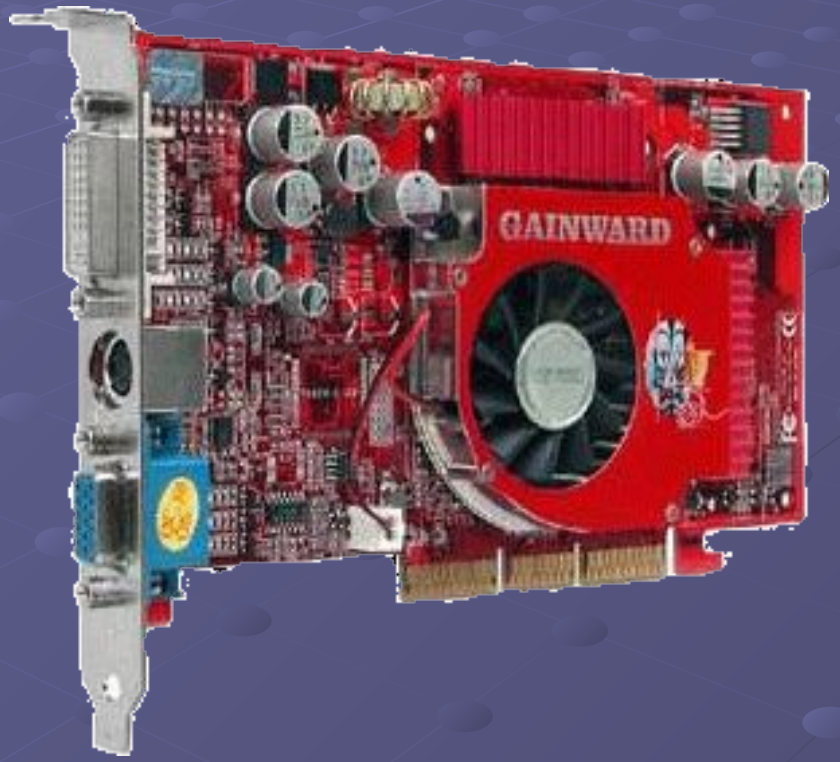
Видеосистема

теоретические основы



Видеосистема современного компьютера включает средства работы с видеоизображениями. Обязательными компонентами системы служат *видеокарта* (графический адаптер, видеоакселератор) и *монитор*, а также обслуживающие их интерфейсы. Дополнительными компонентами часто выступают TV-тюнер, карта видеозахвата, проектор и т.д.

В конце 70-х годов появились первые растровые дисплеи, позволявшие отображать цветное изображение. Формированием картинки занимались специальные устройства – видеоадаптеры.

Основная функция видеоадаптера – преобразование компьютерного сигнала в сигналы подаваемые на монитор

Принципы вывода изображений

Графический режим

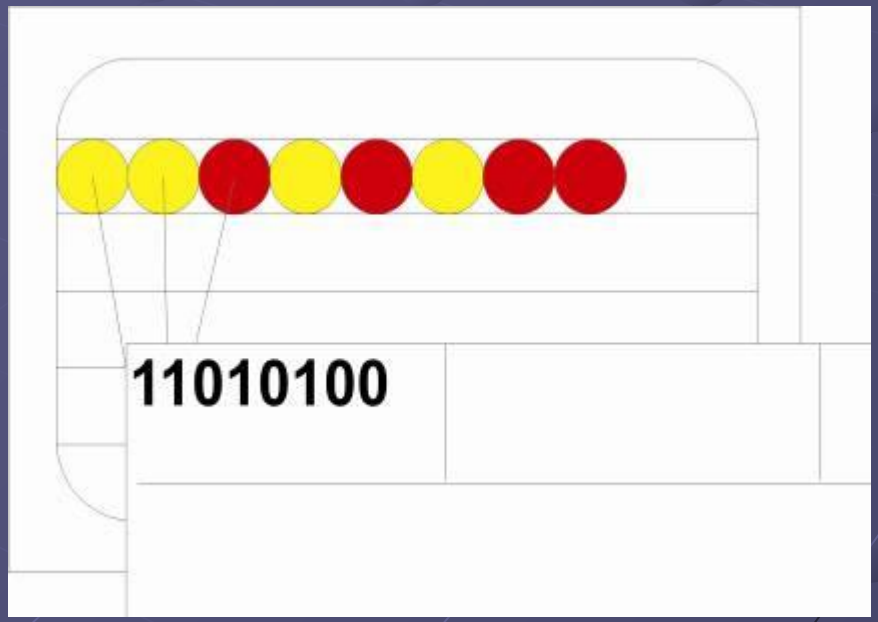
В графическом режиме имеется возможность индивидуального управления свечением каждой точки экрана монитора независимо от состояния остальных. Этот режим обозначают как Gr (Graphics) или APA (All Points Addressable).

В графическом режиме каждой точке экрана (пикселю) соответствует ячейка специальной памяти, которая сканируется схемами адаптера синхронно с движением луча монитора. Такая память называется *видеопамять* (Video Memory, VRAM).

Процесс сканирования видеопамяти называется регенерацией изображения.

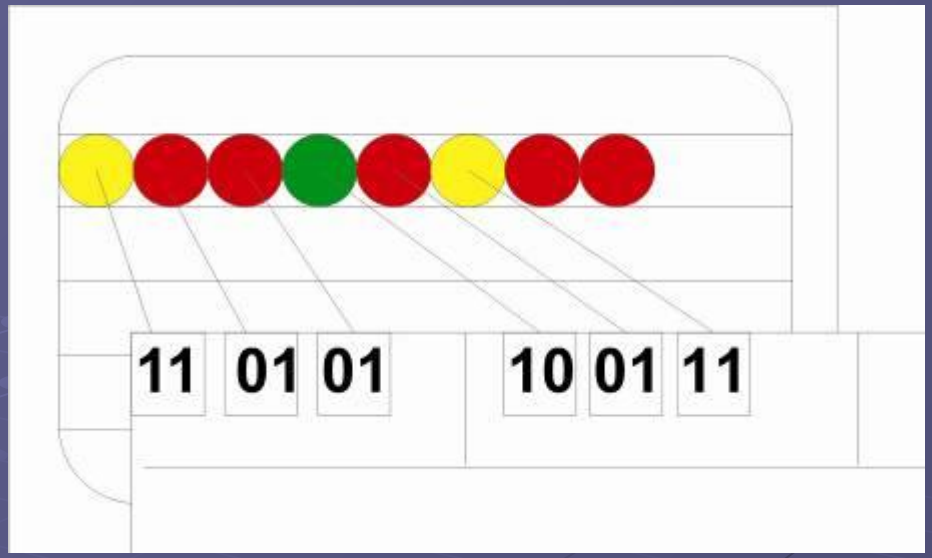
Количество бит видеопамяти, отводимой на каждый пиксель, определяет возможное число состояний пикселя – цветов, градаций яркости и т. п.

