

Анимация



- Записанная анимация и последовательности изображений
- Цифровой кель и спрайтовая анимация
- Анимация ключевых кадров
- Web-анимация и Flash
 - Временная шкала и сцена
 - Символы и построение промежуточных изображений
- Графика движения
- Трехмерная анимация
- Виртуальная реальность
 - Язык моделирования виртуальной реальности
 - Виртуальная реальность QuickTime

Анимацию можно определить как создание движущихся изображений по одному кадру за раз. Этот термин также используется для обозначения последовательности, полученной описанным способом, например “диснеевская анимация” и “Web-анимация”. В XX столетии анимация использовалась в индустрии развлечений, рекламе, обучении, искусстве, пропаганде, в фильмах, а позже — в видео; в настоящее время она широко распространена в WWW и мультимедийных презентациях.

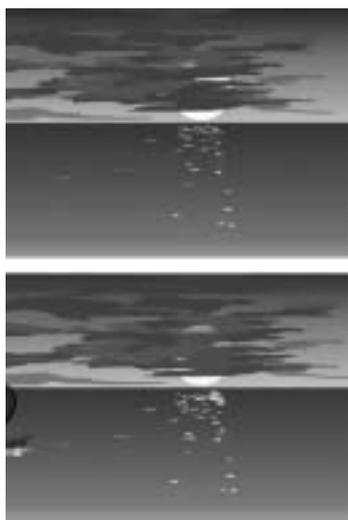
Чтобы понять, как действует анимация, создадим на бумаге последовательность рисунков, отличающихся элементами, которые должны меняться или двигаться. Изменение между соседними рисунками может быть и практически незаметным, и бросающим-

ся в глаза. После создания рисунков их последовательность фотографируется в правильном порядке по одному снимку за раз. При воспроизведении фильма данная последовательность неподвижных изображений будет восприниматься так же, как последовательность кадров, заснятых в реальном времени для “живого” действия: из-за инертности зрительного восприятия ряд неподвижных изображений воспринимается, как непрерывное движущееся изображение. Если нужно создать иллюзию быстрого движения или изменения, разница между соседними изображениями последовательности должна быть значительно большей, чем в том случае, когда изменение должно быть постепенным, а изменения плавными.

“Анимировать” означать “оживлять”, и именно это является сутью процесса: при воспроизведении фильма или видео со стандартной скоростью неподвижные персонажи, объекты, абстрактные формы и т.п., сфотографированные последовательно, кажутся ожившими.

Поскольку фильм демонстрируется со скоростью 24 кадра в секунду, рисование анимации в традиционной среде (как было описано выше) технически требует 24 изображений на каждую секунду фильма, т.е. 1 440 картинок на каждую минуту, а для анимации, сделанной для видео, даже больше. На практике анимацию, которая не требует бессмысленно гладкого движения, можно снимать “с удвоенной скоростью”, т.е. делается не один снимок каждой картинке, а два. В результате получается, что эффективная частота смены кадров равна 12 кадров в секунду для фильма, 15 — для видео NTSC. Действительно, цифровую анимацию можно воспроизводить при таких малых частотах смены кадров, что позволяет существенно сократить объем работ.

Если анимация создается исключительно по рисункам на бумаге, каждый аспект изображения нужно повторять на всех рисунках. Чтобы сократить огромный объем работы, которую необходимо выполнить в этом процессе, было разработано множество других методов анимации (часть из них развилась в ходе постоянного поиска новых выразительных средств). Наиболее известным и самым распространенным из них (по крайней мере, до недавнего времени) был метод, при котором использовалась *келевая анимация* (рис. 8.1). Его суть заключается в том, что элементы сцены, которые могут двигаться (например, Гомер Симпсон), изображаются на прозрачном материале, именуемом келем, и накладываются на изображенный отдельно фон (возможно, гостиную Симпсонов).



При производстве последовательности изображений для каждого кадра нужно перерисовывать только движущиеся элементы на келе; фиксированные части сцены изображаются только один раз. Кели можно накладывать друг на друга, меняя различные кели на разных кадрах, чтобы получить более сложную и динамичную сцену. Расширяя этот подход, фон можно нарисовать на длинном листе, распространяющемся далеко за пределы одного кадра, и перемещать его за келями при переходе между кадрами, чтобы создать эффект перемещения по сцене. Концепции и технологии традиционной келевой анимации оказались особенно полезными при передаче цифровой реальности, и многие популярные келевые телевизионные мультфильмы в настоящее время производятся цифровым способом.

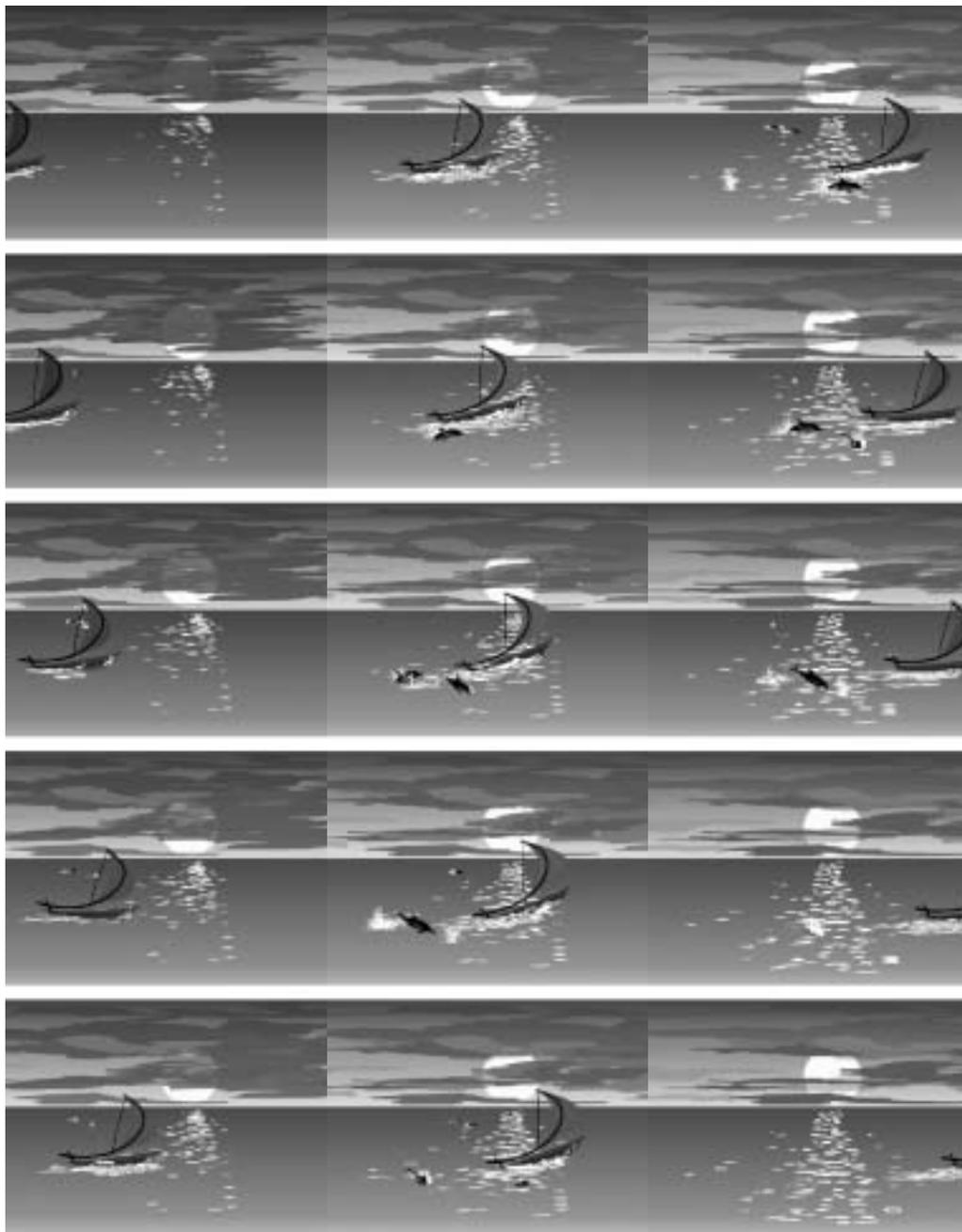


Рис. 8.1. Келевая цифровая анимационная последовательность

В значительной мере из-за огромного влияния студии Уолта Диснея, в которой келевая анимация развилась, в частности, до использования установок с несколькими плоскостями, создающих иллюзию трехмерной сцены, данная технология стала использо-

ваться практически везде, и в мультипликационных сериалах (от “Моряка Папая” до “Симпсонов”), и в полномасштабных фильмах, начиная с “Белоснежки и семи гномов” (1937 г.). В то же время, начиная с первых шагов движущихся изображений в 1890-х годах, анимация успешно создавалась с использованием множества иных средств. Одни художники действительно рисовали каждый кадр отдельно на бумаге, другие рисовали сразу на пленке или соскребали эмульсию с черненной неэкспонированной кинопленки, третьи работали с песком или масляными красками на стекле или мелом на бумаге, для создания нового кадра внося изменения в созданное изображение. Существовала также группа людей, которые работали с аппликациями перед камерой; известным примером аппликативной анимации является известная работа Терри Джиллиама (Terry Gilliam) в сериале “Монти Пайтон”. Иногда аниматоры открывали для себя абсолютно новый способ работы — здесь можно вспомнить булавочный экран Алексея и Паркера, на котором близко расположенные кнопки избирательно вдавливались через доску и освещались так, что отбрасываемые ими на изображение тени формировали изображение, которое менялось для съемки нового кадра.

Совершенно иной альтернативой описанных, по сути двухмерных, форм является трехмерная анимация, или покадровая съемка. В это понятие объединено несколько технологий, использующих миниатюрные трехмерные наборы (подобные наборам сцены), в которых объекты аккуратно перемещаются между кадрами. В качестве объектов могут использоваться шарнирные фигуры, конечности которых можно перемещать, или неподвижные фигуры, части которых заменяются (или подставляются) между кадрами, чтобы создать эффект телодвижений, ходьбы и т.д. Вместо этого можно использовать фигуры и другие объекты, сделанные из податливых материалов (например, пластилина); в этом случае фигуры можно вручную изменять между кадрами, получая естественное движение, а также изменения и преобразования, которые невозможно создать другими средствами. Последняя разновидность анимации (часто называемая *глиняной анимацией*) получила признание после появления работ студии Аардмана (Aardman studio), в частности — фильма “Wallace and Gromit”.

Хотя различные технологии анимации иногда удобно рассматривать по отдельности, часто встречаются гибридные формы анимации, например комбинация келевой и трехмерной анимации. Долгую историю также имеет традиция объединения анимации и живого материала. Пожалуй, наиболее известным примером такого смешения жанров является фильм “Кто подставил Кролика Роджера?” (1988 г.). Однако живое действие и анимация объединялись и ранее, в том числе в фильмах Джорджа Мелиеса (Georges Méliès) и серии Макса Флейшера (Max Fleischer) “Out of the Inkwell” (1920-е гг.), которая внесла значительный вклад в популяризацию анимации как формы развлечения. В последнее время интенсивное внедрение цифровых технологий в кинопроизводство привело к существенному росту интереса к объединению анимации с живым действием, в частности в фильмах со спецэффектами. Аудитория не всегда понимает, что большая часть того, что они считают спецэффектами, получена с помощью обычных технологий анимации, как традиционных (классические фильмы с монстрами, например “Кинг Конг”, 1933 г.), так и цифровых (например, “Матрица” и ее продолжения).

Все традиционные формы анимации имеют аналоги в цифровой реальности. Более того, цифровые технологии предлагают новые возможности для использования в новом контексте анимации и технологий, развившихся из нее.

Записанная анимация и последовательности изображений

Как будет показано далее, цифровая технология проложила новые пути для создания анимации, но, если использовать компьютеры вместе с другими методами, рассмотренными выше, можно эффективно получать анимацию в цифровой форме, подходящую для внедрения в мультимедийную продукцию. В настоящее время подобная подготовка анимации — используя цифровые технологии вместе с видеокамерой и традиционными методами анимации — предлагает аниматору, работающему в цифровой среде, более богатые возможности, чем чисто компьютерные методы, описанные ниже в этой главе.

