы. Разрешение по амплитуде зависит от частоты и наибольшая чувствительность наблюдается в диапазоне от 2 до 4 КHz.

Пространственное разрешение (способность к локализации источника звука) - до 1 градуса.

Звуки различной частоты распространяются в воздухе с разной скоростью. В результате высокочастотная часть спектра от источника находящегося на удалении от слушателя несколько запаздывает.

Человек не в состоянии заметить внезапное исчезновение высоких частот, если оно не превышает порядка 2ms.

Некоторые исследования показывают, что человек в состоянии ощущать частоты выше 20kHz. С возрастом частотный диапазон сужается.

Речь. Частотный спектр, несущий информацию в человеческой речи: от 500 Hz до 2 kHz

Низкие частоты - басы и гласные. Высокие частоты – согласные.

Лучшее сжатие речи достигается с использованием параметрических кодеров (LPC, CELP, и пр.), пытающихся представить речь как набор параметров некоторой речевой модели. Кодеки общего назначения (MPEG и др.), как правило, дают худшее сжатие.

Методы сжатия, основанные на психоакустике.

Представители: MPEG layers 2, MPEG layer 3 (MP3), AAC (Advanced audio coding), TwinVQ, Ogg Vorbis, и др.

Алгоритм кодека использующего психоакустику обычно состоит из следующих шагов:

Обсчет психоакустической модели (маскирования).

Разделение сигнала на частотные подполосы (FFT, DCT/MDCT, FilterBanks, и т.д.).

Квантование сигнала в подполосах в соответствии с результатами психоакустической модели. Возможно использование одного квантового уровня сразу для нескольких входных значений (векторное квантование - Vector Quantization) - TwinVQ.

Звуковое сопровождение цифрового сигнала вещательного качества (не подвергнутого сжатию) требует всего лишь около 1% общего цифрового потока. Однако наш слух очень чувствителен к искажениям звука, в том числе и вызванным цифровым сжатием, поэтому там, где в этом нет жизненной необходимости, звук лучше вообще не сжимать. Если же речь идет о 10 - кратном и более высоком сжатии видеосигнала, то цифровые потоки сжатого видео и несжатого звука окажутся сравнимыми, поэтому цифровое сжатие звукового сигнала станет целесообразным.

Методы сокращения звуковых данных схожи с методами сжатия видеоинформации. При цифровом сжатии широко применяется слуховая маскировка, основанная на неспособности нашего уха воспринимать определенные звуки в присутствии других (например, по сравни-

тельной громкости звучания, диапазону и т.д.). Таким образом, ухо человека, как и глаз, способно извлечь лишь некоторую часть информации, содержащейся в приходящем звуке.

Назовем основные методы, применяющиеся при цифровом сжатии звука:

- диапазонная регулировка усиления, которая обеспечивает увеличение уровня слабых сигналов для того, чтобы они превысили минимальный уровень шума, причем увеличение усиления должно компенсироваться в декодере;
- квантование, которое обеспечивает устранение слабых звуков. Это уменьшает длину слова отсчетов, что дает экономию при хранении, но может повысить минимальный уровень шума. Применяя маскировку, можно повысить уровень шума, который, однако, будет оставаться неслышимым;
- кодирование с предсказанием, основанное на монотонной природе звуковых колебаний. Вследствие этого можно передавать только разность между предсказанным и фактическим значением, а приемник, где содержится аналогичный предсказатель, будет добавлять только эту разность и восстанавливать исходное значение;
- полосное кодирование, базирующееся на разбиении звукового спектра по диапазонам частот (полосам). Уровни сигналов в большинстве таких полос будут меньше наивысшего уровня сигнала. В полосном кодировании используется тот эффект, что реальные звуки не обладают равномерным распределением спектральной энергии. При этом число используемых частотных полос зависит от того, какие другие методы применяются вместе с полосным кодированием;
- спектральное кодирование, основанное на преобразовании Фурье (ДКП) или на волновом преобразовании (Wavelet Transform). Поскольку спектр звукового сигнала меняется медленно, то его коэффициенты требуется передавать реже, чем звуковые отсчеты.

В большинстве кодеров используется сочетание полосного и спектрального кодирования, причем их действие основано на маскировке шума, вызываемого квантованием.

Традиционные методы сжатия без потерь (Huffman, LZW, итд.) обычно плохо применимы для сжатия аудио информации (по тем же причинам что и при сжатии визуальной информации).

Ниже перечислены некоторые методы сжатия с потерями:

Сжатие тишины(пауз) - определяет периоды "тишины", работает аналогично run - length кодированию.

ADPCM - Adaptive Differential Pulse Code Modulation (в русскоязычной литературе применяется термин адаптивная дельта - импульсно - кодовая модуляция (АДИКМ)).

Например, стандарт ССИТТ G.721 -- от 16 до 32 Kbits/sec. Кодирование разницы между двумя или более последовательными отсчетами; затем разность квантуется --> при квантова-

нии часть информации теряется. Квантование адаптивно (меняет параметры в зависимости от сигнала), в результате меньшее количество бит необходимо для достижения лучшего SNR. Необходимо предсказывать как звук изменится --> сложно

Apple разработал собственную систему названную ACE/MACE. Сжатие с потерями, пытается предсказать, каково будет значение следующего отсчета. Сжатие порядка 2:1.

Linear Predictive Coding (LPC) - пытается описать сигнал с помощью "речевой модели" и передает параметры модели --> звучит как компьютерно синтезированная речь, 2.4 kbits/sec.

Code Excited Linear Predictor (CELP) - тоже самое что и LPC, однако дополнительно передает ошибку квантования (используя предопределенный набор "кодовых слов") --> телефонное качество при 4.8 kbits/sec

MP3 -- сокращение от MPEG Layer3. Это один из потоковых форматов хранения и передачи аудиосигнала в цифровой форме, разработанный Fraunhofer IIS и THOMSON, позднее утвержденный как часть стандартов сжатого видео и аудио MPEG1 и MPEG2. Данная схема является наиболее сложной схемой семейства MPEG Layer 1/2/3. Она требует наибольших затрат машинного времени для кодирования по сравнению с двумя другими и обеспечивает более высокое качество кодирования. Используется главным образом для передачи аудио в реальном времени по сетевым каналам и для кодирования CD Audio.

MP3 -- потоковый формат. Это означает, что передача данных происходит потоком независимых отдельных блоков данных - фреймов. Для этого исходный сигнал при кодировании разбивается на равные по продолжительности участки, именуемые фреймами и кодируемые отдельно. При декодировании сигнал формируется из последовательности декодированных фреймов.

Высокая степень компактности MP3 достигается с помощью дополнительного квантования по установленной схеме, позволяющей минимизировать потери качества.

Подобные техники называются адаптивным кодированием и позволяют экономить на наименее значимых с точки зрения восприятия человеком деталях звучания. Степень сжатия, и, соответственно, объем дополнительного квантования, определяются не форматом, а самим пользователем в момент задания параметров кодирования. Ширина потока (bitrate) при кодировании сигнала, аналогичного CD Audio (44.1kHz 16Bit Stereo) варьируется от наибольшего, 320kbs (320 килобит в секунду, (также пишут kbs, kbps или kb/s), до 96kbs и ниже.

Термин битрейт в общем случае обозначает общую величину потока, количество передаваемой за единицу времени информации, и поэтому не связан с внутренними тонкостями строения потока, его смысл не зависит от того, содержит ли поток моно или стерео, или пятиканальное аудио с текстом на разных языках, или что-либо еще.

Описание процесса кодирования.

Подготовка к кодированию. Фреймовая структура

Перед кодированием исходный сигнал разбивается на участки, называемые фреймами, каждый из которых кодируется отдельно и помещается к конечному файлу независимо от других. Последовательность воспроизведения определяется порядком расположения фреймов. Каждый фрейм может кодироваться с разными параметрами. Информация о них содержится в заголовке фрейма.

Начало кодирования.

Кодирование начинается с того, что исходный сигнал с помощью фильтров разделяется на несколько, представляющих отдельные частотные диапазоны, сумма которых эквивалентна исходному сигналу.

Работа психоакустической модели.

Для каждого диапазона определяется величина маскирующего эффекта, создаваемого сигналом соседних диапазонов и сигналом предыдущего фрейма. Если она превышает мощность сигнала интересующего диапазона или мощность сигнала в нем оказывается ниже определенного опытным путем порога слышимости, то для данного фрейма данный диапазон сигнала не кодируется.

Для оставшихся данных для каждого диапазона определяется, сколькими битами на сэмплы можем пожертвовать, чтобы потери от дополнительного квантования были ниже величины маскирующего эффекта. При этом учитывается, что потеря одного бита ведет к внесению шума квантования величиной порядка 6 dB.

Завершение кодирования.

После завершения работы психоакустической модели формируется итоговый поток, который дополнительно кодируется по Хаффману, на этом кодирование завершается.

Способы кодирования стерео сигнала.

В рамках MP3 кодирование стереосигнала допустимо четырьмя различными методами:

Dual Channel - Каждый канал получает ровно половину потока и кодируется отдельно как моно сигнал. Рекомендуется главным образом в случаях, когда разные каналы содержат принципиально разный сигнал (скажем, текст на разных языках). Выставляется в некоторых кодерах по требованию.

Stereo - Каждый канал кодируется отдельно, но кодер может принять решение отдать одному каналу больше места, чем другому. Это может быть полезно в том случае, когда после отброса части сигнала, лежащей ниже порога слышимости или полностью маскируемой, оказалось, что код не полностью заполняет выделенный для данного канала объем, и кодер имеет возможность использовать это место для кодирования другого канала. В документации к

mp3enc замечено, что этим, например, избегается кодирование "тишины" в одном канале, когда в другом есть сигнал. Данный режим выставлен по умолчанию в большинстве ISO-based кодеров, а также используется продукцией FhG IIS на битрейтах выше 192kbs. Применим и на более низких битрейтах порядка 128kbs... 160kbs.

Joint Stereo (MS Stereo) - Стереосигнал раскладывается на средний между каналами и разностный. При этом второй кодируется с меньшим битрейтом. Это позволяет несколько увеличить качество кодирования в обычной ситуации, когда каналы по фазе совпадают. Но приводит и к резкому его ухудшению, если кодируются сигналы, по фазе не совпадающие. В частности, фазовый сдвиг практически всегда присутствует в записях, оцифрованных с аудиокассет, но встречается и на CD, особенно если CD сам был записан в свое время с аудиоленты.

Joint Stereo (MS/LS Stereo) - Вводит еще один метод упрощения стереосигнала, повышающий качество кодирования на особо низких битрейтах. Состоит в том, что для некоторых частотных диапазонов оставляется уже даже не разностный сигнал, а только отношение мощностей сигнала в разных каналах. Понятно, для кодирования этой информации употребляется еще меньший битрейт.

В отличие от всех предыдущих, этот метод приводит к потере фазовой информации, но выгоды от экономии места в пользу среднего сигнала оказываются выше, если речь идет о очень низких битрейтах.

Этот режим по умолчанию используется продукцией FhG IIS для высоких частот на битрейтах от 96kbs и ниже (другими качественными кодерами этот режим практически не используется).

Но, как уже говорилось, при применении данного режима происходит потеря фазовой информации, также теряется любой противофазный сигнал

Способы хранения MP3.

Стандарт MP3 не определяет никакого точного стандартного математического алгоритма кодирования, его разработка целиком и полностью остается на совести разработчиков кодеров. Вместо этого он определяет общую схему процесса кодирования, а также формат закодированного фрейма. Сами последовательности фреймов могут передаваться потоком (процесс передачи такого потока называется streaming) или храниться в файлах.

MP3 файл, как и поток, состоит из последовательно расположенных фреймов, между которыми может содержаться произвольная информация. Основное требование состоит в том, что не должно быть совпадений с сигнатурой начала фрейма.

Часто к последовательности фреймов добавляют стандартный заголовок метааудиоформата WAV, и получается то, что называют WAV-MP3. Еще чаще к MP3-файлу до-

бавляется информационный блок ID3v2, содержащий информацию об исполнителе, жанре, названии композиции, и другую подобную информацию о треке. Он добавляется в конец файла. В середине пока никто ничего ставить не придумал. Хотя, вообще говоря, может представлять некоторый интерес вставка спецтэга для VBR с информацией о том, в какой части трека мы, собственно, находимся.

Формат QDesign AIF.

Этот формат аудиосжатия был разработан компанией QDesign и впоследствии был активно поддержан концерном Apple/Macintosh. QDesign AIF является доработкой семейства стандартов AIFF, которое представляет собой разновидность мультимедийных стандартов используемых на платформе Apple/Macintosh. Пара QDesign AIF - AIFF является полным аналогом пары WAV – MP3 используемой на платформе Wintel, за исключением степени сжатия.

Малый размер файла полностью соответствует его низкому качеству. Так, из аудиокomпозиции удаляются многие частоты, как из верхнего, так и из нижнего диапазона звукового спектра, в результате звучание становится не только очень глухим, но при этом утрачивает и характерные басы. Очень высок шум квантования. Для уменьшения размера композиции некоторые места преобразовываются из стерео в моно.

Однако QDesign AIF с максимально возможным bitrate 48 kbs все же лучше, чем MP3, AAC, PAC и VQF с этой же шириной потока и безусловно лучше real audio. Этот формат годится только для сетевого радио или для ознакомления с композицией, чтобы впоследствии закачать ее в виде более громоздких, но зато и более качественных файлов в форматах MP3, AAC, VQF.

Композиции в формате QDesign AIF с bitrate 48 kbs примерно соответствуют по качеству MP3 64 kbs, хотя эти форматы довольно несхожи между собой и обеспечивают совершенно разное звучание одних и тех же композиций.

Формат TwinVQ (VQF).

Алгоритм аудиокомпрессии TwinVQ (Transform-domain Weighted Interleave Vector Quantization - векторное квантование с преобразуемыми доменами и взвешенным чередованием) был разработан японской фирмой Nippon Telegraph and Telephone Corp. (NTT).