Так, при одном бите на пиксель возможны только два состояния – светиться и не светиться



Два бита на пиксель позволяли выводить на экран 4 цвета одновременно (адаптеры CGA).

Четыре бита на пиксель обеспечивают поддержку 16 цветов (адаптеры EGA), что достаточно для многих приложений. Например текстовых редакторов, СУБД, САПР.



8 бит на пиксель дают возможность работать с 256-ю цветами (адаптер VGA)

Сейчас используют режимы (адаптеры SVGA):

High Color – 16 бит 65536 цветов

True Color – 24 или 32 бит >16,7 млн. цветов

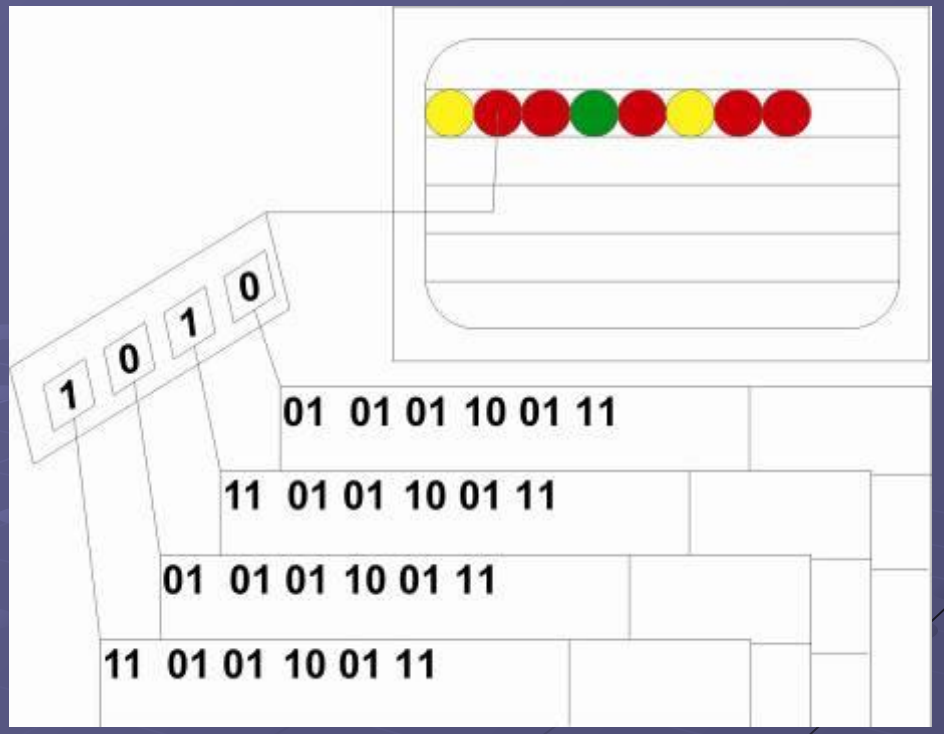
Логически видеопамять может быть организована по-разному, в зависимости от количества бит на пиксель.

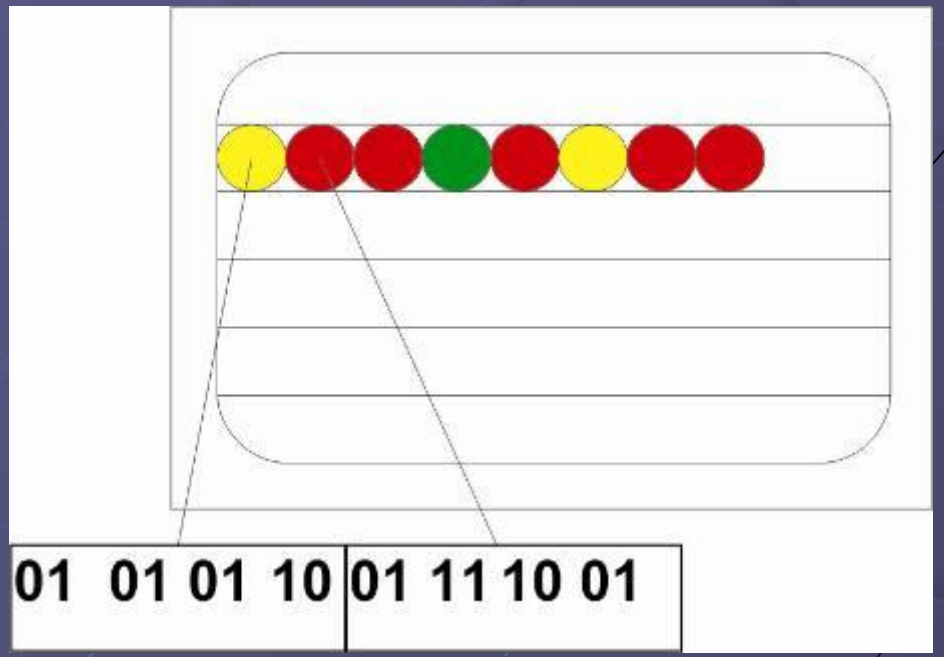
В случае одного или двух бит каждый байт (8 бит) соответствует 8 или 4 соседним пикселям строки. Такой способ называется *линейным.*

В адаптере EGA видеопамять разбивали на 4 слоя – цветовые плоскости. В каждой плоскости используется линейная организация, где каждый байт содержит по одному биту восьми соседних пикселей.

В режимах 8, 16 и 24 бит на пиксель тоже используется линейная организация, но каждый байт отвечает за цвет одного пикселя.

