Урок 4 Видеосистемы

1. Видеоадаптеры

1.1 Введение

Игра, лишенная качественной графики, имеет мало шансов на успех. Красота картинки значит ничуть не меньше, чем смысл происходящего на экране действа. Разработчики непрерывно борются за первенство в сфере графических технологий, а производители железа их с радостью поддерживают. В результате кадры, потрясавшие наше воображение пару лет назад, по нынешним меркам уже считаются нормой, и кажется, что эта гонка не закончится никогда. Но все же — благодаря чему игровой мир обретает красоту и какова расплата за это?

Современная видеокарта (видеоадаптер) - это, по сути, второй самостоятельный компьютер внутри персонального компьютера. Причем, когда человек играет в любимую 3D-игру, процессор видеокарты фактически выполняет большую часть работы, а центральный процессор, отступает на второй план. То есть, реализуется многопроцессорная архитектура, что, если смотреть в корень, есть отступление от основной идеологии персональных компьютеров PC, где центральный процессор выполняет все и вся. Но это итог того, что человек стал требовать от компьютера не просто умения вычислять, а создавать среду общения, где можно было комфортно работать и отдыхать. Если работать только с офисными пакетами программ, то и по скорости работы разницы между новыми и стрыми моделями видеокарт практически нет. Но все меняется, когда запускается трехмерная игра или программа графического моделирования. Вот тут, не говоря о скорости работы, невооруженным взглядом видно, что современный видеоадаптер создает на экране монитора настолько реалистичное изображение, что не всегда можно поверить, что люди и предметы смоделированы математически, а не скопированы с фотографий. Правда, если поэкспериментировать с различными видеокартами, приглядываясь к мелким деталям на изображении, то можно заметить, что в одном случае картинка нравится, а в другом не очень, тут линии какие-то зазубренные, а здесь ровная поверхность воздушного шарика покрыта некими разводами. Иногда в играх пропадают какие-то детали игрового поля или оружие выглядит совсем по-другому. Причина здесь заключается в том, что видеокарта, являясь периферийным устройством, только по интерфейсу должна полностью соответствовать системной плате и монитору, а вот как внутри нее будет моделироваться трехмерное изображение - это остается прерогативой разработчиков. Более мощный (быстрый и с большим набором функций) графический процессор создаст более реалистическое изображение, с все большим количеством визуальных эффектов

Насколько сильно влияют визуальные эффекты на отображаемую картинку? Для ответа на этот вопрос достаточно запустить одну из современных видеоигр на минимальных настройках качества: игра сразу сбрасывает с себя весь визуальный лоск. Даже с качественными текстурами и высоким разрешением о фотореализме не может быть и речи. Для осознания того, каким образом гадкий утенок превращается в прекрасного лебедя, необходимо углубиться в историю развития видеокарт.

1.2 Немного истории

Одним из первых графических адаптеров для IBM PC компьютеров, стала плата MDA (Monochrome Display Adapter) появившаяся в 1981 году, которая работала только в текстовом режиме с разрешением 25x80 символов (физически 720x350 точек) и имела пять атрибутов текста: обычный, яркий, инверсный, подчёркнутый и мигающий. Никакой цветовой или графической информации он передавать не мог, и то, какого цвета будут буквы, определялось моделью использовавшегося монитора, обычно они были чёрно-белыми, янтарными или изумрудными. В

1982 году Фирма Hercules выпускает дальнейшее развитие адаптера MDA, видеоадаптер HGC (Hercules Graphics Controller - графический адаптер Геркулес), который имел графическое разрешение 720x348 точек и поддерживал две графические страницы. Но он всё ещё не позволял работать с цветом.

Первой цветной графической платой стала CGA (Color Graphics Adapter), выпущенная IBM и ставшая основой для последующих стандартов видеокарт. Она могла работать либо в текстовом режиме с разрешениями 40x25 и 80x25 (матрица символа — 8x8), либо в графическом с разрешениями 320x200 или 640x200. В текстовых режимах доступно 256 атрибутов символа — 16 цветов символа и 16 цветов фона (либо 8 цветов фона и атрибут мигания), в графическом режиме 320x200 было доступно четыре палитры по четыре цвета каждая, режим высокого разрешения 640x200 был монохромным. В развитие этой карты появился EGA (Enhanced Graphics Adapter) — улучшенный графический адаптер, с расширенной до 64 цветов палитрой, и промежуточным буфером. Было улучшено разрешение до 640x350, в результате добавился текстовый режим 80x43 при матрице символа 8x8. Для режима 80x25 использовалась большая матрица — 8x14, одновременно можно было использовать 16 цветов, цветовая палитра была расширена до 64 цветов. Графический режим так же позволял использовать при разрешении 640x350 16 цветов из палитры в 64 цвета. Был совместим с CGA и MDA.

Стоит заметить, что интерфейсы с монитором всех этих типов видеоадаптеров были цифровые, MDA и HGC передавали только светится или не светится точка и ещё дополнительный сигнал яркости для атрибута текста «яркий», аналогично CGA по трём каналам (красный, зелёный, синий) передавал основной видеосигнал, и мог дополнительно передавать сигнал яркости (всего получалось 16 цветов), EGA имел по две линии передачи на каждый из основных цветов, то есть каждый основной цвет мог отображаться с полной яркостью, 2/3, или 1/3 от полной яркости, что и давало в сумме максимум 64 цвета.

В ранних моделях компьютеров от IBM, появляется новый графический адаптер MCGA (Multicolor Graphics Adapter — многоцветный графический адаптер). Текстовое разрешение было поднято до 640x400, что позволило использовать режим 80x50 при матрице 8x8, а для режима 80x25 использовать матрицу 8x16. Количество цветов увеличено до 262144 (64 уровня яркости по каждому цвету), для совместимости с EGA в текстовых режимах была введена таблица цветов, через которую выполнялось преобразование 64-цветного пространства EGA в цветовое пространство MCGA. Появился режим 320x200x256, где каждый пиксел на экране кодировался соответствующим байтом в видеопамяти, никаких битовых плоскостей не было, соответственно с EGA осталась совместимость только по текстовым режимам, совместимость с CGA была полная. Из-за огромного количества яркостей основных цветов возникла необходимость использования уже аналогового цветового сигнала, частота строчной развертки составляла уже 31,5 КГц.

Потом IBM пошла ещё дальше и сделала VGA (Video Graphics Array — графический видео массив), это расширение MCGA совместимое с EGA и введённое в средних моделях. Это фактический стандарт видеоадаптера с конца 80-х годов. Добавлены текстовое разрешение 720x400 для эмуляции MDA и графический режим 640x480, с доступом через битовые плоскости. Режим 640x480 замечателен тем, что в нём используется квадратный пиксел, то есть соотношение числа пикселов по горизонтали и вертикали совпадает со стандартным соотношением сторон экрана — 4:3. Дальше появился IBM 8514/a с разрешениями 640x480x256 и 1024x768x256, и IBM XGA с текстовым режимом 132x25 (1056x400) и увеличенной глубиной цвета (640x480x65K).

С 1991 года появилось понятие SVGA (Super VGA — «сверх» VGA) — расширение VGA с добавлением более высоких режимов и дополнительного сервиса, например возможности поставить произвольную частоту кадров. Число одновременно отображаемых цветов увеличивается до 65'536 (High Color, 16 бит) и 16'777'216 (True Color, 24 бита), появляются дополнительные текстовые режимы. Из сервисных функций появляется поддержка VBE (VESA BIOS Extention — расширение BIOS стандарта VESA). SVGA воспринимается как фактический стандарт видеоадаптера где-то с середины 1992 года, после принятия ассоциацией VESA (Video Electronics Standart Association — ассоциация стандартизации видео-электроники) стандарта VBE версии 1.0. До того момента практически все видеоадаптеры SVGA были несовместимы между собой.

Графический пользовательский интерфейс, появившийся во многих операционных системах, стимулировал новый этап развития видеоадаптеров. Появляется понятие «графический ускоритель» (graphics accelerator). Это видеоадаптеры, которые производят выполнение некоторых графических функций на аппаратном уровне. К числу этих функций относятся, перемещение больших блоков изображения из одного участка экрана в другой (например при перемещении окна), заливка участков изображения, рисование линий, дуг, шрифтов, поддержка аппаратного курсора и т. п. Прямым толчком к развитию столь специализированного устройства явилось то, что графический пользовательский интерфейс несомненно удобен, но его использование требует от центрального процессора немалых вычислительных ресурсов, и современный графический ускоритель как раз и призван снять с него львиную долю вычислений по окончательному выводу изображения на экран.

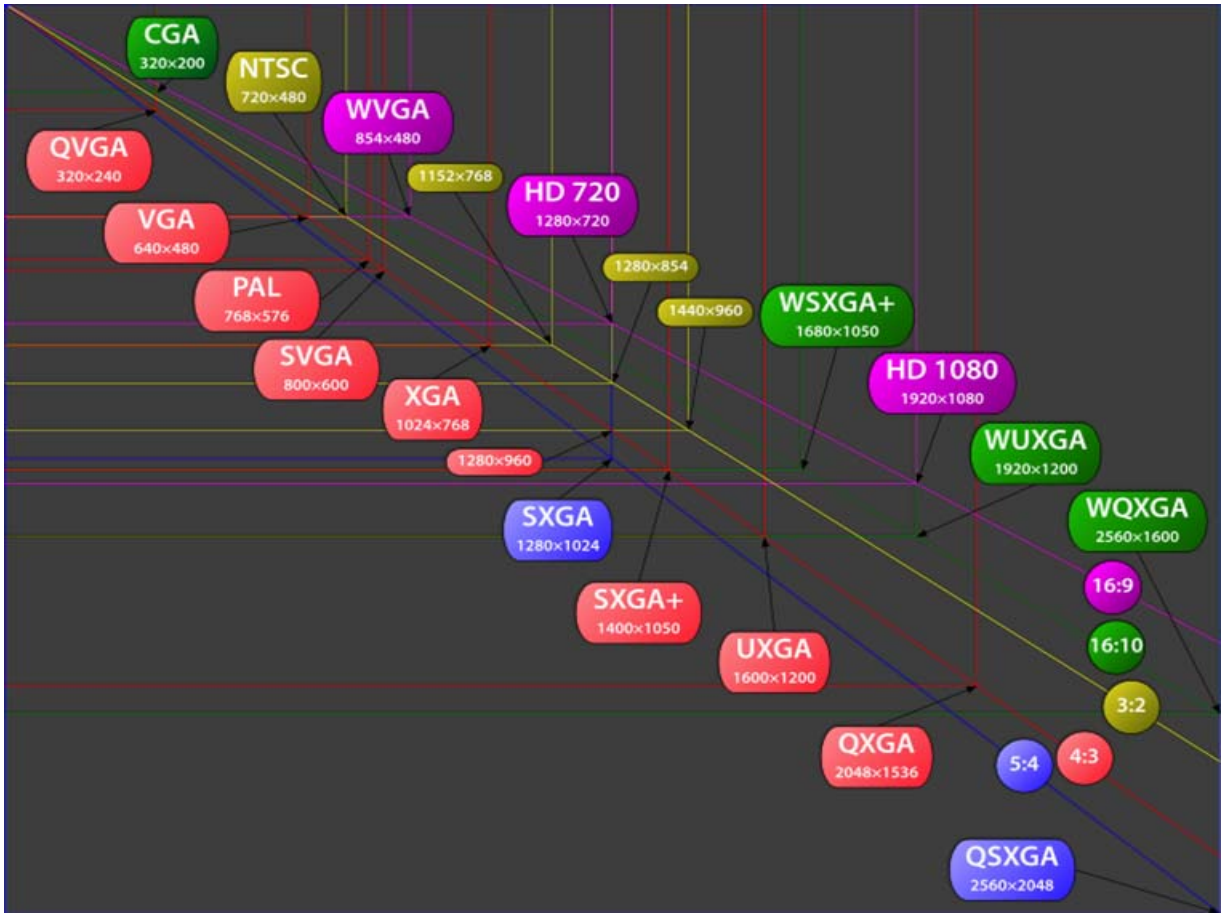


Рисунок – Видеостандарты.

1.3 Основные компоненты.

Так все же давайте попробуем разобрать, из чего состоит наш современный видеоадаптер.
Для работы видеоадаптера необходимы следующие основные компоненты:

- BIOS (Basic Input/Output System — базовая система ввода вывода)
- графический процессор
- видеоконтроллер
- видеопамять
- цифроаналоговый преобразователь, он же DAC (Digital to Analog Converter). Ранее используемый в качестве отдельной микросхемы, DAC зачастую встраивается в гра фический процессор новых наборов микросхем. Необходимость в подобном преобра зователе в цифровых системах (цифровые видеокарты и мониторы) отпадает, однако, пока

живы аналоговый интерфейс VGA и аналоговые мониторы, DAC еще некоторое время будет использоваться

- . видео - ПЗУ
- разъем
- . система охлаждения
- видеодрайвер

Пример современных видео адаптеров вы можете увидеть на рисунке.

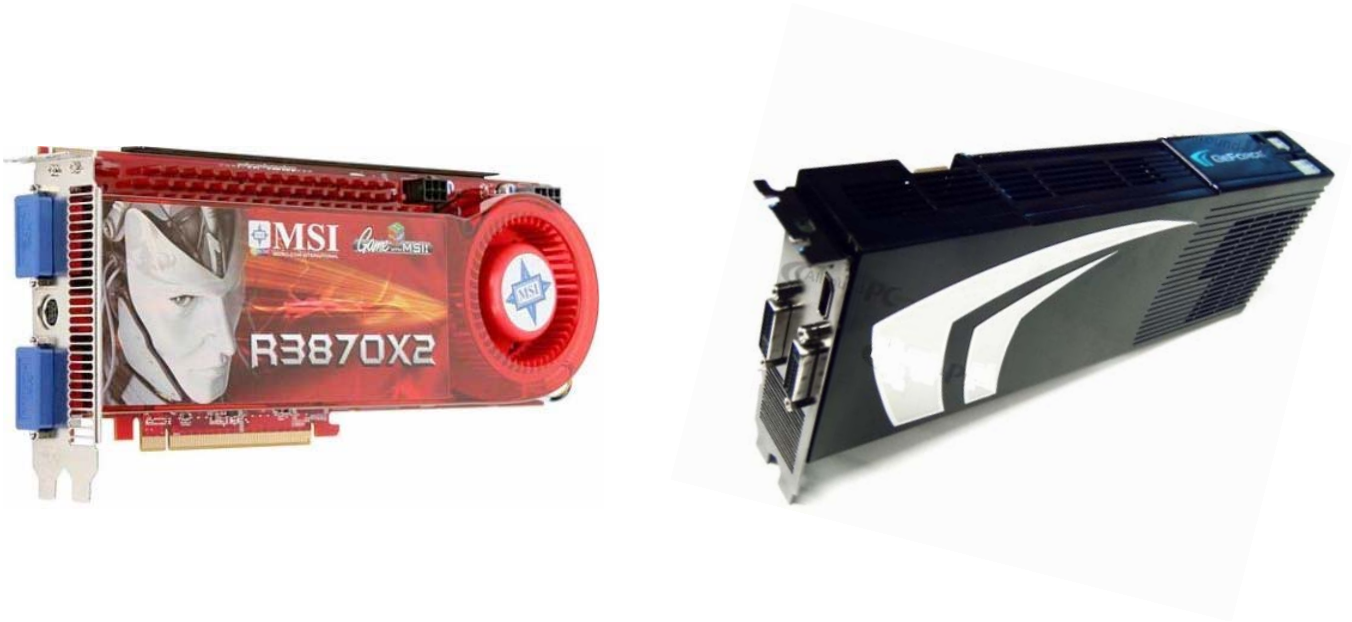


Рисунок – пример современных видеоадаптеров

А теперь по подробнее.

BIOS видеоадаптера

Видеоадаптеры имеют свою BIOS, которая подобна системной BIOS, но полностью незави сима от нее. (Другие устройства в компьютере, такие, как адаптеры SCSI, могут также иметь собственную BIOS.) Если вы включите монитор первым и посмотрите на экран, то сможете увидеть опознавательный знак BIOS видеоадаптера в самом начале запуска системы.

Хранится BIOS видеоадаптера, подобно системной BIOS, в микросхеме ROM; она содер жит основные команды, которые предоставляют интерфейс между оборудованием видеоадап тера и программным обеспечением. Программа, которая обращается к функциям BIOS видео адаптера, может быть автономным приложением, операционной системой или системной BIOS. Обращение к функциям BIOS позволяет вывести информацию о мониторе во время выполнения процедуры POST и начать загрузку системы до загрузки с диска любых других программных драйверов.

Модернизировать BIOS видеоадаптера, как и системную BIOS, можно двумя способами. Если BIOS записана в микросхеме EEPROM, то ее содержимое можно модифицировать с по мощью специальной программы, поставляемой изготовителем адаптера. В противном случае микросхему можно заменить новой, также поставляемой изготовителем. BIOS, которую можно модифицировать с помощью программного обеспечения, иногда называется flash BIOS.

Обновление BIOS видеоадаптера (“прошивка”) может потребоваться в том случае, если старый адаптер используется в новой операционной системе или изготовитель обнаруживает существенный дефект в первоначальном коде программы. Но не впадайте в соблазн модерни зировать BIOS видеоадаптера только потому, что появилась новая, пересмотренная версия.

Старайтесь следовать правилу: не модернизируйте, если в этом нет необходимости.

Графический процессор

Графический процессор, или набор микросхем, “сердце” любого видеоадаптера и характеризует быстродействие адаптера и его функциональные возможности. Два видео адаптера различных производителей с одинаковыми процессорами зачастую демонстрируют схожую производительность и функции обработки графических данных. В основном графический процессор (Graphics processing unit - графическое процессорное устройство) — занимается расчётами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчёты для обработки команд трёхмерной графики. Является основой графической платы, именно от него зависят быстродействие и возможности всего устройства. Современные графические процессоры по сложности, мало чем уступают центральному процессору компьютера, и зачастую превосходят его по числу транзисторов. Архитектура современного GPU обычно предполагает наличие нескольких блоков обработки информации, а именно: блок обработки 2D-графики, блок обработки 3D-графики, в свою очередь, обычно разделяющийся на геометрическое ядро (плюс кэш вершин) и блок растеризации (плюс кэш текстур) и др.

Видеоконтроллер

Устройство отвечающее за формирование изображения в видеопамяти, подаёт команды системе RAMDAC (Random Access Memory Digital-to-Analog Converter) на формирование сигналов развёртки для монитора и осуществляет обработку запросов центрального процессора. Кроме этого, обычно присутствуют контроллер внешней шины данных (например, PCI-E), контроллер внутренней шины данных и контроллер видеопамяти. Ширина внутренней шины и шины видеопамяти обычно больше, чем внешней (64, 128 или 256 разрядов против 16 или 32), во многие видеоконтроллеры встраивается ещё и RAMDAC. Современные графические адаптеры (ATI, nVidia) обычно имеют не менее двух видеоконтроллеров, работающих независимо друг от друга и управляющих одновременно одним или несколькими дисплеями каждый.

Видеопамять

Устройство выполняет роль кадрового буфера, в котором хранится изображение, генерируемое и постоянно изменяемое графическим процессором и выводимое на экран монитора (или нескольких мониторов). В видеопамяти хранятся также промежуточные невидимые на экране элементы изображения и другие данные. Видеопамять бывает нескольких типов, различающихся по скорости доступа и рабочей частоте. Современные видеокарты комплектуются памятью типа GDDR3, GDDR4 и новой анонсированной памятью GDDR5. Следует также иметь в виду, что помимо видеопамяти, находящейся на видеокарте, современные графические процессоры обычно используют в своей работе часть общей системной памяти компьютера, прямой доступ к которой организуется драйвером видеоадаптера через шину PCI-E.

От объема видеопамяти зависит максимальная разрешающая способность экрана и глубина цвета, поддерживаемая адаптером. На рынке в настоящее время предлагаются модели с различным объемом видеопамяти. Хотя больший объем видеопамяти не сказывается на скорости обработки графических данных, однако при использовании увеличенной шины данных или системной оперативной памяти для кэширования часто отображаемых объектов скорость видеоадаптера может существенно увеличиться. Кроме того, объем видеопамяти позволяет видеоадаптеру отображать больше цветов и поддерживать более высокое разрешение, а также хранить и обрабатывать трехмерные текстуры в видеопамяти адаптера, а не в ОЗУ системы.

Объем памяти, необходимый для создания режима с заданным разрешением и количеством цветов, вычисляется следующим образом. Для кодирования каждого пикселя изображения не обходим определенный объем памяти, а общее количество пикселей определяется заданным разрешением. Например, при разрешении **1024x768** на экране отображается **786 432** пикселя.

Если бы это разрешение поддерживало только два цвета, то для отображения каждого пикселя понадобился бы всего один бит памяти, при этом бит со значением 0 определял бы черную точку, а со значением 1 — белую. Отведя на каждый пиксель 24 бит памяти можно

отобразить более 16,7 млн. цветов, так как число возможных комбинаций для 4 разрядного двоичного числа составляет **16 777 216** (**2^{24} = 16 777 216**). Перемножив количество пикселей, используемых при заданном разрешении экрана, на число битов, требующихся для отображения каждого пикселя, получим объем памяти, необходимый для формирования и хранения изображений в этом формате.

Объем ОЗУ = (число точек в строке) x (число строк) x (число байт на одну точку)

Таблица. Минимальный объем памяти для различных режимов отображения (двухмерная графика)

Разрешение, пикселей	Глубина цвета, бит	Количество цветов	Объем модуля, Мбайт	Необходимый объем видеопамяти, байт
640x480	16	65 536	1	614 400
640x480	24	16 777 216	1	921 600
640x480	32	4 294 967 296	2	1 228 800
800x600	16	65 536	1	960 000
800x600	24	16 777 216	2	1 440 000
800x600	32	4 294 967 296	2	1 920 000
1024x768	16	65 536	2	1 572 864
1024x768	24	16 777 216	4	2 359 296
1024x768	32	4 294 967 296	4	3 145 728
1280x1024	16	65 536	4	2 621 440
1280x1024	24	16 777 216	4	3 932 160
1280x1024	32	4 294 967 296	8	5 242 880
1400x1050	16	65 536	4	2 940 000
1400x1050	24	16 777 216	8	4 410 000
1400x1050	32	4 294 967 296	8	5 880 000
1600x1200	16	65 536	4	3 840 000
1600x1200	24	16 777 216	8	5 760 000
1600x1200	32	4 294 967 296	8	7 680 000

Видеоадаптерам, поддерживающим функции трехмерной графики, при заданных глубине цвета и разрешении потребуется больший объем видеопамяти, поскольку данные видеоадаптеры используют еще три типа буфера: **передний буфер, задний буфер и Z буфер**. Объем видеопамяти, который требуется для выполнения той или иной операции, зависит от настроек глубины цвета и Z буфера. При тройной буферизации трехмерным текстурам выделяется больший объем видеопамяти, чем при двойной, однако при этом может снижаться быстродействие некоторых игр. Режим буферизации, как правило, задается в диалоговом окне свойств видеоадаптера.

В таблице представлены требования к объему видеопамяти в некоторых режимах работы видеоадаптеров, поддерживающих функции обработки трехмерной графики.

> ***Замечание:*** *3D Видеоадаптеры обычно работают в 32-битовом режиме, это вовсе не означает, что они могут воспроизвести больше 16 277 216 цветов, характерных для 24-битового режима. Многие графические процессоры и шины видеопамяти оптимизированы для передачи данных в виде 32-битовых слов; однако на самом деле они отображают 24-битовый цвет, хотя и работают в 32-битовом цветовом режиме, которому, казалось бы, соответствуют 4 294 967 296 цветов, характерные для 32-битовой глубины цвет.*

Таблица. Минимальный объем памяти видеоадаптера для различных режимов отображения (трехмерная графика)

Разрешение	Глубина цвета, бит	Глубина Z-буфера, бит	Режим буфера	Объем используемой памяти, Мбайт	Необходимый объем встроенной видеопамяти, Мбайт
640 x 480	16	16	Двойной	1,76	2
			тройной	2,34	4
	24	24	Двойной	2,64	4
			тройной	3,52	4
	32	32	Двойной	3,52	4
			тройной	4,69	8
800 x 600	16	16	Двойной	2,75	4
			тройной	3,66	4
	24	24	Двойной	4,12	8
			тройной	5,49	8
	32	32	Двойной	5,49	8
			тройной	7,32	8
1 024 x 768	16	16	Двойной	4,50	8
			тройной	6,00	8
	24	24	Двойной	6,75	8
			тройной	9,00	16
	32	32	Двойной	9,00	16
			тройной	12,00	16
1 280 x 1 024	16	16	Двойной	7,50	8
			тройной	10,00	16
	24	24	Двойной	11,25	16
			тройной	15,00	16
	32	32	Двойной	15,00	16
			тройной	20,00	32
1 600 x 1 200	16	16	Двойной	10,99	16
			тройной	14,65	16
	24	24	Двойной	16,48	32
			тройной	21,97	32
	32	32	Двойной	21,97	32
			тройной	29,30	32

Основная же причина все большего наращивания памяти видеоадаптера состоит в том, что видеопроцессор, который может самостоятельно, по управляющим командам центрального процессора, строить объемные 3D изображения, а это требует необычайно много ресурсов для хранения промежуточных результатов вычислений и образцов текстур, которыми заливаются условные плоскости моделируемых фигур.

Цифроаналоговый преобразователь

Цифро-аналоговый преобразователь (ЦАП, RAMDAC - Random Access Memory Digital-to-Analog Converter) — служит для преобразования изображения, формируемого видеоконтроллером, в уровни интенсивности цвета, подаваемые на аналоговый монитор. Возможный диапазон цветности изображения определяется только параметрами RAMDAC. Чаще всего RAMDAC имеет четыре основных блока — три цифроаналоговых преобразователя, по одному на каждый цветовой канал (красный, зелёный, синий, RGB), и SRAM для хранения данных о гамма-коррекции. Большинство ЦАП имеют разрядность 8 бит на канал — получается по 256 уровней яркости на каждый основной цвет, что в сумме дает 16,7 млн. цветов (а за счёт гамма-коррекции есть возможность отображать исходные 16,7 млн. цветов в гораздо большее цветовое пространство). Некоторые RAMDAC имеют разрядность по каждому каналу 10 бит (1024 уровня яркости), что

позволяет сразу отображать более 1 млрд. цветов, но эта возможность практически не используется. Для поддержки второго монитора часто устанавливают второй ЦАП. Стоит отметить, что мониторы и видеопроекторы, подключаемые к цифровому DVI выходу видеокарты, для преобразования потока цифровых данных используют собственные цифроаналоговые преобразователи и от характеристик ЦАП видеокарты не зависят.

Центральный процессор компьютера формирует изображение (кадр) в виде массива данных и записывает его в видеопамять, а конкретно - в кадровый буфер. После этого графический контроллер, последовательно, бит за битом, строка за строкой, считывает содержимое кадрового буфера и передает его RAMDAC. Он в свою очередь формирует аналоговый RGB-сигнал, который вместе с сигналами синхронизации передаётся на монитор.

Сканирование видеопамяти осуществляется синхронно с перемещением луча по экрану монитора. Ход электронного луча по экрану изображен на рисунке.

Количество отображённых строк в секунду называется строчной частотой развертки. А частота, с которой меняются кадры изображения, называется кадровой частотой развёртки. Последняя не должна быть ниже 75 Гц, иначе изображение будет мерцать (хотя для многих мерцает и при 75 Гц:). Частота RAMDAC говорит о том, какое максимальное разрешение при какой частоте кадровой развёртки может поддерживать видеоадаптер. От возможностей RAMDAC (частота, разрядность и т.д.) зависит качество получаемого изображения.

К основным характеристикам ЦАП можно отнести:

- **Разрядность** — количество различных уровней выходного сигнала, которые ЦАП может воспроизвести. Обычно задается в битах; количество бит есть логарифм по основанию 2 от количества уровней. Например, однобитный ЦАП способен воспроизвести два уровня, а восьмибитный — 256 уровней. Разрядность тесно связана с эффективной разрядностью (англ. ENOB — Effective Number of Bits), которая показывает реальное разрешение, достижимое на данном ЦАП.
- **Максимальная частота дискретизации** — максимальная частота, на которой ЦАП может работать, выдавая на выходе корректный результат.
- **Монотонность** — свойство ЦАП увеличивать аналоговый выходной сигнал при увеличении входного кода.
- **THD+N** (суммарные гармонические искажения + шум) — мера искажений и шума вносимых в сигнал ЦАПом. Выражается в процентах мощности гармоник и шума в выходном сигнале. Важный параметр при малосигнальных применениях ЦАП.
- **Динамический диапазон** — соотношение наибольшего и наименьшего сигналов, которые может воспроизвести ЦАП, выражается в децибелах. Данный параметр связан с разрядностью и шумовым порогом.

Видео – ПЗУ

Видео-ПЗУ (Video ROM) — постоянное запоминающее устройство, в которое записаны видео-BIOS, экранные шрифты, служебные таблицы и т.п. ПЗУ не используется видеоконтроллером напрямую — к нему обращается только центральный процессор. Хранящийся в ПЗУ видео-BIOS обеспечивает инициализацию и работу видеокарты до загрузки основной операционной системы, а также содержит системные данные, которые могут читаться и интерпретироваться видеодрайвером в процессе работы (в зависимости от применяемого метода разделения ответственности между драйвером и BIOS). На многих современных картах устанавливаются электрически перепрограммируемые ПЗУ (EEPROM, Flash ROM), допускающие перезапись видео-BIOS самим пользователем при помощи специализированных программ.

Система охлаждения

Теперь давайте рассмотрим систему охлаждения GeForce 8800 GTX. Предельно ясно, что она должна быть способной рассеивать такое же количество тепла, что и самые современные процессорные кулеры, но при этом оставаться достаточно компактной: в отличие от

процессорного, графическому кулеру расти практически некуда, он ограничен габаритами видеоадаптера. Как же NVIDIA вышла из положения?

Сегодня очевидно, что система охлаждения современного графического адаптера должна обеспечивать отвод горячего воздуха за пределы корпуса системы – ведь лишние 120-150 Ватт тепла, рассеиваемые внутри корпуса, отнюдь не способствуют улучшению и без того напряженной тепловой обстановки, особенно, если система укомплектована мощным CPU. И если системы охлаждения, разработанные ATI Technologies, используют этот принцип еще со времен Radeon X850, то NVIDIA до сегодняшнего дня следовала ему лишь частично (если не считать печально известной системы охлаждения GeForce FX 5800 Ultra). Как известно, кулер GeForce 7900 GTX отводит за пределы корпуса лишь часть нагретого воздуха, а кулер GeForce 7950 GX2 не делает и этого в силу двухплатной конструкции самой карты. Разрабатывая новую систему охлаждения для семейства GeForce 8800, NVIDIA постаралась учесть старые недостатки и избавиться от них. Вот, что получилось в результате:

Больше всего новый кулер напоминает конструкцию, которая устанавливалась некогда на GeForce 6800 Ultra, но увеличенную в размерах и развернутую на 180 градусов таким образом, чтобы горячий воздух выбрасывался наружу через прорези в крепежной планке видеоадаптера, а не внутрь корпуса системы. Разумеется, здесь применена более сложная конструкция, по сравнению с GeForce 6800 Ultra, так как G80 выделяет значительно больше тепла, нежели NV40. К немалому нашему удивлению, NVIDIA решила не использовать медный радиатор, как это сделала ATI Technologies. Сквозь прорези в кожухе видно, что радиатор, как и прежде, набран из тонких алюминиевых пластин. Демонтировав кожух кулера, мы увидели следующую картину:

Здесь в основном применена та же компоновка, что и в системе охлаждения ATI Radeon X1950 XTX: тепловой поток, генерируемый GPU, отбирается массивным медным основанием и при помощи тепловой трубки равномерно распределяется по радиатору, что значительно улучшает эффективность теплоотвода.

