

# Компьютерная графика в видеотренажерах

М.В.Михайлюк

Центр визуализации и спутниковых информационных технологий,  
Институт микропроцессорных вычислительных систем РАН  
Москва, Россия

## Аннотация

Рассматриваются некоторые основные составляющие использования компьютерной графики в системах визуализации имитационно-тренажерных комплексов. К ним относятся подготовка трехмерных сцен, их предварительная обработка, визуализация и управление объектами, моделирование приборов и окружающей среды.

**Ключевые слова:** система визуализации, видеотренажер, трехмерная сцена, реальный режим времени.

## 1. ВВЕДЕНИЕ

Одной из областей применения компьютерной графики является тренажеростроение. Тренировка космонавтов, пилотов, операторов сложных систем на реальных установках и в реальных условиях слишком дорога, а часто и очень опасна. Альтернативой этому является создание имитационно-тренажерных комплексов, которые в максимально возможной степени приближены к реальным установкам и позволяют тренирующимся приобрести правильные и устойчивые навыки. Схематически тренажерный комплекс можно представить в виде нескольких взаимосвязанных составляющих (см. рис. 1).

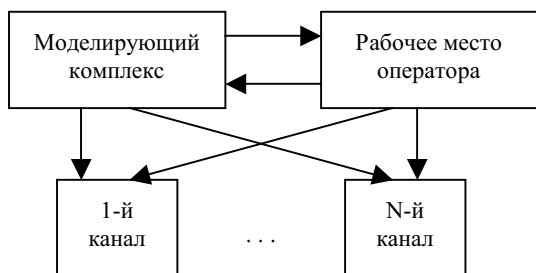


Рисунок 1: Схема тренажерного комплекса.

Рабочее место обычно представляет собой копию реального пульта управления и содержит ручки управления, приборы, шкалы и так далее. Каждый канал представляет собой монитор, в котором синтезируется трехмерная визуальная обстановка, которую оператор должен видеть через стекла, камеры и другие средства наблюдения. Например, при управлении самолетом каналы могут представлять фронтальное и боковые стекла кабины самолета. И, наконец, моделирующий комплекс включает математическую модель процесса и управляет реакцией системы на действия оператора. Поворот оператором некоторой ручки управления отслеживается механической моделью и параметры передаются моделирующему комплексу. На основе математической модели, заложенной в комплексе, рассчитываются новые координаты и ориентации всех объектов сцены и новые параметры передаются по сети системам визуализации всех каналов, которые производят визуализацию в реальном масштабе времени. Такой режим работы предъявляет довольно строгие требования к системам визуализации, работающим в составе имитационно-тренажерных комплексов. Основными требованиями являются следующие:

- достаточно подробная виртуальная модель объектов, а также окружающей среды (небо, звезды, водная и земная поверхность и т.д.);
- хорошее качество визуализации, без ступенек на границах объектов, без исчезновения или мигания мелких деталей (антиалиасинг);
- работа в реальном режиме времени, т.е. визуализация сцены со скоростью не менее 25 кадров в секунду.
- имитация погодных условий (дождь, снег, дымка, туман);
- имитация приборов и специальных средств наблюдения, а также реальных условий работы, включающих помехи, засветки, блики от оптических приборов и т.д.;
- синхронизация каналов, если их несколько, то есть качественная сшивка соседних изображений, формируемых разными компьютерами;
- доступность аппаратных средств для тренажеров широкого распространения.

## 2. ПОДГОТОВКА ТРЕХМЕРНЫХ СЦЕН

Подготовка трехмерных сцен для системы визуализации производится либо с помощью коммерческих продуктов типа 3D Studio MAX, либо с помощью самостоятельно разработанного программного обеспечения. Мы рассмотрим здесь лишь первую возможность. В системах типа 3D MAX принята полигональная модель объектов, то есть все объекты строятся из примитивов (параллелепипедов, цилиндров, сфер и т.д.), которые, в свою очередь, строятся из треугольников (полигонов). Объект можно поместить в нужное место пространства, присвоить ему материал, свойства которого (параметры отражения света) можно задать, а также можно наложить на поверхность объекта текстуру. Кроме объектов можно создать источники света и камеры, задать их свойства и расположить в нужных местах пространства сцены.



Рисунок 2: Виртуальный город

При создании виртуального объекта необходимо отразить все детали для повышения адекватности модели реальному объекту. С другой стороны, если эти детали делать как

трехмерные объекты, то число полигонов в сцене будет очень большим и трудно будет достигнуть приемлемой скорости визуализации. Поэтому приходится идти на компромисс – крупные и важные детали проектировать как части полигональной модели, а мелкие включать в текстуры. Обычно при создании объектов в качестве исходных данных для каркасной модели используются элементы проектной документации, а для создания текстурных карт – фотографии или киносъемки реальных объектов или синтезированные изображения (если объекты в реальности еще не созданы). На рис. 2 приведен пример модели виртуального города, построенного из простых примитивов с использованием текстурных карт.

При построении каркасной трехмерной модели обычно стремятся уменьшить число граней, из которых она строится, так как от числа граней существенно зависит скорость визуализации. Иногда это можно сделать с помощью специальных встроженных программ (plug-in), иногда приходится делать вручную. Например, 3D MAX строит обычную пирамиду как усеченную с очень маленьким верхним основанием и каждое основание разбивается на треугольники центральной точкой (см. рис. 3). Поэтому реально число треугольников в n-гранной пирамиде получается равным  $2n + 2n = 4n$ . Если же склеить центральную точку основания с любой вершиной основания и удалить все треугольники, содержащие ребра верхнего основания, то полученный объект с точки зрения визуализации будет соответствовать обычной пирамиде и содержать  $n+(n-2) = 2n-2$  полигонов. Это приходится выполнять вручную.