Описанные варианты организации видеопамяти представляют собой отображение матрицы пикселей на биты видеопамяти – *Bit Mapping*. Растровый формат хранения изображений, при котором биты отображают пиксели, называют битовой картой - *Bit Map.*





Формирование битовой карты изображения в видеопамяти графического адаптера может производиться под управлением программы выполняемой центральным процессором. Однако это требует передачи большого объема данных от CPU к видеокарте и обратно. Для решения этой проблемы используют несколько путей:

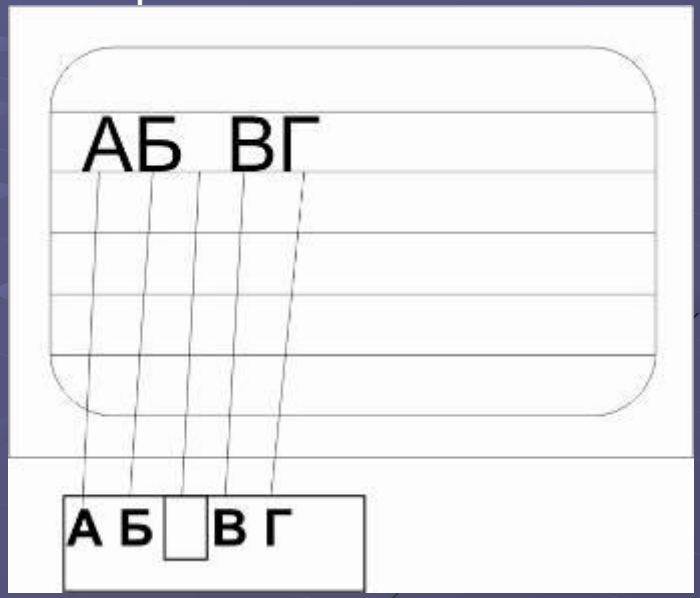
- повышение быстродействия видеопамяти;
- ускорение шины адаптера;
- передача функции формирования битовой карты видеоадаптеру.

Для последнего варианта на видеокарте размещается специализированный графический процессор GPU.

Текстовый режим

В символьном, или текстовом, режиме ячейка видеопамяти хранит информацию о символе, занимающем на экране знакоместо определенного формата. Знакоместо представляет собой матрицу точек, в которой может быть отображен один из символов определенного набора.

В ячейке видеопамяти храниться код символа, определяющий его индекс в таблице символов, и атрибуты символа, определяющие вид его отображения (цвет фона, цвет символа, инверсия, мигание, подчеркивание). Такой режим называют AN или TXT.



В текстовом режиме экран организуется в виде матрицы знакомест. Аналогично организуется видеопамять. Адаптер, работающий в текстовом режиме имеет специальный блок – знакогенератор.

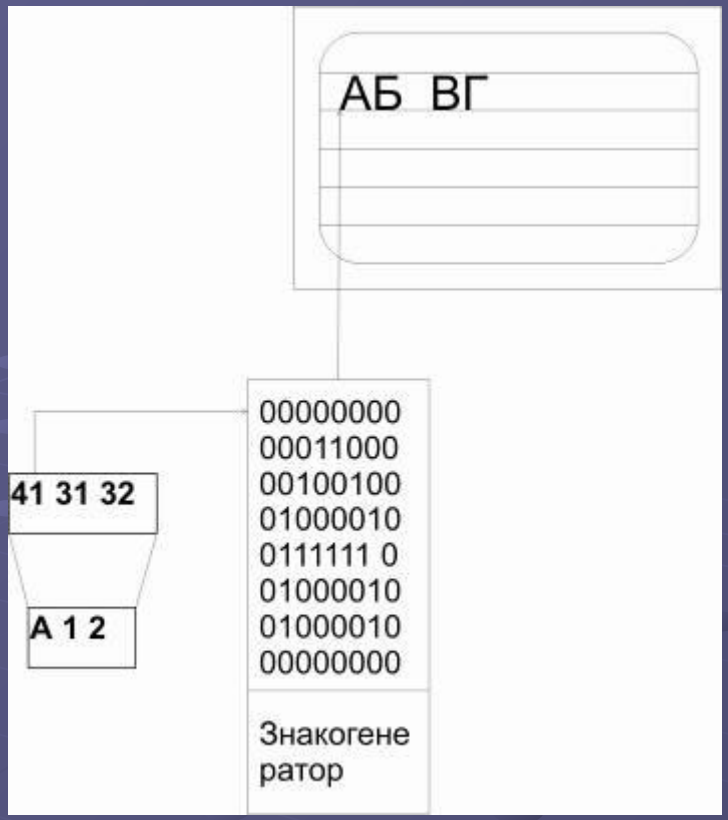
Во время сканирования экрана выборка данных из очередной ячейки производится при подходе к очередному знакоместу. Считанные данные попадают в знакогенератор, который вырабатывает посторочную развертку соответствующего символа – его изображение на экране.

Необходимый объем памяти знакогенератора определяется объемом знакоместа и количеством отображаемых символов.

Минимальный размер знакоместа 8x8 точек. Наилучшее качество обеспечивается при размере 9x16 точек.

Текстовый адаптер имеет аппаратные средства управления курсором. Знакоместо на которое указывают регистры курсора оформляется особым образом. (мигающая полоска)

Несмотря на большее количество узлов, текстовый адаптер дешевле. А программный код для него проще и компактнее.