Алгоритм TwinVQ позволяет сжимать оцифрованные аудиопотоки со степенью сжатия до 1:20. Этот формат по основным концепциям, используемым при компрессии аудиоданных, сильно напоминает MP3, но при этом используется совершенно иная психоакустическая модель. Соответственно и выбор "нужных" и "ненужных" звуков осуществляется по совершенно иным критериям. Размер файлов VQF в среднем на 30-35% меньше, чем MP3, при примерно одинаковом качестве звука. Так, например, качество звучания потока TwinVQ при

bitrate 96 Кбит/с практически идентично качеству звучания потока MPEG-1 Layer 3 (при bitrate 128 Кбит/с) и семейству MPEG-2 AAC (при bitrate 96 Кбит/с). Алгоритм TwinVQ позволяет также как и MP3, производить декодирование и воспроизведение потока непосредственно при его получении (stream playback). Но если композиция в формате VQF и не докачается, то ее все равно можно прослушать до места обрыва. На частотах выше 15 кГц, VQF на 2-3 дБ уступает MP3. При всем при этом, VQF вносит гораздо меньшее искажение формы сигнала в композициях с большим динамическим диапазоном (реальная музыка).

Такое высокое качество при сравнительно низком bitrate достигается благодаря использованию более совершенных алгоритмов сжатия. При этом резко возрастает нагрузка процессора, как при кодировании, так и при декодировании аудиофайлов. Кодеки TwinVQ при компрессии аудиоданных работают, как правило, в 5 - 10 раз медленнее, чем кодеки MPEG Layer 3. При декодировании файлов VQF проигрывателем, нагрузка процессора на 30 - 40% выше, по сравнению с декодированием MP3. Соответственно минимальные требования к конфигурации компьютера для TwinVQ гораздо выше, чем для MP3. В качестве стартового условия для качественного проигрывания аудиокомпозиций в формате VQF нужен как минимум процессор Pentium-100. При компрессии аудиофайлов в формат VQF так же сказывается высокая ресурсоемкость алгоритма TwinVQ.

Помимо более высокой нагрузки процессора при кодировании/декодировании аудиофайлов, для формата VQF характерна очень медленная перемотка, которая реализуется за счет быстрой распаковки файла.

Формат VQF поддерживает bitrate от 8 kbs до 48 kbs на канал (соответственно максимальный bitrate для стереокомпозиций составляет 96 kbs). Поддерживается кодирование с переменным bitrate. Из не стандартных по сравнению с MP3 возможностей можно назвать поддержку несимметричных bitrate, например один канал может кодироваться с bitrate 10 kbs, а другой с bitrate 11 kbs.

Наработки TwinVQ совместно с AAC используются в звуковой части стандарта видеосжатия MPEG-4.

Формат PASC.

Название формата PASC расшифровывается как perceptual audio coding (аудиокодирование основанное на восприятии).

Данный формат был разработан фирмой Lucent Technologies при инвестиционной поддержке компании Bell Labs.

Звучание в целом примерно такого же качества, как и у лучших кодеков из семейства AAC – Liquid Pro AAC и Astrid/Quartex AAC. PASC с bitrate 96 kbs лучше MP3 128 kbs, а PASC с bitrate 128 kbs приближается к CD звучанию, но не достигает его.

При этом достигается высокая скорость кодирования при использовании Audio Library 1.0.

К числу недостатков этого кодека можно отнести и неприменимость сжатых композиций в качестве сетевого формата. Формат не поддерживает потоковую пересылку данных, то есть одновременное воспроизведение и получение аудиокomпозиции. Это формат используется только для домашней аудиотеки и для продажи на CD - дисках.

Кроме того, аудиокomпозиции в данном формате имеют сильную защиту от нелегального. На основании этого факта можно предположить, что механизм защиты содержит привязку в аппаратному обеспечению того компьютера, на котором производилось кодирование (например, привязка к серийному номеру жесткого диска).

Формат WMA (Windows Media Audio).

В этом кодеке было достигнуто качество MP3 128 kbs на bitrate всего 64 kbs.

Сжатая этим кодеком голосовая информация обладала, даже при bitrate всего 64 kbs, очень высокой четкостью. Профессиональными тестерами было установлено, что при bitrate 64 kbs слоговая разборчивость голоса достигала 90%, в то время, как у других форматов аудиосжатия подобный показатель наблюдается при bitrate в 2 - 2.5 раза больше, т.е. при bitrate 128 kbs и 160 kbs соответственно.

Низкобитрейтная версия WMA по качеству ощутимо превосходит Real Audio. Соотношение размер/качество для WMA в 2 - 3 раза выше, чем аналогичный параметр у RA. Более того, WMA превосходит по качеству формат QDesign AIFF. Так, при максимальном для QDesign AIFF bitrate 48 kbs, аудиофайлы WMA имеют гораздо меньше искажений и потерь. Кроме того, формат WMA бесплатен

WMA с bitrate 64 kbs лучше MP3 128 kbs или по крайней мере обладает тем же качеством. Кодек позволяет легко перекодировать из MP3 в WMA с любым bitrate.

В кодеке WMA поддерживаются bitrate от 6 kbs до 96.1 kbs на канал. Помимо этого, кодек содержит своего рода конструктор для пользователей, в котором можно собрать собственный bitrate - произвольно задать количество kbs, задать частоту выборки (по умолчанию 44 КГц) в диапазоне от 22 КГц до 48 КГц и даже изменить алгоритм сжатия .

Скорость кодирования чуть медленнее PAC, но гораздо быстрее VQF, AAC, и QDesign AIFF.

Из недостатков можно назвать: заполнение пауз в аудиокomпозициях посторонними шумами и одноплатформенность WMA (только для пользователей Windows). А также качество WMA 96 kbs несколько ниже MP3 256 kbs.

Формат OggVorbis.

OggVorbis - это универсальный формат аудиокomпрессии. Этот самый молодой (выпущен в 2000г.) формат из всех конкурентов MP3 разработан группой Xiphophorus.

OggVorbis принадлежит к тому же типу форматов аудиосжатия, что и MP3, AAC, VQF, PAC, QDesign AIFF и WMA, то есть к форматам сжатия с потерями. Психоакустическая модель, используемая в OggVorbis по принципам действия близка к MP3, но математическая обработка и практическая реализация этой модели в корне отличается, что позволяет авторам объявить свой формат совершенно независимым от всех предшественников.

Одно из преимуществ формата OggVorbis - это его полная открытость и бесплатность.

Помимо этого, OggVorbis, как спецификация, обладает так же еще целым рядом неоспоримых достоинств. Так, верхняя планка частоты выборки составляет не 44 КГц, как у всех форматов, а 48 КГц, что безусловно более близко к живой музыке по сравнению с CD. Кроме того, число каналов не ограничено двумя как обычно - моно и стерео, а достигает 255. MP3 и OggVorbis трудно сравнивать, но в целом звучание OggVorbis гораздо лучше.

При кодировании кодеки OggVorbis используют VBR (variable bitrate), подобно некоторым MP3 кодекам, что позволяет существенно уменьшить размер композиции, при незначительной потере качества.

В спецификации заложен широкий диапазон - от 8 kbs до 512 kbs. Спецификация OggVorbis содержит очень гибкий и развитый механизм включения комментариев и иллюстраций в тело аудиокomпозиции. Заголовок комментария легко расширяется и позволяет включать тексты любой длины и сложности, перемежающиеся изображениями.

Скорость кодека OggVorbis не быстрее кодека MP3.

OggVorbis, так же как и MP3, изначально разрабатывался как сетевой потоковый формат. Это свойство является очень важным, особенно учитывая мультиплатформенную направленность формата OggVorbis. Интернет-радиостанция, использующая низкобитрейтные версии OggVorbis сможет вещать сразу на всех платформах, тогда как такая же радиостанция, использующая для передачи WMA (в виде ASF), будет ограничена только пользователями Windows.

Основные группы звуковых плат.

Все звуковые платы по назначению можно разделить на три группы:

1. Звуковые, содержащие только тракт цифровой записи/воспроизведения. Эти платы позволяют только записывать или воспроизводить непрерывный звуковой поток, наподобие магнитофона. Вся работа по запоминанию записываемого и подготовке воспроизводимого потока возлагается на программное обеспечение. Оцифрованный звук при этом в самой плате не хранится. Некоторые звуковые платы имеют встроенные сигнальные процессоры для обработки звука в процессе его записи или воспроизведения.

2. Музыкальные, содержащие только музыкальный синтезатор. Такие платы ориентированы прежде всего на генерацию относительно коротких музыкальных звуков по командам от центрального процессора; сами звуки при этом либо создаются параметрически, либо воспроизводятся оцифровки, заранее помещенные в память синтезатора (ПЗУ или ОЗУ). Музыкальные платы не имеют возможности записи звука и, даже при наличии ОЗУ в синтезаторе, не рассчитаны на воспроизведение непрерывного звукового потока, хотя иногда этого можно добиться при помощи особых методов. Некоторые музыкальные платы содержат эффект - процессор для обработки создаваемого звука.

3. Комбинированные, или звуко - музыкальные, с объединенным на одной плате цифровым трактом и музыкальным синтезатором. Обычно под словом "синтезатор" подразумевается WT; платы только с FM - синтезатором, который сильно ограничен для музыкального применения, чаще всего относят к категории чисто звуковых.

По конструкции все платы делятся на обычные, или основные, называемые по традиции "*картами*", которые вставляются в разъем системной магистрали (ISA или PCI), и дочерние, подключаемые к специальному 26-контактному разъему на основной карте. По сути, дочерняя плата как бы "надевается" на разъем, удерживаясь на нем только силой трения контактов и фиксирующих штифтов.

Из - за ограничений интерфейса между основной и дочерней платами дочерние платы могут быть только чисто музыкальными – никаких возможностей по записи/воспроизведению звукового потока они иметь не могут.

Структура современных звуковых плат

В комбинированных картах можно выделить 4 более - менее независимых блока:

1. Блок цифровой записи/воспроизведения, называемый также цифровым каналом, или трактом карты. Осуществляет преобразования аналог -> цифра и цифра -> аналог в режиме программной передачи или по DMA. Состоит из узла, непосредственно выполняющего аналогово - цифровые преобразования – АЦП/ЦАП (международное обозначение - coder/decoder, codec, ADC/DAC), и узла управления. АЦП/ЦАП либо интегрируется в состав одной из микросхем карты, либо применяется отдельная микросхема (AD1848, CS4231, ST1703 и т.п.). От качества применяемого АЦП/ЦАП во многом зависит качество оцифровки и воспроизведения звука; не меньше зависит она и от входных и выходных усилителей.

Цифровой канал большинства распространенных карт (кроме GUS) совместим с Sound Blaster Pro (8 разрядов, 44 кГц - моно, 22 кГц - стерео).

2. Блок синтезатора. Построен либо на базе микросхем FM - синтеза, либо на базе микросхем WT - синтеза, либо того и другого вместе. Работает либо под управлением

драйвера (FM, большинство WT) - программная реализация MIDI, либо под управлением собственного процессора - аппаратная реализация. Почти все FM - синтезаторы совместимы между собой, различные WT- синтезаторы - нет. Большинство WT - синтезаторов содержит встроенное ПЗУ со стандартным набором инструментов General MIDI (128 мелодических и 37 ударных инструментов), а также ОЗУ для загрузки дополнительных оцифрованных звуков, которые будут использоваться при исполнении музыки.

3. Блок MPU. Осуществляет прием/передачу данных по внешнему MIDI-интерфейсу, выведенному на разъем MIDI/Joystick и разъем для дочерних MIDI-плат. Обычно более или менее совместим с интерфейсом MPU-401, но чаще всего требуется программная поддержка.

4. Блок микшера. Осуществляет регулирование уровней, коммутацию и сведение используемых на карте аналоговых сигналов. В состав микшера входят предварительные, промежуточные и выходные усилители звуковых сигналов.

В дочерних платах основными блоками являются собственно музыкальный синтезатор и блок MIDI - интерфейса, через который плата получает MIDI - сообщения с основной карты. Синтезатор обязательно имеет ПЗУ различного объема; наличие ОЗУ возможно, но неудобно, поскольку MIDI является достаточно медленным для загрузки оцифровок интерфейсом. Синтезированный звук возвращается в основную карту по аналоговому стереоканалу.

Технологии создания позиционируемого 3D - звука

Важно видеть различия между типами технологий 3D звука, прежде всего по функциям:

- Stereo Expansion (Расширение стерео): технология, которая оперирует с имеющейся избыточной стерео информацией, надлежащим образом расширяя кажущуюся ширину звукового поля (т.е. главным образом удобная для не - 3D стерео произведений, таких как записанная музыка).

Positional 3D Audio (Позиционируемый 3D звук): технология, которая оперирует с множеством индивидуальных звуковых потоков и пытается определить местоположение каждого из них индивидуально в 3D пространстве.

Virtual Surround (Виртуальный окружающий звук): технология, которая оперирует с декодированными данными в формате surround с целью воспроизведения разнообразных каналов в их истинной перспективе с использованием ограниченного числа источников звука, например воспроизведение пятиканального звука на двух акустических колонках.

Наушники упрощают решение проблемы доставки одного звука к одному уху и другого звука к другому уху. Тем не менее, использование наушников имеет и недостатки.

Даже легкие беспроводные наушники могут быть обременительны. Наушники, обеспечивающие наилучшую акустику, могут быть чрезвычайно неудобными при длительном прослушивании.

Наушники могут иметь провалы и пики в своих частотных характеристиках, которые соответствуют характеристикам ушной раковины. Если такого соответствия нет, то восприятие звука, источник которого находится в вертикальной плоскости, может быть ухудшено. Иначе говоря, мы будем слышать преимущественно только звук, источники которого находятся в горизонтальной плоскости.

При прослушивании в наушниках, создается ощущение, что источник звука находится очень близко. И действительно, физический источник звука находится очень близко к уху, поэтому необходимая компенсация для избавления от акустических сигналов влияющих на определение местоположения физических источников звука зависит от расположения самих наушников.