Вместо того чтобы записывать анимацию на кино- или видеопленке, видеокамера подключается непосредственно к компьютеру, и каждый кадр анимации записывается на диск — все равно, изображен он бумаге или келе, построен с помощью трехмерной установки или создан с использованием другой технологии, которая не зависит от технологии производства фильма. Вместо того чтобы записывать весь поток данных, поступающих с камеры, как при записи живого видео, в каждый момент съемки записывается цифровая версия единственного кадра. Для выполнения подобного *захвата кадров* (frame grabbing) можно использовать одну из множества маленьких утилит. Кроме того, это позволяют делать некоторые программы для редактирования видео; в частности, Premiere имеет команду Stop Frame в меню Capture. Механизмы захвата кадра работают приблизительно по одному принципу: в окне записи показывается, что в данный момент “видит” камера. С помощью этого изображения можно проконтролировать снимок, а затем нажать клавишу для захвата одного кадра, либо помещаемого в файл неподвижного изображения, либо добавляемого в видеоряд AVI или QuickTime. Затем меняется рисунок, положение моделей и т.п. и снимается следующий кадр. Неудовлетворительные кадры можно удалять; обычно программы позволяют при работе с текущим рисунком видеть “призрачное” изображение предыдущего захваченного кадра, что помогает выровнять кадры и внести требуемые изменения. После захвата набора кадров, формирующих последовательность, его можно записать как ролик QuickTime или набор

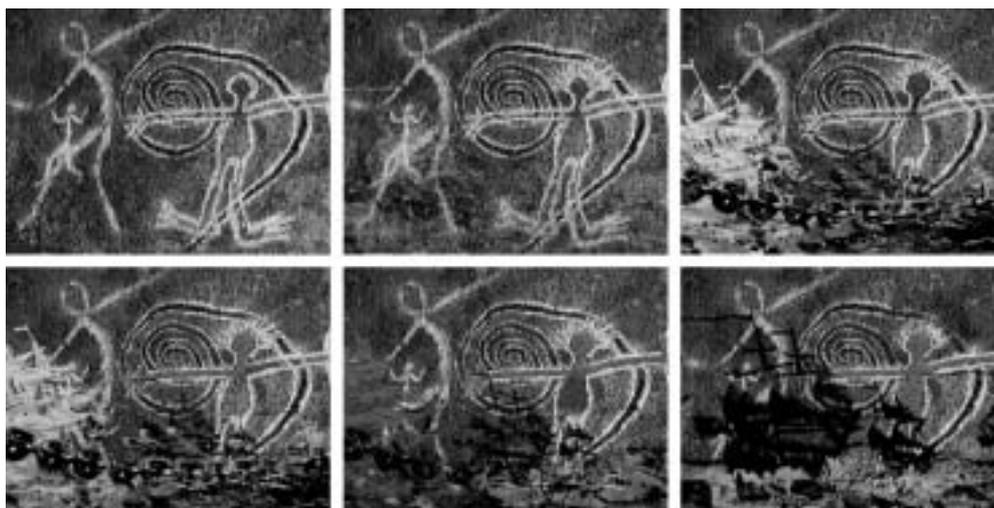


Рис. 8.2. Анимированные стоп-кадры записываются на диск и объединяются с цифровой анимацией

последовательно пронумерованных файлов изображения (см. ниже). Последняя возможность полезна, если, например, отдельные изображения планируется “доводить” в Photoshop или импортировать в Flash.

Запись анимации на диск, произведенная описанным способом, не только открывает возможности нелинейного редактирования и компоновки, описанные в главе 7, но и разрешает комбинировать анимацию в традиционных средах с чисто цифровой анимацией и движущейся графикой. На рис. 8.2 приведен пример работы, созданной именно в такой смешанной манере.

Для определенных типов традиционной анимации даже использовать камеру необязательно. Если вы создали ряд рисунков на бумаге, с помощью сканера из них можно получить набор файлов изображений. Кроме того, манипулировать аппликациями на планшете сканера почти так же просто, как и под прицелом камеры. Мало того, сканер позволяет оцифровать анимацию прямо с неэкспонированной киноплёнки. Вместо видеокамеры можно использовать цифровой фотоаппарат, при условии, что он позволяет загружать изображения прямо на диск. Во всех описанных случаях можно работать при большем разрешении и более богатой цветовой гамме, чем позволяет видеокамера.

Для рисованной анимации можно обойтись без внешней формы и всего процесса оцифровки, используя графическую программу для рисования и записывая сделанную работу, как ролик или последовательность файлов изображений. С помощью кистей Painter или последних версий Photoshop можно создать анимацию, отчасти выглядящую так, как будто она создана с помощью традиционных материалов. Можно воспользоваться и средствами для манипуляции пикселями этих программ и создать работу с характерным цифровым внешним видом. Даже если вы создаете анимацию по кадру за раз, программа часто позволяет сэкономить время — записать макрос или сценарий для перерисовки повторяющихся элементов или для применения фильтров к нескольким кадрам.

Последовательности файлов изображений являются очень гибким представлением анимации. Отдельные файлы можно открывать в графической программе и корректировать; файлы можно удалять из последовательности, заменять или добавлять. Набор изображений можно импортировать в программу редактирования видео и преобразовать в ролик AVI или QuickTime. В то же время управление коллекцией файлов изображений может быть сложным, особенно если со временем их предполагается импортировать в программу редактирования видео. Чтобы не оказалось, что выбирать можно лишь из определенного набора приложений, имена файлов должны соответствовать некоторым договоренностям. Например, на компьютерах Macintosh программа Premiere может импортировать только последовательность файлов PICT, причем все они должны находиться в одной папке и иметь суффикс, состоящий из точки, за которой следует число определенной разрядности, например Animation.001, Animation.002, ..., Animation.449. (Легко догадаться, какими будут последствия, если забыть о необходимых дополняющих нулях.) Если вы внесли изменения в набор изображений, нужно проследить за тем, чтобы нумерация не изменилась, или откорректировать ее.

Некоторые компьютерные программы, включая Painter и Flash (более подробно они рассмотрены ниже), позволяют открывать ролик и модифицировать отдельные его кадры. (Photoshop может открывать файлы в специальном диакопическом формате, экспортировать в который или импортировать из которого можно с помощью Premiere.) Это предоставляет новые возможности. Например, можно рисовать (или каким-либо иным образом изменять) на исходном видеоматериале, что является одним из способов добав-

ления анимации в кадры с естественным движением¹. Другой вариант — отследить на уровне кадр за кадром выбранные элементы “живого” видеоклипа, который затем удалить. Данный процесс (реализуется “в цифре” или иным образом) правильно называется *ротоскопированием*; он долго использовался для создания анимации, точно воспроизводящей формы и естественное движение людей и животных.

Ротоскопирование получило свое название от *ротоскопа*, прибора, запатентованного в 1915 году Максом Флейшером (Max Fleisher) — создателем Бетти Буп и моряка Папая. Устройство Флейшера проектировало материал клипа по кадру за такт на стол с подсветкой, таким образом создавая спроектированное неподвижное изображение, на которое аниматор мог наложить лист с анимированным изображением. После завершения калькирования одного кадра начиналась обработка следующего, для чего пленка вручную прикреплялась к следующему кадру.

Вместо того чтобы применять для записи анимационного ряда набор неподвижных изображений, иногда можно в одном файле “изображения” записать несколько изображений. Хотя данную возможность предлагает множество файловых форматов (в том числе форматы, предназначенные для использования с программами по созданию анимации), чаще всего для этой цели используется формат GIF.

Способность файлов GIF записывать последовательность изображений применялась для получения дешевой и “живой” формы анимации для Web-страниц. Большинство Web-браузеров последовательно отображают все изображения, которые содержатся в GIF-файле, загружая файл. Если отображение происходит достаточно быстро, набор рисунков будет выглядеть, как анимация. Версия GIF89a данного формата предлагает несколько необязательных элементов данных, управляющих поведением этого анимационного формата GIF. В частности, можно установить метку (flag), включающую циклическое воспроизведение анимации либо заданное число раз, либо бесконечно, а также задать минимальную задержку между кадрами, а следовательно — частоту смены кадров. Впрочем, *анимированные GIF-изображения* не совсем надежны как средство добавления анимации на Web-страницы. Пользователи могут изменять способ отображения анимированных GIF-картинок на экране (как и многие другие аспекты поведения браузеров) — циклическое воспроизведение можно отключить, анимацию можно заблокировать; кроме того, если отключить загрузку изображений, анимированные GIF-картинки вообще не будут отображаться (хотя соответствующим образом данная возможность реализована не во всех браузерах). Основным преимуществом анимированных GIF-изображений является то, что они не зависят от сменных модулей (plug-in) или использования сценариев (см. главу 16), так что их можно просматривать в множестве Web-браузеров.

Существует несколько бесплатных или недорогих утилит для основных платформ, позволяющих комбинировать наборы изображений в один анимированный GIF-образ. Premiere и Flash также предоставляют возможность записывать в подобной форме ролики, а с помощью таких специализированных программ графики для Web, как ImageReady и Fireworks, можно создавать анимированные GIF-картинки по наброскам или готовым изображениям. Следовательно, теоретически GIF-файлы можно использовать для хранения любой разновидности анимации. В то же время, даже если анимация GIF соответствующим образом реализована и активизирована, она имеет множество недос-

¹ В сфере компьютерной графики данный процесс рисования на существующих видеокадрах иногда называется ротоскопированием (rotoscoping), но, как объясняется ниже, этот термин неправильный.

татков. В файл невозможно добавить звук, в палитре имеется всего 256 цветов, изображения сжаты с потерями, что может сказаться на их качестве, при этом не дав существенного выигрыша в размере файла (данное соображение весьма серьезно, когда рассматривается использование данного формата для длительных анимационных последовательностей). Обычно кадры анимированного GIF-образа отображаются браузером по мере поступления. Ограничения, накладываемые скоростью сети, означают, что между кадрами возможны нерегулярные (а иногда и чрезмерные) задержки, а значит, указывать в файле частоту смены кадров бессмысленно. В то же время, если анимированная последовательность воспроизводится циклично, после первого воспроизведения она будет целиком скопирована в локальный кэш браузера (если она не слишком велика), и далее скорость воспроизведения будет ограничиваться только возможностями процессора и диска пользователя (которые абсолютно неизвестны аниматору). Вообще, если последовательность не очень короткая, шанс, что анимированная GIF-последовательность будет должным образом воспроизведена при достаточно высокой частоте смены кадров, дающей гладкую анимацию, весьма мал. Следовательно, обычно анимированные GIF-образы используются не для создания реалистичной анимации, а для более стилизованных меняющихся изображений, например таких, которые часто применяются на рекламных щитах. Возможно, по этим причинам GIF-файлы получили широкое распространение в рекламных блоках на Web-сайтах. Пожалуй, справедливым будет утверждение, что из-за простоты, с которой каждый может добавить на Web-страницу анимированную рекламу, в ней используется чуть ли не наихудшая из возможных анимаций. (Возможно, одним из следствий этого стало падение популярности анимированных GIF-изображений в последние годы.)

При записи анимации любой длительности, особенно если она сопровождается звуком, наилучшие результаты дает использование видеоформата. После того как анимированная последовательность нарисована или записана, например, в формате QuickTime (или преобразована в данный формат), получается обычный ролик QuickTime, так что его можно редактировать, объединять с другими клипами, накладывать спецэффекты, добавлять на Web-страницы, в общем, работать с ним, как с другими видеоклипами. В то же время анимационные клипы могут иметь совершенно иные, своеобразные особенности, которые влияют на способ работы с этими файлами. В частности, определенные стили рисованной анимации тяготеют к использованию упрощенных форм и одноцветных областей. (Существуют и исключения из этого правила; характеристики изображений зависят от личного стиля аниматора.) Материал подобного типа может больше подходить для сжатия без потерь, чем другие типы видео. Кодек Animation (QuickTime) разработан специально для того, чтобы использовать преимущества простой рисованной мультипликационной анимации, которая, как будет показано в данной главе, часто объединяется с компьютерной трехмерной анимацией. Сжатие основано на групповом кодировании (Run-Length Encoding — RLE) и при использовании кодека с настройками максимального качества является сжатием без потерь. Существует также режим сжатия с потерями, с помощью которого достигается большее сжатие. Поскольку схема основана на групповом кодировании, данный кодек может хорошо сжимать одноцветные области и хорошо подходит для анимированных последовательностей указанных выше типов.

Цифровой кель и спрайтовая анимация

Обсуждавшаяся выше келевая анимация могла ассоциироваться у вас с уровнями, описанными в главе 3. Уровни позволяют создавать отдельные части неподвижного изображения, например человека и сцену, на фоне которой он перемещается, так что оба образа

можно изменять или перемещать независимо. Кадры анимированной поверхности можно создавать, объединяя уровень фона (статичный) с одним или несколькими уровнями анимации, на которых и происходят все изменения. Следовательно, чтобы создать анимацию, вначале формируется уровень фона для первого кадра, а затем на отдельных уровнях создаются элементы, которые будут двигаться. Если нужно создать иллюзию глубины, между этими уровнями с движением можно ввести несколько промежуточных статичных уровней. Записав первый кадр, мы переносим уровень фона на второй и накладываем на него другие уровни, сделав на них изменения, запланированные для анимации. Таким образом, на каждом кадре не нужно повторно создавать статичные элементы или использовать для этого сценарии.