Основание, радиатор и вентилятор установлены на легкой алюминиевой раме, имеющей выступы в местах контакта с чипами памяти, кристаллом чипа, содержащего TMDS-трансмиттеры и RAMDAC, а также корпусами сильно нагревающихся ключевых МОП-транзисторов системы питания. Во всех случаях в качестве теплопроводящих прокладок используются традиционные для графических карт NVIDIA подушечки из неорганического волокна, пропитанные белой термопастой. В месте контакта медной подошвы с крышкой GPU нанесен слой темно-серой густой термопасты, также хорошо знакомой нашим читателям по многочисленным обзорам мощных видеоадаптеров. Рама имеет прямоугольные прорези в районе вентилятора, что позволяет несколько улучшить охлаждение силовых элементов и PCB путем забора воздуха через эти прорези.

За продувку радиатора отвечает радиальный вентилятор, так называемый "бловер", обладающий двумя преимуществами – во-первых, создаваемый им воздушный поток изначально направлен перпендикулярно оси вращения крыльчатки, а во-вторых, статическое давление потока выше, нежели развиваемое классическим осевым вентилятором той же мощности. При данной компоновке системы охлаждения радиальный вентилятор является оптимальным выбором, позволяющим осуществлять эффективный продув длинного радиатора с малым шагом ребра, имеющего высокое аэродинамическое сопротивление.

Потребляемый ток примененного конструкторами NVIDIA вентилятора составляет 0.48A при напряжении питания 12В, что дает мощность порядка 5.8 Ватт. Скорее всего, при работе вентилятора на максимальных оборотах уровень шума, создаваемый крыльчаткой, был бы невыносимым, но будем надеяться, что система управления оборотами вентилятора, установленная на GeForce 8800 GTX, покажет себя с лучшей стороны.

В целом, система охлаждения NVIDIA GeForce 8800 GTX представляет собой логичную, законченную конструкцию, рассчитанную на охлаждение даже столь мощного чипа, как G80. Несколько настораживает лишь использование в конструкции главного радиатора алюминия вместо меди, что может обернуться на практике необходимостью работы вентилятора на повышенных оборотах, а, следовательно, и повышенным уровнем шума.

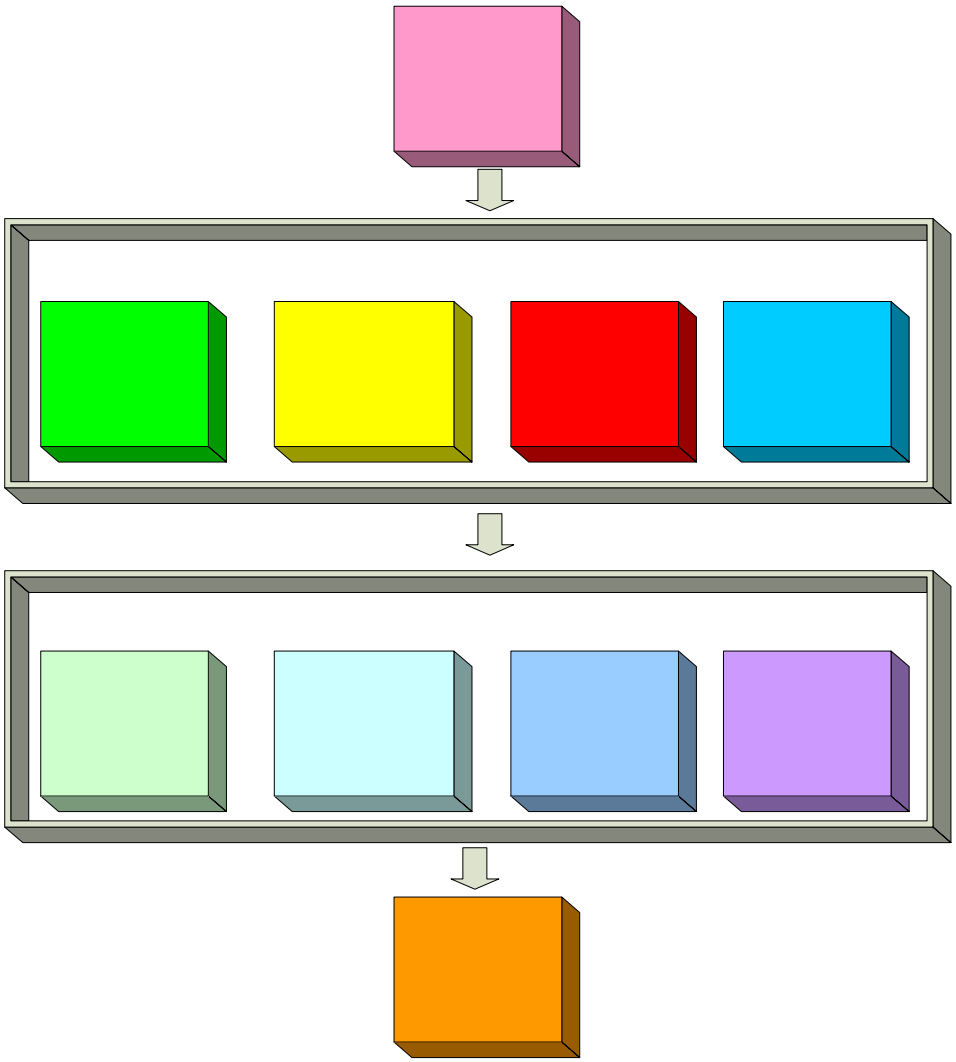
Видеодрайвер

Правильная и полнофункциональная работа современного графического адаптера обеспечивается с помощью видеодрайвера — специального программного обеспечения, поставляемого производителем видеокарты и загружаемого в процессе запуска операционной системы. Видеодрайвер выполняет функции интерфейса между системой с запущенными в ней приложениями и видеоадаптером. Так же как и видео-BIOS, видеодрайвер организует и программно контролирует работу всех частей видеоадаптера через специальные регистры управления, доступ к которым происходит через соответствующую шину.

Большинство производителей видеоадаптеров и наборов микросхем системной логики имеют свои Web сервера, где можно найти информацию о самых последних версиях драйверов. Хотя может пригодиться драйвер, поставляемый с набором микросхем системной логики, однако лучше использовать драйверы, поставляемые производителем адаптера.

1.4 Технология построения 3D сцен.

Теперь хотелось бы рассмотреть, как же Видеоадаптер строит трехмерную сцену. В большинстве подсистем трехмерной графики применяется графический конвейер. Конвейер - это логическая группа вычислений, выполняемых последовательно, которые дают на выходе синтезируемую сцену, Конвейер разделен на множество этапов, на каждом из которых аппаратно или программно выполняется некоторая функция

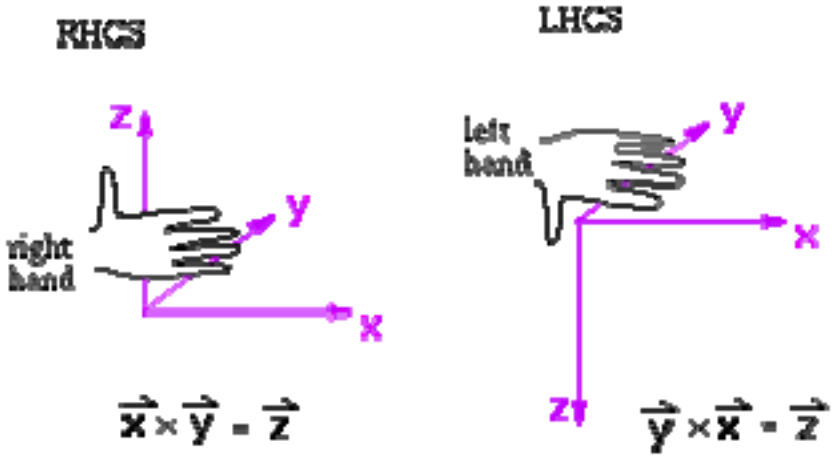


Наличием переходов между этапами конвейера обеспечивается возможность выбора между программной и аппаратной реализацией очередного этапа. Такой подход к настройке конвейера позволяет приложениям трехмерной графики получать преимущества аппаратной реализации, когда таковая доступна. Таким образом, реализация конвейера может чисто программной, полностью аппаратной или смешанной (программно-аппаратной)

Описание сцены (построение геометрической модели)

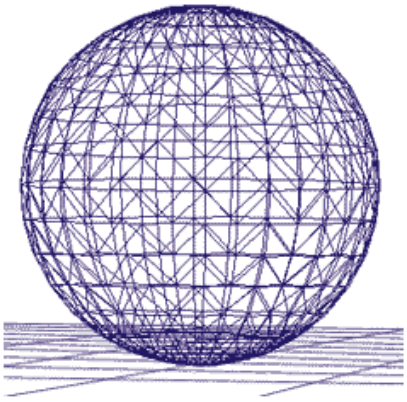
До начала работы графического конвейера и выполнения геометрических преобразований необходимо описать трехмерную сцену, изображение которой необходимо синтезировать.

Трехмерное приложение оперирует объектами, описанными в некоторой глобальной системе координат. Чаще всего здесь используется ортогональная (декартова) система координат, в которой положение каждой точки задается ее расстоянием от начала координат по трем взаимно перпендикулярным осям X, Y и Z



Кроме того необходимо определить состояние объектов, принимающих участие в сцене, которую необходимо отобразить. На первый взгляд к самой графике этот этап отношения не имеет. На самом деле он является определяющим, ибо состояние объектов и их взаимное положение формируют логику последующих действий программы. Например, если вашего персонажа в игре уже «убили», какой смысл вообще выводить трехмерную сцену? С каждым объектом в сцене связана соответствующая текущему моменту геометрическая модель. То есть объект один (например, некий монстр), но с ним сопоставляются разные геометрические модели в зависимости от состояния — жив, ранен убит, трансформировался в другой объект и т. д. Практически все операции на первом этапе выполняет центральный процессор. Результаты его работы пересылаются в графический чипсет посредством драйвера.

Генерация полигонов или тесселяция (tesselation)



Первый этап геометрического преобразования заключается в описании каждого объекта группой треугольников (многоугольников). Треугольники формируются на основе множества вершин, заданных приложением. Объекты, сконструированные из треугольников или многоугольников, называются каркасной (проволочной) моделью

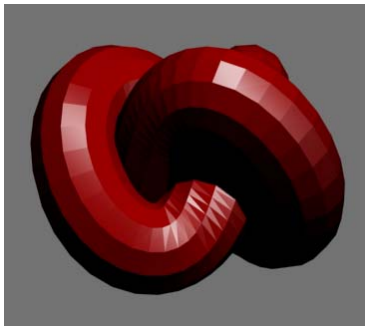
На данном этапе происходит декомпозиция (разделение на примитивы) геометрических моделей. Внешний вид объекта формируется с помощью набора определенных примитивов. В роли примитива как раз и выступают треугольник как простейшая плоская фигура, однозначно располагаемая в трехмерном пространстве. Все прочие элементы состоят из таких треугольников. Таким образом, можно утверждать, что по большей части термины «полигон» и «треугольник» применительно к

игровой графике суть синонимы.

Современные графические процессоры умеют выполнять дополнительные операции, например тесселяцию (Tesselation), то есть разделение исходных треугольников на более мелкие. Некоторые графические чипсеты могут аппаратно обрабатывать геометрические модели, построенные на основе параметрических поверхностей (механизм RT-Patches). Часть графических процессоров умеет превращать плоские треугольники в трехмерные поверхности путем «выдавливания» в третье измерение (механизм N-Patches).

Итоговый результат операций второго этапа пересылается в блок трансформаций и освещения (Transform&Lighting, T&L) геометрического процессора.

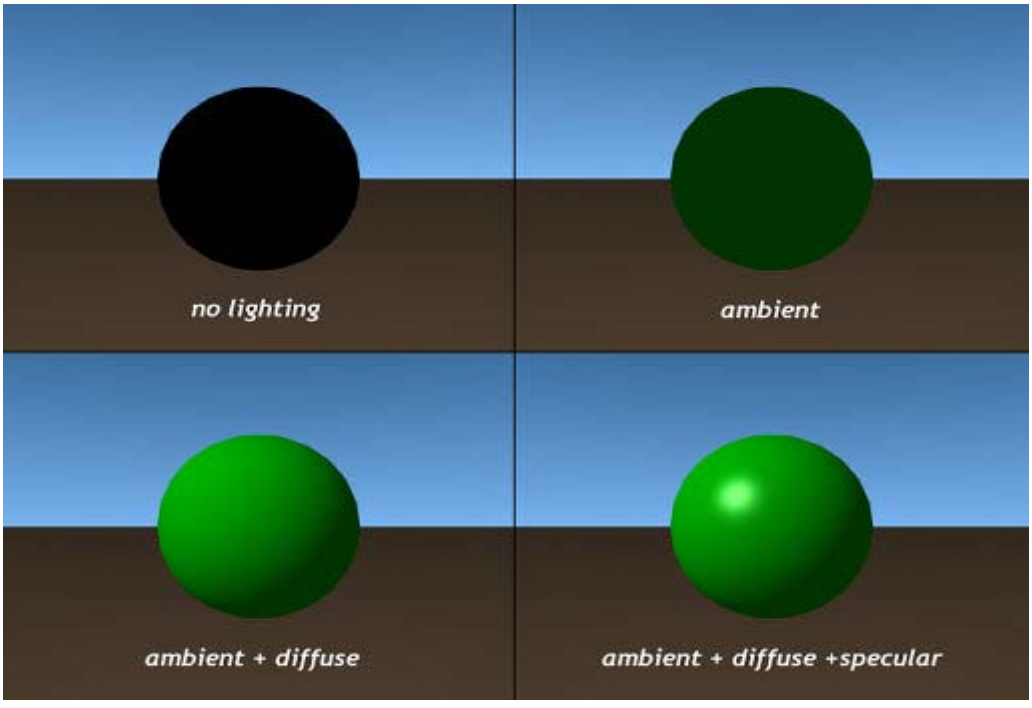
Модельное преобразование или трансформация (transformation)



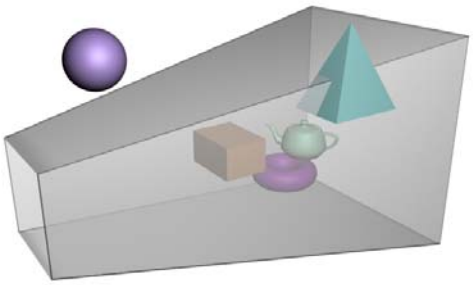
Трансформация - простые объекты чаще всего необходимо определенным образом изменить или трансформировать (transformation), чтобы получился более естественный объект, или имитировать его перемещение в пространстве. Для этого координаты вершин граней объекта (Vertex Shaders) пересчитывают с использованием операций матричной алгебры и геометрических преобразований. В современных видеокартах для этого интенсивно используется геометрический сопроцессор, а в более старых - этим должен заниматься центральный процессор.

Освещение (Lighting)

Расчет освещенности и затенения - для того чтобы объект был виден на экране, надо рассчитать освещенность (lighting) и затенение (shading) каждого элементарного прямоугольника или треугольника. Причем необходимо имитировать реальное распределение освещенности, т.е. требуется скрыть изменения освещенности между прямоугольниками или треугольниками. Модель освещения описывает тип используемых источников света и затем, когда определены свойства освещаемого объекта, формируется эффект освещения. Общепринятые модели освещения включают рассеянный свет, направленный и точечный источники света. Отражающие свойства материала в сочетании с моделью освещения задают цвет объекта



Проверка глубины или Отсечение (clipping)



Завершает стадию геометрических преобразований этап установки. На этапе установки изменяются размеры треугольников в зависимости от положения точки наблюдения сцены. Также удаляются невидимые грани. Хотелось бы заметить что процедура отсечения является очень сложным процессом и проходит в несколько этапов, на разных стадиях обработки изображения

Обработка полигонов (Polygon setup или Triangle setup)

Обработка координат вершин - на этапах моделирования объекта, все координаты вершин, граней получаются в виде чисел с плавающей запятой, но поскольку в видеопамять можно занести только целые числа, необходимо осуществить этап преобразования. На этом же этапе может проводиться сортировка вершин, чтобы отбросить невидимые грани. При расчетах используется субпиксельная коррекция, когда каждый пиксел представляется в виде матрицы субпикселов, с которой проводятся вычисления.
Сама процедура обработки как правило проходит в несколько этапов:

- Прием координат вершин треугольника и их атрибутов: освещенности, координат
- Коррекцию перспективы треугольника. То есть фигура масштабируется в зависимости от ее удаленности.
- Проектирование треугольника на плоскость XY экрана
- Вычисление пикселов, входящих в проекцию треугольника.
- Расчет для этих пикселов атрибутов. Расчет производится интерполяцией значений, задаваемых обычно на вершинах треугольника.
- Посылку обработанного треугольника следующему блоку

На последнем этапе данной стадии происходит проецирование - трехмерный объект преобразуется в двумерный, но запоминаются расстояния вершин граней до поверхности экрана (координата z, r-буфер), на который проецируется объект.

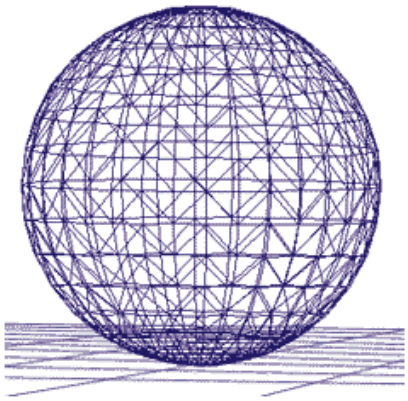
Наложение текстур

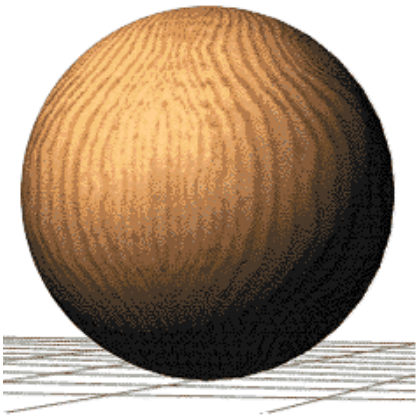
Так-как. возможности процессора видеокарты не бесконечны, поверхность объекта моделируется с помощью ограниченного количества прямоугольников или треугольников, поэтому, чтобы создать реалистичное изображение, на каждую элементарную поверхность накладывают текстуру (texture), имитирующую реальную поверхность. Текстуры хранятся в памяти в виде растровых картинок. Минимальный элемент растровой картинки носит название тексел (texel - TEXture Element). Этап наложения текстур наиболее трудоемок и сложен, причем здесь возникает множество проблем с совмещением краев текстур соседних плоскостей.

То есть всю процедуру можно представить в виде двух этапов:

- Вычисляется накладываемый тексел (иногда весьма изощренно).
- Производится наложение тексела с учетом освещенности пиксела.

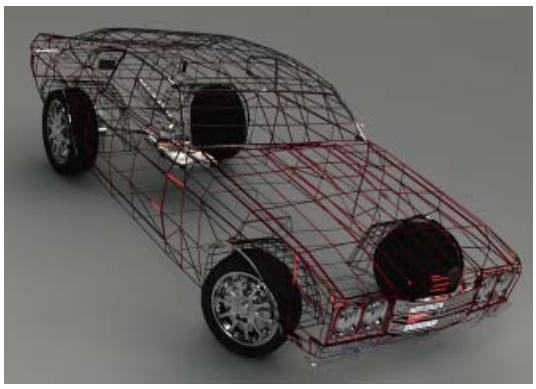
Кроме того, при масштабировании изображения имеет место проблема согласования величины разрешения используемой текстуры с разрешением монитора, т.к. при использовании текстуры с малым разрешением изображение на экране получается в виде набора цветных квадратиков, а при использовании текстур с большим разрешением может не хватить памяти для их хранения.





Наложение эффектов

На данном этапе, мы придаем нашим объектам большей реалистичности. Создаем эффектов прозрачности и полупрозрачности, проводим коррекция цвета пикселов (альфа-смешение, затуманивание) с учетом прозрачности смоделированных объектов, учитываются свойства окружающей объекты среды. Но о том какие методы улучшения качества и реалистичности изображения существуют мы рассмотрим чуть позже, в следующей нашей главе.



Сглаживание

На следующем этапе необходимо выполнить коррекцию дефектов - смоделированные линии и границы объектов, если они не вертикальны или горизонтальны, на экране выглядят угловатыми, поэтому проводят коррекцию изображения, называемую антиалиасинг (anti-aliasing). А также выполняется интерполяция недостающих цветов - если при моделировании объектов использовалось другое количество цветов, нежели чем в текущем режиме видеокарты, то необходимо рассчитать недостающие цвета или удалить избыточные. Этот процесс называется дизеринг (dithering).

1.5 Технологии улучшения качества 3D объектов.

Antialiasing (антиалиасинг)

Одна из наиболее впечатляющих аппаратных технологий, все чаще находящая применение в игровых приложениях, - это устранение ступенчатости, или **антиалиасинг (antialiasing)**.

Эта технология предназначена для устранения одной из ключевых проблем качества синтезированных изображений - лестничного эффекта (также часто называют **алиасингом**, **aliasing**). Часто этот эффект можно заметить на границах объектов или на линиях, близких к вертикальным или горизонтальным, но не строго вертикальных или, соответственно,

горизонтальных. Несомненно, читатель много раз встречался с подобными эффектами, например, в трехмерных играх.

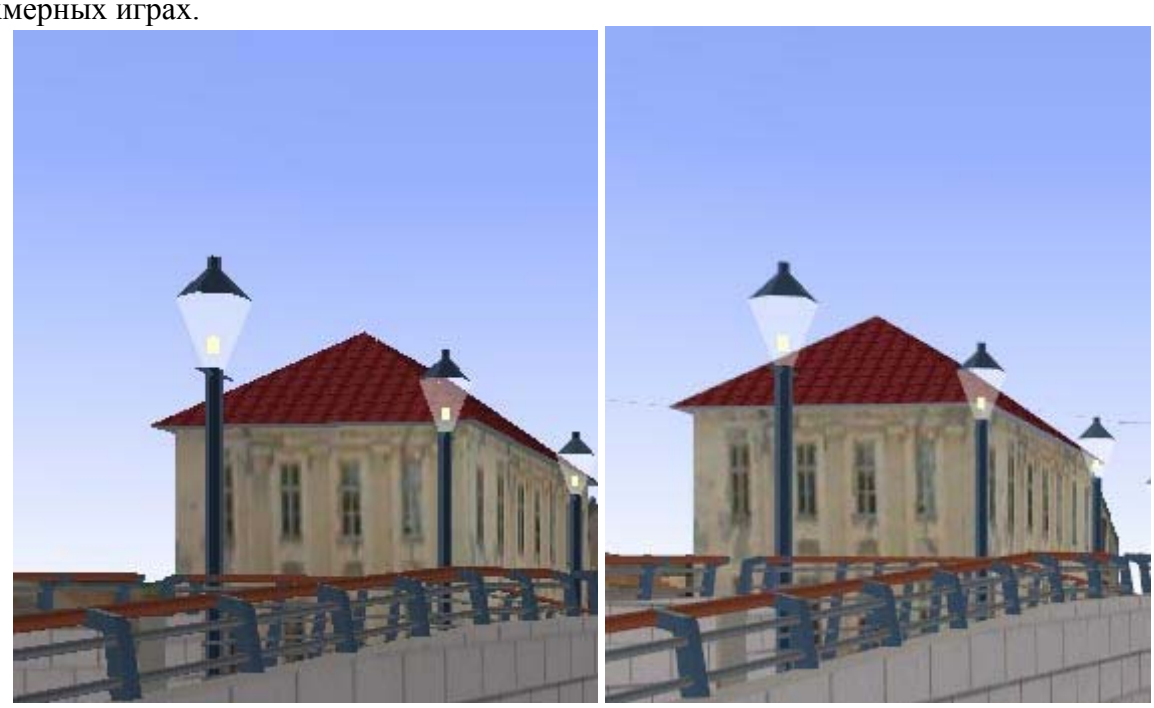
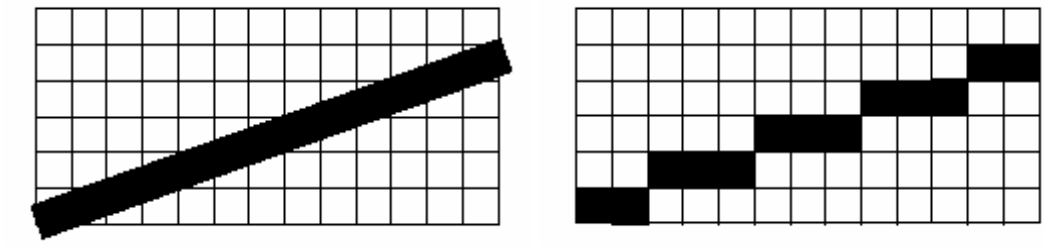


Рисунок - Левое изображение получено в стандартном режиме, правое - со включенным устранением ступенчатости. Обратите внимание на передачу мелких деталей и границы объектов (перила на мосту, фонари и т.п.)

Что является причиной лестничного эффекта? Опуская подробности, можно сказать, что главная причина - "сетка" пикселей на компьютерном мониторе. Эта сетка имеет конечное и весьма небольшое разрешение, а пиксели располагаются в строго фиксированных местах.

Ступенчатость возникает, когда точки на линиях пересекают строки или столбцы пикселей под небольшим углом. Часть линии шириной в один пиксель может попасть на один пиксель экрана, а часть - на другой. Таким образом, получается неопределенность: можно рисовать эту часть как один пиксель на одном ряду, один пиксель на другом ряду или закрашивать оба пикселя. К сожалению, все три способа вносят хорошо заметные дефекты в изображение. Если закрашивать только один пиксель, линия может получиться тоньше чем нужно, и будет хорошо заметен разрыв в том месте, где линия переходит с одного ряда пикселей на другой. Если закрашивать оба - ширина линии в этом месте будет два пикселя. Аналогичные артефакты возникают не только при рисовании линий, а практически на всех границах, встречающихся в изображении.



Как же бороться с этим неприятным эффектом? Можно заметить, что размер одной "ступеньки" никогда не бывает больше, чем один пиксель. Поэтому наиболее естественный и простой путь - это увеличение экранного разрешения. Т.е. увеличение разрешения уменьшает размер пикселя и, следовательно, заметность артефактов. Однако увеличить разрешение можно не всегда. Например, оно может быть ограничено максимальным разрешением монитора или его может ограничивать приложение.

Другой путь - увеличение так называемого эффективного разрешения, т.е. использование специального алгоритма для получения цвета пикселя таким образом, чтобы эмулировать несколько пикселей. Такие методы и называются методами устранения ступенчатости.

Эти методы вычисляют значение цвета как бы внутри экранного пикселя в нескольких точках. Эти точки называются сэмплами (sample). Сэмплы представляют собой те самые дополнительные пиксели изображения, которые увеличивают эффективное разрешение. Значение сэмплов используются для вычисления конечного цвета пикселя.

Supersampling (Суперсэмплинг)

Суперсэмплинг - это технология устранения ступенчатости, которая используется практически во всех современных аппаратных ускорителях. Графический процессор, который использует суперсэмплинг, визуализирует экранное изображение с разрешением, значительно большим чем текущее разрешение дисплея. Существует достаточно много методов выполнения этой операции, при этом их всех можно охарактеризуются числом используемых дополнительных пикселей. После рисования изображения с высоким разрешением, процессор уменьшает размер картинки до разрешения дисплея, причем эта операция производится с соответствующей фильтрацией.

Степень изменения размера изображения определяется отношением числа пикселей в исходном изображении (высокого разрешения) к числу пикселей в выходном изображении. Например, 2x суперсэмплинг пишет в буфер кадра в два раза больше пикселей, 4x - в четыре и т.д. Чем больше это число, тем более качественное устранение ступенчатости можно получить.

Как нетрудно догадаться, использование суперсэмплинга вызывает значительное ухудшение скорости визуализации. Если графический процессор рисует в четыре раза больше пикселей, скорость будет в четыре раза меньше по сравнение со стандартным режимом. И даже хуже, потому что необходимо дополнительно фильтровать изображение высокого разрешения. Вот почему после включения устранения ступенчатости в драйверах видеокарт скорость рисования сильно падает.

NVidia Quincunx (tm)

В режиме QuincunxT (читается - "квинканкс") каждый сэмпл из каждого прохода дополнительно фильтруется с четырьмя соседними из второго изображения. NVidia заявляет, что такая схема дает качество, практически равное 4x. Однако отметим, что улучшение происходит не столько за счет увеличения количества сэмплов (используется два сэмпла, аналогично 2x-шаблону), а размывания изображения за счет смешивания с соседними пикселями. Тем не менее, на практике это дает хороший результат. Существует также NVidia AccuviewT - модифицированная технология, в которой положения сэмплов слегка сдвигаются внутрь основного пикселя для улучшения качества.

ATI Smoothvision (tm)

Эта технология реализована в графических процессорах ATI. В ней используются группы из 16 сэмплов, которые распределяются между различным количеством точек, в зависимости от желаемого качества устранения ступенчатости. В 2x-режиме сэмплы покрывают 8 точек, в 4x режиме - 4 точки и так далее. Каждый пиксель имеет 8 фиксированных псевдослучайных положений, в которые могут попасть сэмплы. Известно, что человеческий глаз хорошо замечает регулярные структуры, что в данном случае является нежелательным. Использование нерегулярного шаблона позволяет создавать более приятные глазу картинки.

Motion Blur (Эффект размывания при движении)

Размывание при движении (motion blur) - еще одна из множества технологий, которые используются для того, чтобы сделать картинки в 3D играх более реалистичными. Если камерой снимается реальный мир, можно заметить, что быстро движущиеся объекты как бы расплываются, размазываются на изображении. Этот эффект и называется размыванием при движении (иногда

мы будем называть его размыванием движения). Его использование может заставить любую игру выглядеть "как в реальном мире".

На самом деле, размывание при движении пришло к нам не из реального мира (у людей объекты в принципе не должны размываться перед глазами), а из кинематографа. Этот эффект есть практически в любом фильме, но вы скорее всего его не замечаете. Как и многие другие артефакты фотографии, он воспринимается как дополнительный эффект, придающий ощущения реализма. В то время как фотографы стараюсь избавиться от артефактов, программисты игр старательно воссоздают их в своих играх. Мы все так привыкли видеть размывание при движении в телевизионных программах и фильмах, что анимация без него выглядит нереалистично.

Во время работы камеры затвор открывает чувствительная пленку на короткое время, в течение которого на нее падает свет. Это вызывает химические реакции, пленка темнеет и в конце концов на ней образуется изображение сцены. Этот процесс известен как экспозиция. Если во время экспозиции сцена меняется, на пленке образуется размытое изображение.

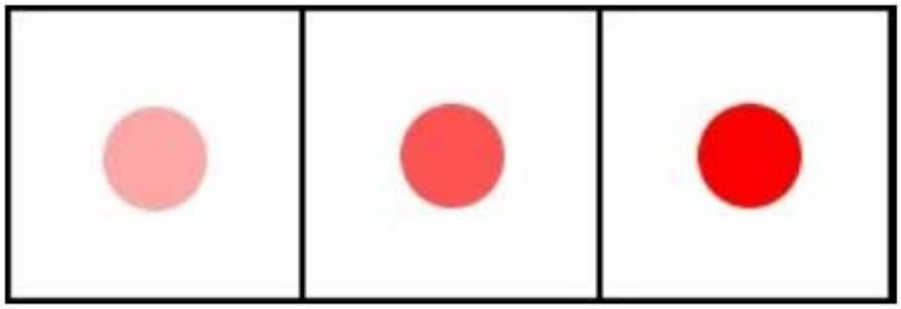


Рисунок - Чем дольше длится экспозиция, тем темнее объект.

