Хорев Сергей Анатольевич

3-D АКСЕЛЕРАТОРЫ.

Что это такое и для чего нужно?

Видеоадаптер, вероятно, является компонентом персонального компьютера, вызывающим наибольшее количество вопросов. И действительно, почему один видеоадаптер стоит тридцать, а другой триста долларов? Почему видеоадаптеры, изготовленные из одинаковых микросхем, могут различаться в стоимости на 30, а то и на 50 %?

Разумно начать с того, что вспомнить,

С ЧЕГО ВСЕ НАЧИНАЛОСЬ, И КАКИМИ БЫЛИ ПЕРВЫЕ ВИДЕОАДАПТЕРЫ.

Видеоадаптер предназначен для того, чтобы преобразовывать информацию, хранящуюся и обрабатываемую компьютером, в визуальную форму, пригодную для воспроизведения на мониторе. Таким образом, комплект видеоадаптер-монитор, по сути дела, представляет интерфейс от машины к человеку. А поскольку более 70% информации мы воспринимаем визуально, от качества этого интерфейса в значительной степени зависит наше восприятие компьютера в целом. Именно поэтому производители компьютеров такое внимание уделяют видеоподсистеме.

Первые компьютеры имели видеоподсистемы, структурная схема которых представлена на рис.1. Надо сказать, за годы существования персональных компьютеров эта схема радикально не изменилась. Однако первоначально весь экран представлялся в виде набора цветных точек-пикселей, и вывод изображения состоял в непосредственной записи в видеопамять цвета точки в ячейку с ее координатами. Такой подход позволял использовать в качестве видеопроцессора относительно простую, а следовательно, недорогую микросхему и оправдывал себя до тех пор, пока основной операционной системой была MS-DOS или ее аналоги.

С ростом популярности Windows возникла потребность в специализирован-

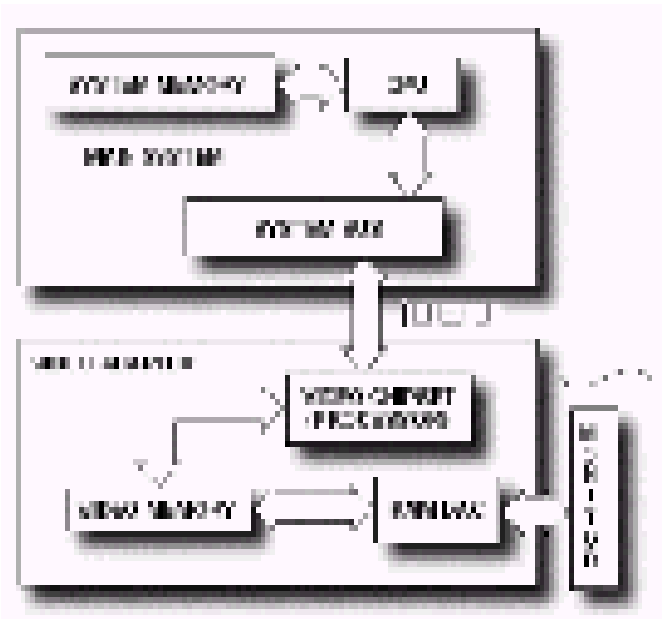

Рисунок 1.
Видеоподсистема персонального компьютера.

ных двумерных или Windows-акселераторах.

ЗАЧЕМ НУЖНЫ ЭТИ WINDOWS-АКСЕЛЕРАТОРЫ?

Непосредственное программирование экрана не могло обеспечить растущих потребностей графической оболочки. Легко представить, сколько операций записи точки необходимо произвести центральному процессору, чтобы нарисовать просто белое окно. Производители видеоадаптеров добавили в свои продукты функции обработки двумерной графики, такие, как прорисовка окон при открытии и свертывании, закраска областей экрана, перемещение областей и аппаратная реализация курсора. Так появилось семейство видеокарт, называемых Windows или GUI (Graphical User Interface) акселераторами, которые стали непременными атрибутами современных компьютеров. Впоследствии эти акселераторы непрерывно совершенствовались. Они приобретали новые функции, например, способность работы с True Type шрифтами, аппаратный скроллинг и т.д.