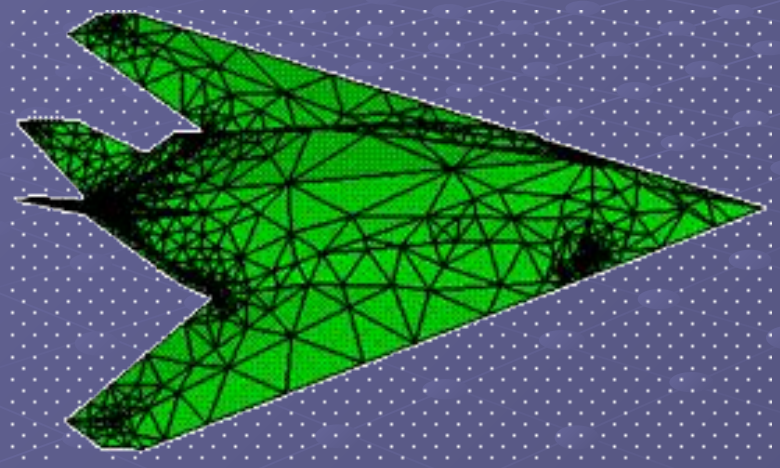


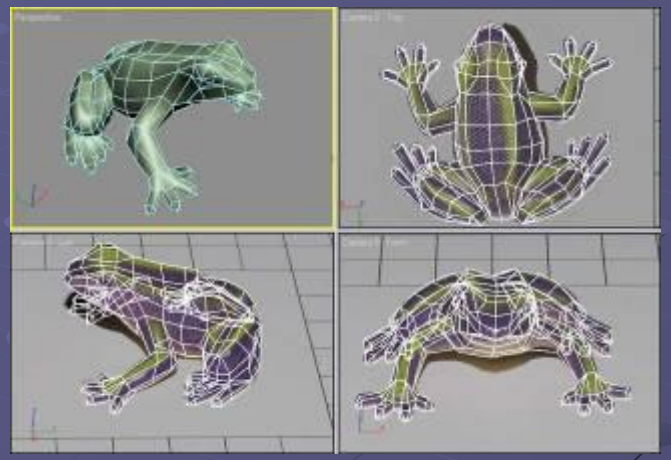
Трехмерная (3D) графика

Потребность в работе с трехмерными изображениями имеется в широком спектре приложений – от игр до САПР используемых в архитектуре, машиностроении.

3D - конвейер

Современные графические процессоры работают с полигональной графикой. То есть любой объект представляется как набор плоских многоугольников, которые рано или поздно разбиваются на простейшие треугольники.



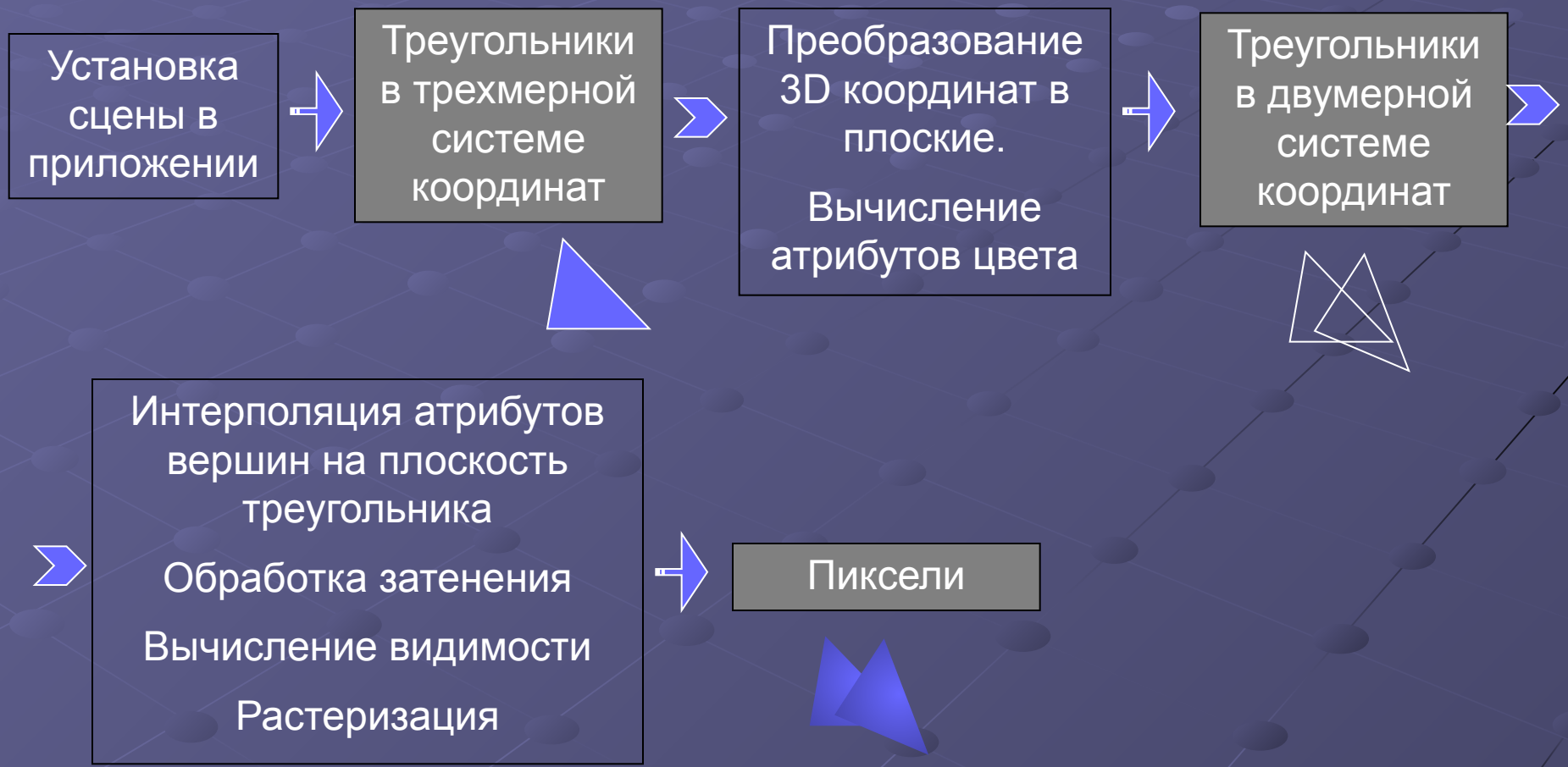


Объект задается вершинами, определяющими ключевые точки, и полигонами, которые образуются линиями, соединяющими вершины. Цвет на полигоны накладывается по специальным алгоритмам закраски, как правило с использованием заранее нарисованных плоских изображений (текстур). Задача GPU сводиться к тому, чтобы нарисовать и закрасить как можно больше полигонов в единицу времени.