Природа компьютерной анимации такова, что объекты не движутся плавно, глаз может замечать скачки при движении объекта. Предположим, что игра работает со скорость 60 FPS, т.е. каждый новый кадр рисуется через 1/60 секунды. Однако время непрерывно и не обновляется только 60 раз в секунду. Также и монитор, и человеческий глаз не могут "обрабатывать" изображения непрерывно, им нужен некоторый интервал времени для обновления изображения. Эффект анимации "скачками", который неизбежен даже при самой быстрой аппаратуре, называется временной ступенчатостью (по аналогии с эффектом ступенчатости изображения, который мы уже рассматривали).

Основная идея технология устранения этого эффекта подобна той, которая использовалась для устранения ступенчатости изображения - суперсэмплинг. Только в данном случае это так называемый временной суперсэмплинг. Для того, чтобы представить непрерывное время в одном кадре, необходимо дополнительно нарисовать несколько кадров-сэмплов, затем усреднить их и таким образом получить нужный эффект. Это усреднение, кстати, и вызовет эффект размывания, причем чем быстрее движется объект, тем сильнее он сдвинется на кадрах-сэмплах и поэтому его размывание будет сильнее.

Например, для размывания движения с использованием 4-х сэмплов необходимо рисовать со скоростью 240 FPS, чтобы получить вывод со скоростью 60 FPS. Чем больше сэмплов используется, тем более качественное изображение получится.

Однако такой метод работает медленно. Для современной игры даже на самой лучшей аппаратуре 60 FPS - предел мечтаний, а про 240 можно и не говорить. Таким образом необходимы быть может менее качественные, но более быстрые решения.

В некоторых компьютерных играх для эмуляции размывания при движении используются "следы" от объектов. Изображение текущего кадра смешивается с изображением предыдущего, т.е. получается что несколько последних кадров все время видны на экране. Это вызывает эффект "расплывания" объектов по экрану. Это, конечно, не настоящие устранение временной ступенчатости, но все же оно дает некоторый эффект.

Появление пиксельных шейдеров в современной аппаратуре позволило разработать новые алгоритмы для размывания движущихся объектов. Эти алгоритмы основываются на вычислении "скорости" движения каждого пикселя из кадра анимации и пост-обработки кадра таким образом, чтобы эмулировать эффект усреднения промежуточных кадров. На первом этапе сцена просто рисуется в буфер кадра и сохраняется в текстуре. Затем для каждой вершины находится разница между ее текущим положением и положением на предыдущем кадре - это и будет вектор скорости движения данной точки. Чем он длиннее, тем быстрее движется точка. Далее для каждого пикселя сцены выбираются несколько соседних точек текстуры (в которой, напомним, хранится изображение текущего кадра) и их значения усредняются и записываются как цвет текущей точки. Соседние точки выбираются по информации о скорости основного пикселя.

Такие технологии работают достаточно быстро и качественно, хотя при этом требуют наличие самых современных видеоадаптеров.

Z-buffer

Часть графической памяти, в которой хранятся расстояния от точки наблюдения до каждого пиксела (значения Z). Z-buffer определяет, какая из многих перекрывающихся точек наиболее близка к плоскости наблюдения.

Также, как большее число битов на пиксель для цвета в буфере кадра соответствует большему количеству цветов, доступных в системе изображения, так и количество бит на пиксель в z-буфере соответствует большему числу элементов. Обычно, z-буфер имеет не менее 16 бит на пиксел для представления глубины цвета. Аппаратные акселераторы 3D графики могут иметь собственный z-буфер на графической карте, чтобы избежать удвоенной нагрузки на системную шину при передаче данных. Некоторые реализации Z-buffer используют для хранения не целочисленное значение глубины а значение с плавающей запятой от 0 до 1.

Z-buffering

Процесс удаления скрытых поверхностей, использующий значения глубины, хранящиеся в Z-буфере. Перед отображением нового кадра, буфер очищается, и значения величин Z устанавливаются равными бесконечности. При рендеринге объекта устанавливаются значения Z для каждого пиксела: чем ближе расположен пиксел, тем меньше значение величины Z. Для каждого нового пиксела значение глубины сравнивается со значением, хранящимся в буфере, и пиксел записывается в кадр, только если величина глубины меньше сохраненного значения.

Z-sorting

Процесс удаления невидимых поверхностей с помощью сортировки многоугольников в порядке низ-верх, предшествующий рендерингу. Таким образом, при рендеринге верхние поверхности обрабатываются последними. Результаты рендеринга получаются верными только, если объекты не близки и не пересекаются. Преимуществом этого метода является отсутствие необходимости хранения значений глубины. Недостатком является высокая загрузка процессора и ограничение на пересекающиеся объекты.

Alpha

Коэффициент прозрачности. В описание цвета (RGB) может входить специальный канал, называемый альфа-каналом, который отвечает за прозрачность данного цвета. Т.о., цвет описывается как ARGB.

Alpha Blending (Alpha pixel blending)

Реальный мир состоит из прозрачных, полупрозрачных и непрозрачных объектов. Alpha Blending -- это способ передачи информации о прозрачности полупрозрачным объектам. Эффект прозрачности и просвечивания достигается путем смешивания значений цветов исходного и результирующего пикселей. Разделение изображения на многоугольники производится с использованием маски, плотность которой зависит от прозрачности объекта. В результате цвет точки является комбинацией цветов переднего и заднего плана. Обычно, Alpha имеет нормализованное значение от 0 до 1 для каждого цветного пиксела. Новый пиксел = (alpha)(цвет пиксела A) + (1 - alpha)(цвет пиксела B)

Alpha Buffer

Альфа буфер. Дополнительный буфер, в котором содержится информация о прозрачности, таким образом, пиксел имеет четырехзначное представление (RGBA), и в 32-разрядном буфере содержится 24 бита информации о цвете, т.е. 8 бит на каждый из цветов (красный, зеленый и синий), и 8 бит на значение alpha.

Double Buffering

Двойная буферизация. Представьте себе старый трюк аниматоров: нарисованный на уголках стопки бумаги персонаж мультика со слегка изменяемым положением на каждом следующем листе; затем, пролистав всю стопку, отогнув уголок, мы увидим плавное движение нашего героя. Практически такой же принцип работы имеет и Double Buffering в 3D анимации, т.е. следующее положение персонажа уже нарисовано до того, как текущая страница не пролистана. Без применения двойной буферизации движущееся изображение не будет иметь требуемой плавности, т.е. 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 использования двух буферов 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 наблюдатель все время видит отличную картинку.

Back buffer

Вторичный буфер. Область памяти, в которой рассчитываются объекты трехмерной сцены. Вывод изображения на экран осуществляется через Front Buffer (первичный буфер). Обычно

процесс копирования содержимого вторичного буфера синхронизируется с обратным ходом луча ЭЛТ монитора. Таким образом достигается плавная смена кадров.

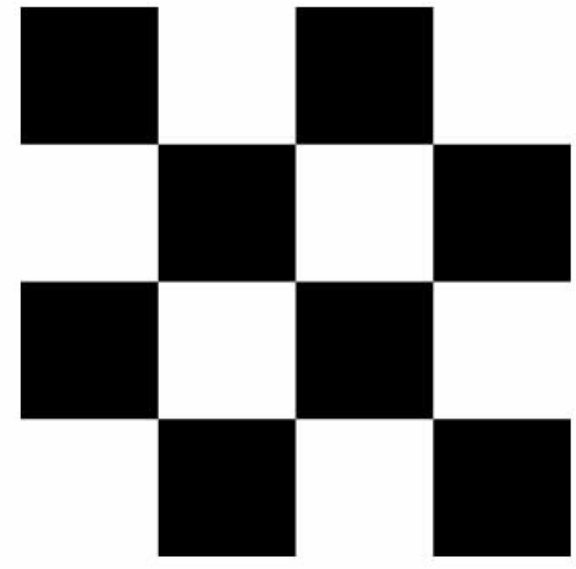
1.6 Методы фильтрации текстур.

Текстурирование является важнейшим элементом сегодняшних 3D приложений, без него многие трехмерные модели теряют значительную часть своей визуальной привлекательности. Однако процесс нанесения текстур на поверхности не обходится без артефактов и соответствующих методов их подавления. В мире трехмерных игр то и дело встречаются специализированные термины типа "мип-мэппинг", "трилинейная фильтрация" и т.п., которые как раз и относятся к этим методам.

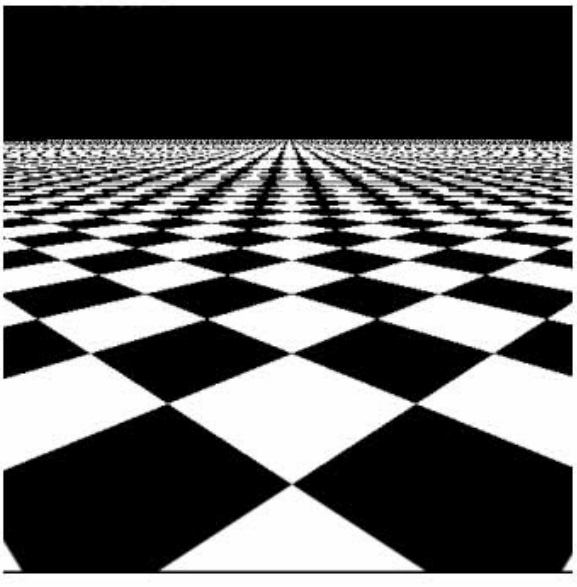
Частным случаем эффекта ступенчатости, рассмотренным ранее, является эффект ступенчатости текстурированных поверхностей, который, к сожалению, нельзя убрать методами мульти- или суперсэмплинга, описанными выше.

Представьте себе черно-белую шахматную доску большого, практически бесконечного размера. Допустим, мы рисуем эту доску на экране и смотрим на нее под небольшим углом. Для достаточно удаленных участков доски размеры клеток неизбежно начнут уменьшаться до размера одного пикселя и меньше. Это так называемое оптическое уменьшение текстуры (minification). Между пикселями текстуры начнется "борьба" за обладание пикселями экрана, что приведет к неприятному мельтешению, что является одной из разновидностей эффекта ступенчатости. Увеличение экранного разрешения (реального или эффективного) помогает только немного, потому что для достаточно удаленных объектов детали текстур все равно становятся меньше пикселей.

С другой стороны, наиболее ближние к нам части доски занимают большую экранную площадь, и можно наблюдать огромные пиксели текстуры. Это называется оптическим увеличением текстуры (magnification). Хотя эта проблема стоит не так остро, для уменьшения негативного эффекта с ней тоже необходимо бороться.



Это текстура шахматной доски



Плоскость с наложенной текстурой. Обратите внимание на искажения, возникающие при уменьшении клеток

Для решения проблем текстурирования применяется так называемая фильтрация текстур. Если разобраться в процессе рисования трехмерного объекта с наложенной текстурой, можно увидеть, что вычисление цвета пикселя идет как бы "наоборот", - сначала находится пиксель экрана, куда будет спроецирована некоторая точка объекта, а затем для этой точки находятся все пиксели текстуры, попадающие в нее. Выбор пикселей текстуры и их комбинация (усреднение) для получения финального цвета пикселя экрана и называется фильтрацией текстуры.

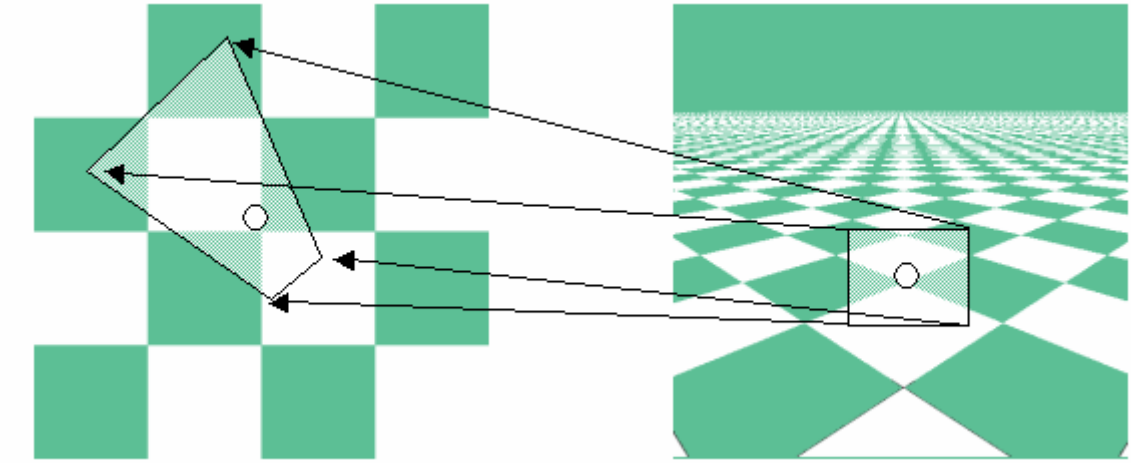


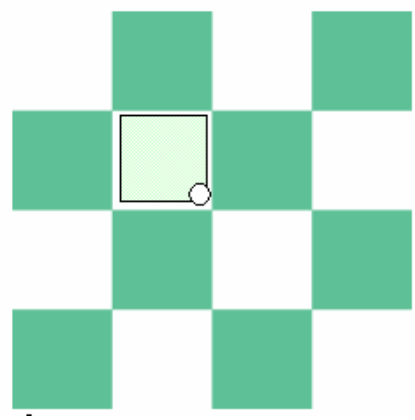
Рисунок. Область экрана и ее образ в текстуре

В процессе текстурирования каждому пикселю экрана ставится в соответствие координата внутри текстуры, причем эта координата не обязательно целочисленная. Более того, пикселю соответствует некоторая область в изображении текстуры, в которую могут попадать несколько пикселей из текстуры. Будем называть эту область образом пикселя в текстуре. Для ближних частей нашей доски пиксель экрана становится значительно меньше пикселя текстуры и как бы находится внутри него (образ содержится внутри пикселя текстуры). Для удаленных, наоборот, в каждый пиксель попадает большое количество точек текстуры (образ содержит в себе несколько точек текстуры). Образ пикселя может иметь различную форму и в общем случае представляет собой произвольный четырехугольник.

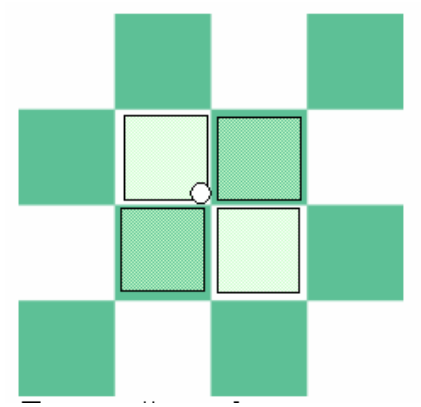
Рассмотрим различные методы фильтрации текстур и их вариации

Nearest Neighbor (Ближайший сосед)

В этом, наиболее простом, методе в качестве цвета пикселя просто выбирается цвет ближайшего соответствующего пикселя текстуры. Этот метод самый быстрый, но и наименее качественный. По сути, это даже не специальный метод фильтрации, а просто способ выбрать хоть какой-то пиксель текстуры, соответствующий экранному пикселю. Он широко применялся до появления аппаратных ускорителей, вместе с широким распространением которых появилась возможность использовать более качественные методы.



Фильтрация методом ближайшего соседа



Билинейная фильтрация

Bilinear (Билинейная фильтрация)

Билинейная фильтрация находит четыре пикселя текстуры, ближайшие к текущей точке экрана и результирующий цвет определяется как результат смешения цветов этих пикселей в некоторой пропорции.

Фильтрация методом ближайшего соседа и билинейная фильтрация работают достаточно хорошо когда, во-первых, степень уменьшения текстуры невелика, а во-вторых, когда мы видим текстуру под прямым углом, т.е. фронтально. С чем это связано?

Если рассмотреть, как описывалось выше, "образ" пикселя экрана в текстуре, то для случая сильного уменьшения он будет включать в себя очень много пикселей текстуры (вплоть до всех пикселей!). Кроме того, если мы смотрим на текстуру под углом, этот образ будет сильно вытянут. В обоих случаях описанные методы будут работать плохо, поскольку фильтр не будет "захватывать" соответствующие пиксели текстуры.

Для решения этих проблем применяют так называемый мип-мэппинг и анизотропную фильтрацию.

Mipmap (Мип-мэппинг)

При значительном оптическом уменьшении точке экрана может соответствовать достаточно много пикселей текстуры. Это значит, что реализация даже самого хорошего фильтра будет требовать достаточно много времени для усреднения всех точек. Однако проблему можно решить, если создавать и хранить версии текстуры, в которых значения будут усреднены заранее.

А на этапе визуализации для пикселя искать нужную версию исходной текстуры и брать значение из нее.

Термин mipmap произошел от латинского multum in parvo - многое в малом. При использовании этой технологии в памяти графического ускорителя в дополнение к изображению текстуры хранится набор ее уменьшенных копий, причем каждая новая ровно в два раза меньше предыдущей. Т.е. для текстуры размером 256x256 дополнительно хранятся изображения 128x128, 64x64 и т.д, вплоть до 1x1.

Далее для каждого пикселя выбирается подходящий уровень мипмапа (чем больше размер "образа" пикселя в текстуре, тем меньший мипмап берется). Далее значения в мипмапе могут усредняться билинейно или методом ближайшего соседа (как описано выше) и дополнительно происходит фильтрация между соседними уровнями мипмапа. Такая фильтрация называется трилинейной. Она дает весьма качественные результаты и широко используется на практике.

Рисунок Уровни мипмапа

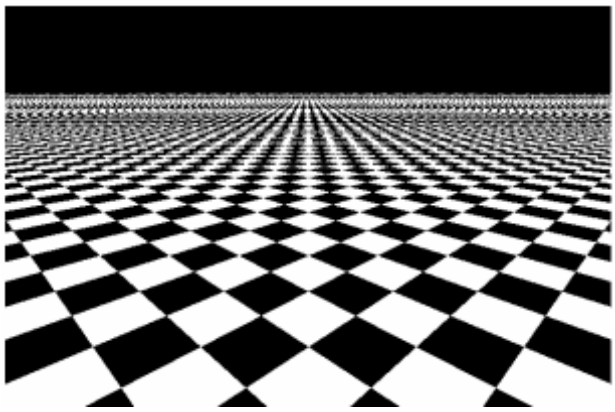
Однако проблема с "вытянутым" образом пикселя в текстуре остается. Как раз по этой причине наша доска на большом расстоянии выглядит очень нечеткой.

Анизотропная фильтрация

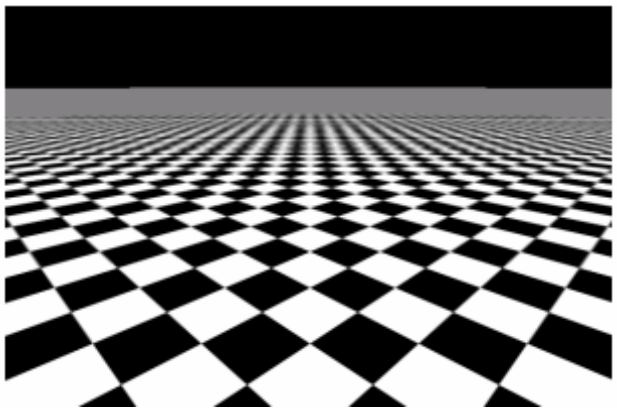
Анизотропная фильтрация - это процесс фильтрации текстуры, специально учитывающий случай вытянутого образа пикселя в текстуре. Фактически, вместо квадратного фильтра (как в билинейной фильтрации), используется вытянутый, что позволяет более качественно выбрать нужный цвет для экранного пикселя. Такая фильтрация используется вместе с мипмэппингом и дает весьма качественные результаты. Однако, существуют и недостатки: реализация анизотропной фильтрации достаточно сложна и при ее включении скорость рисования значительно падает. Анизотропная фильтрация поддерживается последними поколениями графических процессоров NVidia и ATI. Причем с различным уровнем анизотропии - чем больше этот уровень, чем более "вытянутые" образы пикселей можно корректно обрабатывать и тем лучше качество.

Сравнение фильтраций

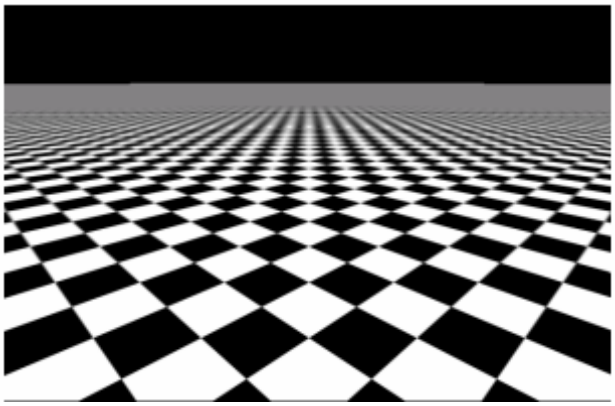
Итог следующий: для подавления артефактов алиасинга текстур аппаратно поддерживаются несколько методов фильтрации, различающиеся по своему качеству и скорости работы. Наиболее простой метод фильтрации - метод ближайшего соседа (который фактически не борется с артефактами, а просто заполняет пиксели). Сейчас чаще всего используется билинейная фильтрация вместе с мип-мэппингом или трилинейная фильтрация. В последнее время графические процессоры начали поддерживать наиболее качественный режим фильтрации - анизотропную фильтрацию.



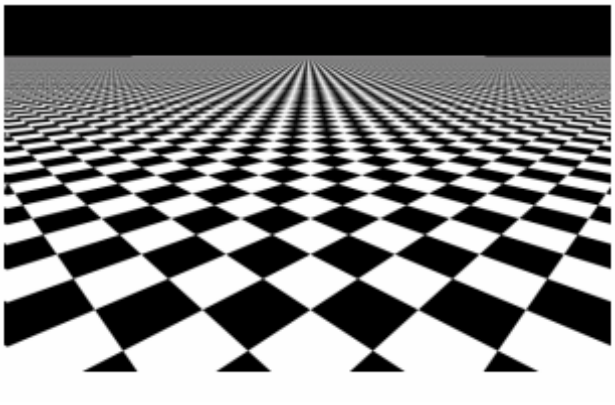
Ближайший сосед



Билинейная



Трилинейная



Анизотропная

1.7 Mapping

Бамп-мэппинг

Бамп-мэппинг (bump mapping) - это тип графических спецэффектов, который призван создавать впечатление "шершавых" или бугристых поверхностей. В последнее время использование бамп-мэппинга стало чуть ли не стандартом игровых приложений.

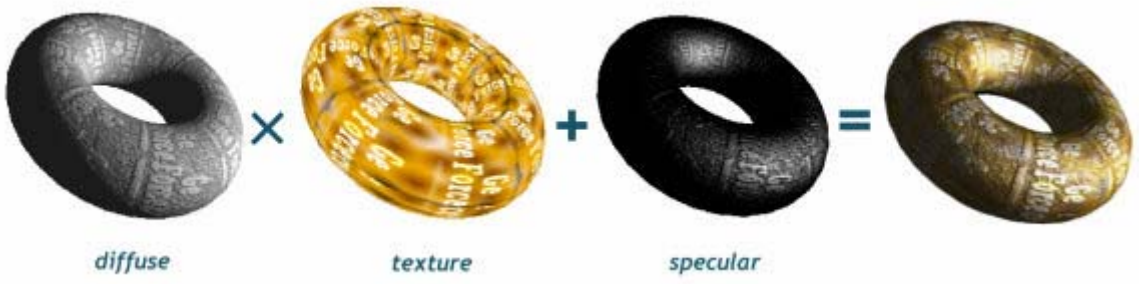
Основная идея бамп-мэппинга - использование текстур для управления взаимодействием света с поверхностью объекта. Это позволяет добавлять мелкие детали без увеличения количества треугольников. В природе мы различаем мелкие неровности поверхностей по теням: любой бугорок будет с одной стороны светлым, а с другой - темным. Фактически, глаз может и не различать изменения в форме поверхности. Этот эффект и используется в технологии бамп-мэппинга. Одна или несколько дополнительных текстур накладываются на поверхность объекта и используются для вычисления освещенности точек объекта. Т.е. поверхность объекта не меняется вовсе, только создается иллюзия неровностей.

Существует несколько методов бамп-мэппинга, но прежде чем мы перейдем к их рассмотрению, необходимо выяснить, собственно как задать неровности на поверхности. Как уже говорилось выше, для этого используются дополнительные текстуры, причем они могут быть разных видов:

Карта нормалей. В этом случае каждый пиксель дополнительной текстуры хранит вектор, перпендикулярный поверхности (нормаль), закодированный в виде цвета. Нормали используются для вычисления освещенности.

Карта смещений. Карта смещений представляет собой текстуру в градациях серого, в каждом пикселе которой хранится смещение от оригинальной поверхности.

Эти текстуры готовятся дизайнерами трехмерных моделей вместе с геометрией и основными текстурами. Существуют и программы, позволяющие получать карты нормалей или смещений автоматически



И посмотрим то же самое на примере игры, Call of Duty 2:

Первый фрагмент картинки - рендеринг без бампмаппинга (нормалмаппинга) вообще, второй (справа-сверху) - бампмаппинг без бликовой составляющей, третий - с бликовой составляющей нормальной величины, какая используется в игре, и последний, справа-снизу - с максимально возможным значением specular составляющей.

Что касается первого аппаратного применения, то некоторые виды бампмаппинга (Emboss Bump Mapping) стали использовать еще во времена видеокарт на базе чипов NVIDIA Riva TNT, однако техники того времени были крайне примитивны и широкого применения не получили. Следующим известным типом стал Environment Mapped Bump Mapping (EMBM), но аппаратной его поддержкой в DirectX в то время обладали только видеокарты Matrox, и опять применение было сильно ограничено. Затем появился Dot3 Bump Mapping и видеочипы того времени (GeForce 256 и GeForce 2) требовали три прохода для того, чтобы полностью выполнить такой математический алгоритм, так как они ограничены двумя одновременно используемыми текстурами. Начиная с NV20 (GeForce3), появилась возможность делать то же самое за один проход при помощи пиксельных шейдеров. Дальше - больше. Стали применять более эффективные техники, такие как Normal Mapping .

Примеры применения бампмаппинга в играх:

Препроцессированный бамп-мэппинг (Pre-calculated bump mapping)

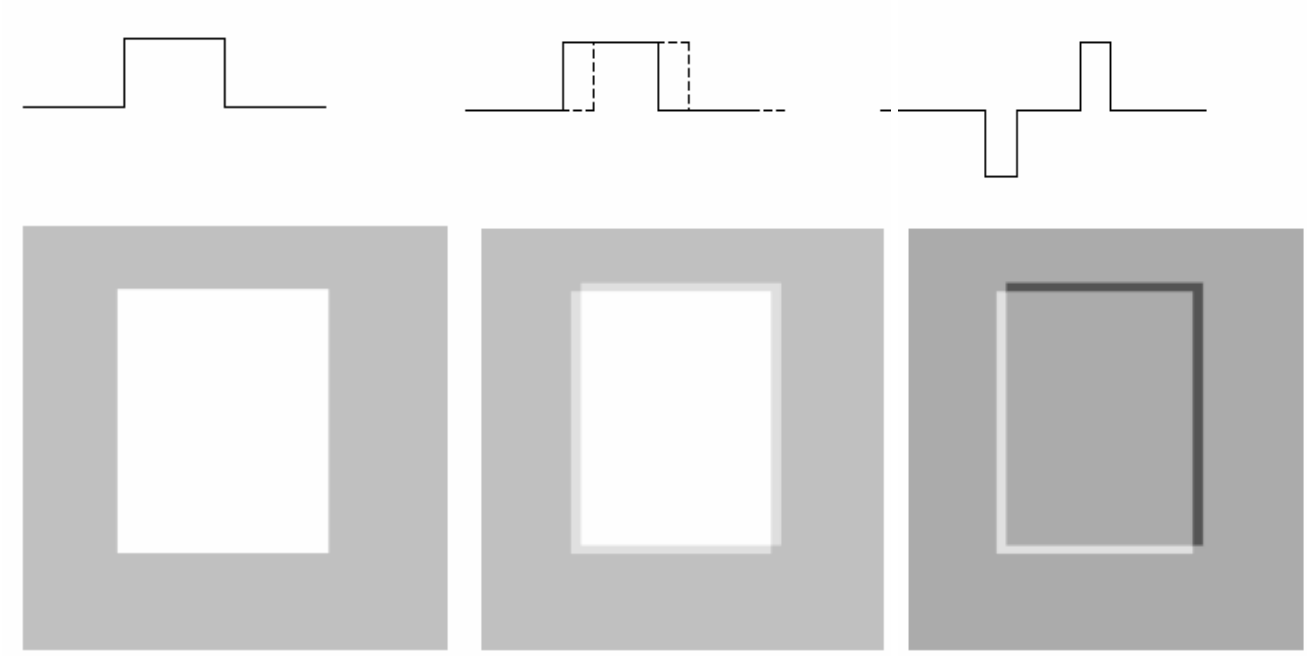
Текстуры, которые будут хранить информацию о поверхности объекта, создаются заранее, до этапа визуализации, путем затемнения некоторых точек текстуры (и, следовательно, самой поверхности) объекта и высветления других. Далее во время рисования используется обычная текстура.

Этот метод не требует никаких алгоритмических ухищрений во время рисования, но, к сожалению, изменений в освещении поверхностей при изменении положений источников света или движения объекта не происходит. А без этого действительно успешной симуляции неровной поверхности не создать. Подобные методы используются для статических частей сцены, часто для архитектуры уровней и т.п

Бамп-мэппинг с помощью тиснения (Emboss bump mapping)

Эта технология применялась на первых графических процессорах (NVidia TNT, TNT2, GeForce). Для объекта создается карта смещений. Рисование происходит в два этапа. На первом этапе карта смещений попиксельно складывается сама с собой. При этом вторая копия сдвигается на небольшое расстояние в направлении источника света. При этом получается следующий эффект: положительные значения разницы определяют освещенные пиксели, отрицательные - пиксели в тени. Эта информация используется для соответствующего изменения цвета пикселей основной текстуры.

Бамп-мэппинг с помощью тиснения не требует аппаратуры, поддерживающей пиксельные шейдеры, однако он плохо работает для относительно крупных неровностей поверхности. Также объекты не всегда выглядят убедительно, это сильно зависит от того, под каким углом смотреть на поверхность.



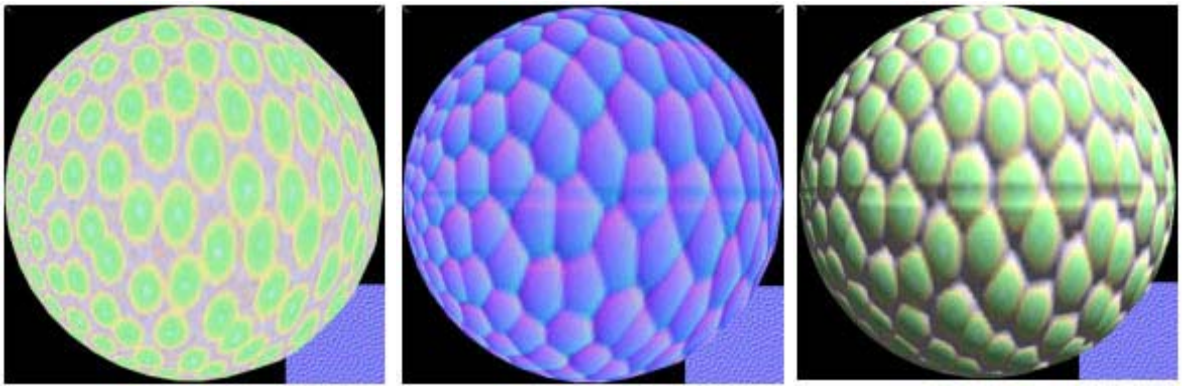
Карта смещений

Карта смещений складывается сама с собой, но сдвинутой на небольшое расстояние

Конечная текстура с эффектом тиснения

Пиксельный бамп-мэппинг (pixel bump mapping)

Пиксельный бамп-мэппинг - на данный момент вершина развития подобных технологий. В этой технологии все вычисляется максимально честно. На вход пиксельному шейдеру дается карта нормалей, из которой берутся значения нормали для каждой точки объекта. Затем значение нормали сравнивается с направлением на источник света и вычисляется значение цвета.