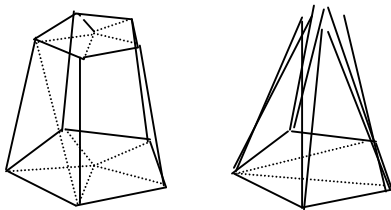


Рисунок 3: Оптимизация пирамиды

Если создается новый объект и его фотографий нет, то можно построить подробную трехмерную модель, а затем с помощью специального IPAS процесса в 3D MAX получить развертку его поверхности, визуализировать ее и использовать в качестве текстурной карты для упрощенной модели того же объекта. В основном этот метод подходит для объектов цилиндрической или конической формы. На рис. 4 слева изображена подробная модель объекта, в центре – развертка его боковой поверхности, а справа – цилиндр, обтянутый полученной текстурой. Такой способ часто дает приемлемые результаты и позволяет сократить число граней в десятки раз за счет включения множества мелких деталей в текстуру.

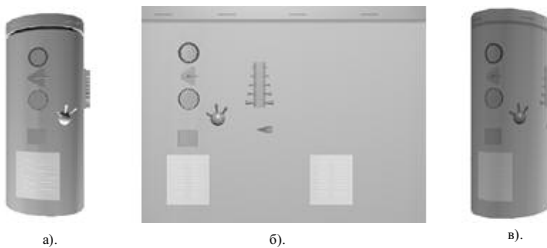


Рисунок 4: Получение текстуры путем развертки поверхности объекта.

### 3. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА

Обычно внутренний формат хранения элементов трехмерной сцены в системе визуализации отличается от выходных форматов системы подготовки сцен. Поэтому необходимо конвертировать элементы из одного формата в другой. Часто написание конвертора является чисто технической задачей, особенно когда известен исходный выходной формат. Например, файлы типа 3ds и max представляют собой последовательность иерархически вложенных информационных кусков - чанков (chunks), каждый из которых содержит идентификационный номер типа куска, его длину и информационную часть. По идентификационному номеру однозначно восстанавливается структура информационной части, в которой также записаны вложенные чанки. Рекурсивно обрабатывая эту структуру, можно выделить любой элемент информации, записанный в 3ds файле сцены и перевести его в другой удобный для дальнейшей обработки формат.

В то же время правильное использование сконвертированных параметров может представлять сложности, так как алгоритмы, использующие их в системе подготовки, часто неизвестны. Поэтому синтезированные изображения объектов и их поведение в системе визуализации может отличаться от изображения и поведения в системе подготовки. Например, интерполяция ориентаций объектов при одних и тех же исходных данных может выполняться по различным алгоритмам и траектории движения в системе 3D MAX и системе визуализации могут отличаться.

После конвертирования производится предварительная обработка трехмерной виртуальной сцены. На этом этапе можно выполнить всю работу, результаты которой не будут меняться от кадра к кадру. В качестве примера таких работ приведем разбиение полигональной модели на множество треугольных стрипов и фэнов и вычисление нормалей к вершинам на основе информации о группах сглаживания. Треугольный стрип представляет собой множество треугольников  $T_i = \{V_i, V_{i+1}, V_{i+2}\}$ , образованных последовательностью вершин  $V_1, V_2, \dots, V_{n+2}$ . Первый треугольник стрипа определяется тремя вершинами (на рис. 5 это  $V_1, V_2$  и  $V_3$ ). Второй треугольник содержит ребро  $V_2V_3$ , поэтому для его задания достаточно указать только одну вершину  $V_4$ . Следующий треугольник содержит ребро  $V_3V_4$  и требуется добавить только вершину  $V_5$  и так далее.

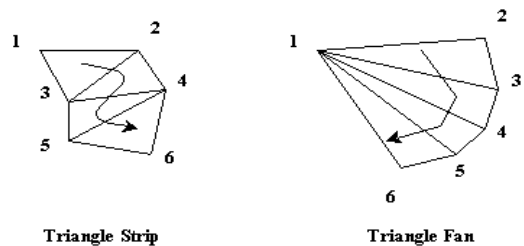


Рисунок 5: Стрипы и фэны.

Треугольный фэн (веер) представляет собой множество треугольников  $T_i = \{V_1, V_{i+1}, V_{i+2}\}$ , образованных последовательностью вершин  $V_1, V_2, \dots, V_{n+2}$ . При задании фэна сначала указывается общая вершина  $V_1$ , а затем остальные вершины в порядке обхода общих ребер. Поэтому стрип из  $n$  треугольников будет задан последовательностью из  $n+2$  вершин вместо  $3n$  вершин, если задавать каждый треугольник отдельно. Так как обработка каждой вершины в графической системе занимает определенное время, то уменьшение числа обрабатываемых вершин ускоряет визуализацию примитива. Поэтому в подсистему предварительной обработки обычно входит модуль автоматического разбиения трехмерных объектов на множество стрипов и фэнов.