К звуковой катушке подводится переменное напряжение - звуковой сигнал в электрической форме. Под действием переменного напряжения по звуковой катушке протекает переменный ток, который создает магнитное поле катушки, взаимодействующее с постоянным магнитным полем магнита, в результате чего звуковая катушка втягивается в керн или выталкивается из него. Вместе со звуковой катушкой движется диффузор, и в процессе движения, подобно поршню насоса, он сжимает воздух перед собой и создает разрежение позади себя. В результате в воздухе возникают сменяющие друг друга сжатия и разрежения, т. е. звуковые волны.

Диффузор электродинамической головки, имея определенную массу и обладая упругостью, является механической колебательной системой. Это означает, что на определенной частоте возникает явление резонанса.

Очевидно, что интенсивность движения диффузора зависит от значения подведенного напряжения и электрического сопротивления головки.

Полное электрическое сопротивление головки (импеданс) - это сопротивление переменному току, измеренное на зажимах головки. На основной частоте механического резонанса подвижной части головки импеданс достигает значительной величины и с понижением частоты быстро падает до сопротивления звуковой катушки постоянному току. На высших звуковых частотах импеданс растет из-за влияния индуктивности катушки. Отметим, что сопротивление, в зависимости от частоты сигнала, может изменяться на $\pm 20\%$ от полного сопротивления. Поэтому номинальным значением полного сопротивления считают сопротивление на фиксированной частоте, равной 1000 Гц (1 кГц), или его минимальное значение в диапазоне частот выше частоты основного механического резонанса.

Расстояния от диффузора до уха для высоких частот достаточно, чтобы в прошлом противофазные волны пришли к пункту своего назначения - уху - в полном согласии, в фазе. А вот на низших частотах и даже на средних сдвиг фаз получается небольшим и звук заметно ослабляется.

Чтобы устранить интерференцию волн низших частот, излучаемых передней и тыльной сторонами диффузора, применяют различные виды акустического оформления громкоговорителя. Иными словами, электродинамические головки размещают в корпусе, который помимо декоративной выполняет роль акустического экрана.

Назначение акустического экрана очевидно: он необходим, чтобы погасить звуковую волну от задней стороны динамика или удлинить ее путь, чтобы в точке приема она стала синфазной с волной, излученной передней стороной.

Есть несколько приемов формирования необходимых акустических характеристик корпуса. Один из них - размещение головок не только на лицевой стенке, но и на боковых. Подобная мера улучшает диаграмму направленности громкоговорителя, поскольку излучение звука происходит равномерно во всех направлениях, при этом появляется ощущение объемности звучания.

В настоящее время громкоговорители РС оформляются в корпусе с закрытой задней стенкой, в корпусе с фазоинвертором или в комбинированном корпусе (с закрытой задней стенкой и фазоинвертором).

Корпус с закрытой задней стенкой представляет собой футляр, имеющий, как правило, форму параллелепипеда. Внутреннее пространство футляра заполняется звукопоглощающим материалом (обычно ватой или мелко нарезанными кусочками поролона). Для эффективного подавления «задней» волны объем такого корпуса должен быть довольно большим, размеры громкоговорителя, соответственно, тоже будут большими, что требует достаточно места для размещения такой АС на рабочем столе.

На высоких частотах длины пути от диффузора до уха достаточно, чтобы в прошлом противофазные волны пришли к пункту своего назначения - уху - в полном согласии, в фазе. А вот на низших частотах и даже на средних сдвиг фаз получается небольшим и звук заметно ослабляется.

Есть несколько приемов формирования необходимых акустических характеристик корпуса. Один из них - размещение головок не только на лицевой стенке, но и на боковых. Подобная мера улучшает диаграмму направленности громкоговорителя, поскольку излучение звука происходит равномерно во всех направлениях, при этом появляется ощущение объемности звучания.

Кроме того, корпусу передаются колебания диффузора и он сам превращается в излучатель звука. Поэтому неудачный выбор конфигурации корпуса АС может вследствие дифракции отрицательно сказаться на равномерности частотной характеристики в области средних частот. Установлено, что наиболее удачным является сферический корпус громкоговорителя: чем более тупые углы прилегают к динамику, тем меньше дифракция. Однако изготовление АС в таком корпусе связано с рядом технологических трудностей, да и ее дизайн не совсем соответствует дизайну компьютерного оборудования.

Наиболее широкое распространение получили АС в корпусе с фазоинвертором. Фазоинвертор представляет собой закрытый футляр с дополнительным отверстием на передней стенке.

Масса воздуха в отверстии, подобно диффузору, является дополнительным излучателем звука, благодаря использованию которого удастся увеличить звуковую отдачу на самых низких частотах. Происходит это потому, что диффузор головки через воздух, сосредоточенный в корпусе, связан с воздухом в фазоинверторе, вследствие чего колебания последнего сдвинуты по фазе по отношению к колебаниям задней стороны диффузора. В результате, колебания воздуха в отверстии почти синфазны с колебаниями передней стороны диффузора.

Правильно сконструированный фазоинвертор не только улучшает частотную характеристику громкоговорителя в области низших частот, но и способствует уменьшению нелинейных искажений в области средних частот.

Кроме того, при использовании фазоинвертора, как правило, не нужно заполнять внутренний объем звукопоглощающим материалом, следовательно, при тех же характеристиках размеры громкоговорителя значительно меньше. Именно поэтому большинство АС для PC оформляются в литом, из специальной пластмассы, корпусе со скругленными углами и фазоинвертором.

В составе современных АС, кроме двух стандартных громкоговорителей, часто используется третий, дополнительный, специально предназначенный для воспроизведения звука низких и сверхнизких (ниже 150 Гц) частот - так называемый сабвуфер (subwoofer),

Сабвуфер (буквально - «подвыватель») представляет собой громкоговоритель довольно большого размера в корпусе с фазоинвертором. Причем мощный низкочастотный динамик, установленный на горизонтальной панели внутри корпуса (рис.), непосредственно в окружающее пространство звуковых волн не излучает, а только возбуждает воздух в фазоинверторе на резонансной частоте последнего (около 50 Гц).

Благодаря использованию высококачественного мощного встроенного усилителя и свойствам фазоинвертора звуковые колебания узкой полосы низших частот излучаются с минимальными искажениями. Это позволяет создать ощущение реальности звуковых эффектов.

Использование акустических колонок позволяет обойти большинство из этих проблем, но при этом не совсем понятно, как можно использовать колонки для воспроизведения бинаурального звука (т.е. звука, предназначенного для прослушивания в наушниках, когда часть сигнала предназначена для одного уха, а другая часть для другого уха). Как только мы подключим вместо наушников колонки, наше правое ухо начнет слышать не только звук, предназначенный для него, но и часть звука, предназначенную для левого уха. Одним из решений такой проблемы является использование техники cross - talk - cancelled stereo или transaural stereo, чаще называемой просто алгоритм crosstalk cancellation (для краткости CC).

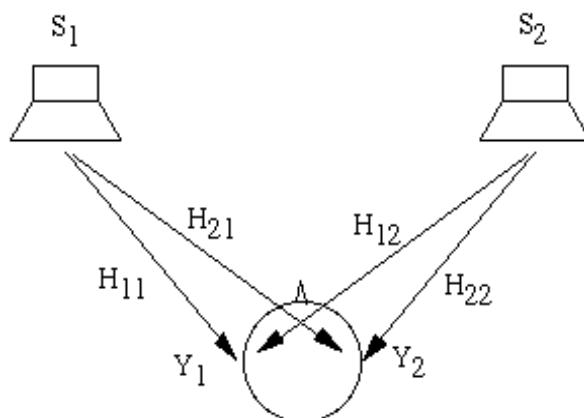


Рис. 27 . Формирование стерео.

Отличительной особенностью технологии A3D фирмы Auzear является использование метода трассировки звука. В играх, например, источниками звуков (цифровых потоков) могут быть персонажи, всевозможные объекты, окружающая среда. Алгоритмы обьсчета путей распространения звуковых волн воспроизводят эффект распространения звука в окружающей среде.

Технология моделирует звуковое окружение, включает дополнительную информацию в звуковой поток в форме различий амплитуд, фаз и временных задержек между выходными каналами.

При моделировании широко используются следующие эффекты.

- Реверберация. Отличается от эффекта “эхо” отсутствием периодичности при повторе звуковых последовательностей. Позволяет моделировать акустические свойства помещения.
- Оклюзия. Моделирует ослабление звука, прошедшего сквозь препятствие из определённого материала (кирпич, металл, бетон и т. д.).
- Обструкция. Моделирует дифракцию звука, закрытого преградой, т. е. звуки, задержанные препятствием.

В результате при прослушивании 3D – звука создаётся ощущение, что источники звука выходят за пределы горизонтальной плоскости.

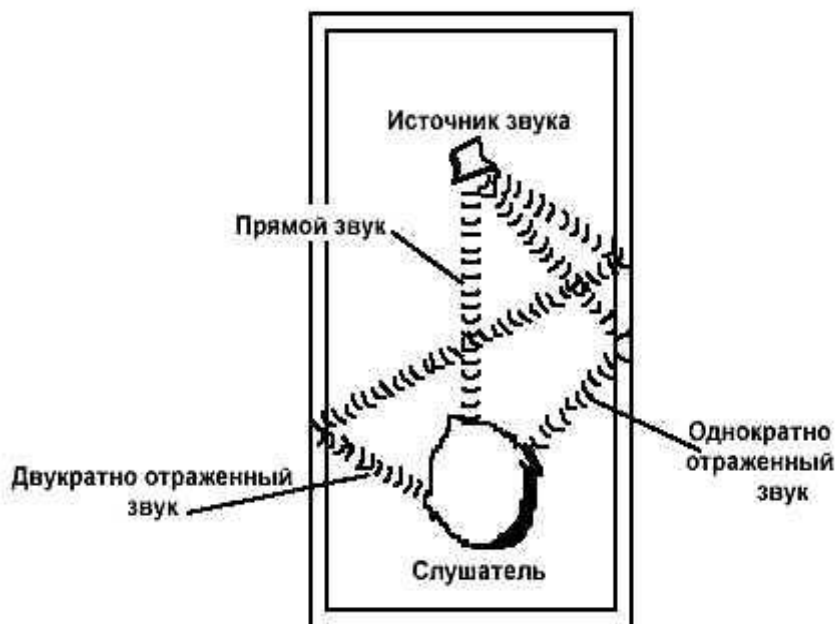


Рис. 28. Формирование трёхмерного звука.

Как видно из рисунка, при распространении звуковые волны достигают слушателя различными путями:

- они могут следовать по прямому пути к слушателю;
- один раз отразившись от объекта (путь однократно отраженного звука);
- отраженный дважды (путь вторично отраженного звука) и более раз;
- звуки могут так же проходить сквозь объекты, такие, как вода или стены.

Существуют три главных компонента: интерфейс A3D, geometry engine (геометрический движок, определяющий геометрию объектов в пространстве) и scene manager (менеджер сцены). Интерфейс A3D является основным компонентом. Один в отдельности он используется для реализации прямых путей распространения звука (direct path). Geometry engine является основным компонентом для обьсчета отраженных и прошедших сквозь препятствия акустических звуковых волн или для Acoustic Wavetracing. Менеджер сцены используется как геометрическим движком, так и интерфейсом A3D для управления сложными звуковыми сценами. Обработка каждого из этих компонентов будет производиться именно в таком порядке. Взаимосвязь и функционирование менеджера сцены, геометрического движка и реализация прямых путей распространения звука показаны ниже:



Рис. 29. Менеджер сцены и геометрический движок.

Видео.

Существует много способов получения видеоматериала. Но видеолента, на использование которой нет официального разрешения, может надолго стать головной болью. Приобретение разрешения также может стать трудно разрешимой и очень дорогой проблемой.

Съёмка видео является наиболее сложным элементом создания мультимедиа и требует специальных знаний. В большинстве случаев студии имеют возможности редактирования и доработки материала, которые на вашем компьютере недоступны. Однако, заказ съёмки в профессиональной киностудии может оказаться слишком дорогим удовольствием. Скорее всего, при создании мультимедиа – фильма стоимость проекта не должна выходить за определённые границы.

В мультимедиа используются те же правила съёмки видео, что и для телевидения. Для воспроизведения на небольшом экране можно избежать вывода кадров с широкой перспективой. Эффект размытого движения панорамы не заметен на небольших окнах. Следует чаще применять средний, близкий и портретный кадр. Рекомендуются больше держать камеру неподвижно, позволять предмету входить в камеру.

Не следует недооценивать значение опоры для видеокамеры. Для всех “домашних” видеосъёмок характерно дрожание изображения. Треножник или даже скрученный свитер на капоте автомобиля помогут улучшить качество съёмки. Если же необходимо производить съёмку с руки, то следует пользоваться камерой с электронной стабилизацией изображения.

Однако, наибольшее отличие любительского видео от профессионального заключается в способности работать со слабым освещением.

Стандартная схема освещения в студии имеет четыре типа освещения: задний свет, прямой свет, антитеневое освещение и фоновое освещение.

Следует придерживаться также следующих правил:

Заголовки должны появляться медленно и присутствовать на экране достаточно долго, затем – медленно растворяться.

Нельзя слишком загромождать экран.

Нельзя использовать слишком часто параллельные линии, окна, концентрические окружности. Эти объекты должны иметь большие размеры и формироваться толстыми линиями.

Соседние цвета должны отличаться по интенсивности. Например, светлый голубой и тёмный красный.

Аналоговый метод передачи видеосигналов

Самым ранним методом передачи видеосигналов является аналоговый метод. Одним из первых видеоформатов на основе этого принципа стал композитный видеосигнал. Композитное аналоговое видео комбинирует все видеокomпоненты (яркость, цвет, синхронизацию и т. п.) в один сигнал. Из-за объединения этих элементов в одном сигнале качество композитного видео далеко от совершенства. В результате мы имеем неточную передачу цвета, недостаточно "чистую" картинку и другие факторы потери качества. Композитное видео быстро уступило дорогу компонентному видео, в котором различные видеокomпоненты представлены как независимые сигналы. Дальнейшие усовершенствования этого формата привели к появлению различных его вариаций: S - Video, RGB, Y, Pb, Pr и др. Тем не менее все вышеперечисленные форматы остаются аналоговыми по своей сути, и, следовательно, обладают одним существенным недостатком: при копировании дубль всегда уступает по качеству оригиналу. Потеря качества при копировании видеоматериала аналогична фотокопированию, когда копия никогда не бывает такой же четкой и яркой, как оригинал.