В профессиональных 3D ускорителях иногда используется метод обратной трассировки лучей (Ray Tracing), который требует больших вычислительных ресурсов

Современный процессор затрачивает на обработку одной точки около 10 тактов и за секунду просчитывает координаты десятка миллионов вершин. Таким образом расчет геометрии сцены не представляет для современного процессора особого труда. Гораздо более ресурсоемким является процесс закраски полигонов и определения видимых поверхностей.

Программно - аппаратное средство преобразующее описание объектов в матрицу ячеек видеопамяти называют *графическим конвеером*.

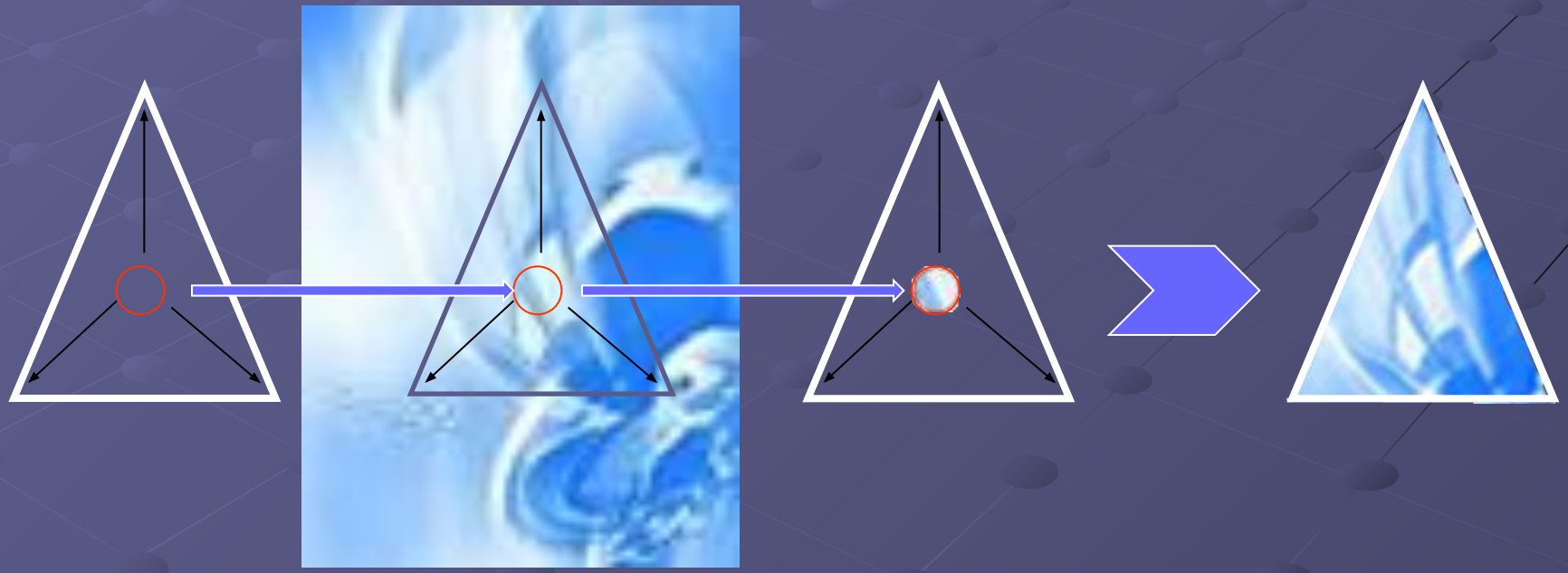


Наложение текстур

Быстродействие графического конвейера и количество конвейеров одновременную работу которых может поддерживать видеокарта сильно влияют на ее производительность.

Реалистичность изображения в трехмерной сцене во многом зависит от качества текстур – заранее нарисованных картинок наложенных на полигоны.

Для каждой вершины указываются ее координаты в плоскости изображения текстуры. При расчете цвета точки полигона учитывается ее расположение относительно вершин полигона и точке присваивается цвет аналогичный цвету соответствующей точки текстуры.

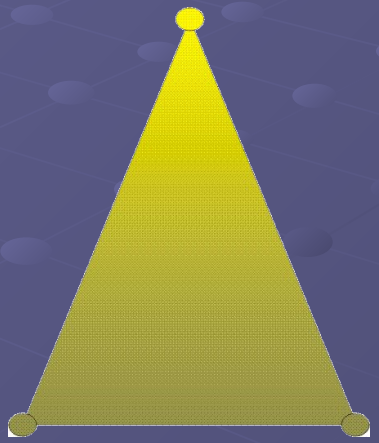


Объекты в сцене, даже с наложенными текстурами выглядят нереалистично если не учитывается их освещенность.

Первые GPU работать с источниками освещения не умели. Освещенность рассчитывалась заблаговременно и создавались соответствующие текстуры освещения (Light maps). Блики света рисовались как текстуры. В динамических сценах, при движении объекта, он смотрелся одинаково на свету и в тени.

Для улучшения реалистичности стали использовать метод Гуро. Определив освещенность вершин полигонов, ее интерполировали на внутренние точки граней.