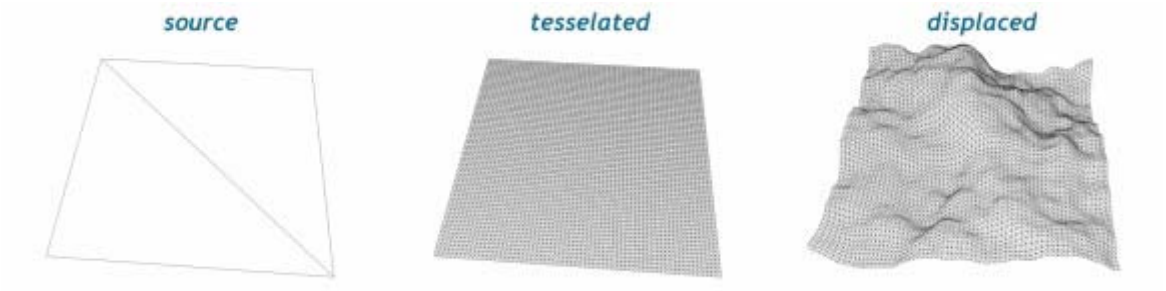


Объект с основной текстурой Карта нормалей Результат бамп-мэппинга

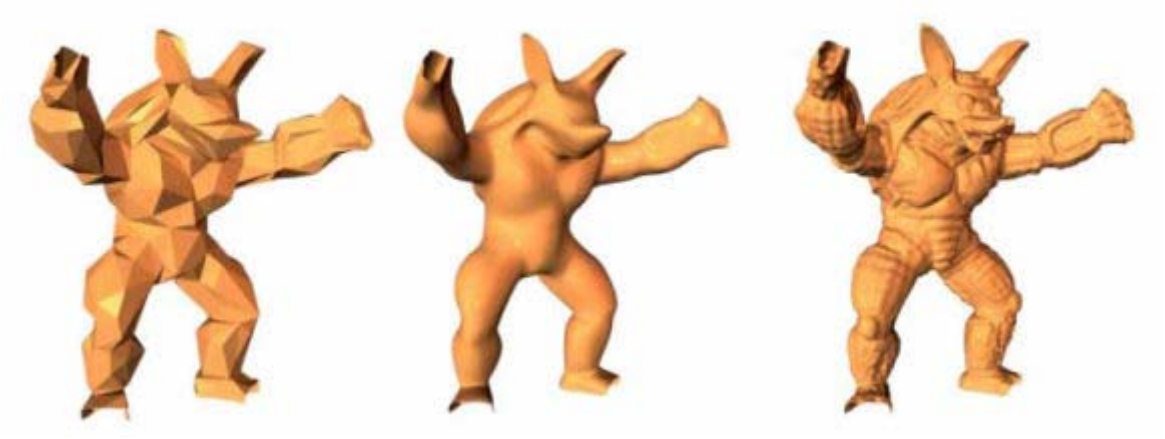
Displacement Mapping

Наложение карт смещения (Displacement Mapping) является методом добавления деталей к трехмерным объектам. В отличие от бампмаппинга и других попиксельных методов, когда картами высот правильно моделируется только освещенность точки, но не изменяется ее положение в пространстве, что дает лишь иллюзию увеличения сложности поверхности, карты смещения позволяют получить настоящие сложные 3D объекты из вершин и полигонов, без ограничений, присущих попиксельным методам. Этот метод изменяет положение вершин треугольников, сдвигая их по нормали на величину, исходя из значений в картах смещения. Карта смещения (displacement map) - это обычно черно-белая текстура, и значения в ней используются для определения высоты каждой точки поверхности объекта (значения могут храниться как 8-битные или 16-битные числа), схоже с bumpmap. Часто карты смещения используются (в этом случае они называются и картами высот) для создания земной поверхности с холмами и впадинами. Так как рельеф местности описывается двухмерной картой смещения, его относительно легко деформировать при необходимости, так как это потребует всего лишь модификации карты смещения и рендеринга на ее основе поверхности в следующем кадре.

Наглядно создание ландшафта при помощи наложения карт смещения представлено на картинке. Исходными были 4 вершины и 2 полигона, в итоге получился полноценный кусок ландшафта.



Большим преимуществом наложения карт смещения является не просто возможность добавления деталей к поверхности, а практически полное создание объекта. Берется низкополигональный объект, разбивается (тесселируется) на большее количество вершин и полигонов. Вершины, полученные в результате тесселяции, затем смещаются по нормали, исходя из значения, прочитанного в карте смещения. Получаем в итоге сложный 3D объект из простого, используя соответствующую displacement карту:



Количество треугольников, созданных при тесселяции, должно быть достаточно большим для того, чтобы передать все детали, задаваемые картой смещений. Иногда дополнительные треугольники создаются автоматически, используя N-патчи или другие методы. Карты смещения лучше использовать совместно с бампмаппингом для создания мелких деталей, где достаточно правильного попиксельного освещения.

Наложение карт смещения впервые получило поддержку в DirectX 9.0. Это была первая версия данного API, которая поддержала технику Displacement Mapping. В DX9 поддерживается два типа наложения карт смещения, filtered и presampled. Первый метод был поддержан забытым уже видеочипом MATROX Parhelia, а второй - ATI RADEON 9700. Filtered метод отличается тем, что позволяет использовать мип-уровни для карт смещения и применять для них трилинейную фильтрацию. В таком методе мип-уровень карты смещения выбирается для каждой вершины на основе расстояния от вершины до камеры, то есть уровень детализации выбирается автоматически. Таким образом достигается почти равномерное разбиение сцены, когда треугольники имеют примерно одинаковый размер.

Наложение карт смещения можно по существу считать методом сжатия геометрии, использование карт смещения снижает объем памяти, требуемый для определенной детализации 3D модели. Громоздкие геометрические данные замещаются простыми двухмерными текстурами смещения, обычно 8-битными или 16-битными. Это снижает требования к объему памяти и пропускной способности, необходимой для доставки геометрических данных к видеочипу, а эти ограничения являются одними из главных для сегодняшних систем. Или же, при равных требованиях к пропускной способности и объему памяти, наложение карт смещения позволяет использовать намного более сложные геометрически 3D модели. Применение моделей значительно меньшей сложности, когда вместо десятков или сотен тысяч треугольников используют единицы тысяч, позволяет еще и ускорить их анимацию. Или же улучшить, применив более сложные комплексные алгоритмы и техники, вроде имитации тканей (cloth simulation).

Другое преимущество в том, что применение карт смещения превращает сложные полигональные трехмерные сетки в несколько двухмерных текстур, которые проще поддаются обработке. Например, для организации Level of Detail можно использовать обычный мип-маппинг для наложения карт смещения. Также, вместо сравнительно сложных алгоритмов сжатия трехмерных сеток можно применять привычные методы сжатия текстур, даже JPEG-подобные. А для процедурного создания 3D объектов можно использовать обычные алгоритмы для двухмерных текстур.

Но у карт смещения есть и некоторые ограничения, они не могут быть применены во всех ситуациях. Например, гладкие объекты, не содержащие большого количества тонких деталей, будут лучше представлены в виде стандартных полигональных сеток или иных поверхностей более высокого уровня, вроде кривых Безье. С другой стороны, очень сложные модели, такие как деревья или растения, также нелегко представить картами смещения. Есть также проблемы удобства их применения, это почти всегда требует специализированных утилит, ведь очень сложно напрямую создавать карты смещения (если речь не идет о простых объектах, вроде ландшафта). Многие проблемы и ограничения, присущие картам смещения, совпадают с таковыми

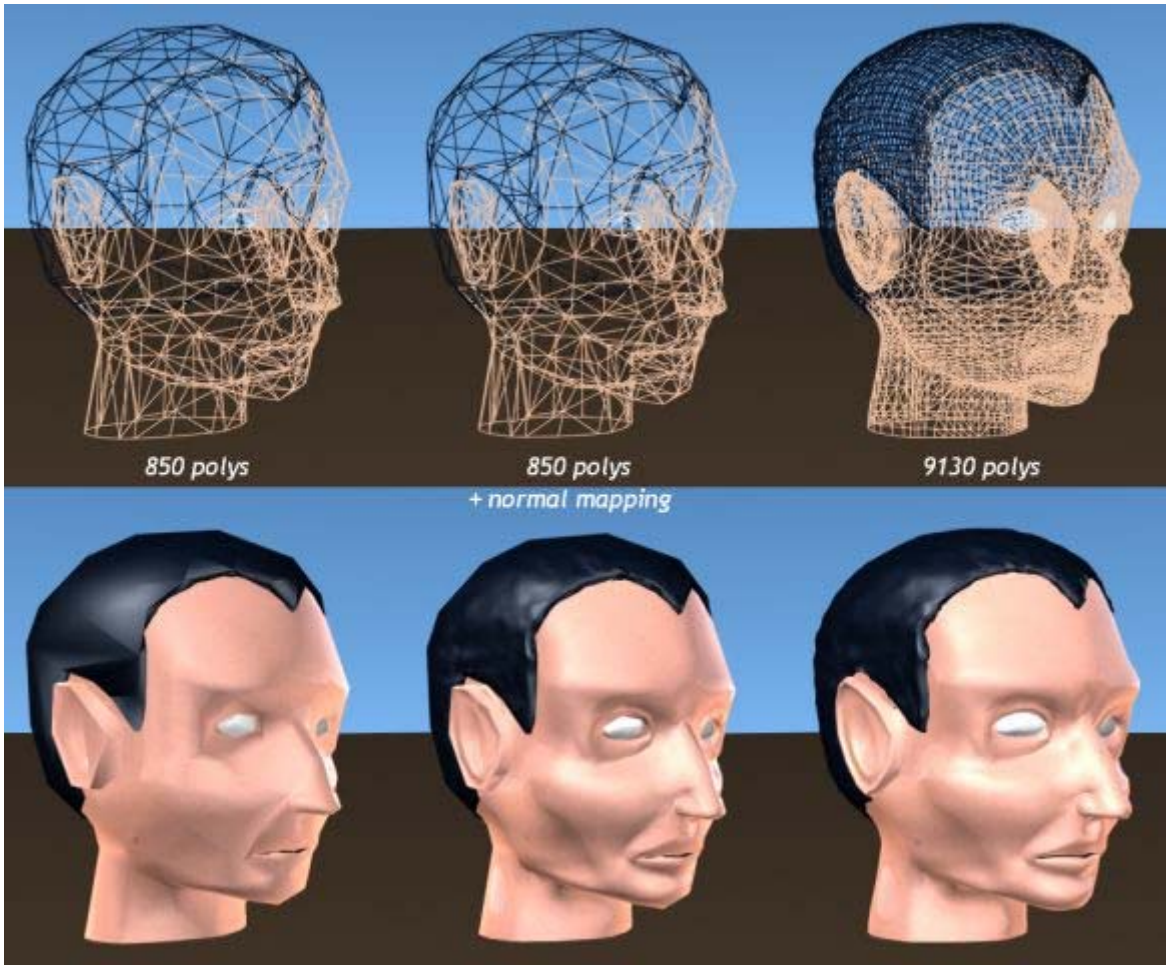
у наложения карт нормалей , поскольку эти два метода по сути - два разных представления похожей идеи.

В качестве примера из реальных игр приведу игру, в которой используется выборка текстур из вершинного шейдера, возможность, появившаяся в видеочипах NVIDIA NV40 и шейдерной модели 3.0. Вершинное текстурирование можно применить для полностью выполняемого видеочипом простого метода наложения карт смещения, без тесселяции (разбиения на большее количество треугольников). Применение такому алгоритму ограничено, они имеют смысл, только если карты будут динамическими, то есть, будут изменяться в процессе. Например, это рендеринг больших водных поверхностей, что и сделано в игре Pacific Fighters:


Normal Mapping

Нормалмаппинг - это улучшенная разновидность техники бампмаппинга, описанной ранее, расширенная ее версия. Бампмаппинг был разработан Блинном (Blinn) еще в 1978 году, нормали поверхности при этом методе наложения рельефа изменяются на основе информации из карт высот (bump map). В то время как бампмаппинг всего лишь изменяет существующую нормаль для точек поверхности, нормалмаппинг полностью заменяет нормали при помощи выборки их значений из специально подготовленной карты нормалей (normal map). Эти карты обычно являются текстурами с сохраненными в них заранее просчитанными значениями нормалей, представленными в виде компонент цвета RGB (впрочем, есть и специальные форматы для карт нормалей, в том числе со сжатием), в отличие от 8-битных черно-белых карт высот в бампмаппинге.

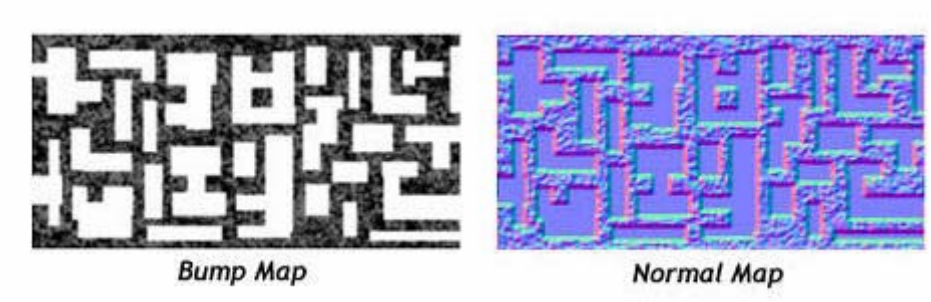
В общем, как и бампмаппинг, это тоже "дешевый" метод для добавления детализации к моделям сравнительно низкой геометрической сложности, без использования большего количества реальной геометрии, только более продвинутый. Одно из наиболее интересных применений техники - существенное увеличение детализации низкополигональных моделей при помощи карт нормалей, полученных обработкой такой же модели высокой геометрической сложности. Карты нормалей содержат более подробное описание поверхности, по сравнению с бампмаппингом и позволяют представить более сложные формы. Идеи по получению информации из высокодетализированных объектов были озвучены в середине 90-х годов прошлого века, но тогда речь шла об использовании для Displacement Mapping . Позднее, в 1998 году, были представлены идеи о перенесении деталей в виде карт нормалей от высокополигональных моделей в низкополигональные.



Карты нормалей предоставляют более эффективный способ для хранения подробных данных о поверхностях, по сравнению с простым использованием большого количества полигонов. Единственное серьезное их ограничение в том, что они не очень хорошо подходят для крупных деталей, ведь нормалмаппинг на самом деле не добавляет полигонов и не изменяет форму объекта, он только создает видимость этого. Это всего лишь симуляция деталей, на основе расчета освещения на пиксельном уровне. На крайних полигонах объекта и больших углах наклона поверхности это очень хорошо заметно. Поэтому наиболее разумный способ применения нормалмаппинга состоит в том, чтобы сделать низкополигональную модель достаточно детализированной для того, чтобы сохранялась основная форма объекта, и использовать карты нормалей для добавления более мелких деталей.

Карты нормалей обычно создаются на основе двух версий модели, низко- и высокополигональной. Низкополигональная модель состоит из минимума геометрии, основных форм объекта, а высокополигональная содержит все необходимое для максимальной детализации. Затем, при помощи специальных утилит они сравниваются друг с другом, разница рассчитывается и сохраняется в текстуре, называемой картой нормалей. При ее создании дополнительно можно использовать и bump map для очень мелких деталей, которые даже в высокополигональной модели не смоделировать (поры кожи, другие мелкие углубления).

Карты нормалей изначально были представлены в виде обычных RGB текстур, где компоненты цвета R, G и B (от 0 до 1) интерпретируются как координаты X, Y и Z. Каждый тексель в карте нормалей представлен как нормаль точки поверхности. Карты нормалей могут быть двух видов: с координатами в model space (общей системе координат) или tangent space (термин на русском - "касательное пространство", локальная система координат треугольника). Чаще применяется второй вариант. Когда карты нормалей представлены в model space, то они должны иметь три компоненты, так как могут быть представлены все направления, а когда в локальной системе координат tangent space, то можно обойтись двумя компонентами, а третью получить в пиксельном шейдере.



Современные приложения реального времени до сих пор сильно проигрывают пререндеренной анимации по качеству изображения, это касается, прежде всего, качества освещения и геометрической сложности сцен. Количество вершин и треугольников, рассчитываемых в реальном времени, ограничено. Поэтому очень важны методы, позволяющие снизить количество геометрии. До нормалмаппинга были разработаны несколько таких методов, но низкополигональные модели даже с бампмаппингом получаются заметно хуже более сложных моделей. Нормалмаппинг хоть и имеет несколько недостатков (самый явный - так как модель остается низкополигональной, это легко видно по ее угловатым границам), но итоговое качество рендеринга заметно улучшается, оставляя геометрическую сложность моделей низкой. В последнее время хорошо видно увеличение популярности данной методики и использование ее во всех популярных игровых движках. "Виной" этому - комбинация отличного результирующего качества и одновременное снижение требований к геометрической сложности моделей. Техника нормалмаппинга сейчас применяется почти повсеместно, все новые игры используют ее максимально широко. Вот лишь краткий список известных ПК игр с использованием нормалмаппинга: Far Cry, Doom 3, Half-Life 2, Call of Duty 2, F.E.A.R., Quake 4. Все они выглядят намного лучше, чем игры прошлого, в том числе из-за применения карт нормалей.

Parallax Mapping/Offset Mapping

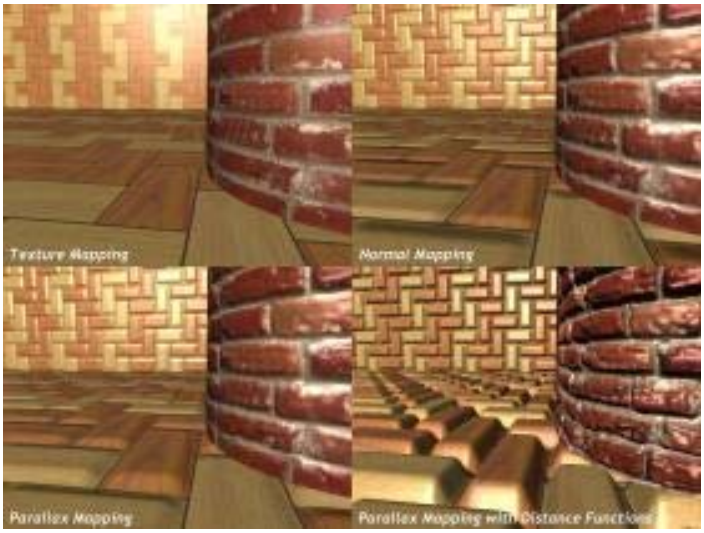
После нормалмаппинга, разработанного еще в 1984 году, последовало рельефное текстурирование (Relief Texture Mapping), представленное Olivera и Bishop в 1999 году. Это метод для наложения текстур, основанный на информации о глубине. Метод не нашел применения в играх, но его идея способствовала продолжению работ над параллаксмаппингом и его улучшении. Kaneko в 2001 представил parallax mapping, который стал первым эффективным методом для попиксельного отображения эффекта параллакса. В 2004 году Welsh продемонстрировал применение параллаксмаппинга на программируемых видеочипах.

У этого метода, пожалуй, больше всего различных названий. Перечислю те, которые встречал: Parallax Mapping, Offset Mapping, Virtual Displacement Mapping, Per-Pixel Displacement Mapping. В статье для краткости применяется первое название.

Параллаксмаппинг - это еще одна альтернатива техникам бампмаппинга и нормалмаппинга, которая дает еще большее представление о деталях поверхности, более натуралистичное отображение 3D поверхностей, также без слишком больших потерь производительности. Это техника похожа одновременно на наложение карт смещения и нормалмаппинг, это нечто среднее между ними. Метод тоже предназначен для отображения большего количества деталей поверхности, чем есть в исходной геометрической модели. Он похож на нормалмаппинг, но

отличие в том, что метод искажает наложение текстуры, изменяя текстурные координаты так, что когда вы смотрите на поверхность под разными углами, она выглядит выпуклой, хотя в реальности поверхность плоская и не изменяется. Иными словами, Parallax Mapping - это техника аппроксимации эффекта смещения точек поверхности в зависимости от изменения точки зрения.

Техника сдвигает текстурные координаты (поэтому технику иногда называют offset mapping) так, чтобы поверхность выглядела более объемной. Идея метода состоит в том, чтобы возвращать текстурные координаты той точки, где видовой вектор пересекает поверхность. Это требует просчета лучей (рейтрейсинг) для карты высот, но если она не имеет слишком сильно изменяющихся значений ("гладкая" или "плавная"), то можно обойтись аппроксимацией. Такой метод хорош для поверхностей с плавно изменяющимися высотами, без просчета пересечений и больших значений смещения. Подобный простой алгоритм отличается от нормалмаппинга всего тремя инструкциями пиксельного шейдера: две математические инструкции и одна дополнительная выборка из текстуры. После того, как вычислена новая текстурная координата, она используется дальше для чтения других текстурных слоев: базовой текстуры, карты нормалей и т.п. Такой метод параллаксмаппинга на современных видеочипах почти также эффективен, как обычное наложение текстур, а его результатом является более реалистичной отображение поверхности, по сравнению с простым нормалмаппингом.



Но использование обычного параллаксмаппинга ограничено картами высот с небольшой разницей значений. "Крутые" неровности обрабатываются алгоритмом некорректно, появляются различные артефакты, "плавание" текстур и пр. Было разработано несколько модифицированных методов для улучшения техники параллаксмаппинга. Несколько исследователей (Yerex, Donnelly, Tatarchuk, Policarpo) описали новые методы, улучшающие начальный алгоритм. Почти все идеи основаны на трассировке лучей в пиксельном шейдере для определения пересечений деталей поверхностей друг другом. Модифицированные методики получили несколько разных названий: Parallax Mapping with Occlusion, Parallax Mapping with Distance Functions, Parallax Occlusion Mapping. Для краткости будем их все называть Parallax Occlusion Mapping.

Методы Parallax Occlusion Mapping включают еще и трассировку лучей для определения высот и учета видимости текселей. Ведь при взгляде под углом к поверхности тексели загораживают друг друга, и, учитывая это, можно добавить к эффекту параллакса больше глубины. Получаемое изображение становится реалистичнее и такие улучшенные методы можно применять для более глубокого рельефа, он отлично подходит для изображения кирпичных и каменных стен, мостовой и пр. Нужно особенно отметить, что главное отличие Parallax Mapping от Displacement Mapping в том, что расчеты все попиксельные, а не повершинные. Именно поэтому метод имеет названия вроде Virtual Displacement Mapping и Per-Pixel Displacement Mapping. Посмотрите на картинку, трудно поверить, но камни мостовой тут - всего лишь попиксельный эффект:

Метод позволяет эффективно отображать детализированные поверхности без миллионов вершин и треугольников, которые потребовались бы при реализации этого геометрией. При этом сохраняется высокая детализация (кроме силуэтов/граней) и значительно упрощаются расчеты анимации. Такая техника дешевле, чем использование реальной геометрии, используется значительно меньшее количество полигонов, особенно в случаях с очень мелкими деталями. Применений алгоритму множество, а лучше всего он подходит для камней, кирпичей и подобного.

Также, дополнительное преимущество в том, что карты высот могут динамически изменяться (поверхность воды с волнами, дырки от пуль в стенах и многое другое). В недостатках метода - отсутствие геометрически правильных силуэтов (краев объекта), ведь алгоритм попиксельный и не является настоящим displacement mapping. Зато он экономит производительность в виде снижения нагрузки на трансформацию, освещение и анимацию геометрии. Экономит видеопамять, необходимую для хранения больших объемов геометрических данных. В плюсах у техники и относительно простая интеграция в существующие приложения и использование в процессе работы привычных утилит, применяемых для нормалмаппинга.

Техника уже применяется в реальных играх последнего времени. Пока что обходятся простым параллаксмаппингом на основе статических карт высот, без трассировки лучей и расчета пересечений. Вот примеры применения параллаксмаппинга в играх:

1.8 Shader (Шейдер).

Шейдер – программysq модуль для визуального определения поверхности объекта.

Это может быть описание освещения, текстурирования, постобработки и т.п. Шейдеры выросли из работ Кука (Cook's shade trees) и Перлина (Perlin’s pixel stream language). Сейчас наиболее известны шейдеры RenderMan Shading Language. Программируемые шейдеры были впервые представлены в RenderMan компании Pixar, там определены несколько типов шейдеров: light source shaders, surface shaders, displacement shaders, volume shaders, imager shaders. Эти шейдеры чаще всего программно выполняются универсальными процессорами и не имеют полной аппаратной реализации. В дальнейшем, многие исследователи описывали похожие на RenderMan языки, но они уже были предназначены для аппаратного ускорения: система PixelFlow (Olano и Lastra), Quake Shader Language (применен id Software в графическом движке игры Quake III, который описывал многопроходный рендеринг), и другие. Peercy сотоварищи разработали технику для того, чтобы программы с циклами и условиями выполнять на традиционных

аппаратных архитектурах при помощи нескольких проходов рендеринга. Шейдеры RenderMan разбивались на несколько проходов, которые комбинировались во фреймбуфере. Позднее появились языки, которые мы видим аппаратно ускоренными в DirectX и OpenGL. Так шейдеры были адаптированы для графических приложений реального времени.

Видеочипы раннего времени не были программируемы и исполняли только заранее запрограммированные действия (fixed-function), например, алгоритм освещения был жестко зафиксирован в железе, и нельзя было ничего изменить. Затем, компании-производители видеочипов постепенно ввели в свои чипы элементы программируемости, сначала это были очень слабые возможности (NV10, известный как NVIDIA GeForce 256, уже был способен на некоторые примитивные программы), которые не получили программной поддержки в Microsoft DirectX API, но со временем возможности постоянно расширялись. Следующий шаг был за и NV20 (GeForce 3) и NV2A (видеочип, примененный в игровой консоли Microsoft Xbox), которые стали первыми чипами с аппаратной поддержкой шейдеров DirectX API. Версия Shader Model 1.0/1.1, появившаяся в DirectX 8, была сильно ограничена, каждый шейдер (особенно это относится к пиксельным) мог быть сравнительно малой длины и сочетать весьма ограниченный набор команд. В дальнейшем, Shader Model 1 (SM1 для краткости) была улучшена с пиксельными шейдерами версии 1.4 (ATI R200), которые предлагали большую гибкость, но также имели слишком ограниченные возможности. Шейдеры того времени писались на так называемом assembly shader language, который близок к ассемблеру для универсальных процессоров. Его низкий уровень доставляет определенные сложности для понимания кода и программирования, особенно, когда код программы большой, ведь он далек от элегантности и структурированности современных языков программирования.

Версия Shader Model 2.0 (SM2), появившись в DirectX 9 (что было поддержано видеочипом ATI R300, ставшим первым GPU с поддержкой шейдерной модели версии 2.0), серьезно расширила возможности шейдеров реального времени, предложив более длинные и сложные шейдеры и заметно расширившийся набор команд. Была добавлена возможность расчетов с плавающей запятой в пиксельных шейдерах, что также стало важнейшим улучшением. DirectX 9, в лице возможностей SM2, также привнес и язык шейдеров высокого уровня - High-Level Shader Language (HLSL), весьма похожий на язык Си. И эффективный компилятор, переводящий HLSL программы в низкоуровневый код, "понятный" для аппаратных средств. Причем, доступно несколько профилей, предназначенных для разных аппаратных архитектур. Теперь, разработчик может писать один код HLSL шейдера и компилировать его при помощи DirectX в оптимальную программу, для установленного у пользователя видеочипа. После этого выходили чипы от NVIDIA, NV30 и NV40, которые улучшили возможности аппаратных шейдеров еще на шаг, добавив еще более длинные шейдеры, возможности динамических переходов в вершинных и пиксельных шейдерах, возможность выборки текстур из вершинных шейдеров и др. С тех пор пока качественных изменений не было, они ожидаются ближе к концу 2006 года в DirectX 10...

В целом, шейдеры добавили к графическому конвейеру множество новых возможностей по трансформации и освещению вершин и индивидуальной обработке пикселей так, как этого хотят разработчики каждого конкретного приложения. И все-таки, возможности аппаратных шейдеров до сих пор не раскрыты в приложениях полностью, а ведь с увеличением их возможностей в каждом новом поколении "железа", мы скоро увидим уровень тех самых шейдеров RenderMan, которые когда-то казались недостижимыми для игровых видеоускорителей. Пока в шейдерных моделях реального времени, поддерживаемых на сегодняшний день аппаратными видеоускорителями, определено лишь два типа шейдеров: Vertex Shader и Pixel Shader (в определении DirectX 9 API). В будущем DirectX 10 к ним обещает добавиться еще и Geometry Shader.

Vertex Shader (Вершинный Шейдер)

Вершинные шейдеры - это программы, выполняемые видеочипами, которые производят математические операции с вершинами (vertex, из них состоят 3D объекты в играх), иначе говоря, они предоставляют возможность выполнять программируемые алгоритмы по изменению параметров вершин и их освещению (T&L - Transform & Lighting). Каждая вершина определяется несколькими переменными, например, положение вершины в 3D пространстве определяется координатами: x, y и z. Вершины также могут быть описаны характеристиками цвета, текстурными координатами и т.п. Вершинные шейдеры, в зависимости от алгоритмов, изменяют эти данные в процессе своей работы, например, вычисляя и записывая новые координаты и/или цвет. То есть, входные данные вершинного шейдера - данные об одной вершине геометрической модели, которая в данный момент обрабатывается. Обычно это координаты в пространстве, нормаль, компоненты цвета и текстурные координаты. Результирующие данные выполняемой программы служат входными для дальнейшей части конвейера, растеризатор делает линейную интерполяцию входных данных для поверхности треугольника и для каждого пикселя исполняет соответствующий пиксельный шейдер. Очень простой и грубый (но наглядный, надеюсь) пример: вершинный шейдер позволяет взять 3D объект сферы и вершинным шейдером сделать из него зеленый куб :).

До появления видеочипа NV20 у разработчиков было два пути, либо использовать собственные программы и алгоритмы, изменяющие параметры вершин, но тогда все расчеты делал бы CPU (software T&L), либо полагаться на фиксированные алгоритмы в видеочипах, с поддержкой аппаратной трансформации и освещения (hardware T&L). Первая же шейдерная модель DirectX означала большой шаг вперед от фиксированных функций по трансформации и освещению вершин к полностью программируемым алгоритмам. Стало возможным, например, выполнять алгоритм скининга полностью на видеочипах, а до этого единственной возможностью было их исполнение на универсальных центральных процессорах. Теперь, с сильно улучшенными со времен упомянутого чипа NVIDIA возможностями, с вершинами при помощи вершинных шейдеров можно делать уже очень многое (кроме их создания, разве что)...

Примеры того, как и где применяются вершинные шейдеры:

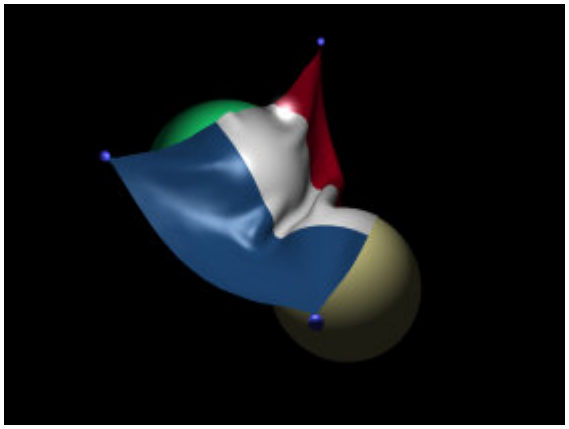
Скининг (skinning). Matrix pallete skinning для скелетной анимации персонажей с большим количеством "костей". Примеры вы видите практически во всех играх. Но приведу один скриншот из Call of Duty 2, над вершинами каждого из персонажей поработал алгоритм скининга. Причем, с шейдерами версии 3.0 сделать скининг стало заметно проще, для шейдеров версии 1.1 нужно было писать несколько шейдеров для каждого вида скининга (с определенным количеством "костей").

