

А. В. КРАПИВЕНКО

МЕТОДЫ И СРЕДСТВА ОБРАБОТКИ АУДИО- И ВИДЕОДАННЫХ

Учебное пособие



Москва
«Вузовская книга»
2010

ББК 32.97
K78

Утверждено на заседании кафедры вычислительной математики и программирования факультета прикладной математики и физики Московского авиационного института (Государственного технического университета) в качестве учебного пособия по спецкурсу «Методы, средства и технологии Multimedia» для студентов, обучающихся по специальности 010501 «Прикладная математика и информатика»

Р е ц е н з е н т ы:
д-р техн. наук А. Д. Тузов;
канд. физ.-мат. наук С. В. Станченко

Крапивенко А. В.
K78 Методы и средства обработки аудио- и видеоданных: учеб. пособие / А. В. Крапивенко. — М.: Вузовская книга, 2010. — 140 с.: ил.

ISBN 978-5-9502-0415-9

Пособие написано на базе конспектов лекций цикла учебных дисциплин «Методы, средства и технологии Multimedia» и является продолжением учебного пособия «Технологии мультимедиа и восприятие ощущений» того же автора. Представлены основные практические алгоритмы и математические методы обработки аудио- и видеоданных, а также альтернативные способы создания мультимедийной продукции.

Для студентов вузов, обучающихся по направлению «Информатика и вычислительная техника», аспирантов и специалистов, а также для тех, кто интересуется вопросами обработки аудио- и видеоданных как несъемлемой части мультимедийных технологий.

ББК 32.97

ISBN 978-5-9502-0415-9

© Крапивенко А. В., 2009
© ЗАО «Издательское предприятие
«Вузовская книга», 2010

ПРЕДИСЛОВИЕ

Представленные в учебном пособии методы и средства мультимедиа являются частью лекционного материала по читаемому автором трехсеместровому циклу спецкурсов на факультете прикладной математики и физики Московского авиационного института с 1997 г. За время преподавания учебной дисциплины «Методы, средства и технологии MultiMedia», практических и лабораторных занятий по спецкурсам, изучения смежной литературы, интересных дискуссий со слушателями было сформировано оригинальное методическое содержание излагаемого материала.

Книга является продолжением учебного пособия «Технологии мультимедиа и восприятие ощущений» (Крапивенко А. В., 2009) [1], в котором основное внимание уделено физиологическим и психологическим особенностям восприятия мультимедийной продукции. В данном пособии описаны конкретные технологии, используемые для создания соответствующего мультимедийного продукта: алгоритмы визуализации, создание виртуальных миров, цифровые интерфейсы передачи данных, аудиоаппаратура, организационные моменты мультимедиaproекта.

Представленные технологии помогут глубже понять принципы функционирования современных мультимедийных систем, узнать, как визуализируется сложная 3D-графика и как можно сделать это самостоятельно, учесть при планировании своего мультимедийного проекта специфику носителей и интерфейсов передачи данных, предусмотреть последовательность организационных действий при старте проекта.

Глава 8 об аудиотехнологиях «Цифровые технологии записи и воспроизведения звука» написана совместно с О. В. Казанцевым. Серьезный вклад в подготовку книги внесли Н. П. Аносова и профессор В. Е. Зайцев.

ГЛАВА 1

ИНСТРУМЕНТЫ МУЛЬТИМЕДИА. МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ СРЕД

Терминология и определения. Классификация инструментальных средств. Способы представления 3D-сцен. Виды геометрического моделирования.

1.1. Место мультимедиа в компьютерных технологиях

Существует большое количество определений термина «*multimedia*», зачастую противоречащих и исключающих друг друга. Весьма размытая область определений рассматриваемого термина находится на стыке нескольких дисциплин, направлений и видов деятельности. В результате термин «*multimedia*» или его русскоязычный аналог «мультимедиа» используется весьма свободно и не всегда точно [1]. Неудивительно, что отношение к слову, а как следствие, и к стоящему за ним понятию, получилось крайне разнообразным и не всегда адекватным. Даже профессионалы при употреблении этого термина имеют в виду не всегда одно и то же. На сегодняшний день существует по крайней мере три различных понимания термина «мультимедиа» [2].

Первое и, вероятно, главное значение этого слова — «мультимедиа как идея», т. е. новая форма выражения художественного творчества новыми компьютерными средствами:

Второе значение слова «мультимедиа», как правило, используется чаще всего и особенно среди профессионалов: так называют программно-аппаратное обеспечение, которое позволяет работать с данными различной природы. Это мультимедийные платы, мультимедийные машины, мультимедийные комплексы и, наконец, программный мультимедийный инструментарий.

Третье значение термина — продукт, составленный из данных всевозможных типов, — структурированный каталог, энциклопедия, учебник. Главные черты такого продукта — объем и разнообразие данных, возможность прямого доступа к ним. В целях улучшения дружественности интерфейса создается система анимированных меню, перекрестных ссылок и интерактивных подсказок, которая служит путеводителем по мультимедиа продукту [3].

Обобщая эти три значения мультимедиа, можно ввести следующее определение.

ОПРЕДЕЛЕНИЕ 1. Под «мультимедиа» понимают современную компьютерную технологию, позволяющую объединить в программно-аппаратной системе различные типы мультимедийных данных (таких, как текст, звук, видео, анимация) для создания единой информационной среды.

Для того чтобы исключить зацикленность определения мультимедиа, введем определение типов мультимедийных данных.

ОПРЕДЕЛЕНИЕ 2. Типы мультимедийных данных изображаются звуками, последовательностями видеокадров, статичными изображениями, вкусовыми, обонятельными и тактильными импульсами — т. е. любыми сигналами (сообщениями), предназначенными для воздействия на органы чувств человека [4].

1.2. Виды инструментальных средств мультимедиа

Инфраструктура мультимедиа — это целый комплекс взаимосвязанных методологических, технических, программных, технологических и производственных систем, позволяющих создавать мультимедийные приложения и продукты.

Инструментальные средства, которые обеспечивают автоматическое создание базовых элементов мультимедийных данных в цифровом представлении и позволяют соединять их в одном программном модуле с единым интерфейсом (мультимедийном приложении) или создавать готовый программный продукт (мультимедиа продукт), принято называть средствами мультимедиа.

В процессе реализации проекта формируются информационные объекты разнообразного содержания, работать с которыми можно в разной последовательности. Такая возможность выбора пользователем режима диалога называется *интерактивной* [5].

Рассмотрим классификацию инструментальных средств по способам создания:

- первая группа предназначена для быстрого создания демонстрационных роликов, презентаций, гипермедиа публикаций;
- вторая группа инструментальных средств объединяет все средства авторских систем;
- третья группа использует средства языков программирования.

Авторские системы мультимедиа занимают промежуточное положение между средствами быстрого создания простой продукции и «тяжелыми» инструментальными платформами, требующими серьезного программирования.

Эти системы представляют собой метод структурирования элементов и объектов мультимедиа, а также способы управления ими с помощью специальных автоматизированных процедур.

Рассмотрим классификацию методов, использованных в авторских системах, предложенную Джеми Сигларом и подробно описанную в работе [5]. Согласно его классификации, можно выделить восемь типов авторских систем, использующих следующие методологии.

1. Язык сценариев (Scripting Language).
2. Изобразительное управление потоками данных (Icon/Flow control).
3. Кадр (Frame).
4. Карточка с языком сценариев (Card/Scripting).
5. Временная шкала (Timeline).
6. Иерархические объекты (Hierarchical objects).
7. Гипермедиассылки (Hypermedia linkage).
8. Маркеры (Tagging).

Методология авторских систем представляет собой совокупность принципов, методов и способов для разработки и реализации проекта мультимедиа.

Методология «Язык сценариев» наиболее близка по форме к традиционному программированию и разработана с учетом принципа алгоритмизации. В ее основе лежит метод объектно-ориентированных языков программирования, который определяет взаимодействие элементов мультимедиа, расположение активных зон, назначение кнопок, синхронизацию кадров и т. д. Использование такого метода увеличивает время разработки, но в результате можно получить более точное взаимодействие элементов и, соответственно, лучшее качество мультимедиа продукта.

Методология «Изобразительное управление потоками данных» основана на методе, который учитывает принцип визуального программирования. Метод используется для быстрого создания прототипа проекта или выполнения задач, которые необходимо завершить в кратчайшие сроки. Данный метод обеспечивает минимальное время обработки данных. В программах, разработанных по данному методу, существует набор (палитра) пиктограмм. Пиктограммы содержат различные функциональные возможности по созданию связей между элементами.

Главным достоинством такого метода является возможность ускорения работы над дизайном мультимедийного приложения, за счет наглядного управления потоками данных.

Авторские системы, построенные на базе этого метода, имеют самые медленные исполняемые модули. Применение авторских систем такого типа — наиболее подходящий путь для построения мультимедийных приложений со сложными функциями взаимодействия (обучающие программы).

Методология «Кадр» подобна методологии изобразительного управления потоками данных. Она включает в себя палитру пиктограмм. Однако связи между пиктограммами могут иметь сложную структуру. Отличие от метода «Изобразительное управление потоками данных» заключается в отсутствии возможности визуального управления потоками данных.

Методология «Карточка с языком сценариев» представляет собой мощный по своим возможностям метод с учетом принципа точной и жесткой структуризации сюжета. Хорошо подходит для гипертекстовых приложений и особенно для прикладных программ с интенсивным перемещением информационных объектов (HyperCard). Такие авторские системы часто используются для разработки прикладных программ общего назначения. Одним из достоинств авторских систем такого типа является наиболее легкий процесс обучения. Авторские системы, построенные на основе метода карточки с языком сценариев, имеют множество шаблонов, примеров и готовых графических элементов пользовательского интерфейса, а также снабжены интерактивными учебными программами.

Главный недостаток авторских систем на основе карточки с языком сценариев — невозможность обеспечить точное управление синхронизацией и выполнение параллельных процессов. Наилучшим применением данных авторских систем является подготовка приложений, которые можно логически организовать в виде отдельных карточек с гипертекстовыми связями между ними.

Методология «Временная шкала» ориентирована на управление поведением информационных объектов с учетом принципа синхронизации их действий. По структуре пользовательского интерфейса авторская система на основе временной шкалы 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 *труппа* (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, основанные на методологии временной шкалы, лучше всего подходят для разработки мультимедийных приложений с интенсивным использованием мультиплексии или таких, где требуется синхронизация различных мультимедийных составляющих. Главный недостаток таких авторских систем — сложность в изучении внутреннего языка сценариев.

Методология «Иерархические объекты» основана на методе объектно-ориентированного программирования, учитывающем принцип визуализации объекта. Благодаря визуальному представлению объектов и информационных составляющих мультимедийного проекта, можно создавать сложные конструкции с развитым сюжетом. Подобные системы обычно довольно дорогие и используются в основном профессиональными разработчиками мультимедийных приложений, так как требуют специальной профессиональной подготовки.

Методология «Гипермедиассылки» подобна методологии «Кадра», в которой показываются концептуальные связи между информационными элементами. Однако ей недостает визуального представления связей. Авторские системы, построенные по этому методу, весьма просты в освоении.

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

Методология «Маркеры» разработана на основе использования специальных команд, которые называются *маркерами страниц*, или тэгами, и используются при связывании страниц для обеспечения взаимодействия и объединения элементов мультимедиа. Метод «Маркеры» имеет ограниченные возможности по отслеживанию связей. Он используется при подготовке диалоговых справочных материалов (словари). С развитием сети Internet авторские системы, созданные на основе метода «Маркеры», нашли широкое применение при создании страниц для узлов глобальной компьютерной сети.

1.3. Моделирование трехмерных сред

При подробном изучении мультимедийных технологий отдельное место занимает моделирование и визуализация трехмерных сред. Методы трехмерного моделирования уже давно используются в системах автоматизированного проектирования, научных визуализациях, обучающих системах. Но их использование для виртуальных погружений, реалистичного моделирования объектов в кино- и игровых индустриях, т. е. в типичных областях мультимедийных технологий, началось сравнительно недавно.

Дальнейшей нашей целью является рассмотрение основных принципов представления и этапов подготовки трехмерных сред, способов геометрического описания объектов сцены, визуализации и интерактивного взаимодействия пользователя с объектами.

Отметим, что при визуализации сцен для вывода на растровые устройства отображения не всегда минимальной единицей отображения является пиксель. Как известно, пиксель при максимальном разрешении монитора представляется одним элементом раstra экрана — *триадой*, которая в свою очередь комбинируется из трех субпикселов, или элементов триад: красного, зеленого и синего цветов. Для дисплеев с цифровым интерфейсом (например, жидкокристаллических) компания Microsoft реализовала технологию ClearType, позволяющую повысить читабельность выводимых на экран шрифтов с помощью прямого управления элементами триад [6].

Эта технология получила широкое распространение среди производителей портативных устройств — PDA, мобильных телефонов, ноутбуков. Ее поддержка реализована также в семействе операционных систем Windows. В отличие от традиционного попиксельного размытия контуров букв во всех направлениях, сутью способа является выборочное сглаживание отдельных сторон букв с помощью управления яркостью отдельных элементов триад. При этом сглаживание не изменяет вертикальных размеров текста (рис. 1).



Рис. 1. Управление элементами триад в технологии ClearType

1.4. Представление 3D-моделей

Напомним ряд положений компьютерной графики. Компонентами геометрической модели являются:

- пространственное расположение объектов;
- форма (геометрия объекта) и атрибуты объектов: цвет, текстура, прозрачность;
- топология (связность с другими объектами).

Существуют несколько распространенных подходов к компьютерному геометрическому моделированию — *каркасный*, *поверхностный*, *твёрдотельный* [7].

Каркасная модель (*wireframe modeling*) полностью описывается в терминах точек и линий. Каркасное («проволочное») моделирование представляет собой моделирование самого низкого уровня и имеет ряд серьезных ограничений, большинство из которых возникает из-за недостатка информации о гранях, заключенных между линиями, и невозможности выделить внешнюю и внутреннюю область изображения твердого объемного тела. Однако каркасная модель требует гораздо меньше оперативной памяти, чем две другие модели, и может оказаться вполне пригодной для решения некоторых задач, относящихся к простым формам.

Поверхностная модель (*surface modeling*) определяется с помощью точек, линий и поверхностей. Таким образом, ее можно рассматривать как модель более высокого уровня, чем каркасная модель, и, следовательно, как более гибкую и многофункциональную. Это наиболее часто встречающаяся модель.

Поверхностное моделирование имеет следующие преимущества по сравнению с каркасным:

- способность изображать сложные криволинейные грани;
- способность выделять грани и таким образом обеспечивать средство получения тоновых трехмерных изображений;
- способность распознавать особые построения на поверхностях, например — отверстия.

Метод поверхностного моделирования наиболее эффективен при проектировании и изготовлении сложных криволинейных поверхностей, таких как корпуса автомобилей.

Хотя методы поверхностного моделирования обладают многими достоинствами, существует ряд ограничений на их использование. Основными являются следующие ограничения:

- возникновение неоднозначностей при моделировании реальных сложных твердых тел;

- недостаточность точности представления некоторых поверхностных моделей для обеспечения надежных данных об объемных трехмерных телах;
- сложность процедур удаления скрытых линий и отображения внутренних областей.

Твердотельная модель (*solid modeling*) описывается в терминах того трехмерного объема, который занимает определяемое этой моделью тело. Таким образом, твердотельное моделирование является единственным средством, которое обеспечивает полное однозначное описание трехмерной геометрической формы. Этот способ моделирования представляет собой самый современный и наиболее мощный из рассматриваемых методов.

Преимущества твердотельных моделей:

- полное определение объемной формы с возможностью разграничения внешней и внутренней областей объекта;
- обеспечение автоматического удаления скрытых линий;
- автоматическое построение трехмерных разрезов компонентов.

Методы твердотельного моделирования, которые обычно используются в прикладных системах, делятся на два класса: метод конструктивного представления (метод C-Rep) и метод граничного представления (метод B-Rep). Они отличаются друг от друга способом хранения моделей.

Представление геометрии объектов при поверхностном моделировании в свою очередь может осуществляться следующими способами:

- 1) *воксельным*;
- 2) *полигональным (триангулярным)*;
- 3) *функционально-параметрическим*.

1.5. Воксельный способ

ОПРЕДЕЛЕНИЕ 3. Пиксел (*pixel*) — слово, образованное от «picture element», т. е. наименьший элемент изображения, выводимого на дисплей. Другое название — *pel*.

Также, пиксел — это двумерные координаты, используемые для определения горизонтальной пространственной позиции точки минимального размера в пределах растрового изображения.

ОПРЕДЕЛЕНИЕ 4. Воксель (*voxel*) — слово, образованное от VOLumized pixEL, т. е. объемный пиксел.

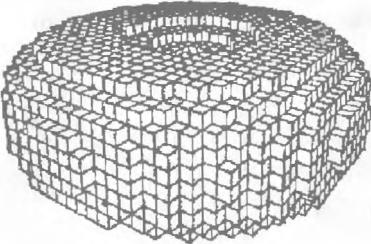
Объемное изображение образуется из элементов, каждый из которых является единичным кубом. Этот куб называется *воксели* и является *квантом объема*. Как пиксел, так и воксель характеризуются значениями, определяющими яркость, цвет, прозрачность и другие характеристики в соответствующей точке исследуемого объекта. Главным отличием вокселя от пикселя является то, что он характеризуется еще и глубиной, т. е. на пиксел накладывается значение глубины данной точки [8].

Алгоритмы создания реалистичных 3D-изображений рассчитаны на многочасовой аппаратно-программный расчет (подобные алгоритмы лежат в основе современной 3D-анимации и спецэффектов, используемых в кино). Однако для рендеринга сцен в реальном времени даже современных аппаратных возможностей может оказаться недостаточно. Существует иной подход, основанный на использовании *воксельной* графики, что значительно уменьшает время расчета при сравнимом качестве [9]. Возможности воксельных ландшафтов хорошо иллюстрируются применением в реальных задачах — авиационных тренажерах, симуляторах и т. д. Использование вокселов существенно уменьшает вычислительную нагрузку на процессор, а при наличии современного 3D-ускорителя более половины итерационных алгоритмов (*z*-буфер, билинейная фильтрация и проч.), используемых в программе, могут быть выполнены его силами (рис. 2).

Широкое применение воксели нашли также в медицинской компьютерной томографии [10]. Изображения с большого количества рентгеновских или ультразвуковых снимков под разными углами (порядка 100–200 снимков) обрабатываются, и создается трехмерный массив плотностей различных участков тканей исследуемого органа (рис. 3). Этот массив представляет собой «объемную картину», элементом которой является воксель.

Следующее прикладное применение вокселов — компьютерные игры. Во многих динамических играх, где требуется визуализация сложных сцен в масштабе реального времени, с помощью трехмерного воксельного массива задается ландшафт или объекты виртуального мира. Однако не стоит приписывать пиксели и вокселя равнотерпимость — это удобно, но не более того. Иногда даже удобнее считать воксели неким параллелепипедом — например в карте высот.

Рис. 2. Воксельное моделирование



Воксельное моделирование



Рис. 3. Пример использования voxelной графики

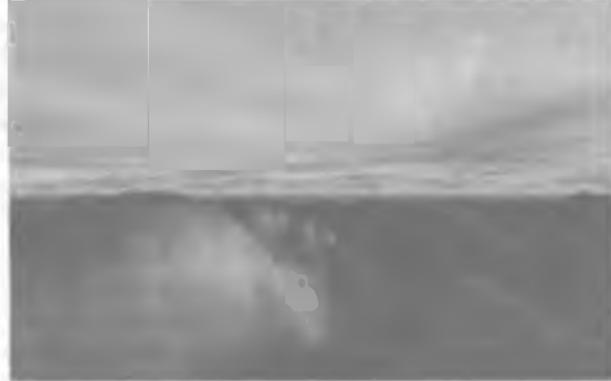


Рис. 4. Вoxelный ландшафт в программе Mars.exe

1.6. Полигональный (треугольный) способ

Поверхностное представление объектов связано с необходимостью задания всех поверхностей, ограничивающих данное тело. В нашем случае то что находится внутри (непрозрачного) тела, мы считаем вакуумом и рассматриваем лишь поверхность, так как при визуальном представлении человек видит внешнюю часть, а не внутреннее содержание объекта. В частности можно считать, что свет распространяется в воздухе беспрепятственно, достигает поверхности, и на поверхности происходят интересующие нас явления.

Поверхности можно задавать аналитическим способом (цилиндрические, сферические поверхности), но их класс ограничен, и поэтому данного способа недостаточно для описания форм более сложных объектов. В силу этих причин используются полигональные сетки, т. е. для объектов, имеющих гладкую, непрерывную поверхность, можно по-

строить полигональную модель, аппроксимируя поверхность с помощью многоугольников — чаще всего треугольников.

Поэтому данный способ представления трехмерных моделей часто называют *триангулярным способом*, так как минимальное количество точек, необходимое для задания ориентации плоскости — три.

Основные типы *векторных геометрических примитивов*:

- точка;
- прямая;
- луч;
- вектор;
- плоскость.

Таким образом, полигональная сетка представляет собой совокупность ребер, вершин и многоугольников. В свою очередь, ребра определяются двумя вершинами, многоугольник — замкнутой последовательностью ребер, и поэтому данное представление (вершины, ребра, грани) не является независимым.

При таком представлении часто возникают задачи, связанные с реальным моделированием: найти ребра, соответствующие данной вершине, или найти вершины, образующие многоугольник и т. д.

На основе геометрических примитивов строится большинство трехмерных моделей, так как в приложениях, критичных к скорости выполнения, очень удобно использовать относительную простоту вычислений, связанных с этими геометрическими примитивами.

1.7. Функционально-параметрический способ

Еще один мощный и гибкий способ представления 3D-моделей — это аппроксимация с помощью *функционально-параметрических поверхностей*, чаще всего — *сплайнов*.

Историю сплайнов принято отсчитывать от момента появления первой работы Шенберга в 1946 г. [11]. Сначала сплайны рассматривались как удобный инструмент в теории и практике приближения функций. Однако довольно скоро область их применения начала быстро расширяться и обнаружилось, что существует много сплайнов самых разных типов. Сплайны стали активно использоваться в численных методах, в системах автоматического проектирования и автоматизации научных исследований и, конечно, в компьютерной графике.

Достаточно типичной является следующая задача: по заданному массиву точек на плоскости (2D) или в пространстве (3D) построить

кривую либо проходящую через все эти точки (задача интерполяции), либо проходящую вблизи от этих точек (задача сглаживания).

Совершенно естественно возникают вопросы:

- 1) В каком классе кривых искать решение поставленной задачи?
- 2) Как искать?

Ответы на поставленные вопросы мы подробно рассмотрим в следующей главе.

1.8. Мультимедийное резюме

В заключение первой главы еще раз необходимо отметить, что рост интереса к мультимедийным технологиям обусловлен рядом причин, среди которых следует выделить следующие:

- появление высокопроизводительных компьютеров, способных поддерживать полноценный мультимедийный интерфейс с пользователем в масштабе реального времени, поскольку в отличие от текста и графики, статичных по своей природе, аудио- и видео-сигналы рассматриваются только в реальном масштабе времени;
- наличие аппаратного и программного мультимедийного обеспечения, работающего по определенному набору стандартов;
- создание инструментальных средств, дающих возможность пользователям создавать свои прикладные мультимедийные программы, не имея большого опыта программирования.

ГЛАВА 2

ФУНКЦИОНАЛЬНО-ПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ. СПЛАЙН-ФУНКЦИИ

*Интерполяционный кубический сплайн. Сплайн Безье.
B-сплайн. NURBS-кривые. NURBS-поверхности.*

2.1. Интерполяционные сплайны

Обратимся для определенности к задаче интерполяции, впервые затронутой в предыдущей главе, и начнем рассмотрение с обсуждения правил выбора класса кривых.

Очевидно, что допустимый класс кривых должен быть таким, чтобы решение задачи было единственным. Кроме того, желательно, чтобы построенная кривая изменялась плавно. Пусть на плоскости задан набор точек (x_i, y_i) , $i=0, 1, \dots, m$, таких, что:

$$x_0 < x_1 < \dots < x_{m-1} < x_m.$$

То обстоятельство, что точки заданного набора занумерованы в порядке возрастания их абсцисс, позволяет искать кривую в классе графиков функций.

Как известно из курса математического анализа, существует интерполяционный многочлен Лагранжа

$$L_m(x) = \sum_{i=0}^m y_i \frac{\omega_m(x)}{(x - x_i)\omega_m(x_i)},$$

где $\omega_m(x) = \prod_{j=0}^m (x - x_j)$, график которого проходит через все заданные точки (x_i, y_i) , $i=0, 1, \dots, m$.

Это обстоятельство и простота описания (заметим, что многочлен однозначно определяется набором своих коэффициентов; в данном случае их число совпадает с количеством точек в заданном наборе) являются несомненными достоинствами построенного интерполяционного многочлена.

Однако нам полезно остановиться и на некоторых недостатках предложенного подхода.

1. Степень многочлена Лагранжа на единицу меньше числа заданных точек. Поэтому, чем больше точек задано, тем выше степень такого

многочлена. И, хотя график интерполяционного многочлена Лагранжа всегда будет проходить через все точки массива, его уклонение от ожидаемого может оказаться довольно значительным (рис. 5).

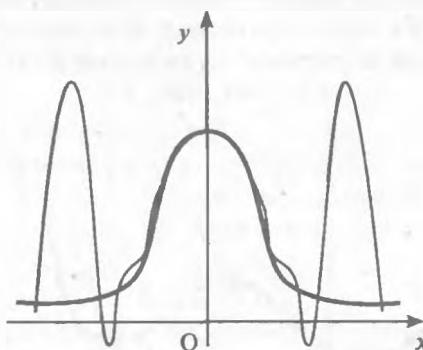


Рис. 5. Многочлен Лагранжа

2. Изменение одной точки (ситуация, довольно часто встречающаяся на практике) требует полного пересчета коэффициентов интерполяционного многочлена и к тому же может существенно повлиять на вид задаваемой им кривой.

С другой стороны, приближающую кривую можно построить и совсем просто: если последовательно соединить точки заданного набора прямолинейными отрезками, то в результате получится ломаная. При такой кусочно-линейной интерполяции требуется найти всего $2t$ чисел (каждый прямолинейный отрезок определяется ровно двумя коэффициентами), но к сожалению, построенная таким образом аппроксимирующая кусочно-линейная функция не обладает нужной гладкостью: уже первая производная этой функции терпит разрывы в узлах интерполяции.

Рассмотрев эти две крайние ситуации, попробуем найти класс функций, которые в основном сохранили бы перечисленные выше достоинства обоих подходов и одновременно были бы в известной степени свободны от их недостатков.

Для этого поступим так: будем использовать многочлены (как и в первом случае) и строить их последовательно, звено за звеном (как во втором случае). В результате получится так называемый *полиномиальный многозвенник*. При подобном подходе важно правильно выбрать степени используемых многочленов, а для плавного изменения результирующей кривой необходимо еще тщательно подобрать коэффициенты многочленов (из условий гладкого сопряжения соседних звеньев).

То что получится в результате описанных усилий, называют сплайн-функциями или просто сплайнами.

Для того чтобы понять, какое отношение имеют сплайн-функции к чертежным сплайнам, возьмем гибкую стальную линейку, поставим ее на ребро и, закрепив один из концов в заданной точке, поместим ее между опорами, которые располагаются в плоскости Oxy в точках (x_i, y_i) , $i=0, 1, \dots, m$, где $x_0 < x_1 < \dots < x_{m-1} < x_m$ (рис. 6).

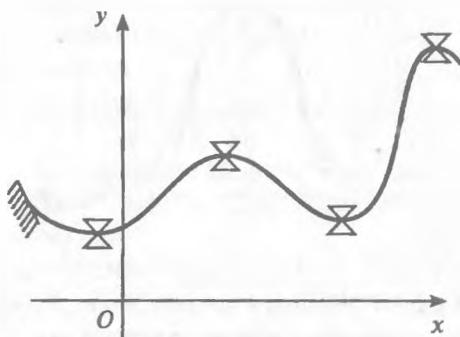


Рис. 6. Чертежный сплайн

Интересно отметить, что функция $y = S(x)$, описывающая профиль линейки, обладает следующими интересными свойствами:

- с довольно большой точностью часть графика этой функции, заключенную между любыми двумя соседними опорами, можно считать многочленом третьей степени;
- на всем промежутке $[x_0, x_m]$ функция $y = S(x)$ дважды непрерывно дифференцируема.

Построенная функция $S(x)$ относится к так называемым интерполяционным кубическим сплайнам. Этот класс в полной мере удовлетворяет высказанным выше требованиям и обладает еще целым рядом замечательных свойств. Переходим, однако, к точным формулировкам.

ОПРЕДЕЛЕНИЕ 5. Интерполяционным кубическим сплайном называется функция $S(x)$, обладающая следующими свойствами:

1. График этой функции проходит через каждую точку заданного массива, $S(x_i) = y_i$, $i = 0, 1, \dots, m - 1$;
2. На каждом из отрезков $[x_i, x_{i+1}]$, $i = 0, 1, \dots, m - 1$, функция является многочленом третьей степени, $S(x) = \sum_{j=0}^3 a_j^i (x - x_i)^j$;
3. На всем отрезке задания $[x_0, x_m]$ функция $S(x)$ имеет непрерывную вторую производную.

Так как на каждом из отрезков $[x_0, x_m]$ сплайн $S(x)$ определяется четырьмя коэффициентами, то для его полного построения на всем отрезке задания необходимо найти $4m$ чисел.

Третье условие будет выполнено, если потребовать непрерывности сплайна во всех внутренних узлах x_i , $i=0, 1, \dots, m-1$ (это дает $m-1$ условий на коэффициенты), а также его первой производной ($m-1$ условий) и второй (еще $m-1$ условий) производных в этих узлах. Вместе с первым условием получаем равенства

$$m-1 + m - 1 + m - 1 + m + 1 = 4m - 2.$$

Недостающие два условия для полного определения коэффициентов можно получить, задав, например, значения первых производных на концах отрезка $[x_0, x_m]$ (границные условия):

$$S(x_0) = l_0, S(x_m) = l_m.$$

Существуют граничные условия и других типов.

Более сложная задача построения по заданному набору точек в трехмерном пространстве интерполяционной функции двух переменных решается похожим образом, при этом получаемая поверхность называется интерполяционным бикубическим сплайном [11].

В целом, достоинства предложенного способа несомненны: для решения линейных систем, возникающих в ходе построения сплайн-функций, существует много эффективных методов; графики построенных сплайн-функций проходят через все заданные точки.

Вместе с тем, изменение лишь одной точки (например, при интерактивном моделировании объекта) при описанном подходе заставляет пересчитывать заново все коэффициенты. К тому же, во многих задачах исходный набор точек задается приближенно и, значит, требование неукоснительного прохождения графика искомой функции через каждую точку этого набора оказывается излишним.

Поскольку интерполяционные кубические сплайны обладают рядом существенных недостатков, мы значительно расширим классы, в которых будет вестись поиск соответствующих кривых и поверхностей. Более точно, мы откажемся от требования однозначного проектирования искомой кривой на координатную ось, а поверхности — на координатную плоскость. Такой подход позволяет ослабить требования к задаваемому массиву.

2.2. Сплайновые кривые. Кривая Безье

В этом разделе более удобно будет пользоваться параметрическими уравнениями кривой [11]. Напомним необходимые понятия.

ОПРЕДЕЛЕНИЕ 6. Параметрически заданной кривой называется множество γ точек $M(x, y, z)$, координаты x, y, z которых определяются соотношениями

$$\begin{aligned} x = x(t), \quad y = y(t), \quad z = z(t), \\ a \leq t \leq b, \end{aligned} \tag{2.2.1}$$

где $x(t), y(t), z(t)$ — функции, непрерывные на отрезке $[a, b]$ (рис. 7).

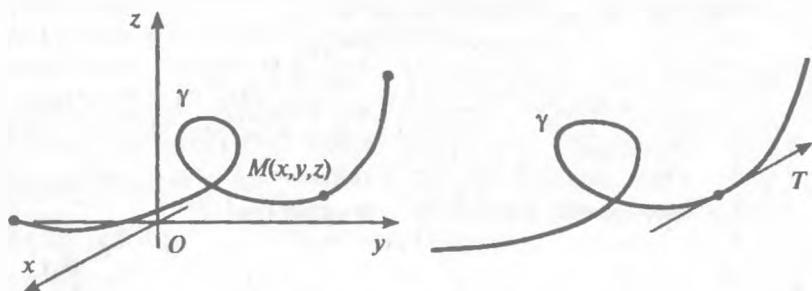


Рис. 7. Сплайновые кривые

Соотношения (2.2.1) называются параметрическими уравнениями кривой γ . Без ограничения общности можно считать, что $a=0$ и $b=1$; этого всегда можно добиться при помощи замены вида

$$u = \frac{t-a}{b-a}.$$

Полезна векторная форма записи параметрических уравнений:

$$\mathbf{r} = \mathbf{r}(t), \quad 0 \leq t \leq 1, \quad \text{где } \mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} + z(t)\mathbf{k}.$$

Параметр t задает ориентацию параметризованной кривой γ (порядок прохождения точек при монотонном возрастании параметра).