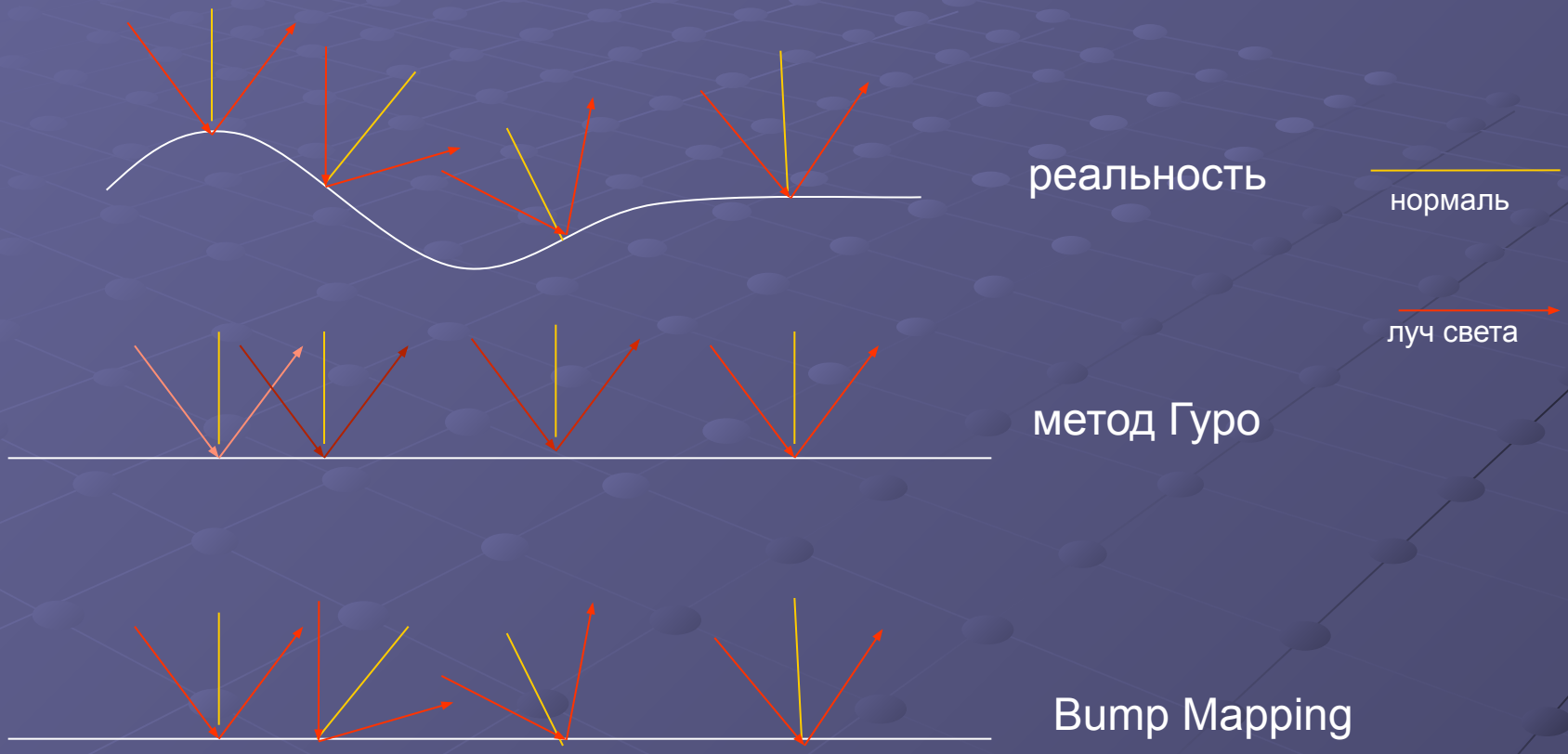
Такой подход лег в основу графических процессоров с фиксированным конвейером (поколение DirectX 7 и OpenGL), получивших блок геометрических вычислений.



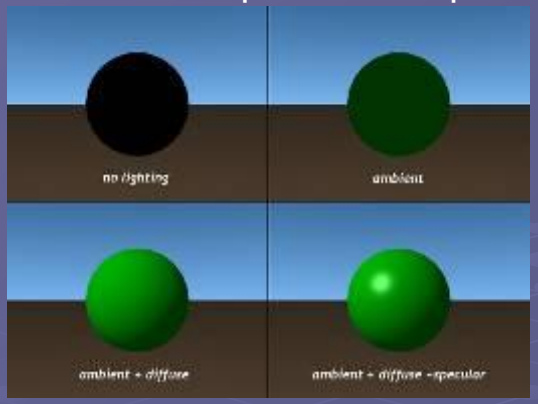
Однако метод Гуро не мог правдоподобно передать фактуру поверхности, так как не учитывал различие в отражении света от неровностей поверхности объекта.

Для решения этой проблемы был разработан метод расчета освещенности с учетом вектора нормали к поверхности в данной точке

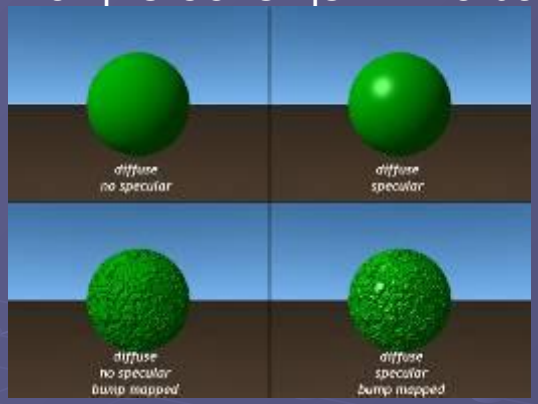
Задав еще одну текстуру специального вида определяющую нормали и модифицировав алгоритм расчета удалось радикально улучшить внешний вид объектов. Подобный метод называют Bump Mapping



Составляющие освещения



Составляющие освещения с bampmapping



Цвет участка поверхности рассчитывается отдельно в каждом из цветовых каналов и складывается из 3-х компонентов: фонового освещения, диффузного отражения, зеркального отражения.

Равномерная (ambient) составляющая освещения - аппроксимация глобального освещения, "начальное" освещение для каждой точки сцены, при котором все точки освещаются одинаково и освещенность не зависит от других факторов.

Диффузная (diffuse) составляющая освещения зависит от положения источника освещения и от нормали поверхности.

Бликовая (specular) составляющая освещения проявляется в бликах отражения лучей света от поверхности. Для ее расчета, помимо вектора положения источника света и нормали, используются еще два вектора: вектор направления взгляда и вектор отражения.

Specular модель освещения впервые предложил Фонг (Phong Bui-Tong). Эти блики существенно увеличивают реалистичность изображения, ведь редкие реальные поверхности не отражают свет. В дальнейшем, исследователи придумывали иные способы вычисления этой составляющей, более сложные (Blinn, Cook-Torrance, Ward).

Итак, Specular Bump Mapping получается таким образом

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


Castrol
MICHELIN
Elbach
SPRINGS
RALLI ART
KONI
Racing
SUBARU
14
SUBARU
RALLY TEAM USA
PIRELLI

Пиксельные шейдеры

Для полноценной реализации Bump Mapping необходим переход от интерполяции и выборки из текстур к обработке формул расчета цвета каждого пикселя объекта а не только вершин. То есть программирование пиксельных конвейеров.