Цифровое видео

Недостатки, присущие аналоговому способу воспроизведения видео, в конце концов привели к разработке цифрового видеоформата. На смену аналоговому видео пришло цифровое. В области профессионального видео применяется несколько цифровых видеоформатов: D1, D2, Digital BetaCam и др. В отличие от аналогового видео, качество которого падает при копировании, каждая копия цифрового видео идентична оригиналу. Хотя современный видеоряд базируется на цифровой основе, практически все цифровые видеоформаты до сих пор в качестве носителя исходного сигнала используют пленку с последовательным доступом. Поэтому большинству профессионалов в области видео все еще привычнее работать с пленкой, чем с компьютером. Конечно, пленка в качестве источника данных все еще остается более предпочтительной, чем жесткий диск компьютера, поскольку вмещает значительно больший объем данных. Но зато для цифрового видеомонтажа использование компьютеров дает ряд существенных преимуществ: не только обеспечивает прямой доступ к любому видеофрагменту (что невозможно при работе с пленкой, поскольку к необходимым участкам можно добраться лишь последовательно просматривая видеоматериал), но и предполагает широкие возможности обработки изображения (редактирование, сжатие). Это достаточно веские причины для перехода видеопроизводства с традиционного оборудования на компьютерное. Компьютерное цифровое видео представляет собой последовательность цифровых изображений и связанный с ними звук. Элементы видео хранятся в цифровом формате. Существует множество способов захвата, хранения и воспроизведения видео на компьютере. С появлением

нием компьютерного цифрового видео стихийно стали возникать самые разнообразные форматы представления видеоданных, что поначалу привело к некоторой путанице и вызвало проблемы совместимости. Однако в последние годы благодаря усилиям Международной организации по стандартизации (ISO -- International Standards Organisation) выработаны единые стандарты на форматы видеоданных, которые мы позже рассмотрим.

Основные характеристики цифрового видео Цифровое видео характеризуется четырьмя основными факторами: частота кадра (Frame Rate), экранное разрешение (Spatial Resolution), глубина цвета (Color Resolution) и качество изображения (Image Quality).

Частота кадра. Стандартная скорость воспроизведения видеосигнала -- 30 кадров/с (для кино этот показатель составляет 24 кадра/с). Каждый кадр состоит из определенного количества строк, которые прорисовываются не последовательно, а через одну, в результате чего получается два полукадра, или так называемых "полей". Поэтому каждая секунда аналогового видеосигнала состоит из 60 полей (полукадров). Такой процесс называется interlaced видео. Между тем монитор компьютера для прорисовки экрана использует метод "прогрессивного сканирования" (progressive scan), при котором строки кадра формируются последовательно, сверху вниз, а полный кадр прорисовывается 30 раз каждую секунду. Разумеется, подобный метод получил название non-interlaced видео. В этом заключается основное отличие между компьютерным и телевизионным методом формирования видеосигнала.

Глубина цвета. Этот показатель является комплексным и определяет количество цветов, одновременно отображаемых на экране. Компьютеры обрабатывают цвет в RGB-формате (красный – зеленый - синий), в то время как видео использует и другие методы. Одна из наиболее распространенных моделей цветности для видеоформатов -- YUV. Каждая из моделей RGB и YUV может быть представлена разными уровнями глубины цвета (максимального количества цветов). Для цветовой модели RGB обычно характерны следующие режимы глубины цвета: 8 бит/пиксель (256 цветов), 16 бит/пиксель (65,535 цветов) и 24 бит/пиксель (16,7 млн. цветов). Для модели YUV применяются режимы: 7 бит/пиксель (4:1:1 или 4:2:2, примерно 2 млн. цветов), и 8 бит/пиксель (4:4:4, примерно 16 млн. цветов).

Экранное разрешение. Еще одна характеристика - экранное разрешение, или, другими словами, количество точек, из которых состоит изображение на экране. Прямой связи между разрешением аналогового видео и компьютерного дисплея нет. Стандартный аналоговый видеосигнал дает полноэкранное изображение без ограничений размера, так часто присущих компьютерному видео. Телевизионный стандарт NTSC (National Television Standards Committee), разработан Национальным комитетом по телевизионным стандартам США. Используемый в Северной Америке и Японии, он предусматривает разрешение 768 на 484. Стандарт PAL (Phase Alternative), распространенный в Европе, имеет несколько большее разрешение --

768 на 576 точек. Поскольку разрешение аналогового и компьютерного видео различается, при преобразовании аналогового видео в цифровой формат приходится иногда масштабировать и уменьшать изображение, что приводит к некоторой потере качества.

Качество изображения. Последняя, и наиболее важна характеристика - это качество видеоизображения. Требования к качеству зависят от конкретной задачи. Иногда достаточно, чтобы картинка была размером в четверть экрана с палитрой из 256-ти цветов (8 бит), при скорости воспроизведения 15 кадров/с. В других случаях требуется полноэкранное видео (768 на 576) с палитрой в 16,7 млн. цветов (24 бит) и полной кадровой разверткой (24 или 30 кадров/с).

Сжатие видео.

Следует исходить из разумной достаточности при определении необходимой степени сжатия. При этом необходимо учитывать, как четыре характеристики (частота кадра, экранное разрешение, глубина цвета и качество изображения) влияют на объем и качество видео. Вы должны ясно себе представлять, какую "цену" придется заплатить за качественное изображение. Чем больше глубина цвета, выше разрешение и лучше качество, тем большая производительность компьютера вам потребуется, не говоря уж о громадных объемах дискового пространства, необходимого под цифровое видео. Учитывая эти характеристики, можно выбрать оптимальный коэффициент сжатия. Надо отметить, что в профессиональном видео действует простое правило - чем ниже коэффициент сжатия, тем лучше. Современные приложения (игры, компьютерные тренажеры, видеокиоски и некоторые деловые пакеты) зачастую не требуют полноэкранного видео. Такие программы обычно используют видео в окне, и для них не требуется оцифровывать целый кадр

Сжатие обычное (в режиме реального времени). Термин real - time (реальное время) имеет много толкований. Применительно к сжатию данных используется его прямое значение, т. е. работа в реальном времени. Многие системы оцифровывают видео и одновременно сжимают его, иногда параллельно совершая и обратный процесс декомпрессии и воспроизведения. Для качественного выполнения этих операций требуются очень мощные специальные процессоры, поэтому большинство плат ввода/вывода видео для PC бытового класса не способны оперировать с полноэкранным видео и часто пропускают кадры. Недостаточная частота кадров является одной из основных проблем для видео на PC. При производительности ниже 24 кадров/с видео перестает быть плавным, что нарушает комфортность восприятия. К тому же, пропущенные кадры могут содержать необходимые данные по синхронизации звука и изображения.

Симметричное и асимметричное сжатие. Этот показатель связан с соотношением способов сжатия и декомпрессии видео. Симметричное сжатие предполагает возможность проиграть видеофрагмент с разрешением 640 на 480 при скорости в 30 кадров/с, если оцифровка и запись его выполнялась с теми же параметрами. Асимметричное сжатие - это процесс обработки одной секунды видео за значительно большее время. Степень асимметричности сжатия обычно задается в виде отношения. Так цифры 150:1 означают, что сжатие одной минуты видео занимает примерно 150 минут реального времени. Асимметричное сжатие обычно более удобно и эффективно для достижения качественного видео и оптимизации скорости его воспроизведения. К сожалению, при этом кодирование полнометражного ролика может занять слишком много времени, вот почему подобный процесс выполняют специализированные компании, куда отсылают исходный материал на кодирование (что увеличивает материальные и временные расходы на проект).

Сжатие с потерей или без потери качества. Как мы уже говорили, чем выше коэффициент сжатия, тем больше страдает качество видео. Все методы сжатия приводят к некоторой потере качества. Даже если это не заметно на глаз, всегда есть разница между исходным и сжатым материалом. Пока существует всего один алгоритм (разновидность Motion-JPEG для формата Kodak Photo CD), который выполняет сжатие без потерь, однако он оптимизирован только для фотоизображений и работает с коэффициентом 2:1.

Сжатие видеопотока и покадровое сжатие. Это, возможно, наиболее обсуждаемый сегодня вид сжатия. Покадровый метод подразумевает сжатие и хранение каждого видеокadra как отдельного изображения. Сжатие видеопотока основано на следующей идее: несмотря на то, что изображение все время претерпевает изменения, задний план в большинстве видеосцен остается постоянным - отличный повод для соответствующей обработки и сжатия изображения. Создается исходный кадр, каждый следующий сравнивается с предыдущим и последующим изображениями, а фиксируется лишь разница между ними. Этот метод позволяет существенно повысить коэффициент сжатия, практически сохранив при этом исходное качество. Однако в этом случае могут возникнуть трудности с покадровым монтажом видеоматериала, закодированного подобным образом.

Коэффициент сжатия. Этот показатель особенно важен для профессионалов, работающих с цифровым видео на компьютерах. Коэффициент сжатия - это цифровое выражение соотношения между объемом сжатого и исходного видеоматериала. Для примера, коэффициент 200:1 означает, что если принять объем полученного после компрессии ролика за единицу, то исходный оригинал занимал объем в 200 раз больший. Обычно, чем выше коэффициент сжатия, тем хуже качество видео. Но многое, конечно, зависит от используемого алгоритма. Для MPEG сейчас стандартом считается соотношение 200:1, при этом сохраняется неплохое

качество видео. Различные варианты Motion- JPEG работают с коэффициентами от 5:1 до 100:1, хотя даже при уровне в 20:1 уже трудно добиться нормального качества изображения. Кроме того, качество видео зависит не только от алгоритма сжатия (MPEG или Motion-JPEG), но и от параметров цифровой видеоплаты, конфигурации компьютера и даже от программного обеспечения (к этим вопросам мы вернемс чуть позже в сравнительном обзоре видеоплат).

AVI (Audio Video Interleave).

Разработанный фирмой Microsoft метод сжатия, записи и воспроизведения движущих изображений (Live Video) и звука на компьютере с использованием только программных средств. Файлы, созданные с использованием этого метода, имеют расширение AVI.

AVI может иметь или не иметь звуковые дорожки. При создании AVI файлов, включающих звуковое сопровождение, важным является правильная синхронизация звука с видеоизображением. Для этого используется технология чередования видеокадров и звука, которой, собственно, и определяется аббревиатура AVI (Audio Video Interleaved). Разные по типу видео и аудиоданные записываются в один файл на диске следующим образом: все информационные потоки разбиваются на множество равных частей (chunks) и затем записываются в один файл друг за другом по очереди

Microsoft Video for Windows использует четыре основных кодека для сжатия AVI файлов цифрового видео - Microsoft Video 1, RLE compression, Cinepak Codec by SuperMatch и Intel Indeo Video R3.2 (INDEO - INtel viDEO).

Кодек Microsoft Video 1 предназначен для сжатия реалистических видеофрагментов и рассчитан на разрешение цвета не более, чем 16 бит.

QuickTime.

QuickTime - стандарт, архитектура программного обеспечения, которая позволяет создавать, объединять и публиковать все типы цифровых мультимедиа данных. Используя QuickTime, программные приложения могут легко работать с широким кругом форматов файлов и кодеков. QuickTime включает три основополагающих элемента - QuickTime видео формат (Movie file format), слой абстрактного носителя (Media Abstraction Layer) и богатый набор встроенных медиа - служб. QuickTime видеоформат хорош тем, что он платформенно независим, открыт для расширения. В связи с названными преимуществами, поддерживается многими производителями. QuickTime видео также поддерживается на платформе Windows. QuickTime слой абстрактного носителя определяет исчерпывающий набор сервисных функций покрывающих все аспекты создания, редактирования и воспроизведения цифрового материала.

Среди них:

- синхронизация по времени;
- компрессия и декомпрессия аудио - и видеоданных;
- преобразование форматов, масштабирование, смешивание и транскодирование;
- аудио - и видеоэффекты и переходы;
- синхронизация чтения и записи;
- захват данных;
- импорт и экспорт данных.

QuickTime поддерживает широкий набор типов мультимедиа данных (видео, аудио, текст, временной код, музыкальный MIDI - интерфейс, спрайты, анимацию и др.).

Формат QuickTime имеет несколько отличных друг от друга кодеков - Video, Animation, Cinepak, Graphics, Photo-JPEG и другие. Наиболее качественными являются кодеки Cinepak и Video.

Далее речь пойдет об основах одного из наиболее продвинутых стандартов, разработанного международным комитетом Motion Pictures Expert Group (сокращенно - MPEG) и принятого в окончательной редакции ISO 11172-2 только в 1993г. Хотя MPEG-стандарт определяет правила кодирования и декодирования цифровых потоков как изображений, так и связанного с ними звука, в этом материале мы остановимся только на видео.

В отличие от AVI-формата, который обычно использует алгоритмы сжатия Motion -JPEG, MPEG - файлы занимают значительно меньше места. Здесь есть свои проблемы: во-первых MPEG, в отличие от AVI очень тяжело редактировать и до последнего времени не было соответствующих редакторов; во-вторых, цена систем для оцифровки MPEG несколько выше, чем на обычные платы ввода видео.

Обзор технологий и алгоритмов сжатия видео.

Векторная квантизация (Vector Quantization, VQ) . Компрессорами, использующими технологию VQ, являются Indeo 3.2 и Cinepac. Оба они применяют цветовую схему YUV. Процесс кодирования очень трудоемок и практически неосуществим без специального дополнительного оборудования. Процесс декодирования очень быстр. Имеют место блоковые искажения при высоких коэффициентах сжатия. Технологии, использующие алгоритмы БДКП, ДВП могут достигать более высоких уровней сжатия.