Кривая γ называется регулярной кривой, если $\mathbf{r}'(t) \neq 0$ в каждой ее точке. Это означает, что в каждой точке кривой существует касательная к ней и эта касательная меняется непрерывно вслед за перемещающейся вдоль кривой ее текущей точкой. Единичный вектор касательной к кривой γ равен

$$\mathbf{T}(t) = \frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}.$$

Если дополнительно потребовать, чтобы задающая кривую векторная функция имела вторую производную, то будет определен вектор кривизны кривой

$$K(t) = \frac{[r'(t) \times r''(t)] \times r'(t)}{|r'(t)|^4},$$

модуль которого характеризует степень ее отклонения от прямой. В частности, если γ — отрезок прямой, то $K=0$.

Необходимо отметить, что параметрические уравнения мы рассматриваем в трехмерном пространстве. Практически все сказанное будет верно и для плоского случая. Дело в том, что параметрическое описание плоской кривой не накладывает никаких ограничений на ее расположение относительно координатных осей; кривая не обязана однозначно проектироваться на координатную ось, как это имеет место в случае ее явного задания $y=y(x)$. В частности, кривая может быть замкнутой, самопересекающейся и т. д.

Рассмотрим некоторые подходы к построению слаживающей кривой. Пусть на плоскости или в пространстве задан упорядоченный набор точек, определяемых векторами V_0, V_1, \dots, V_m . Ломаная $V_0V_1 \dots V_m$ называется контрольной ломаной, порожденной массивом $V=\{V_0, V_1, \dots, V_m\}$.

ОПРЕДЕЛЕНИЕ 7. Кривой Безье, определяемой массивом V , называется кривая, определяемая векторным уравнением

$$r(t) = \sum_{i=0}^m C_m^i t^i (1-t)^{m-i} V_i, \quad 0 \leq t \leq 1,$$

где $C_m^i = \frac{m!}{i!(m-i)!}$ — коэффициенты в разложении бинома Ньютона

(число сочетаний из m элементов по i).

Кривая Безье обладает следующими интересными свойствами:

- 1) она является гладкой;
- 2) начинается в точке V_0 и заканчивается в точке V_m , касаясь при этом отрезков V_0V_1 и $V_{m-1}V_m$ контрольной ломаной;
- 3) функциональные коэффициенты $C_m^i t^i (1-t)^{m-i}$ при вершинах V_i , $i=0, 1, \dots, m$ — суть универсальные многочлены ($m+1$ многочленов Бернштейна); они неотрицательны, и их сумма равна единице:

$$\sum_{i=0}^m C_m^i t^i (1-t)^{m-i} = (t + (1-t))^m = 1.$$

Поэтому кривая Безье целиком лежит в выпуклой оболочке, порождаемой массивом (рис. 8).

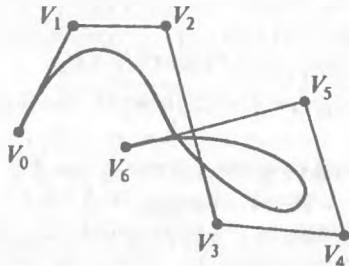


Рис. 8. Кривая Безье

При $m=3$ получаем (элементарную) кубическую кривую Безье, определяемую четверкой точек V_0, V_1, V_2, V_3 и описываемую векторным параметрическим уравнением

$$r(t) = ((1-t)V_0 + 3tV_1)(1-t) + 3t^2V_2)(1-t) + t^3V_3, \quad 0 \leq t \leq 1,$$

или в матричной записи, $r(t) = VMT$, $0 \leq t \leq 1$, где

$$r(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}; \quad V = (V_0 \ V_1 \ V_2 \ V_3) = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \end{pmatrix};$$

$$M = \begin{pmatrix} 1 & -3 & 3 & 1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad T = \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}.$$

Матрица M называется *базисной матрицей* кубической кривой Безье.

Порядок точек в заданном наборе существенно влияет на вид элементарной кривой Безье.

Наряду с отмеченными достоинствами кривые Безье обладают и определенными недостатками. Основных недостатков у элементарных кривых Безье три:

1) степень функциональных коэффициентов напрямую связана с количеством точек в заданном наборе (на единицу меньше);

2) при добавлении хотя бы одной точки в заданный набор необходимо провести полный пересчет функциональных коэффициентов в параметрическом уравнении кривой;

3) изменение хотя бы одной точки приводит к заметному изменению вида всей кривой.

В практических вычислениях часто оказывается удобным пользоваться кривыми, составленными из элементарных кривых Безье, как правило — кубических.

Однако при построении составных кривых часто приходится сталкиваться с ситуацией, когда каждый из регулярных фрагментов, участвующих в создании новой кривой, имеет свою собственную параметризацию. Чтобы учесть это обстоятельство, удобно использовать класс так называемых геометрически-непрерывных кривых.

ОПРЕДЕЛЕНИЕ 8. Составная кривая называется G^1 -непрерывной (геометрически), если вдоль этой кривой единичный вектор ее касательной изменяется непрерывно, и G^2 -непрерывной (геометрически), если вдоль этой кривой изменяется непрерывно и вектор кривизны.

Обратимся к рассмотрению составных кривых Безье. Составная кубическая кривая Безье представляет собой объединение элементарных кубических кривых Безье $\gamma_1, \dots, \gamma_m$, таких, что $r_i(1) = r_{i+1}(0)$, $i = 0, 1, \dots, m-1$, где $r = r_i(t)$, $0 \leq t \leq 1$ — параметрическое уравнение кривой γ_i .

Чтобы составная кривая Безье, определяемая набором вершин $V_0, V_1, \dots, V_{m-1}, V_m$,

1) была G^1 -непрерывной кривой, необходимо, чтобы каждые три точки $V_{3i-1}, V_{3i}, V_{3i+1}$ этого набора лежали на одной прямой;

2) была замкнутой G^1 -непрерывной кривой, необходимо, кроме того, чтобы совпадали первая и последняя точки, $V_0 = V_m$, и три точки $V_{m-1}, (V_0 = V_m), V_1$ лежали на одной прямой;

3) была G^2 -непрерывной кривой, необходимо, чтобы каждые пять точек $V_{3i-2}, V_{3i-1}, V_{3i}, V_{3i+1}, V_{3i+2}$ ($i \geq 1$) заданного набора лежали в одной плоскости.

Составные кубические кривые Безье получили большое распространение в современных компьютерных технологиях. Так, популярный стандарт шрифтов *TrueType* использует сплайны Безье для описания кривизны контуров букв.

Другим примером использования этих полезных кривых являются распространенные векторные редакторы типа *Adobe Illustrator* или *Corel Draw*. При интерактивной работе со сплайнами Безье (рис. 9) пользователь имеет возможность регулировать их кривизну с помощью контрольных управляющих точек (на концах пунктирных линий), представляющих собой не что иное, как вершины опорного графа $V_0, V_1, \dots, V_{m-1}, V_m$.

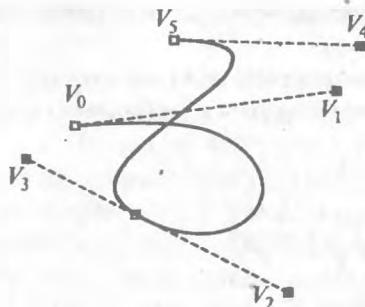


Рис. 9. Работа со сплайнами Безье в CorelDraw

2.3. В-сплайновые кривые

Однако не стоит забывать, что сплайны Безье при всей их распространенности, гибкости, и простоте построения обладают рядом недостатков. Попытаемся найти другой класс кривых, сохраняющих достоинства кривых Безье и лишенных их недостатков.

Так как в векторном уравнении, задающем кривую Безье, векторные составляющие постоянны (это просто вершины массива), то мы уделим основное внимание выбору новых функциональных коэффициентов, стараясь (разумеется по возможности) сохранить при этом замечательные свойства многочленов Бернштейна, ограничив наши рассмотрения кубическими многочленами.

ОПРЕДЕЛЕНИЕ 9. По заданному набору точек V_0, V_1, V_2, V_3 элементарная кубическая В-сплайновая кривая определяется при помощи векторного параметрического уравнения следующего вида:

$$r(t) = \frac{(1-t)^3}{6}V_0 + \frac{3t^3 - 6t^2 + 4}{6}V_1 + \frac{-3t^3 + 3t^2 + 3t + 1}{6}V_2 + \frac{t^3}{6}V_3, \quad 0 \leq t \leq 1,$$

или, в матричной записи, $R(t) = VMT$, $0 \leq t \leq 1$,

$$r(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}, \quad V = (V_0 \ V_1 \ V_2 \ V_3) = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & -3 & 3 & 1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}.$$

Матрица M называется базисной матрицей В-сплайновой кривой.

Функциональные коэффициенты в уравнении, определяющем элементарную В-сплайновую кубическую кривую, неотрицательны, в сум-

мс составляют единицу, универсальны (не зависят от конкретного вида точек в заданной четверке).

Это означает, что рассматриваемый элементарный фрагмент лежит внутри выпуклой оболочки заданных вершин — четырехугольника на плоскости или тетраэдра в пространстве (рис. 10).

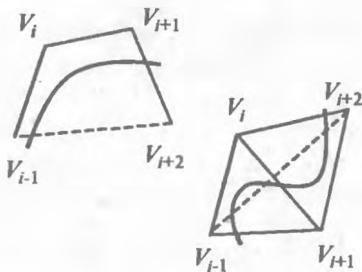


Рис. 10. Элементарная В-сплайновая кубическая кривая

Составная кубическая В-сплайновая кривая, задаваемая параметрическим уравнением

$$r = r(t), \quad 0 \leq t \leq m - 2$$

и определяемая набором точек

$$V_0, V_1, \dots, V_{m-1}, V_m \quad (m \geq 3),$$

представляет собой объединение $m - 2$ элементарных кубических В-сплайновых кривых, $\gamma_3, \dots, \gamma_m$, описываемых уравнениями вида

$$r = r_i(t) = (V_{i-1} V_i V_{i+1} V_{i+2}) M \begin{pmatrix} 1 \\ t-i+1 \\ (t-i+1)^2 \\ (t-i+1)^3 \end{pmatrix},$$

$$i-1 \leq i \leq i, \quad i=1, \dots, m-2.$$

Указанная выше составная В-сплайновая кубическая кривая является G^2 -гладкой кривой и лежит в объединении $m - 2$ выпуклых оболочек, порожденных последовательными четверками точек заданного набора. Добавляя в исходный набор дополнительные точки, можно получать составную В-сплайновую кубическую кривую с разными свойствами.

Область изменения параметра t и расположение на ней точек, соответствующих стыковочным узлам, могут быть совершенно произвольными. Наиболее простой является равномерная параметризация с равноотстоящими целочисленными узлами.

2.4. Рациональные кубические B-сплайны

ОПРЕДЕЛЕНИЕ 10. По заданному набору V_0, V_1, V_2, V_3 рациональная кубическая B-сплайновая кривая в общем случае определяется уравнением следующего вида:

$$r(t) = \frac{\sum_{i=0}^3 w_i n_i(t) V_i}{\sum_{i=0}^3 w_i n_i(t)}, \quad 0 \leq t \leq 1,$$

где

$$n_0(t) = \frac{(1-t)^3}{6}, \quad n_1(t) = \frac{3t^3 - 6t^2 + 4}{6}, \quad n_2(t) = \frac{-3t^3 + 3t^2 + 3t + 1}{6}, \quad n_3(t) = \frac{t^3}{6},$$

а величины w_i , называемые весами (или параметрами формы), — неотрицательные числа, сумма которых положительна.

Замечания:

1. В случае, если все веса равны между собой, приведенное уравнение описывает элементарную кубическую B-сплайновую кривую.
2. Построение составной рациональной B-сплайновой кубической кривой проводится по той же схеме, что и в полиномиальном случае.
3. В последнее время значительный интерес пользователей вызывает класс сплайнов, известный под названием NURBS.

2.5. Области применения NURBS

NURBS (Non-Uniform Rational B-Splines) — рациональные B-сплайны, задаваемые на неравномерной сетке.

В последнее время NURBS-сплайны и поверхности стали основным способом функционально-параметрического описания гладких криволинейных объектов. Такой способ обладает рядом существенных преимуществ [12]:

- предоставляет общее математическое описание как для стандартных аналитических поверхностей (сфера, тор, конус и т. д.), так и для произвольных поверхностей;
- позволяет очень точно моделировать большинство поверхностей, в том числе и достаточно сложных;
- позволяет легко оценить точность и стабильность алгоритма;
- инвариантен при аффинных преобразованиях (таких, например, как перспективная трансформация);

- NURBS обобщают нерациональные B-сплайны, нерациональные и рациональные кривые и поверхности Безье.

Поддержка NURBS реализована во многих профессиональных средах трехмерного моделирования и визуализации, например, таких как 3DStudio MAX, SoftImage, Maya, программном интерфейсе OpenGL (рис. 11).

Одним из главных преимуществ пакетов 3D-моделирования, поддерживающих NURBS-поверхности, является возможность простого создания уровней детализации. Уровни детализации активно применяются, например, в играх, использующих 3D-графику, и языке описания виртуальных миров VRML.

Далее, уровни детализации используются для морфинга уровня детализации (LOD): создается несколько версий одного объекта с разной степенью детализации: самая низкая детализация для отображения объекта на очень далеком расстоянии и самая высокая детализация для самого близкого расстояния. Чем выше детализация, тем из большего числа полигонов состоит объект. Полигоны возникают в процессе тесселяции¹. LOD позволяет уменьшить число полигонов на сцене, и, следовательно, повысить производительность. Например, нет смысла отображать автомобиль во всей его красе на расстоянии, при котором игрок может различить только цвет автомобиля. Переходы между уровнями детализации осуществляется варьированием прозрачности: смена детализации происходит не мгновенно, непрозрачная (старая) версия становится прозрачной, а прозрачная (новая) — более плотной.



Рис. 11. Этапы работы с NURBS-поверхностями

Анимация NURBS-поверхностей также на порядок по своим возможностям превосходит возможности полигональных поверхностей.

¹ Тесселяция (триангуляция) — процесс разбиения поверхности объектов на полигоны.

Как бы не изгибали, крутили и деформировали NURBS-поверхность, острых углов и выпадающих граней не возникает в любом случае.

Существует два типа описания NURBS-кривых и поверхностей — Point и CV (Control Vertex).

Разница между ними заключается лишь в способе управления. Объект «Point» управляется точками, лежащими непосредственно на самом объекте, другими словами, объект проходит через эти точки. Объект «CV» управляет вершинами, которые располагаются вне объекта и связаны между собой линиями (рис. 12) [13].



Рис. 12. Компоненты NURBS-кривой

Это напоминает управляющие вершины Безье. Однако существует важное отличие: управляющие точки Безье влияют сразу на всю поверхность, тогда как управляющие вершины NURBS воздействуют на локальную область, размером которой можно управлять, используя вес. Чем больше вес вершины, тем сильнее влияние этой вершины на поверхность или кривую. Иначе говоря, чем больше вес, тем сильнее меняется форма. Необходимо знать, что одна поверхность или кривая не могут одновременно управляться как точками, так и вершинами, но внутри одного объекта NURBS могут содержаться как подобъекты «Point», так и «CV».

2.6. NURBS-кривые. Определения

Еще раз вернемся к расшифровке аббревиатуры NURBS. Она расшифровывается как Non-Uniform Rational B-Spline(s), или по-русски «неоднородный рациональный би-сплайн». «Неоднородный» означает, что различные области объектов обладают различными свойствами. «Рациональный» означает, что объект NURBS может быть описан при

помощи математических формул. Отличительными характеристиками математики NURBS является форма описания, объединяющая параметрические кривые и поверхности, позволяющая единообразно представлять B-сплайны, кривые Безье, кривые и поверхности канонического вида. NURBS использует математические алгоритмы, которые позволяют виртуально задать любую поверхность или кривую как одно уравнение в виде кусочных полиномов [14].

NURBS-поверхности задаются не только контрольными точками, но и весовыми коэффициентами, заданными для каждой контрольной точки.

NURBS-кривая $C(u)$ является векторной кусочно-непрерывной рациональной полиномиальной функцией и определяется как [15]:

$$C(u) = \frac{\sum_{i=0}^n w_i * P_i * N_{i,k}(u)}{\sum_{i=0}^n w_i * N_{i,k}(u)},$$

где w_i — весовой коэффициент; P_i — контрольная точка (управляющий вектор); $N_{i,k}$ — нормированные базисные B-сплайн функции с шагом k .

Эти B-сплайны определяются рекурсивной функцией

$$N_{i,k}(u) = \frac{u - t_i}{t_{i+k} - t_i} * N_{i,k-1}(u) + \frac{t_{i+k+1} - u}{t_{i+k+1} - t_{i+1}} * N_{i+1,k-1}(u)$$

$$\text{и } N_{i,0}(u) = \begin{cases} 1, & \text{если } t_i \leq u \leq t_{i+1}, \\ 0, & \text{иначе} \end{cases},$$

где t_i — узлы, формирующие вектор узлов $U = \{t_0, t_1, \dots, t_m\}$.

Как показывает формула, определяющая B-сплайны $N_{i,k}(u)$, вектор узлов однозначно определяет B-сплайн. Отношение между числом узлов ($m + 1$), числом шагов (k) функции $N_{i,k}(u)$ и числом контрольных точек ($n + 1$) удовлетворяет равенству $m = n + k + 1$ [16].

Последовательность узлов в векторе подбирается так, чтобы она была неубывающей, т. е. $t_i \leq t_{i+1}$ [17]. Каждая последующая пара узлов представляется как интервал $[t_i, t_{i+1})$ значений параметров для вычисления сегмента поверхности. На рис. 13 показан полный набор B-сплайнов 4-го порядка (кубических) для 9 вершин, построенных на расширенном множестве узлов 0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6 [18]. Параметр t сплайна, построенного на данном множестве узлов, принимает значение на отрезке $0 \leq t \leq 6$.

На рис. 14 по одним и тем же восьми вершинам построены NURBS-кривые 2-го, 4-го, 6-го и 8-го порядка. Из рисунка видно, что чем выше порядок кривой, тем более она гладкая. NURBS-кривая 2-го порядка совпадает со своей характеристической ломаной линией.

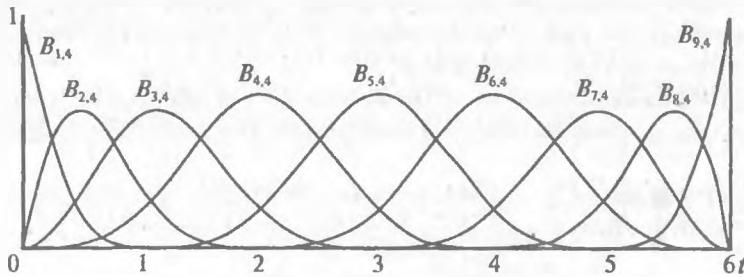


Рис. 13. Набор B-сплайнов 4-го порядка для незамкнутой NURBS-кривой

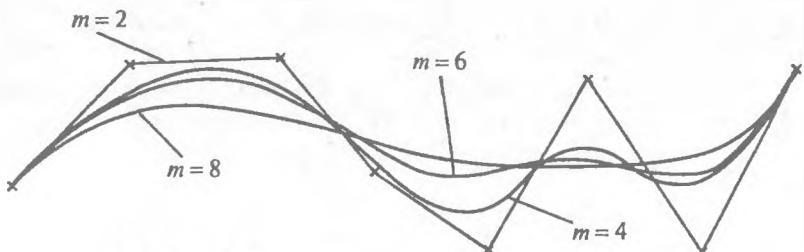


Рис. 14. Влияние порядка на форму NURBS-кривой

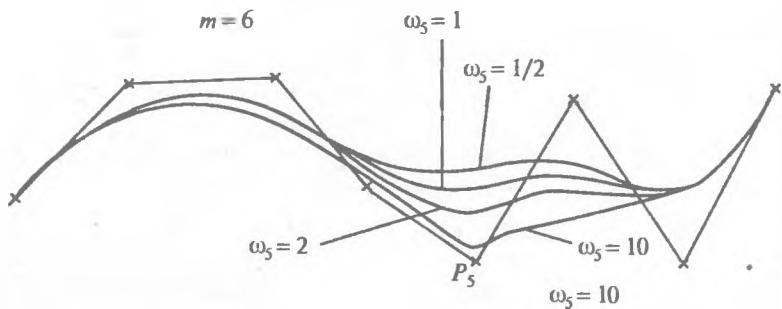


Рис. 15. Влияние веса точки P_5 на форму NURBS-кривой

На рис. 15 показано влияние веса вершины P_5 на форму кривой 6-го порядка. Чем больше вес вершины, тем ближе к ней проходит NURBS-кривая. В общем случае вес вершины может быть нулевым и даже отрицательным.

2.7. Мультимедийное резюме

Рассмотренные сплайн-функции широко используются при функционально-параметрическом моделировании в различных пакетах 2D- и 3D-графики. К преимуществам сплайн-функций, особенно элементарных, относится гладкость функций и сплайновых поверхностей, быстрота вычислений и широкие возможности редактирования.

Однако не все виды сплайнов настолько удобны в использовании. В процессе редактирования традиционного «чертежного» интерполяционного сплайна, при изменении лишь одной контрольной точки необходимо заново пересчитать все остальные коэффициенты.

Для описания геометрически сложных поверхностей при функциональном моделировании 3D-объектов заслуженной популярностью пользуется класс NURBS-кривых и поверхностей, образованных нерациональными B-сплайнами. Их достоинства — весьма точная аппроксимация реальной поверхности, инвариантность ко многим преобразованиям, удобство редактирования методами контрольных точек или вершин.

ГЛАВА 3

ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНЫХ СЦЕН. ПРОСТАЯ МОДЕЛЬ ОСВЕЩЕНИЯ

Этапы подготовки 3D-сцены. Виды текстур. Типы источников освещения. Определение нормали к поверхности. Простая модель освещения.

3.1. Визуализация геометрической модели. Этапы подготовки

Для получения качественной визуализации¹ анимированной трехмерной сцены необходима тщательная подготовка. Этапы подготовки сцены в общем случае можно представить следующим образом:



¹ Процедура визуализации часто называется рендерингом, от англ. «render» — исполнение, визуализация.

Как видно из схемы, рендеринг является лишь завершающим этапом в процессе подготовки сцены к визуализации. Первый этап подробно рассмотрен в предыдущей главе.

Следующий этап — подготовка объектов к анимации. Современные средства 3D-моделирования, например 3DStudio MAX, позволяют запоминать практически любые операции над объектами, от простых преобразований до сложных деформаций. Далее эти операции могут быть анимированы путем указания пользователем на их применение в заданном диапазоне времени. Допускается устанавливать между объектами иерархические взаимосвязи, создавать анимационные последовательности с применением инверсной кинематики или встраивать в сцены системы движущихся микрочастиц, например, для имитации изображения текущей воды или дыма. Для этих частиц задаются разнообразные правила поведения, такие как прилипание к поверхности при соприкосновении или отражение от нее.

3.2. Моделирование текстуры

Для придания более естественного вида сцене желательно иметь возможность менять параметры поверхности (в простейшем случае — цвет) в зависимости от положения точки на ней. Этот процесс называется **текстурированием**.

Существуют разные способы моделирования текстур, но практически все они подразделяются на два основных класса [19]:

- 1) проективные текстуры — статические и анимированные;
- 2) процедурные текстуры.

Представим себе, что необходимо задать определенную текстуру (например мрамор) какому-либо объекту. Возможны два пути.

1. Взять изображение реальной мраморной поверхности и отобразить (спроектировать) его каким-либо образом на поверхность объекта. Т. е. перевести исходные трехмерные координаты точки в двумерные и использовать последние для индексации при наложении изображения.

2. Построить некоторую функцию $C(x, y, z)$, определяющую для каждой точки пространства (x, y, z) цвет таким образом, чтобы объект, цвет которого задается этой функцией, имел вид объекта, сделанного из мрамора.

Первый путь соответствует проективным текстурам. Он наиболее прост, однако имеет целый ряд существенных недостатков: требует большого объема памяти для хранения используемых изображений, обладает сравнительно небольшой гибкостью и к тому же сопряжен

с большими сложностями в подборе способа проектирования («тapping») для объектов сложной формы.

Поэтому в практических задачах, как правило, используется лишь небольшое количество стандартных вариантов проектирования: *плоское* (параллельное проектирование вдоль заданного направления), *цилиндрическое* и *сферическое*. Для параметрически заданных поверхностей часто в качестве проекции точки $[x(u, v), y(u, v), z(u, v)]$ выступают значения параметров (u, v) .

Эти текстуры по своему действию являются цветовыми, т. е. изменяющими цвет в заданной точке. Кроме цветовых существуют также скалярные текстуры, изменяющие один из скалярных параметров, например, коэффициент прозрачности поверхности объекта.

Еще одним типом текстуры является текстура, изменяющая направление вектора нормали в точке. Она служит для моделирования рельефа поверхности («битр»). Аккуратное использование подобных текстур позволяет заметно усилить реалистичность получаемых изображений при сравнительно небольших вычислительных затратах.

Процедурные текстуры не требуют больших затрат памяти и одинаково хорошо работают с объектами любой (сколь угодно сложной) формы. Поскольку подобная функция обычно зависит от большого количества параметров, то, изменяя их, можно легко изменять параметры текстуры. Основным недостатком этого подхода является сложность подбора соответствующей функции.

Рассмотрим создание процедурной текстуры на примере достаточно широкой группы так называемых *шумовых текстур*.

Пусть нужно создать текстуру дерева. Известно, что дерево имеет цилиндрическую структуру (симметрию), направленную, например, вдоль оси Oz . Несложно построить функцию, которая определяет цвет, меняющийся по этому закону, например

$$C(x, y, z) = C_1 + (C_1 - C_2)f(\sqrt{x^2 + y^2}),$$

где C_1 и C_2 — некоторые цвета (светлых и темных колец); $f(t)$ — некоторая неотрицательная периодическая функция, например $\frac{1}{2}(1 + \sin t)$.

Ясно, что изображение, построенное при помощи подобной функции, будет симметричным и слишком правильным. На самом деле деревьев с идеальной структурой колец нет — некоторое случайное искажение присутствует практически всегда.

Для моделирования таких искажений вводится так называемая шумовая функция $\text{Noise}(x, y, z)$. Обычно на шумовую функцию накладываются следующие требования:

- 1) чтобы она была непрерывной функцией;
- 2) принимала значения из отрезка $[0, 1]$ и
- 3) вела себя в некотором смысле аналогично равномерно распределенной случайной величине.

Существует несколько способов построения подобной функции. Простейшим из них является задание случайных значений в узлах некоторой регулярной сетки (например, в точках (i, j, k) , где $i, j, k \in \mathbb{Z}$ — целые числа) и последующая интерполяция на все остальные точки. Кроме шумовой функции Noise довольно часто используется также следующая функция:

$$\text{Turbulence}(p, k) = \sum_{i=1}^k \frac{1}{2^i} \text{Noise}(2^i p).$$

3.3. Атмосферные эффекты, источники освещения

Системы рендеринга современных 3D-пакетов способны создавать сложные эффекты, такие как слоистый туман или светотень при неоднородном освещении, имитирующую прохождение световых лучей через пыль или дым [20]. Предусмотрены даже некоторые методы постобработки видеопоследовательностей, в частности для привнесения в них спецэффектов или оформления переходов между сценами. Можно создать иллюзию искривления пространства, определив невидимые объекты, которые деформируют те, что видимы.

Источники освещения делятся на три базовых типа:

- *Spot* — конус света из определенной точки;
- *Free* — направленный поток параллельных лучей;
- *Omni* — точечный источник, излучающий равномерно во все стороны.

Источники освещения имеют такие характеристики, как интенсивность потока, цвет источника (или даже растровую маску-светофильтр); список объектов, подлежащих исключению из освещения; флаги отбрасывания теней и многие другие.

3.4. Рендеринг, определение нормали к поверхности

В современных программных средах для работы с 3D-графикой реализовано большое количество разнообразных методов визуализации. Регулярно появляются новые методы, совершенствуются старые. Из всего множества методов рендеринга в рамках нашего курса мы рассмотрим следующие базовые, наиболее распространенные методы:

- закраска методом Гуро;
- закраска Фонга;
- трассировка лучей;
- метод излучательности (radiosity).

Реалистичное растровое изображение, геометрического объекта формируется путем определения цвета и яркости его точек при некоторой модели освещенности. В большинстве случаев выполняется построчное сканирование изображения [18]. Для определения отражающей свет точки объекта выполняется пересечение его триангуляционной сетки со сканирующей линией визирования. Эта линия строится из точки изображения перпендикулярно проекционной плоскости (как для параллельной проекции, так и для центральной проекции).

Все методы используют свои оригинальные подходы к расчету интенсивности света в видимых точках сцены, однако есть и общие вопросы, такие как определение нормали к поверхности или нахождение вектора отражения, которые мы рассмотрим предварительно.

Нормаль к поверхности представляет ее локальную кривизну, а следовательно, и направление зеркального отражения. Если известно аналитическое описание поверхности, то нормаль вычисляется непосредственно. Но для многих поверхностей бывает задана лишь их полигональная аппроксимация. Зная уравнение плоскости каждой грани, можно найти направление внешней нормали [21].

Во многих алгоритмах удаления невидимых линий и поверхностей используются только ребра или вершины, поэтому, для того чтобы объединить их с моделью освещения, необходимо знать приближенное значение нормали на ребрах и в вершинах. Пусть заданы уравнения плоскостей полигональных граней, тогда нормаль к их общей вершине равна среднему значению нормалей ко всем многоугольникам, сходящимся в этой вершине. Например, на рис. 16, а направление приближенной нормали в точке V_1 есть

$$\vec{n}_{V_1} = (a_0 + a_1 + a_4)\vec{i} + (b_0 + b_1 + b_4)\vec{j} + (c_0 + c_1 + c_4)\vec{k},$$

где $a_0, a_1, a_4, b_0, b_1, b_4, c_0, c_1, c_4$ — коэффициенты уравнений плоскостей грех многоугольников P_0, P_1, P_4 , окружающих V_1 . Отметим, что если требуется найти только направление нормали, то делить результат на количество граней необязательно (рис. 16, б).

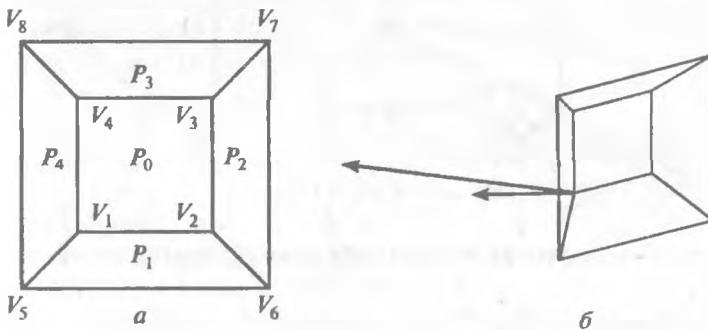


Рис. 16. Алпроксимация нормали к полигональной поверхности

Если же уравнения плоскостей не заданы, то нормаль к вершине можно определить, усредняя векторные произведения всех ребер, пересекающихся в вершине. Еще раз рассматривая вершину V_1 на рис. 16, найдем направление приближенной нормали:

$$n_{V_1} = V_1V_2 \otimes V_1V_4 + V_1V_5 \otimes V_1V_2 + V_1V_4 \otimes V_1V_5.$$

Следует обратить внимание на то, что необходимы только внешние нормали. Кроме того, если полученный вектор не нормируется, то его величина зависит от количества и площади конкретных многоугольников, а также от количества и длины конкретных ребер.

3.5. Определение вектора отражения

Для модели освещения чрезвычайно важно правильно задавать направление векторов отражения. По закону отражения вектор падающего света, нормаль к поверхности и вектор отражения лежат в одной плоскости, причем на этой плоскости угол падения равен углу отражения (рис. 17, а). Фонг вывел отсюда простое решение для случая, когда свет падает вдоль оси z . Это предположение удобно для модели освещения с одним точечным источником. Если начертить системы координат перенести в точку поверхности, то проекции нормали и вектора отражения на плоскость xy будут лежать на одной прямой (рис. 17, б).

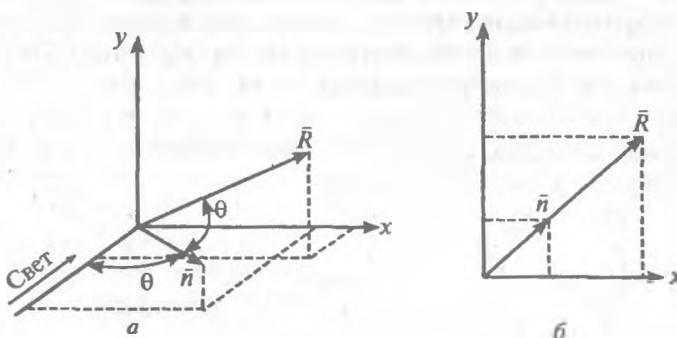


Рис. 17. Расчет направления отражения

Таким образом, $\frac{\bar{R}_x}{\bar{R}_y} = \frac{\bar{n}_x}{\bar{n}_y}$, где $\bar{R}_x, \bar{R}_y, \bar{n}_x, \bar{n}_y$ — суть x - и y -составляющие единичных векторов соответственно отражения и нормали. Отсюда получаем, что:

$$\bar{R}_z = 2\bar{n}_z^2 - 1, \quad \bar{R}_y = 2\bar{n}_z\bar{n}_y, \quad \bar{R}_x = 2\bar{n}_z\bar{n}_x.$$

3.6. Прозрачность и тени

В основных моделях освещения и алгоритмах удаления невидимых линий и поверхностей рассматриваются только непрозрачные поверхности и объекты. Однако существуют и прозрачные объекты, пропускающие свет (такие как стакан, ваза, окно автомобиля, вода). При переходе из одной среды в другую, например, из воздуха в воду, световой луч преломляется; поэтому торчащая из воды палка кажется согнутой. Преломление рассчитывается по закону Снеллиуса, который утверждает, что падающий и преломляющий лучи лежат в одной плоскости, а углы падения и преломления связаны соотношением

$$\eta_1 \sin \theta = \eta_2 \sin \theta',$$

где η_1 и η_2 — показатели преломления двух сред; θ — угол падения; θ' — угол преломления (рис. 18). Ни одно вещество не пропускает весь падающий свет, часть его всегда отражается.