Внедрение мультимедиа создало новые проблемы, вызванные добавлением таких компонентов, как звук и цифровое видео, к набору двумерных графических функций. Сегодня большинство видеоадаптеров поддерживают на аппаратном уровне обработку цифрового видео. Мультимедиа акселератор обычно имеет встроенные аппаратные функции, позволяющие масштабировать видеоизображение по осям x и y, а также аппаратно преобразовывать цифровой сигнал в аналоговый для вывода его на монитор в формате RGB. Мультимедиа акселераторы могут также иметь встроенные возможности декомпрессии цифрового видео (MPEG-1).

Параллельно с увеличением быстродействия и количества функций росло разрешение мультимедиа акселераторов. Первичный стандарт VGA составлял 640×480 пикселей, сегодня же не существует монитора, поддерживающего разрешение менее 1024×768 пикселей, поэтому и видеокарт с меньшим разрешением не выпускается. Что касается видеопамяти, то для этих графических карт легко посчитать необходимый размер: (разрешение по горизонтали)×(разрешение по вертикали)×(глубину цвета). Например: 1024×768×3(байта)= 2359296 байт, то есть для получения разрешения 1024×768 пикселей при 16777216 цветах требуется 2359296 байт видеопамяти. На практике это означает, что такое разрешение обеспечивают мультимедиа акселераторы с видеопамятью 4 МБ.

Видеоконтроллеры, на базе которых выпускались видеоадаптеры, становились все совершеннее, все быстрее приобретали новые возможности; например, S3-Trio 64V+ фирмы S3 может работать с тремя независимыми потоками видеоданных: данные от центрального процессора, цифровое видео и аппаратный курсор. Потенциал, заложенный в эту микросхему, столь высок, что видеоадаптеры на ее основе до сих пор пользуются спросом. Более того, встречаются даже видеоадаптеры на микросхемах других фирм, которые “делают вид”, что это S3-Trio 64V+. Мне кажется, что это высшее признание.

Однако эта почти идиллическая и уж во всяком случае явно эволюционная картина была буквально взорвана примерно полтора года назад появлением 3-D.

ЧТО ЖЕ ТАКОЕ 3-D?

3-D, как следует из самого названия, это изображение трехмерных объектов на экране монитора, скорее, даже не просто отдельных объектов, а реального, или, точнее, виртуального мира. Зачем это нужно? Апологеты 3-D графики говорят о преимуществах, которые дает перенос приложений с высокопроизводительных рабочих станций типа Silicon Graphics на PC платформу. Говорят, что интернет-приложения сильно выиграют от невероятной маневренности, интуитивности и гибкости, которые обеспечивает применение трехмерного графического интерфейса.

Взаимодействие в World Wide Web будет гораздо проще и удобнее, если будет происходить в трехмерном пространстве. Однако даже самые ярые энтузиасты признают, что развлекательные приложения - главный потребитель и заказчик ускорителей 3-D графики. Развлекательные приложения - это прежде всего и в основном игры, значит, если мы не собираемся играть, нам 3-D акселератор не нужен? Значит, не нужен. Однако не следует забывать, что 3-D графика - это уже технология. Можно работать с текстами в редакторе WordPad, однако большинство предпочитает Word. Это тоже технология. Ну, а цена не столь велика. Если мне не изменяет память, то самая покупаемая сегодня видеокарта - 3-D акселератор на микросхеме S3-ViRGE с 2МБ памяти стоит около 35 долларов или даже дешевле. Ею оснащаются большинство компьютеров начального уровня.

РОЛЬ API

3-D графика не смогла бы стать технологией, если бы не было разнообразного инструментария, позволяющего с ней работать, а для изображения трехмерных объектов на экране монитора требуется проведение серии процессов (обычно называемых конвейером). Программный интерфейс приложений (API) как раз и состоит из функций, управляющих 3D конвейером на программном уровне, но при этом может использовать преимущества аппаратной реализации 3D в случае наличия этой возможности. Если имеется аппаратный акселератор, API использует его преимущества, если нет, то API работает с оптимальными настройками, рассчитанными на самые обычные системы. Таким образом, благодаря применению API, любое количество программных средств может поддерживаться любым количеством аппаратных 3D акселераторов.