Деформация объектов. Как самый явный и эффектный пример - создание реалистичных волн в динамике. Примеры подобных решений наблюдаются в играх F.E.A.R. и Pacific Fighters, причем в последнем сделана, пожалуй, самая реалистичная вода реального времени, применяются вершинные шейдеры 3.0 и доступ к текстурам из них, настоящий Displacement Mapping в дополнение к Bump Mapping:

Конечно, похожий эффект волн в динамике, как в F.E.A.R., может быть запрограммирован и на пиксельном уровне (Morrowind), но в данном случае речь об изменении реальной геометрии, что всегда реалистичнее выглядит.

Анимация объектов. Например, травы и деревьев в одном из первых применений - 3DMark 2001 SE, алгоритм анимации был значительно улучшен в следующем 3DMark 03:

Имитация ткани (Cloth Simulation) - для имитации поведения подобных ткани материалов, которой очень не хватает в большинстве игр. Наиболее просто понять, о чем речь, по такой картинке:



Toon shading/Cel shading. Используется в некоторых играх для создания специального эффекта "мультяшного" изображения:


Pixel Shader (Пиксельный Шейдер)

Пиксельные шейдеры - это программы, выполняемые видеочипом во время растеризации для каждого пикселя изображения, они производят выборку из текстур и/или математические операции над цветом и значением глубины (Z-buffer) пикселей. Все инструкции пиксельного шейдера выполняются попиксельно, после того, как операции с трансформированием и освещением геометрии завершены. Пиксельный шейдер в итоге своей работы выдает конечное значение цвета пикселя и Z-значение для последующего этапа графического конвейера, блендинга. Наиболее простой пример пиксельного шейдера, который можно привести: банальное мультитекстурирование, просто смешение двух текстур (diffuse и lightmap, например) и наложение результата вычисления на пиксель.

До появления видеочипов с аппаратной поддержкой пиксельных шейдеров, у разработчиков были лишь возможности по обычному мультитекстурированию и альфа-блендингу, что существенно ограничивало возможности по многим визуальным эффектам и не позволяло делать многое из того, что сейчас доступно. И если с геометрией еще что-то можно было делать программно, то с пикселями - нет. Ранние версии DirectX (до 7.0 включительно) всегда выполняли все расчеты повершинно и предлагали крайне ограниченную функциональность по попиксельному освещению (вспоминаем EMBM - environment bump mapping и DOT3) в последних версиях. Пиксельные шейдеры сделали возможным освещение любых поверхностей попиксельно, используя запрограммированные разработчиками материалы. Появившиеся в NV20 пиксельные шейдеры версии 1.1 (в понимании DirectX) уже могли не только делать мультитекстурирование, но и многое другое, хотя большинство игр, использующих SM1, просто использовали традиционное мультитекстурирование на большинстве поверхностей, выполняя более сложные пиксельные шейдеры лишь на части поверхностей, для создания разнообразных спецэффектов (все знают, что вода до сих пор является наиболее частым примером использования пиксельных шейдеров в играх). Сейчас, после появления SM3 и поддерживающих их видеочипов, возможности пиксельных шейдеров доросли уже до того, чтобы с их помощью делать даже трассировку лучей (raytracing), пусть пока с некоторыми ограничениями.

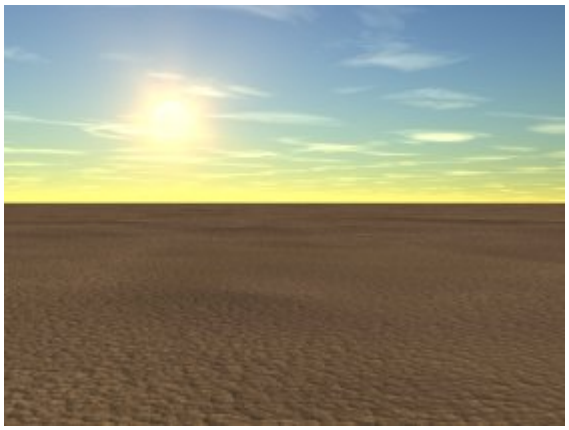
Примеры применения пиксельных шейдеров:

- **Мультитекстурирование**. Несколько слоев текстур (colormap, detailmap, lightmap и т.д.). Используется вообще во всех играх.


Попиксельное освещение. Bump mapping . Normal mapping . С недавних пор применяется практически везде.


Постобработка кадра. Все эти эффекты Bloom , Depth of Field и Motion Blur ...


Процедурные текстуры , такие, как текстура дерева или мрамора. Примеры:



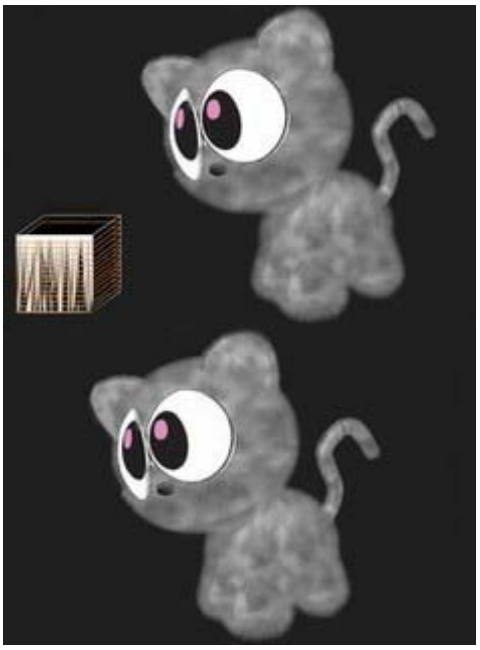

Geometry Shader (Геометрические шейдеры)

Геометрический шейдер, также как и вершинный, имеет дело с вершинами полигонов, однако в отличие от последнего его назначение — генерация геометрических данных «на лету».

«Геометрия», с которой приходится работать системе рендеринга, состоит из двух компонентов: вершин и связей между ними (ребер, граней). Вершинные шейдеры, как ясно из их названия, работают исключительно с вершинами, с их координатами, нормалями, цветом, но они не имеют абсолютно никакой информации о том, к каким треугольникам принадлежит данная вершина и какие вершины являются ее соседями. Кроме того, вершинный шейдер не может удалить вверенную ему в этом вызове вершину или добавить новую, он может только изменять ее позицию и другие атрибуты.

После того как вершинные шейдеры отработают все объекты, и персонажи окажутся в нужных местах и нужных позах, данный результат передает геометрическому шейдеру, который, в отличие от вершинного, оперирует ребрами и треугольниками.

Это открывает массу интересных возможностей. Самое простое, что сразу же приходит в голову, – это определение контуров объекта. Действительно, если у нас есть информация о ребре и двух прилежащих к нему треугольниках, то мы легко можем определить, лежит ли это ребро на контуре объекта или нет: если один из треугольников соседей «смотрит» на нас, а второй «не смотрит», то ребро лежит на контуре, если нет – то нет. Зачем вообще может понадобиться определение контура? Хотя бы для того, чтобы нарисовать мех.



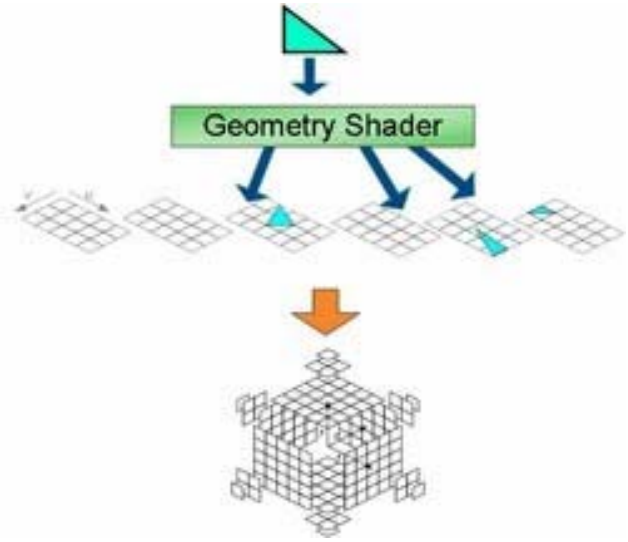
Тоесть с начала рисуем «оболочки» вокруг объекта, «Эффект меха» присутствует, но по контурам объекта он выглядит весьма неряшливо, с помощью геометрического шейдера находим контуры и обрисовываем их «ворсинками», результат – вполне реалистичный и аккуратный мех.

Аналогичным образом геометрические шейдеры могут использоваться для построения «теневых объемов». На сегодняшний день существуют два основных метода создания теней: с помощью карт теней и с помощью теневых объемов. Каждый из них имеет свои достоинства и недостатки. Карты теней, к примеру, не боятся самых сложных сцен с очень большим количеством деталей и самозатенением объектов, зато они нередко являются причиной дефектов изображения и хуже работают на больших расстояниях, чем теневые объемы.

Идея метода теневых объемов состоит в том, чтобы взять плоскую фигуру – контур затеняющего объекта – и «выдавливанием» превратить его в трехмерную (это, собственно, и будет «теневой объем»). Затем, определяя места пересечения теневого объема и других объектов, можно определять, какие пиксели на экране будут затенены.

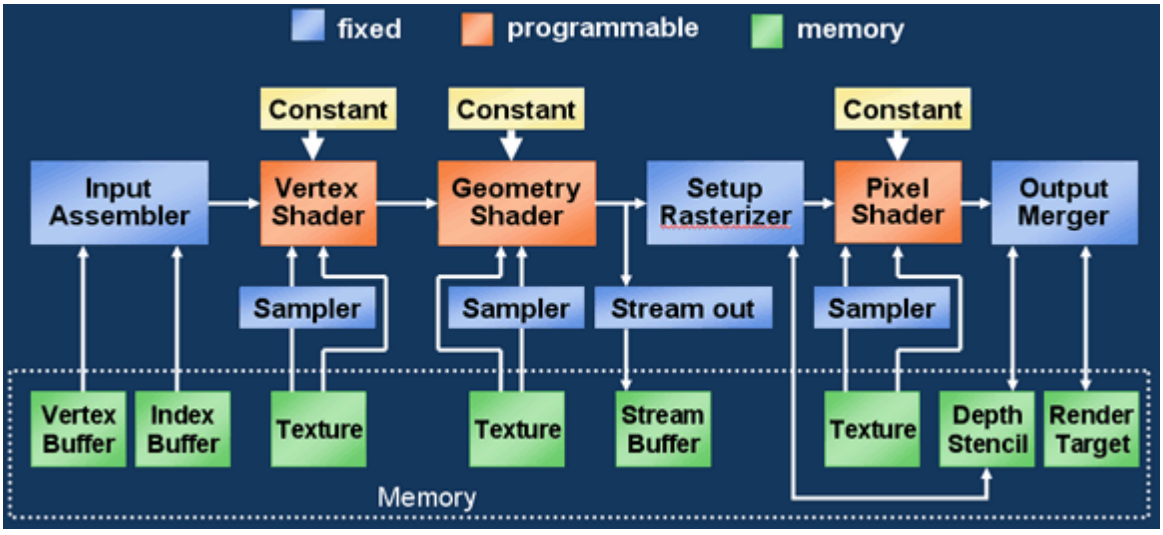
Теневые объемы, к примеру, применяются в Doom 3 и играх на основе его движка – его характерной особенностью являются очень ровные и резкие края теней, хотя в последнее время их и научились искусственно сглаживать. Вплоть до появления DirectX 10 использование теневых объемов было достаточно проблематичным из-за того, что большую часть обсчета теней приходилось осуществлять на центральном процессоре. А это, помимо прочего, означает, что CPU приходилось еще и лишний раз делать скиннинг моделей, то есть дублировать работу, выполняемую вершинными шейдерами на GPU. Разумеется, такое затенение было ресурсоемким и недостаточно качественным: чтобы сильно не напрягать процессор, для скиннинга и последующего создания теневых объемов персонажей использовали упрощенные модели. Теперь, с переносом всех вычислений на графическую карту, все будет намного эффективнее, логичнее и удобнее.

Тем разработчикам, которые предпочитают использовать вместо теневых объемов карты теней, геометрические шейдеры также могут помочь. Благодаря возможности создания нескольких треугольников из одного теперь можно будет рендерить по нескольку теневых карт (в том числе и кубические текстуры) за один проход. К



Кубическая текстура – это шесть текстур, как бы «сложенных» в виде кубика. Такие карты часто используются в качестве теневых, однако еще чаще их применяют для симуляции отражающих поверхностей. Скажем, когда вы едете на автомобиле в Need for Speed, игра несколько раз в секунду (частота зависит от настроек качества) рендерит упрощенную версию окружения в кубическую текстуру, а потом накладывает ее на поверхность автомобиля, создавая эффект глянцевой поверхности, отражающей все вокруг. Благодаря геометрическим шейдерам кубические текстуры можно будет рендерить намного быстрее, эффективнее и с большим количеством деталей, это значит, что отражение теперь не будет меняться рывками, а количество объектов увеличится (сейчас это обычно только здания и огни, но не машины конкурентов).

1.9 Основной принцип работы Графического конвейера.



Теперь давайте, в общих чертах, попробуем разобраться, как это все работает. Данная унифицированная схема графического конвейера позволяет получать доступ к текстурам на трех стадиях конвейера — на стадии вершинного шейдера, затем на стадии геометрического шейдера и на стадии пиксельного шейдера, разумеется. Везде доступны сэмплеры, позволяющие выбрать сразу несколько соседних значений из текстуры по одним рассчитанным текстурным координатам, что может существенно упростить реализацию собственных алгоритмов фильтрации или работы с нетривиальными представлениями данных и всяческими специальными картами.

После уже привычной стадии вершинного шейдера следует геометрический шейдер (Geometry Shader). Этому шейдер, доступны уже все собранные из вершин треугольники перед отрисовкой, как целостные объекты. Т.е. он может производить какие-либо операции над треугольниками целиком. В том числе, учитывая какие-то контрольные или дополнительные параметры вершин. Можно изменить параметры, можно рассчитать новые, специфичные для всего треугольника, и передать их затем в пиксельный шедер. Можно пометить треугольник (чтобы затем обработать его по разному, в зависимости от значения метки), или выбросить его из кандидатов на отрисовку. К сожалению этот шейдер не может, как ожидалось ранее, произвольно создавать новую геометрию и новые треугольники на выходе, но сразу за ним следует еще одна новая стадия — вывод потока (Stream Out).

Кроме того, данные могут быть повторно возвращены обратно в буфер (память). Прошедшие обработку в вершинной части конвейера, до передачи и отрисовки их в пиксельной части. Затем их можно снова использовать по своему усмотрению. Таким образом можно реализовать новую геометрию, в том числе реализовать тесселяцию поверхностей на большее кратное число треугольников по тому или иному алгоритму. Для этого надо воспользоваться двумя проходами вершинной части, где первый экспортирует поток данных и коэффицентами деления у потоков между ними (помните, еще в DX 9c появилась возможность устанавливать коэффициент, на который будут делиться индексы вершин при собирании их из нескольких независимых потоков, отдельный для каждого потока). Это несколько менее естественно, чем просто генерация новых вершин и новой геометрии в вершинном шейдере, но теперь такая возможность существует. Можно даже сразу зациклить данные по маршруту выход -> вход и воспользовавшись метками осуществить циклический отбор и деградирование геометрии, например, упростив модель.

1.10 Аппаратно программные интерфейсы (API)

Аббревиатура API означает Application Programming Interface, что переводится, как интерфейс прикладного программирования. Различные 3D API (Direct3D (составная часть DirectX), OpenGL, Glide, Metal и многие другие) предназначены в основном для унификации и облегчения разработки приложений, использующих трехмерную графику и игры.

Поясняю на примере. С каждым годом выходят всё новые и новые видеокарты, и, кроме того, на рынке итак уже очень много разношёрстных старых карт. Для того чтобы, программист смог использовать все новые технологии и особенности, заложенные в определённой модели, ему необходимо искать документацию и особенности по ней. Согласитесь, очень сложно делать поддержку для n моделей старых видеокарт, и ещё учитывать то, что продукт должен прекрасно идти на последующих m новых видеокартах.

Для решения этой проблемы и были придуманы 3D API. С API работать намного удобнее. Они дают нам унификацию всех моделей видеокарт, всех брендовых производителей, аппаратную поддержку многих технологий, удобную (для программиста) эмуляцию программными средствами того, что не поддерживается аппаратным обеспечением.

Таким образом, у программиста появляется куда больше времени, которое он может потратить, допустим, на программирование геймплея игры, ну или AI (Artificial Intelligence, искусственный интеллект).

OpenGL

OpenGL (Open Graphics Library — открытая графическая библиотека) — спецификация, определяющая независимый от языка программирования кросс-платформенный программный интерфейс для написания приложений, использующих двумерную и трехмерную компьютерную графику.

OpenGL ориентируется на следующие две задачи:

- скрыть сложности адаптации различных 3D-ускорителей предоставляя разработчику единый API
- скрыть различия в возможностях аппаратных платформ, требуя реализации недостающей функциональности с помощью программной эмуляции

Основным принципом работы OpenGL является получение наборов векторных графических примитивов в виде точек, линий и многоугольников с последующей математической обработкой полученных данных и построением растровой картинки на экране и/или в памяти. Векторные трансформации и растеризация выполняются графическим конвейером (graphics pipeline), который по сути представляет из себя дискретный автомат. Абсолютное большинство команд OpenGL попадают в одну из двух групп: либо они добавляют графические примитивы на вход в конвейер, либо конфигурируют конвейер на различное исполнение транформаций.

OpenGL является низкоуровневым процедурным API, что вынуждает программиста диктовать точную последовательность шагов, чтобы построить результирующую растровую графику (императивный подход). Это является основным отличием от дескрипторных подходов, когда вся сцена передается в виде структуры данных (чаще всего дерева), которое обрабатывается и строится на экране. С одной стороны императивный подход требует от программиста глубокого знания законов трёхмерной графики и математических моделей, с другой стороны — даёт свободу внедрения различных инноваций.

Одно из основных преимуществ OpenGL состоит в том, что он является бесплатным для использования продуктом, с полным открытым исходным кодом. Кроме того, OpenGL можно использовать с тем языком, который является ближе всего к вам: C, C++, C#, Java, Delphi, Visual Basic и другие.

DirectX

Классическое определение представляет DirectX как совокупность интерфейсов прикладного программирования - Application Programming Interface, API, для удобного программирования приложений под операционные системы Microsoft Windows, главным образом, для программирования игр. Говоря проще, разнообразные библиотеки–API из комплекта DirectX, представляют собой готовые наборы функций для облегчения труда программистов. Теперь им не нужно каждый раз создавать ряд типичных процессов для работы со звуком, видео и т.п., для этого в комплекте DirectX имеется ряд стандартных "кубиков" и инструментов для создания мультимедийных приложений и игрушек под Windows.

Практически все части DirectX API представляют собой наборы COM-совместимых объектов.

В целом, DirectX подразделяется на:

- **DirectX Graphics**, набор интерфейсов.
- **DirectInput**: интерфейс, используемый для обработки данных, поступающих с клавиатуры, мыши, джойстика и пр. игровых контроллеров.
- **DirectPlay**: интерфейс сетевой коммуникации игр.
- **DirectSound**: интерфейс низкоуровневой работы со звуком (формата Wave)
- **DirectMusic**: интерфейс воспроизведения музыки в форматах Microsoft.

- **DirectShow**: интерфейс, используемый для ввода/вывода аудио и/или видео данных.
- **DirectSetup**: часть, ответственная за установку DirectX.
- **DirectX Media Objects**: реализует функциональную поддержку потоковых объектов (например, энкодеры/декодеры)

Ранее DirectX вкладывался разработчиками в дистрибутивы игр, но сейчас он включён в стандартный набор ПО Windows. На данный момент самой свежей версией является DirectX 10.1. Зачастую, свежие версии DirectX поставляются вместе с игровыми приложениями, так как DirectX API обновляется достаточно часто, и версия, включённая в ОС Windows зачастую является далеко не самой новой.

Среди ключевых нововведений, реализованных в DirectX10, любители новых игр по достоинству оценят следующие:

- Более реалистичная анимация шерсти меха и растений
- Более мягкие и более чёткие тени
- Более насыщенные ландшафты с более сложной окружающей обстановкой
- Значительно более тщательно прорисованный лес, более масштабные и детальные сцены баталий
- Более динамичные и чаще меняющиеся по ходу событий сценарии игр
- Больший реализм и уменьшение смазывания движущихся объектов
- Объёмные эффекты
- Уточнённый, более реалистичный дым и облака
- Более реалистичные отражения и преломления на отражающих поверхностях – воде, автомобилях, стекле и др.
- Снижение загрузки CPU, перераспределение обсчёта ряда процессов на GPU, снижение вероятности подтормаживания и зависания системы при сложном геймплее.

К примеру можно привести изображение из **Flight Simulator**

DirectX9 рендеринг

DirectX10 рендеринг

1.11 Технологии объединения видеоадаптеров.

О технологиях увеличения производительности графической подсистемы путем объединения двух видеокарт в одну упряжку говорится уже давно.

Как вы знаете, история вопроса насчитывает уже не один год, и попытки объединить мощность двух графических чипов делались довольно давно. Родоначальником движения за удвоение принято считать компанию 3dfx с ее технологией Scan Line Interface (SLI), впоследствии приобретенной NVIDIA. Для компании ATI объединять мощь двух видеопроцессоров в одну упряжку тоже не впервой. Еще в 1999 году увидела свет видеокарта Rage Fury Maxx, на которой было установлено два видеочипа, не говоря уже о профессиональных видеорешениях (назвать их видеокартами язык не поворачивается, поскольку это целый программно-аппаратный комплекс), предназначенных для использования в авиа-тренажерах для подготовки пилотов. Подобные системы одновременно задействуют чуть ли не десятки видеопроцессоров. Однако их главная задача - не максимальный FPS (количество кадров в секунду), а максимально реалистичная картинка, выдаваемая на экраны тренажера, при сохранении приемлемого FPS. Стоят такие комплексы не один десяток, а то и сотню тысяч условных единиц. Однако спустимся с небес на землю, и посмотрим, каким образом высокие технологии проникают на массовый рынок.

Как обычно получается, здоровая конкуренция приводит к тому, что производители рано или поздно приходят к некоторому общему знаменателю, касательно использования тех или иных технологических решений. Похожим образом ситуация развивается и в секторе видеосистем. За последнее время слова SLI и CrossFire касались уха, наверное, каждого человека кто хоть иногда интересуется компьютерными "железками". И, что интересно, развитие данных технологий в последнее время идет по схожему пути.

Рассмотрим сходства и различия между SLI и CrossFire. Обе технологии требуют соответствующей "обвязки". Один из основных элементов - материнская плата, допускающая установку двух видеокарт и обязательно поддерживающая соответствующую технологию. То есть требуется материнская плата с двумя слотами PCI-E, способных работать со скоростью 8x каждый. До недавнего времени, помимо "правильной" материнской платы, требовалось и некоторое дополнительное "приспособление", чтобы объединить две видеокарты в одно целое. В технологии SLI это был мостик, накидывающийся сверху на соответствующие разъемы видеокарт, по технологии CrossFire использовался внешний шнур, подключаемый к разъему DVI. Как видите, вот тут то и начинаются различия.

Технология объединения видеокарт от NVIDIA изначально предполагала использование двух идентичных видеокарт. В настоящее время данное требование несколько ослабло, в том смысле, что не требуется две совершенно идентичные видеокарты, достаточно установить две видеокарты на одинаковых GPU. Объединение вычислительной мощности видеокарт

осуществлялось драйверами. Именно поэтому компании NVIDIA потребовалось некоторое время для их отладки, поскольку первое время системы с SLI были несколько нестабильны.

Компания ATI пошла по другому пути. Изначально предполагалось, что видеокарты, объединяемые по технологии CrossFire, будут неравноправны по выполняемым функциям. Одна из видеокарт является Master-картой и содержит в себе важнейший элемент - Composite Engine. Это такая микросхема, которая выполняет объединение вычислительной мощности видеокарт на аппаратном уровне. Зачем такие сложности, спросите вы? Идея, положенная за основу выбранной технологии, очень проста - избежать влияния драйверов и других программных компонентов на производительность связки двух видеокарт. К тому же, как известно с давних времен, специализированные аппаратные решения намного лучше справляются со своими обязанностями по сравнению с программными средствами, выполняемыми на процессорах универсальной архитектуры (то есть - CPU). Таким образом, получается что, выбрав независимость от драйверов, компания ATI лишилась унификации видеокарт в связке CrossFire. Или наоборот. Отказавшись от унификации видеокарт в связке CrossFire, компания ATI приобрела независимость от производительности драйверов. Это уже вопрос выбранной точки зрения на данный вопрос. К сожалению, с первой реализацией CrossFire на базе Radeon X800/850 не все пошло гладко, как раз из-за аппаратных ограничений Composite Engine. Главная проблема была в максимальном поддерживаемом режиме - 1600x1200@60 Гц вызвал много нареканий за малую частоту вертикального обновления. Урок был извлечен, и следующая реализация CrossFire избавилась от "детских" болезней.

Сейчас мы рассмотрим с вами развитие технологий SLI и CrossFire после чего у вас возможно появится собственное мнение, на то какая же из технологий лучше.

NVIDIA SLI

SLI — технология, позволяющая использовать мощности нескольких видеокарт для обработки трехмерного изображения.

Для построения компьютера на основе SLI необходимо иметь:

- материнскую плату с двумя и более разъемами PCI Express, поддерживающую технологию SLI.
- мощный блок питания;
- видеокарты GeForce 6/7/8/9 или Quadro FX с шиной PCI Express;
- мост, объединяющий видеокарты.

Поддержка чипсетов для работы со SLI осуществляется программно. Видеокарты должны принадлежать к одному классу, при этом версия BIOS плат и их производитель значения не имеют.

SLI-систему можно организовать двумя способами:

- С помощью специального мостика;
- Программным путем.

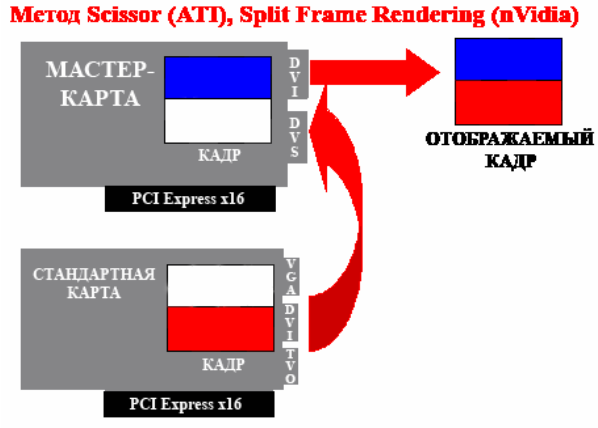
В последнем случае нагрузка на шину PCIe возрастает, что плохо сказывается на производительности.

В последнее время получила распространение система Quad SLI. Она предполагает объединение в SLI-систему двух двухчиповых плат (x2) Таким образом, получается, что в построении изображения работают 4 чипа. На рисунке изображена как раз такая система, два видео адаптера GeForce 9800GX2 объединены в режим SLI с помощью специального моста.

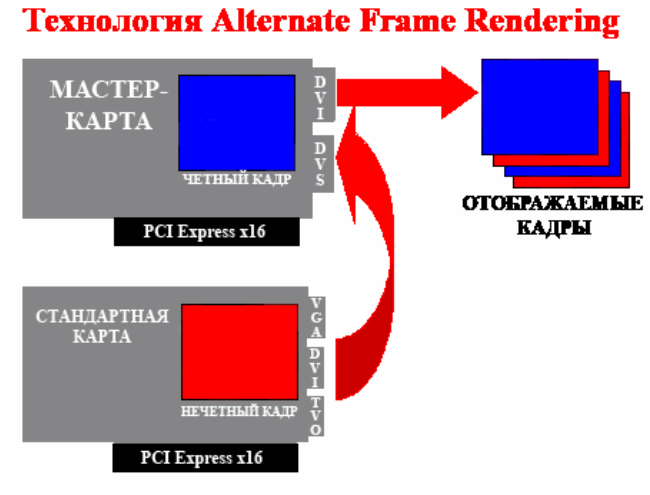
Рисунок – Видеокарты в режиме SLI

Алгоритмы построения изображений

- **Split Frame Rendering**
  Изображение разбивается на несколько частей, количество которых соответствует количеству видеокарт в связке. Каждая часть изображения обрабатывается одной видеокартой полностью, включая геометрическую и пиксельную составляющие.



- **Alternate Frame Rendering**
  Обработка кадров происходит поочередно: одна видеокарта обрабатывает только четные кадры, а вторая — только нечетные. Однако у этого алгорима есть недостаток. Дело в том, что один кадр может быть простым, а другой сложным для обработки.
  Данный алгоритм, был запатентован ATI во время выпуска двухчиповой видеокарты



- **SLI AA**
  Данный алгоритм нацелен на повышение качества изображения. Одна и та же картинка генерируется на всех видеокартах с разными шаблонами сглаживания. Видеокарта производит сглаживание кадра с некоторым шагом относительно изображения другой видеокарты. Затем полученные изображения смешиваются и выводятся. Таким образом достигается максимальные четкость и детализованность изображения. Доступны следующие режимы сглаживания: 8x, 10x, 12x, 14x, 16x и 32x. Метод SLI AA (NVIDIA) мало отличается от аналога от ATI. Однако становятся доступны два новых режима сглаживания: SLI AA 8x и SLI AA 16x. Первый — это комбинация 4-кратного мультисэмплинга каждой из видеокарт (MSAA 4x + MSAA 4x), а второй — 4-кратного мульти- и 2-кратного суперсэмплинга, то есть 8xS + 8xS. В результате применения новых режимов возрастает четкость изображения и детальность сцены. Особенно это касается мелких и удаленных от зрителя объектов. Есть и более экзотический режим SLI AA 32x, правда, воспользоваться им будет сложно — слишком высока нагрузка.

ATI CrossFire

ATI CrossFire — технология, позволяющая одновременно использовать мощности двух и более видеокарт Radeon для построения трёхмерного изображения.

Каждая из видеокарт, используя определённый алгоритм, формирует свою часть изображения, которое передаётся в чип Composing Engine мастер-карты, имеющий собственную буферную память. Этот чип объединяет изображения каждой видеокарты и выводит финальный кадр.

В будущем видеокарты на основе CrossFire облегчат работу процессора с графикой. Две видеокарты будут обрабатывать графику, а одна — физику.

Для построения компьютера на основе CrossFire необходимо иметь:

- материнскую плату с двумя и более разъёмами PCI Express x16 с чипсетом AMD или Intel определённой модели;
- мощный блок питания;
- видеокарты с поддержкой CrossFire.

Видеокарты должны быть одной серии, но необязательно одной модели. При этом быстродействие и частота CrossFire-системы определяется характеристиками чипа наименее производительной видеокарты.

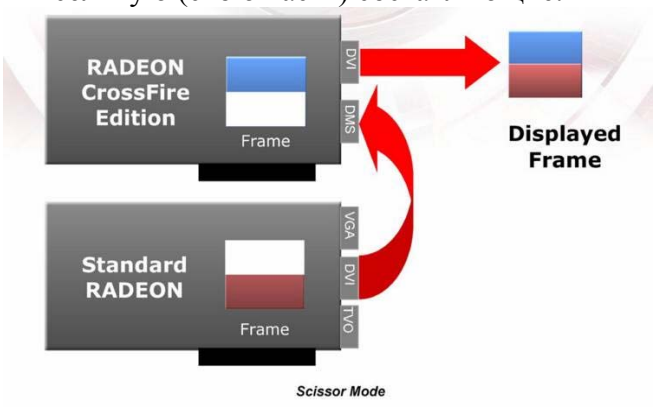
CrossFire-систему можно организовать тремя способами:

- Внешнее соединение — видеокарты объединяются с помощью кабеля, при этом карта, на которой распаян чип Compositing Engine, называется мастер-картой (Master card). Остальные видеокарты могут быть любыми в пределах серии.
- Внутреннее соединение — видеокарты соединены посредством гибкого мостика. Драйвером определяется, какая из них будет мастер-картой.