Так же, как и отражение, пропускание может быть зеркальным (направленным) или диффузным. Направленное пропускание свойственно прозрачным веществам, например стеклу. Если смотреть на объект

сквозь такое вещество, то, за исключением контурных линий криволинейных поверхностей, искажения происходить не будет. Если свет при пропускании через вещество рассеивается, то мы имеем диффузное пропускание. Такие вещества кажутся полупрозрачными или матовыми. Если смотреть на объект сквозь такое вещество, то он будет выглядеть нечетким или искаженным. Все это необходимо иметь в виду при создании реалистичных изображений.



Рис. 18. Геометрия преломления

Тени. Если положения наблюдателя и источника света совпадают, то теней не видно, но они появляются, когда наблюдатель перемещается в любую другую точку. Изображение с построенными тенями выглядит гораздо реалистичнее, и, кроме того, тени очень важны для моделирования. Например, особо интересующий нас участок может оказаться невидимым из-за того, что он попадает в тень.

Наблюдения показывают, что тень состоит из двух частей: полутени и полной тени. Полная тень — это центральная, темная, резко очерченная часть, а полутень — окружающая ее более светлая часть. Обычно рассматриваются точечные источники, создающие только полную тень. Распределенные источники света конечного размера создают как тень, так и полутень: в полной тени свет вообще отсутствует, а полутень освещается частью распределенного источника. Из-за больших вычислительных затрат, как правило, рассматривается только полная тень, образуемая точечным источником света. Сложность вычислений зависит и от положения источника. Легче всего, когда источник находится в бесконечности, и тени определяются с помощью ортогонального проецирования. Сложнее, если источник расположен на конечном расстоянии, но вне поля зрения; здесь необходима перспективная проекция. Самый трудный случай, когда источник находится в поле зрения. Тогда необходимо делить пространство на секторы и искать тени отдельно для каждого сектора.

3.7. Простая модель освещения

Световая энергия, падающая на поверхность, может быть поглощена, отражена или пропущена [21]. Частично она поглощается и превращается в тепло, а частично отражается или пропускается. Объект можно увидеть, только если он отражает или пропускает свет; если же объект поглощает весь падающий свет, то он невидим и называется абсолютно черным телом. Количество поглощенной, отраженной или пропущенной энергии зависит от длины волны света. При освещении белым светом, в котором интенсивность всех длин волн снижена примерно одинаково, объект выглядит серым. Если поглощается почти весь свет, то объект кажется черным, а если только небольшая его часть — белым. Если поглощаются лишь определенные длины волн, то у света, исходящего от объекта, изменяется распределение энергии и объект выглядит цветным. Цвет объекта определяется поглощаемыми длинами волн.

Свойства отраженного света зависят от строения, направления и формы источника света, от ориентации и свойств поверхности. Отраженный от объекта свет может также быть диффузным или зеркальным. Диффузное отражение света происходит, когда свет как бы проникает под поверхность объекта, поглощается, а затем вновь испускается. При этом положение наблюдателя не имеет значения, так как диффузно отраженный свет рассеивается равномерно по всем направлениям. Зеркальное отражение происходит от внешней поверхности объекта.

Свет точечного источника отражается от идеального рассеивателя по закону косинусов Ламберта: интенсивность отраженного света пропорциональна косинусу угла между направлением света и нормалью к поверхности, т. е.

$$I = I_l k_d \cos \theta, \quad 0 \leq \theta \leq \frac{\pi}{2},$$

где I — интенсивность отраженного света; I_l — интенсивность точечного источника; k_d — коэффициент диффузного отражения ($0 \leq k_d \leq 1$); θ — угол между направлением света и нормалью к поверхности (рис. 19).

Если $\theta > \frac{\pi}{2}$, то источник света расположен за объектом. Коэффициент диффузного отражения k_d зависит от материала и длины волны света, но в простых моделях освещения обычно считается постоянным.

Поверхность предметов, изображенных при помощи простой модели освещения с диффузным отражением ЛамBERTA, выглядит блеклой и матовой. Предполагается, что источник точечный, поэтому объекты, на которые не падает прямой свет, кажутся черными. Одна-

ко на объекты реальных сцен падает еще и рассеянный свет, отраженный от окружающей обстановки, например от стен комнаты. Рассеянному свету соответствует распределенный источник. Поскольку для расчета таких источников требуются большие вычислительные затраты, в машинной графике они заменяются на коэффициент рассеяния — константу, которая входит в формулу в линейной комбинации с членом Ламберта:

$$I = I_a k_a + I_l k_d \cos\theta, \quad 0 \leq \theta \leq \frac{\pi}{2},$$

где I_a — интенсивность рассеянного света; k_a — коэффициент диффузного отражения рассеянного света ($0 \leq k_a \leq 1$).

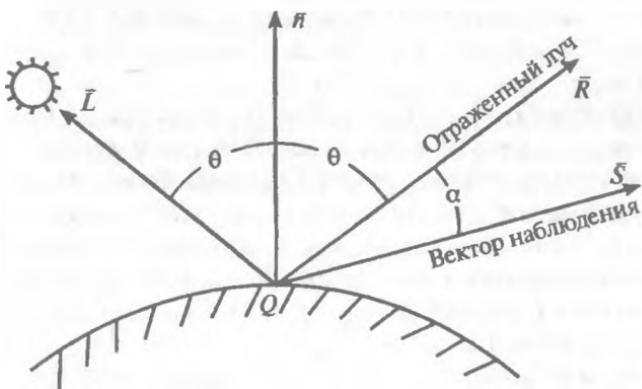


Рис. 19. Диффузное и зеркальное отражения

Пусть даны два объекта, одинаково ориентированные относительно источника, но расположенные на разном расстоянии от него. Если найти их интенсивность по данной формуле, то она окажется одинаковой. Это значит, что когда предметы перекрываются, их невозможно различить, хотя интенсивность света обратно пропорциональна квадрату расстояния от источника, и объект, лежащий дальше от него, должен быть темнее. Если предположить, что источник света находится в бесконечности, то диффузный член модели освещения обратится в ноль. В случае перспективного преобразования сцены в качестве коэффициента пропорциональности для диффузного члена можно взять расстояние d от центра проекции до объекта. Но если центр проекции лежит близко к объекту, то $1/d^2$ изменяется очень быстро, т. е. у объектов, лежащих примерно на одинаковом расстоянии от источника, разница интенсивностей чрезмерно велика. Как показывает опыт, боль-

шой реалистичности можно добиться при линейном затухании. В этом случае модель освещения выглядит так:

$$I = I_a k_a + \frac{I_l k_d \cos \theta}{d + K},$$

где K — произвольная постоянная.

Если предполагается, что точка наблюдения находится в бесконечности, то d определяется положением объекта, ближайшего к точке наблюдения. Это означает, что ближайший объект освещается с полной интенсивностью источника, а более далекие — с уменьшенной. Для цветных поверхностей модель освещения применяется к каждому из трех основных цветов.

Интенсивность зеркально отраженного света зависит от угла падения, длины волны падающего света и свойств вещества. Основное уравнение Френеля приводится в любой книге по геометрической оптике. Зеркальное отражение света является направленным. Угол отражения от идеальной отражающей поверхности (зеркала) равен углу падения, в любом другом положении наблюдатель не видит зеркально отраженный свет. Это означает, что вектор наблюдения \vec{s} (рис. 19) совпадает с вектором отражения \vec{R} , и угол α равен нулю. Если поверхность не идеальна, то количество света, достигающее наблюдателя, зависит от пространственного распределения зеркального отраженного света. У гладких поверхностей распределение узкое или сфокусированное, у шероховатых — более широкое.

Благодаря зеркальному отражению на блестящих предметах появляются световые блики. Из-за того, что зеркально отраженный свет сфокусирован вдоль вектора отражения, блики при движении наблюдателя тоже перемещаются. Более того, так как свет отражается от внешней поверхности (за исключением металлов и некоторых твердых красителей), то отраженный луч сохраняет свойства падающего. Например, при освещении блестящей синей поверхности белым светом возникают белые, а не синие блики.

В простых моделях освещения обычно пользуются эмпирической моделью Буи-Туонга Фонга, так как физические свойства зеркального отражения очень сложны. Модель Фонга имеет вид

$$I_s = I_l w(i, \lambda) \cos^n \alpha,$$

где $w(i, \lambda)$ — кривая отражения, представляющая отношение зеркально отраженного света к падающему как функцию угла падения i и длины волны λ ; n — степень, аппроксимирующая пространственное распределение зеркально отраженного света. На рис. 20 показана функция $\cos^n \alpha$

при $-\pi/2 \leq \alpha \leq \pi/2$ для различных n : большие значения n дают сфокусированные пространственные распределения характеристик металлов и других блестящих поверхностей, а малые — более широкие распределения для неметаллических поверхностей, например бумаги.

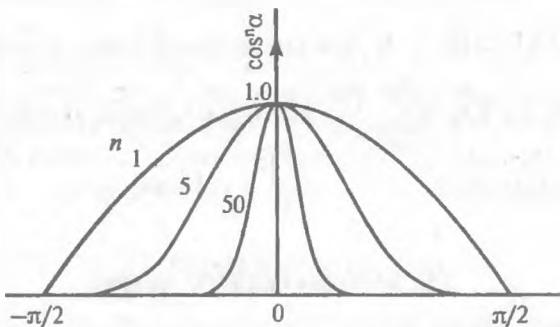


Рис. 20. Приближенные функции пространственного распределения для зеркального отражения

Коэффициент зеркального отражения зависит от угла падения, однако даже при перпендикулярном падении зеркально отражается только часть света, а остальное либо поглощается, либо отражается диффузно. Эти соотношения определяются свойствами вещества и длиной волны. Коэффициент отражения для некоторых неметаллов может быть всего 4 %, в то время как для металлических материалов — более 80 %. Коэффициенты отражения для типичных веществ при нормальном падении света представлены как функции длины волны. При падении под скользящим углом ($\theta = 90^\circ$) отражается весь падающий свет (коэффициент отражения 100 %).

Объединяя эти результаты с формулой рассеянного света и диффузного отражения, получим следующую модель освещения:

$$I = I_a k_a + \frac{I_t}{d+K} (k_d \cos \theta + w(i, \lambda) \cos^n \alpha).$$

Функция $w(i, \lambda)$ довольно сложна, поэтому ее обычно заменяют константой k_s , которая либо выбирается из эстетических соображений, либо определяется экспериментально. Таким образом,

$$I = I_a k_a + \frac{I_t}{d+K} (k_d \cos \theta + k_s \cos^n \alpha).$$

В машинной графике эта модель часто называется *функцией закраски* и применяется для расчета интенсивности или тона точек объекта или пикселов изображения. Чтобы получить цветное изображение, нужно

найти функции закраски для каждого из трех основных цветов. Константа k_s , обычно одинакова для всех трех основных цветов, поскольку цвет зеркально отраженного света определяется цветом падающего.

Если имеется несколько источников света, то их эффекты суммируются.

ОПРЕДЕЛЕНИЕ 12. В этом случае простая модель освещения определяется как $I = I_a k_a + \sum_{j=1}^m \frac{I_j}{d+K} (k_d \cos \theta_j + k_s \cos^n \alpha_j)$, где m — количество источников света.

3.8. Мультимедийное резюме

Создание сложной трехмерной сцены состоит из последовательности трудоемких и высокотехнологичных процессов: начиная с геометрического моделирования объектов сцены и заканчивая расстановкой источников света, камер, выбора модели освещения и рендеринга.

Перед осуществлением финальной визуализации сцены алгоритмы рендеринга решают ряд общих задач, таких как текстурирование, определение нормали к поверхности в местах острых изломов, проецирование теней и моделирование прозрачности.

Эмпирическая модель Фонга позволяет существенно упростить вычислительно емкие калькуляции освещенности точек объекта, основываясь на приближении физических свойств зеркального отражения.

ГЛАВА 4

МЕТОДЫ РЕНДЕРИНГА 3D-СЦЕН

Сканирующая плоскость, метод Гуро. Интерполяция вектора нормали, метод Фонга. Глобальная модель освещенности Уиттеда. Прямая и обратная трассировки лучей. Метод излучательности.

4.1. Закраска методом Гуро

Если при построении полигональной поверхности для каждой грани используется по одной нормали, то простая модель освещения создает изображение, состоящее из отдельных многоугольников. Методом Гуро можно получить более сглаженное изображение [21]. В этом методе по значениям нормалей в вершинах каждого треугольника вначале определяются интенсивности отраженного света в его углах. Нормали к поверхности аппроксимируются в вершинах так, как описано в предыдущей главе. Затем с помощью билинейной интерполяции вычисляется интенсивность каждого пикселя на сканирующей строке.

Рассмотрим, например, участок полигональной поверхности (рис. 21).

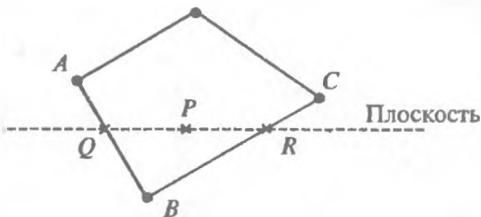


Рис. 21. Сканирующая плоскость

Значение интенсивности в точке P определяется линейной интерполяцией интенсивности в точках Q и R . Для получения интенсивности в точке Q — пересечении ребра многоугольника со сканирующей строкой — нужно линейной интерполяцией интенсивностей A и B найти

$$I_Q = u I_A + (1 - u) I_B \quad 0 \leq u \leq 1, \text{ где } u = AQ/AB.$$

Аналогично для получения интенсивности R линейно интерполируются интенсивности в вершинах B и C , т. е.

$$I_R = wI_B + (1-w)I_C \quad 0 \leq w \leq 1, \text{ где } w = BR/BC.$$

Наконец, линейной интерполяцией по строке между Q и R находится интенсивность P , т. е.

$$I_P = tI_Q + (1-t)I_R \quad 0 \leq t \leq 1, \text{ где } t = QP/QR.$$

Значения интенсивности пикселов вдоль сканирующей строки можно вычислять инкрементально. Для двух пикселов в t_1 и t_2 на сканирующей строке

$$I_{P_2} = t_2 I_Q + (1-t_2)I_R,$$

$$I_{P_1} = t_1 I_Q + (1-t_1)I_R.$$

Вычитая, получим, что вдоль строки

$$I_{P_2} = I_{P_1} + (I_Q - I_R)(t_2 - t_1) = I_{P_1} + \Delta I \Delta t.$$

При применении закраски Гуро к полигональной аппроксимации качество изображения неплохое, но возможно проявление эффекта полос Маха. Это происходит потому, что такой метод интерполяции обеспечивает лишь непрерывность значений интенсивности вдоль границ многоугольников, но не обеспечивает непрерывности изменения интенсивности.

Закраска Гуро лучше всего выглядит в сочетании с простой моделью с диффузным отражением, так как форма бликов при зеркальном отражении зависит от выбора многоугольников, представляющих объект или поверхность.

4.2. Закраска Фонга

Закраска Фонга требует больших вычислительных затрат, однако она позволяет разрешить многие проблемы метода Гуро. При закраске Гуро вдоль сканирующей строки интерполируется значение интенсивности, а при закраске Фонга — вектор нормали. Затем он используется в модели освещения для вычисления интенсивности пикселя. При этом достигается лучшая локальная аппроксимация кривизны поверхности и, следовательно, получается более реалистичное изображение. В частности, правдоподобнее выглядят зеркальные блики.

При закраске Фонга нормали в вершинах имеют свое истинное направление, а в пределах каждого треугольника принимается линейный закон их изменения от вершины к вершине. После этого билинейной интерполяцией вычисляется нормаль в каждом пикселе.

Несмотря на то, что векторы нормалей в вершинах нормированы, интерполируемый вектор в общем случае не имеет единичной длины, и его следует нормировать [18].

Хотя метод Фонга устраняет большинство недостатков метода Гуро, он тоже основывается на линейной интерполяции. Поэтому в местах разрыва первой производной интенсивности заметен эффект полос Маха, хотя и не такой сильный, как при закраске Гуро. Однако иногда этот эффект проявляется сильнее у метода Фонга, например для сфер. Кроме того, оба метода могут привести к ошибкам при изображении невыпуклых многоугольников.

Также возникают трудности, когда любой из этих методов применяется при создании видеопоследовательности. Например, закраска может значительно изменяться от кадра к кадру. Это происходит из-за того, что правило закраски зависит от поворотов, а обработка ведется в пространстве изображения. Поэтому, когда от кадра к кадру меняется ориентация объекта, его закраска (цвет) тоже изменяется, причем достаточно заметно. Существуют модификации методов закраски Гуро и Фонга, инвариантные относительно поворота.

4.3. Глобальная модель освещения с трассировкой лучей

Одним из наиболее распространенных и наглядных методов построения реалистических изображений является метод трассировки лучей, позволяющий строить фотoreалистичные изображения сложных сцен с учетом таких эффектов, как отражение и преломление. Отличительной чертой метода является его крайняя простота и наглядность [11].

Название «трассировка лучей» происходит от слова «путь» (трасса), что связано с моделированием геометрического пути каждого светового луча, участвующего в построении изображения [22].

Ключевая задача метода трассировки лучей — определение освещенности произвольной точки объекта и той части световой энергии, которая уходит в заданном направлении. Эта энергия складывается из двух частей — непосредственной (первичной) освещенности, т. е. энергии, непосредственно получаемой от источников света, и вторичной освещенности, т. е. энергии, идущей от других объектов.

Введем некоторые ограничения на рассматриваемую сцену:

- будем рассматривать только точечные источники света;
- при трассировке преломленного луча будем игнорировать зависимость его направления от длины волны;

- будем считать освещенность объекта состоящей из диффузной и зеркальной частей (с заданными весами).

Через каждый пиксел экрана луч трассируется до ближайшего пересечения с объектами сцены. Из точки пересечения выпускаются лучи ко всем источникам света для проверки их видимости и определения непосредственной освещенности точки пересечения. Выпускаются также отраженный и преломленный лучи, которые трассируются, в свою очередь, до ближайшего пересечения с объектами сцены, и так далее. Получается рекурсивный алгоритм обратной трассировки лучей.

Отдельно остановимся на модели освещения. Она предназначена для того, чтобы рассчитать интенсивность отраженного к наблюдателю света в каждой точке (пикселе) изображения. Модель может быть локальной или глобальной. В первом случае во внимание принимается только свет, падающий от источника (источников), и ориентация поверхности. Во втором учитывается также свет, отраженный от других объектов сцены или пропущенный сквозь них [20].

Глобальная модель воспроизводит чрезвычайно важные эффекты; некоторые из них показаны на рис. 22.

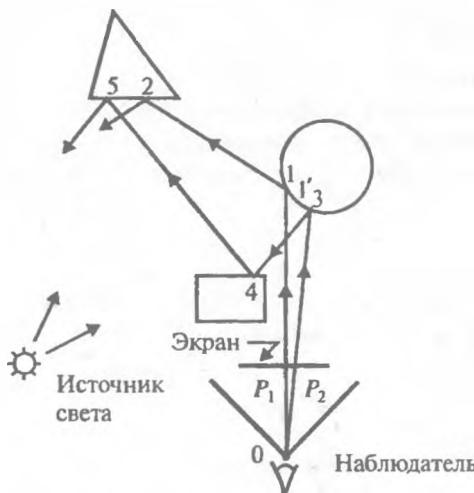


Рис. 22. Глобальное освещение

Сфера, треугольная призма и параллелепипед на рисунке непрозрачны, их поверхность зеркальна. Наблюдатель смотрит из точки 0 на точку 1 на сфере. При этом он видит не только сферу, но и точку 2 на

призме. Призма, загороженная от наблюдателя параллелепипедом, становится видимой из-за отражения в сфере. Точка 5 на призме видима еще более косвенно. Она отражается от обратной стороны параллелепипеда в точке 4 к точке 3 на сфере, а затем — к наблюдателю. Кроме того, наблюдатель видит точку 5 в точке $5'$ после одного отражения от сферы, поэтому на сфере находятся два изображения призмы. Изображение, расположенное вокруг точки $5'$, перевернуто, так как получается после одного отражения; вокруг точки 3 не перевернуто, так как получается после двух отражений. Интенсивность второго изображения меньше. Наконец, в сфере отражается обратная сторона параллелепипеда, т. е. наблюдатель видит ее, хотя на нее не падает прямой свет источника. Эта сторона освещается рассеянным светом и светом, отраженным от других объектов сцены.

Отсюда следует, что обычная операция отбрасывания задних граней, применяемая при удалении невидимых поверхностей, здесь не годится, как и предварительная сортировка по глубине. Поэтому из всех алгоритмов удаления невидимых граней остается только метод трассировки лучей. Таким образом, глобальная модель освещения является частью алгоритмов выделения видимых поверхностей путем трассировки лучей. Первые работы в этой области принадлежат Уиттеду и Кэю. Алгоритм Уиттеда более общий, он был расширен и часто используется.

Уиттед пользуется моделью с такими же членами рассеянного и ламбертовского диффузного отражения, а также зеркального отражения Фонга, как и в локальной простой модели освещения, рассмотренной в предыдущей главе. Члены, соответствующие глобальному зеркальному отражению и пропусканию, рассчитываются по правилу, показанному на рис. 23.

Здесь трассируется луч \bar{v} , падающий на поверхность в точке Q . В этой точке он отражается в направлении r и, если поверхность прозрачна, преломляется в направлении \bar{p} . Здесь I_s — интенсивность света, падающего в точку Q по направлению \bar{p} . Этот свет преломляется и достигает наблюдателя, находящегося в направлении $-\bar{v}$.

Аналогично I_s — интенсивность зеркально отраженного света, падающего в направлении $-\bar{r}$ и отраженного к наблюдателю в точке Q ; \bar{n} — нормаль к поверхности в точке Q ; \bar{L}_j — направление на j -й источник света; \bar{S} и \bar{R} — локальные векторы наблюдения и отражения, η — показатель преломления среды, n — степень пространственного распределения Фонга для зеркального отражения.



Рис. 23. Зеркальное отражение и преломление в глобальной модели освещения Уиттеда

Тогда наблюдаемая интенсивность I выражается формулой

$$I = k_a I_a + k_d \sum_j I_{l_j} (\bar{n} \times \bar{L}_j) + k_s \sum_j I_{l_j} (\bar{S} \times \bar{R}_j)^n + k_s I_s + k_t I_t,$$

где k_a, k_d, k_s — коэффициенты рассеянного, диффузного и зеркального отражения, а k_t — коэффициент пропускания. Уиттед считает их постоянными, однако с помощью любой из рассмотренных выше моделей освещения можно найти зависимость этих коэффициентов от угла падения и длины волны. Две суммы по j — это диффузное и зеркальное отражение от множества источников.

В глобальной модели освещения предполагается, что падающий луч v в точке Q отражается в направлении r и пропускается сквозь поверхность в направлении \bar{p} , как показано на рис. 23. Это значит, что в точке Q образуются еще два луча, для которых нужно найти все пересечения с объектами сцены. Т. е. для каждого луча \bar{v} строится соответствующее дерево, состоящее из ветвей отражения и преломления. Такой процесс повторяется, пока не останется ни одного пересечения. Ветвь кончается, когда луч покидает сцену.

Для того чтобы определить интенсивность в каждом пересечении луча с поверхностью, надо пройти дерево в обратном направлении. Ин-

тенсивность в узлах рассчитывается в соответствии с моделью освещения, причем для каждого следующего узла она ослабляется с расстоянием между точками пересечения. После прохода всего дерева получается окончательная интенсивность пикселя.

Теоретически дерево трассировки луча может быть бесконечно глубоким. В качестве критерия остановки обычно используется отсечение по глубине (не более заданного количества уровней рекурсии) и по весу (чем дальше, тем меньше вклад каждого луча в итоговый цвет пикселя, и, как только этот вклад опускается ниже некоторого порогового значения, дальнейшая трассировка этого луча прекращается).

4.4. Распределенная трассировка лучей

Необходимо отметить, что ряд изображений, построенных обычными методами обратной трассировки лучей, несут в себе существенные погрешности, визуально наиболее заметными из которых являются «лестничные» линии (рис. 24) и пропадающие точки.

Эти характерные линии для классической трассировки лучей носят названия *«aliasing defects»*. Они связаны с тем, что каждый пиксель фактически трактуется как бесконечно малая точка на регулярной сетке, хотя на самом деле пиксель является прямоугольной областью, и его цвет определяется путем суммирования по всем лучам, проходящим через эту область, т. е. является интегралом по этой области.

Одним из наиболее распространенных и мощных средств, используемых для борьбы с этими эффектами, является так называемая *распределенная трассировка лучей* (*distributed ray tracing*). При этом для вычисления соответствующего интеграла

Рассмотрим случайную точку, равномерно распределенную в области пикселя, и оттрассируем луч, проходящий через эту точку. Тогда освещенность, приносимая таким лучом, является случайной величиной и ее математическое ожидание — соответствующим интегралом. Поэтому для вычисления цвета пикселя достаточно отследить несколько равномерно распределенных случайных лучей и взять среднее значение.

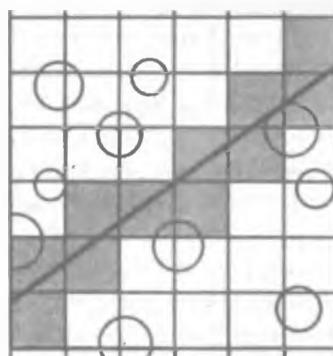


Рис. 24. Лестничный эффект

Основным недостатком метода распределенной трассировки лучей является заметное увеличение объема вычислений. Поэтому на практике применяются аддитивные методы, ограничивающие количество выпускаемых первичных лучей. Все эти методы основаны на статистическом анализе значений, принесенных по уже оттрассированным лучам для данного пикселя (а иногда и соседних).

Этот метод позволяет полностью избавиться от aliasing-погрешностей, при этом в изображение вносится высокочастотный шум, гораздо менее заметный для глаз, чем исходные погрешности. Распределенная трассировка лучей позволяет также моделировать целый ряд дополнительных эффектов, таких, как неточечные источники света, нечеткие отражения, глубина резкости и другие.

4.5. Методы оптимизации трассировки лучей

Метод трассировки лучей отличает высокий объем вычислений, причем по оценкам, до 95 % работы в сложных сценах уходит на проверки пересечения луча с объектами сцены. Для реальных сцен, содержащих многие тысячи и десятки тысяч объектов, простой перебор для определения ближайшей точки пересечения луча с объектами сцены просто неприемлем.

Существуют методы, позволяющие заметно сократить для каждого луча количество проверяемых объектов. Сделаем краткий обзор этих методов.

Предположим, что в рассматриваемой сцене имеется сложный объект. Поместим его внутрь достаточно простой выпуклой фигуры (например сферы). Ясно, что если луч не пересекает эту фигуру, то он не сможет пересечь и охватываемый ею сложный объект. Более того, построенную вспомогательную фигуру пересечет лишь небольшая часть рассматриваемых лучей и только эти лучи нужно проверить на наличие пересечения со взятым объектом сцены. Это означает, что подобная двухэтажная проверка оказывается заметно проще, чем непосредственная проверка пересечения лучей со сложным объектом:

- поиск точек пересечения со сферой прост и алгоритмически легко реализуем;
- количество лучей, которые нужно проверить на пересечение с исходным объектом, становится значительно меньше исходного.

Тем самым выигрыш очевиден.

Можно пойти дальше — описать простую фигуру сразу вокруг нескольких объектов. Тогда если луч не пересекает ограничивающую фи-

туру, то он не сможет пересечь ни одного из содержащихся внутри объектов.

Следующий шаг — оптимизация случая, когда луч все-таки пересекает ограничивающую фигуру. Вначале рассмотрим метод дерева ограничивающих фигур.

Разобьем содержащиеся внутри фигуры объекты на группы и вокруг каждой из них опишем новую фигуру, содержащуюся в исходной. Это позволит вместо непосредственной проверки содержащихся внутри объектов проверить сначала ограничивающие их фигуры (объемы).

Продолжая подобный процесс, приходим к следующей организации сцены: вся сцена содержится внутри некоторой фигуры B_1 , все объекты разбиты на несколько групп, и вокруг каждой из них описано по ограничивающей простейшей фигуре B_{11}, \dots, B_{1n_1} , все объекты, содержащиеся внутри каждой из этих фигур, снова разбиты на группы, вокруг каждой из которых описано по ограничивающей фигуре и т. д.

В результате приходим к древовидной организации сцены. В качестве ограничивающих тел обычно выбирают сферы или пересечение полупространств. Количество проверок на пересечения для данного метода составляет $O(\log n)$, где n — общее количество объектов в сцене.

Основным недостатком метода дерева ограничивающих фигур является необходимость построения структуры ограничивающих фигур, что зачастую представляет собой значительные сложности.

В отличие от метода дерева ограничивающих фигур (объемов), разбивающего объекты на группы, метод разбиения пространства проводит разбиение самого пространства. Рассмотрим простейший вариант этого метода — метод равномерного разбиения пространства.

Опишем вокруг всей сцены прямоугольный параллелепипед и разобьем его на $n_1 \times n_2 \times n_3$ равных частей. Для каждой из этих частей составим список всех объектов, границы которых пересекают данный блок.

Процесс нахождения ближайшего пересечения луча с объектами сцены начинается с определения части, содержащей начало луча, и двух списков — списка уже проверенных объектов и списка найденных точек пересечения, отсортированный по расстоянию до начала луча.

Проверим все объекты из списка текущего блока, которые еще не были проверены. После этого добавим проверенные объекты в список уже проверенных, а все найденные пересечения — в список точек пересечения. Если ближайшая точка пересечения из списка найденных

содержится в текущем блоке, то она возвращается как искомое пересечение. В случае если в текущем блоке нет пересечений, находится следующий блок, через который проходит луч, и для него повторяется та же процедура. В случае если следующего блока нет (луч выходит из сцены), алгоритм возвращается отсутствие пересечений.

К несомненным преимуществам относится простота разбиения, возможность использования алгоритма Брезенхайма для нахождения следующего блока и направленный перебор вдоль луча, когда найденное пересечение гарантированно является ближайшим. При соответствующем выборе шагов разбиения среднее количество проверяемых объектов практически не зависит от общего количества объектов в сцене.

4.6. Метод излучательности (Radiosity)

Основными недостатками метода трассировки лучей являются неэффективность работы с диффузными поверхностями, а также то, что определение освещенности поверхностей проводится параллельно с построением изображения и зависит от положения наблюдателя. Поэтому любое изменение положения наблюдателя ведет к полному пересчету всей сцены.

Метод излучательности устраняет эти недостатки, обеспечивая одновременно и высокую точность при работе с диффузными объектами, и отдельное вычисление глобальной освещенности независимо от положения наблюдателя.

В основе метода излучательности лежит закон сохранения энергии в замкнутой системе. Все объекты разбиваются на фрагменты и для этих фрагментов составляются сравнения баланса энергии.

Пусть все объекты являются чисто диффузными, т. е. отражают (рассекают) свет равномерно по всем направлениям. Разобьем всю сцену на n фрагментов и обозначим:

B_i — энергия, отбрасываемая i -м фрагментом сцены;

E_i — собственная излучательность фрагмента;

F_{ij} — доля энергии j -го фрагмента, попадающая на i -й фрагмент (коэффициенты формы);

ρ_i — коэффициент отражения.

Тогда уравнения баланса энергии имеют вид

$$B_i = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j, \quad i = 1, \dots, n.$$

Эти соотношения можно переписать в следующей форме:

$$\sum_{j=1}^n (\delta_{ij} - \rho_i F_{ij}) B_j = E_i, \quad i = 1, \dots, n,$$

или в матричном виде $(I - \rho F)B = E$, где I — единичная матрица.

В результате получаем систему линейных алгебраических уравнений. Из закона сохранения энергии следует, что

$$\sum_{j=1}^n F_{ij} < 1$$

для всех i . Тем самым эта линейная система обладает так называемым диагональным преобладанием, что позволяет использовать для ее решения эффективные итерационные методы (типа Гаусса — Зейделя), дающие за небольшое число итераций вполне удовлетворительное решение.

Соответствующую последовательность приближений к решению можно построить, например, по следующим формулам:

$$B_i^{(0)} = E_i,$$

$$B_i^{(k+1)} = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j^{(k)}.$$

Итерационный процесс прекращается, когда разница между двумя последовательными приближениями оказывается меньше заданной точности.

Для определения цвета фрагмента соответствующие линейные системы записываются для каждой из трех основных цветовых составляющих, причем коэффициенты формы определяются только геометрией сцены и от цвета не зависят.