Для приложений общего и развлекательного направления существуют следующие API:

- Microsoft Direct3D
- Criterion Renderware
- Argonaut BRender
- Intel 3DR

Компания Apple продвигает свой собственный интерфейс Rave, созданный на основе их собственного API Quickdraw 3D.

Для профессиональных приложений, работающих под управлением WindowsNT, доминирует интерфейс OpenGL. Компания Autodesk, крупнейший производитель инженерных приложений, разработала свой собственный API, называемый Heidi. Свои API разработали и такие компании, как Intergraph - RenderGL и 3DFX - GLide.

Существование и доступность 3D интерфейсов, поддерживающих множество графических подсистем и приложений, в свою очередь, увеличивает потребность в аппаратных ускорителях трехмерной графики, работающих в режиме реального времени.

Итак, как же строится 3-D изображение?

МОДЕЛЬ

Как мы уже знаем, для изображения трехмерных объектов требуется выполнение ряда операций с преобразованием результата в двумерный вид. Первоначально, объект представляется в виде набора точек или координат в трехмерном пространстве. Трехмерная система координат определяется тремя осями: горизонтальной, вертикальной и глубины, обычно называемых, соответственно осями x, y и z. Объектом может быть дом, человек, машина, самолет или целый 3D мир, и координаты определяют положение вершин (узловых точек), из которых состоит объект, в пространстве.

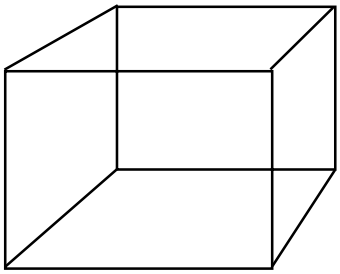

Рисунок 2.
Каркасная модель куба.

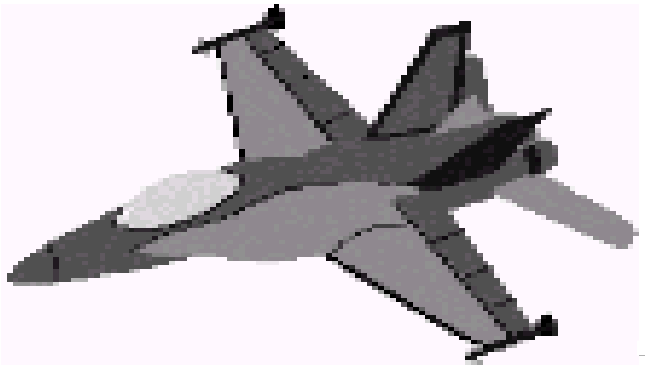

Рисунок 3.
Модель самолета с закрашенными поверхностями.

Соединив вершины объекта линиями, мы получим каркасную модель, называемую так из-за того, что видимыми являются только края поверхностей трехмерного тела. Каркасная модель определяет области, составляющие поверхности объекта, которые могут быть заполнены цветом, текстурами и освещаться лучами света. Даже при таком упрощенном объяснении конвейера 3D графики становится ясно, как много требуется вычислений для прорисовки трехмерного объекта на двумерном экране. Можно представить, насколько увеличивается объем требуемых вычислений над системой координат, если объект движется. Именно этот огромный объем вычислений и обусловил появление специализированных 3D акселераторов.

ТЕХНОЛОГИЯ 3D-ГРАФИКИ.

Часть вычислительных операций, связанных с отображением и моделированием трехмерного мира, переложено на 3D-акселератор, который является сердцем 3D-видеокарты. Центральный процессор теперь практически не занят вопросами отображения, образ экрана формирует видеокарта. В основе этого процесса лежит реализация на аппаратном уровне ряда эффектов, а также применение математического аппарата. Попробуем разобраться, что же конкретно умеет графический 3D-процессор.

В качестве примера рассмотрим гипотетический симулятор автогонок и на его примере проиллюстрируем, какие усилия предпринимаются для создания реалистичности отображения поверхностей дороги или зданий, стоящих на обочине, и т.д.