Если анимируется простое движение, может потребоваться переместить или преобразовать изображения лишь нескольких уровней. Рассмотрим простой пример. Предположим, что нужно анимировать движение планеты на фоне звезд. Первый кадр может состоять из фонового уровня, содержащего звездное поле, и уровня переднего плана с изображением планеты. Чтобы создать следующий кадр, эти два уровня копируются и с помощью инструмента движения изображение планеты смещается на небольшое расстояние. Действуя подобным образом, можно сформировать последовательность, в которой планета движется на заданном фоне. (Если не требуется, чтобы планета двигалась по прямой линии, изображение придется не только смещать, но и поворачивать.)

Используя уровни в качестве цифрового эквивалента келей, можно сэкономить время аниматора, но в том виде описанная выше техника не меняет способ записи полной анимации: каждый кадр записывается, как файл изображения, а последовательность позднее преобразуется в ролик QuickTime, анимированную GIF-картинку или другое удобное представление. Очевидно, что последовательность, кадры которой построены из одного набора элементов, записывается с существенной избыточностью. Возможно, если конечная последовательность будет сжата, от избыточной информации придется избавиться, но сжатие готовой последовательности вряд ли будет настолько же успешным, как запись последовательности в форме, в которой уже учтена избыточность. В общих чертах это означает однократную запись всех статичных уровней и всех объектов (т.е. непрозрачных частей) на других уровнях плюс описание принципа преобразования движущихся элементов при переходе между кадрами.

Данная форма анимации, основанной на движущихся объектах, называется *спрайтовой анимацией*, а объекты именуются *спрайтами*. Чтобы получить более сложное движение, с каждым спрайтом можно соотнести набор изображений, иногда называемых *гранями* (face). Обычно этого достаточно, например, для создания “цикла ходьбы” человекоподобного персонажа (рис. 8.3). Последовательно изменяя положение спрайта и циклически переставляя грани, можно заставить персонаж идти.

QuickTime поддерживает траектории спрайтов, когда анимация записывается в форме “выборки ключевых кадров” с последующими “корректирующими выборками”. Выборка ключевого кадра содержит изображения всех граней спрайтов, используемых в данной анимации, и значения пространственных свойств (положение, ориентация, видимость и т.д.) каждого спрайта; кроме того, указывается, какая грань отображается в данный момент. Корректирующие выборки не содержат информации об изображении, в них записываются только новые значения свойств спрайтов, которые изменились на данный момент. Следовательно, эти выборки могут быть очень маленькими. Отметим, что спрайтовые траектории QuickTime можно объединять в ролики с обычным видео и звуковыми дорожками.



Рис. 8.3. Спрайтовые грани цикла ходьбы

Мы описали спрайтовую анимацию как способ записи анимационной последовательности, но она также часто используется по-другому. Вместо того чтобы записывать изменения свойств спрайтов, изменяющиеся значения можно генерировать динамически с помощью программы. Таким образом, простые движущиеся последовательности, которые можно описать алгоритмически, записываются еще компактнее. Более того, вычисление свойств спрайтов можно сделать зависимым от таких внешних событий, как движение мыши и другой пользовательский ввод. Иначе говоря, движение и внешний вид анимированных объектов может контролировать пользователь. Такой способ использования спрайтов широко применяется в компьютерных играх, но с его помощью можно создавать и динамическую форму взаимодействия в других контекстах, например в тренажерах.

Анимация ключевых кадров

На протяжении 1930- и 1940-х годов ведущие американские производители мультфильмов, возглавляемые Уолтом Диснеем, разработали принцип массового производства анимации. Основным моментом этой разработки было разделение труда. Как конвейер Генри Форда при производстве автомобилей стал возможен благодаря разбиению сложных заданий на маленькие повторяющиеся подзадания, которые можно было выполнить относительно неквалифицированными рабочими, так и идея Диснея заключалась в разбиении процесса создания ряда изображений на подзадачи, по крайней мере часть из которых мог выполнять относительно неквалифицированный персонал. От снижения требований к квалификации персонала Дисней в анимации выиграл меньше, чем Форд на производстве: разработка персонажей, придумывание идей и сюжетных линий, проверки и некоторая анимация всегда требовали опытных и талантливых художников. Однако, когда дело доходило до производства конечных келлей фильма, действия талантливых аниматоров сводились к созданию *ключевых кадров*.

Данный термин уже встречался в контексте сжатия видео, а также в связи с траекториями спрайтов QuickTime. Тогда ключевыми назывались кадры, которые записывались целиком, а для промежуточных кадров записывались только разности. В традиционной анимации значение этого термина имеет несколько другой оттенок: ключевые кадры обычно создаются «ведущим аниматором», и на них вырисовываются поза и детальные характеристики персонажей² в важных точках анимации. Обычно на ключевых кадрах изображают крайние точки движения (начало и конец прогулки, верхнюю и нижнюю точку падения и т.д.), которые относительно полно определяют, что будет происходить на промежуточных кадрах, хотя ключевые кадры могут создаваться и для любого момента, в который происходят существенные изменения. После этого помощники аниматора практически механически могут рисовать промежуточные кадры. С каждым ведущим аниматором могут работать несколько таких помощников, повышающих его производительность. (Кроме того, трудоемкую задачу переноса рисунков на кель и их раскрашивания также переложили на плечи подчиненных.)

Построение промежуточных кадров можно описать математическим термином *интерполяция*: расчет значений функции, проходящей между известными точками. Интерполяция — это одна из тех задач, которые весьма удачно решаются с помощью компьютерных программ, так что значения, которые нужно вычислить, и связи между ними можно выразить численно. Обычно связь между двумя ключевыми кадрами «рукописной» анимации слишком сложна, чтобы ее можно было привести к численному виду,

² Массовое производство в анимации ассоциируется практически исключительно с мультипликационными персонажами.

подходящему для компьютерной обработки. Однако это не запрещает людям пытаться решить данную задачу хотя бы из-за существенной возможной экономии труда.

Все цифровые изображения в некотором смысле представлены численно, но цифровое представление векторных изображений гораздо проще представления растровых образов, что делает их существенно привлекательнее с точки зрения численной интерполяции. Говоря точнее, преобразования, которые можно применить к векторным формам (трансляция, поворот, масштабирование, отражение и сдвиг), являются арифметическими операциями, которые поддаются интерполяции. Следовательно, движение, состоящее из комбинации этих операций, можно сгенерировать в процессе численного построения промежуточных кадров, начав с пары данных ключевых кадров. Сказанное означает, что мультипликацию можно создавать цифровыми средствами в таких программах, как Flash, при значительном сокращении работы по сравнению с традиционными методами.

Если рассматривать только прямолинейное движение, простейшей формой интерполяции является *линейная*. Это означает, что объект проходит равные расстояния между любой парой соседних кадров, а общее расстояние, пройденное в кадре, равно общему расстоянию между начальным и конечным ключевыми кадрами, деленному на количество кадров в последовательности. Проще говоря, символ движется с постоянной скоростью, что порождает две проблемы.

Во-первых, движение начинается и заканчивается мгновенно, объекты выходят на полную скорость, только начав движение, и поддерживают ее до момента остановки. В действительности никакое тело так не движется. Для получения более естественного движения программы, в которых реализована интерполяция движения (в том числе Flash), используют технику, перенесенную из “рукотворной” анимации: переход от неподвижного положения к движению делается постепенным с помощью маленьких, плавно увеличивающихся приростов положения на нескольких первых кадрах (т.е. объект *ускоряется* до конечной скорости). Обратным процессом является *замедление* движения. На рис. 8.4 показано, как со временем меняются горизонтальное смещение и скорость объекта при его перемещении от исходной точки в ключевом кадре 1 (kf_1) с координатами (0, 0) к конечной точке в ключевом кадре 2 (kf_2) с координатами (50, 50) при использовании линейной интерполяции по 50 кадрам. На рис. 8.5 и 8.6 показано, как можно модифицировать изменения при ускорении или замедлении движения при использовании *квадратичной* интерполяции, т.е. при постоянном ускорении. Возможны и более сложные стили, которые могут быть предпочтительными в различных ситуациях. При использовании ускорения и замедления в Flash аниматор может устанавливать степень ускорения с помощью ползунка, положение которого изменяется от максимального ускорения через движение с постоянной скоростью до максимального торможения. По сути, при этом кривая смещения изменяется от формы, подобной показанной на рис. 8.5, к кривым с менее явным изгибом, переходя в прямую линию (рис. 8.4), а затем изгибаясь в обратную сторону (рис. 8.6), т.е. ускорение изменяется от некоторого максимального положительного значения через нуль до максимального отрицательного значения.

Во-вторых, рассмотрим рис. 8.7, на котором показано, как изменяются смещение и скорость, если добавить к исходной последовательности вторую, состоящую из 50 кадров, на которых объект перемещается из точки kf_2 с координатами (50, 50) в точку kf_3 с координатами (75, 75). Поскольку обе последовательности интерполируются отдельно как прямые линии, в точке kf_2 возникает резкая неоднородность; из графика скорости четко видно, что в этой точке анимации наблюдается как бы внезапное замедление. Такое движение является неестественным, а следовательно, нежелательным. Аккуратно настраивая ползунки ускорения, данную неоднородность можно сгладить, однако существует более общее решение этой проблемы. В главе 4 мы сформулировали самое привлекательное

свойство кривых Безье: их можно соединять гладко, совмещая касательные векторы. Следовательно, используя для интерполяции ключевых кадров вместо прямых линий кривые Безье, можно добиться гладкого движения. Обратите внимание: не объекты должны следовать по траекториям, имеющим форму кривых Безье, а скорость, с которой изменяются свойства, нужно интерполировать, используя кривую Безье. Flash не позволяет применять нелинейную интерполяцию, но существуют другие, более сложные системы анимации по ключевым кадрам, которые предлагают такую возможность. Отметим, что, хотя выше мы рассматривали только смещение в одном направлении, нужно понимать, что подобные соображения применимы ко всем преобразованиям, поддающимся интерполяции.

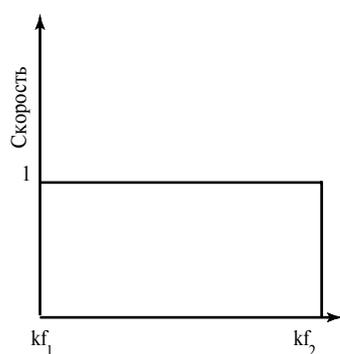
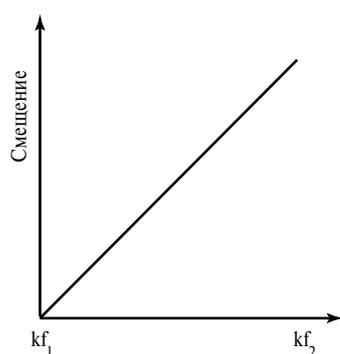


Рис. 8.4. Линейная интерполяция движения

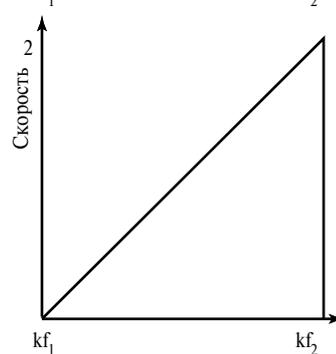
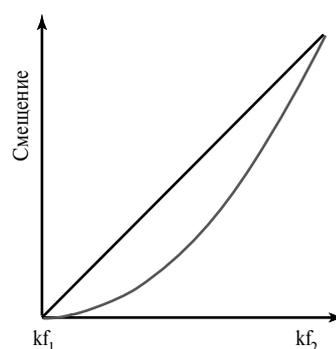


Рис. 8.5. Ускорение

Web-анимация и Flash

Анимацию можно добавлять на Web-страницы в форме анимированных GIF-файлов или встроенного видео, но наиболее популярным форматом Web-анимации является формат *Shockwave Flash* (SWF), который обычно генерируется с использованием Macromedia Flash. SWF — это векторный формат анимации, что делает его особенно удачным в сфере Web-анимации, поскольку графические объекты можно компактно представить в векторной форме и, как подчеркивалось в предыдущем разделе, движение можно представить численными операциями с векторными данными. Следовательно, анимация в формате SWF может иметь более низкие требования к полосе пропускания, чем видео или любой битовый формат. Платой за уменьшенную полосу является то, что векторная анимация не дает всех тех возможностей, которые предлагает битовое отображение.