Обычно после определения освещенности каждого фрагмента производится билинейная интерполяция освещенности по всем объектам, дающая естественное плавное освещение.

После выбора точки наблюдения объекты сцены проектируются на поверхность экрана и строится изображение.

Наиболее трудоемким шагом метода излучательности является вычисление коэффициентов формы, хранящих в себе информацию о геометрии сцены. Рассмотрим этот процесс подробнее.

Выберем фрагменты A_i и A_j и элементарные участки dA_i и dA_j , на них с нормалями соответственно \bar{n}_i и \bar{n}_j (рис. 25). Тогда доля энергии элемента dA_j , попадающей на элемент dA_i , будет равна

$$F(dA_j, dA_i) = \frac{\cos \varphi_i \cos \varphi_j}{\pi r^2},$$

где r — расстояние между элементами dA_i и dA_j ; φ_i и φ_j — углы между нормалями к ним и соединяющим их отрезком.

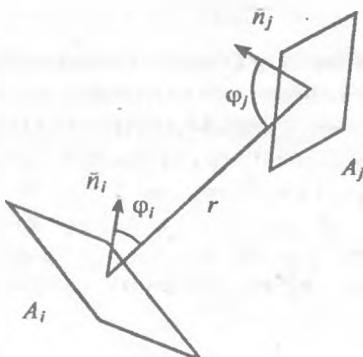


Рис. 25. Элементарные участки метода излучательности

В результате двойного интегрирования получаем следующее соотношение:

$$F_{ij} = F(A_i, A_j) = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r^2} dA_i dA_j.$$

Легко видеть, что

$$A_i F_{ij} = A_j F_{ji}.$$

Интегральная формула не учитывает объектов, закрывающих часть одного фрагмента от другого. Для их учета в подынтегральное выражение нужно добавить еще один множитель — функцию HID_{ij} , принимающую значения из отрезка $[0, 1]$ и характеризующую степень видимости точки фрагмента A_i из точки фрагмента A_j :

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cos \varphi_j}{\pi r^2} HID_{ij} dA_j dA_i.$$

Точное вычисление этого интеграла, как правило, представляет значительные трудности. Поэтому для отыскания коэффициентов формы используется ряд достаточно эффективных приближенных методов.

Для сокращения объема вычислений иногда используется *метод прогрессивной излучательности*, в котором также строится последовательность приближенных решений, но на этот раз коэффициенты формы вычисляются не все сразу, а только по мере необходимости.

Для получения очередного приближения выбирается фрагмент A_i с наибольшей излучательностью, вычисляются коэффициенты формы от него ко всем остальным фрагментам и затем излучательности всех этих фрагментов корректируются по формуле

$$B_i^{(k+1)} = E_j + \rho_j F_{ij} B_j^{(k)}, \quad j = 1, 2, \dots, n.$$

После того как искомые значения B_i найдены, строится изображение сцены с использованием какого-либо метода удаления невидимых поверхностей, например z -буфера. При этом значения излучательности обычно интерполируются для получения плавного перехода освещенности между отдельными фрагментами.

4.7. Мультимедийное резюме

Рассмотренные в данной главе методы рендеринга являются базовыми алгоритмами для многих пакетов 3D-графики. Несмотря на относительную простоту вычислений, методы Гуро и Фонга пользуются популярностью в приложениях реального времени, когда необходимо очень быстро (пусть и не очень качественно) визуализировать 3D-сцену.

По сравнению с рассмотренной в предыдущей главе простой моделью освещенности, глобальная модель освещения Уиттеда дополнительно способна учитывать такие физические свойства объектов как прозрачность, преломление и отражение от внутренних стенок объектов, а также является частью алгоритмов выделения видимых поверхностей путем трассировки лучей.

Сами алгоритмы обратной трассировки лучей, хотя и отличаются высоким качеством синтезируемого изображения, но выделяются среди прочих методов высокой вычислительной нагрузкой. Поэтому для оптимизации расчетов разработаны различные способы оптимизации вычислений: метод дерева ограничивающих фигур, метод равномерного разбиения пространства и другие.

Один из самых качественных методов рендеринга — это, бесспорно, метод излучательности. Основанный на законе сохранения энергии в замкнутой системе, он способен визуализировать наиболее сложные зеркальные отражения и атмосферные эффекты.

ГЛАВА 5

ПРОГРАММНЫЕ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

*Сфера применения. Техника панорамной съемки. Аватары.
Основы VRML. Интерактивность виртуальных объектов.*

В предыдущих главах мы уже касались понятия «виртуальная реальность» (*Virtual Reality*), не вводя четкого определения.

ОПРЕДЕЛЕНИЕ 13. Системой виртуальной реальности называется комплекс программно-аппаратных средств, создающих эффект погружения пользователя в искусственно созданную трехмерную среду.

Эффект погружения заключается не только в ощущении человеком своего пассивного присутствия в виртуальной сцене. Главным условием для достижения максимально полноценного эффекта является *возможность интерактивного взаимодействия* пользователя с окружающей его виртуальной средой.

На сегодняшний день известно достаточно много областей, в которых используются VR-технологии: визуализация, моделирование, перемещение в виртуальном пространстве при поиске информации (навигация), развлечения и игры. Виртуальная реальность помогает инженерам и ученым визуально представить сложные системы в тех случаях, когда создать реальный объект очень сложно или дорого. Работа с визуальной моделью составляет основу имитационного моделирования. Визуализация и моделирование — важные области практического применения VR-технологий, но они менее ориентированы на Internet, чем навигация, воспроизведение присутствия и игровые среды.

Помимо игровых виртуальных сред, одной из наиболее популярных областей применения программных технологий виртуальной реальности являются интерактивные WEB-приложения:

- виртуальные руководства пользователя с 3D-разделами;
- конструкторские и инженерные приложения со сложным моделированием;
- курсы дистанционного обучения;
- сценарное 3D-моделирование происходящих событий;
- виртуальные экскурсии по городам, музеям, замкам;

- демонстрация продающейся или сдающейся в наем недвижимости;
- дизайн внутренних интерьеров;
- бизнес-приложения и т. п.

Для создания виртуальных сред в среде Internet используется широкий спектр программных средств: запатентованные VR-технологии; языки VRML, X3D, 3DMLW; системы подготовки панорамных изображений, например, QuickTime Virtual Reality или Microsoft Surround Video.

5.1. Панорамные видеотехнологии

Панорамные видеотехнологии создают виртуальную среду, в которой угол обзора составляет 360 градусов за счет склейки отдельных изображений в круговую панораму. Искажения при склейке автоматически корректируются программным путем. В зависимости от метода съемки изображение на экране охватывает обычно примерно 70 градусов круговой панорамы. Повороты осуществляются простым перемещением курсора к краю экрана, а перемещение — указанием и нажатием кнопки мыши. При этом горизонтальное панорамирование неограничено, а вертикальное смещение определяется снятым фотокамерой изображением.

Для получения качественных исходных изображений требуется специальный круговой штатив с автоматизацией шага съемки или вращающаяся панорамная камера, принцип действия которой — аналог непрерывной съемки изображения на полоску пленки, протягиваемой в направлении, обратном направлению вращения камеры (рис. 26).

Индикатор угла на вращающейся головке обеспечивает камере большую устойчивость и сбалансированность во время работы. Встроенный уклонометр определяет горизонтальность текущей позиции относительно предыдущей, обеспечивая качественную передачу окружающего мира.

Чтобы можно было рассматривать конкретные объекты с любой степе-



Рис. 26. 360-градусная камера от Cedric Tay

нью детализации и с любого угла, они фотографируются или рендерятся отдельно и компонуются с общей панорамой. При щелчке мышью на таком объекте открывается специальное окно, и объект заполняет весь экран. Это позволяет реализовать интересные возможности. Например, пользователь может прогуливаться по музею, «брать» отдельные экспонаты и рассматривать их.

Типичным примером использования панорамных технологий являются системы интерактивного кругового обзора в компьютерных играх жанра «quest». Как правило, они реализуются с помощью собственных программных решений, хотя для достижения быстрого результата можно воспользоваться упомянутыми выше технологиями от Apple или Microsoft.

5.2. Язык VRML, аватары

VRML — Virtual Reality Modeling (Markup) Language, язык моделирования виртуальной реальности [23]. Во-первых, VRML — это стандартный формат файлов для визуализации трехмерной интерактивной векторной графики, пока еще широко используемый в среде Internet, несмотря на стагнацию стандарта в последнее десятилетие и появление более технологичных стандартов, таких как X3D или 3DMLW.

Во-вторых, это собственно язык описания трехмерных сцен. В нем определены наиболее распространенные в 3D-приложениях семантические сущности, а именно: иерархические преобразования, источники света, возможность произвольной смены точки наблюдения, геометрические тела, анимация, различные свойства материалов и атмосферных сред [24]. Например, вершины и грани многогранников могут указываться вместе с цветом поверхности, текстурами, блеском, прозрачностью и т. д.

В-третьих, VRML — это простой мультиплатформенный текстовый язык, обеспечивающий технологию для интеграции двумерных и трехмерных объектов, текста и мультимедийных данных в единой согласованной модели виртуального мира. Ссылки на другие WEB-ресурсы могут связываться с графическими объектами, и при щелчке пользователя на объекте происходит переход по новому адресу. Мультимедийные данные (видеоролики, звуки, эффекты освещения и т. п.) могут активироваться как реакция на действия пользователя или же на другие внешние события, например таймеры. Также возможно добавление вызова внешнего программного Java-кода из VRML-файла.

В качестве удачного развития технологий, базирующихся на VRML, следует упомянуть о продуктах отечественной компании Parallel Graphics (Paragraph). В линейке ее пользующейся заслуженным успехом продукции — визуальные VRML-редакторы и системы научной и промышленной визуализации, такие как 3D-визуализация авиационных деталей для поддержки инженеров компании Boeing по всему миру через сеть Internet. На рис. 27 приведен пример использования VRML-технологий при коллективном проектировании через сеть Internet обстановки на рабочем месте сотрудников..

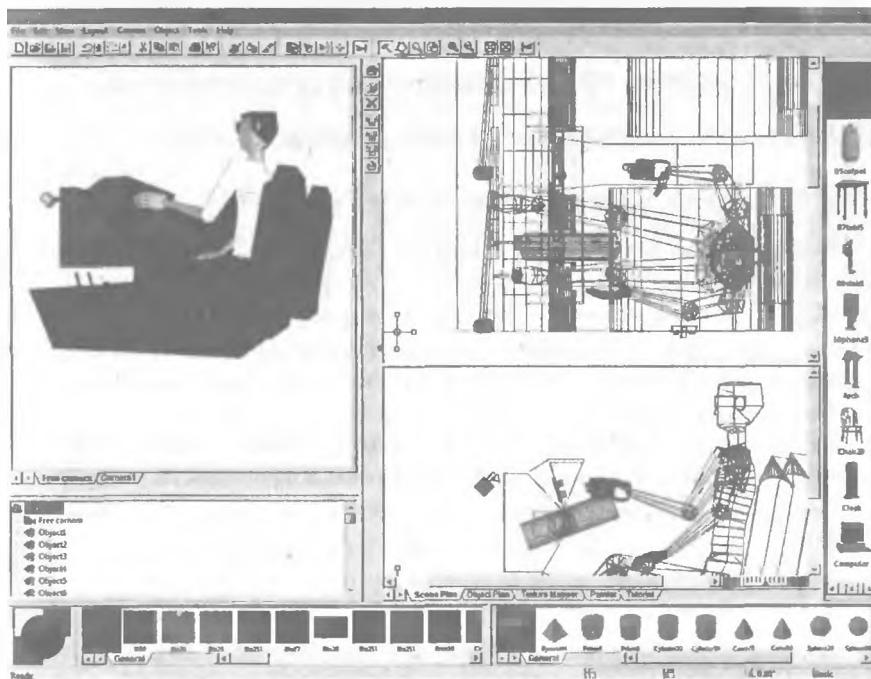


Рис. 27. VRML-среда Parallel Graphics для университета Нюрнберга

Наиболее спорной является концепция, согласно которой VRML является основой киберпространства и виртуального сообщества. Критики этой концепции справедливо указывают на отсутствие как минимум поддержки сетевых протоколов и протоколов баз данных, необходимых для многопользовательских симуляций и виртуальных встреч аватаров.

Термин «аватар» образован от слова «аватара» индуистской религиозной традиции, как правило, обозначающего любое из воплощений многоликого бога Вишну, и трактуется в контексте мультимедийных технологий как проводник по виртуальному миру.

Аватар представляет собой анимированный объект VRML-графики, обычно с человеческими чертами. Может управляться как пользователем («виртуальный персонаж» [25]), так и компьютером («электронный гид»). В первом случае подразумевается свободное двухстороннее общение, во втором случае, интерфейс взаимодействия с гидом сильно ограничен запрограммированными сценариями развития диалога.

5.3. Структура VRML. Создание и воспроизведение сцен

Рассмотрим типовые средства работы с VRML-сценами.

1. Трехмерные среды для создания VRML-сцен.

Большинство программ трехмерного моделирования могут сохранять объекты и сцены в формате VRML. Также существует специализированное программное обеспечение, прежде всего это продукт Internet Space Builder (ISB) от фирмы Parallel Graphics. В ISB неплохо сочетаются простота обращения и возможности построения трехмерных сцен с их сохранением в формате VRML.

VRML-файлы часто называются мирами и имеют расширение .wrl (например, submarine.wrl). Встречаются текстовые файлы и файлы, сжатые с использованием алгоритма компрессии gzip для уменьшения размера.

2. Средства визуализации VRML-сцен.

Визуализация VRML-сцен работает в соответствии с установками конкретного WEB-сервера. Клиентский браузер посыпает запросы на WEB-сервер, который должен возвратить запрошенный документ в формате стандарта MIME (Multipurpose Internet Mail Extension). Если сервер сконфигурирован правильно, то он обнаружит, что VRML-файл имеет тип «x-world/x-vrml». Браузер анализирует эту MIME-информацию и с помощью специальных подключаемых модулей (helper application, plug-in) воспроизводит VRML-сцены [26]. Основными подключаемыми к браузеру модулями являются Cortona от Parallel Graphics и CosmoPlayer от CosmoSoftware.

3. Прямое текстовое кодирование на языке VRML.

Файл VRML состоит из следующей последовательности основных функциональных блоков: заголовка, описания прототипов, описания трехмерной сцены, маршрутов событий.

Для упрощения идентификации файла VRML, каждый файл начинается со строки:

```
#VRML V2.0 <кодировка> [комментарий] <перенос строки>
```

Элементы заголовка внутри угловых скобок являются обязательными (все элементы заголовка до поля <кодировка> должны быть разделены только одним пробелом). Обычно в этом поле указывают «utf8», т. е. кодировку Unicode. Элементы в квадратных скобках можно использовать при необходимости. После кодировки может сразу идти перенос строки или любое количество разделителей в виде пробелов или табуляций, после них текст комментария и, как обязательный элемент, <перенос строки>.

В языке VRML каждая строка, начинающаяся со знака «#», считается комментарием и игнорируется браузером. Данный символ имеет значение только для заголовка файла [27].

Пример заголовка:

```
#VRML V2.0 utf8 «Пример заголовка»
```

или

```
#VRML V2.0 utf8  
# Пример заголовка  
# и комментария
```

После данного заголовка идут описания трехмерных объектов, их параметров и свойств, а также события и их обработчики для создания динамики в виртуальном мире.

Существует три версии языка VRML. Версия 1.0 была создана на базе разработок компании Silicon Graphics и практически не используется. Позже появилась версия 2.0, которая имела гораздо больше возможностей. Именно эту версию языка разработчики отправили на стандартизацию в ISO/IEC. После этого была разработана новая версия языка — VRML97. Она мало отличается от 2.0 и содержит лишь небольшие изменения и поправки.

5.4. Основные понятия языка

1. Имена.

Как и в любом другом языке, в VRML можно использовать имена для различных объектов. Имена могут быть практически любыми, но на них все же накладываются некоторые ограничения:

- имена в языке регистра-зависимые (так же, как в C++);
- в именах можно использовать только первую половину кодовой таблицы (т. е. имена, состоящие из русских букв использовать нельзя);
- нельзя использовать управляющие символы от 0 до 32;
- имена не должны содержать синтаксических символов, которые используются в языке. К этим символам относятся # , . [\] { } () ' " ;
- имена не должны начинаться с цифр.

2. Узлы.

Под объектами виртуального мира понимаются не только все трехмерные тела в нем, но и любые мультимедийные сообщения — звуки, изображения, анимированные ссылки. Все такие объекты в VRML носят название **узлы** (*nodes*). Связано это с тем, что весь трехмерный мир представляет собой дерево объектов. Всего в VRML97 определено 54 типа узлов.

Все узлы делятся на три типа:

- 1) группировочные;
- 2) конечные;
- 3) вспомогательные.

Группировочные узлы предназначены для объединения вложенных узлов в ветви так называемого «дерева сцены». Группировочные узлы называются *предком* (родителем), а подчиненные ему узлы — *потомками* (детьми).

Конечные узлы определяют конкретные геометрические примитивы, звуки, видео, камеры. Данные узлы не могут иметь вложенных узлов-детей, за исключением специфических вспомогательных.

Вспомогательные узлы — это специфические узлы, располагающиеся только в определенных узлах. К таким узлам можно отнести *сенсоры* (см. ниже), узлы-интерполяторы, которые меняют объекты сцены в соответствии с заданным набором опорных точек, и некоторые другие.

3. Поля и события.

Любой объект в VRML имеет набор уникальных параметров, отличающих данный объект от других аналогичных. К примеру, у двух кубов могут различаться параметры «размер» и «цвет». Такие параметры объектов называются *полями* (field) узлов. У каждого узла в VRML существует определенный набор полей. Каждое поле имеет свой тип и значение по умолчанию. При описании объекта сцены можно задавать только требующиеся поля. Все остальные неописанные поля будут использовать значения по умолчанию. Описание узла имеет следующий вид:

```
Тип_узла {  
    поле_1 значение  
    поле_2 значение  
    ...  
    поле_N значение  
}
```

Кроме полей узел может иметь набор событий. Под *событиями* подразумевается средство обмена информацией между узлами. Узел может получать информацию с помощью входящих (eventIn) событий и передавать информацию с помощью исходящих (eventOut) событий. По входящим событиям узел может определять изменения в трехмерном мире и менять свое состояние по заданному алгоритму. При этом о любых своих изменениях узел сообщает миру через исходящие события.

Для того чтобы объекты мира могли изменять свои параметры, практически каждый узел в VRML имеет набор изменяемых полей, называемых *регулируемыми полями* (exposedField). С каждым таким полем связано два события: входящее и исходящее. Если поле имеет название Abcd, то входящее событие получает имя set_Abcd, а исходящее событие Abcd_changed. При помощи входящего события другие узлы могут устанавливать значение поля Abcd. После любого изменения данного поля узел генерирует исходящее событие Abcd_chnaged.

Таким образом, объявление регулируемого поля Abcd выглядит следующим образом:

```
exposedField Abcd
```

Это будет эквивалентно объявлению одного поля и двух связанных с ним событий:

```
field Abcd  
eventIn set_Abcd  
eventOut Abcd_changed
```

Такие регулируемые поля необходимы для создания анимации.

5.5. Координатная система. Примитивы VRML, цвет и текстура

Принятая в VRML система координат является правосторонней трехмерной Картезианской координатной системой. Это означает, что ось *X* горизонтальна, ось *Y* вертикальна, а ось *Z* направлена от экрана.

В VRML приняты следующие единицы измерения:

- расстояние и размер: метры;
- углы: радианы;
- остальные значения: выражаются в диапазоне от 0 до 1.

В качестве примера можно привести цветовое пространство. Цвета в VRML задаются в RGB-кодировке, но компоненты цвета измеряются вещественной величиной от 0 до 1 (где 1 — максимальное значение составляющей цвета). Таким образом, красный цвет задается как (1, 0, 0), а серый, как (0.5, 0.5, 0.5)=(.5, .5, .5).

В VRML определены четыре базовые фигуры: куб (точнее прямоугольный параллелепипед), сфера, цилиндр и конус. Эти фигуры называются *примитивами* (primitives). Набор примитивов невелик, однако комбинируя их, можно строить достаточно сложные трехмерные изображения.

Цвет фигуры, определяется с помощью объекта *Material*.

```
Material {
    ambientColor 0.2 0.2 0.2
    diffuseColor 0.8 0.8 0.8
    specularColor 0 0 0
    emissiveColor 0 0 0
    transparency 0
}
```

Параметры *ambientColor*, *diffuseColor*, *specularColor* и *emissiveColor* управляют цветами и указываются в палитре RGB. Параметр *transparency* определяет степень прозрачности, причем максимальная прозрачность достигается при значении, равном единице.

Для имитирования различных поверхностей в VRML существует объект *<Texture2>*. В качестве текстуры легче всего использовать обычный графический файл, например в GIF-формате. В таком случае нужно только указать путь к файлу в параметре *filename* объекта *<Texture2>*.

```
#VRML V1.0 ascii
Texture2 {
    filename "marble.gif"
```

```

image    0 0 0 .
wrapS REPEAT
wrapT REPEAT
}
Cube {
    width 1
    height 1
    depth 1
}

```

Параметры wrapS и wrapT могут принимать значения REPEAT или CLAMP и управляют проектированием текстуры по горизонтальной и вертикальной осям.

5.6. Положение объектов в пространстве.

Гиперссылки. Узел Anchor

По умолчанию любой описанный объект будет располагаться точно по центру окна браузера. По этой причине, если описать, к примеру, два одинаковых цилиндра, они сольются друг с другом. Для того чтобы изменить положение второго цилиндра, используется узел Translation, определяющий координаты объекта:

```

Translation {
    translation 1 2 3 #т. е. соответственно
    x=1 y=2 z=3
}

```

Вообще говоря, указываемые в <Translation> координаты не являются абсолютными. Фактически это координаты относительно предыдущего узла Translation.

В VRML принято следующее правило: узлы, модифицирующие свойства фигур (Translation, Material и т. п.), действуют на все фигуры, описанные далее. Чтобы ограничить область действия модифицирующих узлов, фигуры необходимо сгруппировать с помощью узла Separator:

```

Separator {
    другие узлы
}

```

Узел Separator работает как контейнер, он может содержать любые другие узлы, и основным его предназначением является именно ограничение области действия узлов типа Translation и Material. Рекоменду-

ется всегда и везде использовать узел **Separator**. Он не только избавит от ошибок, связанных с относительностью координат, но и сделает VRML-код более простым и понятным.

Для вращения фигур вокруг осей координат применяется узел **Rotation**:

```
Rotation {
    rotation 0 1 0 1.57
}
```

Первые три цифры определяют, будет ли осуществлен поворот вокруг соответственно осей *x*, *y* и *z*, а четвертая — задает угол вращения в радианах. В приведенном выше фрагменте поворот осуществляется вокруг оси *y* на 90°.

Узел **Scale** масштабирует фигуры по одному или нескольким измерениям. Три цифры, стоящие после параметра **scaleFactor**, определяют коэффициенты масштабирования относительно осей *x*, *y* и *z*:

```
Scale {
    scaleFactor 1 1 1
}
```

В языке VRML возможность перемещения между виртуальными мирами или переход по гиперссылке осуществляется с помощью узла **Anchor**. Как правило, для пользователя это выглядит как объект в виртуальном мире, щелчок по которому мышкой переносит ассоциированного с пользователем аватара в другое место. Этот узел реализуется следующим образом:

```
Anchor {
    children [ ]
    description      " "
    parameter [ ]
    url [ ]
    bboxCenter 0 0 0 #(-inf, inf)
    bboxSize -1 -1 -1 #(0,inf)or-1 -1 -1
}
```

Узел **Anchor** — это группирующий узел. В его поле **children** может находиться целый список других узлов (к примеру, геометрических). Щелчок мыши по изображению любого из узлов-потомков приведет к переходу по гиперссылке указанной в поле **url**. В поле **description** помещаются словесные пояснения для данной ссылки. В поле **parameter** можно ввести дополнительную информацию (например, название фрейма). **BboxCenter** и **bboxSize** описывают параметры параллелепипе-

да, содержащего всех потомков этого узла (если в качестве параметров `bboxSize` указывается `-1`, то размеры сторон параллелепипеда вычисляет браузер). Эти данные позволяют ускорить рендеринг сцены.

5.7. Сенсоры окружающей среды. Узлы-манипуляторы

Сенсоры окружающей среды не принимают непосредственно ввод данных от пользователя, но вместо этого контролируют течение процессов в окружающей среде, таких как течение времени и расположение пользователя. К этому типу относятся:

- сенсор времени (`TimeSensor`);
- сенсор видимости (`VisibilitySensor`);
- сенсор приближения (`ProximitySensor`);
- сенсор коллизий (`Collision`).

`TimeSensor` (*сенсор времени*) — это, по своей сути, таймер, он не имеет конкретной позиции в виртуальном мире и не имеет ассоциированных с ним геометрических объектов. Этот узел является одним из важнейших для анимации узлов. Он может использоваться для того, чтобы порождать регулярные события, разовые события или управлять узлами-интерполяторами.

`VisibilitySensor` (*сенсор видимости*) — уже более интерактивный сенсор. Он представляет собой некоторый параллелепипед, генерирующий события, как только он попадает в поле зрения пользователя или выходит из него. По этим событиям можно точно определить время и характер произошедшего.

`ProximitySensor` (*датчик перемещений*) похож на `VisibilitySensor`, но события `isActive` возникают в момент входа или выхода из заданного параллелепипеда. Также этот сенсор отслеживает перемещения внутри заданного параллелепипеда и генерирует сообщения при изменении позиции или ориентации пользователя.

Узел `Collision` (*сенсор коллизий*) — это группирующий узел, с помощью которого можно включать и выключать распознавание коллизий (столкновений) между потомками этого узла и пользователем.

К узлам-манипуляторам относятся датчик касания (`TouchSensor`), сферический сенсор (`SphereSensor`), цилиндрический сенсор (`CylinderSensor`) и плоский сенсор (`PlaneSensor`).

Датчик касания (`TouchSensor`) реагирует на взаимодействие трехмерного эквивалента курсора мыши с геометрическим объектами —

потомками своего родительского узла. Этот узел генерирует соответствующие события при попадании курсора внутрь области, описанной этими геометрическими объектами. Если при этом удерживалась нажатой кнопка мыши, то генерируется еще одно сообщение (*isActive* со значением *TRUE*), если же был произведен одинарный щелчок, то генерируется исходящее событие *touchTime*.

Следующие три сенсора относятся к группе, именуемой датчиками бусковировки:

- *сферический сенсор* (*SphereSensor*) позволяет вращать объект вокруг центральной точки самого сенсора. Этот сенсор, как и все, улавливает щелчок мыши на геометрических объектах-потомках его родительского узла;
- *цилиндрический сенсор* (*CylinderSensor*) похож на сферический, но вращение может происходить только вокруг одной оси;
- *плоский сенсор* (*PlaneSensor*) позволяет осуществлять перемещение вдоль осей *X* и *Y* в своей локальной системе координат.

5.8. Мультимедийное резюме

Как уже говорилось в начале главы, виртуальные технологии традиционно используются в ряде конкретных областей для создания виртуальных сред моделирования объектов, обстановки и происходящих событий.

Более широкое распространение таких технологий сдерживают технические трудности. Так, файл с достаточно сложным и детализированным VRML-миротвором занимает весьма значительный объем, что при условии текстовой интерпретации VRML-файла приложением-визуализатором — неприемлемо по скорости. Этот аспект, а также несогласованность и частичная несовместимость программ для просмотра виртуальных миров сильно сдерживает развитие данного направления Internet-технологий [28].

Тем не менее, гибкость описания виртуальной среды, изначально заложенная в формат языка VRML, позволяет быстро и сравнительно легко получать простые геометрические модели или предметы обстановки виртуальных помещений; сохранять их в текстовом мультиплатформенном файле, воспроизводить с помощью обычного Internet-браузера при установленном бесплатном приложении.

ГЛАВА 6

ИСПОЛЬЗОВАНИЕ ФРАКТАЛОВ В МУЛЬТИМЕДИЙНЫХ ТЕХНОЛОГИЯХ

История возникновения. Виды фракталов по Калге. Дробная размерность. Снежинка Коха. L-системы. Множества Мандельброта и Жюлиа. Хаос и синергетика.

Слово фрактал образовано от латинского *«fractus»* и в переводе означает «состоящий из фрагментов». Оно было предложено Бенуа Мандельбротом в 1975 г. для обозначения нерегулярных, но самоподобных структур, которыми он занимался. Рождение фрактальной геометрии принято связывать с выходом в 1977 г. книги Мандельброта «Фрактальная геометрия природы». Интересно то, что в его работах использованы изыскания многих ученых, работавших в период 1875—1925 гг., таких, как Пуанкаре, Фату, Жюлиа, Кантор. Понятие фрактальной (дробной) размерности появилось в 1919 г. в работе Феликса Хаусдорфа. Но только в наше время удалось объединить их работы в единую систему. За этот короткий промежуток времени вышли десятки книг и сотни статей, в которых обсуждаются как чисто математические проблемы (например связность множеств), так и вопросы взаимоотношения фракталов и искусства, фракталов и хаоса, синергетики.

Применительно к контексту излагаемого материала, фракталы обладают *самостоятельной художественной ценностью*, и, как следствие, широко используются в различных мультимедийных технологиях [1]. Один из примеров — геометрические фракталы применяются в так называемых генераторах ландшафтов («террагенах») для задания карты высоты местности.

Основным свойством фракталов является *самоподобие*, т. е. их вид не претерпевает существенных изменений при рассмотрении с любым увеличением [29]. В самом простом случае небольшая часть фрактала при увеличении дает образ всего фрактала. Вообще же рассматриваются ситуации с нелинейным преобразованием подобия, подобием разных по размеру частей и т. п.

Для того чтобы представить себе все многообразие фракталов, удобно прибегнуть к их классификации по способу построения, предложенному Калге в 1988 г. В ней фракталы делятся на три группы: геометрические, арифметические и стохастические [30].

6.1. Геометрические фракталы. Самоподобие

Геометрические фракталы получают с помощью некоторой ломаной линии (или в трехмерном случае — поверхности), называемой **аттрактором** или **генератором**. Пусть, например, исходная фигура — это отрезок. За один шаг алгоритма все отрезки текущей фигуры заменяются ломаной линией генератора. Итоговое фрактальное изображение получается в результате бесконечного повторения этой процедуры (т. е. перехода к пределу). В этом случае части полученной фигуры будут подобны всей или почти всей фигуре (см. ниже снежинку Коха, рис. 29) [1].

Примерами геометрических фракталов служат:

- кривая дракона;
- снежинка Коха;
- кривая Леви;
- кривая Минковского;
- кривая Пеано.

К геометрическим фракталам также относят фракталы, получаемые с помощью так называемых *Итеративных систем функций* (Iterated Function System), например:

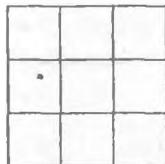
- папоротник Барнсли;
- квадрат Кантора;
- ковер Серпинского;
- губка Менгера;
- дерево Пифагора.

Для определенного класса таких систем последовательность итераций сходится к неподвижному изображению. При этом начальное изображение несущественно, а конечное зависит лишь от коэффициентов преобразований в системе. С помощью этого алгоритма получают достаточно реалистичные изображения папоротника, листьев, камней, облаков и т. п. Итеративные системы функций также используются во фрактальной компрессии изображений [31].

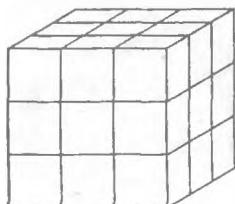
Рассмотрим *парадокс дробной размерности*, основанной на самоподобии фракталов. Традиционное восприятие нашего Декартова мира не позволяет допустить, что существуют системы, в которых размерности мира могут быть дробными. Однако во фрактальной теории — это

так. Разделим отрезок прямой на N равных частей. Тогда каждую часть можно считать копией всего отрезка, уменьшенной в $1/r$ раз. Очевидно, N и r связаны соотношением $Nr=1$ (рис. 28, верхний ряд):

$$N=3 \quad r=1/3 \quad d=1$$



$$N=9 \quad r=1/3 \quad d=2$$



$$N=27 \quad r=1/3 \quad d=3$$

Рис. 28. Связь размерности и коэффициента подобия

Если квадрат разбить на N равных квадратов (с площадью в $1/r^2$ раз меньше площади исходного), то соотношение запишется как $Nr^2=1$. Если куб разбить на N равных кубов (с объемом в $1/r^3$ раз меньше объема исходного), то соотношение примет следующий вид: $Nr^3=1$. Заметим, что размерность d объекта, будь то одномерный отрезок, двумерный квадрат или трехмерный куб, появляется как степень r в соотношении между N , числом равных подобъектов, и коэффициентом подобия. А именно:

$$Nr^d=1. \quad (6.1.1)$$

Множества, построенные на рис. 28, обладают целой размерностью. Зададимся вопросом, возможно ли такое построение, при котором показатель d в равенстве (6.1.1) не является целым, т. е. таким, что при разбиении исходного множества на N непересекающихся подмножеств, полученных масштабированием оригинала с коэффициентом r , значение d не будет выражаться целым числом. Ответ — да. Такое множество называют *самоподобным фракталом*. Величину d называют *фрактальной* (дроб-

ной) размерностью Хаусдорфа или размерностью подобия. Явное выражение для d через N и r находится логарифмированием обеих частей (6.1.1):

$$d = \frac{\log N}{\log (1/r)}. \quad (6.1.2)$$

Логарифм можно взять по любому положительному основанию, отличному от единицы, например, по основанию 10 или по основанию $e \approx 2,7183$.