Второй возможной составляющей этапа предварительной обработки является вычисление нормалей. Так как закраска граней производится на основе нормалей в их вершинах, то правильное задание нормалей позволяет сделать плавный переход между гранями (т.е. сделать невидимыми ребра). На рис. 6 приведена модель космического модуля, корпус которой реализован двенадцатигранным цилиндром. Однако за счет сглаживания он представляется совершенно гладким. Множество граней, между которыми переход должен быть сглажен, задается в 3D MAX так называемой группой сглаживания. Если нормали не меняются в процессе визуализации, то их вычисление на основе этих групп сглаживания естественно включить в этап предварительной обработки. В этот этап входят и другие задачи, например, вычисление координат охватывающего объект параллелепипеда или центр и радиус описанной сферы, расчет векторов касательных к сплайновым интерполяционным кривым в ключевой анимации, подготовка таблиц для упрощенных моделей расчета освещенности и так далее.

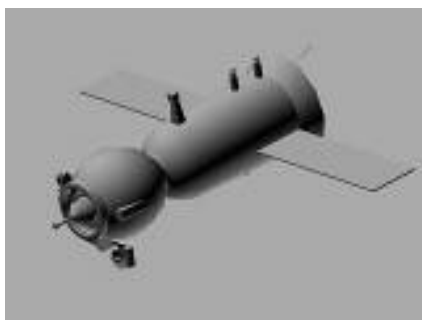


Рисунок 6: Сглаженная модель космического модуля.

#### 4. ВИЗУАЛИЗАЦИЯ ОБЪЕКТОВ ВИРТУАЛЬНОЙ СЦЕНЫ.

Основными целями при визуализации трехмерных объектов являются хорошее качество, реалистичность и скорость. Качество визуализации определяется многими составляющими, среди которых поддерживаемое разрешение экрана, используемые модели освещенности и закраски граней, алгоритмы сглаживания краев при растеризации (антиалиасинг) и т.д. Реалистичность

изображения определяется качеством используемых текстур. Кроме того, важное значение имеют такие составляющие как тени, отражения и блики от источников света. Тени дают другой вид объекта (со стороны источника света), за счет чего мы можем лучше представить взаимное расположение объектов и источников света в сцене. Примером является тень от мишени ориентира при стыковке космических аппаратов. Тени не отбрасываются автоматически виртуальными объектами, их необходимо создавать искусственно. При этом трудно ожидать, что каждый объект в сцене будет отбрасывать тень на все остальные объекты. Да и реализовать отбрасывание тени каждым объектом на все остальные в реальном режиме времени практически невозможно. Одним из подходов является задание множества объектов, отбрасывающих тень (затеняющих) и множества объектов, "принимающих" тень (затеняемых) и построение соответствующей модели тени для каждой пары. Если поверхность, на которую падает тень, является плоской или близка к плоской, то можно использовать метод проективного создания теней, при котором затеняющий объект просто проецируется на плоскую поверхность, а затем визуализируется как отдельный примитив. В этом случае вычисление тени включает два этапа: вычисление матрицы проекции и визуализация спроецированного объекта с нужным цветом. Очевидно, вычислительная сложность метода напрямую зависит от сложности затеняющего объекта. Для получения реалистичности тени выполняется смешение ее цвета с цветом поверхности, на которую она падает. На рис. 7 показан пример тени, отбрасываемой мишенью при стыковке к модели международной космической станции.

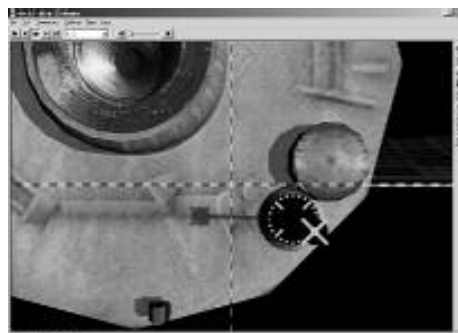


Рисунок 7: Реализация тени от мишени космического модуля.

Если поверхность, на которую падает тень, имеет сложную форму, используют другие методы, например, метод теневых объемов. В этом методе тень, отбрасываемая объектом, рассматривается как полигональный объем, то есть объемное тело, заданное многоугольниками. «Теневой объем» формируется из лучей, испускаемых источником освещения, далее проходящих через контур затеняющего объекта (со стороны источника света) и продолжающихся за пределы сцены. Заданный таким образом, теневой объем является усеченной пирамидой. Для определения того, какие части объектов сцены будут находиться в теневом объеме, используется буфер трафарета (stencil buffer, далее S-буфер). Метод относится к классу многопроходных и включает следующие этапы:

- сначала вся сцена выводится с атрибутами рисования в тени (выключен источник света);
- затем рисуется теневой объем (только в S-буфере), при этом операции в S-буфере задаются таким образом,

что после данного шага ненулевое значение получают затеняемые пиксели;

- и, наконец, сцена выводится второй раз с включенным источником света и включенным тестом трафарета с условием его прохождения при равенстве значения в S-буфере нулю.

Следует отметить, что эта процедура неинвариантна относительно положения наблюдателя, а именно, S-буфер необходимо инициализировать различными значениями в зависимости от того, внутри или снаружи теневого объема находится наблюдатель. Скорость метода зависит как от сложности затеняющего объекта, так и сложности всей сцены (поскольку она выводится дважды).

Блики (flares) образуются в оптической системе объектива камеры при наличии в поле зрения ярких источников света. Являясь специфическим искажением изображения, блики, тем не менее, специально имитируются средствами компьютерной графики для повышения реализма синтезируемых изображений. Повышение реалистичности виртуальных сцен при добавлении эффекта бликов можно объяснить следующими причинами:

- Данное явление привычно для человеческого глаза, поэтому, чисто субъективно, изображение с данным эффектом кажется более реальным.
- Добавление эффекта бликов дает впечатление повышенной яркости, что помогает расширить воспринимаемый динамический диапазон синтезируемого изображения.