Основная идея векторной квантизации заключается в разбиении изображения на блоки (размером 4x4 пикселя в цветовой схеме YUV для компрессоров Indeo и Cinepac). Как правило, некоторые блоки оказываются похожими друг на друга. В этом случае компрессор

идентифицирует класс похожих блоков и заменяет их одним общим блоком. Кроме того, генерируется двоичная таблица (карта) таких общих блоков из самых коротких кодовых слов. VQ-декодер затем, используя таблицу, собирает изображение поблочно из общих блоков. Ясно, что данный способ кодирования с потерями качества, так как, строго говоря, схожесть блоков весьма относительна. Здесь допускается аппроксимация реальных блоков изображения к общему, их объединяющему. Процесс кодирования длителен и трудоемок, так как кодеру необходимо выявлять принадлежность каждого блока изображения к какому-нибудь общему блоку. Однако задача декодирования в этом случае сводится к задаче построения изображения по заданной карте из общих блоков и не занимает много аппаратных и временных ресурсов. Таблицу или карту также называют еще и кодовой книгой, А двоичные коды, входящие в нее, — кодовыми словами, соответственно. Наибольшее сжатие с использованием алгоритма VQ достигается путем уменьшения числа классов общих блоков, то есть предположением о схожести относительно большего числа блоков изображения, и, как следствие, уменьшением кодовой книги. По мере уменьшения размеров кодовой книги качество воспроизводимого видео ухудшается. В результате на изображении появляется искусственная “блочность”.

Дискретное Косинусное Преобразование (ДКП) См. JPEG.

Дискретное Wavelet - преобразование (DWT)

Компрессоры, использующие DWT (Discrete Wavelet Transform): Intel Indeo 5.x; Intel Indeo 4.x

Большинство как статических, так и динамических изображений, сжатых при помощи алгоритма DWT, не имеет характерной для алгоритма ДКП блочной структуры. Наблюдается относительно высокое качество изображений, сжатых с использованием DWT, превосходит качество изображений, сжатых при помощи ДКП, при тех же коэффициентах сжатия. DWT несколько размывает, закругляет острые контуры изображения (так называемый контурный шум или эффект Гиббса).

DWT - алгоритм основан на передаче сигнала, например изображения, через пару фильтров: низкочастотный и высокочастотный. Низкочастотный фильтр выдает грубую форму исходного сигнала. Высокочастотный фильтр выдает сигнал разности или дополнительной детализации. В свою очередь, результат на выходе высокочастотного фильтра (добавочный сигнал детализации) может быть подвергнут той же процедуре и так далее.

Простым примером DWT является DWT Хара:

Входной сигнал $x[n]$ есть множество выборок с индексом n . Низкочастотный фильтр Хара (Haar Low Pass Filter) есть арифметическое среднее двух удачных выборок:

$$g[n] = 1/2 * (x[n] + x[n+1])$$

Высокочастотный фильтр Хара (Haar High Pass Filter) есть средняя разность двух удачных выборок:

$$h[n] = 1/2 * (x[n+1] - x[n])$$

При этом:

$$x[n] = g[n] - h[n] \quad x[n+1] = g[n] + h[n]$$

Выходные последовательности $g[n]$ и $h[n]$ содержат избыточную информацию. Таким образом, что для воспроизведения исходного сигнала $x[n]$ достаточно взять только четные или только нечетные его выборки. Как правило, берутся четные выборки. Таким образом, исходный сигнал $x[n]$ получается только из: $g[0], g[2], g[4], \dots, h[0], h[2], h[4]$,

$$x[0] = g[0] - h[0]$$

$$x[1] = g[0] + h[0] \quad x[2] = g[2] - h[2] \quad x[3] = g[2] + h[2] \text{ и так далее...}$$

Выход низкочастотного фильтра, как уже отмечалось, представляет собой грубую аналогию исходного сигнала. Если исходным сигналом является изображение, то на выходе низкочастотного фильтра получится расплывчатое, размытое изображение с низким разрешением. Выход высокочастотного сигнала добавляет детали к изображению. В сочетании с выходом низкочастотного фильтра может быть воспроизведено, таким образом, исходное изображение. Грубая форма исходного сигнала (сигнал на выходе низкочастотного фильтра) иногда называют основным уровнем (base layer), а дополнительный сигнал детализации — уровнем улучшения (enhancement layer). Сигнал на выходе высокочастотного фильтра $h[n]$ может быть пропущен снова через пару фильтров, и процесс, таким образом, может быть повторен, пока не будет достигнута достаточная степень детализации исходного сигнала $x[n]$. Однако ясно, что никакого сжатия здесь не достигается. Преобразование попросту воспроизводит то же количество битов, которое было в исходном сигнале. Выходные значения называются коэффициентами преобразования, или коэффициентами wavelet-преобразования.

Преобразование Хара используется в основном в области сжатия изображений. Для других целей используются более сложные фильтры преобразований. Сжатие же достигается в основном за счет применения некоторой формы квантизации (скалярной или векторной) к добавочному сигналу детализации. Далее к полученным коэффициентам преобразования применяется техника вероятностного (энтропийного) кодирования.

Разность кадров.

Компрессорами, использующими технологию разницы кадров, являются: Cinepac

В целом может обеспечивать сжатие, лучшее, чем независимое сжатие отдельных кадров. Возникающие в ходе кодирования ошибки накапливаясь, требуют наличия дополнительного ключевого кадра.

Алгоритм разницы кадров использует то обстоятельство, что во многих видео изображение от кадра к кадру мало чем различается. По мере применения алгоритма векторной квантизации для кодирования каждого следующего кадра и получения при этом малых коэффициентов, которые трудно кодируются, в кадры постепенно вкрадывается ошибка. Это требует включения в видеоряд так называемых ключевых кадров, которые кодируются без учета предыдущих и являются так называемыми “опорными точками” в видео.

Компенсация движения.

Компрессорами, использующими технологию компенсации движения, являются: MPEG-1,2 и 4.

По сравнению с механизмом разницы кадров механизм компенсации движения позволяет достигать большей степени сжатия. Кодирование весьма трудоемко и требует специальной аппаратуры. Технология компенсации движения используется в таких международных стандартах сжатия цифрового видео, как: MPEG, H.261 и H.263. Наибольшее сжатие достигается в сценах с пониженным движением.

Компенсация движения основана на использовании ряда сложных алгоритмов. Сфера, где данная технология сжатия эффективна, как правило, сводится к видеоряду, в котором объект изменяет свое местоположение относительно неподвижного фона. Объекты, изменяющиеся по форме, приближающиеся или удаляющиеся (движущаяся камера), не подлежат эффективному сжатию посредством алгоритма компенсации движения. Сжатие возможно заданием вектора смещения элементов изображения вместо хранения больших значений новых координат данных элементов изображения. Основным блоком (относительно которого задается вектор смещения остальных блоков) может являться любой блок изображения размером 16x16 пикселей, максимально похожий на кодируемый (предсказываемый) блок. Ясно, что кадр, на который ссылаются таким образом другие кадры, должен быть декодирован ранее. Однако совсем не обязательно, чтобы опорный кадр предшествовал предсказываемому кадру. MPEG позволяет производить предсказание в обоих направлениях путем введения так называемых В - (bidirectionally predicted) кадров.

Алгоритм компрессии MPEG - 1.

Технология сжатия видео в MPEG решает следующие задачи:

Устранение временной избыточности видео (в пределах коротких интервалов времени большинство фрагментов сцены остаётся неподвижными).

Устранение пространственной избыточности изображения путём подавления мелких несущественных деталей.

Использование более низкого цветового разрешения YOV.

Повышение информационной плотности результирующего потока путём выбора оптимального математического кода для его описания.

На этапе предварительной обработки входной видеосигнал оцифровывается и форматируется согласно заданному размеру и цветовой выборке (например, для PAL 352*288 и 2:1:1). Изображение делится на макроблоки размером 16*16 пикселей.

Каждый макроблок состоит из шести блоков, четыре из которых несут информацию о яркости, а два – определяют цвет. Каждый блок имеет размер 8*8. Блоки являются основными единицами, над которыми осуществляется дискретное косинусоидальное преобразование и квантование.

I - кадры - независимо сжатые (I - Intrapictures). Являются опорными и кодируются независимо методом JPEG. Требуют для своего кодирования большого объема информации и обеспечивают большую точность восстановления. Стандарт не ограничивает алгоритма определения вектора смещения. Кадры I - типа сжаты описанным выше методом и не содержат ссылок на какие-либо другие кадры. Средняя степень сжатия составляет примерно 2 бита на точку. Подобный кадр является первым кадром последовательности, с него начинается новый видеоряд, при этом между двумя I - кадрами может находиться не более 12 кадров других типов. Обычно последовательность выглядит следующим образом:

I V R V V R V V R V V I V R V V R V

R - кадры - сжатые с использованием ссылки на одно изображение (R - Predicted). Макроблоки R – типа могут кодироваться по уже описанной схеме и предсказываемые, которые формируют разность, текущего макроблока и подобного ему предыдущего I или R блока. В основном, в R – блоках используется вторая схема.

B - кадры - сжатые с использованием ссылки на два изображения (B - Bidirection). Могут быть I, R и B – типа. Могут кодироваться с предсказанием вперед, назад, и интерполируемые, как полусумма обоих предсказаний. В данном случае кодеру предстоит наиболее сложная работа по выбору наиболее эффективного варианта, реализующего компромисс между длиной кода и качеством восстанавливаемого изображения. Положительным является тот факт, что при этом не происходит накапливания ошибок в отличие от предыдущих этапов.

BC - кадры - независимо сжатые с большой потерей качества (используются только при быстром поиске).

Каждый кадр, таким образом, кроме служебной информации включает в себя переменное число кодов

Кадры Р - типа, или предсказанные кадры, обычно следуют через два В - кадра.



Рис. 30. Кадры Р и В – типа.

Поскольку В - кадры ссылаются как назад, так и вперед по последовательности, то для оптимизации процесса воспроизведения кадры в потоке перераспределяются следующим образом: компенсация движения является методом увеличения степени сжатия Р - и В - кадров путем использования информации из предыдущих кадров. Этот алгоритм действует на уровне макроблоков и обеспечивает примерно трехкратное сжатие по сравнению с I - кадрами.



Рис. 31. Двунаправленное предсказание

Сжатый с его использованием кадр содержит дополнительно вектор перемещения макроблока, на который имеется ссылка и отличия между кодируемым макроблоком и макроблоком по ссылке.

Обычно не вся информация в кадре может быть эффективно описана методом компенсации движения. В подобном случае макроблок, не имеющий аналогов в предыдущих кадрах, кодируется так же, как и все макроблоки в I - кадре.

Для В - кадра компенсация движения может осуществляться как вперед, так и назад, поэтому для данного типа возможно четыре варианта кодирования макроблоков:



Рис. 32. Варианты кодирования макроблоков.

- кодирование без компенсации движения;
- компенсация с прямым предсказанием;
- компенсация с обратным предсказанием ;
- компенсация с двунаправленным предсказанием.

Обратное предсказание применяется тогда, когда нет возможности выполнить прямое.

Частота I - кадров выбирается в зависимости от требований на время произвольного доступа. Соотношение P - и B - кадров подбирается, исходя из требований к величине компрессии и ограничений декодера. Именно варьирование частоты кадров разных типов обеспечивает алгоритму необходимую гибкость и возможность расширения.

При выборе кодека для видео надо иметь ввиду следующее.

Видео вводится в компьютер для двух целей. Первая из них – компьютерный монтаж. Вторая – создание видеоролика в формате AVI или QuickTime для последующего использования в ролике мультимедиа. При этом рекомендуется использовать различные кодеки для сжатия информации. В случае монтажа следует пользоваться кодеками None, Component Video, Motion – JPEG, применение других кодеков может привести к ухудшению фрагмента. Для создания фильма для показа на компьютере следует использовать кодеки, которые ухудшают качество картинки, но при этом значительно уменьшают объём фильма. Необходимо также учитывать количество данных, которые надо прочитать за единицу времени. При выборе кодека надо иметь ввиду эту величину.

Cinepak. Один из старейших кодеков, появившихся на рынке. Первоначально был разработан фирмой SuperMatch. Однако, впоследствии был переработан другой фирмой - Radius. Свою популярность кодек получил благодаря использованию минимальных ресурсов процессора. Максимальный коэффициент компрессии составляет 10:1. Кодек выдает среднее качество видеоизображения из - за чрезмерной пикселеризации (изображение выглядит "шероховатым"). К недостаткам кодека также относится изменение насыщенности цвета в видеоизображении, что влечет за собой определенный визуальный эффект - изображение становится более желтым. Cinepak - наиболее асимметричный кодек. Кодирование видеоизображения в 8- и 24 - битном разрешении цвета. Алгоритм сжатия использует векторную квантизацию (vector quantization) и межкадровую разность. Плохо работает с видеоизображением, частота воспроизведения которого превышает 15 кадров в секунду. Кодек хорошо сжимает синтезированное динамическое видеоизображения - 2D и 3D анимацию. В случае черно - белого видеоизображения, кодек представляет возможность произвести сжатие в 8-битном режиме с 256 оттенками серого. В настоящее время корпорация Compression Technologies продает улучшенный кодек CinepakPro, в котором устранены основные проблемы которые встречались у стандартного Cinepak. CinepakPro полностью совместим с Cinepak.

VDOWave от VDONET. Основанный на преобразовании элементарной волны кодек видео. Microsoft лицензировал VDOWAVE как часть NetShow. Имеются две версии VDOWAVE кодер - декодера. VDOWave 2.0 - кодек фиксированной задержки пакетов в канале, который использует Microsoft, четырехсимвольный код VDOM.. VDOWave 3.0 - "масштабируемый" кодек видео. Этот кодек использует Microsoft четырехсимвольный код

(FOURCC) VDOW. В NetShow 2.0, автономная установка клиентов устанавливает VDOWAVE декодер. VDOWave использует комбинацию алгоритмов сжатия элементарной волны и компенсации движения.