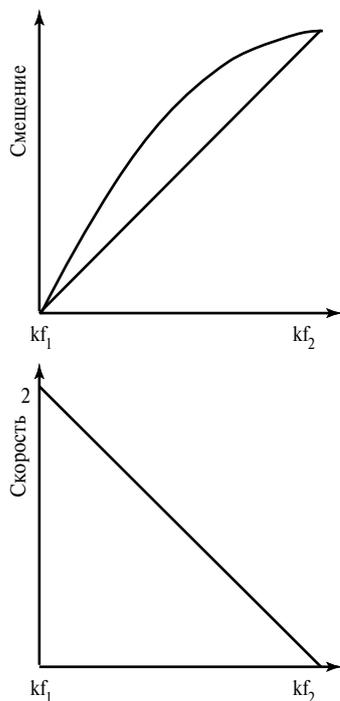


Рис. 8.6. Замедление

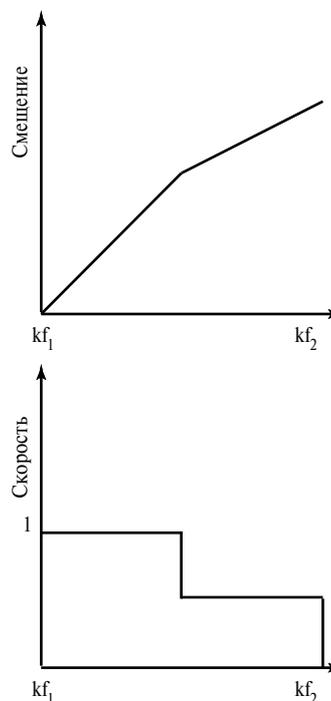


Рис. 8.7. Резкое изменение скорости

Хотя файлы SWF *могут* быть маленькими, требовать только небольшой ширины полосы, это не всегда так. Flash позволяет импортировать битовые изображения в анимацию, а определенные стили рисования и анимации дают векторную анимацию существенного размера. Фактически загрузка многих анимаций Flash, распространенных в Web, требует неприемлемо длительного времени при доступе к Internet через модем. Если вопрос ширины полосы для вас не важен, стоит попытаться не использовать битовые отображения, рисовать простые векторные формы и максимально применять в своей анимации интерполированное движение.

Flash — это больше чем программа анимации. Она поддерживает мощный язык подготовки сценариев, названный ActionScript, который позволяет интерактивно взаимодействовать с анимацией и строить Web-приложения с пользовательскими интерфейсами, созданными в Flash. К данным сторонам Flash мы еще обратимся в главе 16. Сейчас же отметим, что с помощью сценариев можно создавать анимацию в Flash, используя для задания положений символов клипа специальные алгоритмы. Вместо формирования движения вручную можно указать, что объекты движутся согласно законам физики или некоторым математическим алгоритмам, например, определяющим полет стаи птиц как группы.

Временная шкала и сцена

Для организации анимации, создаваемой в Flash, используется *временная шкала* (timeline) — графическое представление последовательности кадров, подобная времен-

ной шкале в приложениях для редактирования видео. Анимацию можно формировать из отдельных кадров, последовательно располагая ключевые кадры на временной шкале.

Сцена (stage) Flash — это подокно, в котором кадры формируются путем рисования объектов. Объекты на сцене можно создавать, применяя некоторые встроенные средства рисования, подобные соответствующим инструментам Illustrator и Freehand (хотя и проще их), или импортировать из других приложений. Растровые изображения в форматах JPEG и PNG также можно импортировать и использовать для создания векторных объектов. В кадре Flash объекты можно использовать в растровой форме, но в таком случае их нельзя поворачивать или масштабировать, не исказив. Flash комплексно поддерживает текст; символы эскизных шрифтов можно разбивать на компоненты, которые можно по отдельности редактировать или анимировать. Для организации элементов кадра можно использовать уровни; также они играют ведущую роль в интерполяции движения. На рис. 8.8 показаны сцена и временная шкала в Flash. Интерфейс Flash также содержит инструментальную панель со средствами рисования, набор палитр для смешивания и выравнивания цветов, применения преобразований, задания параметров печати (набора текста) и т.д.

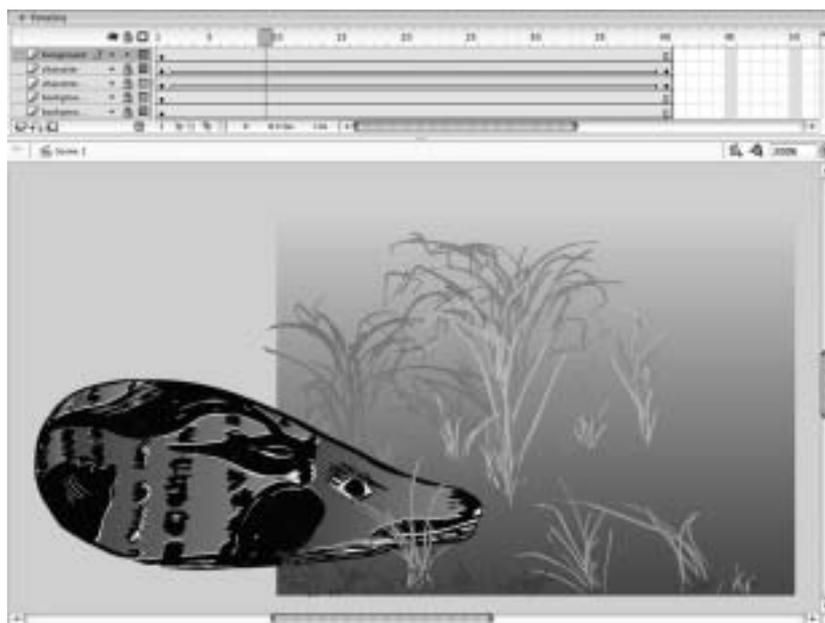


Рис. 8.8. Временная шкала (вверху) и сцена (внизу) простого ролика Flash

Когда ролик только начинается создаваться, он содержит единственный пустой ключевой кадр. Когда на временную шкалу непосредственно после существующего ключевого кадра добавляется новый ключевой кадр, работа начинается с копирования содержимого предыдущего ключевого кадра. Поскольку для большинства анимаций характерны незначительные изменения между соседними кадрами, эффективный способ работы с анимациями, создаваемыми кадр за кадром, заключается в последовательном добавлении ключевых кадров в конец существующей последовательности и внесении изменений в их содержимое. Для облегчения работы с анимацией подобного типа Flash предлагает средство кальки; при активизации этой возможности под текущим кадром отображается до

пяти полупрозрачных предыдущих кадров. Это позволяет легко отслеживать изменения между кадрами и правильно выравнивать объекты.

Помимо ключевых кадров, на временной шкале могут располагаться обычные. Они не содержат собственных объектов и при воспроизведении ролика продолжают отображать содержимое ближайшего предшествующего ключевого кадра, т.е. базируются на ключевых кадрах. Кадры и ключевые кадры можно независимо добавлять на разные уровни, поэтому один уровень может содержать статичное фоновое изображение, а движущиеся объекты могут располагаться на вышестоящем уровне. Уровень фона будет иметь только один ключевой кадр в самом начале ролика, а на уровнях с движением ключевые кадры будут вставлены во всех точках, в которых объект движется (иногда это могут быть все кадры).

Символы и построение промежуточных изображений

Графические объекты можно записывать в библиотеке в специальной форме так называемых *символов*, что позволяет повторно использовать эти объекты. На сцене можно размещать множественные *экземпляры* символа. Все они могут быть по сути идентичными, но также их можно преобразовывать, изменяя размер и ориентацию различных экземпляров. Следует помнить, что экземпляры связаны с символами; если изменить символ, автоматически изменятся и все его экземпляры.

Поскольку при анимации через интерполяцию практически по определению объекты используются повторно, интерполяция (или построение промежуточных изображений; *tweening*, как это называется в Flash) движения объекта превращает его в символ. Чтобы формировать движение, используя промежуточные изображения, можно применять несколько подходов. Простейший вариант — выбирать ключевой кадр на временной шкале и рисовать объект на сцене. Для этого из меню *Insert* выбирается команда *Create Motion Tween*; в результате формируются промежуточные изображения, и (как побочный эффект) объект записывается в библиотеке как символ. В конце последовательности промежуточных изображений создается другой ключевой кадр, и символ переходит в положение, которое он имеет в этом новом кадре. Построение промежуточных изображений представляется на временной шкале как стрелка между двумя ключевыми кадрами (рис. 8.8). При этом для двух уровней символов строятся промежуточные изображения, в результате чего мы наблюдаем рыбу, проплывающую на экране по прямой линии (рис. 8.9; здесь показан только каждый четвертый кадр анимации). Анимация может создаваться как последовательность сегментов, для которых между упорядоченными вручную ключевыми кадрами автоматически строятся промежуточные изображения.

Перемещение одного символа на сцене по прямой линии не позволяет художнику полностью раскрыть свой талант. Построение промежуточных изображений можно применять к различным уровням, размещая ключевые кадры в разных местах и при этом допуская независимую анимацию большого количества символов, каждый из которых является частью некоторого объекта. Чтобы еще больше упростить анимацию движения, объект можно пустить по траектории, нарисованной на невидимом уровне. Данная *траектория движения* необязательно должна быть прямой линией, так что движение, для отображения которого при разрешенном только “прямоугольном движении” понадобилось бы построить множество ключевых кадров, с помощью промежуточных изображений создается на основе всего двух ключевых кадров. Наконец, хотя описанный процесс называется построением промежуточных изображений для движения, таким же образом можно интерполировать изменение размера, ориентации, прозрачности и цвета объекта.

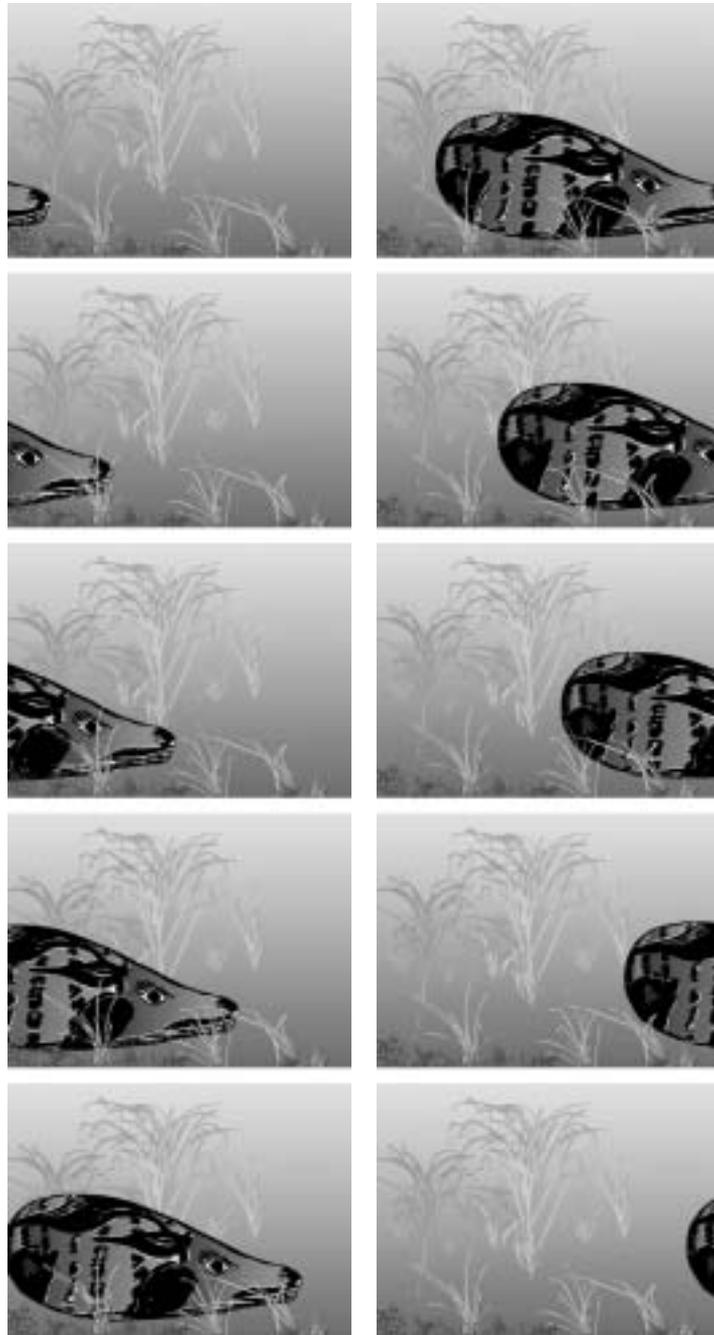


Рис. 8.9. Простая анимация, созданная путем построения промежуточных изображений

Помимо построения промежуточных изображений для движения, Flash поддерживает построение промежуточных изображений для формы, или трансформацию (morphing), как это обычно называется. Трансформация является разновидностью интерполяции,

когда формы графических объектов преобразуются между ключевыми кадрами, например превращение квадрата в круг. В анимационной последовательности, приведенной в начале данной главы, а также на рис. 8.10–8.12, движение брызг и ряби на море создано именно с помощью трансформации.