6.2. Дробная размерность снежинки Коха

Граница снежинки, придуманной Гельгом фон Кохом в 1904 г. (рис. 29), описывается кривой, составленной из трех одинаковых фракталов дробной размерности $d \approx 1,2618$.

Рассмотрим процесс определения размерности подробнее. Каждая третья снежинки строится итеративно, начиная с одной из сторон равностороннего треугольника. Пусть K_0 — начальный отрезок. Уберем среднюю треть и добавим два новых отрезка такой же длины, как показано на рис. 30. Назовем полученное множество K_1 . Повторим данную процедуру многократно, на каждом шаге заменяя среднюю треть двумя новыми отрезками. Обозначим через K_n фигуру, получившуюся после n -го шага.

Очевидно, что вследствие бесконечного уменьшения (стремления к нулю), последовательность кривых $\{K_n\}_{n=1}^{\infty}$ сходится к некоторой предельной кривой K . Рассмотрим некоторые ее свойства.

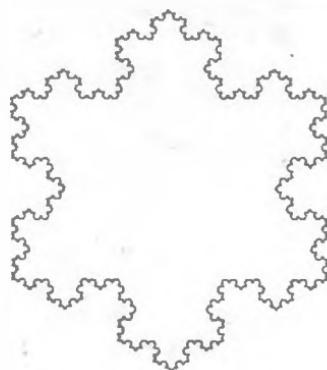


Рис. 29. Снежинка Коха



Рис. 30. Этапы построения снежинки Коха

Если взять копию K , уменьшенную в три раза ($r=1/3$), то все множество K можно составить из $N=4$ таких копий. Следовательно, отно-

шение самоподобия выполняется при указанных N и r , а размерность фрактала

$$d = \log(4) / \log(3) \approx 1,2618.$$

Еще одно важное свойство, которым обладает граница снежинки Коха — ее бесконечная длина (см. теорему 6.1). Это может показаться удивительным, обычно гладкие кривые всегда имеют конечную длину (в чем можно убедиться интегрированием). Мандельброт в этой связи опубликовал ряд увлекательных работ, в которых исследуется вопрос измерения длины береговой линии Великобритании. В качестве модели он использовал фрактальную кривую, напоминающую границу снежинки за тем исключением, что в нее введен элемент случайности, учитывающий случайность в природе. В результате оказалось, что кривая, описывающая береговую линию, имеет бесконечную длину.

Теорема 6.1. Граница снежинки Коха имеет бесконечную длину.

Доказательство. Достаточно показать, что каждый из трех идентичных фракталов K , полученных итерациями (рис. 30), имеет бесконечную длину. Пусть исходный отрезок K_0 имеет единичную длину. Тогда длина кривой K_1 равна $4/3$. Длина кривой K_2 равна $16/9$. Продолжая таким же образом, имеем, что кривая K_n после n -го шага имеет длину $4^n/3^n$. Следовательно, длина предельной кривой K равна бесконечности:

$$\lim_{n \rightarrow \infty} (4^n/3^n) = \infty$$

Теорема доказана.

6.3. L-системы. Тетр-графика

Понятие L-систем, тесно связанное с самоподобными фракталами, появилось только в 1968 г. благодаря Аристиду Линден-Майеру. Изначально L-системы были введены при изучении формальных языков, а также использовались в биологических моделях селекции. С их помощью можно строить многие известные самоподобные фракталы, включая снежинку Коха и ковер Серпинского (рис. 31).

Некоторые другие классические построения, например кривые Пеано (ра-

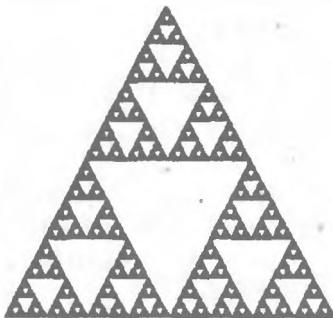


Рис. 31. Ковер Серпинского

боты Пеано, Гильберта, Серпинского), также укладываются в эту схему. И конечно, *L*-системы открывают путь к бесконечному разнообразию новых фракталов, что и послужило причиной их широкого применения в компьютерной графике для построения фрактальных деревьев и растений. Мы ограничимся случаем детерминированных *L*-систем и графикой на плоскости.

Для графической реализации *L*-систем в качестве подсистемы вывода используется так называемая тетл-графика (*turtle* — черепаха). При этом точка «черепашка» передвигается по экрану дискретными шагами, как правило, прорисовывая свой след, но при необходимости может перемещаться без рисования, с поднятым пером. Для управления черепашкой доступны три параметра (x, y, α) , где (x, y) — координаты черепашки; α — направление, в котором она смотрит. Черепашка обучена интерпретировать и выполнять последовательность команд, задаваемых кодовым словом, буквы которого читаются слева направо. Кодовое слово представляет собой результат работы *L*-системы и может включать следующие буквы:

- F — переместиться вперед на один шаг, прорисовывая след;
- B — переместиться вперед на один шаг, *не* прорисовывая след;
- $+$ — увеличить угол α на величину θ ;
- $-$ — уменьшить угол α на величину θ .

Размер шага и величина приращения по углу θ задаются заранее и остаются неизменными для всех перемещений черепашки. Если начальное направление движения α (угол, отсчитываемый от положительного направления оси *X*) не указано, то полагаем α равным нулю.

ОПРЕДЕЛЕНИЕ 14. Формально, детерминированная *L*-система состоит из алфавита, слова инициализации, называемого *аксиомой* или *инициатором*, и набора порождающих правил, указывающих, как следует преобразовывать слово при переходе от уровня к уровню (от итерации к итерации).

К примеру, можно заменять букву *F* при помощи порождающего правила *newf*=*F*—*F*+*F*—*F*, что соответствует *L*-системе для снежинки Коха, рассматриваемой ниже. Символы «+», «-», не обновляются, а просто остаются на тех местах, где они встретились. Обновление букв в данном слове предполагается одновременным, т. е. все буквы слова одного уровня обновляются раньше любой буквы следующего уровня.

L-система, соответствующая снежинке Коха (рис. 29), задается следующим образом:

$$\theta = \pi/3.$$

Аксиома: $F++F++F;$

Порождающее правило: $newf = F - F++F - F.$

Графическое представление аксиомы $F++F++F$ — равносторонний треугольник. Черепашка делает один шаг вперед; затем угол α увеличивается на $2\pi/3$; и черепашка делает еще один шаг вперед; угол α снова увеличивается на $2\pi/3$; и черепашка делает еще шаг.

На первом шаге каждая буква F в слове-инициаторе $F++F++F$ заменяется на $F - F++F - F$:

$(F - F++F - F) ++ (F - F++F - F) ++ (F - F++F - F).$ Убирая скобки, получаем

$F - F++F - F + F - F++F - F++F - F++F - F - F++F - F.$

Повторяя этот процесс, на втором шаге получим

$F - F++F - F - F - F++F - F++F - F++F - F - F++F - F++F - F++F - F - F++F - F - F - F++F - F++F - F - F++F - F - F++F - F + \dots$

После некоторого, достаточно большого количества таких рекурсивных итераций получается весьма реалистичное изображение снежинки Кюха.

6.4. Алгебраические фракталы

Это самая обширная группа фракталов. Получают их в процессе итераций функций или систем функций. Наиболее известны изображения множеств Мандельброта и Жюлиа, получаемые при итерациях комплексных многочленов второй степени.

Строят алгебраические фракталы следующим образом. Задается фрактальная функция, на комплексной плоскости выбирается прямогольная область и на ней фиксируется сетка — пиксельный растр изображения. От выбранной исходной точки зависит начальное значение функции. Каждое применение функции к комплексному числу-точке переводит ее в другое число. При большом количестве итераций можно отследить характер поведения получившейся последовательности — сходится она, расходится, остается ограниченной или ведет себя хаотически.

Характер поведения влияет на цвет выбранной исходной точки — например, все расходящиеся точки можно покрасить в белый, сходящиеся — в черный. Одним из самых распространенных способов цветного раскрашивания точек-узлов сетки будет сравнение текущего значения с заранее выбранным числом, которое считается «бесконечным»,

т. е. цвет исходной точки равен номеру итерации, на которой функция достигла «бесконечности», или черному в противном случае

Меняя функцию, положение изображаемой области и алгоритм выбора цвета, можно получать сложные фрактальные картины с причудливыми многоцветными узорами. Несмотря на примитивность алгоритма и используемых функций, получаемые изображения весьма нетривиальны [1].

Кроме того, можно использовать не только классические функции — многочлены второй степени. Фракталы получают и при итеративном поиске корней уравнений (метод Ньютона), и итеративном решении систем дифференциальных уравнений (метод Эйлера), рассмотрении модели магнетизма. Получаемые при этом иллюстрации являются двумерными графиками зависимости числового параметра от начальных данных и имеют самостоятельную научную ценность.

Заметим также, что, например, системы дифференциальных уравнений элементарным способом позволяют строить фрактальную функцию не только для двумерного, но и для n -мерного пространства. Следовательно, используя алгоритм построения изображений фигур, задаваемых с помощью неявных функций, мы получим трехмерные фрактальные структуры.

Рассмотрим построение алгебраических фракталов на примере тесно связанных между собой множеств Жюлия и Мандельброта.

6.5. Множество Мандельброта и множества Жюлия

Идея, использованная Мандельбротом, состояла в том, чтобы вместо действительных чисел рассмотреть комплексные и наблюдать процесс $z_0 \mapsto z_1 \mapsto z_2 \mapsto \dots$ не на прямой, а в плоскости. Правило $z_n \mapsto f(z_n)$ указывает, куда должна переместиться точка в комплексной плоскости. Конкретный вид правила не является существенным, поскольку различные правила могут порождать то же самое множество Мандельброта.

Процесс Мандельброта основан на простой формуле

$$z_{n+1} = f(z_n) = z_n^2 + c.$$

Выбрав произвольное число z_0 , возведем его в квадрат и прибавим константу c для того чтобы получить z_1 ; затем повторим вычисления для того чтобы получить z_2, z_3 и т. д. Возьмем простейшее из возможных значений константы c , а именно $c=0$. Тогда при каждой итерации вычисляется точный квадрат числа: $z_0 \mapsto z_0^2 \mapsto z_0^4 \mapsto z_0^8 \mapsto \dots$. Для этой по-

следовательности в зависимости от выбора z_0 существуют три возможности [29].

1. Числа получаются все меньшими и меньшими, их последовательность приближается к нулю. В этом случае считается, что ноль является *аттрактором* для процесса $z \mapsto z^2$. Все точки, находящиеся на расстоянии меньше единицы от этого аттрактора, движутся к нему.

2. Числа становятся все большими и большими, стремясь к бесконечности. Тогда считается, что бесконечность также является *аттрактором* для этого процесса. Все точки, лежащие на расстоянии больше единицы от нуля, движутся к бесконечности.

3. Точки находятся и продолжают оставаться на расстоянии единицы от нуля. Их последовательности лежат на границе двух *областей притяжения*, в данном случае на окружности единичного радиуса с центром в нуле.

Иными словами, плоскость делится на две зоны влияния или *бассейна притяжения* неподвижных точек 0 и ∞ , а границей между ними является просто окружность, точки которой никуда не притягиваются.

Более интересная ситуация возникнет, если мы выберем ненулевое значение c , например $c = -0,15 + 0,55i$. Здесь для последовательности $z_0 \mapsto z_1 \mapsto z_2 \mapsto \dots$ также имеются три из перечисленных выше возможностей, но внутренний аттрактор (отмеченный точкой на рис. 32) уже не является нулем, а граница уже не является гладкой.

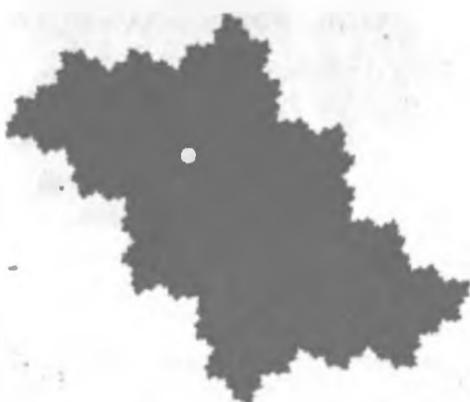


Рис. 32. Бассейн притягивающей неподвижной точки

Из рисунка видно, что эта граница сильно изломана. Причем при увеличении она выглядит столь же изломанной, как и без увеличения.

Именно это Б. Мандельброт назвал фрактальной структурой такой границы. Она напоминает линию морского берега, многие другие естественные границы, которые становятся явно тем длиннее, чем более мелкий масштаб используется для их измерения. Одной из характерных особенностей этой границы является ее **самоподобие**. Если взглянуть на любой из ее поворотов или заливов, то можно обнаружить, что одна и та же форма встречается в различных местах и имеет разные размеры.

Границы такого рода в математике называют **множествами Жюлия**. Во время первой мировой войны французские математики Гастон Жюлия и Пьер Фату изучали их свойства для общего случая рациональных отображений в комплексной плоскости.

Существует правило, указывающее, какой вид имеет множество Жюлия при каждом конкретном выборе c . Это правило и приводит к множеству Мандельброта M . Оно представлено на рис. 33 как закрашенная черным цветом часть комплексной плоскости графика переменной c . Каждое комплексное число c либо принадлежит черной структуре M , либо нет.

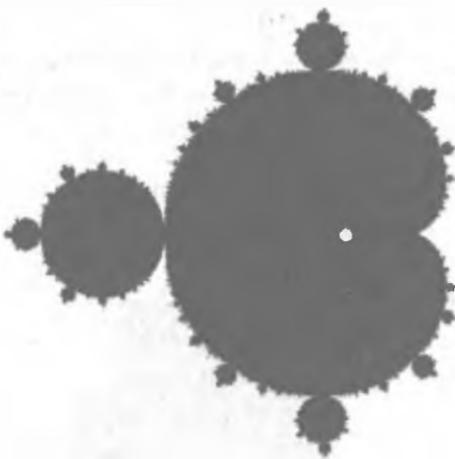


Рис. 33. Множество Мандельброта

Соответствующие множества Жюлия процесса $z \mapsto z^2 + c$ существенно различаются. Они представляют собой связные структуры, когда c лежит внутри M , и рассыпаются на бесконечное число фрагментов, когда c лежит снаружи. Поэтому граница множества M вызывает исключительный интерес. Представим себе некоторый путь в c -плоскости, начинающийся внутри M и заканчивающийся вне него.

Если менять c , двигаясь вдоль этого пути, то самые качественные изменения происходят с соответствующими множествами Жюлиа тогда, когда c пересекает границу M : они, как будто «взорвавшись», превращаются в облако из бесконечного числа точек. В этом смысле граница множества M определяет момент математического фазового перехода для множеств Жюлиа отображения $z \mapsto z^2 + c$.

Упрощенно говоря, если зафиксировать c и изменять z_0 в поле комплексных чисел, то получим множество Жюлиа, а если зафиксировать $z_0 = 0$ и изменять параметр c , то получим множество Мандельброта. Если взять z_0 далеко от нуля, то последовательность будет быстро стремиться к бесконечности. Это, конечно, верно также и тогда, когда точка z_n для некоторого n находится далеко от нуля. Но существует и такие значения z_0 , для которых последовательность z_n никогда не уходит далеко, а всегда остается ограниченной.

Итак, запишем точные определения [32].

ОПРЕДЕЛЕНИЕ 15. Множество Мандельброта M для полинома $f_c(z) = z^2 + c$ определяется как множество всех $c \in C$, для которых орбита точки 0 ограничена:

$$M = \left\{ c \in C : \left\{ f_c^{(n)}(0) \right\}_{n=0}^{\infty} \text{ ограничена} \right\}.$$

ОПРЕДЕЛЕНИЕ 16. Множество Жюлиа функции f , обозначаемое как $J(f)$, определяется как

$$J(f) = \partial \left\{ z : f^{(n)} \rightarrow \infty \text{ при } n \rightarrow \infty \right\}.$$

Таким образом, множество Жюлиа функции f есть граница множества точек $f(z)$, стремящихся к бесконечности при итерировании $f(z)$.

6.6. Стохастические фракталы

Последним классом по Калге являются стохастические фракталы. В процессе создания всех природных объектов всегда есть случайность. Самым известным их применением является использование датчика случайных чисел в геометрических фракталях. При этом деревья становятся быть симметричными, береговая линия — правильной, за счет чего конечные синтезированные объекты становятся более похожими на природные.

Двумерные геометрическо-стохастические фракталы используют при моделировании рельефа местности, поверхности моря. Заметим, что при генерации гор потребуется множество других алгоритмов пост-обработки. Они «пробывают» скалы для горных рек, определят местонахождение озер, ледников, леса, выберут границу снегов, цвет растительности и горных пород.

В компьютерной графике известны применения геометрических фракталов при получении изображений деревьев, кустов, береговой линии. Двухмерные геометрические фракталы используются для создания объемных текстур. Так, природные ландшафты, особенно горные и пересеченные местности, легко приблизить с помощью самоподобия на построенной сетке географических высот. Пусть имеется разбиение ландшафта на симметричные области (квадраты). На первом этапе задается случайное значение для верхней левой ячейки каждого квадрата. На следующем этапе сетка разбивается на четыре подсети, в каждую подсеть копируются значения квадрата, значения в ячейках задаются как среднее арифметическое между соседними по горизонтали и вертикали проинициализированными ячейками плюс случайная величина (которая тем меньше, чем мельче разбиение на подсети) и т. д.

6.7. Хаос и синергетика

Фрактальная теория тесно связана с хаотической динамикой. В процессе изучения физических процессов, уравнения их описывающие начинают вести себя непредсказуемо. Оказалось, что характер стохастического возмущения таких уравнений удобно описывать в терминах фрактальной теории.

Под *хаосом* понимают очевидно непредсказуемое поведение, возникающее в детерминированных системах из-за сильной чувствительности к начальным условиям. Хаос возникает в динамических системах, если две сколь угодно близкие начальные точки расходятся экспоненциально так, что их будущее поведение невозможно предсказать при известных обстоятельствах.

Существенная зависимость от начальных условий — основная черта, присущая хаотической динамике.

Рассмотрим метрическое пространство (X, d) . Будем называть отображение $f : X \mapsto X$ хаотическим, если выполняются следующие условия:

- 1) f обладает существенной зависимостью от начальных условий;
- 2) f транзитивно;
- 3) периодические точки f плотны в X .

Может создаться впечатление, что хаотическое поведение не поддается никаким методам математического анализа, применимым в обычных случаях. Но оказывается, что и здесь есть свои закономерности: отношение длин следующих друг за другом интервалов удвоения возмущений является постоянной величиной [29]. Она равняется 4.669162... и называется константой Фейхенбаума. Это число является универсальной характеристикой сценария удвоения периода для целого класса одномерных процессов. За подробным анализом сценариев хаотического поведения функций можно обратиться к [32].

В научных кругах растет уверенность в том, что процессы образования структур и самоорганизация развиваются в соответствии с небольшим числом сценариев, не зависящих от конкретной системы. В Германии, например, Герман Хакен из Штутгартского университета с конца 60-х гг. направлял свою деятельность на создание «синергетики» [33]. Являясь одним из основателей теории лазеров, он обнаружил, что образование внутренних структур в лазере происходит в соответствии с законами, очень напоминающими конкуренцию молекулярных видов, которую описал Манфред Эйген (Институт Макса Планка в Геттингене) в своих исследованиях ранней эволюции жизни.

Синергетика целенаправленно пытается отыскивать правила, по которым возникает порядок в сложных системах. Большую работу по анализу самоорганизующихся динамических систем в неравновесной термодинамике провел лауреат Нобелевской премии, сын русского эмигранта бельгийский подданный Илья Пригожин [34].

Под синергетическим принципом понимают природную способность любой системы (микросистемы) к самоорганизации, т. е. к изменению своих качественных характеристик за счет внутреннего потенциала энергетического обмена между ее элементами.

В последнее время появились интересные работы по синергетике в социологии. В них, в частности, постулируется, что социум — это хаотическая система, так как каждый член этой системы ведет себя непредсказуемо. Однако социум от этого не разваливается, и тенденция поведения всей системы, тем не менее, поддается описанию на макроуровне.

Ождалось, что компьютеры наведут полный порядок и дисциплину во всех областях жизни, но именно они дали возможность лучше понять гармонию и хаос. Более того, и по объективному измерению, и по субъективному восприятию стало очевидно, что в технических объектах по сравнению с природными теряется роскошь некоторой нерегулярности, беспорядка и непредсказуемости.

6.8. Мультимедийное резюме

Визуализированные фракталы обладают самостоятельной художественной ценностью, поэтому их часто используют для реалистичного моделирования сложных природных объектов.

Существуют основные, объединяющие все типы фракталов, свойства:

- они содержат произвольно малые масштабы, поэтому могут быть увеличены до бесконечности;
- они слишком нерегулярны, чтобы быть описанными на традиционном геометрическом языке;
- они имеют некоторую форму самоподобия, чаще всего нестрогую;
- они имеют периметр бесконечной длины и дробную «фрактальную» размерность, называемую также размерностью Хаусдорфа.

Большинство геометрических фракталов удобно строить с помощью рекурсивной «черепашьей» графики и формализма, называемого *L*-системами.

Хаотическое поведение многих динамических систем, появляющееся при небольших изменениях начальных условий может быть описано в терминах фрактальной теории, что позволяет изучать не только физические и математические модели, но и социальные или биологические.

ГЛАВА 7

ТЕХНОЛОГИИ ХРАНЕНИЯ И ПЕРЕДАЧИ МУЛЬТИМЕДИЙНЫХ ДАННЫХ

Специфика мультимедийных носителей. Форматы оптических дисков. Устройство DVD. Blu-ray. Флэш-память. FireWire и USB. Bluetooth и WiFi.

Помимо рассмотрения базовых подходов и технологий, используемых для создания мультимедийной продукции, необходимо изучить способы представления, хранения и передачи мультимедийных данных. Основной интерес, в отличие от универсальных компьютерных накопителей, представляют технологии хранения, разработанные преимущественно в целях представления мультимедийных данных и учитывающие их специфику.

Среди носителей мультимедийных данных (впрочем, как и среди носителей информации в целом) можно выделить три группы:

- 1) магнитные носители (наиболее традиционная и распространенная группа);
- 2) оптические и гибридные магнитооптические диски;
- 3) накопители, использующие флэш-память.

Характеристики и особенности традиционных магнитных накопителей хорошо известны и не требуют широкого освещения в данной книге. Стоит лишь упомянуть о специализированных жестких дисках, используемых при обработке аудио- и видеоинформации. Основное их отличие от любых других жестких дисков заключается не только в повышенной пропускной способности, но и в *равномерности скорости записи и чтения с носителя*. Это требование действительно специфично для мультимедиа, поскольку пользователю более критично воспроизведение аудио или видео без рывков и задержек, чем небольшие различия в качестве проигрываемого материала.

Как правило, такие диски используются в профессиональных комплексах и не имеют массового распространения. Также для обеспечения высокой пропускной способности достаточно часто используют RAID-массивы. Эта технология позволяет записывать информацию

параллельными потоками сразу на несколько обычных жестких дисков, что обеспечивает необходимую скорость и надежность.

В отличие от накопителей на магнитных дисках, оптические диски разрабатывались в первую очередь как носители высококачественной аудио- и видеоинформации. И только по прошествии некоторого времени они нашли свое применение в компьютерных технологиях. Причем емкость и скорость работы оптических дисков то опережали показатели магнитных носителей, то снова отставали от них. Для более глубокого понимания принципов работы рассмотрим эволюцию оптических носителей.

7.1. Эволюция оптических дисков

Идея использовать лазер для записи и считывания информации возникла еще в 60-х гг. Но тогда отсутствовали необходимые технологии и, вероятно, спрос. А на практике все началось в 1972 г., когда компании Philips и MCA впервые продемонстрировали LaserDisk, а в 1978-м этот формат, известный также как LD, появился на рынке.

Диск диаметром примерно 30 см вмещал на каждой стороне до часа видео, качество которого удалось перекрыть только много лет спустя, с появлением DVD. Более поздние выпуски включали еще и многоканальное звуковое сопровождение и субтитры.

К недостаткам LD можно отнести большие габариты, высокую цену, отсутствие интерактивного меню. Вдобавок выяснилось, что диски подвержены процессу «лазерного гниения» — окисления отражающего слоя. Однако долгое время LD были единственным выбором для желающих иметь доступ к видео высокого качества [35].

Технология CD-ROM была разработана совместными усилиями компаний Philips и Sony в 1982 г. Компакт-диски являлись тогда первым и самым передовым средством сверхточной передачи аудиоинформации. Благодаря таким качествам, как широкий динамический диапазон, пониженный уровень шума, неуязвимость и долговечность, компакт-диски уже через 5 лет вытеснили виниловые грампластинки с передовых позиций и заняли их место в качестве основных носителей стереофонической информации.

Возможность записи музыкальных произведений на компакт-диски была одним из основных технических требований при разработке CD-систем. 70 минут звучания музыки, достаточные для того чтобы вместить Девятую симфонию ре-минор Бетховена, — это блок данных, общий объем которого превышает 600 Мб.

Но очень скоро звук стал лишь одной из многих разновидностей использования компакт-дисков. Первоначальное имя пришлось расширить, чтобы отличить музыкальные компакт-диски от всех остальных. Музыкальный CD-диск стал называться CD-DA, т. е. цифровой аудио компакт-диск.

Возможности произвольного доступа к данным, простая и дешевая процедура копирования делали CD-ROM идеальным средством распространения информации. Одна и та же базовая универсальная технология позволяет хранить данные, звук и видео на достаточно емком и в то же время дешевом носителе. Благодаря этому компакт-диск долгое время является ядром мультимедийных средств. Фактически CD-диски стали настолько распространенными, что изготовители ПК считали отсутствие устройства CD-ROM в составе своих компьютеров нестандартной комплектацией.

Преимущества оптического излучения для записи информации весьма многочисленны. Например, напряженность магнитного поля жестких дисков резко падает при удалении от поверхности носителя, в связи с чем головки чтения/записи должны находиться на расстоянии микронов от поверхности. А оборудование, генерирующее луч света, который осуществляет запись или считывание с оптического запоминающего устройства, может располагаться на гораздо большем удалении от носителя.

Основой системы CD-ROM является собственно диск. Размер диска составляет 120 мм в диаметре, диаметр отверстия в центре компакт-диска равен 15 мм.

В основе оптической системы кодирования данных лежит изменение отражающей способности носителя при помощи механических изменений текстуры поверхности. Фактически диск играет роль отражающего зеркала, которое покрыто темными углублениями (ямками или питами), с помощью которых и осуществляется кодирование данных. Для нанесения этих углублений на поверхность диска используется луч лазера. Затем эти углубления, которые являются механической особенностью поверхности эталонного мастер-диска, могут копироваться при помощи штамповочного оборудования. Запись производится только на одной стороне компакт-диска, затем диск подвергается металлизации (отражающая поверхность) и ламинированию (защита от химических и физических повреждений) [36].

Вместо того чтобы использовать ряд концентрических дорожек (как в случае магнитных компьютерных систем памяти), дорожка данных на CD-диске — это одна длинная непрерывная спираль. Проигрыватель

компакт-дисков просматривает дорожку, начиная от центра диска и двигаясь затем по направлению к его внешнему краю. Для считывания данных с диска в проигрывателе необходимо наличие фотодетектора — электрического устройства, реагирующего на различные уровни яркости путем изменения своего сопротивления.

Чтобы максимально увеличить доступный объем памяти диска, пишущая система компакт-дисков выполняет запись с постоянной линейной скоростью. При чтении с внутренних областей компакт диск вращается быстрее, чем при считывании с внешних, так что каждую секунду под головкой считывания возникают равные по длине отрезки дорожки.

7.2. Стандарты CD-ROM

Как и в случае с другими системами массовых запоминающих компьютерных устройств, стандартизация CD-ROM происходит на нескольких уровнях — на уровне аппаратных средств и на уровне программного обеспечения.

На аппаратном уровне системы компакт-диска управляются несколькими стандартами. По существующей традиции промышленные стандарты CD-ROM были названы цветами обложек книг документации, определяющими эти стандарты. Существуют следующие книги и стандарты.

Red Book (Красная книга) описывает цифровую аудиосистему CD-DA — первоначальное применение компакт-диска, где звуковая информация хранится в цифровой форме. Название «Красная книга» связано с международным стандартом (ISO 10149), который был опубликован в книге с красным переплетом. Этот стандарт определяет элементы цифрового кодирования и частоты выборки, включая скорость передачи данных и точный вид используемой импульсно-кодовой модуляции. Стандарт Красной книги используется для дисков двух различных емкостей: одни могут содержать 650 Мб данных или 74 минуты музыки, другие — 700 Мб данных или 80 минут музыки.

В соответствии с этими стандартами информационное пространство CD-ROM состоит из следующих частей.

1. *Область входа* (Lead In), содержащая оглавление (Table of Contents или ТОС), адреса всех треков, их длину на диске, суммарное время записей и информацию о диске; ТОС присутствует на всех CD. Если ее нет, то диск будет нечитаем на проигрывателе CD или дисководе CD-ROM.

2. Область данных (программ или звуковых треков).
3. Область вывода (Lead Out), в которой находится специальная метка, указывающая на конец записи.

При записи диска используется специальный метод представления цифровой информации, обеспечивающий ее коррекцию и резко снижающий вероятность ошибок. При этом для представления одного байта информации используется не обычные 8 бит, а целых 14. Преобразование такого представления в обычное осуществляется при помощи специальной таблицы — EFM (Eight of Fourteen Modulation, модуляция «восемь на четырнадцать»).

Верхняя граница частот несколько превышает 15 кГц, нижняя граница частот близка к 0. Система хранит звуковые данные с разрешающей способностью, равной 16 бигам, так что каждый аналоговый звуковой уровень количественно определен как один из 65 536 уровней. Это соответствует динамическому диапазону в 96 дБ (т. е. составляет $20\log_2^{16}$). Чтобы согласовать значение верхней границы частот с работой фильтров слаживания, система использует частоту выборки, равную 44,1 кГц (впрочем, точная цифра 44,1 кГц — наследие от аналоговых систем, такая частота дискретизации была установлена для телевизионного форматов NTSC и PAL).

Green Book (Зеленая книга) управляет дисками CD-I (интерактивными компакт-дисками), разработана компанией Philips в 1986 г. как стандарт аппаратного и программного обеспечения с целью объединения на одном диске текстовой, звуковой и видеинформации. В соответствии со стандартом Зеленої книги, диск CD-I использует адаптивную импульсно-кодовую дельта-модуляцию для сжатия большего количества звуковой информации — до 2 полных часов высококачественного стереозвучания или до 20 часов монозвучания. Возможности CD-I позволяют чередовать аудиодорожки, видеодорожки и дорожки данных на диске.

Orange Book (Оранжевая книга) — это служебное издание, где описывается стандарт CD-R (систем записываемых компакт-дисков).

Yellow Book (Желтая книга) описывает стандарты хранения гибридных данных для дисков CD-ROM. Это наиболее распространенный компьютерный стандарт мультимедийных средств.

Кроме того, некоторые изготовители пробовали создать собственные типы компакт-дисков и разработали стандарты, которые и по сей день остаются частными стандартами. Среди них: *видеоинтерактивная система* (Video Interactive System), разработанная корпорацией Microsoft & Tandy Corporation; *CD-телевидение* (CD-TV), частный стандарт видеопамяти, разработанный несуществующей в настоящее время компа-

нией Commodore International; *MMCD* — стандарт мультимедийных средств для карманных проигрывателей компакт-дисков, разработанный корпорацией Sony; *PhotoCD* — стандарт для сохранения высококачественных фотографических изображений, разработанный компанией Eastman Kodak Company.

Несмотря на то, что аппаратные стандарты определяют процесс форматирования компакт-диска и методы хранения данных, они не уточняют, каким образом операционные системы и прикладные программы используют дисковую память [37]. Они схожи с низкоуровневым форматом жесткого диска. В специализированных аппаратных прикладных программах, подобных системам аудио компакт-дисков, такой уровень стандартизации является вполне достаточным.

White Book (Белая книга) — формат VideoCD — появился в 1993 г. как результат работы VideoCD-консорциума, в который вошли Philips, Sony, Matsushita и JVC. Данные на дисках VideoCD кодируются с помощью формата MPEG-1, качество видео соответствует VHS.

Емкость дисков VideoCD часто превышает 650 Мб, отведенных под данные на CD-ROM. Дело в том, что данные на VideoCD записаны в специальном формате, который опускает исправление ошибок в пользу большей вместимости и непрерывности поступления данных в буфер видеовывода. Это естественно для потокового видео (см. выше специфику воспроизведения мультимедийных данных), но определенно не рекомендуется для обычных данных. На один диск может вмещаться до 74 минут видео в формате MPEG-1 со стереозвуковым сопровождением при постоянной скорости считывания 1,1 Мбит/с.