Точная психофизическая модель явления весьма сложна и должна учитывать огромное количество параметров, поэтому строится упрощенная модель. Она состоит из основного блика, включающего радужный ореол, ресничную корону и эффект свечения и некоторого числа вторичных бликов, которые выглядят обычно как несколько полупрозрачных колец различного размера и положения. Основной блик расположен вокруг источника света, а количество, размер и расположение вторичных бликов зависят от характеристик объектива, однако, в любом случае, общее свойство вторичных бликов заключается в том, что они располагаются на одной линии, проходящей через центр изображения. Данная линия, которую условно можно назвать вектором бликов, зависит лишь от положения источника света в экранной системе координат (см. рис. 8).

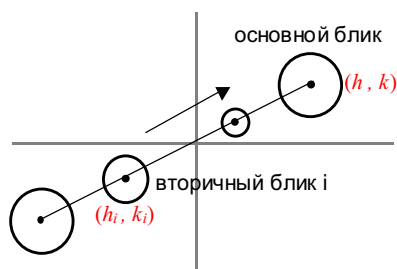


Рисунок 8: Схема расположения бликов.

Если координаты источника света  $i$ , соответственно, основного блика, в системе координат, связанной с экраном, равны  $(h, k)$ , то координаты вторичных бликов в этой системе будут кратны им, т.е. равны  $a_i \cdot (h, k)$  для некоторых коэффициентов  $a_i$ . Размер и цвет вторичных бликов индивидуальны для каждого конкретного объектива, поэтому выбираются либо произвольно, либо на основе практических наблюдений для

заданной оптической системы. Отметим, что визуализация бликов производится на заключительном этапе построения сцены поверх уже ранее визуализированных объектов. На рис. 9 изображен пример модели бликов.

Задача повышения скорости визуализации решается на каждом этапе. На этапе предварительной обработки производится оптимизация организации и хранения данных, а также представление множества полигонов в виде специальных конструкций (стрипов и фанов). В процессе самой визуализации также используется ряд методов. Прежде всего следует отметить оптимизацию подачи данных на конвейер визуализации и отбрасывание объектов, не попадающих в пирамиду видимости.

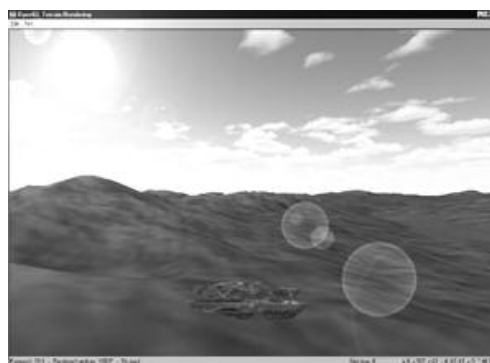


Рисунок 9: Пример моделирования бликов.

Каждое изменение параметров визуализации всегда приводит к дополнительным издержкам по переконфигурированию конвейера. Проведенные исследования показали, что наиболее "дорогими" являются операции по изменению параметров материала, а именно переключение текстур, изменение цветов и параметра степени зеркального отражения. Естественный порядок обработки сцены состоит в переборе всех объектов, для каждого из которых перебираются все материалы, из которых состоят его части. Простое изменение этого порядка (т.е. для каждого материала выводятся части всех объектов с данным материалом) позволяет повысить скорость визуализации на 10-20%. Использование многих стандартных средств графической библиотеки (дисплейные списки, массивы вершин, буферы вершин, текстурные объекты и т.д.) также позволяют увеличить скорость синтеза изображений.

Очевидно, что обработка полигонов для объектов, не попадающих в пирамиду видимости или полностью перекрываемых другими объектами, является излишней. Конечно, графическая система производит отсечение таких объектов, однако все же тратит на них время (например, трансформируя объекты, рассчитывая освещенность их вершин и производя отсечение каждого полигона в отдельности). Для экономии этого времени можно осуществлять предварительную проверку пересечения пирамиды видимости и сферы (или параллелепипеда), охватывающей объект. Если объект вообще не попадает в пирамиду видимости, он не посылается на визуализацию, если пересекается с ней – он передается на конвейер. Дальнейшая визуализация только видимых объектов позволит уменьшить объем передаваемых данных, то есть нагрузку на системную шину, и, соответственно, снизить количество обрабатываемых данных.

Если объект попадает в поле зрения, но перекрывается другими объектами, то его полигоны проходят все стадии конвейера, их отбрасывание осуществляется лишь на стадии теста глубины и то лишь в случае, если части сцены, перекрывающие этот объект, нарисованы ранее. Для

анализа и выявления таких объектов существует много методов, среди которых следует отметить упорядочение объектов или их полигонов по глубине, двоичное деление пространства, порталную технологию и т.д.

## 5. СОЗДАНИЕ И ВИЗУАЛИЗАЦИЯ ФОНА СЦЕНЫ.

Моделирование окружающего фона сцены также повышает реалистичность сцены. В качестве фона окружающей обстановки рассматривается множество сильно удаленных от наблюдателя объектов, то есть объектов, расстояние до которых велико по сравнению с перемещениями наблюдателя в течение всего времени процесса работы модели. К такому фону могут относиться, например, небо с облаками и солнечным диском, удаленные горы или здания.