Комплекты команд, позволяющие программировать пиксельные конвейеры графического процессора называют – *пиксельные шейдеры*. (DirectX 8 и старше).

Как правило, программы-шейдеры пишут на специальных разновидностях ассемблера, привязанного к конкретному GPU. В последнее время появились языки программирования высокого уровня для создания шейдеров. Например HLSL.

В целом, от пиксельных шейдеров зависит качество отрисовки цвета, фактуры, материала и освещенности объекта.

Примеры использования пиксельных шейдеров:

Мультитекстурирование. Несколько слоев текстур (colormap, detailmap, lightmap и т.д.)

Попиксельное освещение. Bump mapping. Normal mapping

Примеры использования пиксельных шейдеров:

Постобработка кадра. Bloom, Depth of Field и Motion Blur

Процедурные текстуры, такие, как текстура дерева или мрамора

Вершинные шейдеры (Vertex Shader)

Естественным развитием блока геометрических вычислений стал программируемый блок геометрических преобразований, поддерживающий блок *вершинных шейдеров.*

Программируемый вершинный блок способен имитировать мимику персонажа, переливы меха, развевающиеся волосы, стелящуюся траву, и выполнять прочие геометрические преобразования «на лету».

Программированию поддаются практически все параметры связанные с обработкой вершин: геометрические координаты вершины, текстурные координаты, цвет вершины, параметры смешивания, прозрачность.

Начиная с шейдеров версии 3.0 (DirectX 9), поддерживается использование в вычислениях текстуры и задавать уникальные свойства вершин.

Примеры применения Vertex Shader:

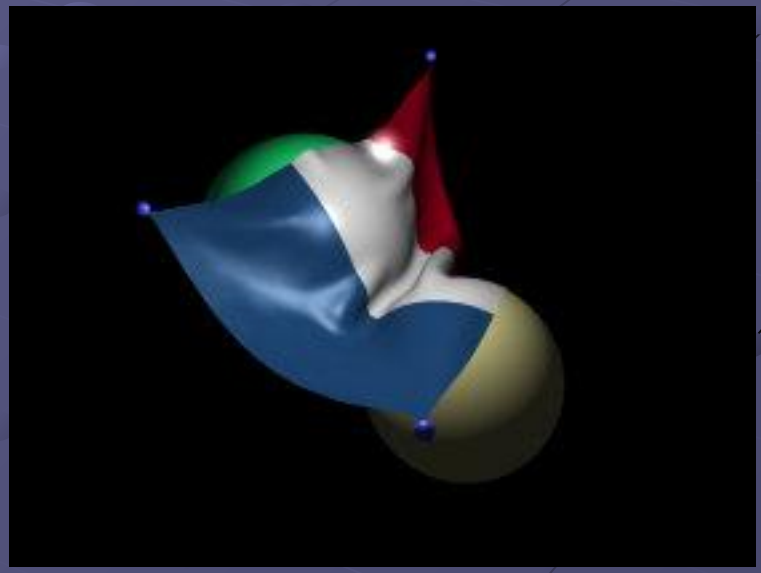
Скининг (skinning). Matrix pallete skinning для скелетной анимации персонажей с большим количеством "костей".

Деформация объектов. Как самый явный и эффектный пример - создание реалистичных волн в динамике. Примеры подобных решений наблюдаются в игре Pacific Fighters, где применяются вершинные шейдеры 3.0 и доступ к текстурам из них.

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

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

Имитация ткани (Cloth Simulation) - для имитации поведения подобных ткани материалов.

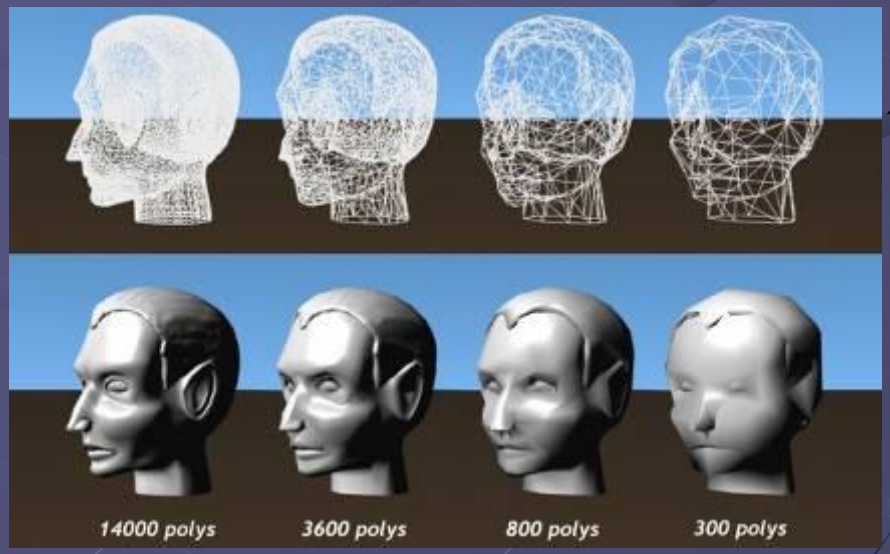


Технологии 3D графики
Уровень детализации LOD

Современные приложения 3D графики позволяют создавать реалистичные объекты с высокой степенью детализации. Однако отображать все объекты трехмерной сцены с максимальной детализацией нерационально, так как человеческое зрение не воспринимает слишком маленькие детали удаленных объектов. А, значит. Не имеет смысла тратить ресурсы на их расчет.Таким образом, на каждый момент времени необходимо просчитать для всех объектов необходимую степень детализации – *LOD уровень.*

Существуют два подхода – статический и динамический LOD. В первом случае заранее создаются упрощенные варианты максимально детализированного объекта. В ходе построения сцены просчитывается удаление объекта от наблюдателя и выбирается соответствующий вариант.

В динамическом LOD используют различные алгоритмы, позволяющие плавно регулировать число полигонов в объекте в зависимости от расстояния до наблюдателя.