SVCD. Естественно, что по мере развития технологий качество VideoCD перестало удовлетворять потребителей. К 1997 г. с появлением стандарта MPEG-2 возникло сразу несколько форматов, использующих его для записи видео на CD. На самом большом для VideoCD рынке (китайском) определились несколько крупных участников, один из них, компания C-Cube Microsystems вместе с китайскими партнерами создала собственный формат CVD (China Video Disk). Другой формат, HQ-VCD, предложил VideoCD-консорциум.

А в 1998 г. специальным комитетом при правительстве Китая, который состоял из экспертов и представителей фирм-производителей, был разработан и принят формат SuperVCD, фактически включивший в себя и CVD, и HQ-VCD. В порядке отступления можно заметить, что привлечение новых технологий и защиту национальной экономики китайское руководство рассматривает как одну из важных задач, и в данном случае в принятии нового стандарта сыграло немалую роль жела-

ние Китая не зависит от DVD-консорциума и не платить лицензионные отчисления иностранным компаниям. И, конечно, собственный стандарт — предмет национальной гордости.

С разрешением до 480×576 , качество видео SVCD при просмотре на телевизоре среднего класса приближено к DVD. Также есть поддержка многоканального звука. К недостаткам SuperVCD дисков нужно отнести недостаточную вместительность — время воспроизведения составляет не больше 45 минут.

Таблица 1.
Сравнительные характеристики некоторых типов VideoCD

	SVCD	CVD	VCD 3.0	VCD 2.0	DVD
Формат видеокомпрессии	MPEG-2	MPEG-2	MPEG-1	MPEG-1	MPEG-2
Скорость считывания, Мбит/с	переменная, 1,7–2,6	переменная, 1,7–2,	постоянная, 1,1	постоянная, 1,1	переменная 3–9,8
Минимальная скорость привода	2x CD	2x CD	1x CD	1x CD	~9
Время воспроизведения, мин	45	45	74	74	≤ 4 ч
Уровень интерактивности ¹	Высокий	Средний	Высокий	Средний	Высокий
Разрешение, точек	NTSC: 480×480 PAL: 480×576	NTSC: 480×480 PAL: 480×576	NTSC: 352×240 PAL: 352×288	NTSC: 352×240 PAL: 352×288	NTSC: 720×480 PAL: 720×576
Формат изображения	4:3/16:9	4:3	4:3	4:3	4:3/16:9
API	Поддержка JAVA		Поддержка JAVA		Интерактивное меню
Распространенность	Средняя	Низкая	Низкая	Высокая	Высокая

Обратите внимание, что в таблицу также добавлен формат DVD. Рассмотрим его подробнее.

¹ Обозначает, насколько пользователь сможет влиять на процесс воспроизведения. Например, если он может только включать и выключать субтитры, это низкий уровень интерактивности, а когда появляются меню, ссылки и пр., то это позиционируется как высокий уровень.

7.3. Устройство DVD

Как уже говорилось, появление компакт-дисков послужило отправной точкой в развитии компьютерных мультимедийных технологий. Однако области применения и компьютеров, и оптических дисков быстро расширяются. Одновременно лавинообразно растут и объемы информации, которую нужно где-то хранить. Современные компьютерные игры с их все более интерактивным мультимедийным содержимым давно перестали умещаться на компакт-дисках; емкости диска не хватает и для записи полнометражного видеофильма.

Предвидя подобную ситуацию, производители электроники заблаговременно стали разрабатывать следующее поколение оптических носителей информации. Как и в случае с компакт-дисками, в роли двигателя прогресса вновь выступила индустрия развлечений. Перед разработчиками была поставлена цель: увеличить емкость дисков до такой степени, чтобы на одной их стороне можно было целиком поместить высококачественный цифровой фильм.

Поначалу ведущие фирмы не согласовывали свои усилия в этом направлении, что в последствии вылилось в войну форматов и значительно подорвало интерес потребителей к новой продукции (сходная ситуация была, например, с бытовыми видеомагнитофонами). Пытаясь избежать подобного развития событий, основные производители стали координировать свою деятельность с целью выработки унифицированного стандарта. В конце первого этапа интеграционного процесса начали вырисовываться контуры двух форматов: MMCD (MultiMedia Compact Disk), продвигаемого фирмами Sony и Philips; и SD (Super Density), за которым стояли сразу семь компаний (Toshiba, Time Warner и ряд других).

Для окончательного согласования единого стандарта в сентябре 1995 г. десять компаний (Hitachi, Matsushita Electronic Industrial, Mitsubishi Electric, Philips Electronics, Pioneer Electronics, Sony, Thomson Multimedia, Time Warner, Toshiba и Victor Company of Japan) создали организацию DVD Consortium (в 1997 г. переименована в DVD Forum). Новые диски получили название DVD. Первоначально эта аббревиатура означала цифровой видеодиск (Digital Video Disk). Однако, в связи с тем, что новый продукт годился не только для записи видеофильмов, но и для хранения компьютерных данных и программ, ее стали расшифровывать как цифровой универсальный диск (Digital Versatile Disk). Окончательное соглашение, определяющее спецификации на DVD-видеоплееры, накопители DVD-ROM и DVD-диски, было утверждено лишь в апреле 1997 г. (первые устройства и диски появи-

лись в продаже несколько раньше). Новый формат отвечает всем условиям и требованиям, предъявляемым как кинопромышленностью в лице Studio Advisory Committee, так и компьютерной индустрией в лице Technical Working Group.

Логический формат DVD-видео обеспечивает разрешение 720×480 точек в NTSC и 720×576 в PAL формата MPEG-2 с переменной скоростью 3–9,8 Мбит/с, в зависимости от того, насколько быстро происходят изменения в кадрах в каждый момент. В среднем, как правило, эта величина составляет 3,5 Мбит/с. Соответственно, два часа видео без меню и прочих добавок занимают около 3 Гб. DVD обеспечивает более четкое и качественное изображение, чем лазерный диск (LD) и более насыщенный звук, чем CD. Помимо этого возможен многоканальный звук на нескольких языках, субтитры, разные соотношения сторон изображения, множество опций воспроизведения, поддерживается также система сложных динамичных меню и перекрестных ссылок.

Внутреннее устройство DVD. Внешне диски DVD выглядят как обычные диски CD-ROM. Диски DVD имеют тот же диаметр (120 мм) и толщину (1,2 мм), что и традиционные компакт-диски, однако в отличие от последних, запись на них может производиться с обеих сторон (хотя допускается и односторонняя запись). Основное требование при разработке DVD было простым: увеличить емкость хранимых данных за счет расположения как можно большего числа питов вдоль треков на диске, при этом технология изготовления должна быть дешевой.

Результатом исследований стала разработка более высокочастотного полупроводникового лазера с меньшей длиной волны, вследствие чего стало возможным использовать питы более маленького размера [38]. Кроме того, для новых моделей приводов DVD-ROM был разработан принципиально новый оптический канал. В конструкции этого канала нашли применение несколько инноваций. Прежде всего речь идет о «голографическом модуле». В его состав входит полупроводниковый лазер нового типа. Этот тип лазеров называется VCSEL (Vertical Cavity Surface Emitting Laser — лазер с вертикальным эмиттированием через полость в поверхности) и отличается от обычных полупроводниковых лазеров направлением, в котором формируется лазерный луч: если обычно луч проходит в продольном направлении активного слоя сквозь его торец (торцевые лазеры), то в новом устройстве излучение происходит в поперечном направлении через специальное отверстие в электроде (рис. 34). Основные преимущества применения новой лазерной технологии VCSEL:

- значительное снижение потребляемой мощности: если торцевые лазеры потребляют 50 мА, то VCSEL — только 8 мА. Хотя речь



Рис. 34. Лазер VCSEL

идет о малых токах, нужно учитывать и размеры активного элемента, и достигаемый уровень надежности при почти трехкратном снижении потребляемого тока;

- повышение эффективности за счет снижения потерь при формировании пятна от лазерного луча округлой формы. В обычных торцевых устройствах форма пятна от луча довольно далека от требуемой формы, тогда как в VCSEL луч необходимого профиля формируется уже при излучении;
- снижение температуры, выделяемой активным узлом. Для VCSEL-устройств она составляет $70\text{ }^{\circ}\text{C}$, а для торцевых лазеров — $100\text{ }^{\circ}\text{C}$. Снижение температуры приводит к упрощению конструкции и повышению общей надежности устройства.

Оригинальность оптического канала основывается также на конструкции объектной линзы с кольцеобразной маской. Благодаря применению этой инновации привод DVD может работать с дисками CD/CD-R/CD-RW.

И все-таки, главное отличие DVD от CD заключается, естественно, в увеличенной плотности записи информации. За счет перевода считывающего лазера из инфракрасного диапазона (длина волны 780 нм) в красный (с длиной волны 650 или 635 нм), усовершенствования оптической головки, увеличения числовой апертуры объектива до 0,6 (против 0,45 в CD), достигается более чем двукратное уплотнение дорожек и снижение длины питов.

Емкость, стороны и слои DVD. Большинство дисков DVD имеют емкость 4,7 Гб. Но применение схем удвоения плотности и их комбинирования позволяет иметь диски большей емкости: от 8,5 и 9,4 Гб до 17 Гб.

Существуют следующие структурные типы DVD.

Single Side / Single Layer (односторонний/однослойный): это самая простая структура DVD-диска. На таком диске можно разместить до 4,7 Гб.

Single Side / Dual Layer (односторонний/двухслойный): этот тип дисков имеет два слоя данных. Первый слой делается полупрозрачным, таким образом, используя двойную фокусировку, можно считывать данные с обоих слоев. Оба слоя считаются с одной стороны и на таком диске можно разместить 8,5 Гб данных.

Double Side / Single Layer (двусторонний/однослойный): производители могут изготавливать диски DVD толщиной 0,6 мм, что вдвое меньше толщины стандартного диска CD. Это дает возможность соединить два диска обратными сторонами и получить емкость в 9,4 Гб (по 4,7 Гб на каждой стороне).

Double Side / Double Layer (двусторонний/двухслойный): структура этого диска обеспечивает возможность разместить на нем до 17 Гб данных (по 8,5 Гб на каждой стороне).

Защита авторских прав. В связи с возможностью хранить на DVD отличные цифровые копии фильмов кинокомпании подготовили комплекс мер по защите авторских прав. Основная их часть — это шифрование видеоданных, размещаемых на DVD. Сейчас наибольшее распространение получил такой способ защиты, как запись DVD-диска для специальной зоны. Весь мир территориально разделен на 6 зон, к которым приписываются как сами проигрыватели, так и диски¹:

- 1) Северная Америка;
- 2) Япония, Европа, Средний Восток, Южная Африка;
- 3) Юго-восточная Азия (включая Гонг-Конг);
- 4) Австралия, Новая Зеландия, Центральная и Южная Америка;
- 5) Северо-западная Азия (куда входит Россия), Северная Африка;
- 6) Китай.

7.4. Диски Blu-ray

Blu-ray Disc, BD — формат оптического носителя, используемый для записи и хранения цифровых данных, включая видео высокой четкости с повышенной плотностью. Изначально продвижением и разработкой стандарта занимались следующие компании: Sony, Matsushita (Panasonic),

¹ Конечно, существуют способы изменить настройки аппаратуры, а в некоторых случаях вообще можно превратить плеер в мультизонный.

Samsung, LG, Philips, Thomson, Hitachi, Sharp и Pioneer. В последнее время компания Toshiba отказалась от продвижения своего конкурирующего формата HD DVD и также примкнула к этому консорциуму.

Как следует из названия, в Blu-ray для записи и воспроизведения диска вместо красного лазера, который используется в DVD и CD-ROM, применен синий лазер (blue-violet laser), что обусловило название стандарта. Интересно, что буква «е» была намеренно исключена из слова «blue», чтобы получить возможность зарегистрировать торговую марку, так как выражение «blue ray» является часто используемым и не могло быть зарегистрировано.

У синего лазера длина волны составляет 405 нм, что значительно меньше длины волны красного лазера (650 нм). Меньшая длина волны — соответственно меньшая интерференция отраженного луча, что позволило сузить дорожку вдвое по сравнению с обычным DVD-диском (до 0,32 мкм) и увеличить плотность записи данных. Это в свою очередь привело к значительному увеличению емкости носителя.

Согласно объявленной спецификации, Blu-ray Disc — перезаписываемый диск следующего поколения со стандартным CD/DVD размером 12 см с максимальной емкостью записи на один слой и одну сторону до 27 Гб. Таким образом, однослойный диск Blu-ray может хранить 23,3/25/27 или 33 Гб, двухслойный диск может вместить 46,6/50/54 или 66 Гб. Ожидается появление дисков BD вместимостью 100 и 200 Гб с использованием соответственно четырех и восьми слоев. В зависимости от используемого метода сжатия видео, на один слой возможно записать от двух до четырех часов видео высокой четкости.

7.5. Накопители на флэш-памяти

После рассмотрения второй группы носителей — оптических дисков обратимся к накопителям с использованием флэш-памяти. Такое название накопителей связано с типом использованного в них ППЗУ¹ — энергонезависимой перезаписываемой оперативной памяти (FLASH). Сочетая малые размеры и энергонезависимость с достаточно большими объемами памяти, они стали почти идеальным средством для хранения информации в портативных устройствах.

К наиболее типичным представителям семейства Flash-карт можно отнести: *Memory Stick* (MS, размер 21,5×50×2,8 мм), *MultiMedia Card*

¹ Полупостоянное запоминающее устройство.

(MMC, $32 \times 24 \times 1,4$ мм), Secure Digital Memory Card (SD Card, $32 \times 24 \times 2,1$ мм), а также SmartMedia Card и Compact Flash Card.

Уже по приведенному списку можно сделать вывод, что война форматов между производителями на этом рынке далека от завершения. Независимые производители автономных устройств для считывания Flash-карт стараются обеспечить мультиплатформенную совместимость, выпуская универсальные считыватели (так называемые Card-readers).

Кроме того, набирают популярность так называемые твердотельные накопители, используемые в портативных компьютерах и других мобильных смарт-устройствах. Такие накопители конструктивно также представляют собой карты флэш-памяти с увеличенной емкостью и быстродействием.

7.6. Интерфейсы передачи мультимедийных данных

Основным требованием к интерфейсам передачи мультимедийных данных являются скорость передачи (в том числе минимально гарантированная пропускная способность), простота исполнения и использования. Выполнение этих требований позволяет объединять такими интерфейсами разнородное мультимедийное оборудование для передачи, обработки и хранения данных. На сегодняшний момент этим требованиям в той или иной степени отвечают несколько интерфейсов. Из наиболее распространенных интерфейсов передачи MultiMedia данных на основе кабельных соединений назовем FireWire (IEEE 1394) и USB X.0. К беспроводным интерфейсам с некоторой степенью условности можно отнести технологии Bluetooth и WiFi (IEEE 802.11)¹.

IEEE 1394.

IEEE 1394, или FireWire, — это последовательная высокоскоростная шина, предназначенная для обмена цифровой информацией между компьютером и другими электронными устройствами [39]. Благодаря невысокой цене и большой скорости передачи данных эта шина получила широкое распространение в виде стандарта интерфейса ввода-вывода. Ее изменяемая архитектура и одноранговая топология делают FireWire идеальным вариантом для подключения жестких дисков и устройств обработки аудио- и видеинформации, прежде всего видео-

¹ Эти беспроводные интерфейсы разрабатывались как универсальные и обладают значительно меньшей скоростью передачи данных.

камер. Эта шина также идеально подходит для работы мультимедийных приложений в реальном времени.

В зависимости от версии стандарта, поддерживается пропускная способность шины на уровнях 100, 200 и 400 Мбит/с. В зависимости от возможностей подключенных устройств, одна пара устройств может обмениваться сигналами на скорости 100 Мбит/с, в то время как другая на той жешине — на скорости 400 Мбит/с. Такие высокие показатели пропускной способности последовательной шины практически исключают необходимость использования параллельных шин.

FireWire удовлетворяет всем основным требованиям к мультимедийным интерфейсам:

- цифровой интерфейс; предельная теоретическая длина шины 224 м;
- тонкий кабель и небольшой размер разъемов;
- отсутствие терминаторов, идентификаторов устройств или необходимости предварительной установки. Максимальное расстояние между двумя устройствами в цепочке по IEEE-1394a — 4,5 м, по IEEE-1394b — 100 м;
- возможность переконфигурировать шину без выключения компьютера;
- небольшая стоимость для конечных пользователей;
- скорость передачи данных до 400 Мбит/с по стандарту IEEE-1394a и 800 Мбит/с по стандарту IEEE-1394b;
- произвольная топология — равноправие устройств, допускающее различные конфигурации. 16-ти разрядный адрес позволяет адресовать до 64 К узлов на шине;
- возможность обработки мультимедийного сигнала в реальном времени с гарантированной пропускной способностью;
- открытая архитектура — отсутствие необходимости использования специального программного обеспечения.

Провода питания рассчитаны на ток до 1,5 А при напряжении от 8 до 40 В и поддерживают работу всей шины, даже когда некоторые устройства выключены. Они также делают ненужными кабели питания во многих устройствах.

Протокол IEEE-1394 реализует три нижних уровня эталонной модели Международной организации по стандартизации OSI: *физический, канальный и сетевой*. Кроме того, существует менеджер шины, которому доступны все три уровня. На физическом уровне обеспечивается электрическое и механическое соединение с коннектором, на других уровнях — соединение с прикладной программой.

Шина USB.

Шина USB (Universal Serial Bus) — универсальная последовательная шина. Первый утвержденный вариант стандарта 1.0 появился 15 января 1996 г. [40]. Разработка стандарта была инициирована фирмами Intel, DEC, IBM, NEC, Northern Telecom и Compaq.

Основная цель стандарта, поставленная перед его разработчиками, — создать реальную возможность пользователям работать в режиме Plug&Play с периферийными устройствами. Это означает, что должно быть предусмотрено подключение устройства к работающему компьютеру, автоматическое распознавание его немедленно после подключения и последующей установки соответствующих драйверов. Кроме этого, желательно питание маломощных устройств подавать с самой шины. Скорость шины должна быть достаточной для подавляющего большинства периферийных устройств. Попутно решается историческая проблема нехватки ресурсов на внутренних шинах IBM PC совместимого компьютера — контроллер USB занимает только одно прерывание независимо от количества подключенных к шине устройств.

Технические характеристики USB, такие как скорость обмена и максимальная длина кабеля, существенно различаются в зависимости от версии стандарта

Топология USB практически не отличается от топологии обычной локальной сети на витой паре, обычно называемой «звездой». Даже терминология похожа — размножители шины также называются «хабами». Вместо любого из устройств в одном из узлов звезды также может стоять хаб. Основное отличие от топологии обычной локальной сети — компьютер (или host устройство) может быть только один. Хаб может быть как отдельным устройством с собственным блоком питания, так и встроенным в периферийное устройство. Наиболее часто хабы встраиваются в мониторы и клавиатуры.

Так как обмен данными по USB идет только между компьютером и периферийным устройством (между устройствами обмена нет), то устройства с большими объемами приема и/или передачи данных должны подключаться либо к самому компьютеру, либо к свободному ближайшему узлу.

Интерфейс беспроводной связи на малых расстояниях — Bluetooth.

В начале 1998 г. Ericsson, IBM, Intel, Toshiba и Nokia объединились для совместной разработки технологии беспроводного соединения мобильных устройств. 20 мая 1998 г. произошло официальное представление специальной рабочей группы (SIG — Special Interest Group),

призванной обеспечить беспрепятственное внедрение технологии, получившей название Bluetooth¹. Вскоре в группу вошли 3COM/Palm, Axis Communication, Motorola, Compaq, Dell, Qualcomm, Lucent Technologies, UK Limited, Xircom. Благодаря своей простоте, технология Bluetooth нашла свое применение в области создания небольших локальных сетей и беспроводного объединения устройств в пределах офисов и жилых домов.

В отличие от технологии инфракрасной связи IrDA (Infrared Direct Access), работающей по принципу «точка — точка» в зоне прямой видимости, технология Bluetooth разрабатывалась для работы как по принципу «точка — точка», так и в качестве многоточечного радиоканала, управляемого многоуровневым протоколом, похожим на протокол мобильной связи GSM.

Технология использует небольшие приемопередатчики малого радиуса действия, либо непосредственно встроенные в устройство, либо подключаемые через свободный порт или PC-карту. АдAPTERы работают в радиусе 10–40 м и, в отличие от IrDA, не обязательно в зоне прямой видимости, т. е. между соединяемыми устройствами могут быть различные препятствия или стены.

Bluetooth работает на частоте 2,45 ГГц (полоса промышленного, научного и медицинского применения ISM — Industry, Science, Medicine). Радиоканал обеспечивает скорость 721 Кбит/с и передачу трех голосовых каналов. Сейчас до 80 % выпускаемых мобильных телефонов оснащаются устройствами Bluetooth. На рынке появилось большое количе-

¹ Историческая справка. Буквальный перевод слова Bluetooth с английского — «синие зубы». Однако своим названием технология Bluetooth обязана вовсе не дантистам, а историческому недоразумению. Так английские летописцы обозвали датского короля викингов, жившего в 910–940 гг. Король Гаральд Блютус (Harald Bluetooth) вошел в историю как собиратель земель скандинавских. В частности, ему приписывается объединение Дании и Норвегии (а технология Bluetooth должна объединить телекоммуникационную и компьютерную индустрию). Вероятно, по аналогии с этим технология Bluetooth также призвана объединить мир мобильной электроники.

Есть две версии происхождения синезубого королевского прозвища. Первая — у него действительно были зубы патологического цвета. Но правдоподобнее всего выглядит другая версия: Bluetooth — исковерканное на английский манер прозвище короля на языке викингов: у короля была смуглая кожа и темные волосы, что нетипично для расы викингов (белокожих и светловолосых). За этот «генетический кульбит» Гаральда прозвали Чернявеньким (в скандинавском оригинале — Bletand).

ство различных передатчиков, подключаемых через PC-карты, USB, или через CF-разъемы, таких как, например, беспроводные телефонные гарнитуры hands-free.

В разрезе изучения мультимедийных технологий, можно упомянуть разработку компанией Toshiba устройства, использующего Bluetooth для проведения видео-конференций на базе видеостандарта MPEG-4.

Беспроводной интерфейс IEEE 802.11.

Еще одной эффективной технологией для организации беспроводных сетей является технология IEEE 802.11 (WiFi), использующая диапазон частот, свободных в большинстве стран от лицензирования и получения разрешений на использование радиочастот [41]. Основное распространение получили спецификации 802.11 версий a, b, e, g, h, k.

Радиус сотов, если не применять специальных направленных антенн, обычно не превышает 100 м, а скорость передачи заметно падает при удалении от базовой станции. Для 802.11b критическими являются расстояния около 30 м (скорость передачи 5,5 Мбит/с) и 50 м, где скорость падает до 2 Мбит/с. В случае 802.11a на расстоянии 30 м этот показатель падает уже втрое до 18 Мбит/с, а на 50 м — обычно не превышает 6 Мбит/с.

Кроме собственно выбора радиочастотного диапазона и типа модуляции одной из самых больших проблем беспроводных сетей с момента их появления была *низкая защищенность от перехвата данных*. В отличие от сетей кабельных, где перехват информации едва ли возможен без физического доступа к среде передачи (собственно кабелю или коммутационному оборудованию), беспроводные сети оказываются практически беззащитными, если не применять специальных мер шифрования трафика.

Еще одна проблема, решение которой значительно отличается от аналогичного для проводных сетей, — обеспечение прикладных сервисов требуемой полосой пропускания и допустимой задержкой передачи данных (QoS). Трудность ее решения в том, что количество ошибочных пакетов в беспроводных сетях может достигать 10–20 %, скорость передачи данных зависит от местоположения клиента и может изменяться во время соединения.

7.7. Мультимедийное резюме

В данной главе была рассмотрена специфика хранения и передачи мультимедийных данных, связанная прежде всего со значительными объемами и требованием равномерной скорости сохранения и передачи информации.

Среди трех основных групп носителей мультимедийной информации своим разнообразием выделяются оптические диски. Технологии CD-ROM, DVD, Blu-ray не полностью сменили друг друга и используются в различных технологических нишах.

Интерфейсы передачи мультимедийных данных классифицируются по способу соединения: кабельные и беспроводные. Из типичных представителей первого семейства по своей популярности выделяются интерфейсы FireWire (IEEE 1394) и USB X.0. Среди беспроводных стандартов следует отметить формат Bluetooth, широко используемый в миниатюрных цифровых устройствах, и семейство WiFi, предназначеннное для быстрой передачи значительных объемов данных.

ГЛАВА 8

ЦИФРОВЫЕ ТЕХНОЛОГИИ ЗАПИСИ И ВОСПРОИЗВЕДЕНИЯ ЗВУКА

Интерференция и гармонический анализ. Амплитудно-частотная характеристика. Фазовые и нелинейные искажения. Аддитивная и частотная модуляция. Синтез на основе таблиц волн. Устройство звуковой платы. MIDI-интерфейс.

Независимо от области применения, компьютер — это прежде всего средство сбора, хранения и обработки информации. Звук — один из важнейших носителей информации, соответственно появление аудиоустройств в составе компьютера закономерно. Для понимания процессов цифровой обработки аудиоданных рекомендуется предварительно ознакомиться с основными физическими сведениями о звуковых волнах и психофизиологией восприятия звука, подробно рассмотренных в книге [1].

8.1. Спектральное представление звука

Звуковая волна простейшей формы — чистый звуковой тон определенной частоты есть синусоидальное, или гармоническое, колебание. Для ее описания используют следующее уравнение:

$$x(t) = x_m \sin(2\pi ft + \varphi),$$

где x_m — амплитуда колебаний; $\pi = 3,141\dots$ — число «пи»; f — частота колебаний; φ — фазовый угол.

Частота f характеризует число периодов колебания в единицу времени — секунду. Колебания можно также охарактеризовать периодом повторения $T = 1/f$. Период — это время, в течение которого завершается один полный цикл периодического колебания (например, от одной вершины синусоиды до другой). Период колебаний или частота определяют тональность (высоту) звука, а амплитуда колебаний — его громкость.

Чисто синусоидальные звуковые колебания встречаются в природе крайне редко. Большинство звуковых сигналов имеют резко несинусоидальную форму. Они могут быть повторяющимися — периодическими

(например, звук скрипки при равномерном движении струны) или непериодическими (например, звуки речи).

Если частицы воздуха подвергаются одновременному воздействию нескольких волн, то наблюдается явление *интерференции* — сложения волн. При сложении волн с разными частотами и амплитудами, даже если эти волны синусоидальны, результирующая волна может иметь весьма сложный вид. При сложении двух простых синусоидальных колебаний получается периодическое колебание сложной формы. Складывая не два, а большее количество синусоидальных колебаний с кратными частотами, с различными амплитудами и фазами, можно каждый раз получать все новые и новые результирующие колебания, но всякий раз результатом сложения будет периодическое колебание.

Справедливо и обратное: всякое периодическое колебание, форма которого как угодно сложна, можно представить как сумму гармонических колебаний, имеющих кратные частоты. При цифровой обработке звука важную роль играет теорема Фурье, гарантирующая, с одной стороны, возможность синтеза колебаний сложной формы на основе гармонических колебаний, а с другой стороны — возможность разложения сложных колебаний в ряд простых. Разложение сложного звука в ряд простых тонов называется *гармоническим анализом*.

На рис. 35 показано, как с помощью сложения синусоидальных колебаний — гармоник создается симметричный сигнал прямоугольной формы, называемый меандром. Такой сигнал описывается следующим рядом, в котором учитываются только нечетные гармоники:

$$X(t) = \sum_{k=1}^{k=\infty} \frac{X_m \sin(2\pi k f_1 t)}{k(2\pi k f_1)}, \text{ где } k \text{ — нечетное число.}$$

Теоретически предполагается, что синтез ведется по бесконечному числу гармоник (номер гармоники k меняется от 1 до бесконечности), на практике же число гармоник всегда ограничивают. Так, на рисунке представлен синтез положительной полуволны меандра (отрицательная полуволна симметрична) для разного, но конечного числа гармоник. При этом для симметричных колебаний амплитуды четных гармоник равны 0, и они при синтезе не используются.

С ростом числа гармоник результирующее колебание приближается к прямоугольному колебанию. Однако в моменты разрывов меандра возникают характерные выбросы — *осцилляции*. Их появление называется эффектом Гиббса и связано с подменой бесконечного ряда гармоник конечным. К счастью, реальные звуковые колебания не имеют раз-

рывов, что облегчает их синтез конечным числом гармоник, но не исключает этот эффект вообще. Для подавления эффекта Гиббса служат специальные фильтры, ослабляющие высшие гармоники. Чем плавнее идет спад частотной характеристики фильтра на высоких частотах, тем меньше проявление эффекта Гиббса.

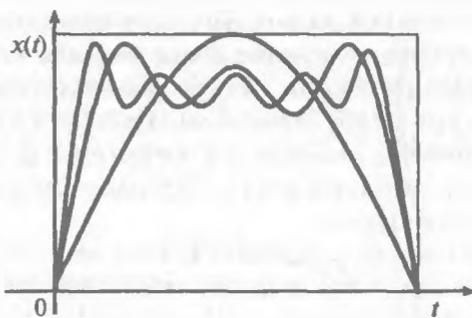


Рис. 35. Синтез меандра по гармоникам

Отложим по горизонтальной оси частоты синусоидальных колебаний, составляющих сигнал, а по вертикальной оси — их амплитуды. Высота каждой линии равна амплитуде соответствующего гармонического колебания. Такое представление сложного колебания (см. рис. 36) называется *амплитудным спектром*. Периодическому сигналу соответствует линейчатый амплитудный спектр.

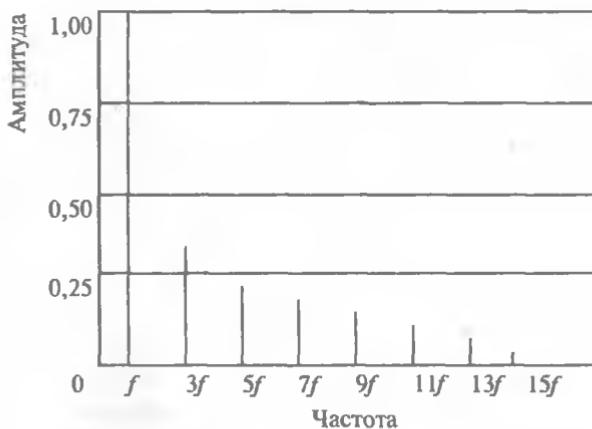


Рис. 36. Амплитудный спектр периодического сигнала

Такое представление сложного колебания весьма наглядно, однако оно не содержит информации о фазе гармоник. Так как ухо человека при восприятии звукового сигнала не реагирует на начальные фазы отдельных составляющих — важен лишь спектр амплитуд.

Непериодические колебания также можно представить в виде суммы гармонических колебаний, но частоты колебаний заполняют весь спектр, образуя сплошной, или непрерывный спектр. Непериодические колебания выражаются не через ряд Фурье, а через интеграл Фурье.

Отдельный прямоугольный импульс имеет сплошной спектр, в котором присутствуют колебания всевозможных частот. Фрагмент синусоиды, число полных колебаний в котором равно, например шести, также будет иметь сплошной спектр. Фрагмент синусоиды не описывает периодический процесс.

Так как в природе бесконечных сигналов не существует, то имеет смысл говорить лишь о том, в какой степени реальный процесс приближается к периодическому. Если число колебаний в течение рассматриваемого промежутка времени велико, то не будет большой ошибкой замена ограниченной во времени синусоиды на неограниченную. В то же время фрагмент синусоиды, состоящий из небольшого числа колебаний, нельзя заменить неограниченной во времени синусоидой.

8.2. Параметры устройств электроакустики

К сожалению, все известные устройства электроакустики неизбежно создают искажения звуковых сигналов. Под искажениями подразумевают отличия формы колебаний после их усиления, преобразования и обработки от формы исходного сигнала. Самыми распространенными являются линейные и нелинейные искажения, а также шумы и наводки.

Линейные искажения проявляются в неодинаковом усилении, передаче или воспроизведении различных гармонических составляющих спектра звуковых сигналов независимо от их уровня. Обычно их оценивают относительным уровнем снижения амплитуды гармонического сигнала $X_m(f)$ на заданной частоте f по сравнению с амплитудой сигнала $X_m(f_0)$ на некоторой средней частоте f_0 — обычно это частота около 1 кГц. Уровень снижения (иногда напротив, увеличения) оценивается логарифмической величиной.

$$Y(f) = 20 \log(X_m(f)/X_m(f_0)).$$

Эта величина используется также для оценки неравномерности так называемой амплитудно-частотной характеристики (АЧХ) сквозного

тракта аудиоустройств и их отдельных звеньев, например, микрофона, усилителя и громкоговорителя. АЧХ характеризует зависимость от частоты уровня выходного сигнала электроакустических устройств или пропорциональных этому уровню величин — например, коэффициента усиления электронного усилителя.

Как правило, у HiFi-устройств неравномерность АЧХ не превосходит 1 дБ. Для снятия АЧХ сквозного канала используют генераторы звуковых частот и измерители амплитудного или среднего значения звуковых сигналов. Типичный вид АЧХ сквозного тракта электроакустических устройств представлен на рис. 37.