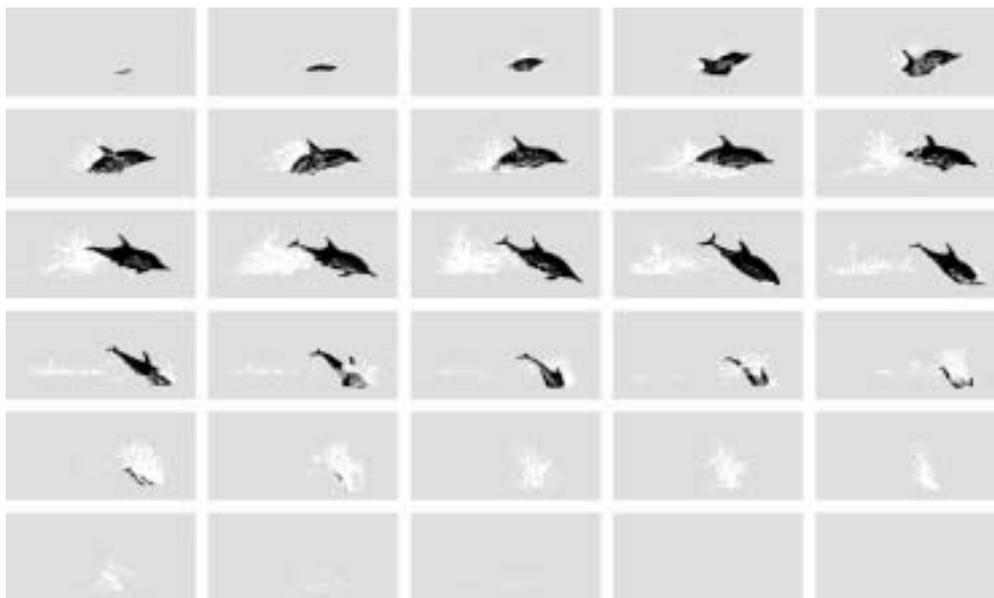


Рис. 8.10. Символ клипа в форме прыгающего дельфина (для большей отчетливости наложен на серый фон)



Рис. 8.11. Символы клипа, синхронизированные с другими действиями в основном клипе

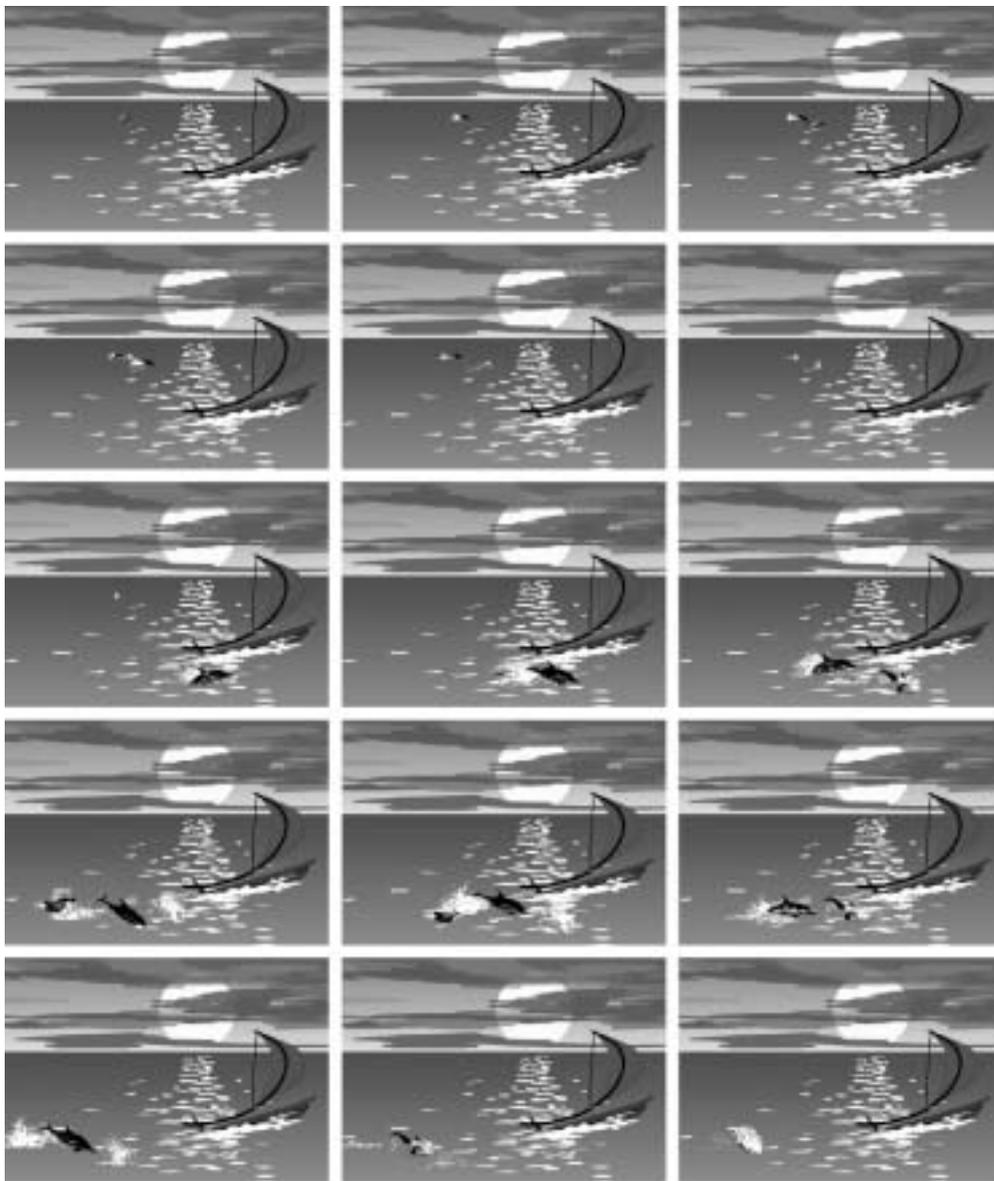


Рис. 8.12. Символы клипа, воспроизводимые независимо от основного клипа

Фактически в Flash существует три основных вида символов. *Графические* символы представляют собой просто векторные объекты с возможностью повторного использования. Символы, создаваемые при построении промежуточных изображений для движения по умолчанию, являются графическими символами. Специальным типом символов являются *кнопочные* символы, используемые для интерактивного взаимодействия с роликами Flash (подробно рассмотрены в главе 16). *Символы клипа* — это независимые анимации с собственными временными шкалами, которые воспроизводятся на фоне основного клипа. Например, резвящиеся дельфины, живописно выпрыгивающие из воды в

анимации кораблика, были добавлены в ролик, изначально содержащий только корабль, море и облака. Для этого потребовалось создать символ клипа — прыгающего дельфина (см. рис. 8.10); затем несколько экземпляров этого символа было добавлено в основной клип, и так появилась группа дельфинов, сопровождающих кораблик.

Поскольку экземпляры символа клипа имеют собственные временные шкалы, они могут воспроизводиться даже после окончания основного клипа. На рис. 8.12 показано, что группа дельфинов продолжает прыгать даже после прекращения движения корабля (пользователь остановил воспроизведение ролика). Сравнивая эти кадры с рисунками в начале главы, видим, как можно создавать сложную анимацию, используя независимое движение ее элементов. На рис. 8.13 показано, как общая композиция создается из множества уровней с применением промежуточных изображений для формы и движения, а также клипов. Обратите внимание на то, что изображения на сцене выходят за пределы кадра; при построении промежуточные изображения войдут в кадры через некоторое время.

Природа векторного рисования и построения промежуточных изображений позволяет создавать компактные анимации. Файл SWF состоит из *элементов* (item), которые разделены на две большие категории: *определяющие* и *управляющие элементы*. Первые используются для записи определений символов, применяемых в анимации, в словарь; последние — это инструкции по размещению, удалению или движению символа (определяемого именем в словаре). Размещение и движение задаются с помощью матриц преобразования, так что положение и разнообразное масштабирование и/или вращение задаются одновременно. Таким образом, файл SWF чем-то похож на программу и включает определения некоторых объектов, а также инструкции, управляющие ими. Данные SWF кодируются в бинарной форме и сжимаются, что дает весьма маленькие файлы.

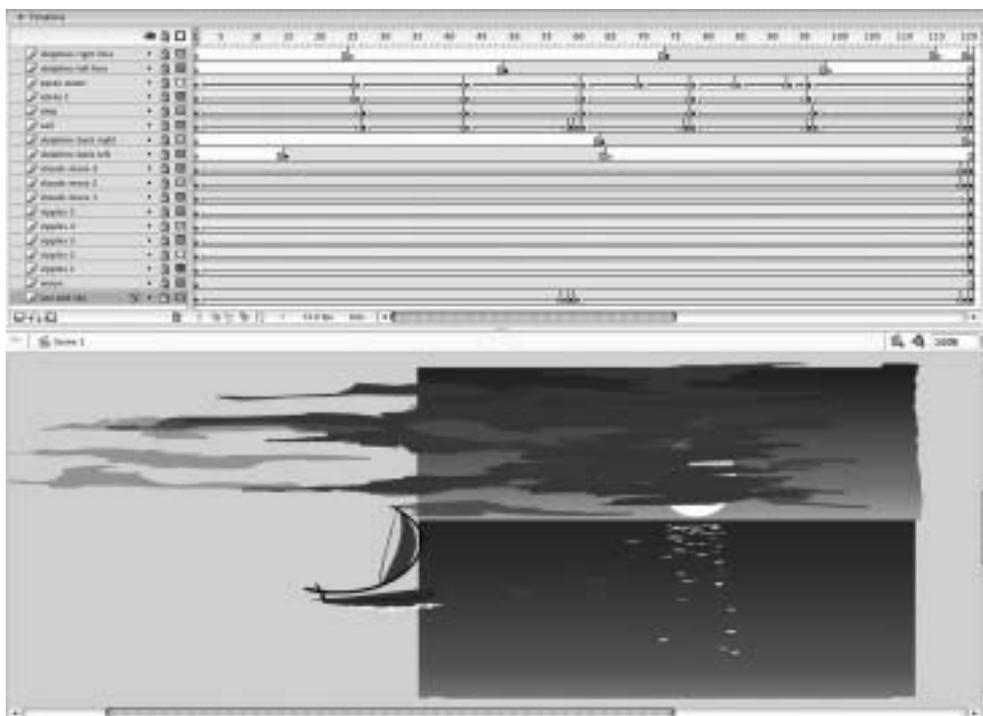
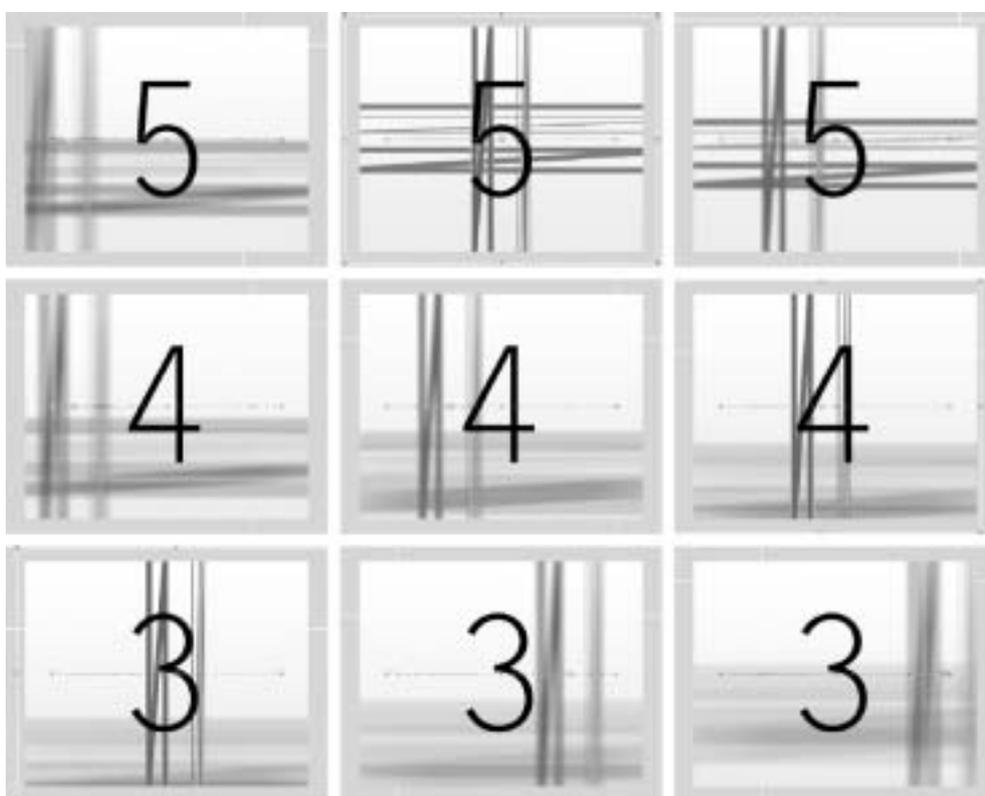


Рис. 8.13. Временная шкала и сцена для сложной анимации Flash

Графика движения

Интерполяцию ключевых кадров можно применять и к растровым изображениям. Поскольку эти изображения не содержат идентифицируемых объектов, необходимо обособить различные элементы анимации с помощью уровней. Аналогия с келевой анимацией более или менее точна — каждый уровень подобен прозрачному слою киноплетки, на которой что-то изображено. Уровни можно перемещать независимо, так что анимацию можно создавать, размещая различные элементы на различных уровнях и перемещая или изменяя уровни при переходах между кадрами. Если движение или модификацию уровней легко описать алгоритмически, возможна интерполяция ключевых кадров, подобно тому, как помощники интерполируют ключевые кадры ведущего аниматора. Обычно при переходе между ключевыми кадрами уровень может изменять свое положение, поворачиваться или масштабироваться. Данные геометрические преобразования легко поддаются интерполяции, но поскольку сейчас мы рассматриваем растровые отображения, возможно, потребуется повторная дискретизация, а следовательно, возможна потеря качества, как объяснялось в главе 5.