Intel Indeo 3.1/3.2. Один из старейших кодеков, появившихся на рынке. Качество сжатого видеоизображения немного лучше, чем у Cinepak. Однако, кодек более требователен к ресурсам процессора. Indeo 3.1/3.2 представляет собой менее асимметричный кодек - время компрессии и декомпрессии видеоизображения примерно равны. Максимальный коэффициент компрессии составляет 15:1. Кодирование видеоизображения в 24-битном разрешении цвета. Алгоритм сжатия использует векторную квантизацию и межкадровую разность. Кодек наиболее подходит для сжатия изображения с "говорящими головами". Процесс сжатия видеоизображения на порядок быстрее, чем у Cinepak. Видеоизображение, сжатое при помощи кода Indeo R3.2, имеет цветовые артефакты - изображение расплывается и получает красно-синий оттенок.

Indeo 4.1. Indeo Video Interactive, Indeo 4.1, является следующей принципиально новой версией кода от Intel, основанной на гибридном алгоритме DWT. Indeo 4.1 поддерживает множество особенностей в дополнение к новому алгоритму сжатия, например прозрачность. При низкой скорости передачи данных при применении алгоритмов, использующих дискретное преобразование элементарной волны (DWT), проявляется размытие в гранях объектов и также артефакты "окружения" около граней, но нет блочных артефактов, явно видимых при ДКП.

Indeo Video Interactive 5.x. Indeo 5.1 использует новый улучшенный алгоритм DWT для увеличения качества видео. Indeo 5.x включает такие особенности как прогрессивная загрузка для Internet, прозрачность, спрайты, и т.д. Все выпуски Indeo 5.x, используют четырех-символьный код IV50.

ClearVideo (RealVideo). ClearVideo - кодек видео от Iterated Systems. ClearVideo использует рекурсивное сжатие. Рекурсивное кодирование видео достаточно медленное (требовательно к вычислительным ресурсам). Качество видео аналогично или несколько выше MPEG - 1.

SFM (Surface Fitting Method) Этот кодек рассчитан на использование при большой задержке пакетов в канале для ISDN и коммутируемого соединения. SFM использует алгоритм контур - основанного кодирования изображения.

QPEG. Q - group производит avi кодек известный как QPEG. На 6/27/97, QPEG поддерживал 8 разряда цвета. Известно, что Q-группа планирует поддержку 16 и 24 разряда цвета, MMX поддержку, и другие дополнительные особенности в будущем. Q-группа также работает над MPEG-4 для PC.

Н.261. Н.261 - рекомендация, разработанная и оптимизированная для передачи цветного видеоизображения по каналам данных системы ISDN со скоростями 64 Кбит/с и фиксированными значениями задержек пакетов в канале. В Н.261 реализована комбинация алгоритмов ДКП и компенсации движения движения В алгоритме применяется формат CIF (Common Intermediate format) с разрешением 352x288 пикселей для яркостного сигнала или QCIF (Quarter CIF) с четвертной частью разрешения CIF (то есть 176x144). Скорость кадровой развертки равна 29.97 кадра в секунду. Необходимо отметить, что при использовании формата CIF, цветоразностные сигналы передаются с разрешением 176x144 пикселя, каждый пиксель описывается восемью битами.

Н.263. Н.263 - стандарт, основанный на ДКП и компенсации движения. Н.263 имеет множество усовершенствований, главным образом в области компенсации движения, по сравнению с более ранним стандартом Н.261. Это позволяет добиться еще более высокого сжатия видео.

Кодек (30 кадров/с, 160x120,24 бита, 239 кадров(7,996 с))	Размер файла	Коэффициент сжатия/ требуемая пропу- сканая способ- ность/PSNR
Full Frames (Uncompressed)	13.1 MB	1 : 1 / 13.5 Mbps /---
Cinepak Codec by Radius Version 1.8.0.12 Качество сжатия 100, ключевой кадр каждые 15 кадров	0.85 MB	15.9 : 1 / 848 Kbps / 31.08 dB
Cinepak Codec by Radius Качество сжатия 85, ключевой кадр каж- дые 15 кадров	0.78 MB	17.7 : 1 / 780 Kbps / 30.49 dB
Cinepak Codec by Radius Качество сжатия 50, ключевой кадр каж- дые 15 кадров	0.42 MB	32.93 : 1 / 420 Kbps / 27.087 dB
Cinepak Codec by Radius Качество сжатия 25, ключевой кадр каж- дые 15 кадров	0.23 MB	59.6 : 1 / 231 Kbps / 23.762 dB

Intel Indeo 5.10 Качество сжатия 85, ключевой кадр каж- дые15 кадров	0.993 MB	13.2 : 1 / 992 Kbps / 32.43 dB
Intel Indeo 5.10 Качество сжатия 50, ключевой кадр каж- дые15 кадров	0.22 MB	59.6 : 1 / 216 Kbps / 29.6 dB
Intel Indeo 5.10 Качество сжатия 25, ключевой кадр каж- дые15 кадров	0.177 MB	74 : 1 / 176 Kbps / 28.4 dB
Intel Indeo 5.10 Качество сжатия 10, ключевой кадр каж- дые15 кадров	0.158 MB	82.9 : 1 / 152 Kbps / 27.82 dB
Indeo V 4.11.15.62 Качество сжатия 85, ключевой кадр каж- дые15 кадров	1.564 MB	8.8 : 1 / 1.564 Mbps / 28.686 dB
Indeo V 4.11.15.62 Качество сжатия 50, ключевой кадр каж- дые15 кадров	0.297 MB	46.5 : 1 / 297 Kbps / 26.622 dB
Indeo V 4.11.15.62 Качество сжатия 85, ключевой кадр каж- дые15 кадров	0.256 MB	53.9 : 1 / 256 Kbps / 25.389 dB
Microsoft Video 1 Качество сжатия 100, ключевой кадр каждые15 кадров	5.198 MB	2.7 : 1 / 5.198 Mbps / 32.209 dB
Microsoft Video 1 Качество сжатия 75, ключевой кадр каж- дые15 кадров	0.79 MB	17.5 : 1 / 790 Kbps / 30.286 dB
Microsoft Video 1 Качество сжатия 50, ключевой кадр каж- дые15 кадров	0.17 MB	82.8 : 1 / 166 Kbps / 23.915 dB
Microsoft Video 1 Качество сжатия 25, ключевой кадр каж-	0.08 MB	163.7 : 1 / 84 Kbps / 18. 524 dB

дые15 кадров		
--------------	--	--

Общее описание MPEG.

Слово MPEG является сокращением от Moving Picture Expert Group - названия экспертной группы ISO, действующая в направлении разработки стандартов кодирования и сжатия видео- и аудиоданных. Официальное название группы - ISO/IEC JTC1 SC29 WG11. Часто аббревиатуру MPEG используют для ссылки на стандарты, разработанные этой группой. На сегодняшний день известны следующие:

MPEG - 1 - предназначен для записи синхронизированных видеоизображения (обычно в формате SIF, 288*358) и звукового сопровождения на CD - ROM с учетом максимальной скорости считывания около 1.5 Мбит/с. Качественные параметры видеоданных, обработанных MPEG - 1, во многом аналогичны обычному VHS - видео, поэтому этот формат применяется в первую очередь там, где неудобно или непрактично использовать стандартные аналоговые видеоносители.

MPEG - 2 - предназначен для обработки видеоизображения соизмеримого по качеству с телевизионным при пропускной способности системы передачи данных в пределах от 3 до 15 Мбит/с, профессионалы используют и большие потоки. аппаратуре используются потоки до 50 Мбит/с. На технологии, основанные на MPEG - 2, переходят многие телеканалы, сигнал сжатый в соответствии с этим стандартом транслируется через телевизионные спутники, используется для архивации больших объёмов видеоматериала.

MPEG - 3 - предназначался для использования в системах телевидения высокой чёткости (high-defenition television, HDTV) со скоростью потока данных 20 - 40 Мбит/с., но позже стал частью стандарта MPEG - 2 и отдельно теперь не упоминается. Кстати, формат MP3, который иногда путают с MPEG - 3, предназначен только для сжатия аудиоинформации и полное название MP3 звучит как MPEG Audio Layer III.

MPEG - 4 - задает принципы работы с цифровым представлением медиа - данных для трех областей: интерактивного мультимедиа (включая продукты, распространяемые на оптических дисках и через Сеть), графических приложений (синтетического контента) и цифрового телевидения.

Области применения DVD.

DVD – стандарт, призванный обеспечить совместимость носителей и информационного обеспечения для бытовой и компьютерной техники. DVD - диск имеет множество приложе-

ний: в качестве носителя видеоизображений (DVD - Video), аудиозаписи (DVD - Audio), хранения данных (DVD - ROM), для записи информации (DVD - R) и т.д. Стандарты DVD - диска (физические особенности, файловая система, области использования) описываются в регламентирующих изданиях, полностью совместимых с "Yellow Book" (описание стандарта CD - ROM) и "Red Book" (описание стандарта Audio CD). Они делятся на следующие группы:

Book A Read only - Стандарт диска, предназначенного только для считывания

Book B Video - Стандарт диска для видеоинформации

Book C Audio - Стандарт диска для аудиоинформации

Book D Write-once - Стандарт диска для однократной записи

Book E Rewritable - Стандарт многократно перезаписываемого диска

Структура организации данных на диске.

В DVD - дисках (книги А, В, С) будет применяться единая файловая система как для компьютерных приложений, так и для видеопрограмм, названная UDF Bridge, которая является комбинацией систем Micro UDF и ISO - 9660. Структура представления данных UDF (Universal Disk Format), разработанная ассоциацией OSTA (Optical Storage Technology Association) на базе стандарта ISO / IEC 13346, определяет все параметры структур файла и диска, а также метода записи и считывания информации. Файловая система UDF - это очень гибкий, многоплатформенный и многопользовательский стандарт, который был адаптирован для применения в DVD. В частности, гарантирована совместимость со стандартом ISO-9660 для работы с существующим программным обеспечением. Окончательный вариант универсальной файловой структуры DVD еще не сформирован. Он будет зависеть от новых требований, предъявляемых компанией Microsoft к операционной системе. Поэтому на первом этапе домашние DVD - плееры будут использовать файловую структуру UDF, в то время как компьютерные DVD - устройства пока ограничатся стандартом ISO-9660, до тех пор, пока UDF не распространится повсеместно.

Таким образом, после полной реализации заложенных в UDF возможностей пользователь сможет одновременно хранить на одном DVD - диске видеофильмы, аудиозаписи, оцифрованные фотографии и свои компьютерные файлы. DVD - диск должен стать единым носителем и для Macintosh, и для DOS / Windows, и для OS/2, и для Unix.

Основные черты.

диаметр DVD - диска составляет 120 мм (4,75 дюйма), а толщина - 1,2 мм. DVD-диск может быть как односторонним, так и двухсторонним. Причем на каждой стороне DVD - диска может быть один или два рабочих слоя. Таким образом, различают следующие типы DVD - дисков:

DVD - 5 (Single-sided, single-layer disc) - Односторонний однослойный диск 4,7 Гбайт.

DVD - 9 (Single-sided, double-layer disc) - Односторонний двухслойный диск 8,5 Гбайт.

DVD - 10 (Double-sided, single-layer disc) - Двухсторонний однослойный диск 9,4 Гбайт.

DVD - 18 (Double-sided, double-layer disc) - Двухсторонний двухслойный диск 17 Гбайт.

Конструктивно двухсторонний DVD - диск выглядит как два склеенных между собой CD - диска толщиной 0,6 мм.

Метод записи	Цифровой (MPEG - 2)			
Ёмкость	1 сторона – 1 слой, 4.7 ГБ	1 сторона –2 слоя, 8.5 ГБ	2 стороны – 1 слой, 9.4 ГБ	2 сторона – 2 слоя, 17 ГБ
Время воспроизведения	133 мин.	242 мин.	266 мин.	484 мин.
Диаметр диска	12 см. / 8 мин			
Горизонтальное разрешение	500 и более линий			
Аудио	Dolby Digital / Линейное PCM			
Качество звука	48 кГц, 99 кГц / 16, 20, 24 бит.			

Такие средства можно условно разделить на три группы:

- специализированные программы, предназначенные для быстрой подготовки определенных типов мультимедийных приложений (презентаций, публикаций в Internet);
- авторские средства разработки (специализированные инструментальные средства для создания мультимедийных приложений);
- языки программирования.

Наша цель - познакомить вас с авторскими средствами разработки, о которых написано незаслуженно мало, хотя в последнее время ситуация начала меняться.

Провести четкую границу между указанными группами практически очень сложно. К примеру, одна из лучших презентационных программ Astound обладает некоторыми чертами авторского средства разработки, многие авторские средства позволяют распространять созданные с их помощью приложения через Internet и т. д.

По большому счету есть два основных способа создать мультимедийное приложение: использовать специализированные средства разработки или заняться программированием напрямую. Когда речь идет о презентациях, второй способ бессмыслен, в остальных случаях возможны варианты. Первый способ дает экономию средств и времени, но мы проигрываем в эффективности работы программы. Это плата за скорость разработки. Непосредственное программирование - более дорогое удовольствие, но и некоторые авторские программы недешевы. Кроме того, вы сталкиваетесь с необходимостью овладения специальными приемами для работы с ними и целым рядом ограничений, хотя и тут можно найти выход из положения. Оптимальным был бы путь посередине - применение готовых пакетов с расширением их функций при помощи языков программирования, но он, к сожалению, не всегда осуществим.

Очевидно, что задача выбора необходимого средства создания мультимедиа-приложения не так проста, как кажется на первый взгляд, и универсального решения, годного на все случаи жизни, не имеет. Поэтому весьма важен в процессе разработки именно сам этап выбора, поскольку если вы ошибетесь, то время и деньги могут быть потеряны напрасно, причем иногда это невозможные потери. Мы попытаемся помочь вам осмысленно сделать первый шаг по верному пути. Всякие рекомендации - вещь субъективная: их стоит принимать во внимание, но не нужно следовать им буквально, ведь существует множество нюансов, которые довольно трудно учесть. Так что конечный выбор - за вами, но, естественно, он должен быть разумным.