- $K(f)/K(f_0)$

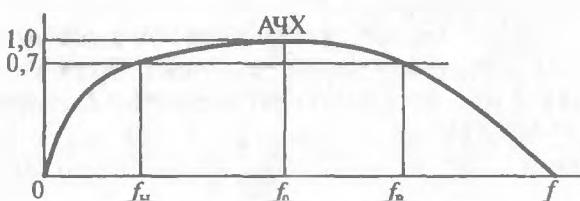


Рис. 37. Амплитудно-частотная характеристика

АЧХ может строиться в линейном или логарифмическом масштабе по оси частот. Поскольку диапазон звуковых частот достаточно широк, логарифмическое представление часто оказывается более удобным.

АЧХ всех реальных устройств имеет спады в области низких и высоких частот. Если они находятся за пределами звукового диапазона частот (от 20 до 20 кГц), то играют даже положительную роль — отсеивают сигналы, которые слух не воспринимает, но которые могут вызвать перегрузку акустических систем или даже барабанной перепонки уха. Если же спады АЧХ лежат в пределах звуковых частот, то они ведут к заметному ухудшению качества звучания.

Хотя слух людей нечувствителен к фазовым искажениям отдельных синусоидальных сигналов и даже их совокупности, есть убедительные данные о том, что мы способны оценивать различие форм сигналов, обусловленные фазовыми искажениями и ограничениями крутизны изменения звуковых сигналов. Поэтому там, где это возможно, разработчики HiFi-аппаратуры резко ужесточают требования к ней. К примеру, многие современные усилители высокого класса усиливают сигналы в диапазоне частот от 0 (постоянный ток) до сотен кГц и даже

нескольких МГц при практически нулевых фазовых искажениях в рабочем диапазоне частот.

Следующий тип искажений — нелинейные искажения. Причины их возникновения очень разнообразны. К примеру, у микрофона может быть неравномерность магнитного поля, в котором перемещается катушка, или конструктивное ограничение пределов перемещения мембранны. Те же причины ведут к возникновению нелинейных искажений у громкоговорителей (динамиков). Здесь они намного выше, так как амплитуда колебаний диффузора велика и нередко достигает нескольких миллиметров. При большой громкости и воспроизведении звуков ударных инструментов это хорошо заметно на глаз: диффузор динамика как бы «выпрыгивает» из него или «всасывается» внутрь.

Усилители построены на принципиально нелинейных активных приборах — электронных лампах, биполярных или полевых транзисторах, либо на интегральных микросхемах с этими приборами. К тому же их мощность всегда ограничена энергетическими возможностями источников электропитания.

Нелинейные искажения характерны тем, что степень их проявления зависит от амплитуды сигналов и вида сквозной передаточной характеристики устройств электроакустики. Под такой характеристикой подразумевается зависимость мгновенного значения сигнала на выходе устройства от мгновенного значения сигнала на входе. Лишь небольшой участок этой характеристики бывает линейным, в целом характеристика нелинейная. Это означает, что если входной сигнал синусoidalный, то выходной будет отличен от синусоиды.

Из спектрального представления сигналов следует, что нелинейные искажения порождают в выходном сигнале новые гармонические составляющие, отсутствующие во входном сигнале. Поэтому количественной мерой величины нелинейных искажений принято считать относительный уровень высших гармонических составляющих, возникающих при подаче на вход устройства синусоидального сигнала. Обычно этот уровень (коэффициент нелинейных искажений, или гармоник) измеряется в процентах:

$$K_{\Gamma} = \frac{\sqrt{X_{m_2}^2 + X_{m_3}^2 + X_{m_4}^2 + \dots}}{X_{m_1}} \cdot 100 \text{ \%}.$$

Еще совсем недавно уровень $K_{\Gamma} < 1 \text{ \%}$ считался вполне достаточным. Однако в наше время даже средние HiFi-усилители имеют $K_{\Gamma} < 0,01\%$. На качество звучания оказывает значение не только величина амплитуд высокочастотных гармоник, но и то, насколько быстро убывает их

спектр по мере роста номеров (частот) гармоник. Не случайно усилители на приборах с малой нелинейностью (электронных лампах и полевых транзисторах) дают качество звучания явно выше, чем усилители на обычных биполярных транзисторах.

Теоретически для вычисления амплитуд гармоник используется спектральный анализ колебаний. Он реализуется достаточно сложными и трудоемкими способами. Ускоренный анализ, как и синтез, может реализовываться с помощью специального алгоритма — быстрого преобразования Фурье (БПФ, или по-английски FFT). БПФ успешно используется для обработки сигналов в ряде мультимедийных программных систем.

Существует еще один вид искажений — **интермодуляционные искажения**. Они обусловлены тем, что в нелинейных системах сильный сигнал с одной частотой (чаще всего низкой) может вызвать амплитудную модуляцию другого, более слабого сигнала. Чистота звучания при этом ухудшается.

Есть и такой коварный вид искажения, как **биения**, т. е. появление разностной частоты при воздействии на нелинейную систему двух высокочастотных гармонических колебаний с незначительно отличающимися частотами. Биения могут сильно ухудшить качество воспроизведения звука — многие знают это из практики прослушивания радиопередач, когда звук принимаемой станции сопровождается биениями — сильным свистом от работающей рядом станции. Еще хуже, если частоты биений оказываются порядка 1 кГц, поскольку ухо наиболее чувствительно к этим частотам.

Еще один важный параметр электроакустических устройств — относительный уровень помех, наводок и шумов. **Помехами** обычно называют различные случайные звуки, отсутствующие в исходном звуковом сигнале. Есть два особых вида помех — шумы и фон. **Шумы** — это характерное шипение, например, грампластинки, магнитофонной ленты или не очень хорошего усилителя. А **фон** связан с тем, что большинство электроакустических устройств питается от промышленной сети переменного тока с напряжением 220 В и частотой 50 Гц. Это напряжение проникает тем или иным способом в цепи усилителя и воспринимается как гул частоты 50 Гц.

Уровень помех (УП) определяется как:

$$\text{УП} = 20 \log(U_{\text{Пом}} / U_{\text{МАКС}}),$$

где $U_{\text{МАКС}}$ — максимальный уровень напряжения выходного сигнала (звука), $U_{\text{Пом}}$ — уровень помех.

К примеру, если $U_{\text{МАКС}}$ в 1000 раз превосходит $U_{\text{ПОМ}}$, то УП = –60 дБ. У современных устройств электроакустики он достигает –100 дБ. Это фактически означает ничтожно малые уровни шумов и помех, достижимые только при использовании цифровых методов записи и воспроизведения звуковых сигналов.

Наконец, устройства электроакустики характеризуются *динамическим диапазоном* (ДД):

$$\text{ДД} = 20 \log(U_{\text{МАКС}} / U_{\text{МИН}}),$$

где $U_{\text{МИН}}$ — минимальный различимый на уровне помех (в том числе и акустических) уровень сигнала. При его вычислении надо учитывать не только электрические шумы и наводки, но и типовой уровень акустических помех в помещении — например, шумы улицы, бытовых приборов, разговора и т. д.

Значение ДД, равное 55–60 дБ, считается приемлемым для HiFi-устройств. Сообщения об устройствах с динамическим диапазоном до 90–100 дБ — не более чем рекламный трюк разработчиков электроакустической аппаратуры. Такой динамический диапазон едва ли достижим в реальных условиях прослушивания.

Помимо рассмотренных параметров мультимедийных устройств электроакустики, важное значение могут иметь и другие параметры. К примеру, усилители на плате аудиоадаптера характеризуются выходной мощностью при заданном коэффициенте нелинейных искажений. Если она имеет порядок нескольких ватт и усилитель рассчитан на низкоомную нагрузку (обычно 4 Ом), значит, смело можно подключать к их выходам звуковые колонки небольшой мощности — либо специально входящие в комплект мультимедийные устройства, либо от других устройств электроакустики, например FM-ресивера. Если мощность усилителя составляет лишь доли ватта, то к нему можно подключать только стереотелефоны или активные акустические колонки со своими усилителями мощности и источниками питания.

8.3. Основные методы синтеза звука

Звуки, созданные с помощью музыкальных инструментов, состоят из нескольких характерных фаз.

Атака (attack) — начальная фаза образования звука, в течение которой уровень сигнала быстро нарастает до максимального значения, затем немного спадает. Длительность атаки для различных музыкальных

инструментов варьируется от единиц до нескольких десятков или даже сотен миллисекунд;

Поддержка (sustain) — следующая после атаки фаза, в течение которой уровень сигнала примерно постоянен. Ощущение высоты звука формируется как раз во время поддержки;

Затухание (release) — участок относительно быстрого уменьшения величины сигнала. Данная фаза является последней.

Огибающая колебаний во время атаки, поддержки и затухания называется *амплитудной огибающей* (рис. 38).

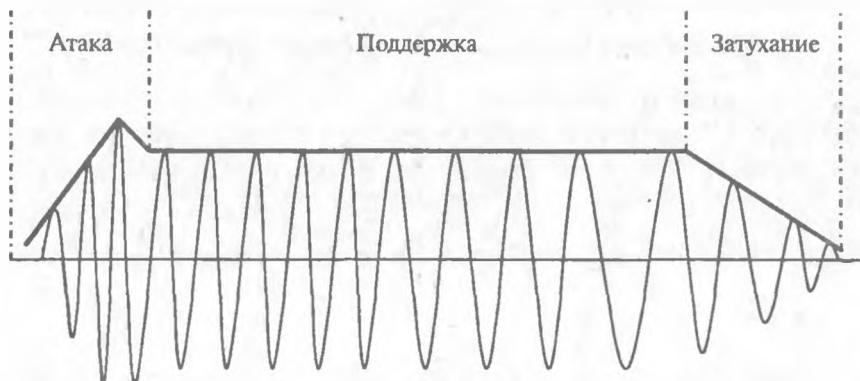


Рис. 38. Амплитудная огибающая звукового сигнала

Различные музыкальные инструменты имеют разную форму амплитудной огибающей, тем не менее, отмеченные фазы характерны практически для всех инструментов. В процессе воспроизведения звука его спектр изменяется во времени, т. е. каждая гармоника имеет свою огибающую, отличную от результирующей. Для создания электронного аналога реального звука необходимо воссоздать огибающие всех гармоник, из которых состоит реальный звук.

Рассмотрим основные методы синтеза звука, применяемые в синтезаторах музыкальных инструментов.

Аддитивный синтез. Суть метода состоит в том, что результирующий звук формируется путем сложения нескольких исходных звуковых волн. При цифровом аддитивном синтезе отдельно формируются N гармоник с частотами от $f_1(t)$ до $f_N(t)$ и амплитудами от $A_1(t)$ до $A_N(t)$, зависящими от времени. Затем эти гармоники складываются. Огибающие всех N составляющих процессор формирует при воспроизведении звука. Среди частот гармоник могут быть и не кратные. Для получения

всех частотных точек звукоряда необходимо использовать несколько сотен составляющих.

Для генерации синусоидального сигнала используют ГУК (генератор, управляемый кодом) (рис. 39).



Рис. 39. Структурная схема генератора, управляемого кодом

ГУК состоит из формирователя фазы, собранного на сумматоре и регистре; ПЗУ, в котором записаны значения (отсчеты) функции $\sin(x)$ для одного периода в равноотстоящих точках; перемножителя параллельных кодов и ЦАП (цифро-анalogового преобразователя) с ФНЧ (фильтром низких частот) на выходе. Описанный метод позволяет непосредственно формировать сигнал с соответствующим динамическим спектром, но требует значительное количество входных данных: N значений частоты и амплитуды.

Метод волновой формы. Метод волновой формы является разновидностью нелинейного синтеза. При этом методе для синтеза одного музыкального звука используется сигнал одного генератора. Гармоники при этом получают в результате нелинейных искажений (рис. 40).

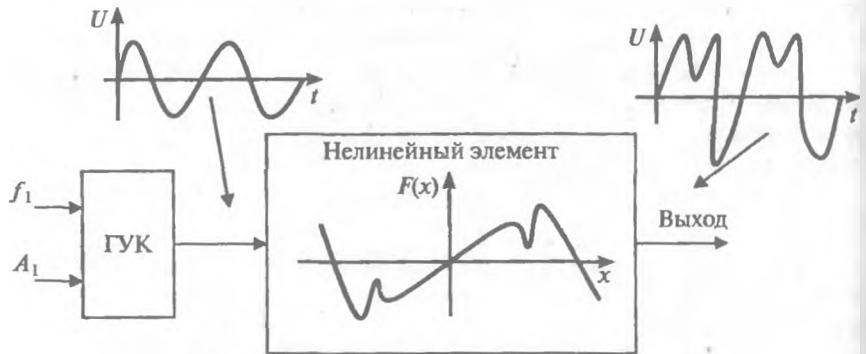


Рис. 40. Схема синтеза звука методом волновой формы

Синусоидальный сигнал с амплитудой A_1 и частотой f_1 пропускают через нелинейный элемент с передаточной характеристикой $F(x)$ и на выходе получают сигнал той же частоты, но с другой амплитудой и обогащенный гармониками. Используя нелинейные элементы специального вида, можно получить спектры, характерные для тех или иных инструментов. Получение динамического спектра не представляет труда, так как спектр зависит от амплитуды A_1 .

Недостаток метода состоит в том, что он не позволяет раздельно управлять амплитудой и спектром выходного сигнала.

Субтрактивный метод. Этот метод синтеза заключается в том, что новый звук создается путем изменения соотношения между отдельными составляющими первоначального колебания. Он реализуется в два этапа. На первом этапе генерируется сигнал, основная частота которого соответствует частоте требуемой ноты. Данный сигнал должен содержать максимально возможное количество гармоник. На втором этапе с помощью частотных фильтров из первоначального колебания выделяют гармоники, характерные для имитируемого звука. Чаще всего в качестве первоначального сигнала используется последовательность коротких прямоугольных импульсов, как наиболее богатый гармониками сигнал.

Синтез звука на основе частотной модуляции. Частотная модуляция — процесс модуляции, при котором мгновенная частота несущего сигнала изменяется под воздействием модулирующего сигнала, а отклонение несущей частоты от среднего значения пропорционально амплитуде модулирующего сигнала. Если в качестве модулирующего и модулируемого сигналов взять синусоидальные, то ЧМ-сигнал можно описать выражением

$$U = A_c \sin[2\pi f_c t + m_f \sin(2\pi f_m t)],$$

где f_c — несущая частота; f_m — частота модулирующего сигнала; m_f — индекс частотной модуляции; A_c — амплитуда несущей.

Если $m_f = 0$, то в спектре выходного сигнала содержится только несущая гармоника. Однако если $m_f \neq 0$, то в спектре появляются составляющие с частотами $f_c \pm Kf_m$, где K — натуральное число. Итак, изменения всего один параметр — индекс модуляции, можно варьировать спектр в широких пределах. Этот принцип положен в основу метода ЧМ-синтеза.

В устройстве использованы два ГУК. Генератор огибающей 1 управляет амплитудой выходного сигнала, генератор огибающей 2 обеспечивает изменение индекса модуляции. Такая упрощенная модель не позволяет получить большого разнообразия изменений спектра, поэтому

в реальных синтезаторах используется не два, а шесть и более ГУК, модулирующих друг друга.

В микросхеме OPL2 используется 18 генераторов и двухоператорная базовая модель синтеза, в OPL3 — 80 генераторов и четырехоператорная базовая модель.

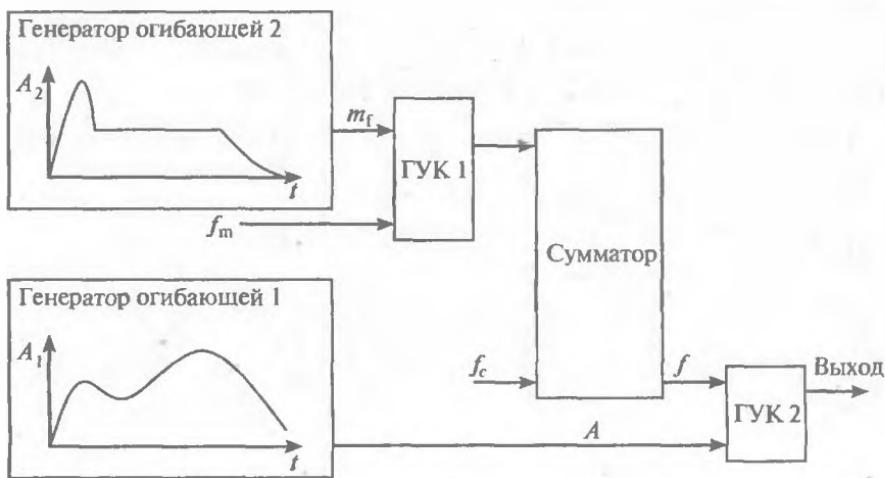


Рис. 41. Блок-схема ЧМ-синтеза

Синтез на основе таблиц волн (Wave Table). Это метод синтеза, основанный на воспроизведении заранее записанных в цифровом виде звучаний инструментов — сэмплов. Для изменения высоты звука сэмпл проигрывается с разной скоростью, а чтобы при этом сильно не изменился характер звучания инструмента, сэмпл составляется из нескольких фрагментов для разных диапазонов нот.

В сложных синтезаторах используется параллельное проигрывание нескольких сэмплов на одну ноту и дополнительная обработка звука (модуляция, фильтрование, различные «оживляющие» эффекты и т. д.). Инструменты с малой длительностью звучания обычно записываются полностью, а для остальных может записываться лишь начало/конец звука и небольшая «средняя» часть, которая затем проигрывается в цикле в течение нужного времени.

Достоинства данного метода — предельная реалистичность звучания классических инструментов и простота получения звука. Недостатки — наличие жесткого набора заранее подготовленных тембров, многие параметры которых нельзя изменять в реальном времени, а также

большие объемы памяти для хранения сэмплов (иногда до нескольких Гб на инструмент).

Надо заметить, что в большинстве синтезаторов звуковых плат, для которых заявлен метод ВТ-синтеза, на самом деле реализован более старый и простой «сэмплерный» метод. При синтезе звука в этих устройствах атака и затухание реализуется предопределенным сэмплом, длительность звучания которого не может быть изменена, отчего атака и затухание звука звучат всегда с одинаковой длительностью, и только средняя часть может быть произвольной длительности.

Синтез на основе физического моделирования. Это сравнительно новый метод синтеза звука. В отличие от синтеза звука на основе таблицы волн, где в качестве исходных данных выступают оцифрованные звуки реальных музыкальных инструментов, записанные в ПЗУ, физическое моделирование предусматривает использование математических моделей звукообразования реальных музыкальных инструментов для генерации в цифровом виде соответствующих волновых форм, которые затем конвертируются в звук при помощи ЦАП. Алгоритмы моделирования очень сложны и требуют больших вычислений, что делает подобные устройства более дорогими. Однако данный метод синтеза позволяет воспроизводить звуки реальных инструментов наиболее точно.

В последнее время появилось большое число программных синтезаторов, работающих на основе методов физического моделирования звука.

8.4. Устройство звуковой платы

Состав и устройство звуковой платы зависит от конкретной реализации. Так, например, на некоторых звуковых платах отсутствует синтезатор или цифро-аналоговый преобразователь. Однако все устройства звуковой платы можно объединить по функциональному назначению в логические блоки и рассматривать их отдельно. В общем случае можно выделить следующие блоки:

- блок записи/воспроизведения звука;
- блок синтезатора;
- блок внешних интерфейсов;
- блок микшера.

Рассмотрим их подробнее.

Блок записи/воспроизведения звука. Данный блок предназначен для аналого-цифрового и цифро-аналогового преобразований в режиме

программной передачи звуковых данных или передачи их по каналам прямого доступа к памяти. Блок включает в себя следующие компоненты: АЦП; ЦАП; блок управления.

Основные характеристики блока:

- динамический диапазон входных сигналов;
- частота дискретизации;
- тип и разрядность АЦП;
- алгоритм и коэффициент компрессии аудиоданных;
- количество каналов АЦП;
- тип и разрядность ЦАП;
- количество каналов ЦАП;
- возможность одновременной записи и воспроизведения (*симплекс, полудуплекс, дуплекс*).

Блок записи/воспроизведения не связан с блоком синтезатора и управляет собственным драйвером.

Блок синтезатора. Блок предназначен для синтеза произвольных звуковых сигналов, включая голоса музыкальных инструментов. Чаще всего управляется через MIDI-интерфейс. Состав элементов блока определяется способом синтеза, реализуемого звуковой платой. Блок синтезатора при программной реализации MIDI работает под управлением драйвера, а при аппаратной реализации — под управлением собственного процессора.

Тип синтезатора звуковой платы во многом определяет качество звучания. Так, FM-синтезатор воспроизводит звучание музыкальных инструментов не вполне достоверно: инструменты звучат очень бедно, со «звенящим» оттенком, а имитация классических музыкальных инструментов вообще весьма условна. WT-синтезатор воспроизводит инструменты более точно: звучание инструментов «живое», «сочное», классические инструменты звучат естественно, а синтетические — более приятно.

Важной характеристикой синтезатора, независимо от его типа, является *полифония*. *Полифония* (максимальное число голосов синтезатора) — это предельное количество одновременно воспроизводимых синтезатором элементарных звуков (нот). Для плат с FM-синтезатором полифония составляет обычно 20 голосов, для плат с WT-синтезатором — 32 голоса. Однако не следует путать количество полифонических голосов синтезатора с разрядностью АЦП/ЦАП.

Для плат с FM-синтезатором основной характеристикой является число так называемых операторов синтеза. Эта характеристика опреде-

ляет качество имитации «живых инструментов». Например, для микросхемы OPL2 число операторов равно 2, для микросхемы OPL3 число операторов равно 4.

Для плат с предопределенным набором синтезируемых звуков (WT-синтезаторы с ПЗУ) важной характеристикой является набор инструментов. В понятие набор инструментов входит количество инструментов и поддерживаемый стандарт (General MIDI, MIDI XG и т. д.).

Для плат с WT-синтезатором, имеющих ОЗУ, важнейший параметр — это его объем. В конечном итоге именно он определяет количество инструментов синтезатора и качество синтеза. Минимальный объем ОЗУ необходимый для нормальной работы с WT-синтезатором составляет 8 Мб (хотя эта величина зависит от стиля, в котором работает музыкант).

Важным параметром синтезатора является используемая система команд. Практически все FM-синтезаторы совместимы друг с другом, а WT-синтезаторы несовместимы.

Блок внешних интерфейсов обеспечивает обмен данными с физическими устройствами, являющимися внешними по отношению к звуковой плате. Рассмотрим основные интерфейсы звуковых плат.

Системный интерфейс — интерфейс звуковой платы по отношению к материнской плате компьютера (например PCI). Данный интерфейс определяет производительность звуковой системы в целом. Некоторые модели WT-синтезаторов могут хранить сэмплы в обычном ОЗУ компьютера, пользуясь системным интерфейсом.

MIDI-интерфейс (MIDI Processing Unit, MPU) осуществляет обмен MIDI-командами с внешними и/или внутренними MIDI-устройствами. Внешние MIDI-устройства, например, MIDI-клавиатура или тон-генератор, соединяются с компьютером с помощью специального адаптера, подключаемого по разъему Joystick/MIDI, расположенному на монтажной скобе звуковой платы. В описаниях большинства звуковых плат указывается, что ее внешний MIDI-интерфейс совместим с изделием Roland MPU-401. Это означает, что он может работать в режиме приема и передачи команд, совпадающем с режимом DUMB MODE для MPU-401. Обычно MPU, реализующий такой режим, называют MIDI UART.

Интерфейсы S/PDIF (Sony/Philips Digital Interface Format) и *AES/EBU* (Audio Engineers Society/European Broadcast Union) предназначены для последовательной передачи звуковых сигналов между устройствами в цифровой форме. Для кодирования звукового сигнала используется цифровой сигнал с полосой частот от 5 до 10 МГц (в зависи-

симости от частоты дискретизации). Интерфейс S/PDIF используется для передачи 16-разрядных стереофонических данных с любой частотой дискретизации. Для передачи используется коаксиальный кабель с волновым сопротивлением 75 Ом или оптоволоконный кабель. Данный интерфейс в основном предназначен для бытовой радиоаппаратуры. Интерфейс AES/EBU обеспечивает передачу любого монофонического или стереофонического сигнала с переменной частотой дискретизации. Передача осуществляется по протоколу RS-422. Данный интерфейс предназначен в основном для профессионального использования.

Интерфейс для подключения приводов CD-ROM имелся практически на всех звуковых платах, хотя в современных компьютерах надобность в этом интерфейсе отпала.

Блок микшера осуществляет коммутацию и сведение входных и выходных сигналов, а также регулирование их уровней. Основными характеристиками этого блока являются:

- тип и количество входных каналов;
- тип и количество выходных каналов;
- возможность и реализация (механически или программно) управления уровнями каналов;
- возможность внутренней коммутации каналов (например, возможность коммутировать выход синтезатора или ЦАП со входом АЦП).

8.5. Цифровой MIDI-интерфейс музыкальных инструментов

MIDI (Musical Instrument Digital Interface) — это цифровой интерфейс музыкальных инструментов. Создан в 1982 г. ведущими фирмами-производителями музыкального оборудования — Yamaha, Roland, Korg, E-mi и др. Изначально был предназначен для замены принятого в то время стандарта управления с помощью аналоговых сигналов. Впоследствии интерфейс MIDI стал стандартом де-факто в области электронных музыкальных инструментов и компьютерных синтезаторов.

MIDI-интерфейс представляет собой событийно-ориентированный протокол связи между инструментами. Всякий раз, когда исполнитель осуществляет какое-либо воздействие на органы управления музыкального инструмента (нажатие/отпускание клавиш, педалей, изменение положений регуляторов и т. п.), формируется соответствующее MIDI-сообщение, в тот же момент посылаемое через интерфейс. Ин-

струмент, получая сообщения, отрабатывает их так же, как и при воздействии на собственные органы управления. Таким образом, поток MIDI-сообщений является как бы отображением действий исполнителя, сохраняет присущий ему стиль исполнения — динамику, технические приемы и т. п. При записи на устройства хранения информации все MIDI-сообщения снабжаются временными метками и образуют своеобразную «партитуру». При воспроизведении по этим меткам полностью и однозначно восстанавливается исходный MIDI-поток.

Главное назначение MIDI — хранение и передача музыкальной информации в нотной записи. Это может быть управление электронными музыкальными инструментами в реальном времени, запись MIDI-потока, формируемого при игре исполнителя, на носитель данных с последующим редактированием и воспроизведением, синхронизация различной аппаратуры (синтезаторы, ритм-машины, магнитофоны, блоки обработки звука, световое оборудование, дымогенераторы и т. п.).

Существуют устройства, управляемые только через интерфейс MIDI. Наиболее распространенным таким устройством является тон-генератор. *Тон-генератор* — это устройство, предназначеннное для синтеза и управляемое только через MIDI-интерфейс (обычный синтезатор, только без клавиатуры).

Для формирования MIDI-сообщений могут служить устройства, не содержащие средств синтеза звука. Это так называемые MIDI-контроллеры: клавиатура, педаль, рукоятка с несколькими степенями свободы, ударная установка с датчиками способа и силы удара, а также струнный или духовой инструмент с датчиками и анализаторами способов воздействия и приемов игры. Тон-генератор с достаточными возможностями управления может весьма точно воспроизвести оттенки звучания инструмента по сформированному контроллером MIDI-потоку.

Спецификация MIDI состоит из аппаратной спецификации самого интерфейса и спецификации формата данных — описания системы передаваемых сообщений. Соответственно различают аппаратный интерфейс и формат MIDI-данных (MIDI-партитуру). Интерфейс используется для физического соединения источника и приемника сообщений, формат данных — для создания, хранения и передачи MIDI-сообщений. В настоящее время эти понятия стали совершенно самостоятельными: по интерфейсу могут передаваться данные любого формата, а MIDI-формат может применяться только для обработки партитур, без вывода на устройство синтеза.

Аппаратная реализация интерфейса MIDI представляет собой обычный последовательный асинхронный интерфейс типа «токовая петля»

со скоростью передачи данных $31\,250 + 1\%$ бит/с. Интерфейс реализуется активным передатчиком с уровнем 5 мА. Для соединения устройств используется двужильный экранированный кабель длиной не более 15 м, практически нечувствительный к наводкам извне, поэтому экран необходим только для защиты от излучаемых помех.

Один MIDI-передатчик допускает подключение до четырех приемников, так что можно создавать сеть MIDI-устройств, выстраивая их по цепочке и в нескольких направлениях.

MIDI-интерфейс является событийно-ориентированным. Обмен данными осуществляется при помощи блоков данных произвольной длины. Эти блоки данных называют MIDI-сообщениями. Все существующие MIDI-сообщения можно разделить на типы, изображенные на рис. 42.

Канальные сообщения предназначены для передачи команд и параметров отдельным MIDI-устройствам сети в целях управления звучанием музыкального инструмента и определения реакции музыкального инструмента на голосовые сообщения. Это достигается включением в код команды адреса MIDI-устройства или, как его еще называют, MIDI-канала. Такие MIDI-сообщения называются *канальными* (*Channel Message*). Всего в протоколе MIDI существует 16 каналов, т. е. можно адресовать 16 MIDI-устройств. Адресация в MIDI не является однозначной. Несколько инструментов могут быть настроены на один и тот

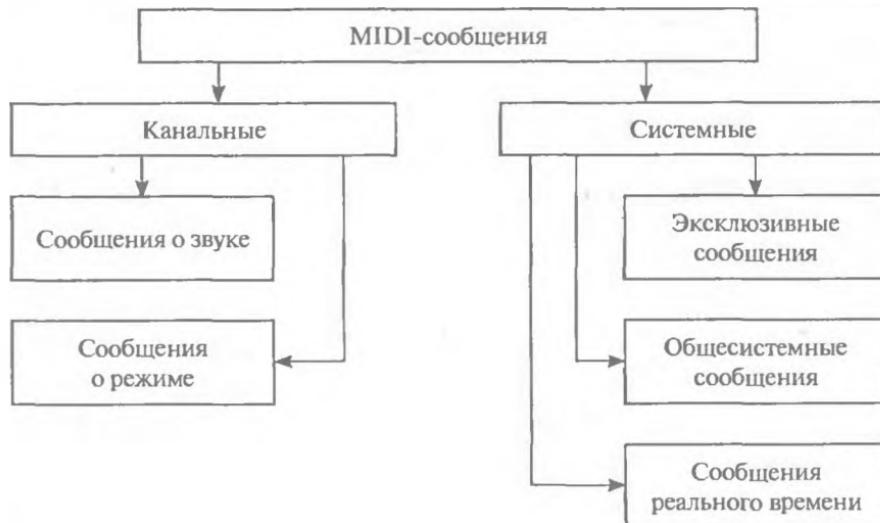


Рис. 42. Типы MIDI-сообщений

же MIDI-канал. В таком случае канальное сообщение может быть послано группе инструментов.

Системные сообщения — это сообщения предназначенные для управления всеми MIDI-устройствами сети. В зависимости от функционального назначения они могут быть следующих типов: *эксклюзивные сообщения, общесистемные сообщения, сообщения реального времени*.

8.6. Мультимедийное резюме

Рассмотрены основные принципы спектрального разложения звука, сопутствующие понятия — интерференция волн, осцилляция гармоник, амплитудный спектр. Упомянутый здесь эффект Гиббса имеет более широкое значение для мультимедийных технологий, в частности, он оказывает значительное негативное влияние на качество JPEG/MPEG-сжатия изображений и видео при высоких степенях компрессии.

Перечислены базовые параметры устройств электроакустики, в том числе амплитудно-частотной характеристики, по которым можно судить о качестве пропускного тракта. Классификация линейных и нелинейных искажений позволяет глубже понять специфику цифровой обработки аудиоинформации.

Многие основные методы синтеза звука реализованы на аппаратном уровне звуковых плат, поэтому понимание основ их функционирования позволяет подобрать оптимальный способ представления звука при создании мультимедийной продукции.

ГЛАВА 9

ЭТАПЫ РАЗРАБОТКИ МУЛЬТИМЕДИА ПРИЛОЖЕНИЙ

Использование мультимедиа в веб-дизайне. Основные этапы разработки проекта. Структура бизнес-плана. Методология MSF. Юридические аспекты защиты продукции.

Создание любого мультимедийного продукта требует тщательного предварительного планирования как содержательной тематики, контента, так и целого ряда организационных моментов. Процесс создания продукта начинается с формулирования идеи проекта или концепции, а затем анализа различных методов отображения необходимой информации, которые смогут наиболее точно отобразить эту концепцию [5].

Сначала создаются план разработки проекта и общая стратегия его реализации. В стратегии необходимо учитывать уровень профессионализма сотрудников, время на разработку, финансовые возможности и риски, техническую, инструментальную базы и ресурсы, которые имеются в распоряжении или потребуются в дальнейшем. Разработка стратегии создания мультимедийного проекта так же важна, как и разработка структуры содержания самого приложения, которое в дальнейшем может быть растиражировано.

Идея проекта должна полностью соответствовать цели создания мультимедийного продукта. Целевые функции проекта должны формулироваться на основе тех программно-аппаратных и экономических возможностей, которые имеются у разработчиков и последующих пользователей.

Часто разработчики мультимедийных приложений, увлекаясь многочисленными возможностями инструментальных средств разработки и перенасыщая проект всевозможными эффектами, искажают саму идею в процессе работы. Поэтому следует помнить главное правило при разработке проекта: лишь взведенное гармоничное участие всех мультимедийных элементов позволяет создать качественный продукт.