Ведущим настольным приложением для анимации описанного типа является AfterEffects. Из-за своего смешанного происхождения AfterEffects хорошо совместимо с Photoshop и Illustrator. Изображение Photoshop можно импортировать в AfterEffects с сохранением всех уровней (включая настроечные) и альфа-каналов; картинки Illustrator можно (также с сохранением уровней) импортировать и растеризовать. Таким образом, распространен следующий способ работы: использовать средства и инструменты Photoshop или Illustrator для подготовки элементов анимации на отдельных уровнях и им-



портировать результат в AfterEffects, где эти уровни анимируются. Изображения Photoshop должны подготавливаться с соответствующими разрешением и размером для предполагаемой среды доставки. Если их нужно представлять в уменьшенном масштабе, они должны быть достаточно большими, чтобы выдержать максимальное уменьшение, которое может применяться. Файлы Illustrator могут либо растеризоваться в момент, когда они импортируются, а затем трактуются как битовые отображения, либо непрерывно растеризоваться для каждого кадра анимации. Это означает, например, что если они будут масштабироваться, детали не будут потеряны.

Простейшие анимации создаются путем изменения положения уровней, для чего либо перетаскиваются уровни на экране, либо вводятся координаты нового положения, а затем интерполируется движение между ключевыми кадрами. Комбинируя уровни, добавляя эффекты и фильтры, которые могут изменяться со временем, можно получать движущиеся графические изображения. Импортируя набор неподвижных изображений, показанных на рис. 8.14, в AfterEffects и анимируя их описанным способом, получим видеоряд “Обратный отсчет”. Кроме интерполированного движения целых битовых изображений, не использовались никакие движущиеся элементы. Эффекты, которых можно достичь, применяя к растровым изображениям движение и нестационарные фильтры, имеют больше общего с компьютерной графикой, чем с традиционной мультипликацией или художественной анимацией. Часто их называют более характерным термином *графика движения*. Многие из этих технологий впервые опробовались при изображении названий художественных фильмов, и, в основном, они применяются при создании титров.

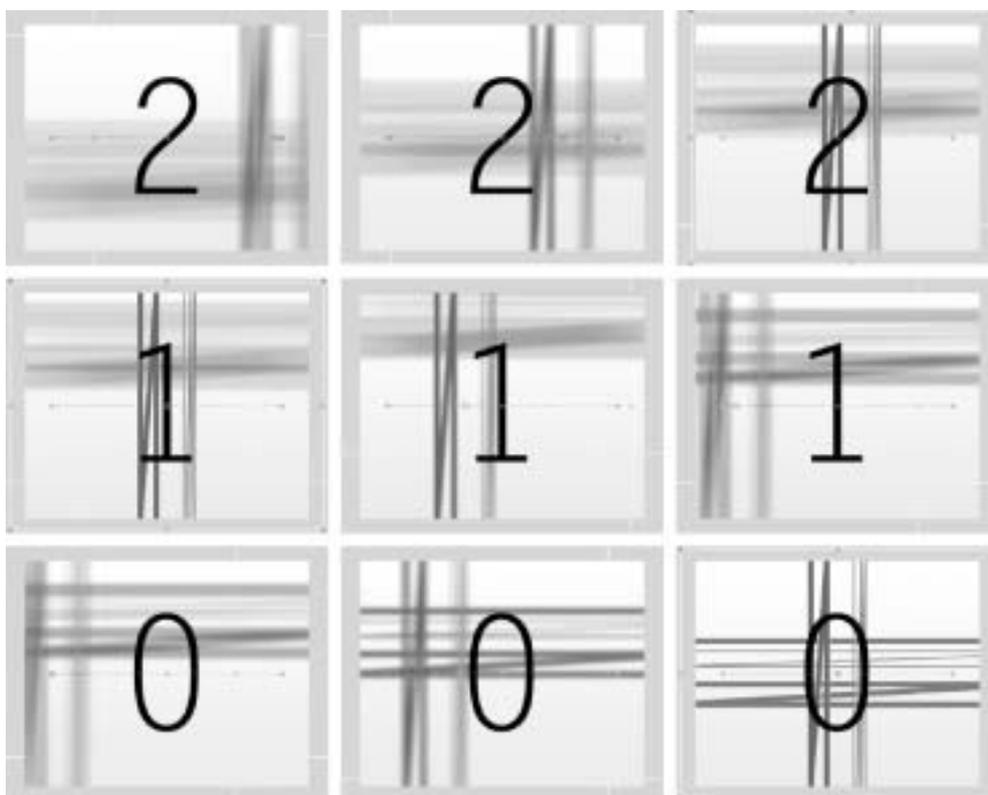


Рис. 8.14. Пример графики движения

Интерполяцию можно применить и к другим свойствам уровня. В частности, можно варьировать ориентацию уровня, имитируя его поворот. Угол поворота для ключевых кадров задается вручную, а для промежуточных вычисляется с помощью интерполяции. Также поворот можно определять автоматически вместе с движением из условия сохранения ориентации уровня относительно траектории движения. В ключевых кадрах (с последующей интерполяцией в промежуточных) может задаваться масштабирование, создающее эффект перспективы, что формирует иллюзию приближения к сцене или отдаления от нее, уменьшения или увеличения масштаба.

AfterEffects поддерживает линейную интерполяцию и интерполяцию Безье в пространственной и временной областях. Линейная интерполяция приводит к резким изменениям направления, а интерполяция Безье дает гладкие изменения направления. Существует несколько видов пространственной интерполяции, которые задаются движением уровня в окне, в котором отображается изображение. Временная интерполяция меняет скорость изменения положения во времени. Как и ранее, это может быть линейная интерполяция с постоянной скоростью, мгновенным разгоном и остановкой, как обсуждалось при рассмотрении Flash, или интерполяция Безье с плавным ускорением. Методы пространственной и временной интерполяции независимы: можно, например, использовать линейную временную интерполяцию с траекториями движения Безье и наоборот.

Степень контроля над интерполяцией пространственных свойств, предлагаемых AfterEffects, значительна. Используя обычный инструмент “кривые Безье”, можно перерисовать графики изменения значения со временем. Ключевые кадры вставляются автоматически при добавлении к графику контрольных точек. Абсолютные значения можно вводить численно, что позволяет полностью контролировать положение и скорость движения. Тем не менее существуют ограничения на тип движения, которое можно создать, интерполируя положение, угол и размер отдельного уровня. Объекты движутся, как единое целое, скользя нереально гладко, как в простых малобюджетных аппликативных анимациях, которые используются в некоторых образовательных телепередачах и программах дошкольной подготовки. Поэтому для стилизованного движения чаще применяется анимация ключевых кадров битовых изображений. Как отмечалось в главе 7, блуждающее каше часто формируется путем анимации неподвижного изображения в AfterEffects. Другим популярным приложением является анимация текста. Отдельные символы или слова можно разместить на разных уровнях и анимировать. Также текст можно расположить на траектории (как в Illustrator), а затем показать динамику его движения по траектории.

Как следует из приведенного примера с “Обратным отсчетом”, растровое представление позволяет изменять во времени не только положение, угол и размер изображения, но и другие его свойства. Таким образом, помимо геометрических преобразований, можно добиться более радикального временного изменения уровней. Как описывалось в главе 5, к растровым изображениям можно применить множество различных эффектов и фильтров. Большинство из этих фильтров имеет параметры, например размывание по Гауссу или яркость светящихся краев. Подобные параметры можно изменять во времени, используя тот же механизм интерполяции ключевых кадров, который применяется для интерполяции движения. В результате можно получить уникальные эффекты.

Например, на рис. 8.15 показана последовательность кадров, извлеченных из титров короткометражного фильма. Надпись “part of” появляется из темноты как неопределенное пятно, становится четче и ярче, достигает максимальной четкости, в таком виде сохраняется несколько секунд, а затем снова растворяется в темноте. Этого эффекта можно достичь, применив гауссово размывание к тексту и варьируя яркость. Действительный текст — это обычное неподвижное изображение, сделанное в Photoshop и не изменяющееся никаким иным способом. На рис. 8.15 также показаны графики использованных

гауссова размывания и вариации яркости. Размывание начинается с очень большого значения, при котором текст практически не виден; на некотором ключевом кадре яркость существенно снижается. Значения интерполируются до точки, в которой размывание удаляется, а яркость восстанавливается. Чтобы обеспечить гладкое затухание, используется интерполяция Безье. Затем между парой ключевых кадров значения остаются постоянными, что видно из горизонтальных фрагментов графиков, а затем для создания

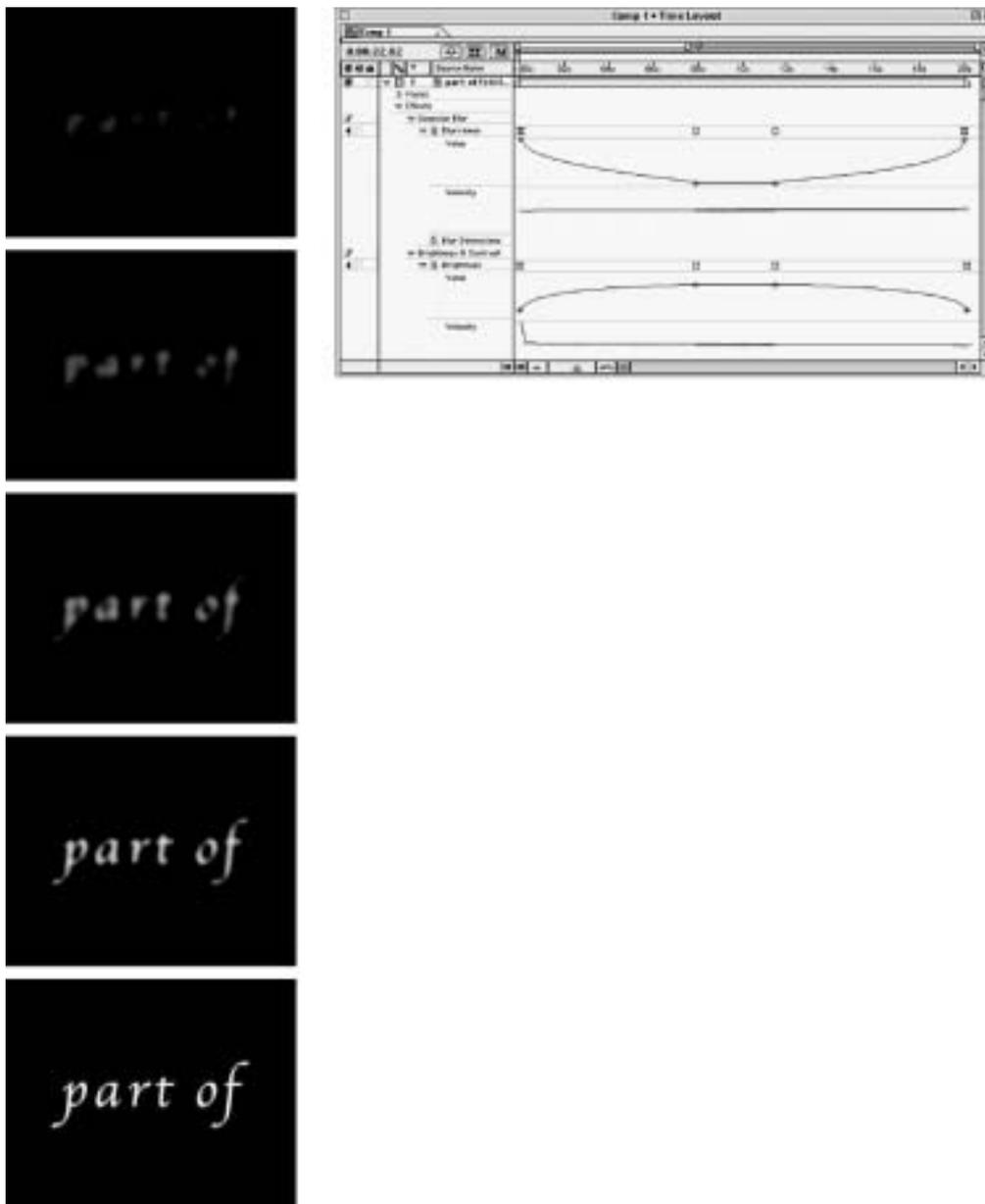


Рис. 8.15. Интерполирующие фильтры

обратного затухания надписи применяется симметричная интерполяция, переводящая изображение в ключевой кадр со значениями, идентичными значениям на первоначальном кадре.