Простейший способ моделирования фона заключается в использовании цилиндрической карты среды, в которой одно изображение (например, неба), заикленное по горизонтали, накладывается на цилиндр, расположенный вокруг наблюдателя. Именно такой метод использован в морском тренажере (см. рис. ???). Однако «верх» цилиндра остается пустым, поэтому этот метод можно использовать только в таких симуляторах, где наблюдатель не может смотреть вверх (например, в автомобильных). Другим методом является создание сферической карты среды, при котором текстура накладывается на полусферу, аппроксимируемую полигонами. Любопытно, что эта аппроксимация может быть очень грубой, например можно аппроксимировать сферу кубом, реализовав так называемый метод кубической карты среды (skybox). Его реализация включает предварительное создание кубических карт среды и затем их визуализацию в специальном режиме. Кубическая карта среды образуется из шести квадратных изображений, которые накладываются на соответствующие грани куба, центр которого размещается в точке наблюдения.

Шесть изображений для кубической карты представляют собой перспективные проекции окружающей фоновой обстановки с полем зрения 90 градусов по базовым осям  $\pm X$ ,  $\pm Y$ ,  $\pm Z$  – соответственно виды справа (Right), слева (Left), сверху (Up), снизу (Down), сзади (Back) и спереди (Front) (см. рис 10).

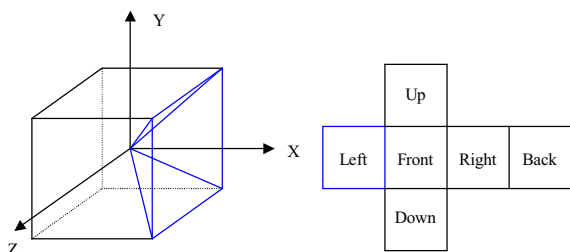


Рисунок 10: Схема развертки кубической карты.

На рис. 11 показан пример изображений для кубической карты. Использование кубической карты среды заключается в визуализации шести полигонов, образующих куб, так, что его центр располагается в точке наблюдения и наложения на эти полигоны соответствующих текстур кубической карты. На рис. 12 представлены изображения из разных точек зрения, полученные при использовании карты среды.

Использование метода кубических карт среды для отображения фона сцены позволяет существенно повысить реализм синтезируемых виртуальных сцен без привнесения

значительных издержек при его использовании. Кроме того, в процессе полета

можно направлять камеру во все стороны (в частности, вверх), а также выполнять на моделируемом летательном аппарате перевороты, не опасаясь выйти за пределы формируемого изображения.

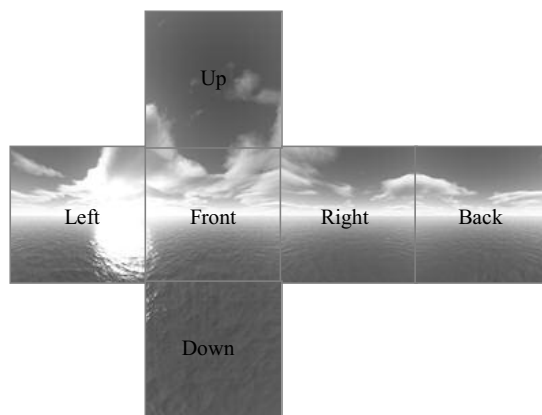


Рисунок 11: Развертка кубической карты среды.



Рисунок 12: Примеры реализации кубической карты среды.

## 6. МОДЕЛИРОВАНИЕ ЛАНДШАФТА БЛИЖНЕГО ПЛАНА.

Кроме отдельных объектов в видеотренажерах возникает необходимость имитации местности, окружающей тренируемого, в частности моделировании ландшафта ближнего плана. Примерами ландшафта ближнего плана могут служить береговая линия, лесополоса, а также любая другая местность, окружающая наблюдателя при движении по определенной траектории. И, если в некоторых случаях можно ограничиться плоским моделированием окружающей местности, как, например, лесополоса вдоль дороги, то задачи имитации сильно пересеченной и гористой местности являются более сложными. Основными характеристиками компьютерной модели имитируемого рельефа должны быть ее объемность и максимальная реалистичность. Такие требования связаны с тем, что моделируемая местность, как правило, служит основным ориентиром для тренируемого.

Исходными данными для моделирования рельефа местности служат топографические карты, в которых линиями уровня показаны изменения высоты рельефа. На основании этих линий строятся отдельные элементы ландшафта, которые можно смоделировать k-уровневыми усеченными пирамидами с n-угольными основаниями, искривив их в соответствии с линиями уровня. Затем эти элементы соединяются в сплошной рельеф и удаляются те грани, которые заведомо не попадут в поле зрения наблюдателя (нижние основания и задние грани пирамид). Для каждого элемента выбирается подходящая текстурная карта и способ ее наложения, соответствующий степени изменения рельефа. Если текстурное покрытие рельефа однородно и не содержит каких-либо ярко выраженных особенностей, можно использовать одну и ту же

текстурную карту, накладывая ее мозаичным образом в горизонтальном и вертикальном направлениях для всех фрагментов рельефа. Проецирование текстурной карты на отдельные участки может производиться несколькими различными способами. Для высоких, вытянутых вверх объектов типа отдельных гор приемлемым является цилиндрическое проецирование. Для низких, приземистых холмов лучшие результаты даст планарное проецирование текстурной карты сверху вниз. Для остальных объектов возможно выделение групп граней и осуществление бокового планарного проецирования отдельно на каждую группу.

На рисунке 13 показана каркасная модель участка береговой линии, а на рисунке 14 - эта же модель, обтянутая текстурой. При этом была использована одна и та же текстурная карта.