9.1. Связь usability и мультимедийных эффектов

В качестве примера перенасыщения проекта разнообразными эффектами и снижения в связи с этим его пользовательских качеств, удобства работы («usability») рассмотрим актуальную проблему применения технологии Flash¹ при создании сайтов, насыщенных мультимедийным контентом.

Снижение уровня интерактивности сайта. Интерактивная сущность, заложенная в фундамент навигации пользователя по сети Интернет, заключается в том, что она позволяет пользователю самому управлять процессом работы. Пользователь самостоятельно передвигается по ссылкам, выбирая когда (и куда) ему нужно переместиться. Flash-презентации, напротив, превращают работу в Интернете в некое подобие просмотра телевизора. Такие сайты заставляют пассивно находиться перед экраном и просматривать последовательность кадров. Пользователь бездействует, начинает скучать, и его уже не впечатляют эти кадры, как бы хорошо они не выглядели.

Использование нестандартного HTML-кода приносит дополнительные неудобства. Если пользователь перемещается внутри Flash-объекта, нажатие кнопки «Back» избавляет от этого объекта полностью, вместо того чтобы вернуть в предыдущее его состояние. Не работает изменение цвета ссылок. Пользователь не видит, какие страницы он уже посещал, а какие нет, теряет чувство ориентации.

Нет возможности сделать текст больше или меньше. Пользователи вынуждены использовать размер шрифта, определенный разработчиком. Для пользователей, не обладающих хорошим зрением, он обычно слишком мал.

Зачастую Flash-заставки ухудшают общий дизайн сайта. «Splash pages» — страницы заставки, обычно содержащие красивую картинку и надпись «Вход» — первые принялись за деградацию дизайна. К счастью, сегодня абсолютное большинство профессиональных сайтов отказались от идеи использования «Splash pages».

¹ Технология Flash позволяет использовать в веб-страницах анимированную векторную графику, объединять и синхронизировать ее со звуковыми потоками и событиями ввода/вывода. Для создания flash-объектов используется программа Macromedia Flash. При публикации автоматически создается три вида файлов Один в формате FLA (Flash Animation), с которым впоследствии можно будет продолжить работу в Macromedia Flash. Другой в формате SWF (ShockWave Format), служащий для проигрывания объекта Flash-плеерами (плагинами), и третий файл — HTML-оболочка для Flash-объекта, в котором автоматически создаются внешние инструкции браузеру, такие как размер и цвет фона.

Использование мультимедийных объектов в веб-дизайне.

Правило 1: не использовать Flash, когда нужно озвучить нажатие кнопки, проиграть маленькую анимацию, сделать то, что можно сделать силами скриптов и языка разметки. Интерфейс браузеров разрабатывался годами, не нужно тратить время посетителя на изучение изысков и «революционных» представлений в области дизайна окон браузеров.

Правило 2: не пытаться экономить на статусной строке и системных кнопках. Если же новому дизайну столь необходимо место этих двух полосок экрана по 20 пикселов каждая, то нужно дополнительно снабдить посетителя всеми потерянными функциями интерфейса браузера.

Правило 3: в интерфейс Flash-объекта обязательно нужно включить кнопку «закрыть окно» и окно с процентом загружаемого файла, желательно в килобайтном эквиваленте.

Правило 4: создавая дизайн интерфейса, необходимо его активные элементы выполнять в цвете, отличном от цвета всех остальных элементов, и цветом же показывать изменение его состояния (нажатие, перетаскивание). В идеале курсор снабжается соответствующими подсказками, возникающими в момент пересечения активных областей. При переходе от блока к блоку сайта нужно сохранять выбранные единожды цветовые схемы.

Правило 5: изменения дизайна курсора, не надо забывать об его основной функции — возможности запустить активный элемент. Не стоит оставлять от курсора мыши один хвост.

Правило 6: создавая полноценный мультимедийный сайт, необходимо иметь в виду, что ролики могут быть просмотрены пользователем в разгар рабочего дня, где «техно-саунд» нежелателен. Лучше предупредить пользователя заранее о звуковых эффектах и дать возможность быстро отключить звук, предусмотрев понятную и легко находимую пиктограмму. При отключении звука нужно дублировать титрами звуковые сообщения на сайте.

Правило 7: предоставлять возможность посетителю пропускать заставки, по нажатию кнопки «пропустить интро».

Правило 8: максимально минимизировать время загрузки модулей сайта. Подгружать элементы интерфейса поблочно, по мере запроса, в уже загруженные оболочки интерфейса сайта.

Правило 9: экономить время посетителя, подгружая большие текстовые файлы извне. Активнее пользоваться системными шрифтами, делая возможным производить распечатку больших фрагментов текста. Не использовать шрифты меньше 8 пунктов, и цвета, неконтрастные с фоном.

Правило 10: ставить на первой странице скрипт, который подгружал бы вариант сайта именно для разрешения монитора посетителя. Не дезориентировать пользователя большими пустыми местами вокруг окон с информацией.

9.2. Этапы разработки мультимедийного проекта

При разработке любого проекта, а тем более мультимедийного, необходимо учитывать принцип оптимального сочетания потребностей и существующих экономических, технических и творческих возможностей того коллектива, который будет его реализовывать. Принято выделять несколько основных этапов разработки проекта мультимедиа.

1. Разработка идеи или концепции. Составление эскизного проекта.
2. Анализ технико-экономического обоснования (ТЭО), бизнес-планирование.
3. Проектирование. Написание сценариев, технических заданий.
4. Создание информационных объектов.
5. Интеграция информационных объектов с помощью инструментальных средств по сценарию.
6. Формирование пользовательского интерфейса.
7. Отладка и тестирование.
8. Выпуск и реализация.
9. Сопровождение продукта, поддержка и переиздание.

Разработка идеи является главным и самым важным этапом. Сначала устанавливается *целевая функция* мультимедийного продукта. Для этого необходимо ответить по крайней мере на следующие вопросы:

- Что является главным в проекте?
- Какова цель проекта, что хотите создать?
- Какие измеримые результаты и в какой срок хотите получить?

На этом этапе в максимальной мере проявляется интуиция создателей продукта, их умение предвидеть технологическое и финансовое продолжение развития идеи.

9.3. Составление бизнес-плана мультимедийной продукции

Бизнес-план определяет всю систему организационно-плановых мероприятий и экономических параметров. В бизнес-плане указываются источники финансирования, просчитывается рентабельность и возможные риски. Обычно бизнес-план на разработку мультимедийных приложений состоит из следующих разделов [42]:

- 1) РЕЗЮМЕ;
- 2) ОПИСАНИЕ ПРОДУКЦИИ;
- 3) ОПИСАНИЕ РЫНКОВ СБЫТА;
- 4) ОЦЕНКА КОНКУРЕНТОВ;
- 5) СТРАТЕГИИ МАРКЕТИНГА;
- 6) ПЛАН ПРОИЗВОДСТВА;
- 7) ОРГАНИЗАЦИОННЫЙ ПЛАН;
- 8) ЮРИДИЧЕСКИЙ ПЛАН;
- 9) ОЦЕНКА РИСКОВ И СТРАХОВАНИЕ;
- 10) ФИНАНСОВЫЙ ПЛАН.

Резюме, как правило, пишется в завершение работы над бизнес-планом, поскольку в нем необходимо ясно и лаконично перечислить, *ЧТО* и *КОГДА* планируется создать, *ЧТО* для этого нужно сделать, и *КАКИЕ* финансовые результаты будут получены. Резюме не должно занимать больше одной страницы.

В разделе **описания продукции** рассматривается сам мультимедийный продукт, его основные уникальные (продающие) моменты. Указывается место продукта на рынке, перечисляются методы сбыта.

При описании **рынков сбыта** основное внимание уделяется типологии и сегментации клиентурных рынков, моделируется портрет типового пользователя будущего продукта, приводятся маркетинговые расчеты емкости целевых сегментов рынка.

В разделе **оценка конкурентов** анализируются технические и ценовые показатели конкурирующих продуктов, их сильные и слабые стороны.

Стратегии маркетинга заключаются в рассмотрении вопросов ценообразования на продукт, определении прогнозируемого спроса на него. Даётся оценка производительности производственно-сбытовых мощностей по реализации продукта.

В **плане производства** перечисляются основные технологические подходы, анализируется программный инструментарий разработки проекта. Указывается стоимость необходимого материально-технического обеспечения.

Организационный план представляет собой перечень мероприятий, необходимых для успешного выполнения проекта. Решаются кадровые вопросы, на участников проекта формируется штатное расписание. Указывается профессиональный уровень разработчиков, их зарплата, необходимые социальные гарантии.

В **юридическом плане** прорабатываются вопросы авторских и имущественных прав, очерчивается договорная база.

Оценка рисков и страхования дается для получения степени возможных финансовых, организационных, политических и валютных рисков и анализа путей их устранения.

В **финансовом плане** сводятся воедино графики платежей и поступлений со всех разделов. Указываются источники финансирования, степень заемных средств, минимальная, средняя и максимальная расчетные выручки. Приводится показатель рентабельности проекта.

Существуют различные типы бизнес-планов, но в подавляющем большинстве такой план создается для поиска инвестиций и пишется под конкретного инвестора или группу инвесторов, таких как высокотехнологичные бизнес-инкубаторы. Но даже если уже имеется положительное решение инвестора, написание бизнес-плана позволит более четко изложить основные характеристики проекта, систематизировать организационные, финансовые и юридические мероприятия. Поэтому считаем, что при проектировании нового мультимедийного продукта бизнес-план необходимо готовить при любых обстоятельствах.

9.4. Существующие методологии разработки проектов

В настоящее время существует достаточное количество методик по созданию программного обеспечения. Мы кратко рассмотрим универсальную методологию Microsoft Solutions Framework (MSF), так как она является достаточно широкой и включает в себя не только описание процесса проектирования приложения, но и модель группы разработки, самого приложения и других смежных вопросов [43]. Кроме того, эта методология (позволяя избежать проблем старых каскадной и спиральной моделей создания программных продуктов) является более традиционной и привычной для разработчиков, чем, например, методика экстремального программирования.

Принципы разработки приложений MSF — это набор моделей, принципов и методов, которые помогают организации более эффективно создавать программное обеспечение, в том числе мультимедийное. Обеспечивая ощутимый прогресс и четкое руководство, MSF позволяет сделать приложение гибким и способным реагировать на изменяющиеся потребности организации. Ядро этой системы составляют пять основных моделей:

- 1) модель проектной группы;
- 2) модель процесса разработки ПО;

- 3) модель управления рисками;
- 4) модель процесса проектирования;
- 5) модель приложения.

Модель проектной группы. Эта модель относится к группе, работающей над проектом, описывает роли, обязанности каждого участника, распределение ответственности и порядок работы. Гибкость позволяет привести модель в соответствие с характером проекта, размером группы и квалификацией участников. Использование этой модели и ее основных принципов помогает сформировать неравнодушную, энергичную и эффективную команду.

Модель процесса разработки. Эта модель описывает организационную структуру процесса разработки и руководство им в течение всего времени выполнения проекта. Отличительные особенности модели — этапность, итеративность и гибкость. Она описывает фазы, этапы, виды деятельности и результаты процесса разработки приложения и их связь с моделью проектной группы MSF. Использование этой модели обеспечивает контроль над ходом разработки проекта, минимизацию рисков, повышение качества и сокращение сроков выполнения проекта.

Модель управления рисками. Эта модель предлагает организованный путь активного управления рисками проекта. Она описывает порядок и условия реализации упреждающих решений и мер для постоянного выявления потенциальных проблем, позволяет обнаружить наиболее существенные риски и реализовать стратегии их устранения. Использование этой модели и ее основных принципов помогает команде сосредоточиться на наиболее важных моментах, принимать верные решения.

Модель процесса проектирования. Эта модель описывает трехфазный, ориентированный на конечного пользователя, непрерывный процесс разработки, характеризующийся параллельным и итерационным выполнением проекта и таким образом способствующий его эффективности и гибкости. Три фазы разработки — концептуальное, логическое и физическое проектирование — реализуют точку зрения на проект трех аудиторий: конечных пользователей, проектной группы и разработчиков. Продвижение от концептуального проекта к физическому превращает набор сценариев использования в совокупность компонентов и сервисов, образующих приложение, реализующее требования заказчика и пользователей. Таким образом, приложение разрабатывается не ради демонстрации технологических возможностей, а для решения насущных проблем пользователей.

Модель приложения. Эта модель реализует логичный, трехуровневый, ориентированный на сервисы метод проектирования и разработки программного обеспечения. Применение пользовательских сервисов, бизнес-сервисов и сервисов данных позволяет реализовать параллельную разработку, обеспечивает оптимальное использование технологии, облегчает эксплуатацию и сопровождение, обеспечивает максимальную гибкость развертывания, поскольку сервисы, составляющие приложение, могут располагаться и на единственном персональном компьютере, и на различных серверах и компьютерах клиентов, установленных в разных странах.

MSF — не готовое решение, а каркас, который можно адаптировать для нужд любой организации. Один из элементов этого каркаса — модель проектной группы. Она описывает структуру группы и принципы, которым надо следовать для успешного выполнения проекта.

Чтобы проект считался удачным, следует решить определенные задачи.

Удовлетворить требования заказчика — проект должен выполнить требования заказчиков и пользователей, иначе ни о каком успехе не может быть и речи, возможна ситуация, когда бюджет и график соблюdenы, но проект провалился, так как не выполнены требования заказчика.

Соблюсти ограничения — разработчики проекта должны уложиться в финансовые и временные рамки.

Выполнить спецификации, основанные на требованиях пользователей. Спецификации представляют собой подробное описание продукта, создаваемое группой для заказчика; соглашение между проектной группой и клиентом; регулирование вопросов, касающихся приложения. В основе этого требования лежит принцип «сделать все, что обещано».

Выпустить продукт только после выявления и устранения всех проблем — не существует программ без дефектов, однако группа должна найти и устраниТЬ их до выпуска продукта в свет, причем устранением ошибки считается не только ее исправление, но и, например, занесение в документацию способа ее обхода; даже такой способ устранения проблем предпочтительнее, чем выпуск приложения, содержащего невыявленные ошибки, которые в любой момент могут преподнести неприятный сюрприз пользователям и разработчикам.

Повысить эффективность труда пользователей — новый продукт должен упрощать работу пользователей и делать ее более эффективной.

Поэтому приложение, обладающее массой возможностей, применять которые сложно или неудобно, считается провальным.

Гарантировать простоту развертывания и управления — эффективность развертывания непосредственно влияет на оценку пользователем качества продукта, например, ошибка в программе установки может создать у пользователей впечатление, что и само приложение небезгрешно. От проектной группы требуется не только подготовить продукт к развертыванию и гладко провести его, но и обеспечить пользователей поддержкой, организовав сопровождение приложения.

Для достижения этих целей в модели проектной группы выполняемые задачи распределяются по шести ролям: менеджмент продукта, менеджмент программы, разработка, тестирование, обучение пользователей и логистика. Люди, выполняющие конкретную роль, должны рассматривать проект со своей «колокольни» и обладать необходимой для этого квалификацией.

9.5. Защита интеллектуальных прав на мультимедиа-продукцию

Необходимо отдельно коснуться проблемы защиты интеллектуальных прав. Это юридически очень сложная и обширная тема, но разработчики мультимедийных проектов должны хотя бы в общем виде знать, как они могут *защитить свой продукт*, как *обезопасить себя от обвинений в нелицензионном использовании чужих материалов* и как *бороться с компьютерным пиратством*.

Следует отметить, что авторские права не распространяются на идеи, методы, процессы, системы, способы, концепции, принципы, открытия, факты. Поэтому такой вид защиты, как патент, в случае программного обеспечения не подходит. Патенты выдаются на «открытия и изобретения, явным образом не следующие из текущего уровня развития науки и техники». Следовательно, программы не патентуются, а две программы, реализующие одну и ту же идею разными способами — это две разные программы. Однако свою программу все же можно защитить, поскольку программа — это не голая идея или метод, а его *реализация*.

Основным действующим законом нашей страны в области результатов интеллектуальной деятельности является Четвертая часть Гражданского Кодекса РФ, сменившая закон РФ «Об авторском праве и смежных правах».

Поэтому вначале определимся, какие именно объекты подлежат защите по закону.

Согласно ст. 1225 ГК РФ результатами интеллектуальной деятельности являются:

- произведения науки, литературы и искусства (в том числе аудиовизуальные произведения);
- программы для электронных вычислительных машин;
- базы данных;
- фонограммы;
- сообщение в эфир или по кабелю радио- или телепередач;
- изобретения;
- секреты производства (ноу-хай);
- фирменные наименования;
- товарные знаки и знаки обслуживания;
- коммерческие обозначения.... .

Не являются объектами авторского права:

- официальные документы законодательного, административного и судебного характера;
- государственные символы и знаки;
- произведения народного творчества;
- сообщения о событиях и фактах, имеющие информационный характер [44].

Соответственно, при разработке собственных мультимедийных проектов нужно руководствоваться простым правилом — если в проекте используются заимствованные произведения, подпадающие под действие ст. 1225 ГК РФ, необходимо разыскать правообладателей и письменно согласовать с ними этот вопрос.

Интеллектуальные права подразделяются на:

- 1) исключительное право (*имущественное*) — владеть, пользоваться, распоряжаться;
- 2) личное неимущественное (*авторское*) и иные права (право доступа и т. п.).

На практике, создатели программ часто путают эти права между собой. Поэтому рассмотрим их подробнее.

Авторское право возникает в силу *факта его создания*. По ст. 1261, авторские права на все виды программ для ЭВМ охраняются так же, как авторские права на произведения литературы. *Программой для ЭВМ* является представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определенного результата, включая подготовительные материалы, полученные в ходе разработки

программы, и порождаемые ею аудиовизуальные отображения. Т.е. мультимедийная продукция также защищена авторским правом.

При отсутствии доказательств иного автором произведения считается лицо, указанное в качестве автора на оригинале или экземпляре произведения. Для указания исключительных авторских прав авторы вправе снабдить свое произведение знаком охраны авторского права, состоящим из трех элементов (хотя отсутствие копирайта на произведении совсем не означает отсутствие авторских прав):

- латинской буквы С в окружности: ©;
- имени (наименования) обладателя исключительных авторских прав;
- года первого опубликования произведения.

Со своей стороны мы рекомендуем обратить внимание на ст. 1262 ГК РФ о государственной регистрации программ для ЭВМ: «Правообладатель в течение срока действия исключительного права на программу для ЭВМ или на базу данных может по своему желанию зарегистрировать такую программу или такую базу данных в федеральном органе исполнительной власти по интеллектуальной собственности».

Таким образом, можно дополнительно защитить свой мультимедийный проект путем депонирования одного экземпляра (например, листинга программы) в соответствующем *fедеральном органе исполнительной власти по интеллектуальной собственности*, оплатив пошлину. Сам факт депонирования экземпляра не является фактом признания авторских прав, но фиксирует суть передаваемого на хранение произведения и дату размещения. Такое действие может существенно помочь при установлении в суде факта первенства разработки и истинного автора проекта.

Авторские права действуют пожизненно (и даже в течение 50–70 лет после смерти). Но имущественными правами на произведение автор может спокойно распоряжаться по своему усмотрению. Можно воспроизводить произведение в любой форме, показывать публично, продавать частично или полностью и т. п. В момент полной продажи произведения по договору возникает момент перехода исключительных имущественных прав на произведение к покупателю. После этого «допускается дальнейшее распространение без согласия автора и без выплаты авторского вознаграждения».

Кому принадлежат имущественные права на гениальную программу, которую между делом написал на работе программист — ему или работодателю? Для правильного ответа на этот вопрос необходимо внимательно изучить трудовой договор и должностные обязанности этого программиста. Если во вмененные ему обязанности это входит (возмож-

но, общими формулировками, например «создание инструментальных программных средств»), то — работодателю (ст. 1295 о служебном произведении). Если же ни в каких официальных документах это не упоминается, то тогда имущественные права принадлежат программисту, хотя в этом случае он справедливо может заработать выговор, штраф или даже увольнение за занятие личными делами в рабочее время.

Здесь возникает другой актуальный вопрос, тесно связанный с проблемой пиратства: какие правоустанавливающие документы являются документальным подтверждением имущественных прав юридического лица на продаваемый им мультимедийный продукт? Чаще всего в качестве таких документов могут встречаться трудовые договоры (фирма наняла программистов под конкретный проект), договоры подряда (фирма заключила договор с другой фирмой на разработку продукта), договоры купли-продажи (фирма «А» купила продукт у фирмы «Б», принадлежавший последней на праве собственности).

Если программа была зарегистрирована ранее, то договоры об отчуждении исключительного права на зарегистрированную программу и переход исключительного права на такую программу к другим лицам без договора подлежат государственной регистрации в федеральном органе по интеллектуальной собственности.

О пользовании программами, в том числе инструментарием разработчика. В лицензионных соглашениях с конечными пользователями компания-правообладатель предоставляет «неисключительное имущественное право на одну копию программного обеспечения». Причем в этом же лицензионном соглашении компания должна четко оговорить все возможные действия с программой или результатами ее работы, которые вправе совершить пользователь. В частности, пользователь имеет право самостоятельно устранять ошибки в программе, вносить в нее любые изменения, не связанные с устранением программно-аппаратной защиты (если она, конечно, реализована), сделать с программы резервную копию, и даже ее декомпилировать. Если же пользователь не согласен с условиями лицензионного соглашения, то он должен вернуть программу производителю, не вскрывая упаковки с электронным носителем, на котором записана эта программа.

9.6. Мультимедийное резюме

При создании мультимедийной продукции необходимо уделять критичное внимание процессу управления проектом разработки. Этот процесс включает в себя план разработки проекта; стратегию реализа-

ции; основные цели, которые преследует проект; бизнес-план. При формулировании целей необходимо заботиться, чтобы цели были реалистичными, подкрепленными ресурсами, измеримыми и ограниченными по времени (так называемый критерий SMART).

Бизнес-план является одним из важнейших первоначальных документов, создаваемых до начала интенсивной реализации проекта. Помимо функции поиска инвестора, бизнес-план содержит общее организационное описание предполагаемых действий и ожидаемой отдачи от проекта.

Из различных проектных методологий в данной главе кратко рассмотрена модель Microsoft Solutions Framework, позволяющая управлять всем жизненным циклом проекта.

При разработке мультимедийной продукции особое внимание разработчиков должно быть удалено вопросу интеллектуальных прав на используемые в проекте чужие программы и материалы и проблеме официальной авторской защиты результатов собственного труда.

БИБЛИОГРАФИЯ

1. Крапивенко А. В. Технологии мультимедиа и восприятие ощущений: учеб. пособие. М.: БИНОМ. Лаборатория знаний, 2009.
2. Балаховская Т. Три значения мультимедиа // КомпьютерПресс. 1995. № 2.
3. Сторчак В. В. Знакомьтесь — multimedia. М.: Компьютеры + программы, 1993.
4. Крапивенко А. В. Кросс-кадровая интерполяция видеоданных: дис. ... канд. физ.-мат. наук. М.: МАИ, 2001.
5. Алексеева М. Б., Балан С. Н. Технология использования систем мультимедиа: учеб. пособие. СПб.: Изд. дом «Бизнес-пресса», 2002.
6. ClearType. <http://research.microsoft.com/~jplatt/cleartype>.
7. Троицкий Д. И. Автоматизация конструкторского труда. <http://home.uic.tula.ru/~ams/WIN/BOOK/MAN/EU/LECTURE2/lecture2.htm>.
8. Крапивенко А. В., Сулеев И. М. Генерация ландшафтов с использованием воксельной графики // САПР. 1999. № 8.
9. Пореев В. Н. Компьютерная графика. СПб.: БХВ-Петербург, 2002.
10. Зефиров С. Некоторые результаты размышлений об использовании вокселяй. <http://www.codenet.ru/progr/video/voxel>.
11. Шикин Е. В., Боресков А. В. Компьютерная графика: динамика, реалистические изображения. М.: Наука, 1990.
12. Piegl L. A. On NURBS: A Survey // IEEE Computer Graphics and Applications. 1991. Vol. 11. № 1. P. 55–71.
13. NURBS-моделирование. Вып. 2. <http://grdragon.narod.ru/subscribe/NURBS/2.htm>.
14. Виды кривых и поверхностей, используемых в CAD/CAM системах. <http://www.polymir.narod.ru/Articles/curves.html>.
15. Altman M. Nonuniform Rational B-Splines — NURBS. <http://www.cs.wpi.edu/~matt/courses/cs563/talks/nurbs.html>.
16. State of the Art in Computer Graphics // Visualization and Modeling / eds. D. F. Rogers, R. A. Earnshaw. N.Y.: Springer-Verlag, 1991. P. 225–269
17. Кудря В. Н. Исследование и разработка алгоритмов синтеза моделей геометрических объектов. http://masters.donntu.edu.ua/2003/fvti/kudrya/diss/index.htm#5_1.
18. Голованов Н. Н. Геометрическое моделирование. М.: Изд-во физико-математической литературы, 2002.
19. Шикин Е. В., Боресков А. В. Компьютерная графика. Полигональные модели. М.: Диалог-МИФИ, 1995.
20. Gee W. Возможности 3D Studio MAX для Windows NT // Computer Week. 1996. № 41.

21. Роджерс Д. Алгоритмические основы машинной графики: пер. с англ. М.: Мир, 1989.
22. Иванов В. П., Батраков А. С. Трехмерная компьютерная графика / под ред. Г. М. Полищук. М.: Радио и связь, 1995.
23. Миланина М. VRML в примерах. www.firststeps.ru/www/vrml/vrml1.html.
24. Аврамова О. Д. Язык VRML: практическое руководство. М.: Диалог-МИФИ, 2000.
25. Suler J., The Psychology of Avatars and Graphical Space in Multimedia Chat Communities. Rider University — May 1996. <http://www1.rider.edu/~suler/psycyber/psycyber.html>.
26. Тиммел Э., Сандерс К., Скотт Ч., Вольф П. Создание VRML-миров. Киев: BHV, 1997.
27. The Virtual Reality Modeling Language Specification. Version 2.0., 1996. <http://vrml.sgi.com/moving-worlds/index.html>.
28. Елисеев Г. А. Моделирование виртуального торгового комплекса на языке VRML. <http://www.elias.ru>.
29. Пайтген Х.-О., Рихтер П. Х. Красота фракталов. М.: Мир, 1993.
30. Ватолин Д. Применение фракталов в машинной графике // Computer World. 1995. № 15.
31. Гребеник В. Усовершенствованный алгоритм сжатия изображений на основе ИФС // Тр. конф. GRAPHICON 1998.
32. Кроновер Р. М. Фракталы и хаос в динамических системах. М.: Постмаркет, 1999.
33. Хакен Г. Синергетика: Иерархии неустойчивостей в самоорганизующихся системах. М.: Мир, 1985.
34. Пригожин И., Стенгерс И. Порядок из хаоса: Новый диалог человека с природой. М.: Прогресс, 1986.
35. Тодоренко О. Радуга оптических форматов // КомпьюТerra. 2000. 15 августа.
36. Рош У. Библия Мультимедиа. Киев: Диасофт, 1998.
37. Мюррей Дж., ван Райпер У. Энциклопедия форматов графических файлов. Киев: BHV, 1997.
38. DVD: диски, проигрыватели, рекордеры. <http://intercomp.net.ru/storage/dvd.htm>.
39. IEEE 1394 — новая последовательная шина. www.ixbt.ru.
40. Шина USB. http://www.spline.ru/SplineRazdel_4.html.
41. Леонов С. Абвгдейка // КомпьюТerra. 2002. 22 марта.
42. Крапивенко А. В. Бизнес-план мультимедийного проекта. <http://www.elias.ru>.
43. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD. М.: Изд-во торговый дом «Русская редакция», 2002.
44. Авторское право в шоу-бизнесе: Нормативные акты и комментарии / авт.-сост. И. А. Силонов — М.: Агентство «Издательский сервис»; Норма, 2001.

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	3
ГЛАВА 1. ИНСТРУМЕНТЫ МУЛЬТИМЕДИА. МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ СРЕД	4
1.1. Место мультимедиа в компьютерных технологиях	4
1.2. Виды инструментальных средств мультимедиа	5
1.3. Моделирование трехмерных сред	9
1.4. Представление 3D-моделей	10
1.5. Воксельный способ	11
1.6. Полигональный (треугольный) способ	13
1.7. Функционально-параметрический способ	14
1.8. Мультимедийное резюме	15
ГЛАВА 2. ФУНКЦИОНАЛЬНО-ПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ. СПЛАЙН-ФУНКЦИИ	16
2.1. Интерполяционные сплайны	16
2.2. Сплайновые кривые. Кривая Безье	20
2.3. В-сплайновые кривые	24
2.4. Рациональные кубические В-сплайны	26
2.5. Области применения NURBS	26
2.6. NURBS-кривые. Определения	28
2.7. Мультимедийное резюме	31
ГЛАВА 3. ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНЫХ СЦЕН. ПРОСТАЯ МОДЕЛЬ ОСВЕЩЕНИЯ	32
3.1. Визуализация геометрической модели. Этапы подготовки	32
3.2. Моделирование текстуры	33
3.3. Атмосферные эффекты, источники освещения	35
3.4. Рендеринг, определение нормали к поверхности	36
3.5. Определение вектора отражения	37
3.6. Прозрачность и тени	38
3.7. Простая модель освещения	40
3.8. Мультимедийное резюме	44
ГЛАВА 4. МЕТОДЫ РЕНДЕРИНГА 3D-СЦЕН	45
4.1. Закраска методом Гуро	45
4.2. Закраска Фонга	46

4.3. Глобальная модель освещения с трассировкой лучей	47
4.4. Распределенная трассировка лучей	51
4.5. Методы оптимизации трассировки лучей	52
4.6. Метод излучательности (Radiocity)	54
4.7. Мультимедийное резюме	57
ГЛАВА 5. ПРОГРАММНЫЕ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ	58
5.1. Панорамные видеотехнологии	59
5.2. Язык VRML, аватары	60
5.3. Структура VRML. Создание и воспроизведение сцен	62
5.4. Основные понятия языка	64
5.5. Координатная система. Примитивы VRML, цвет и текстура	66
5.6. Положение объектов в пространстве. Гиперссылки. Узел Anchor	67
5.7. Сенсоры окружающей среды. Узлы-манипуляторы	69
5.8. Мультимедийное резюме	70
ГЛАВА 6. ИСПОЛЬЗОВАНИЕ ФРАКТАЛОВ В МУЛЬТИМЕДИЙНЫХ ТЕХНОЛОГИЯХ	71
6.1. Геометрические фракталы. Самоподобие	72
6.2. Дробная размерность снежинки Коха	74
6.3. L-системы. Тетр-графика	75
6.4. Алгебраические фракталы	77
6.5. Множество Мандельброта и множества Жюлиа	78
6.6. Стохастические фракталы	81
6.7. Хаос и синергетика	82
6.8. Мультимедийное резюме	84
ГЛАВА 7. ТЕХНОЛОГИИ ХРАНЕНИЯ И ПЕРЕДАЧИ МУЛЬТИМЕДИЙНЫХ ДАННЫХ	85
7.1. Эволюция оптических дисков	86
7.2. Стандарты CD-ROM	88
7.3. Устройство DVD	92
7.4. Диски Blu-ray	95
7.5. Накопители на флэш-памяти	96
7.6. Интерфейсы передачи мультимедийных данных	97
7.7. Мультимедийное резюме	102
ГЛАВА 8. ЦИФРОВЫЕ ТЕХНОЛОГИИ ЗАПИСИ И ВОСПРОИЗВЕДЕНИЯ ЗВУКА	103
8.1. Спектральное представление звука	103
8.2. Параметры устройств электроакустики	106

8.3. Основные методы синтеза звука	110
8.4. Устройство звуковой платы	115
8.5. Цифровой MIDI-интерфейс музыкальных инструментов	118
8.6. Мультимедийное резюме	121
ГЛАВА 9. ЭТАПЫ РАЗРАБОТКИ	
МУЛЬТИМЕДИА ПРИЛОЖЕНИЙ	122
9.1. Связь usability и мультимедийных эффектов	123
9.2. Этапы разработки мультимедийного проекта	125
9.3. Составление бизнес-плана мультимедийной продукции	125
9.4. Существующие методологии разработки проектов	127
9.5. Защита интеллектуальных прав на мультимедиа-продукцию	130
9.6. Мультимедийное резюме	133
БИБЛИОГРАФИЯ	135

Учебное издание
Крапивенко Андрей Викторович
МЕТОДЫ И СРЕДСТВА ОБРАБОТКИ
АУДИО- И ВИДЕОДАННЫХ

Учебное пособие

Книга издана в авторской редакции

Ответственный редактор *М. Б. Гаража*
Технический редактор *П. С. Корсунская*
Корректор *А. П. Гундаев*
Компьютерная верстка *К. И. Бобрусь*

Подписано в печать 05.05.10. Формат 60×84/16.
Печать офсетная. Бумага офсетная. Гарнитура «Ньютон».
Усл. печ. л. 8,13. Тираж 300 экз.

ЗАО «Издательское предприятие «Вузовская книга»
125993, Москва, А-80, ГСП-3, Волоколамское шоссе, д. 4,
МАИ, Главный административный корпус, к. 301а.
Тел./факс (499) 158-02-35. E-mail: vbook@mail.ru; vbook@mai.ru
www.vuzkniga.ru