Для этого применяется распространенный метод, называемый текстурирование (texture mapping). Это самый распространенный эффект для моделирования поверхностей. Например, фасад здания потребовал бы отображения множества граней для моделирования множества кирпичей, окон и дверей. Однако текстура (изображение, накладываемое на всю поверхность сразу) дает больше реализма, но требует меньше вычислительных ресурсов, так как позволяет оперировать со всем фасадом как с единой поверхностью. Перед тем, как поверхности попадают на экран, они текстурируются и затеняются. Все текстуры хранятся в памяти, обычно установленной на видеокарте. Кстати, здесь нельзя не заметить, что применение ускоренного графического порта AGP (Accelerated Graphics Port) делает возможным хранение текстур в системной памяти, а ее объем гораздо больше.

Очевидно, что, когда поверхности текстурируются, необходим учет перспективы, например, при отображении дороги с разделительной полосой, уходящей за горизонт. Перспективная коррекция необходима для того, чтобы текстурированные объекты выглядели правильно. Она гарантирует, что битмэп правильно наложится на разные части объекта – и на те, которые ближе к наблюдателю, и на более далекие.

Коррекция с учетом перспективы - очень трудоемкая операция, поэтому иногда можно встретить не совсем верную ее реализацию.

При наложении текстур, в принципе, также можно увидеть швы между двумя ближайшими битмэпами. Или, что бывает чаще, в некоторых играх при изображении дороги или длинных коридоров

заметно мерцание во время движения. Для подавления этих трудностей применяется фильтрация (обычно Bi- или tri-линейная).

Билинейная фильтрация - метод устранения искажений изображения. При медленном вращении или движении объекта могут быть заметны перескакивания пикселей с одного места на другое, что и вызывает мерцание. Для снижения этого эффекта при билинейной фильтрации для отображения точки поверхности берется взвешенное среднее четырех смежных текстурных пикселей.

Трилинейная фильтрация несколько сложнее. Для получения каждого пикселя изображения берется взвешенное среднее значение результатов двух уровней билинейной фильтрации. Полученное изображение будет еще более четкое и менее мерцающее.

Текстуры, с помощью которых формируется поверхность объекта, изменяют свой вид в зависимости от изменения расстояния от объекта до положения глаз зрителя. При движущемся изображении, например, по мере того, как объект удаляется от зрителя, текстурный битмэп должен уменьшаться в размерах вместе с уменьшением размера отображаемого объекта. Для того, чтобы выполнить это преобразование, графический процессор преобразует битмэпы текстур вплоть до соответствующего размера для покрытия поверхности объекта, но при этом изображение должно оставаться естественным, то есть объект не должен деформироваться непредвиденным образом.

Для того чтобы избежать непредвиденных изменений, большинство управляющих графикой процессов создают серии предфильтрованных битмэпов - текстур с уменьшенным разрешением, этот процесс называется mip mapping. Затем графическая программа автоматически определяет, какую текстуру использовать, основываясь на деталях изображения, которое уже выведено на экран. Соответственно, если объект уменьшается в размерах, размер его текстурного битмэпа тоже уменьшается.

Но вернемся в наш гоночный автомобиль. Сама дорога уже выглядит реалистично, но проблемы наблюдаются с ее краями! Вспомните, как выглядит линия,

Рисунок 4.

Иллюстрация процесса mip mapping. Верхняя строка – соответствующие битмэпы для разного размера объекта. В нижней строке битмэпы увеличены до одного размера для пояснения технологии.

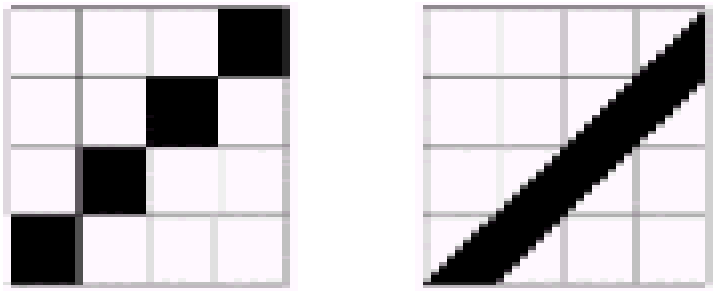

Рисунок 5.
Иллюстрация процесса anti-aliasing. Слева линия имеет рваные края, справа гладкие.