Итак, наиболее простым способом разработки мультимедийных приложений является использование современных программ для создания презентаций. С них и начнем.

Мультимедиа - быстро и просто

Современные программы создания презентаций все больше ориентируются именно на мультимедиа. Наиболее интересным примером может служить программа PowerPoint 97 фирмы Microsoft. По количеству изобразительных и анимационных эффектов она становится вровень со многими авторскими инструментальными средствами мультимедиа. Наличие сценария без возможности выбора отличало раньше программы для разработки презентации от авторских систем. Но теперь и с этим покончено. В PowerPoint 97 презентация не должна от начала до конца следовать жесткому сценарию - он может свободно разветвляться в зависимости от реакции пользователя.

Программа PowerPoint 97 позволяет создавать сложные программные надстройки за счет использования Visual Basic. Встроенная поддержка Internet и другие разнообразные усовершенствования сделали эту программу лидером в мире мультимедийных презентаций, а наличие русскоязычной версии позволило решить все проблемы, связанные с применением англоязычного интерфейса.

Среди других презентационных программ необходимо отметить Macromedia Action!, Gold Disk Astound и Asymetrix Compel. О них достаточно подробно рассказывалось на страницах "Мира ПК". Мы же обратим взор к авторским средствам разработки.

Авторские средства разработки и их классификация

Авторское средство разработки (авторская система) представляет собой программу, которая имеет предварительно подготовленные элементы для разработки интерактивного программного обеспечения. Такие системы различаются по своей специализации, возможностям и легкости освоения. В настоящее время не существует автоматизированной авторской системы, позволяющей полностью построить приложение только по принципу "укажи и щелкни", хотя современные средства подходят к этому достаточно близко.

Применение авторской системы - это фактически ускоренная форма программирования: вы не обязаны вникать в тонкости языка или, хуже того, в детали функционирования Windows API (Application Programming Interface - интерфейс прикладных программ), но должны понимать, как программы работают. Вместе с тем не надо пугаться слова "программирование". Многие системы имеют довольно дружелюбный пользовательский интерфейс, а для осуществления простейших проектов можно вообще обойтись без этого процесса.

В общем случае для разработки интерактивного мультимедийного проекта в авторской системе требуется значительно меньше времени, чем при использовании средств чистого программирования. Это означает снижение стоимости работ в несколько раз. Однако на создание компонентов мультимедиа (графика, текст, видео, звук, мультипликация и т. д.) выбор авторской системы вообще не влияет; выигрыш во времени при подготовке конечного про-

дукта в этом случае получается за счет ускоренного построения прототипа, а не из-за выбора авторской системы вместо какого-нибудь языка программирования.

Что касается классификации авторских систем, то в этом направлении уже предпринято достаточно много попыток. В их основе лежит так называемая авторская метафора - методология, в соответствии с которой авторская система выполняет свои задачи. Хотелось бы подчеркнуть, что:

- границы между различными метафорами довольно размытые;
- некоторые авторские системы имеют черты нескольких метафор;
- классификация авторских систем по метафорам не является достаточно точной.

Классификация, предложенная Джеми Сигларом, представляется сегодня наиболее полной. Ее мы и примем за основу в нашем дальнейшем путешествии в мир авторских систем.

Согласно этой классификации можно выделить восемь типов авторских систем, использующих следующие метафоры:

- язык сценариев (Scripting Language);
- изобразительное управление потоком данных (Icon/Flow Control);
- кадр (Frame);
- карточку с языком сценариев (Card/Scripting);
- временную шкалу (Timeline);
- иерархические объекты (Hierarchical Object);
- гипермедиа-ссылки (Hypermedia Linkage);
- маркеры (Tagging).

Заметим, что классификация сама по себе не является самоцелью. Это лишь средство для обоснованного выбора необходимого инструмента в соответствии со спецификой вашего мультимедийного проекта и его бюджета. Рассмотрим типы авторских систем более подробно.

Язык сценариев

Авторский метод "Язык сценариев" наиболее близок по форме к традиционному программированию. Этот мощный, объектно-ориентированный язык программирования определяет (с помощью специальных операторов) взаимодействие элементов мультимедиа, расположение активных зон, назначение кнопок, синхронизацию и т. д. Он является обычно центральной частью такой системы; редактирование элементов мультимедиа внутри программы (графических изображений, видео, звука и т. д.) представлено либо в минимальном виде, либо отсутствует вообще. Языки сценариев изменяются. При выборе системы обратите внимание на то, в какой степени язык основан на объектах или объектно-ориентирован. Использование этого метода несколько увеличивает период разработки (требуется дополнительное время на

индивидуальное изучение возможностей системы), но в результате можно получить более мощное взаимодействие элементов. Так как многие языки сценариев - интерпретирующие, подобные системы имеют довольно низкое быстродействие по сравнению с другими авторскими средствами.

К системам, основанным на языке сценариев, относятся:

- Grasp (фирмы Paul Mace Software), DOS;
- Tempra Media Author (фирмы Mathematica), DOS;
- Ten Core Language (фирмы Computer Teaching), DOS, Windows;
- Media View (фирмы Microsoft), Windows.

Примером мультимедийного приложения, выполненного с использованием системы Grasp, может служить CD-ROM Space Shuttle. Он был разработан фирмами Amazing Media и Follett Software в 1993 г. и поставлялся с торговой маркой The Software Toolworks (в том числе и в наши магазины). Этот компакт-диск представляет собой энциклопедию по американской космической программе Space Shuttle с кратким описанием истории проекта, процесса подготовки астронавтов и 53 конкретных полетов. Здесь широко используются звуковые комментарии к неподвижным изображениям и оцифрованное видео, правда, не всегда достаточно хорошего качества. Приложение запускается в системе DOS непосредственно с CD-ROM.

Изобразительное управление потоком данных

Этот авторский метод обеспечивает минимальное время разработки; лучше всего он подходит для быстрого создания прототипа проекта или выполнения задач, которые необходимо завершить в кратчайшие сроки. Его основа - палитра пиктограмм (Icon Palette), содержащая всевозможные функции взаимодействия элементов программы, и направляющая линия (Flow Line), которая показывает фактические связи между пиктограммами. Авторские системы, построенные на базе этого метода, имеют самые медленные исполняемые модули, потому что каждое взаимодействие влечет за собой всяческие перестановки. Однако наиболее развитые пакеты, такие как Authorware или IconAuthor, являются чрезвычайно мощными и обладают большим потенциалом.

Главное достоинство рассматриваемого метода состоит в том, что он позволяет ускорить работу над дизайном приложения. Вы перемещаете пиктограммы из палитры на бланк страницы, и получающийся документ становится проектом вашего приложения. Далее нужно дважды щелкнуть на пиктограммах, и появившиеся диалоговые окна будут ждать от вас команд для связывания составляющих в единое целое и формирования диалога с пользователем.

Применение авторских систем этого типа - наиболее подходящий путь для построения мультимедийных приложений со сложными функциями взаимодействия, подобных про-

граммам машинного обучения и мультимедийным киоскам. Такие авторские системы могут стоить очень дорого - до нескольких тысяч долларов. Чем же обусловлена столь высокая цена? Дело в том, что разработчики продают вам не только программное обеспечение, но и право на распространение созданных с его помощью приложений большим тиражом.

По легкости освоения эти программы занимают промежуточное положение между авторскими системами на базе метафоры "карточка с языком сценариев" и системами, основанными на временной шкале.

Конечно, здесь имеется значительное число функций и переменных, которые требуют изучения. Однако если вы уже создали с помощью палитры пиктограмм свое приложение, то нет ничего проще, чем формировать новые приложения на базе ваших шаблонов.

К системам, основанным на изобразительном управлении потоком данных, относятся:

- Authorware (фирмы Macromedia), Windows, MacOS;
- IconAuthor (фирмы Aim Tech), Windows, Unix, OS/2;
- TIE (фирмы Global Information Systems), Windows, Unix.

В качестве примера мультимедийного приложения, выполненного с использованием авторской системы Authorware, можно привести CD-ROM "ABBA ностальгия". Диск разработан российской фирмой ВЕКС в 1996 г. Это интерактивная энциклопедия творчества известной шведской группы ABBA на русском языке. Она содержит почти 100 страниц текста, более 400 фотографий, записи (6,5 часа музыки, включая все хиты!), в том числе редкие и ранее не публиковавшиеся в России, биографии и описания творческого пути солистов ансамбля в период с 1969 по 1995 г. Использование мощного авторского инструмента позволило выполнить проект силами небольшого коллектива разработчиков. Приложение запускается в системе Windows непосредственно с CD-ROM.

Кадр

Метод "Кадр" подобен методу изобразительного управления потоком данных. В него тоже обычно включается палитра пиктограмм (Icon Palette); однако связи, прорисованные между пиктограммами, могут представлять собой сложные ветвящиеся алгоритмы. Авторские системы, построенные по этому методу, - очень быстрые, но требуют применения хорошего автоматического отладчика, поскольку ошибки визуальны неувимы. Самые лучшие программы такого рода, например Quest, позволяют связать компилируемый язык с языком сценариев (при создании приложения в качестве языка сценариев используются Си или Apple Media Kit).

К системам, основанным на кадре, относятся:

- Quest (фирмы Allen Communication), Windows;
- Apple Media Kit (фирмы Apple), MacOS;

- Ten Core Producer (фирмы Computer Teaching), DOS, Windows;
- CBT Express (фирмы Aim Tech), Windows, Unix, OS/2.

Карточка с языком сценариев

Это весьма мощный по своим возможностям (через включенный язык сценариев) метод, требующий, однако, точной и жесткой структуризации сюжета. Он превосходно подходит для гипертекстовых приложений и особенно для прикладных программ с интенсивным перемещением (наиболее яркий пример - известная игра Myst, разработанная в авторской системе HyperCard).

Возможности программ этого типа легко расширяемы с помощью модулей XCMD и DLL. Такие системы часто используются для разработки прикладных программ общего назначения, а их лучшие представители позволяют все объекты (включая индивидуальные графические элементы) подготавливать внутри авторской системы. Многие развлекательные и игровые программы проходят этап создания прототипа по данному методу до кодирования на компилирующем языке программирования.

Одно из достоинств - наиболее легкий процесс обучения. Системы поставляются с множеством шаблонов, примеров и готовых графических элементов пользовательского интерфейса, а также с интерактивными учебными программами. Благодаря этому освоение происходит достаточно быстро.

Программы Astound и Compel, занимающие промежуточное положение между программами создания презентаций и авторскими системами, тоже иногда относят к этому типу авторских систем. Очень простые в освоении, они позволяют разрабатывать довольно интересные приложения.

Главный недостаток авторских систем на основе карточки с языком сценариев - невозможность обеспечить точное управление синхронизацией и выполнение параллельных процессов. К примеру, звуковой файл должен запускаться и заканчиваться прежде, чем сможет начаться следующее событие по сценарию.

Наилучшее применение для этих авторских систем - подготовка приложений, которые можно логически организовать в виде отдельных карточек с гипертекстовыми связями между ними.

К системам, основанным на карточке с языком сценариев, относятся:

- HyperCard (фирмы Apple Computer), MacOS;
- SuperCard (фирмы Allegiant Technologies), MacOS;
- Multimedia ToolBook (фирмы Asymetrix), Windows.

Примером мультимедиа-приложения, созданного с использованием авторской системы Toolbook, может служить CD-ROM "Английский на каждый день", разработанный россий-

ской фирмой New Media Generation в 1996 г. Специфика интенсивного курса изучения английского языка по методу Т.А. Графовой хорошо сочеталась с данной метафорой. Разработка была выполнена на достаточно высоком уровне и отмечена за отличный дизайн. Приложение функционирует в среде Windows и требует установки отдельных файлов на жесткий диск для ускорения работы.

Временная шкала

По структуре пользовательского интерфейса авторская система на основе метода "Временная шкала" напоминает звуковой редактор для многоканальной записи. Синхронизируемые элементы показываются в различных горизонтальных "дорожках" с рабочими связями, отраженными через вертикальные столбцы. Основными элементами данного метода являются "группа" (cast) - база данных объектов и партитура (score) - покадровый график событий, происходящих с этими объектами. Главное достоинство метода заключается в том, что он позволяет написать сценарий поведения для любого объекта. Каждое появление объекта из группы в одном из каналов партитуры называется спрайтом (sprite) и также считается самостоятельным объектом. Для управления спрайтами в зависимости от действий пользователя в пакет встраивается объектно-событийный язык сценариев (Scripting language). Подобные системы используются при создании многих коммерческих прикладных программ.

Авторские системы на базе временной шкалы лучше всего подходят для подготовки приложений с интенсивным использованием мультипликации или таких, где требуется синхронизация различных мультимедийных составляющих. Эти системы легко расширяются с целью обработки других функций (таких как гипертекст) через модули типа XOBJ, XCMD и DLL. Их основной недостаток - сложность освоения из-за необходимости изучения достаточно мощного языка сценариев.

К системам, основанным на временной шкале, относятся:

- Director (фирмы Macromedia), Windows, MacOS;
- Power Media (фирмы RAD Technologies), Windows, MacOS, Unix;
- MediaMogul (фирмы Optimage), для платформы CD-i.

Наиболее известная система, построенная по данному методу, является и самой популярной авторской системой мультимедиа вообще. Это Macromedia Director. С ее помощью разрабатываются достаточно сложные коммерческие приложения и даже компьютерные игры. Как пример можно привести "Frankenstein. Through the eyes of the monster" - довольно сложную приключенческую игру, по своему построению сходную с Myst. Игрок выступает в роли монстра, созданного доктором Виктором Франкенштейном. Цель игры - путешествуя по замку доктора и его окрестностям, раскрывать страшные тайны и разгадывать многочисленные загадки.