- Программный метод — видеокарты не соединяются, обмен данными идёт по шине PCI Express x16, при этом их взаимодейтсвие реализуется с помощью драйверов. Недостатком данного способа являются потери в производительности на 10-15% по сравнению с двумя вышеназванными способами.

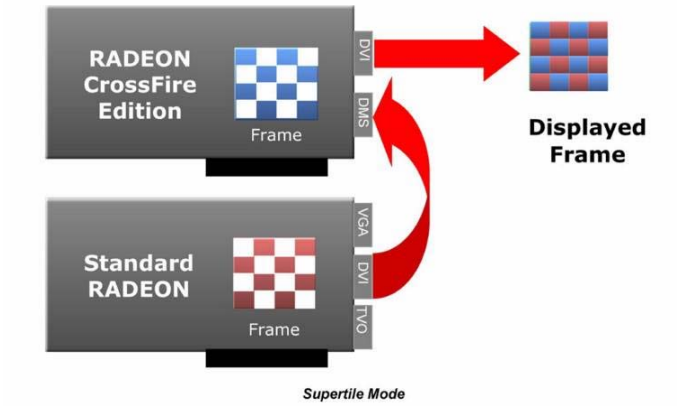
Алгоритмы построения изображений:

- **Scissor**
  В случае с Scissor как это было и с Split Frame Rendering экран разделяется на несколько частей, каждая из которых обрабатывается отдельной видеокартой. При использовании двух плат изображение делится на две части по горизонтали. Нагрузка между видеокартами распределяется динамически, то есть разделение экрана происходит пропорционально загруженности сцены. Обе платы обрабатывают как геометрическую (полностью), так и пиксельную (свою часть) составляющие.



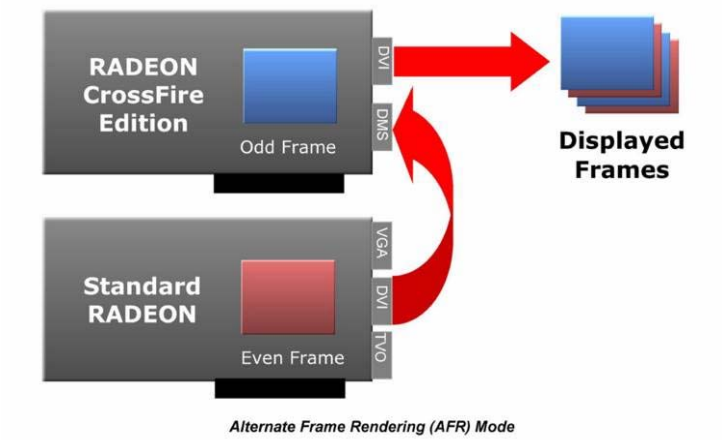
Scissor Mode

- **SuperTiling**
  Метод SuperTiling поддерживается только ATI. Картинка разбивается на отдельные участки и принимает вид шахматной доски. Каждая видеокарта обрабатывает свою часть изображения — квадратик 32 на 32 пикселя. Таким образом, нагрузка по закраске делится примерно поровну, а вот геометрическая дублируется — обе видеокарты рассчитывают одни и те же данные. Этот режим особенно пригодится в играх, где не делается упор на геометрическую составляющую.



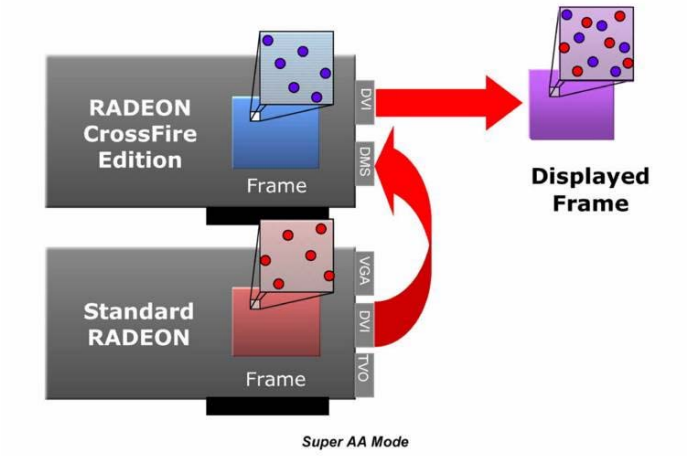
Supertile Mode

- **Alternate Frame Rendering**
  Технология Alternate Frame Rendering была запатентована ATI во времена двухчиповой карты Rage Fury MAXX. Сейчас AFR используют обе компании. В основе метода лежит поочередная обработка кадров: одной плате достаются четные кадры, другой — нечетные. В идеале такой подход должен обеспечивать двукратный прирост скорости. На деле все иначе. В силу того, что технология основана на принципах параллельной работы, вся геометрия, шейдерные программы и прочее обрабатываются на обеих платах — это плюс. Но из-за особенностей SLI/CrossFire возникают заметные задержки в реакции системы на действия пользователя (рывки). Почему? Да все просто — один кадр может быть довольно простым для обработки, а следующий во много раз сложнее. Тут-то и начинаются проблемы, времени на построение кадра требуется больше.



Alternate Frame Rendering (AFR) Mode

- **SuperAA**
  Вышеобозначенные режимы, в отличие от предыдущих, нацелены на повышение качества картинки, а не быстродействия. При использовании SuperAA (ATI) обе карты генерируют одно и то же изображение, но используют разные шаблоны полноэкранного сглаживания (FSAA). Видеокарта выполняет над кадром сглаживание с некоторым сдвигом относительно результатов другой платы. Затем происходит смешивание обеих картинок и на экран выводится результат. Таким образом, происходит удвоение качества сглаживания без потерь скорости (в сравнении с одной картой). Доступные режимы SuperAA: 8x, 10x (8x + 2xS), 12x и 14x (12x + 2xS).



Super AA Mode

Технологии SLI и CrossFire можно оценивать по-разному. Для кого-то это — шанс установить на своем домашнем ПК новые рекорды, исчисляемые в очках и кадрах в секунду. Есть и те, кто скептически относится к приросту производительности, который явно не соразмерен денежным затратам. Третьи скептически смотрят на «двухакселераторные» тандемы из-за сырости драйверов, над которыми еще предстоит работать и работать. По-своему правы и те, и другие, и десятые.

На наш взгляд, очень хорошо, что производители железа стремятся предоставить больше вариантов для создания мощного игрового ПК. При этом выбор остается за вами, а уж вам нужно тщательно все взвесить: что вы хотите от видеоподсистемы и готовы ли за это платить.

2. Мониторы

2.1 Введение

Информационную связь между пользователем и компьютером обеспечивает монитор. Первые микрокомпьютеры представляли собой небольшие блоки, в которых практически не было средств индикации. Все, что имел в своем распоряжении пользователь, — это набор мигающих светодиодов или возможность распечатки результатов на принтере. По сравнению с

современными стандартами первые компьютерные мониторы были крайне примитивны; текст отображался только в одном цвете (как правило, в зеленом), однако в те годы это было важнейшим технологическим прорывом, поскольку пользователи получили возможность вводить и выводить данные в режиме реального времени. Затем появились цветные мониторы, увеличился размер экрана, и жидкокристаллические панели перекочевали с портативных компьютеров на рабочий стол пользователей.

В наши дни компьютерные мониторы достигли высшей ступени развития, что не избавляет пользователя от необходимости разбираться в аппаратном обеспечении. Медленный ви деоадаптер может затормозить работу даже самого быстрого компьютера. А неправильное со четание монитора и видеоадаптера не только не позволит полноценно выполнять поставленные задачи, но может привести к ухудшению зрения.

Система отображения состоит из двух главных компонентов:

- . монитора (дисплея);
- . видеоадаптера (называемого также видеоплатой или графической платой).

Видеоадаптеры мы уже с вами обсудили, теперь давайте поговорим более подробно про мониторы

Так как же выбрать монитор? Так, чтобы было удобно и безопасно работать, чтобы голова не болела, а глаза не уставали, чтобы было комфортно играть и работать? На все эти вопросы мы и попытаемся дать ответ.

Понятно, что критериев, определяющих правильный выбор монитора, очень много. Более того, для разных целей выбираются разные мониторы. Стоимость мониторов может очень существенно отличаться, их возможности и технические параметры тоже различны. Мы рассмотрим разные типы мониторов, поговорим о рекомендациях при покупке, о том, как выбрать монитор именно для ваших нужд.

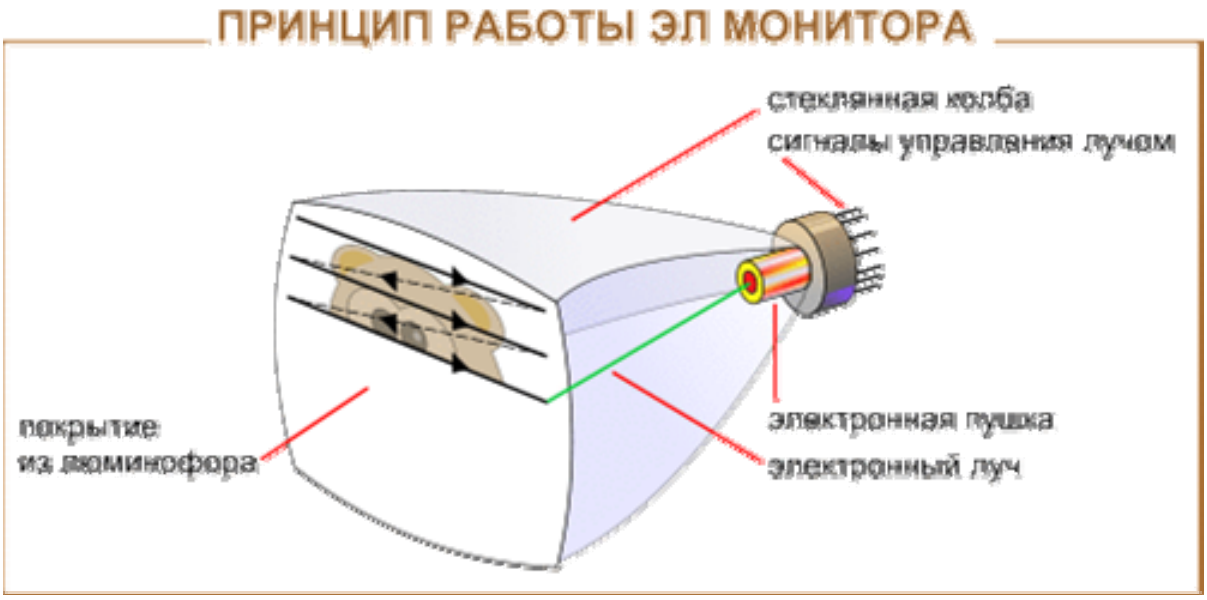
Разумеется, при выборе монитора мы, волей-неволей, ориентируемся на рекламу. Но, по понятным причинам, в рекламе производители делают акцент на тех характеристиках монитора, которые выгодны именно производителям. Мы рассмотрим, на что следует обратить особое внимание при покупке, и о каких характеристиках следует знать точно. Также мы рассмотрим преимущества и недостатки разных типов мониторов, начиная с традиционных мониторов с Электронно-Лучевой Трубкой (ЭЛТ или Cathode Ray Tube (CRT)) и современных жидкокристаллических (ЖК) мониторов (Liquid Crystal Display (LCD)), и заканчивая плазменными дисплеи, OLEP и Field Emission Display (FED). Мы уделим особое внимание таким параметрам, как поддерживаемые разрешения и частоты обновления, соответствие стандартам безопасности и поддержка режимов энергосбережения.

2.2 Основные принципы работы CRT-мониторов

До недавнего времени самым распространенный типом мониторов был CRT-мониторы. Как видно из названия, в основе всех подобных мониторов лежит катодно-лучевая трубка, но это дословный перевод, технически правильно говорить "электроннолучевая трубка" (ЭЛТ).

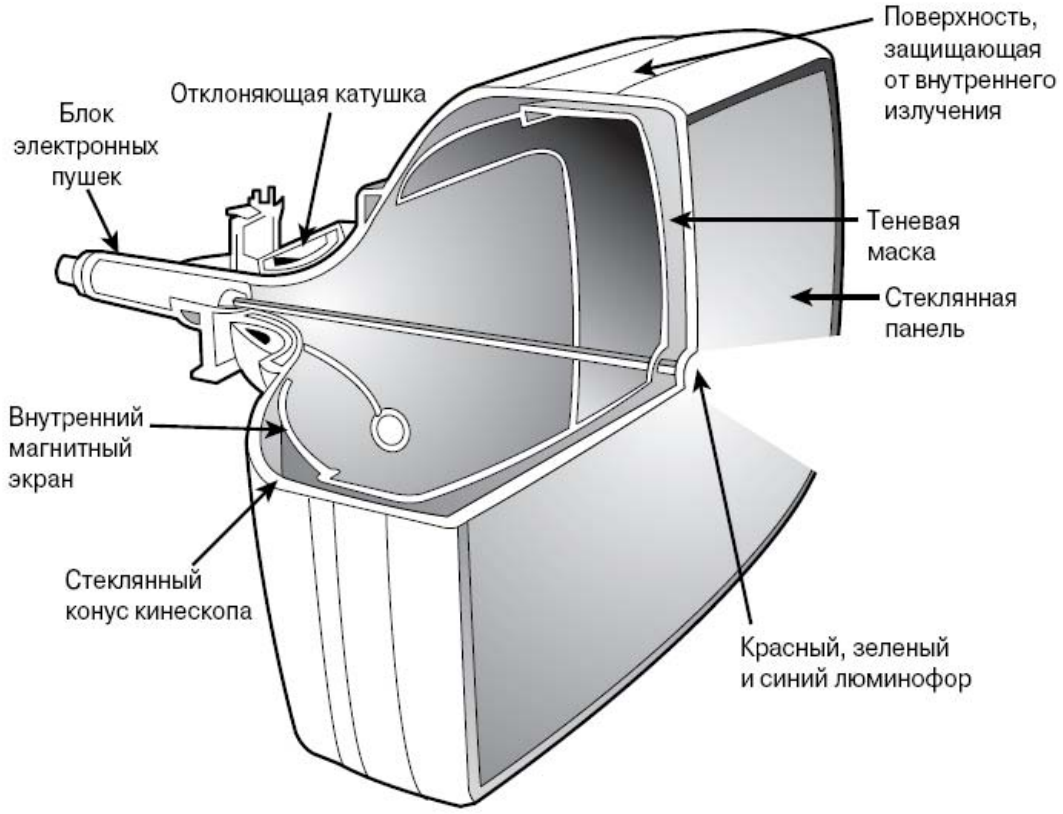
Используемая в этом типе мониторов технология была создана много лет назад и первоначально создавалась для применения в осциллографах. Развитие этой технологии, применительно к созданию мониторов, за последние годы привело к производству все больших по размеру экранов с высоким качеством и низкой стоимостью. Сегодня найти в магазине 14" или 15" монитор почти невозможно (разве только это не магазин б/у техники), а лет 5 назад это был стандарт. Сегодня стандартными являются 17" мониторы, и наблюдается явная тенденция в сторону 19" экранов. Скоро 19" мониторы станут стандартным устройством, особенно в свете существенного снижения цен на них, а на горизонте уже 21" мониторы и более.

Информация на мониторе может отображаться несколькими способами. Самый распростра ненный — отображение на экране электроннолучевой трубки (ЭЛТ), такой же, как в телевизоре. ЭЛТ представляет собой электронный вакуумный прибор в стеклянной колбе, в горловине ко торого находится электронная пушка, а на дне — экран, покрытый люминофором.



Нагреваясь, электронная пушка испускает поток электронов, которые с большой скоростью устремляются к экрану. Поток электронов (электронный луч) проходит через фокусирующую и отклоняющую катушки, которые направляют его в определенную точку покрытого люминофором экрана. Под воздействием ударов электронов люминофор излучает свет, который видит пользователь, сидящий перед экраном компьютера. В электроннолучевых мониторах используются три слоя люминофора: красный, зеленый и синий. Для выравнивания потоков электронов применяется так называемая теневая маска — металлическая пластина, имеющая щели или отверстия, которые разделяют красный, зеленый и синий люминофор на группы по три точки каждого цвета. Качество изображения определяется типом используемой теневой маски; на резкость изображения влияет расстояние между группами люминофоров (шаг расположения точек).

На рисунке показан разрез типичного электроннолучевого монитора. Химическое вещество, используемое в качестве люминофора, характеризуется временем послесвечения, которое отображает длительность свечения люминофора после воздействия электронного пучка. Время послесвечения и частота обновления изображения должны соответствовать друг другу, чтобы не было заметно мерцание изображения (если время послесвечения очень мало) и отсутствовала размытость и удвоение контуров в результате наложения последовательных кадров (если время послесвечения слишком велико).



Электронный луч движется очень быстро, прочерчивая экран строками слева направо и сверху вниз по траектории, именуемой растром. Период сканирования по горизонтали определяется скоростью перемещения луча поперек экрана.

В процессе развертки (перемещения по экрану) луч воздействует на те элементарные участки люминофорного покрытия экрана, где должно появиться изображение. Интенсивность луча постоянно меняется, в результате чего изменяется яркость свечения соответствующих участков экрана. Поскольку свечение исчезает очень быстро, электронный луч должен вновь и вновь пробегать по экрану, возобновляя его. Этот процесс называется возобновлением (или регенерацией) изображения.

В большинстве мониторов частота регенерации, которую также называют частотой вертикальной развертки, во многих режимах приблизительно равна 85 Гц, т.е. изображение на экране обновляется 85 раз в секунду. Снижение частоты регенерации приводит к мерцанию изображения, которое очень утомляет глаза. Следовательно, чем выше частота регенерации, тем комфорт нее себя чувствует пользователь. В некоторых дешевых мониторах частота регенерации без мерцания возможна только при разрешениях 600x480 и 800x600; следует приобретать монитор, поддерживающий достаточную частоту регенерации при разрешении 1 024Ч768 и выше.

Очень важно, чтобы частота регенерации, которую может обеспечить монитор, соответствовала частоте, на которую настроен видеоадаптер. Если такого соответствия нет, изображение на экране вообще не появится, а монитор может выйти из строя. В целом видеоадаптеры обеспечивают намного большую частоту регенерации, чем поддерживается большинством мониторов. Именно поэтому изначальная частота регенерации, определенная для большинства видеоадаптеров с целью избежать повреждения монитора, составляет 60 Гц. Частоту можно изменить с помощью диалогового окна **Свойства: Экран (Display Properties)**.

В одних мониторах установлена фиксированная частота развертки, в других поддерживаются разные частоты в некотором диапазоне (такие мониторы называются многочастотными — multiple frequency monitor). Большинство современных мониторов многочастотные, т.е. могут работать с разными стандартами видеосигнала, которые получили довольно широкое распространение. Для обозначения мониторов такого типа производители

используют различные термины: синхронизируемые (multisync), многочастотные (multifrequency), многорежимные (multiscan), автосинхронизирующиеся (autosynchronous) и с автонастройкой (autotracking).

Экраны мониторов могут быть двух типов: выпуклые и плоские. До недавнего времени большинство экранов были выпуклыми, т.е. экран изгибался к краям корпуса. Этот принцип применялся в производстве львиной доли ЭЛТ мониторов и телевизоров. Несмотря на низкую стоимость подобного экрана, выпуклая поверхность приводила к искажению изображения и появлению бликов, особенно если монитор располагался в ярко освещенной комнате. Чтобы уменьшить уровень отблеска света типичного выпуклого экрана, в некоторых мониторах используется специальное антибликовое покрытие.

Обычно экран искривлен как по вертикали, так и по горизонтали. В некоторых моделях (Sony FD Trinitron и Mitsubishi DiamondTron NF) используется конструкция Trinitron, в которой поверхность экрана имеет небольшую кривизну только в горизонтальном сечении. Кривизна вертикального сечения экрана равна нулю; подобная трубка называется плоской (flat square tube — FST).

В настоящее время большинство мониторов оснащены экранами, плоскими в горизонтальном и вертикальном сечении. Подобные экраны уже появлялись на рынке в конце 1980 х годов (Zenitn FTM), однако не получили широкого распространения. Среди основных типов плоских ЭЛТ стоит отметить технологии FD Trinitron компании Sony, DiamondTron NF от NEC/Mitsubishi, Perfect Flat от ViewSonic и Flatron от LG. Плоский экран отбрасывает гораздо меньше бликов и обеспечивает высококачественное насыщенное изображение с минимальными искажениями.

Все мы знаем или слышали о том, что наши глаза реагируют на основные цвета: красный (Red), зеленый (Green) и синий (Blue) и на их комбинации, которые создают бесконечное число цветов.

Люминофорный слой, покрывающий переднюю часть электронно-лучевой трубки, состоит из очень маленьких элементов (настолько маленьких, что человеческий глаз их не всегда может различить). Эти люминофорные элементы воспроизводят основные цвета, фактически имеются три типа разноцветных частиц, чьи цвета соответствуют основным цветам RGB - отсюда и название группы из люминофорных элементов - триады.

Люминофор начинает светиться, как было сказано выше, под воздействием ускоренных электронов, которые создаются тремя электронными пушками. Каждая из трех пушек соответствует одному из основных цветов и посылает пучок электронов на различные частицы люминофор, чье свечение основными цветами с различной интенсивностью комбинируется, и, в результате, формируется изображение с требуемым цветом. Например, если активировать красную, зеленую и синюю люминофорные частицы, то их комбинация сформирует белый цвет.

Для управления электронно-лучевой трубкой необходима и управляющая электроника, качество которой во многом определяет и качество монитора. Кстати, именно разница в качестве управляющей электроники, создаваемой разными производителями, является одним из критериев, определяющих разницу между мониторами с одинаковой электронно-лучевой трубкой.

Понятно, что электронный луч, предназначенный для красных люминофорных элементов, не должен влиять на люминофор зеленого или синего цвета. Чтобы добиться такого действия используется специальная маска.

Хотя тип маски является не критичным при выборе монитора, тем не менее представляется интересным рассмотреть различные типы масок, так как маски определенного типа лучше подходят для тех или иных работ, выполняемых за компьютером.

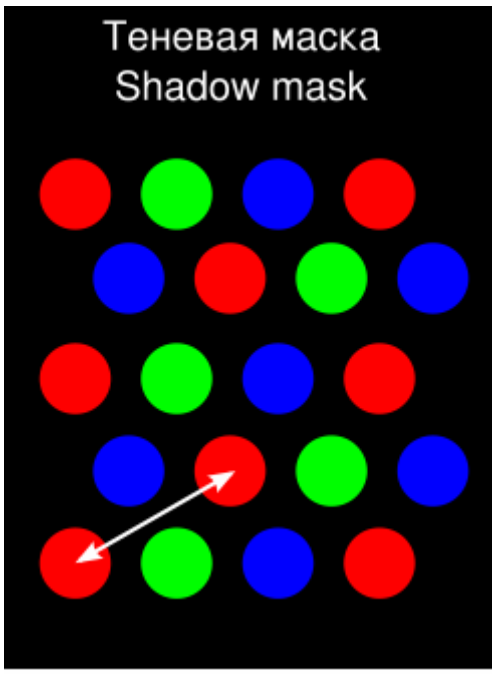
Тип маски можно определить в терминах формы и расположения **зерен (dots)** люминофора на экране. Следует отметить, что именно зерно является минимальным "атомом" изображения на экране, а пиксел может состоять из нескольких зерен (в зависимости от разрешения)

А теперь давайте обсудим в чем жеж различия между разными типами масок

2.2.1 Различные типы масок

Shadow Mask (Теневая маска, Дельтовидная технология)

В данной технологии цветной элемент состоит из трех зерен, расположенных в вершинах правильного треугольника. Резкость изображения определяется расстоянием между геометрическими центрами соседних элементов.

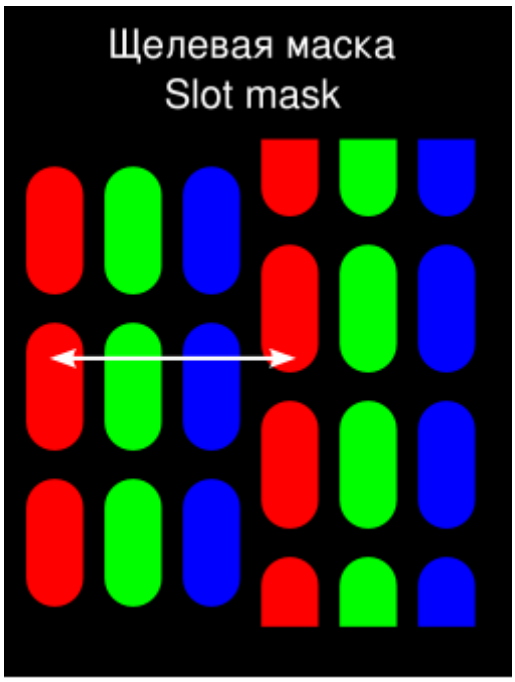


Основное достоинство дельтовидной технологии: точная четкая картинка, диагональные линии не имеют зазубрин. Основным же недостатком считается большое расстояние между зернами, отчего такие трубки имеют не слишком насыщенный цвет.

Теневая маска (shadow mask) - это самый распространенный тип масок для CRT-мониторов. Например, в плоских трубках Infinite Flat Tube (серия DynaFlat) от Samsung используется теневая маска. Теневая маска состоит из металлической сетки перед частью стеклянной трубки с люминофорным слоем. Как правило, большинство современных теневых масок изготавливают из инвара (invar, сплав железа и никеля). Отверстия в металлической сетке работают, как отверстия в сите, именно этим обеспечивается то, что электронный луч попадает только на требуемые люминофорные элементы, и только в определенных областях.

Минимальное расстояние между люминофорными элементами одинакового цвета называется **dot pitch (шаг точки)** и является важным параметром качества изображения. Шаг точки обычно измеряется в миллиметрах (мм). Чем меньше значение шага точки, тем выше качество воспроизводимого на мониторе изображения.

Slot Mask (Щелевая маска)



Щелевая маска (slot mask) - это технология, широко применяемая компанией NEC, под именем "CromaClear". В данном случае люминофорные элементы расположены в вертикальных эллиптических ячейках, а маска сделана из вертикальных линий. Фактически, вертикальные полосы разделены на эллиптические ячейки, которые содержат группы из трех люминофорных элементов трех основных цветов. Минимальное расстояние между двумя ячейками называется slot pitch (щелевой шаг). Чем меньше значение щелевого шага, тем выше качество изображения на мониторе.

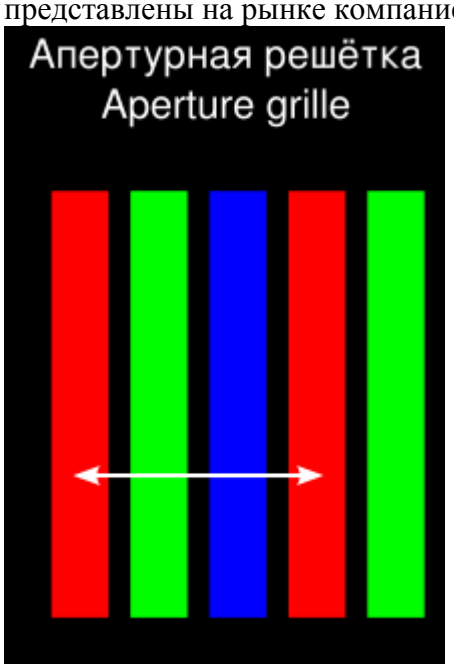
Преимущества щелевой маски: Более насыщенный цвет, нежели при использовании дельтовидной технологии, за счет большей площади светящегося люминофора, однако такая маска несколько проигрывает

в четкости наклонных линий.

Щелевая маска используется, помимо мониторов от NEC (где ячейки эллиптические), в мониторах Panasonic с трубкой PureFlat (ранее называвшейся PanaFlat). LG использует плоскую щелевую трубку Flatron с шагом 0,24 в своих мониторах.

Aperture Grill (Апертурная решетка)

Есть и еще один вид трубок, в которых используется "Aperture Grill" (апертурная, или теневая решетка). Эти трубки стали известны под именем Trinitron и впервые были представлены на рынке компанией Sony еще в 1982 году.



Апертурная решетка (aperture grill) - это тип маски, используемый разными производителями в своих технологиях для производства кинескопов, носящих разные названия, но имеющих одинаковую суть, например, технология Trinitron от Sony или Diamondtron от Mitsubishi. Это решение не включает в себя металлическую решетку с отверстиями, как в случае с теневой маской, а имеет решетку из вертикальных линий. Вместо точек с люминофорными элементами трех основных цветов апертурная решетка содержит серию нитей, состоящих из люминофорных элементов, выстроенных в виде вертикальных полос трех основных цветов. Такая система обеспечивает высокую контрастность изображения и хорошую насыщенность цветов, что вместе обеспечивает высокое качество мониторов с трубками на основе этой технологии. Маска, применяемая в трубках фирм Sony, Mitsubishi, ViewSonic, представляет собой тонкую фольгу, на которой процарапаны тонкие вертикальные линии. Она держится на горизонтальных проволочках, тень от проволочки пользователь ВИДИТ на экране в виде тонкие полосы (или полос), пересекающих по диагонали экран. Эта проволочка применяется для гашения колебаний и называется **damper wire**. Ее хорошо видно, особенно при светлом фоне изображения на мониторе. Некоторым пользователям эти линии принципиально не нравятся, другие же, наоборот, довольны и используют их в качестве горизонтальной линейки ;-).

Как и любой другой технологии, у апертурной решетки есть свои достоинства и недостатки. Основное достоинство это технологии: великолепные по насыщенности цвета, кроме того, экран таких мониторов гораздо менее выпуклый, чем при применении теневых масок и щелевых масок, так как в принципе не имеет причин быть искривлен по вертикали. Однако, современные технологии изготовления плоских трубок, позволяют и при применении теневой или щелевой маски добиться плоского экрана. Основной же недостаток ступенчатость и нечеткость диагональных линий, отчего проектировщики не любят мониторы с применением апертурной решетки.

Кроме nuj хотелось бы заметить что нельзя напрямую сравнивать размер шага для трубок разных типов: шаг точек (или триад) трубки с теневой маской измеряется по диагонали, в то время как шаг апертурной решетки, иначе называемый strip pitch (горизонтальный шаг) точек, - по горизонтали. Поэтому при одинаковом шаге точек трубка с теневой маской имеет большую плотность точек, чем трубка с апертурной решеткой. Для примера: 0,25 мм strip pitch (для апертурной решетки) приблизительно эквивалентен 0,27 мм dot pitch.

Как вы уже поняли, все типы масок имеют свои недостатки: у теневой маски - слабая насыщенность цвета, у апертурной решетки - плохие наклонные линии, у щелевой маски

совокупность и достоинств и недостатков обоих технологий. В целом, применение того или иного типа маски в конкретном экземпляре монитора - не самый главный, но тем не менее засуживающий внимания параметр.