Помимо изменения параметров фильтров неподвижных изображений и применения нестационарных эффектов, можно применять новые эффекты, которые стали возможными при добавлении к изображениям временного изменения. На рис. 8.16 показана часть альтернативной версии анимации “Обратный отсчет”, в которой к числам применяется эффект разрушения.

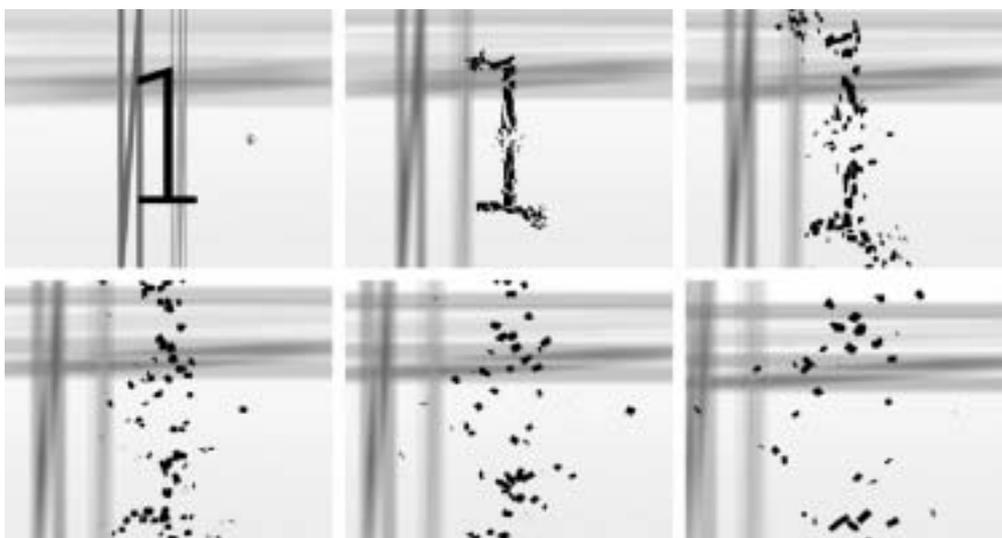


Рис. 8.16. Чисто временной эффект

Трехмерная анимация

Трехмерную анимацию легко описать, но сложно сделать. Для понимания сути процесса не требуются никакие новые концепции, кроме описанных в данной главе и главе 4. Свойства трехмерных моделей определяются численными величинами. Изменение чисел приводит к изменению таких свойств, как положение объекта в пространстве, его ориентация, характеристики поверхности и даже форма. Интенсивность и направление источников света плюс положение и ориентация камеры также определяются численно. Таким образом, чтобы анимировать трехмерную сцену, нужно всего лишь задать исходную сцену, а затем визуализировать ее, как первый кадр анимации, изменяя параметры, визуализируя следующий кадр и т.д. Поскольку изменяются численные значения, возможна определенная интерполяция ключевых кадров; для организации анимации удобно использовать временную шкалу; кроме того, для описания движения можно применять трехмерные траектории (часто представляются трехмерными сплайнами Безье). Поскольку трехмерные модели нужно визуализировать как двухмерные изображения, что подразумевает наличие точки наблюдения или камеры, а также движущихся объектов на сцене, мы можем перемещать камеру, заставляя ее “летать” над ландшафтом или перемещаться вокруг определенных объектов (рис. 8.17).

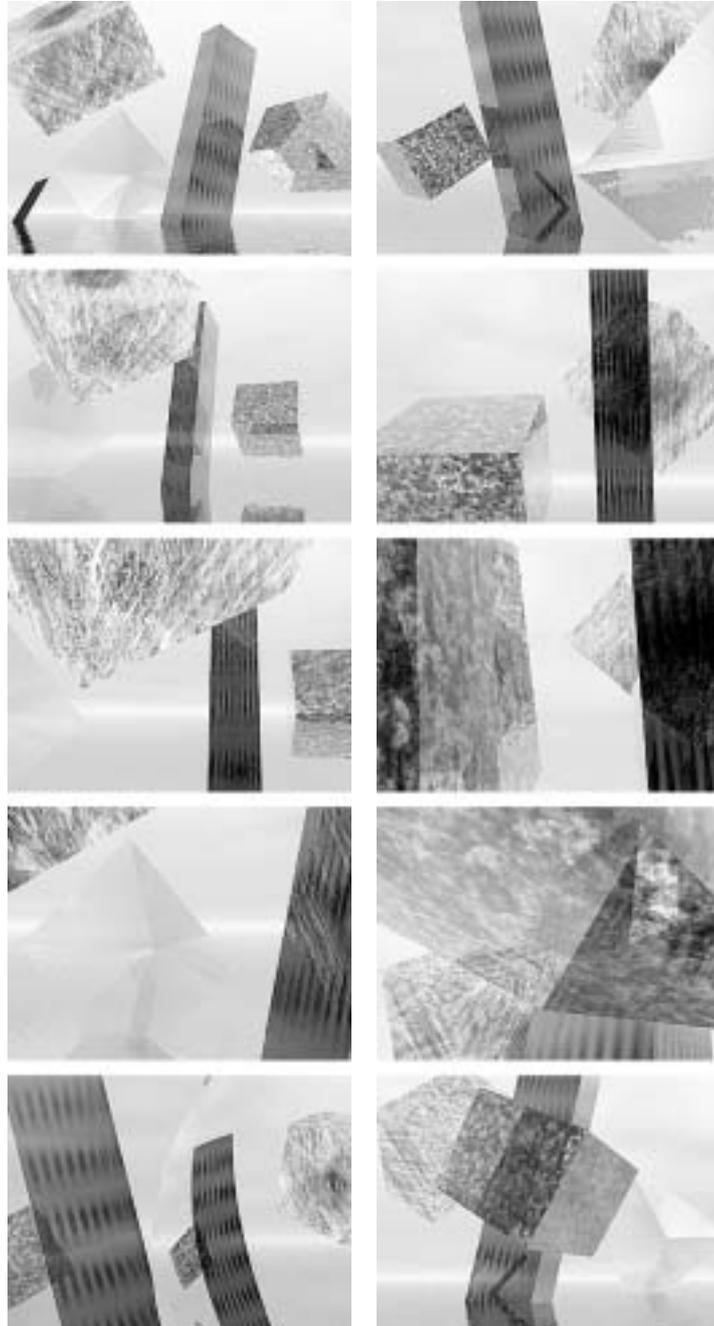


Рис. 8.17. Движение камеры вокруг трехмерной сцены

Если простую трехмерную анимацию (например, поворачивающиеся логотипы и вращающиеся глобусы) действительно можно сделать очень просто, и существуют пакеты, специализирующиеся на решении подобных задач, высококачественная фотореали-

стичная анимация, подобная используемой в телерекламе, музыкальных видеоклипах и фильмах со спецэффектами, требует огромных ресурсов: времени, мощности процессора и памяти, специализированного программного обеспечения, но прежде всего — высококвалифицированного персонала. Производители мультимедиа редко могут предложить такие ресурсы. По этим причинам наше описание трехмерной анимации будет ограниченным, а читателям, серьезно интересующимся данным вопросом, мы рекомендуем обратиться к соответствующей литературе, указанной в конце книги.

Существует несколько факторов, делающих трехмерную анимацию сложнее, чем она кажется на первый взгляд. Первый — большинство людей испытывают сложности с наглядным представлением трехмерных сцен. Если добавить время, получим четыре измерения, с которыми нужно работать посредством двухмерного компьютерного экрана. Данная сложность усугубляется второй проблемой — вычислительной мощностью, необходимой для визуализации трехмерной анимации. Такие сложные алгоритмы затенения, как построение хода лучей, требуют значительного времени для обработки одного изображения. В анимации каждую секунду необходимо обрабатывать от 12 до 30 изображений. В результате генерация полностью визуализированных изображений предварительного просмотра возможна только на мощнейших рабочих станциях или в сетях распределенной обработки данных.

Имея большой бюджет, терпение и опыт, данные проблемы можно преодолеть, но при этом неизбежно придется изменять множество параметров, чтобы получить правдоподобное движение. На наивысшем уровне трехмерной компьютерной анимации распространено следующее решение: предоставить аниматору богатый интерфейс, дающий полный контроль над движением. В контексте анимации фигур это похоже на действия кукловода при управлении марионеткой — иногда даже используются скафандры, оборудованные датчиками движения, подобные применяющимся для управления анимированными марионетками. Это сильно контрастирует с подходом к анимации, используемым программистами, инстинкты которых выступают за глобальную автоматизацию. В попытке преодолеть ограничения простых схем интерполяции разрабатывались схемы автоматического формирования правдоподобного движения в трех измерениях.

Одним из ключевых подходов является разработка определенных схем поведения, которые могут приписываться объектам и определять их взаимодействие. Один из примеров схемы поведения — “Объект А должен указывать на объект В”. Данная схема чрезвычайно полезна, когда указывающий объект — это камера или источник света. Если камера “смотрит” на объект, она будет удерживать его в поле зрения при любом движении; прожектор, направленный на объект, автоматически будет следовать за ним, как, например, реальный прожектор следует за актером на сцене. Реальные объекты на сцене также можно заставить указывать друг на друга; например, можно сделать, чтобы подсолнечник всегда был повернут к солнцу. Возможны и вариации этой схемы поведения — сопровождение одного объекта другим, т.е. повтор его движения на расстоянии. Данный эффект можно использовать для анимации сцен погони. Как и в случае с указанием, эту схему поведения также можно навязать камере, после чего она будет следовать за объектом или символом по сцене, заходя даже туда, куда реальная камера физически не смогла бы попасть.

В некоторых системах трехмерной анимации реализована схема поведения, основанная на физических законах движения. Например, они позволяют пользователю указывать, что объект ускоряется от нулевой скорости под воздействием внешней силы, величина которой задается как параметр. Развивая эту схему, можно заставить объекты реалистично сталкиваться с твердыми поверхностями или отскакивать от них рикошетом. Подобное поведение основано на простых законах физики, определяющих возможное движение несколькими уравнениями. К сожалению, многие реалистичные типы движения невозможно описать так просто, поэтому нужно использовать другие методы.