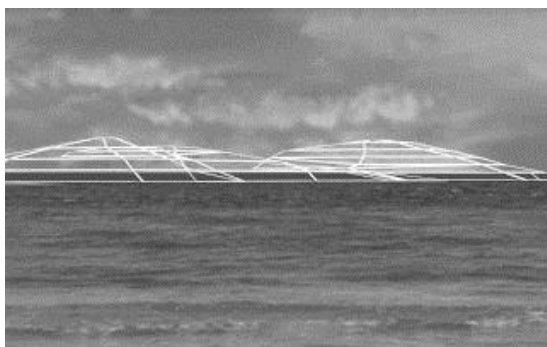


Рисунок 13: Каркасная модель.



Рисунок 14: Модель с текстурой.

## 7. МОДЕЛИРОВАНИЕ ЛАНДШАФТА ШИРОКОГО ОБЗОРА.

Под ландшафтами широкого обзора понимаются большие участки местности, наблюдаемые с не очень большой высоты (например, при полете на вертолете) так, что детали рельефа достаточно хорошо просматриваются. Это означает, что нельзя заменить карту рельефа плоской текстурой, как при полете самолетов на большой высоте. Для создания ландшафтов широкого обзора используется сетка высот. Эта сетка может быть получена на основе съемки и оцифровки реального участка местности или сформирована случайным образом компьютерной программой для моделирования незнакомой местности. В последнем случае необходимо сформировать и задать в качестве входа некоторые параметры формируемой местности, такие, как тип местности, максимальную и минимальную высоты, степень перепада высот близлежащих точек и т.д. Поверхность ландшафта формируется на основе сетки высот с помощью естественной триангуляции. Для осуществления

динамической подкачки вся сетка разбивается на квадратные участки фиксированной длины, а для увеличения скорости визуализации (обеспечения нужной частоты кадров) множество полигонов каждого квадрата задается в виде одного стрипа, который хранится в системной структуре графической системы (дисплейном списке).

Очевидно, что чем более густая сетка наложена на участок, тем выше качество полученной модели. С другой стороны тем большее число полигонов придется визуализировать и в конце концов графическая система может просто не успевать производить визуализацию в реальном режиме времени. Кроме того, если участок местности представляет собой равнину или близкую к ней поверхность, то нет смысла задавать ее большим числом мелких треугольников, а достаточно разбить квадрат всего на два больших треугольника. Отсюда возникает идея заменить, где это возможно, маленькие треугольники на большие, снизив тем самым нагрузку на систему визуализации и повысив ее скорость. До какой степени объединять треугольники – определяется уровнем детализации. При визуализации в каждом кадре определяется оптимальный уровень детализации всех вершин полигонов и производится объединение полигонов малого размера в более крупные. При этом последовательно используются два критерия оптимизации. Первый критерий основан на том, что для удаленных треугольников их горизонтальный размер может быть меньше пиксельного разрешения экрана. Поэтому пару таких соседних треугольников одного уровня можно склеить в один треугольник (без потери качества изображения) и рекурсивно продолжать этот процесс, увеличивая размеры и уменьшая общее число треугольников. Второй критерий основан на объединении двух соседних треугольников, которые не сильно выходят за пределы одной плоскости. Этот критерий очень важен при триангуляции плоской поверхности, для которой достаточно задать большие треугольники вместо множества маленьких. В соответствии с этим критерием два треугольника объединяются, если их общая вершина изменяет свою высоту после объединения не больше, чем на  $\epsilon(h, l)$ , где  $\epsilon(h, l)$  - некоторая функция, зависящая от высоты  $h$  и дальности  $l$  точки наблюдения.

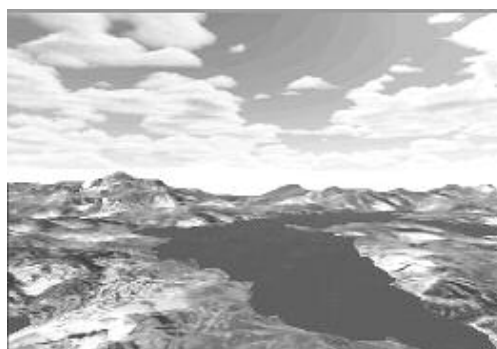


Рисунок 15: Ландшафт с одной текстурной картой .

При этом очень важно, что множество полученных треугольников можно записать в виде одного стрипа, что позволяет добиться ускорения визуализации в случае использования систем, поддерживающих обработку стрипов. Естественно, что при объединении треугольников, примыкающих к границам квадрата, необходимо производить соответствующие изменения и в соседнем квадрате. При моделировании реальных участков местности, конечно, необходимо иметь текстурные карты этих участков. Искусственные же участки местности можно



закрашивать различным цветом в зависимости от высоты и угла наклона поверхности. Точки, расположенные выше некоторого уровня, а также склоны крутых гор закрашиваются цветом камня, остальные - цветом травы. На рис. 15 показан пример такого ландшафта, в котором также использована лишь одна текстурная карта.

Для визуализации достаточно больших площадей (например, в вертолетных тренажерах) надо обеспечить динамическую подкачку участков местности в пирамиде видимости и удаление из памяти участков, не попадающих в поле зрения оператора тренажера.