проведенная на экране не параллельно его краю. Вот и у нашей дороги появляются “рваные края”. И для борьбы с этим недостатком изображения применяется anti-aliasing.

Это способ интерполяции пикселей для получения более ровных границ объекта. Наиболее часто используемая техника - создание плавного перехода от цвета линии или края к цвету фона. Цвет точки, лежащей на границе объектов, определяется как среднее цветов двух граничных точек. Однако в некоторых случаях побочным эффектом anti-aliasing является смазывание (blurring) краев.

Мы подходим к ключевому моменту функционирования всех 3D-алгоритмов. Предположим, что трек, по которому ездит наша гоночная машина, окружен большим количеством разнообразных объектов - строений, деревьев, людей. Тут перед 3D-процессором встает главная проблема, как определить, какие из объектов находятся в области видимости и как они освещены. Причем, знать, что видимо в данный момент, недостаточно. Необходимо иметь информацию и о взаимном расположении объектов. Для решения этой задачи применяется метод, называемый z-буферизация. Это самый надежный метод удаления скрытых поверхностей. В так называемом z-буфере хранятся значения глубины всех пикселей (z-координаты). Когда рассчитывается (рендерится) новый пиксель, его глубина сравнивается со значениями, хранимыми в z-буфере, а конкретнее, - с глубинами уже срендеренных пикселей с теми же координатами x и y. Если новый пиксель имеет значение глубины больше какого-либо значения в z-буфере, новый пиксель не записывается в буфер для отображения, если меньше - то записывается. Z-буферизация при аппаратной реализации значительно увеличивает производительность.

Разрешающая способность z-буфера - самый главный его атрибут. Она критична для высококачественного отображения сцен с большой глубиной. Чем выше

Рисунок 6.
Применение техники anti-aliasing к реальной картинке.
Слева - исходная фотография, справа - после обработки.

разрешающая способность, тем выше дискретность z-координат и точнее выполняется рендеринг удаленных объектов. Если при рендеринге разрешающей способности не хватает, то может случиться, что два перекрывающихся объекта получат одну и ту же координату z, в результате аппаратура не будет знать какой объект ближе к наблюдателю, что может вызвать искажение изображения. Для избежания этих эффектов профессиональные платы имеют 32-разрядный z-буфер и оборудуются большими объемами памяти.

Кроме вышеперечисленных основ, трехмерные графические платы обычно имеют возможность воспроизведения некоторого количества дополнительных функций. Например, если бы Вы на своем гоночном автомобиле въехали в песок, то обзор бы затруднился поднявшейся пылью. Для реализации таких и подобных эффектов применяется fogging (затуманивание). Этот эффект образуется за счет комбинирования смешанных компьютерных цветовых пикселей с цветом тумана (fog) под управлением функции, определяющей глубину затуманивания. С помощью этого же алгоритма далеко отстоящие объекты погружаются в дымку, создавая иллюзию расстояния.

Реальный мир состоит из прозрачных, полупрозрачных и непрозрачных объектов. Для учета этого обстоятельства применяется alpha blending - способ передачи информации о прозрачности полупрозрачных объектов. Эффект полупрозрачности создается путем объединения цвета исходного пиксела с пикселом, уже находящимся в буфере. В результате цвет точки является комбинацией цветов переднего и заднего плана. Обычно, коэффициент alpha имеет нормализованное значение от 0 до 1 для каждого цветного пиксела. Новый пиксел = (alpha)(цвет пиксела A) + (1 - alpha)(цвет пиксела B).