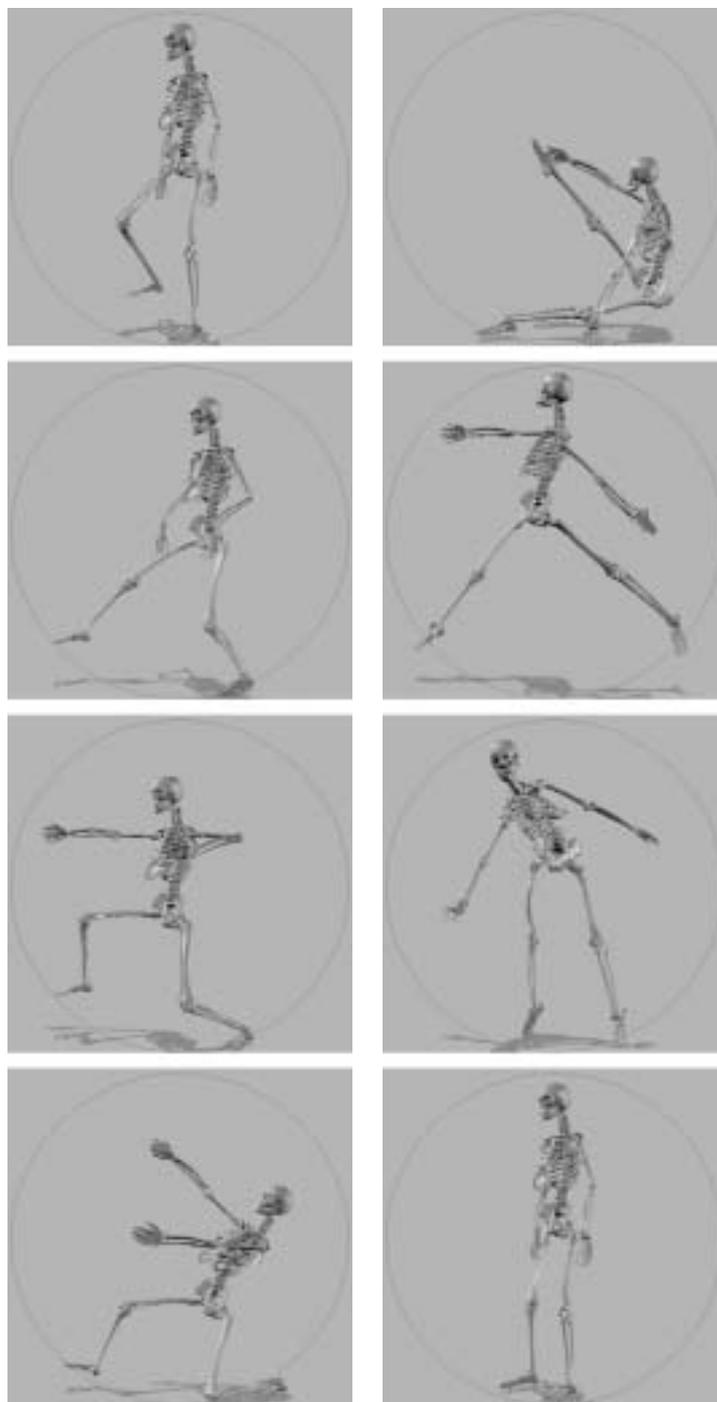


Рис. 8.18. Танцующий скелет

Движение тел безотносительно к массе или силе называется *кинематикой*, т.е. важно только то, как могут двигаться тела, а не что заставляет их двигаться. В анимации кинематика более всего полезна в связи с шарнирными структурами, такими как конечности человека или тела животных. Поскольку эти структуры соединены в единое целое, различные части вашей руки, например, могут двигаться только определенным образом. Чтобы получить реалистичное движение, на трехмерную модель руки нужно наложить такие *кинетические условия*, какие имеются у реальной руки; например, если плечо поднято вверх, вслед за ним поднимаются предплечье и кисть. Если в реальной жизни движение плеча определяет движение остальной части руки, в системах анимации подобное моделирование движения (от начала цепочки соединенных элементов к концу) не очень удобно. Гораздо легче расположить объект, который находится в конце цепочки (скажем, кисть руки), а затем определить положение остальных элементов (остальной части руки), не противоречащее положению первого элемента. Как правило, ограничения на движение накладываются исходя из крайних ситуаций; например, при ходьбе ступня должна при каждом шаге подниматься над землей и опускаться на землю. Чтобы заставить фигуру идти правильно, вначале нужно разобраться, как движутся конечности при движении бедра, поэтому для программы, решающей задачу движения ноги, оптимально начинать с указанного аниматором расположения ступни, и т.д. Данный тип моделирования называется *обратной кинематикой*, поскольку здесь “танцуют” от эффекта к его причине. Обратную кинематику можно применять к любой структуре, которая моделируется как связанная цепочка объектов. Как правило, программы трехмерной анимации, поддерживающие подобные структуры, также позволяют использовать обратную анимацию. Например, в Poser можно задать автоматическое применение обратной кинематики к рукам и ногам фигуры (рис. 8.18).

Немного поэкспериментировав, можно обнаружить, что расчет движения с использованием обратной кинематики не является совсем уж прямолинейным. В частности, кинематические условия для рук и ног не полностью определяют возможные движения конечностей, согласующиеся с движением крайних точек. Попробуйте коснуться указательным пальцем правой руки кончика носа. Сколько при этом существует различных траекторий движения локтя правой руки? Чтобы зафиксировать определенный тип движения, нужно добавить другие условия, например минимум потенциальной энергии всей структуры. Чтобы сформировать движение, которое нарушает эти условия, но является физически возможным, нужно отказаться от обратной кинематики и вручную разместить элементы тел.

Виртуальная реальность

Обычно фразу “виртуальная реальность” используют для описания крайней степени восприятия искусственного мира. С помощью установленных на голове дисплеев, чувствительных к движениям головы, изображение проектируется в глаза пользователя. Далее эти изображения модифицируются при перемещении головы, так что пользователю кажется, будто он находится внутри трехмерного мира и осматривается в нем. “Информационные перчатки” отслеживают движения рук, позволяя дисплею отображать кисти пользователя. Тактильные интерфейсы обеспечивают тактильную обратную связь, так что пользователи могут трогать и ощущать объекты в виртуальном мире. В крайнем проявлении виртуальная реальность подобного типа может стать наивысшей степенью развития мультимедийных средств, одновременно воздействуя на все чувства.

Высокая стоимость аппаратуры интерфейса, необходимой для создания глубокой виртуальной реальности (плюс понятное нежелание части преимущественно совершен-

нолетней аудитории погружать тело в странное электронное устройство), ограничила область применения этой технологии в летных и промышленных тренажерах и в специальных играх. В настоящее время развилось иное видение виртуальной реальности как трехмерной графики, которую можно исследовать. Впрочем, даже такая разновидность виртуальной реальности не получила широкого признания, в основном “благодаря” высоким требованиям к вычислительной мощности, приводящим к неутешительным результатам на домашних компьютерах. В то же время две технологии виртуальной реальности заслуживают отдельного обсуждения, поскольку их можно более или менее удачно внедрить в Web-страницы. Кроме того, с повышением мощности компьютеров эти технологии могут стать более важными и соответствующими ожиданиям энтузиастов.

Язык моделирования виртуальной реальности

Язык моделирования виртуальной реальности (Virtual Reality Modeling Language — VRML) был разработан на волне энтузиазма по поводу виртуальной реальности и World Wide Web в 1994 году для создания механизма распределения виртуального мира по Internet с использованием в качестве интерфейса Web-браузеров. До этого VRML был текстовым языком, позволяющим описывать трехмерные объекты и сцены способом, подобным записи программ на обычных языках программирования. Язык VRML 1.0 предоставлял несколько большие возможности, и основной дополнительной его особенностью была возможность внедрения в сцену гиперссылок, использование URL. Позже VRML 2.0 добавил поддержку интерактивности посредством сценариев, как будет описано в главе 16, и позволил внедрять в миры VRML видео и аудио. В декабре 1997 года VRML получил статус стандарта ISO.

VRML разрешил спецификацию объектов через их геометрию и материал, из которого они состоят. Появилась возможность накладывать на поверхность текстуру, а сами объекты размещать в трехмерном пространстве с помощью преобразований. Сцены стало возможным освещать по-разному, задавая тип и положение источников света. Таким образом, основной язык позволял описывать сцены, подобные тем, которые строятся в обычных программах трехмерного моделирования, хотя ему недоставало некоторых замысловатых особенностей, в частности — сплайнов NURBS и меташариков. Описание является явным; например, ландшафт моделируется как сетка возвышений — разновидность объекта VRML, задающего набор точек в виде сетки, каждый узел которой имеет свое возвышение над нулевым уровнем. В VRML размеры сетки и высота каждой точки должны задаваться явно. Из сказанного понятно, что построение сцен VRML вручную — это утомительный и чреватый возникновением ошибок процесс. Большинство людей, занимающихся трехмерным моделированием, создают VRML-образы как выход обычных интерактивных средств моделирования, поскольку таким образом генерировать сцены гораздо проще.

Может показаться, что VRML — это не более чем альтернативное представление выхода программы моделирования, и в основном это правда. Как отмечалось ранее, данный язык предлагает дополнительные возможности интерактивного взаимодействия и встроенных средств мультимедиа, но основное отличие связано не с языком, а со способом этого отображения. После того как файл VRML загружен в подходящий браузер (Web-браузер с соответствующим модулем или специализированный VRML-браузер), пользователь может исследовать описываемый этим файлом трехмерный мир, т.е. точку наблюдения можно перемещать в пространстве модели, как будто пользователь сам движется в этом мире. В данном контексте VRML заслуживает того, чтобы его называли разновидностью виртуальной реальности.

Чтобы создать иллюзию движения в трехмерном пространстве, VRML нужно визуализировать в реальном времени. Как уже неоднократно отмечалось, реалистичная визуализация трехмерных моделей является вычислительно сложной задачей, которая обычно под силу только специальному аппаратному обеспечению, например видеокартам с 3D-ускорителем. Это одно из главных препятствий для широкого распространения VRML, хотя более существенным может быть и отсутствие широкой поддержки VRML со стороны ведущих производителей программного обеспечения. На момент написания данной книги было распространено множество форматов (в большинстве своем запатентованных) доставки трехмерных моделей через World Wide Web, причем ни один из них пока не получил всеобщего признания.

Виртуальная реальность QuickTime

Виртуальная реальность QuickTime (QuickTime Virtual Reality — QTVR) относится к самому начальному уровню виртуальной реальности. Существует два типа роликов QTVR: панорамные картины и клипы с объектами. Первые представляют обзор сцены на 360° (интерьер комнаты или долина, окруженная горами). Используя мышь, пользователь может перетаскивать сцену, как бы осматриваясь. Кроме того, возможно уменьшение или увеличение масштаба, благодаря чему сцену можно наблюдать более или менее детализировано. Клипы с объектами, наоборот, позволяют пользователю исследовать (также с помощью мыши) объект с различных сторон, как бы ходя вокруг него. Ролики QTVR любого типа могут содержать *горячие точки* — активные области, содержащие ссылки на другие ролики. Щелчок на горячей точке приводит к замене текущего ролика клипом, привязанным к этой точке. Основное назначение подобных точек — создание дверей, щелкая на которых, пользователь переходит в другую комнату.

Ролики QTVR можно создавать в некоторых программах, например в Bryce. Также их можно строить по фотографиям, а чтобы получить лучшие результаты, камера (для панорамной съемки) или объект устанавливается на специальную вращающуюся платформу. Последовательно поворачивая платформу с небольшим угловым шагом, получаем набор снимков, которые затем сканируются (или считываются в компьютер, если используется цифровая камера), и специальное программное обеспечение их “сшивает”, визуализируя результат как QTVR. Задача платформы — гарантировать, что отдельные снимки совмещаются идеально. Если для съемки панорамы используется обыкновенный тренажник или для съемки роликов с объектами — поворотный стол либо другое подобное устройство, возможны разрывы; программное обеспечение, с помощью которого выполняется сшивание, может попытаться устранить эти неоднородности, но в результате изображения могут исказиться.

Поскольку QTVR является частью QuickTime, панорамные картины и клипы объектов можно объединять с аудио и видео. Радует еще и то, что их можно просматривать с помощью любого программного обеспечения, которое использует QuickTime; в частности, модули QuickTime для Web-браузеров позволяют встраивать QTVR в Web-страницы так же, как видеоклипы (см. главу 12).

QTVR и VRML можно считать пародией на исходное представление о глубокой виртуальной реальности, но их достоинством является реализуемость без специальных устройств сопряжения или мощных рабочих станций. Они предлагают новые подходы к сопряжению с мультимедиа, основанные на организации средств информации в трехмерном пространстве.

Упражнения

1. Назовите преимущества и недостатки использования сканера или цифрового фотоаппарата для записи традиционных художественных произведений в форме анимационной последовательности. Для каких типов анимации (если такие существуют) стоит использовать видеокамеру, подключенную к компьютеру?
2. Как рисованную анимацию можно встроить в “живое” видео, не используя программу создания спецэффектов?
3. Если анимационную последовательность нужно записать как ролик QuickTime, какие факторы влияют на выбор кодека? При каких обстоятельствах анимационную последовательность выгодно рассматривать как видеоряд с натуральным движением?
4. Когда для записи анимационной последовательности стоит использовать анимированный GIF? Какие проблемы сопутствуют такому решению?
5. Чем дорожка спрайтовой анимации радикально отличается от видеодорожки, содержащей анимацию?
6. Для каких работ идеально подходит спрайтовая анимация и почему?
7. Какие проблемы связаны с использованием стандартной линейной интерполяции для создания промежуточных кадров, находящихся между ключевыми?
8. Как использовать средства ускорения Flash для показа объекта, который разгоняется и замедляется?
9. Опишите движение объекта, положение которого анимируется в AfterEffects с интерполяцией траектории движения по методу Безье и линейной интерполяцией скорости.
10. В каком-либо растровом графическом приложении, например Photoshop или Painter, создайте очень простые титры для видеоклипа (одно изображение) и запишите их в виде файла неподвижного изображения. Используя любые доступные средства (Premiere, AfterEffects и т.д.), создайте приятные 10-секундные титры, просто применив к этому изображению нестационарные эффекты и фильтры. (Если вы хотите достичь более сложного результата и имеете необходимые инструменты, можете создать собственное изображение из нескольких уровней и анимировать их по отдельности.)
11. Содержат ли модели, генерируемые трехмерными приложениями, достаточно информации, чтобы, объединив их с тактильным интерфейсом, можно было обеспечить обратную связь с пользователем по осязанию? Если нет, то какая нужна дополнительная информация?