## 8. ДВИЖЕНИЕ УПРАВЛЯЕМЫХ ОБЪЕКТОВ.

Если моделирующий комплекс и система визуализации работают на разных компьютерах, то для связи используется подсистема сетевого взаимодействия. Она включает программные средства приема по сети и обработки новых значений параметров (положения, углы поворота, скорости и т.д.) управляемых объектов. Моделирующий комплекс производит расчет изменения параметров модели при изменении положения ручек управления и новые значения параметров передаются по сети в комплекс визуализации. Управляющий комплекс с достаточно высокой частотой посылает в систему визуализации информационные пакеты, содержащие полную информацию об изменениях положений, ориентаций и других параметров объектов сцены. Получая эти пакеты, система визуализации использует содержащуюся в них информацию для визуализации очередных кадров. При этом часто возникает такая ситуация, что пора визуализировать следующий кадр, а информация для него еще не пришла из моделирующего комплекса. Это может быть связано со сложными расчетами математической модели или задержками в сети. В этом случае приходится экстраполировать необходимые параметры на основе их предыдущих значений чтобы сгладить «дергания» объектов, вызванные неравномерностью поступлений информационных пакетов.

Видеотренажер может иметь несколько видеоканалов. Каждый из них соответствует отдельному окну наблюдения (например, переднее, правое и левое окна самолета) и представлен отдельным монитором, подключенным к отдельному компьютеру. В этом случае одной из основных проблем является синхронизация, которая позволила бы на нескольких компьютерах одновременно представлять разные части одной и той же сцены (например, моделировать вид сцены из фронтального и боковых стекол летящего самолета).

Проблема шивки и синхронизации нескольких изображений отпадает, если использовать технологию вывода изображения на сферический экран. Это позволяет получать изображение с углом обзора до 180 градусов в обоих направлениях из одного источника видео. Для такой технологии нужен сферический экран, сферическое проецирование и специальное программное обеспечение сферической коррекции изображения.

## 9. ДВИЖЕНИЕ НЕУПРАВЛЯЕМЫХ ОБЪЕКТОВ.

Кроме тех объектов, которыми управляет тренирующийся, в сцене могут быть и другие движущиеся объекты. Например,

в автомобильном тренажере это могут быть встречные или попутные машины, в космических тренажерах – Земля, планеты и так далее. Конечно движение таких объектов можно сделать управляемым от вычислительного комплекса в соответствии с некоторой математической моделью, однако часто их движение заранее определено и управление ими включается в систему визуализации. Например, Земля вращается вокруг своей оси с определенной скоростью и нет смысла для каждого кадра передавать ее очередные углы ориентации. Можно передать эту скорость один раз и система визуализации будет использовать ее по умолчанию все время тренировки. Часто, однако, движение неуправляемых объектов более сложное и мы вынуждены задавать траекторию их движения. Задавать параметры этой траектории для каждого кадра слишком накладно, поэтому в системах визуализации используется ключевая анимация. В ней параметры объекта задаются для некоторого числа ключевых кадров, а во всех промежуточных кадрах система визуализации производит расчет их значений с помощью интерполяции. Рассмотрим этот вопрос на примере координат объекта. Пусть в двух соседних ключевых кадрах объект находился в точках  $P_1$  и  $P_2$  и имел векторы скоростей, соответственно,  $V_1$  и  $V_2$ . Если рассматривать параметрическое представление участка траектории между точками  $P_1$  и  $P_2$ , то эти точки соответствуют значениям параметра  $t = 0$  и  $t = 1$ . Для определения положения объекта в момент времени  $t$  используется эрмитов сплайн

$$H(t) = (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_2 + (t^3 - 2t^2 + t)V_1 + (t^3 - t^2)V_2$$

Легко проверить, что  $H(0) = P_1$ ,  $H(1) = P_2$ ,  $H'(0) = V_1$ ,  $H'(1) = V_2$ , то есть эрмитов сплайн проходит через заданные точки  $P_1$  и  $P_2$  и имеет в них заданные скорости  $V_1$  и  $V_2$ . Обычно в ключевых кадрах задаются только положения  $P_1$  (координаты) объекта, а его скорость в каждой точке  $P_1$  надо вычислять по этим координатам. В процедуру этого вычисления вводится ряд параметров, а именно натяжение (tension), непрерывность (continuity) и общий наклон (bias) вектора скорости при прохождении точки  $P_1$ . Получаемый при этом сплайн носит название ТСВ-сплайна или сплайна Кочанека-Бартельса по именам его авторов.

Кроме положения объекта важным параметром является его ориентация в трехмерном пространстве. Ориентацию можно трактовать как поворот объекта (без перемещения) из некоторого стандартного положения. Поворот можно задать с помощью матрицы  $3 \times 3$  или с помощью углов Эйлера. Первый способ требует задания 9 значений, а второй является неоднозначным, так как двум разным наборам углов Эйлера могут соответствовать одинаковые ориентации. Это вносит большие сложности при интерполяции ориентаций, заданных углами Эйлера. Лучшим способом задания ориентаций является использование кватернионов. Не вдаваясь в подробности, скажем лишь, что для кватерниона  $q = (\cos\phi, v\sin\phi)$  преобразование  $R(u) = quq^{-1}$  трехмерного пространства  $R^3$  ( $u, v \in R^3$ ) является линейным и осуществляет поворот пространства вокруг вектора  $v$  на угол  $2\phi$ . Таким образом ориентации объекта в ключевых кадрах можно задавать с помощью кватернионов (4 числа для каждого кадра). Однако вычисление значений кватернионов для промежуточных кадров является более сложной задачей, чем вычисление координат объекта и требует использования интерполяции на четырехмерной единичной сфере (сферической линейной интерполяции).

С помощью ключевой анимации можно выполнять морфинг объектов, изменять свойства их материалов, менять параметры настройки камер и так далее. Все это делается аналогично интерполяции координат объекта.