Иерархические объекты

Здесь, как и в объектно-ориентированном программировании, применяется метафора объекта. Хотя научиться работать с этими средствами разработки непросто, благодаря визуальному представлению объектов и информационных составляющих мультимедийного проекта можно создавать достаточно сложные конструкции с развитым сюжетом. Типичным представителем такого рода средств является mTropolis - одна из наиболее перспективных авторских систем. Подобные системы обычно довольно дорогие и используются в основном профессиональными разработчиками мультимедийных приложений.

К системам, основанным на иерархических объектах, относятся:

- mTropolis (фирмы mFactory), Mac;
- New Media Studio (фирмы Sybase), Unix, Windows (только 95 или NT);
- Fire Walker (фирмы Silicon Graphic Studio), для платформы SGI.

Гипермедиа-ссылки

Метафора гипермедиа-ссылки подобна метафоре кадра, в которой показываются концептуальные связи между элементами; однако ей недостает визуального представления связей. Авторские системы, построенные по этому методу, весьма просты в освоении, хотя для эффективной работы с ними требуется обучение.

При использовании авторских систем с гипермедиа-ссылками можно создавать разнообразные гипертекстовые приложения с элементами мультимедиа. Они имеют те же области применения, что и системы, построенные по методу "Карточка с языком сценариев", но более гибки (за счет отказа от карточек).

К системам, основанным на гипермедиа-ссылках, относятся:

- HyperMethod (фирмы Prog. Systems AI Lab), DOS, Windows;
- Formula Graphic (фирмы Harrow Media), Windows;
- HM-card, Windows;
- Everest (фирмы Intersystem Concepts), Windows.

Авторская система HyperMethod уже знакома читателям (см. "Мир ПК", #11/97). Она применяется для разработки самых разнообразных мультимедийных приложений. В частности, с ее помощью подготовлена энциклопедия на CD-ROM "Русский музей. Живопись".

Маркеры (теги)

Системы на базе маркеров используют специальные команды - теги в текстовых файлах (например, SGML/HTML и WinHelp), чтобы связать страницы для обеспечения взаимодействия и объединения элементов мультимедиа. Они имеют, как правило, ограниченные возможности по отслеживанию связей и лучше всего подходят для подготовки диалоговых справочных материалов, подобных словарям и руководствам. С развитием Internet такие системы

нашли широкое применение и при создании страниц для узлов этой глобальной компьютерной сети.

К системам, основанным на маркерах, относятся:

- Hot Dog (фирмы Sausage Software), Windows;
- WebAuthor (фирмы Quarterdeck), Windows;
- FrontPage (фирмы Vermeer), Windows, MacOS;
- HoTMetaLPro (фирмы SoftQuad), Windows, MacOS, Unix;
- Adobe PageMill (фирмы Adobe), MacOS;
- Arachnophilia, Windows.

Число редакторов, предназначенных для создания HTML-страниц, стремительно растет день ото дня. Они распространяются на коммерческой основе или как условно-бесплатное программное обеспечение, немало и бесплатных программ. Причем качество программы совсем не обязательно определяется тем, к какой стоимостной категории она относится.

Конечно, мир авторских систем не ограничивается перечисленными выше программами. Достаточно полные списки, представленные в Internet, насчитывают порядка 70 таких систем, и число их (в которое не включены программы создания презентаций и разнообразные HTML-редакторы) постоянно увеличивается. Но для человека, делающего свой выбор, нужно начинать со знакомства с лучшими из них.

Использование языков программирования

Как мы уже подчеркивали, в сравнении с авторскими средствами разработки универсальные языки программирования оказываются более гибкими и обеспечивают возможность получения более быстродействующего приложения. Но лучшие представители мира авторских систем довольно успешно пытаются преодолеть все препятствия. В современных условиях гибкость и быстрота работы иногда отходят на второй план, уступая место высокой скорости разработки. Этим и объясняется возросший интерес к таким системам со стороны разработчиков. В России распространение авторских систем сдерживается непомерными ценами на них, да и вообще приобрести их довольно сложно. Кроме того, для многих пользователей, особенно непрофессионалов в компьютерной технике, англоязычный интерфейс системы может перечеркнуть все ее достоинства. Но вернемся к программированию.

Если спросить у профессиональных российских разработчиков мультимедийных приложений, какие средства они используют, то ответ будет однозначным - языки программирования, причем чаще всего Си++, Delphi, реже Visual Basic. Немногочисленные авторские системы применяются лишь в единичных случаях. Но ситуация постепенно меняется. Все больше авторских инструментов появляется у нас на легальных условиях, и их уже можно

купить. Но вот стоит ли? Вопрос, конечно, интересный, и об этом имеет смысл поговорить подробнее.

Правильный выбор инструмента

Создание мультимедийного приложения начинается вовсе не с выбора необходимого средства разработки. Прежде всего нужно определить, какую информацию и каким образом вы собираетесь использовать. И только после этого можно переходить к выбору инструмента, который позволит вам наиболее полно выразить свои идеи.

Предположим, что вопрос о том, что за приложение вы хотите создать, уже решен и наступил момент отбора необходимых средств реализации проекта. Руководствуясь приведенными выше рекомендациями, попытайтесь найти наиболее подходящий для вашей задачи тип авторской системы. Подберите программы, относящиеся к нужному вам типу.

На выбор авторской системы влияют следующие факторы:

- тип платформы разработки;
- цена (включая лицензионные отчисления за распространение разработанных приложений);
- расширяемость (работа с DLL или XCMD);
- подход к программированию;
- наличие инструментов отладки и тестирования приложений;
- возможности форматирования текста и печати;
- интерактивные возможности;
- возможность управления внешними устройствами;
- поддержка OLE;
- возможности встроенного редактора компонентов мультимедиа;
- наличие средств организации проекта;
- поддержка баз данных;
- контроль над синхронизацией воспроизведения элементов мультимедиа;
- техническая поддержка;
- наличие обучающей программы;
- качество печатной документации;
- поддержка по "горячей" телефонной линии.

Постарайтесь узнать о предварительно выбранных вами программах побольше. Зарубежные специалисты в области мультимедиа советуют попросить для этого демонстрационные версии у разработчиков. Например, демонстрационные диски с системами Director, Authorware и Icon Author высылаются бесплатно или за минимальную плату. Поработав с такими версиями, вы лучше уясните возможности и ограничения систем. Дополнительную ин-

формацию по многим из них можно получить через Internet, обратившись на узлы фирм-разработчиков. В этом поиске вам должна помочь приведенная здесь таблица.

В России наибольшей популярностью пользуются персональные компьютеры с операционными системами DOS и Windows. Именно для этой платформы ниже даются рекомендации по выбору авторской системы с учетом возможности приобретения тех или иных программных продуктов у нас в стране.

Web-приложения

Если вы не хотите стать профессиональным Web-мастером и создаете HTML-страницы от случая к случаю, то наилучшим выбором для вас будет текстовый редактор Word 97 (и никаких проблем с языком!).

Более качественную работу обеспечивают бесплатные или условно-бесплатные HTML-редакторы, которые можно найти в Internet. В частности, попробуйте программу Arachnophilia, свободно распространяемую самим разработчиком Полем Латусом. Об этой программе стоит сказать пару слов.

Во-первых, у нее не возникает трудностей при создании русских HTML-страниц, что является большой редкостью даже для коммерческих редакторов. Во-вторых, она удобна для любого пользователя: от новичка до профессионала. К примеру, у известной программы FrontPage есть нехорошая привычка исправлять набранный вами исходный текст, даже если он полностью соответствует спецификации HTML, что ограничивает ваши творческие возможности. Автор Arachnophilia предлагает простой способ борьбы с этим злом, и поэтому совместное использование этих двух редакторов позволит снять подобные ограничения. Учтите и тот факт, что облегченная версия Front Page с набором шаблонов под названием FrontPad поставляется вместе с пакетом Internet Explorer 4.0, распространяющимся бесплатно.

Презентации

Тем, кто создает презентации не часто, можно рекомендовать русскоязычную версию программы PowerPoint 97. Она входит в состав Office 97. Стоит обратить внимание и на многообещающую новинку "Клуба голосовых технологий" - презентационную программу "Говорящая мышь для дома" со встроенным синтезатором речи. Все тексты будут читаться с правильным произношением, причем вам предоставляется возможность настроить его в соответствии со своими запросами. Единственным недостатком этих двух программ является их ориентация на Windows 95. Для пользователей Windows 3.1 хорошим выбором будет система Macromedia Action! Если вы сочтете необходимым приобрести пакет Macromedia Director, то разработанные в нем элементы сможете использовать в Action! без всякого преобразования.

Прототипы приложения

Для разработки прототипов лучше всего подойдут авторские системы, основанные на интерактивном управлении потоком данных или использующие карточку с языком сценариев. Пожалуй, возможностей демонстрационных версий Authorware или Icon Author будет достаточно для быстрого создания прототипа вашего приложения. Полные версии пока еще очень дороги, тем более для непрофессионалов. Хорошим выбором окажется и Multimedia Toolbook. Эта система вам так понравится, что вы, быть может, пожелаете и окончательный вариант выполнить с ее помощью.

Интерактивные программы

В данном случае имеет смысл применить Macromedia Director. Но помните, что для освоения этой программы вам придется приложить некоторые усилия. Если вы не хотите тратить деньги на приобретение авторской системы, остановите свой выбор на MediaView, HM-Card или Formula Graphics.

MediaView - название новой версии комплекта Multimedia Viewer Publishing Toolkit. Ранее он стоил 695.00 долл. Теперь распространяется бесплатно. По сравнению с предыдущей версией эта программа стала более трудной для изучения и освоения (вы должны хоть немного уметь программировать на Visual Basic), но зато и более мощной.

HM-Card - условно-бесплатная программа, по своим возможностям близкая к HyperMethod. По заявлению разработчиков, для создания приложения с ее помощью знать язык программирования вам не требуется.

Formula Graphics - бесплатная программа, которая обеспечивает быстрый и простой путь для реализации мультимедийных проектов в среде Windows. Она имеет свои плюсы и минусы, но факт ее бесплатного распространения многое решает. Зарегистрируйте вашу версию у разработчиков, и вы получите доступ к дополнительным средствам пакета.

Обучающие программы

Для создания обучающих программ некоторые фирмы выпускают отдельные версии своих основных продуктов. Например, существует версия Multimedia Toolbook СВТ со специальными шаблонами для разработки таких программ. Если вы в процессе обучения собираетесь применять мультипликацию, то можете остановиться на Macromedia Director.

Гипертекстовые приложения

Здесь предпочтение следует отдать системам на основе гипермедиа-ссылок (HM-Card или любая программа этого типа) и карточек с языком сценариев (Multimedia Toolbook). Обратите внимание на программу HyperMethod. Невысокая цена, простота изучения базовых возможностей (так что первое приложение вы сможете создать без программирования), быстрота расстановки гиперсвязей с учетом падежей, поддержка мультимедийных функций и совместимость с HTML - вот далеко не полный перечень ее отличительных особенностей. По

примеру своих зарубежных коллег российские разработчики выпустили демонстрационную версию и учебные материалы. И быть может, эта программа станет вашим лучшим помощником при построении мультимедийных приложений

Контрольные вопросы.

1. Гипертекст.

[Ответ.](#)

2. Способ формирования шрифтов TrueType.

[Ответ.](#)

3. Понятие о вёрстке.

[Ответ.](#)

4. Векторная графика. Структура изображения.

[Ответ.](#)

5. Векторная графика. Удаление невидимых линий, Z – буфер.

[Ответ.](#)

6. Растровая графика. Сжатие методом JPEG.

[Ответ.](#)

7. Растровая графика. Эффекты. Акварелизация.

[Ответ.](#)

8. Звук. MP – 3. Основные этапы сжатия.

[Ответ.](#)

9. Звук. 3D – звук.

[Ответ.](#)

10. Видео. Видеостандарты.

[Ответ.](#)

11. Видео. DVD.

[Ответ.](#)

12. Видео. Сжатие MPEG. Основные этапы.

[Ответ.](#)

13. Авторские системы. Метод временной шкалы.

[Ответ.](#)

Литература.

1. Роджерс Д. Алгоритмические основы машинной графики. Пер. с англ. М.: Мир, 1989. 512 с.
2. Павлидис Т. Алгоритмы машинной графики и обработки изображений. Пер. с англ. М.: Радио и связь, 1986.
3. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики: В 2 - х книгах. Пер. с англ. М.: Мир, 1985.
4. Гилой В. Интерактивная машинная графика. Пер. с англ. М. : Мир, 1981.
5. Шикин Е.В., Боресков А.В. Компьютерная графика. Динамика, реалистические изображения. М.: "ДИАЛОГ - МИФИ", 1995. 228 с.
6. Капелев В. В. Основы технологии наборных процессов: Учебное пособие.- М.: Изд – во МГАП “Мир книги”, 1997. - 288 с.
7. Кирьянов Д. В., Кирьянова Е. Н. Самоучитель Adobe Premier 6.5. - СПб.: БХВ - Петербург, 2004. - 480 с.: ил.
8. Пэрент Р. Компьютерная анимация: Пер. с англ. - М.: КУДИЦ - ОБРАЗ, 2004.
9. Божко А. Н., Жук Д. М., Маничев В. Б. Компьютерная графика: Учеб. пособие для вузов.- Изд - во МГТУ им. Н. Э. Баумана, 2007.- 392 с.: ил.- (Информатика в техническом университете).
10. Кречман Д. Л., Пушков А. И. Мультимедиа своими руками. – СПб.: БХВ - Петербург, 1999. – 528 с.: ил.
11. Sound Forge 8. Звуковая студия. Фирменное руководство от Sony [пер. с англ.]Scott R. Garrigus.- Москва: Изд - во Триумф: Thomson Course Technology PTR.- 480 с. ил.
12. Вельтмандер П.В. Введение в машинную графику: Учеб. пособие/ Новосиб. Ун - т. Новосибирск, 1995. 77 с.
13. Цифровая обработка изображений в информационных системах/ Грузман И.С. [и др.]: Учебное пособие. - Новосибирск: Изд - во НГТУ, 2000. - 168.