Очевидно, что для создания реалистичной картины происходящего на экране необходимо частое обновление его содержимого. При формировании каждого следующего кадра, 3D-акселератор проходит весь путь подсчета заново, поэтому он должен обладать немалым быстродействием. Но в 3D-графике применяются и другие методы придания плавности движению. Ключевой - Double Buffering. Представьте себе старый трюк аниматоров, рисовавших на уголках стопки бумаги персонаж мультика, со слегка изменяемым положением на каждом следующем листе. Пролистав всю стопку, отгибая уголок, мы увидим плавное движение нашего героя. Практически такой же принцип работы имеет и Double Buffering в 3D анимации, то есть следующее положение персонажа уже нарисовано до того, как текущая страница будет пролистана. Без применения двойной буферизации изображение не будет иметь требуемой плавности, то есть будет прерывистым. Для двойной буферизации требуется наличие двух областей, зарезервированных в буфере кадров трехмерной графической платы; обе области должны соответствовать размеру изображения, выводимого на экран. Метод использует два буфера для получения изображения: один для отображения картинки, другой для рендеринга. В то время, как отображается содержимое одного буфера, в другом происходит рендеринг. Когда очередной кадр обработан, буфера переключаются (меняются местами). Таким образом, играющий все время видит устойчивую картинку.

В заключение обсуждения алгоритмов, применяемых в 3D-графических акселераторах, попробуем разобраться, каким же образом применение всех эффектов по отдельности позволяет получить целостную картину. 3D-графика реализуется с помощью многоступенчатого механизма, называемого, как мы помним, конвейером, точнее, конвейером рендеринга.

Применение конвейерной обработки позволяет еще ускорить выполнение расчетов за счет того, что вычисления для следующего объекта могут быть начаты до окончания вычислений предыдущего.

Конвейер рендеринга может быть разделен на 2 стадии: геометрическая обработка и растеризация.

На первой стадии геометрической обработки выполняется преобразование координат (вращение, перенос и масштабирование всех объектов), отсечение невидимых частей объектов, расчет освещения, определение цвета каждой вершины с учетом всех световых источников и процесс деления изображения на более мелкие формы. Для описания характера поверхности объекта она делится на всевозможные многоугольники. Наиболее часто при отображении графических объектов используется деление на треугольники и четырехугольники, так как они легче все-