2.2 Жидкокристаллические дисплеи (Liquid Crystal Display)

Несмотря на постоянное падение цен, жидкокристаллические дисплеи продолжают оставаться более дорогими, чем аналогичные ЭЛТ мониторы (хотя последние в последние вообще трудно найти в продаже). Тем не менее следует учитывать тот факт, что видимая область экрана жидкокристаллического дисплея выше, чем у классического монитора

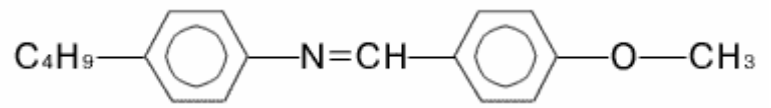
В портативных компьютерах в настоящее время используются цветные аналоговые или цифровые активные матрицы. Монохромные жидкокристаллические дисплеи уже давно не применяются в ПК, хотя остаются популярными для карманных компьютеров серии Palm и иногда применяются в промышленных цифровых устройствах. Дисплеи с пассивной матрицей и двойным сканированием были популярны в дешевых портативных компьютерах (ноутбуках), однако большинство продаваемых сегодня недорогих ноутбуков оснащены цветными аналоговыми или цифровыми матрицами, которые ранее использовались только в дорогостоящих моделях.

Дисплеи с пассивной матрицей до сих пор применяются в цифровых карманных компьютерах или промышленных устройствах, так как обладают привлекательной ценой и более высокой надежностью по сравнению с компьютерами, оснащенными экранами с активной матрицей.

Замечание *В большинстве дисплеев с пассивной матрицей используется технология транзисторов с полной переориентацией (**supertwist numatic design** или **STN**). Панели с активной матрицей, в свою очередь, основаны на тонкопленочных транзисторах (**thin film transistor - TFT**).*

2.2.1 Историческая справка

Жидкие кристаллы - это вещество, которое обладает свойствами как жидкости, так и твердого тела. Одно из самых важных свойств жидких кристаллов, которое используется в ЖК дисплеях - возможность изменять свою ориентацию в пространстве в зависимости от прикладываемого напряжения. Как обычно и происходит в науке, жидкие кристаллы были открыты случайно. В 1888 году Фридрих Рейнзер (Friedrich Reinitzer), австрийский ботаник, изучал роль холестерина в растениях. Один из экспериментов заключался в нагреве материала. Ученый обнаружил, что кристаллы становятся мутными и текут при 145,5º, а далее кристаллы превращаютсяв жидкость при 178,5º. Фридрих поделился открытием с Отто Леманном (Otto Lehmann), немецким физиком, который обнаружил у жидкости свойства кристалла в отношении реакции на свет. С тех пор и пошло название "жидкие кристаллы". Метоксибензидин бутиланалин (methoxybenzilidene butylanaline) Молекула, обладающая свойствами кристалла



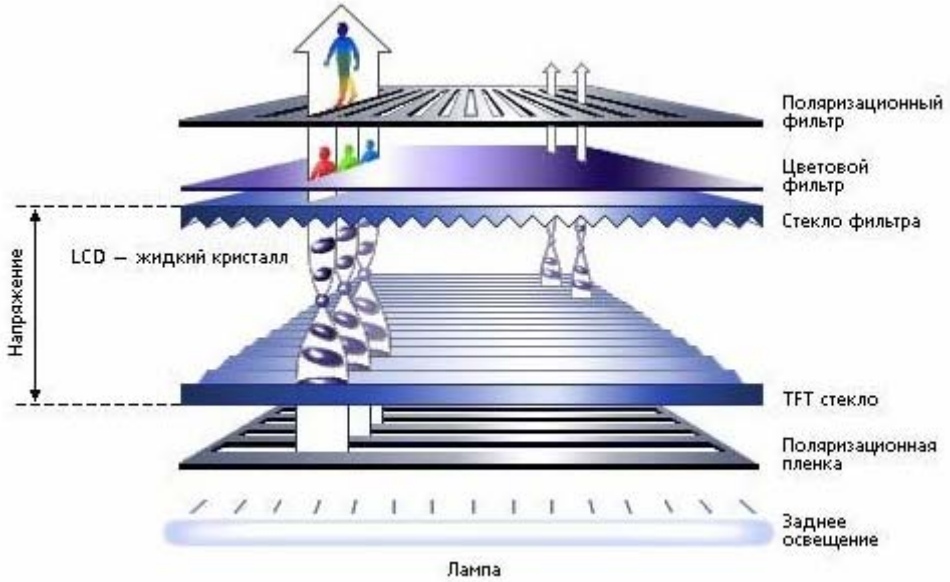
Почти столетие это открытие относилось к рангу удивительных особенностей природы, пока в 70-х годах XX века компания Radio Corporation of America не представила первый работающий монохромный экран на жидких кристаллах. Вскоре после этого технология начала проникать на рынок потребительской электроники, в частности, наручных часов и калькуляторов. Однако до появления цветных экранов было ещё очень далеко.

2.2.2 Принцип работы

Из всего ряда плоских дисплеев LCD выделяются тем, что сама жидкокристаллическая панель не является источником света; она лишь пропускает через себя свет, излучаемый неоновой лампой. Подтип таких дисплеев, TFT LCD, принято также называть жидкокристаллическими дисплеями с активной матрицей. Аббревиатура TFT (тонкоплёночный транзистор) обозначает управляющий элемент матрицы, контролирующий работу каждого отдельного пикселя.

Чтобы понять, как LCD контролирует яркость, нужно вспомнить эффект поляризации света из курса общей физики. Если не вдаваться в подробности, то данный эффект можно описать так: свет поляризуется, проходя через первый специальный фильтр, характеризуемый определённым углом поляризации. Для человеческого глаза ничего не меняется, только в два раза падает яркость света. Но если за первым фильтром поставить ещё один такой же, то свет будет либо полностью им поглощаться (если угол поляризации второго фильтра перпендикулярен углу первого), либо беспрепятственно проходить (если углы совпадают).

Плавное изменение угла второго фильтра позволяет плавно регулировать интенсивность света. Общий принцип действия всех TFT LCD показан на рисунке ниже: свет от неоновой лампы проходит через систему отражателей, направляется через первый поляризационный фильтр и попадает в слой жидких кристаллов, контролируемый транзистором; затем свет проходит через цветовые фильтры (как и в CRT, каждый пиксель матрицы строится из трёх компонент цвета – красной, зелёной и синей). Транзистор создаёт электрическое поле, задающее пространственную ориентацию жидких кристаллов. Свет, проходя через такую упорядоченную молекулярную структуру, меняет свою поляризацию, и в зависимости от неё будет либо полностью поглощён вторым поляризационным фильтром на выходе (образуя чёрный пиксель), либо не будет поглощаться или поглотится частично (образуя различные цветовые оттенки, вплоть до чистого белого).



В упрощенном виде матрица жидкокристаллического дисплея состоит из следующих частей:

- галогенная лампа подсветки;
- система отражателей и полимерных световодов, обеспечивающая равномерную подсветку;
- фильтр-поляризатор;
- стеклянная пластина-подложка, на которую нанесены контакты;
- жидкие кристаллы;
- ещё один поляризатор;
- снова стеклянная подложка с контактами.

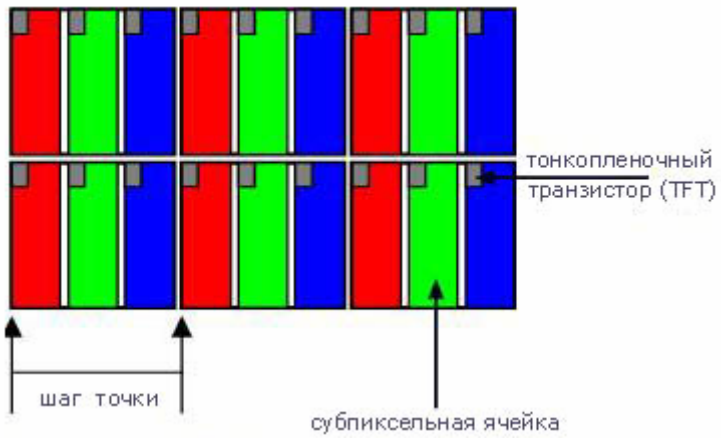
Поляризация, лежащая в основе LCD технологии, имеет и свои минусы. Один из главных – сокращение угла обзора жидкокристаллического дисплея, и производители LCD панелей это учитывают. В настоящее время распространены три технологии, позволяющие если не искоренить, то хотя бы значительно уменьшить такой недостаток.

Глобально матрицы делятся на пассивные (простые) и активные. В пассивных матрицах управление производится попиксельно, т.е. по порядку от ячейки к ячейке в строке. Проблемой, встающей при производстве ЖК-экранов по этой технологии, стало то, что при увеличении диагонали увеличиваются и длины проводников, по которым передаётся ток на каждый пиксель. Во-первых, пока будет изменён последний пиксель, первый успеет потерять заряд и погаснуть. Во-вторых, большая длина требует большего напряжения, что приводит к росту помех и наводок. Это резко ухудшает качество картинки и точность цветопередачи. Из-за этого пассивные матрицы применяются только там, где не нужны большая диагональ и высокая плотность отображения.

Для преодоления этой проблемы были разработаны активные матрицы. Основой стало изобретение технологии, известной всем по аббревиатуре TFT, что означает Thin Film Transistor – тонкоплёночный транзистор. Благодаря TFT, появилась возможность управлять каждым пикселем на экране отдельно. Это резко сокращает время реакции матрицы и делает возможными большие диагонали матриц. Транзисторы изолированы друг от друга и подведены к каждой ячейке матрицы. Они создают поле, когда им приказывает управляющая логика – драйвер матрицы. Для того, чтобы ячейка не потеряла заряд преждевременно, к ней добавляется небольшой конденсатор, который играет роль буферной ёмкости. С помощью этой технологии удалось радикально уменьшить время реакции отдельных ячеек матрицы.

2.2.3 Строение пиксела TFT

Цветные фильтры для красного, зелёного и синего цветов интегрированы в стеклянную основу и расположены близко друг к другу. Каждый пиксел (точка) состоит из трёх ячеек указанных цветов (субпикселей). Это означает, что при разрешении 1280 x 1024 точки экран содержит ровно 3840 x 1024 транзистора и пиксельных элемента. Шаг пиксела для 15.1" TFT-дисплея (1024 x 768 точек) составляет примерно 0.0188" (или 0.30 мм), а для 18.1" TFT (1280 x 1024 точки) примерно 0.011" (или 0.28 мм).

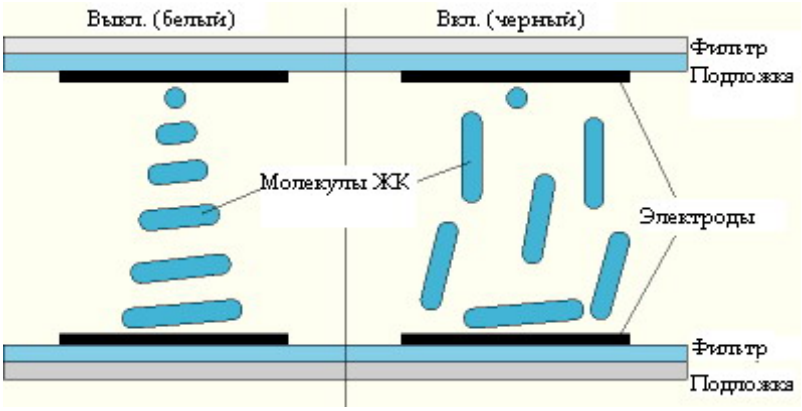


2.2.4 Типы TFT матриц

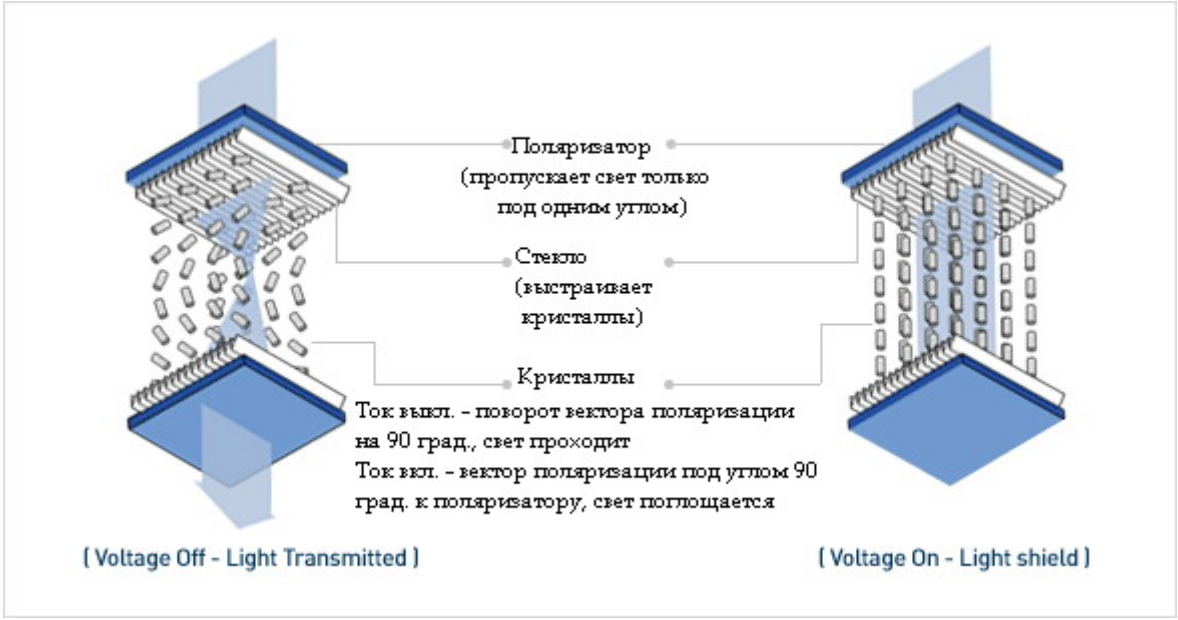
TN TFT или **TN+Film TFT (Twisted Nematic + Film)**

Самый распространённый тип цифровых панелей основан на технологии, сокращённо называемой TN TFT или TN+Film TFT (Twisted Nematic + Film). Термин Film обозначает дополнительное наружное плёночное покрытие, позволяющее увеличить угол обзора со стандартных 90 градусов (по 45 с каждой стороны) до приблизительно 140 градусов. Схема работы TN TFT дисплея показана на рисунке ниже:

- Когда транзистор находится в выключенном состоянии, то есть не создаёт электрическое поле, молекулы жидких кристаллов находятся в своём нормальном состоянии и выстроены так, чтобы менять угол поляризации проходящего через них светового потока на 90 градусов (жидкие кристаллы образуют спираль). Поскольку угол поляризации второго фильтра перпендикулярен углу первого, то проходящий через неактивный транзистор свет будет без потерь выходить наружу, образуя яркую точку, цвет которой задаётся световым фильтром.
- Когда транзистор генерирует электрическое поле, все молекулы жидких кристаллов выстраиваются в линии, параллельные углу поляризации первого фильтра, и тем самым никоим образом не влияют на проходящий через них световой поток. Второй поляризующий фильтр поглощает свет полностью, создавая чёрную точку на месте одной из трёх цветовых компонент.



TN TFT – первая технология, появившаяся на рынке LCD, которая до сих пор чувствует себя уверенно в категории бюджетных решений, поскольку создание подобных цифровых панелей обходится относительно дёшево. Но, как и многие другие дешёвые вещи, LCD мониторы на матрице TN TFT не лишены недостатков. Во-первых, чёрный цвет у старых моделей таких дисплеев больше смахивает на тёмно-серый (поскольку очень трудно было развернуть все жидкие кристаллы строго перпендикулярно к фильтру), что приводит к низкой контрастности картинки. С годами технологический процесс совершенствовался, и новые TN панели демонстрируют значительно увеличившуюся глубину тёмных оттенков. Во-вторых, в случае отказа транзистора (а такое бывает) на экране образуется посторонняя «мёртвая» яркая точка, которая для глаза намного заметнее «мёртвой» чёрной. Но эти два недостатка не мешают данной технологии занимать место лидера среди бюджетных панелей, поскольку главным фактором для бюджетных решений всё равно остаётся стоимость.



Итак, выделим достоинства и недостатки матриц TN+film (во всех исполнениях) на сегодняшний день:

Достоинства:

- высокая скорость переключения ячеек
- низкая цена

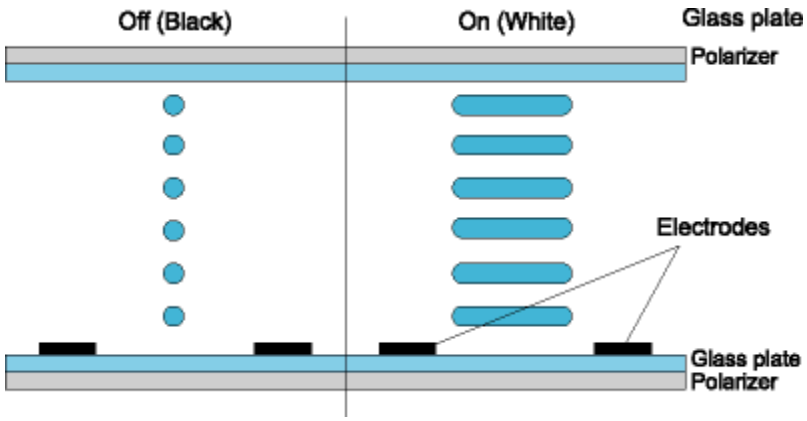
Недостатки:

- абсолютно низкое качество цветопередачи
- малые углы обзора
- низкая контрастность (соотношение между белым и чёрным)

Super-TFT или IPS (In-Plane Switching)

Один из вариантов борьбы с недостатками предложила технология Super-TFT или IPS (In-Plane Switching), разработанная компанией Hitachi. IPS позволила расширить угол обзора до примерно 170 градусов за счёт более точного механизма управления ориентацией жидких кристаллов, что и явилось её главным достижением. Такой важный параметр как контрастность остался на старом уровне TN TFT, а время отклика даже стало больше. Рассмотрим, чем отличается принцип работы Super-TFT от TN TFT:

- При отсутствии электрического поля молекулы жидких кристаллов выстроены вертикально и не влияют на угол поляризации проходящего через них света. Поскольку углы поляризации фильтров перпендикулярны, то свет идущий через выключенный транзистор полностью поглощается вторым фильтром.
- Создаваемое электродами поле поворачивает молекулы жидких кристаллов на 90 градусов относительно позиции покоя, меняя тем самым поляризацию светового потока, который пройдёт второй поляризующий фильтр без помех.

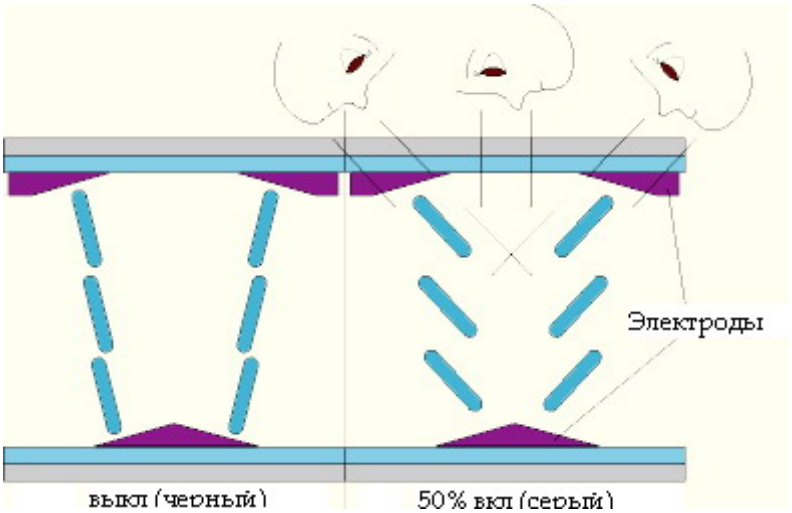


При подаче напряжения молекулы выравниваются параллельно подложке.

Очевиден плюс такого подхода: «мёртвые» пиксели будут гаснуть, а не светиться, как в обычном TN TFT, что менее заметно для глаза..
Благодаря параллельному плоскости экрана расположению жидких кристаллов углы обзора достигают 170° как в горизонтальном, так и в вертикальном направлениях. Время отклика у IPS-матриц невелико - порядка 8 мс. Но, к сожалению, у этой технологии есть и свои минусы (как и у других решений, впрочем). На каждую ячейку в такой матрице приходятся два электрода, расположенных на одной из подложек. Из-за этого шаг между ячейками довольно велик и требуется более мощный источник подсветки, чтобы обеспечить хорошую яркость изображения. Впрочем, решение проблемы с углами обзора, хорошая контрастность и неплохое время отклика привлекли к IPS внимание многих производителей LCD-панелей. Сегодня это одна из наиболее распространенных технологий
Недостатки технологии обусловлены её достоинствами.

- Во-первых, чтобы повернуть весь массив расположенных параллельно кристаллов, требуется время. Поэтому время реакции у мониторов на базе IPS, а также эволюционных продолжений этой технологии S-IPS (Super-IPS) и DD-IPS (DualDomain-IPS) выше, чем у TN+film. Среднее значение для этого типа матриц – 35-25 мс.
- Во-вторых, расположение электродов на одной подложке требует большего напряжения для создания достаточного поля, чтобы повернуть кристаллы в нужное положение. Поэтому мониторы на основе IPS-матриц потребляют больше электроэнергии.
- В-третьих, требуются более мощные лампы, чтобы просветить панель и при этом обеспечить достаточную яркость.
- В-четвёртых, эти панели банально дороги, и до недавнего времени устанавливались только в мониторы с большими диагоналями.
- Одним словом, мониторы на основе матриц этого типа остаются идеальным выбором для дизайнеров и других специалистов, работа которых критична к качеству цветопередачи и некритична к скорости переключения ячеек.

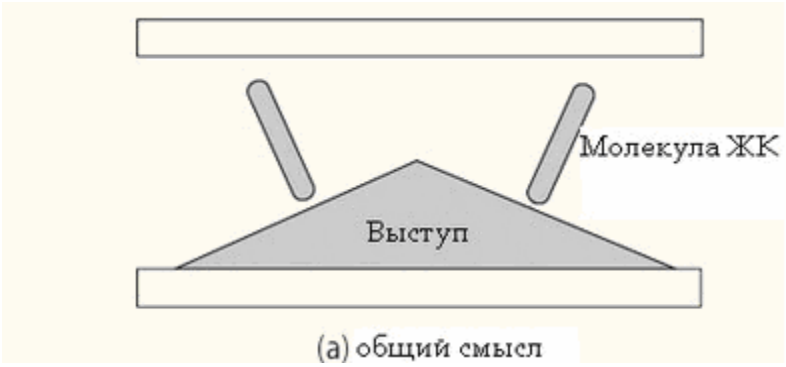
MVA/PVA (Multi-domain Vertical Alignment, PVA - Patterned Vertical Alignment)

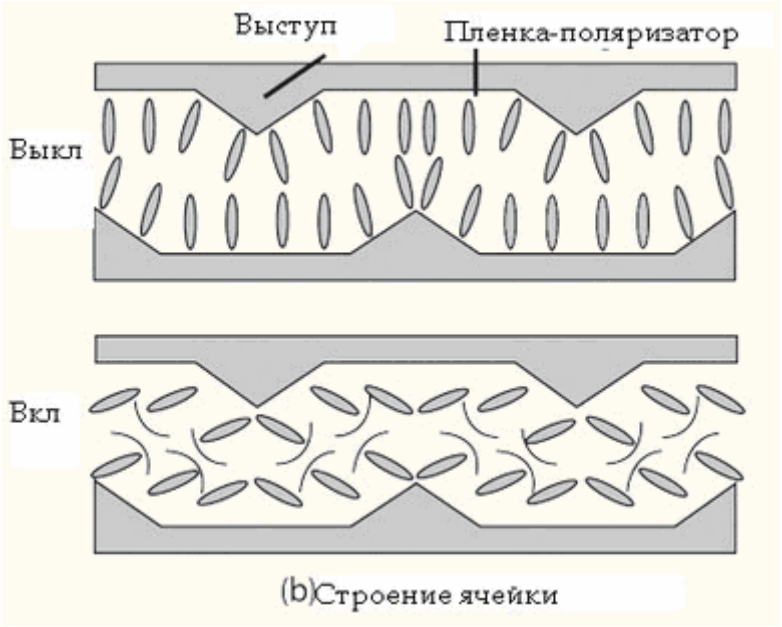


Поскольку с недостатками TN+film бороться стало практически невозможно, а повысить быстродействие S-IPS – так и просто нереально, компания Fujitsu разработала и представила в 1996 году технологию VA (Vertical Alignment). Для коммерческого использования, впрочем, эта технология не подходила и была развита до MVA (Multi-Domain Vertical Alignment). Технология должна была служить компромиссом между быстродействием TN и качеством изображения S-IPS. Потому и реализация во многом схожа с IPS.

В этих матрицах кристаллы располагаются параллельно друг к другу и под углом 90° ко второму фильтру. Таким образом, свет попадает во второй фильтр с осью поляризации, направленной под углом 90° к плоскости поляризации фильтра, и поглощается. В результате мы получаем незасвеченный чёрный цвет на экране. Подавая напряжение на ячейку, мы поворачиваем кристаллы и получаем светящийся пиксель.

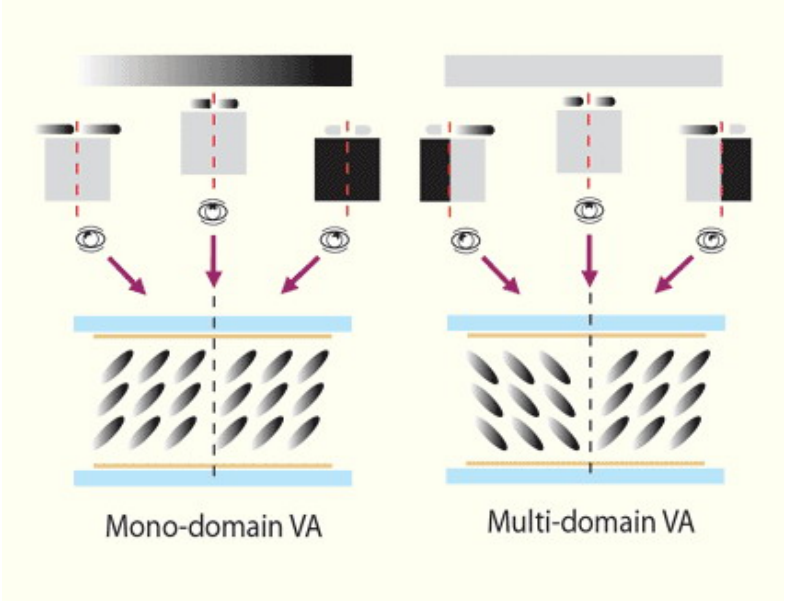
Недостатком первых матриц VA было то, что цвет резко изменялся при смене угла обзора по горизонтали. Для того, чтобы понять это явление, представьте себе, что кристаллы повернуты на 45 градусов и показывают светло-красный цвет. Теперь смещаемся в одну сторону. Угол обзора растёт, и мы получаем уже намного более насыщенный красный цвет. Смещаясь в другую сторону, мы видим, как цвет уходит в противоположную часть спектра и становится зелёным. Поэтому и была разработана MVA. Суть её состоит в том, что поляризационные фильтры были значительно усложнены, а на стеклянную подложку стали наноситься не плоские электроды, а своеобразные треугольники.





Строение MVA

При отключённом токе кристаллы всегда выстраиваются перпендикулярно подложке, так что, с какой бы стороны мы ни смотрели, всегда будет чёрный. При включённом же токе, как всегда, кристаллы поворачиваются на нужный угол и поворачивают вектор поляризации света. Вот только угол этот – между плоскостью электрода и кристалла. Если мы смотрим под углом, мы всегда увидим только одну зону, кристаллы в которой расположены как раз в таком положении, чтобы не искажать цвет. Вторая зона видна не будет.

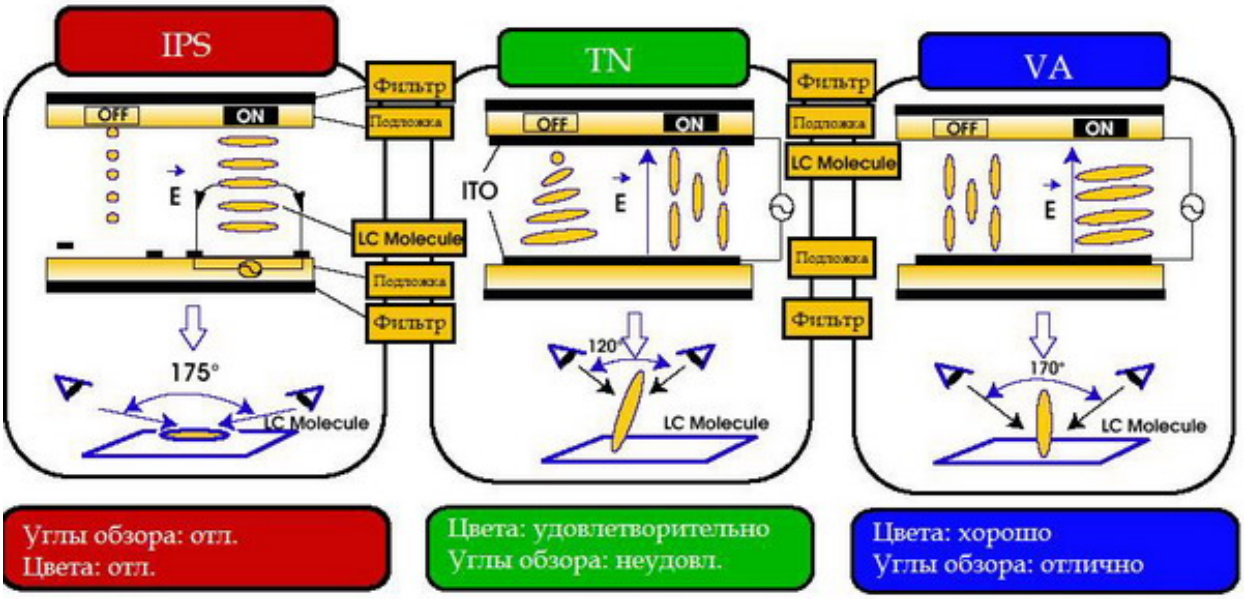


Подобное решение значительно усложняет как фильтры-поляризаторы, так и сами панели, потому что каждую точку на экране нужно дублировать для двух зон.

Как и в S-IPS, у MVA недостатки обусловлены достоинствами. Налицо всё та же инерционность – время отклика выше, чем у TN. Впрочем, на данный момент отличие уже абсолютно некритично: значение достигло 5 мс. Контрастность и яркость намного лучше S-IPS, до 1000:1. Цветопередача матриц MVA считается компромиссной между TN и S-IPS: она не настолько хороша, чтобы применять её для серьёзной работы с полиграфией и дизайном, но намного превышает жутковатые показатели TN+film.

Компания Samsung не пожелала платить лицензионные отчисления Fujitsu и разработала PVA. Впрочем, технологии эти очень похожи, а отличия незначительны. Единственное существенное – большая контрастность, что только плюс. Поэтому довольно часто в характеристиках монитора в графе «тип матрицы» пишут MVA/PVA.

Теперь если объединить все изображения принципов работы каждой из матриц вместе получим следующее вид



2.2.5 Основные параметры ЖК-мониторов

Несмотря на то, что время отклика ячейки – далеко не самый важный показатель, чаще всего при выборе монитора покупатель обращает внимание только на этот фактор. Собственно, именно поэтому TN+film и доминирует. Однако при выборе конкретной модели стоит обдуманно взвешивать все характеристики монитора.

Время отклика

Этот показатель означает минимальное время, за которое ячейка жидкокристаллической панели изменяет цвет. Существуют два способа измерения скорости матрицы: black to black, чёрный-белый-чёрный, и gray to gray, между градациями серого. Эти значения очень сильно различаются.

При изменении состояния ячейки между крайними положениями (чёрный-белый) на кристалл подаётся максимальное напряжение, поэтому он поворачивается с максимальной скоростью. Именно так получены значения в 8, 6, а иногда и 4 мс в характеристиках современных мониторов.