## 10. МОДЕЛИРОВАНИЕ ПРИБОРОВ.

Естественно, в любом моделируемом комплексе имеется достаточно много приборов, по которым оператор следит за состоянием комплекса, параметрами происходящих процессов, проводит ориентировку в пространстве и во времени и т.д. В тренажерных комплексах многие из этих приборов моделируются физически на рабочем месте оператора и управляются от моделирующего комплекса. Например, это могут быть спидометры, показатели высоты полета, датчики давления и т.д. Однако, если прибор как-то связан с визуальной обстановкой (которая является виртуальной), то его также приходится моделировать в трехмерной сцене. Такими приборами могут быть локаторы, эхолоты, системы ориентации и т.д. Рассмотрим, например, оптический визир ВСК на космическом аппарате. Прибор ВСК обеспечивает контроль ориентации космического аппарата за счет наличия в нем центральной зрительной трубы и восьми зрительных трубок, составляющих периферийную систему. В системе визуализации центральная зрительная труба моделируется камерой, которая во время ориентации направлена к центру Земли (а во время стыковки – параллельно поверхности Земли), а периферийная система – восемью другими камерами, оси которых расположены на поверхности конуса, построенного на центральной оси. Их направление выбрано так, что во время ориентировки в них видны края Земли. Таким образом, по расположению границ Земли в периферийных окнах можно судить о том, насколько правильно сориентирован космический аппарат. При правильной ориентации аппарата граница Земли проходит через середину каждого из периферийных окон (см. рис. 16).

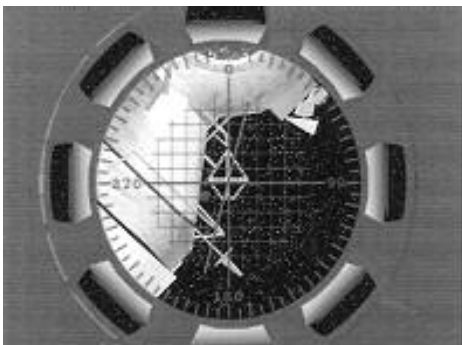


Рисунок 16: Модель прибора ВСК.

Таким образом, для формирования изображения, видимого через этот прибор, необходимо использовать девять камер, жестко связанных с моделью управляемого космического модуля. Каждая из них выводит изображение в свою область вывода экрана (Viewport). Так как каждое из окон не является прямоугольным, то приходится дополнительно использовать маски в буфере трафарета. Для контроля орбитальной ориентации используется и другие приборы. Например, визир широкоугольный с точной вертикалью (ВШТВ) содержит широкоугольный объектив с углом поля зрения 185°. Нам трудно представить, как будет выглядеть изображение в таком объективе. Это так называемый «рыбий глаз», т.е. считается, что так видит рыба, вытщенная из воды. Промоделировать такой объектив средствами стандартных камер визуальных систем невозможно. Поэтому используется математическая модель, по которой трехмерная сцена отображается через такой объектив на выделенную область экрана.

## 11. ЗАКЛЮЧЕНИЕ.

Мы рассмотрели лишь некоторые основные составляющие использования компьютерной графики в видеотренажерах. Конечно, эта область стремительно развивается вместе с развитием вычислительной техники, совершенствованием графических методов и алгоритмов, созданием новых графических библиотек, ростом аппаратной поддержки графических функций, развитием новых взглядов на обучение и приобретение навыков персонала. Исследования в области виртуальной реальности, нейронных сетей, искусственного интеллекта несомненно, будут предъявлять новые требования к системам визуализации в видеотренажерных комплексах. Разработка новых методов представления и моделирования трехмерного пространства, например таких, как задание объектов поверхностями второго порядка, представление внутренней структуры объектов с помощью трехмерных текстурных (воксельных) карт, интерфейсы взаимодействия объектов сцены уже сейчас ставят целую серию новых сложных задач перед исследователями в области компьютерной графики.

## 12. ЛИТЕРАТУРА.

- [1] Долговесов Б.С. Семейство компьютерных систем визуализации Альбатрос // Автометрия, № 6, 1994.
- [2] Dolgovesov B.S. et al. ARIUS – Family of real-time 3D image generation for PC platform // JANES SIMULATION AND TRAINING SYSTEMS, Editor I.W.Strachham MBE AFC FRAs, Bentworth, ALTON, Hampshire, England, 1998.
- [3] Михайлюк М.В., Решетников В.Н. Визуализация трехмерных сцен в реальном режиме времени. Программные продукты и системы № 1, 1999.
- [4] Михайлюк М.В. Основы компьютерной графики. М. 2001.
- [5] Роганов В.Р. Методы формирования виртуальной реальности., Пенза, ПензГУ, 2002.
- [6] A.Kapustin, V.Eroukhimov, A.Malashkina, The implementation of the real-time terrain visualization algorithm, GraphiCon '99, Conference Proceedings, 1999.
- [7] Н.А.Елыков, И.В.Белаго, М.М.Лаврентьев, Ю.Ю.Некрасов. Генерация изображения морской поверхности в реальном времени, Труды конференции Графика'2000, М.
- [8] Тезисы 4-й международной научно-практической конференции «Пилотируемые полеты в космос», Звездный городок, март 2000.
- [9] Материалы научно-технической конференции «Тренажерные технологии и симуляторы 2002», СПб, 2002.

## Об авторе

Михайлюк Михаил Васильевич, д.ф.м.н., профессор, заведующий отделом компьютерных средств визуализации в Центре визуализации и спутниковых информационных технологий Института микропроцессорных вычислительных систем РАН.

E-mail: [mix@mail.imvs.ru](mailto:mix@mail.imvs.ru)