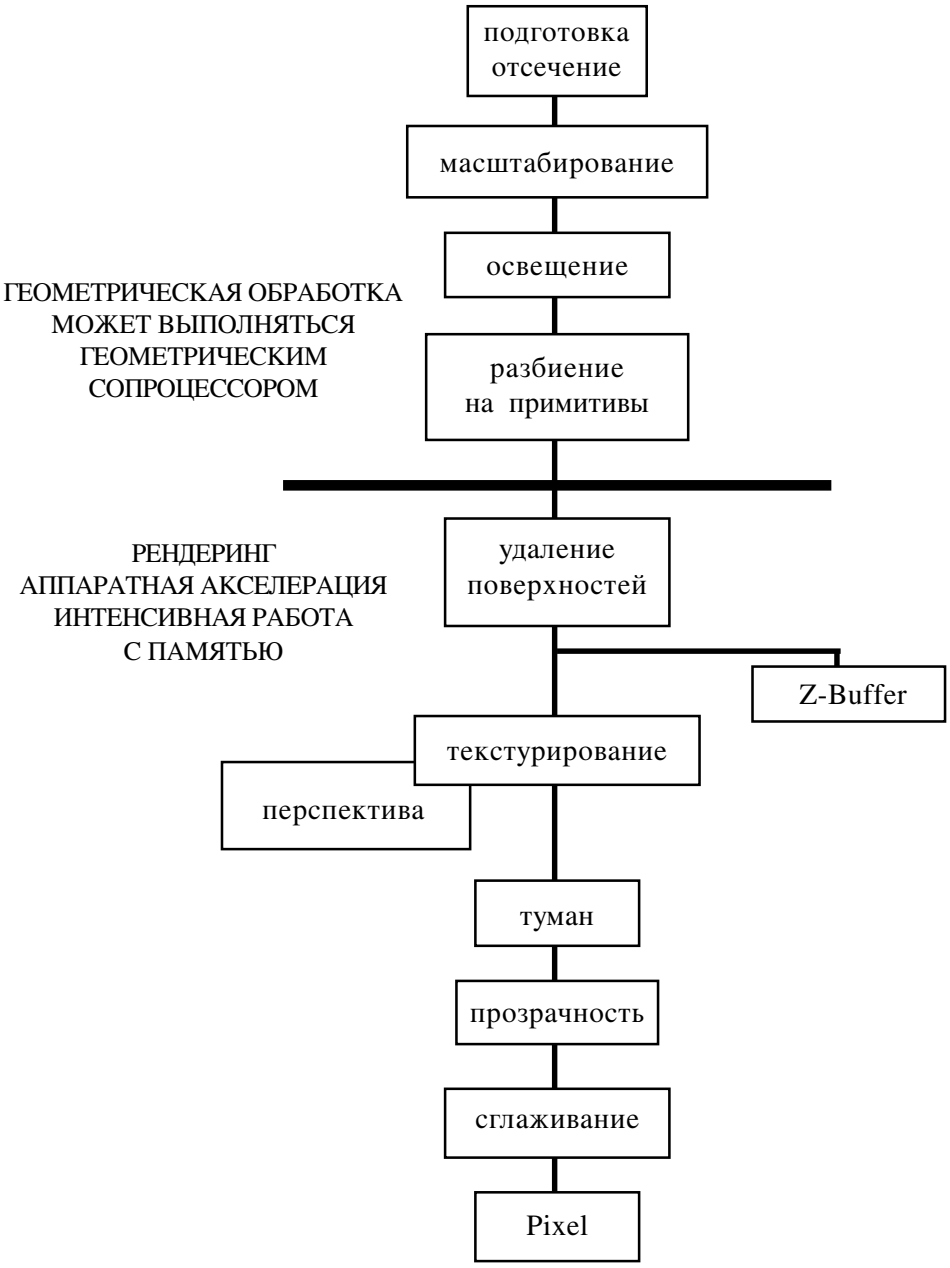

Рисунок 7.
Конвейер рендеринга.

го обсчитываются и ими легко манипулировать. При этом координаты объектов переводятся из вещественного в целочисленное представление для ускорения вычислений.

На второй стадии к изображению применяются все описанные эффекты в следующей последовательности: удаление скрытых поверхностей, наложение с учетом перспективы текстур (используя z-буфер), применение эффектов тумана и полупрозрачности, anti-aliasing. После этого очередная точка считается готовой к помещению в буфер со следующего кадра.

НЕОБХОДИМЫЕ ОБЪЕМЫ ПАМЯТИ.

Значительное усложнение технологии потребовало значительного увеличения объема памяти видеокарты. Если у 2D видеокарт требовалась память только под буфер кадра, для 3D акселераторов необходимы front buffer, back buffer, z-buffer и место под кэш текстур. Front buffer и back buffer равны буферу кадра, z-buffer несколько меньше, а объем кэша текстур определяется программным обеспечением. Для примера возьмем разрешение 640x480 точек при 16 млн. цветов, тогда front buffer и back buffer составляют примерно по 900КБ, z-buffer – примерно 600КБ (если он 16-ти битный), все остальное остается под кэш текстур. Это означает, что для работы в этом режиме 3D акселератору потребуется по крайней мере 4МБ. Так что, встретив где-нибудь в прайслисте 3D акселератор 40МБ памяти, не удивляйтесь. Утешает одно: сегодняшние 4МБ видеокарты стоят столько же, сколько и 1-мегабайтные видеокарты пару лет назад.

ВМЕСТО ЗАКЛЮЧЕНИЯ, ИЛИ ДАВАЙТЕ ПОМЕЧТАЕМ.

Немного придя в себя от обилия информации, попробуем представить, какими будут акселераторы следующего поколения. Лично мне кажется, что за 3-D акселераторами последуют 4-D акселераторы. Добавится новая координата - время. И если сейчас задаются размеры, координаты, тип поверхности и характер освещения, то в будущем можно будет вводить описание поведения объекта. Например, прыгающий мячик будет отскакивать от пола на все меньшую высоту, пока окончательно не опустится на пол, вентилятор будет вращаться, покачиваясь вправо-влево и т.д.

А может быть, все будет и не так. Но в чем вы можете быть уверены, так это в том, что технический прогресс не остановится, и завтра будет что-то новое, о чем мы вам с удовольствием расскажем.

При подготовке статьи использовались материалы фирм Diamond Multimedia (США), Hercules Computer Technology (США), S3 (США), iXBT (Россия).

Хорев Сергей Анатольевич, технический отдел фирмы MicroXperts.

НАШИ АВТОРЫ