При смещении кристаллов между градациями серого на ячейку подаётся намного меньшее напряжение, потому что позиционировать их нужно точно для получения нужного оттенка. Поэтому и времени для этого затрачивается намного больше (для матриц 16 мс – до 27-28 мс).
Лишь недавно в конечных продуктах смогли воплотить достаточно логичный способ решения этой проблемы. На ячейку подаётся максимальное напряжение (или сбрасывается до нуля), а в нужный момент моментально выводится на нужное для удержания положения кристалла. Сложностью является чёткое управление напряжением с частотой, превышающей частоту развёртки. Кроме того, импульс нужно высчитывать с учётом начального положения кристаллов. Однако Samsung уже представила модели с технологией Digital Capacitance Compensation, дающей показатели 8-6 мс для матриц PVA.

Контрастность

Значение контрастности определяется по соотношению яркости матрицы в состоянии «чёрный» и «белый». Т.е. чем меньше засвечен чёрный цвет и чем выше яркость белого, тем выше контрастность. Этот показатель критичен для просмотра видео, изображений и, в принципе, для хорошего отображения любого изображения. Выглядит как, например, 250:1, т.е. яркость матрицы в «белом» состоянии – 250 кд/м 2, а в «чёрном» – 1 кд/м 2. Впрочем, такие значения возможны только в случае TN+film, для S-IPS среднее значение – 400:1, а для PVA – до 1000:1.
Впрочем, заявленным в характеристиках монитора значениям стоит верить только с натяжкой, потому что это значение замеряется для матрицы, а не для монитора. И замеряется оно на специальном стенде, когда на матрицу подаётся строго стандартное напряжение, подсветка питается строго стандартным током и т.д.

Яркость

Измеряется в кд/м 2. Важна для работы с изображениями, для красочных игр и видео. Зависит от мощности лампы подсветки и, косвенно, от типа матрицы (помните недостатки S-IPS?).

Углы обзора

Обычно указываются значения 170°/170°, впрочем, для TN+film это значение – не больше чем декларация. Требованием при определении углов обзора является сохранение контрастности не ниже 10:1. При этом абсолютно безразлична цветопередача в таком положении, даже если цвета будут инвертированы. Также учитываем, что углы определяются в центре матрицы, а на углы мы, естественно, изначально смотрим под углом.

Цветопередача

До пересечения рубежа в 25 мс при переключении ячейки в порядке чёрный-белый-чёрный все матрицы TN отображали честный 24-битный цвет. Однако в гонке скоростей AU Optronics решила честную цветопередачу отбросить. Начиная с матриц со скоростью 16 мс, все TN+film обеспечивают только 262 тысячи оттенков (18 бит). Большее же количество оттенков обеспечивается двумя путями: либо перемешиванием точек с разными цветами (дизеринг), либо сменой цвета ячейки при каждом обновлении картинки (Frame Rate Control, FRC). Второй способ «честней», потому как человеческий глаз всё равно не успевает заметить смены цвета на каждом кадре. Подчеркиваем, только матрицы TN+film – 18-битные, матрицы, произведённые по другим технологиям, поддерживают 24-битную цветопередачу.

2.3 Плазменные дисплеи.

Сравнительно недавно, в 90-е гг прошлого века на экранов магазинов появилась альтернативная технология - плоскопанельный плазменный дисплей. Такие телевизоры имеют широкие экраны, больше самых больших ЭЛТ, при этом они всего около 15 см. в толщину. `Бортовой компьютер` плазменной панели последовательно зажигает тысячи и тысячи крошечных точек-пикселей. В большинстве систем покрытие пикселей использует три цвета - красный, зеленый и синий. Комбинируя эти цвета телевизор может создавать весь цветовой спектр.

Таким образом, каждый пиксель создан из трех ячеек, представляющих собой крошечные флуоресцентные лампы. Как и в ЭЛТ-телевизоре, для создания всего многообразия оттенков цветов меняется интенсивность свечения ячеек.

Основа каждой плазменной панели - это собственно плазма, т. е. газ, состоящий из ионов (электрически заряженных атомов) и электронов (отрицательно заряженных частиц). В нормальных условиях газ состоит из электрически нейтральных, т. е. не имеющих заряда частиц. Отдельные атомы газа содержат равное число протонов (частиц с положительным зарядом в ядре

атома) и электронов. Электроны `компенсируют` протоны, таким образом, что общий заряд атома равен нулю.

Если ввести в газ большое число свободных электронов, пропустив через него электрический ток, ситуация меняется радикально. Свободные электроны сталкиваются с атомами, `выбивая` все новые и новые электроны. Без электрона меняется баланс, атом приобретает положительный заряд и превращается в ион.

Когда электрический ток проходит через образовавшуюся плазму, отрицательно и положительно заряженные частицы стремятся друг к другу.

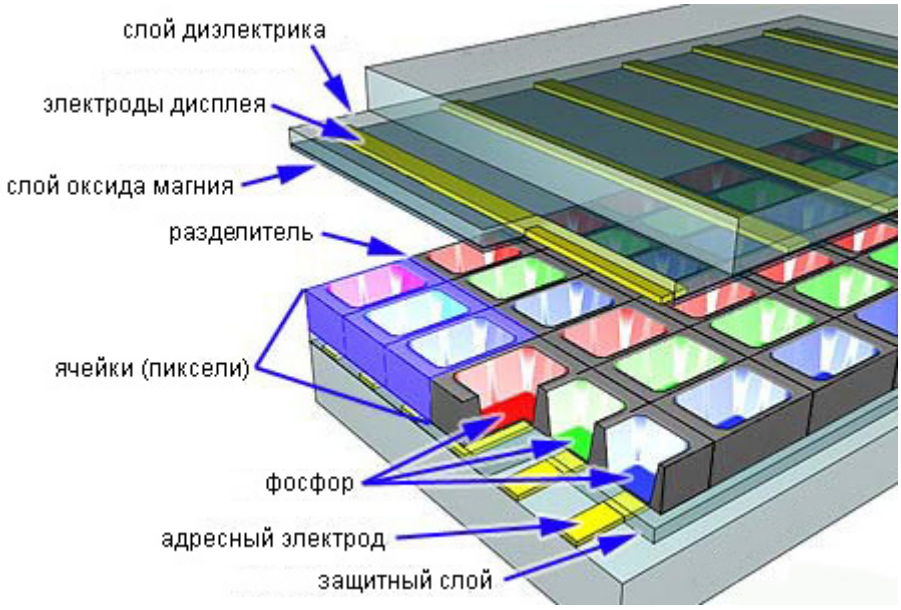
Среди всего этого хаоса частицы постоянно сталкиваются. Столкновения `возбуждают` атомы газа в плазме, заставляя из высвобождать энергию в виде фотонов.

В плазменных панелях используются в основном инертные газы - неон и ксенон. В состоянии `возбуждения` они испускают свет в ультрафиолетовом диапазоне, невидимом для человеческого глаза. Тем не менее, ультрафиолет можно использовать и для высвобождения фотонов видимого спектра.

2.3.1 Внутреннее строение дисплея

В плазменном телевизоре `пузырьки` газов неона и ксенона размещены в сотни и сотни тысяч маленьких ячеек, сжатых между двумя стеклянными панелями. Между панелями по обеим сторонам ячеек расположены также длинные электроды. `Адресные` электроды находятся за ячейками, вдоль задней стеклянной панели. Прозрачные электроды покрыты диэлектриком и защитной пленкой оксида магния (MgO). Они располагаются над ячейками, вдоль передней стеклянной панели.

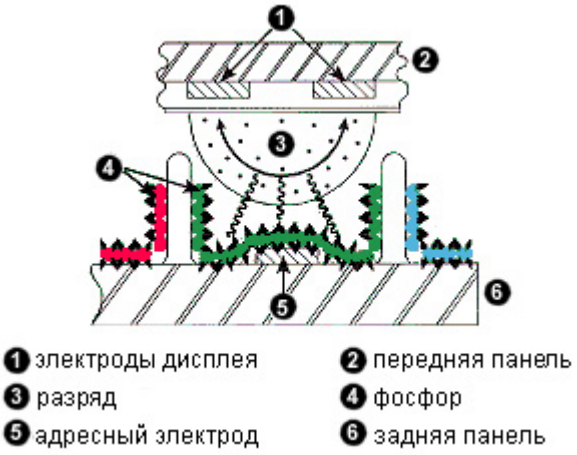
Обе `сетки` электродов перекрывают весь дисплей. Электроды дисплея выстроены в горизонтальные ряды вдоль экрана, а адресные электроды расположены вертикальными колонками. Как видно на рисунке ниже, вертикальные и горизонтальные электроды формируют базовую сетку.



Для того, чтобы ионизировать газ в отдельной ячейке, компьютер плазменного дисплея заряжает те электроды, которые на ней пересекаются. Он делает это тысячи раз за малую долю секунды, заряжая каждую ячейку дисплея по очереди.

Когда пересекающиеся электроды заряжены, через ячейку проходит электрический разряд. Поток заряженных частиц заставляет атомы газа высвобождать фотоны света в ультрафиолетовом диапазоне.

Фотоны взаимодействуют с фосфорным покрытием внутренней стенки ячейки. Как известно, фосфор - материал, под действием света сам испускающий свет. Когда фотон света взаимодействует с атомом фосфора в ячейке, один из электронов атома переходит на более высокий энергетический уровень. После чего электрон смещается назад, при этом высвобождается фотон видимого света.



Пиксели в плазменной панели состоят из трех ячеек-субпикселей, каждая из которых имеет свое покрытие - из красного, зеленого или синего фосфора. В ходе работы панели эти цвета комбинируются компьютером, создаются новые цвета пикселя. Меняя ритм пульсации тока, проходящего через ячейки, контрольная система может увеличивать или уменьшать интенсивность свечения каждого субпикселя, создавая сотни и сотни различных комбинаций красного, зеленого и синего цветов.

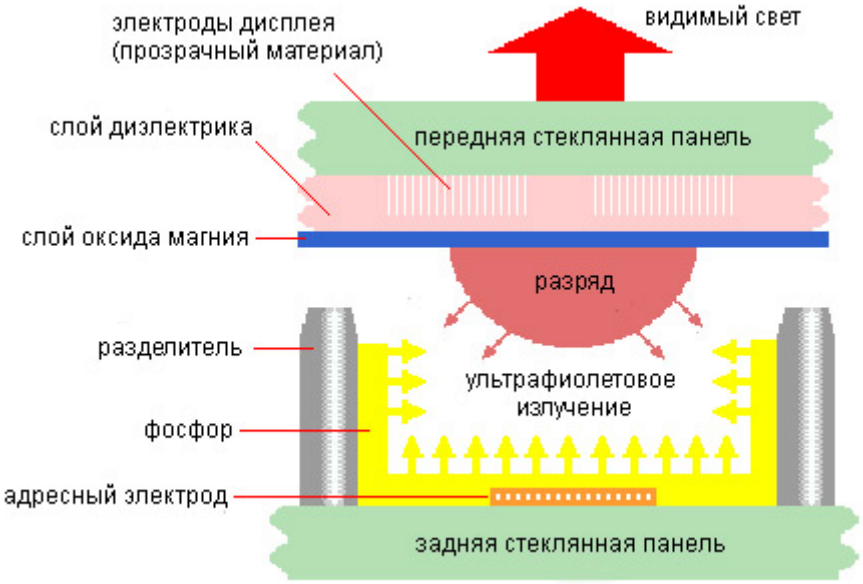
Главное преимущество производства плазменных дисплеев - возможность создавать тонкие панели с широкими экранами. Поскольку свечение каждого пикселя определяется индивидуально, изображение выходит потрясающе ярким, причем при просмотре под любым углом. В норме насыщенность и контрастность изображения несколько уступает лучшим моделям ЭЛТ-телевизоров, но вполне оправдывает ожидания большинства покупателей. Главный недостаток

плазменных панелей - их цена. Дешевле пары тысяч долларов новую плазменную панель купить невозможно, модели hi-end класса обойдутся в десятки тысяч долларов. Впрочем, с течением времени технология значительно усовершенствовалась, цены продолжают падать. Сейчас плазменные панели начинают уверенно теснить ЭЛТ-телевизоры. особенно это заметно в богатых, технологически развитых странах. В ближайшем будущем `плазма` придет в дома даже небогатых покупателей.

2.3.2 Описание работы плазмы другими словами

Плазменные панели немного похожи на ЭЛТ-телевизоры - покрытие дисплея использует способный светиться фосфоросодержащий состав. В то же время они, как и LCD, используют сетку электродов с защитным покрытием из оксида магния для передачи сигнала на каждый пиксель-ячейку. Ячейки заполнены интернтыми, т. н. `благородными` газами - смесью неона, ксенона, аргона.

Проходящий через газ электрический ток заставляет его светиться. По сути, плазменная панель представляет собой матрицу из крошечных флуоресцентных ламп, управляемых при помощи встроенного компьютера панели. Каждый пиксель-ячейка является своеобразным конденсатором с электродами. Электрический разряд ионизирует газы, превращая их в плазму - т. е. электрически нейтральную, высокоионизированную субстанцию, состоящую из электронов, ионов и нейтральных частиц. Будучи электрически нейтральной, плазма содержит равное число электронов и ионов и является хорошим проводником тока. После разряда плазма испускает ультрафиолетовое излучение, заставляющий светиться фосфорное покрытие ячеек-пикселей. Красную, зеленую или синюю составляющую покрытия.



На самом деле каждый пиксель делится на три субпикселя, содержащих красный, зеленый либо синий фосфор. Для создания разнообразных оттенков цветов интенсивность свечения каждого субпикселя контролируется независимо. В кинескопных телевизорах это делается путем изменения интенсивности потока электронов, в `плазме` - при помощи 8-битной испульсной кодовой модуляции. Общее число цветовых комбинаций в этом случае достигает 16,777,216 оттенков.

Тот факт, что плазменные панели сами являются источником света, обеспечивает отличные углы обзора по вертикали и горизонтали и великолепную цветопередачу (в отличие от, например, LCD, экраны в которых обычно нуждаются в подсветке матрицы). Впрочем, обычные плазменные дисплеи в норме страдают от низкой контрастности. Это обусловлено необходимостью постоянно подавать низковольтный ток на все ячейки. Без этого пиксели будут `включаться` и `выключаться` как обычные флуоресцентные лампы, то есть очень долго, непозволительно увеличивая время отклика. Таким образом, пиксели должны оставаться выключенными, в то же время испуская свет низкой интенсивности, что, конечно, не может не сказаться на контрастности дисплея. В конце 90-

х гг. прошлого века Fujitsu удалось несколько смягчить остроту проблемы, улучшив контрастность своих панелей с 70:1 до 400:1. К 2000 году некоторые производители заявляли в спецификациях панелей контрастность до 3000:1, сейчас - уже 10000:1+.

Процесс производства плазменных дисплеев несколько проще, чем процес производства LCD. В сравнении с выпуском TFT LCD-дисплеев, требующим использования фотолитографии и высокотемпературных технологий в стерильно чистых помещениях, `плазму` можно выпускать в цехах погрязнее, при невысоких температурах, с использованием прямой печати. Тем не менее, век плазменных панелей недолог - совсем недавно среднестатистический ресурс панели равнялся 25000 часов, сейчас он почти удвоился, но проблему это не снимает. В пересчете на часы работы плазменный дисплей обходится дороже LCD. Для большого презентационного экрана разница не очень существенная, однако, если оснастить плазменными мониторами многочисленные офисные компьютеры, выигрыш LCD становится очевидным для компании-покупателя.

Еще один важный недостаток `плазмы` - большой размер пикселей. Большинство производителей неспособны создавать ячейки менее 0,3 мм - это больше, чем зерно стандартного компьютерного монитора. Непохоже, чтобы в ближайшем будущем ситуация изменилась к лучшему. На среднесрочную перспективу такие плазменные дисплеи подойдут в качестве домашних телевизоров и презентационных экранов до 70+ дюймов размером. Если `плазму` не уничтожат LCD и появляющиеся каждый день новые дисплейные технологии, через какой-нибудь десяток лет она будет доступна любому покупателю.

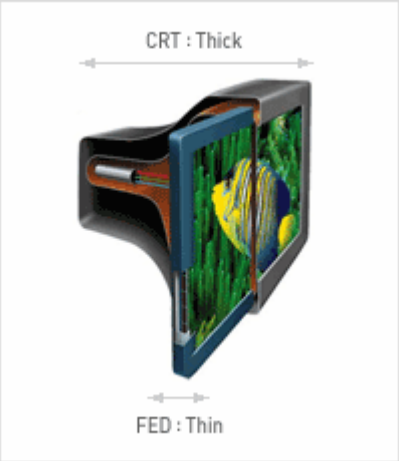
2.4 FED.

Технологии, которые применяются при создании мониторов, могут быть разделены на две группы:

1) мониторы, основанные на излучении света, например, традиционные CRTмониторы и плазменные, т.е. это устройства, элементы экрана которых излучают свет во внешний мир

2) мониторы трансляционного типа, такие, как LCD-мониторы.

Одним из лучших технологических направлений в области создания мониторов, которое совмещает в себе особенности обеих технологий, описанных нами выше, является технология



FED (**Field Emission Display**). Мониторы FED основаны на процессе, который немного похож на тот, что применяется в CRT-мониторах, так как в обоих методах применяется люминофор, светящийся под воздействием электронного луча. Главное отличие между CRT и FED мониторами состоит в том, что CRT-мониторы имеют три пушки, которые испускают три электронных луча, последовательно сканирующих панель, покрытую люминофорным слоем, а в FED-мониторе используется множество маленьких источников электронов, расположенных за каждым элементом экрана, и все они размещаются в пространстве, по глубине меньшем, чем требуется для CRT. Каждый источник электронов управляется отдельным электронным элементом, так же, как это происходит в LCD-мониторах, и каждый пиксель затем излучает свет, благодаря воздействию электронов на люминофорные элементы, как и в традиционных CRT-мониторах. При этом FED-мониторы очень тонкие.

2.5 Безопасность мониторов.

Теперь, после того, как мы поговорили о принципах работы, логично перейти к вопросу о стандартах безопасности. Тем более, что на всех современных мониторах можно встретить наклейки с аббревиатурой TCO или MPRII. На очень старых моделях встречаются еще и надписи "Low Radiation", которые на самом деле ни о чем не говорят. Просто когда-то,

исключительно в маркетинговых целях, производители из ЮгоВосточной Азии привлекали этим внимание к своей продукции. Никакой защиты подобная надпись не гарантирует.

С целью снижения риска для здоровья различными организациями были разработаны рекомендации по параметрам мониторов, следуя которым производители мониторов борются за наше здоровье. Все стандарты безопасности для мониторов регламентируют максимально допустимые значения электрических и магнитных полей, создаваемых монитором при работе. Практически в каждой развитой стране есть собственные стандарты, но особую популярность во всем мире (так сложилось исторически) завоевали стандарты, разработанные в Швеции и известные под именами TCO и MPRII. Расскажем о них подробнее.

TCO

"...TCO (The Swedish Confederation of Professional Employees, Шведская Конфедерация Профессиональных Коллективов Рабочих), членами которой являются 1.3 миллиона шведских профессионалов, организационно состоит из 19 объединений, которые работают вместе с целью улучшения условий работы своих членов. Эти 1.3 млн. человек представляют широкий спектр рабочих и служащих из государственного и частного сектора экономики.

TCO никак не связана с политикой или религией, что является одной из определяющих причин, позволяющей объединяться различным коллективным членам под крышей одной организации.

Учителя, инженеры, экономисты, секретари и няньки - лишь немногие из групп, которые все вместе формируют TCO. Это означает, что TCO отражает большой срез общества, что обеспечивает ей широкую поддержку..."

Это была цитата из официального документа TCO. Дело в том, что более 80% служащих и рабочих в Швеции имеют дело с компьютерами, поэтому главная задача TCO - это разработать стандарты безопасности при работе с компьютерами, т.е. обеспечить своим членам и всем остальным безопасное и комфортное рабочее место. Кроме разработки стандартов безопасности, TCO участвует в создании специальных инструментов для тестирования мониторов и компьютеров.

Стандарты TCO разработаны с целью гарантировать пользователям компьютеров безопасную работу. Этим стандартам должен соответствовать каждый монитор, продаваемый в Швеции и в Европе. Рекомендации TCO используются производителями мониторов для создания более качественных продуктов, которые менее опасны для здоровья пользователей. Суть рекомендаций TCO состоит не только в определении допустимых значений различного типа излучений, но и в определении минимально приемлемых параметров мониторов, например, поддерживаемых разрешений, интенсивности свечения люминофора, запас яркости, энергопотребление, шумность и т.д. Более того, кроме требований, в документах TCO приводятся подробные методики тестирования мониторов.

Рекомендации TCO применяются как в Швеции, так и во всех Европейских странах для определения стандартных параметров, которым должны соответствовать все мониторы. В состав разработанных TCO рекомендаций сегодня входят четыре стандарта: TCO'92, TCO'95, TCO'99 и TCO '03, нетрудно догадаться, что цифры означают год их принятия. Большинство измерений во время тестирований на соответствие стандартам TCO проводятся на расстоянии 30 см спереди от экрана и на расстоянии 50 см вокруг монитора. Для сравнения: во время тестирования мониторов на соответствие другому стандарту MPRII все измерения производятся на расстоянии 50 см спереди экрана и вокруг монитора. Это объясняет то, что стандарты TCO более жесткие, чем MPRII.

TCO '92

Стандарт TCO'92 был разработан исключительно для мониторов и определяет величину максимально допустимых электромагнитных излучений при работе монитора, а также устанавливает стандарт на функции энергосбережения мониторов. Кроме того, монитор, сертифицированный по TCO'92, должен соответствовать стандарту на

энергопотребление NUTEK и соответствовать Европейским стандартам на пожарную и электрическую безопасность.

TCO '95

Стандарт TCO'92 рассчитан только на мониторы и их характеристики относительно электрических и магнитных полей, режимов энергосбережения и пожарной и электрической безопасности. Стандарт TCO'95 распространяется на весь персональный компьютер, т.е. на монитор, системный блок и клавиатуру, и касается эргономических свойств, излучений (электрических и магнитных полей, шума и тепла), режимов энергосбережения и экологии (с требованием к обязательной адаптации продукта и технологического процесса производства на фабрике). Заметим, что в данном случае термин "персональный компьютер" включает в себя рабочие станции, серверы, настольные и напольные компьютеры, а также компьютеры Macintosh.

Стандарт TCO'95 существует наряду с TCO'92 и не отменяет последний. Требования TCO'95 по отношению к электромагнитным излучениям мониторов не являются более жесткими, чем по TCO'92.

Отметим, что LCD и плазменные мониторы также могут быть сертифицированы по стандартам TCO'92 и TCO'95, как, впрочем, и портативные компьютеры.

К слову, мыши не подлежат сертификации TCO'95 :)

В разработке стандарта TCO'95 принимали совместное участие четыре организации: TCO, Naturskyddforeinegen, NUTEK и SEMKO AB.

Naturskyddforeinegen (The Swedish Society for Nature Conservation) - Шведское общество защиты природы. Это их знак в виде летящего сокола размещен на эмблеме TCO'95.

NUTEK (The National Board for Industrial and Technical Development in Sweden) - Шведская правительственная организация, занимающаяся исследованиями в области энергосбережения и эффективного использования энергии.

Компания SEMKO AB занимается тестированием и сертификацией электрических продуктов. Это независимое подразделение группы British Inchcape. SEMKO AB разработала тесты для TCO'95 сертификации и проверки сертифицированных устройств.

TCO '99

TCO'99 предъявляет более жесткие требования, чем TCO'95, в следующих областях: эргономика (физическая, визуальная и удобство использования), энергия, излучение (электрических и магнитных полей), окружающая среда и экология, а также пожарная и электрическая безопасность. Стандарт TCO'99 распространяется на традиционные CRT-мониторы, плоскопанельные мониторы (Flat Panel Displays), портативные компьютеры (Laptop и Notebook), системные блоки и клавиатуры.

Экологические требования включают в себя ограничения на присутствие тяжелых металлов, броминатов и хлоринатов, фреонов (CFC) и хлорированных веществ внутри материалов.

Любой продукт должен быть подготовлен к переработке, а производитель обязан иметь разработанную политику по утилизации, которая должна исполняться в каждой стране, в которой действует компания.

Требования по энергосбережению включают в себя необходимость того, чтобы компьютер и/или монитор после определенного времени бездействия снижали уровень потребления энергии на одну или более ступеней. При этом период времени восстановления до рабочего режима потребления энергии, должен устраивать пользователя.

TCO '03

Спустя 4 года после TCO '99, TCO Development продолжает трудиться во славу все более строгих требований по безопасности, предъявляемых к различным устройствам. В конце января 2003 года началась сертификация по новому стандарту TCO'03. Так и хочется спросить, потирая руки: "Ну-с, что новенького?". На этот раз стандарт безопасности, в соответствии с веяниями моды, вновь концентрирует свое внимание на дисплеях мониторов. При этом затрагиваются оба распространенных типа экранов электронно-лучевых и жидкокристаллических. Разумеется, требования к ним несколько отличаются друг от друга.

Перечень основных требований к параметрам мониторов на базе электронно-лучевых трубок выглядит следующим образом:

1. Максимальная яркость должна составлять не менее 120 кд/м2, при допустимых ранее 100 кд/м2. Весьма существенная поправка к TCO'99.

2. Мониторы с диагональю от 22 дюймов и более, должны поддерживать частоту 85 Гц при разрешении 1600x1200. Ранее допускалась поддержка данной частоты на более низком разрешении - 1280x1024, при диагонали от 21 дюйма и более.
3. Монитор должен обладать возможностью поворота корпуса в вертикальной плоскости на угол не менее 20o - это что-то новенькое! В описании TCO'99 подобные злобные требования отсутствовали.

Ограничения также коснулись значений цветовой температуры для того, чтобы добиться максимально возможной по точности цветопередачи. Кроме того, весьма четко TCO'03 ограничивает содержание вредных веществ в используемых при изготовлении мониторов материалах.

По отношению к обретающим в последнее время исключительную популярность жидкокристаллическим мониторам, стандарт TCO'03 не менее строг:

1. На расстоянии 50 см, видимая плотность пикселей должна достигать 30 пикселей на градус. Иначе, требуемое стандартное разрешение должно составлять: для 15- и 16-дюймовых дисплеев не менее 1024x768, больше или равно 1280x1024 для экранов с размером диагонали от 17 до 19 дюймов и не менее 1600x1200 для 21-дюймовых дисплеев.

2. Максимальная яркость увеличилась по сравнению с предыдущим стандартом. Теперь ее значение должно достигать 150 кд/м2, тогда как раньше было достаточно 125 кд/м2.

Некоторые изменения коснулись неравномерности подсветки по полю экрана. Добавились требования по равномерности яркости подсветки в зависимости от угла наблюдения в вертикальной и горизонтальной плоскостях. Также ЖК мониторов коснулись требования, аналогично применяемые к мониторам на базе ЭЛТ, касающиеся злополучной цветопередачи и соответствия используемых при изготовлении вредных материалов экологическим нормам.

Ряд производителей уже может похвастаться обновленными, сертифицированными линейками моделей. Первыми привели в соответствие новому стандарту свои мониторы EIZO, Hyundai, LG, Belinea, Philips и Samsung. Теперь, приобретая новый монитор, вы можете обратить особенное внимание на наличие приметных наклеек, с обозначением соответствия стандарту TCO'03.

MPR II

Это еще один стандарт, также разработанный в Швеции, где правительство и неправительственные организации очень сильно заботятся о здоровье населения страны.

MPRII был разработан SWEDAC (The Swedish Board for Technical Accreditation) и определяет максимально допустимые величины излучения магнитного и электрического полей, а также методы их измерения. MPRII базируется на концепции о том, что люди живут и работают в местах, где уже есть магнитные и электрические поля, поэтому устройства, которые мы используем, такие, как монитор для компьютера, не должны создавать электрические и магнитные поля, большие чем те, которые уже существуют. Заметим, что стандарты TCO требуют снижения излучений электрических и магнитных полей от устройств настолько, насколько это технически возможно, вне зависимости от электрических и магнитных полей, уже существующих вокруг нас. Впрочем, мы уже отмечали, что стандарты TCO жестче, чем MPRII.

Постарайтесь учесть все приведенные выше рекомендации при выбор монитора, но помните две вещи: идеальных мониторов не бывает, и, самое главное - все решают Ваши глаза и Ваши ощущения.

3. Рекомендации по материалам Cisco

А теперь как обычно давайте рассмотрим материалы предлагаемые вам для изучения Cisco Network Academy, сегодня мы с вами ознакомимся с Основные сведения об операционных системах. Вы узнаете о компонентах, функциях и терминологии, относящейся к операционным системам а также с основным предназначения операционной системы.

Глава 5: Основные сведения об операционных системах

Цель данной главы познакомить вас с сравнительным анализом операционных систем с учетом их предназначения, ограничений и совместимости. Рассмотреть каким образом происходит установка операционной системы, и ознакомить с применением наиболее распространенных методов профилактического обслуживания операционных систем.

5.1 Объяснение предназначения операционной системы

Во всех компьютерах операционная система (ОС) обеспечивает интерфейс для взаимодействия между пользователями, приложениями и аппаратным обеспечением. Операционная система загружает компьютер и управляет файловой системой. Почти все современные операционные системы могут поддерживать несколько пользователей, задач или ЦП.

По завершении данного раздела вы освоите: основные характеристик современных операционных систем, и объяснение концепций операционных систем.

5.2 Описание и сравнение операционных систем с учетом их предназначения, ограничений и совместимости

Клиент может попросить технического специалиста выбрать и установить операционную систему. Тип выбранной ОС зависит от требований клиента к компьютеру. А правильно выбрать Операционную систему исходя из требований и поможет вам данный раздел.

5.3 Определение операционной системы на основе потребностей компьютера

Цель данного раздела познакомить вас с приложениями и окружением, совместимыми с операционной системой. Определение минимальных требований к аппаратному обеспечению и совместимости с платформой ОС. И возможна ли вообще инсталляция данной операционной системы на этот компьютер.

5.4 Установка операционной системы

В качестве технического специалиста вам придется выполнять новую установку операционной системы. А это придется делать: при передаче компьютера от одного сотрудника к другому; при повреждении операционной системы; при установке на компьютере нового жесткого диска. И как правильно это сделать вы и узнаете в данном разделе.

5.5 Навигация по графическому пользовательскому интерфейсу (Windows)

Для того, что бы вы максимально быстро могли помочь разобраться с неисправностью в операционной системе, или же подсказать возможную проблему и ее решение, вам необходимо хорошо разбираться и ориентироваться в пользовательском интерфейсе.

5.6 Определение и применение стандартных процедур профилактического обслуживания для операционных систем

План по профилактическому обслуживанию операционной системы составляется для того, чтобы избежать проблем в будущем. Профилактическое обслуживание необходимо выполнять регулярно. А как запланировать профилактическое обслуживание, и правильно провести его, вы узнаете ознакомившись с данным разделом.

5.7 Поиск и устранение неисправностей операционной системы

Проблемы операционных систем могут быть вызваны комбинацией проблем оборудования, приложений и настройки. И вы как технические специалист должны уметь анализировать проблемы и выявлять причины возникновения ошибки для восстановления работы операционной системы. Этот процесс называется поиском и устранением неполадок.

Теперь надеемся у вас уже не возникнет проблем с выполнением процедур резервного копирования которая позволяла бы восстанавливать утерянные данные. С Процедурой профилактического обслуживания, поиска и устранения неполадок в операционной.

В дополнительные сведения об операционных системах, а это глава 12.
Вы узнаете о различиях между обычными и сетевыми операционными системами. Практические занятия помогли вам познакомиться с Windows XP, процедурами создания разделов, настройкой виртуальной памяти и планирования задач. Кроме того, вы получили рекомендации по оптимизации операционных систем, в том числе по устранению неполадок с компьютером.

Поэтому мы настоятельно рекомендуем вам ознакомится с содержимым 12 главы, с последующей сдачей модульного экзамена, для того, что бы быть уверенными, что вы в полном объеме усвоили материал данной главы.