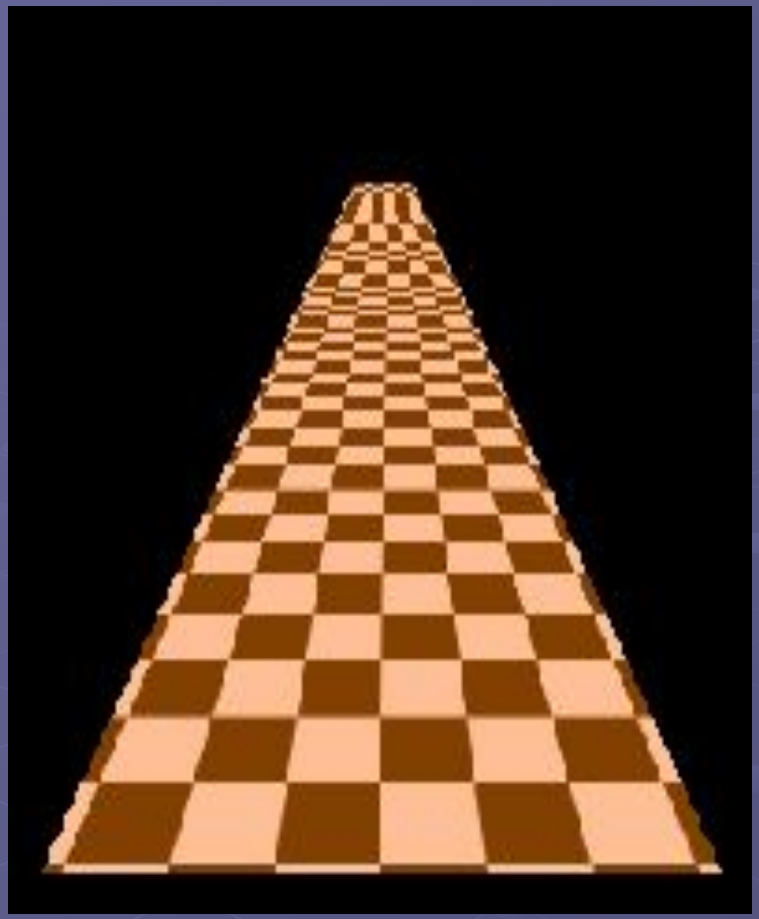
Уровень детализации не обязательно относится только к геометрии, метод может применяться и для экономии других ресурсов - при выборе техник освещения, техник текстурирования.

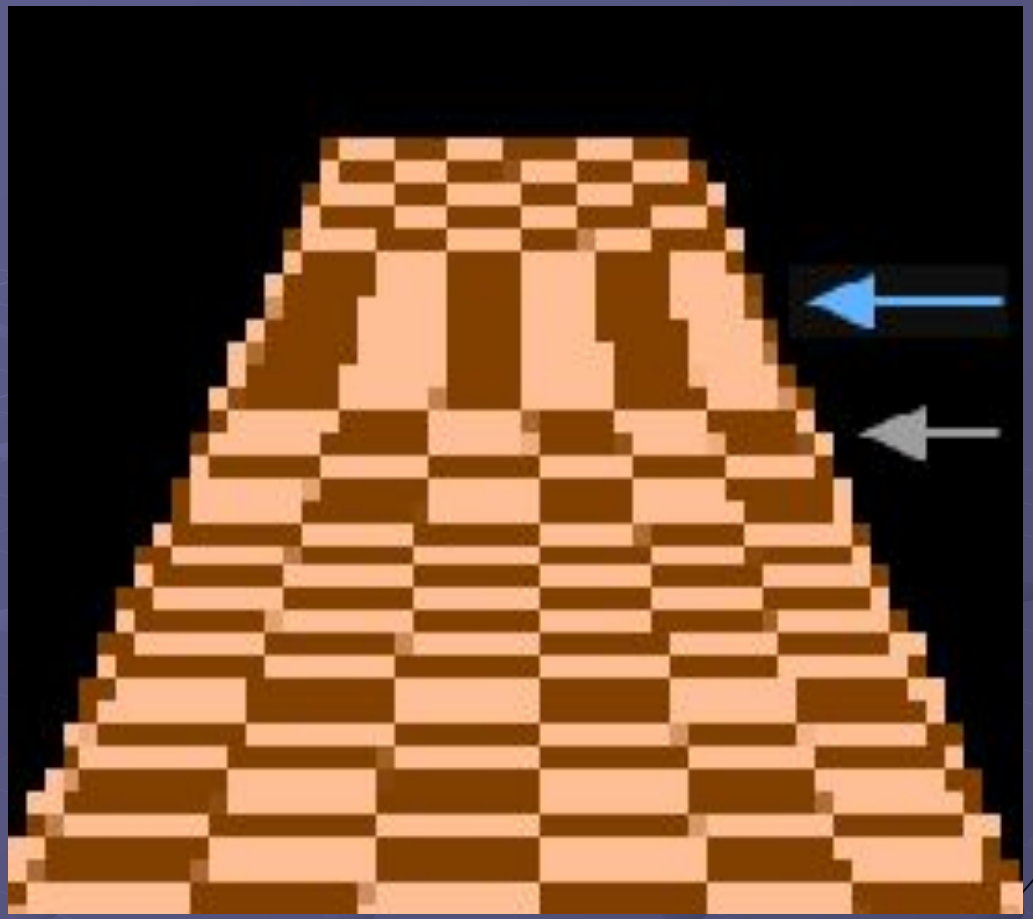
Кроме расстояния от камеры, для LOD могут иметь значение и другие факторы - общее количество объектов на экране (когда один-два персонажа в кадре, то используются сложные модели, а когда 10-20, они переключаются на более простые) или количество кадров в секунду (задаются пределы значений FPS, при которых изменяется уровень детализации, например, при FPS ниже 30 снижаем сложность моделей на экране, а при 60, наоборот, повышаем). Другие возможные факторы, влияющие на уровень детализации - скорость перемещения объекта (ракету в движении вы рассмотреть вряд ли успеете, а вот улитку - запросто), важность персонажа с игровой точки зрения.

У каждого метода свои преимущества и недостатки. Статическое управление иногда вызывает эффект дерганья при смене детализации. Динамическое потребляет большее количество ресурсов, может наблюдаться эффект волнистости поверхности

Билинейная фильтрация

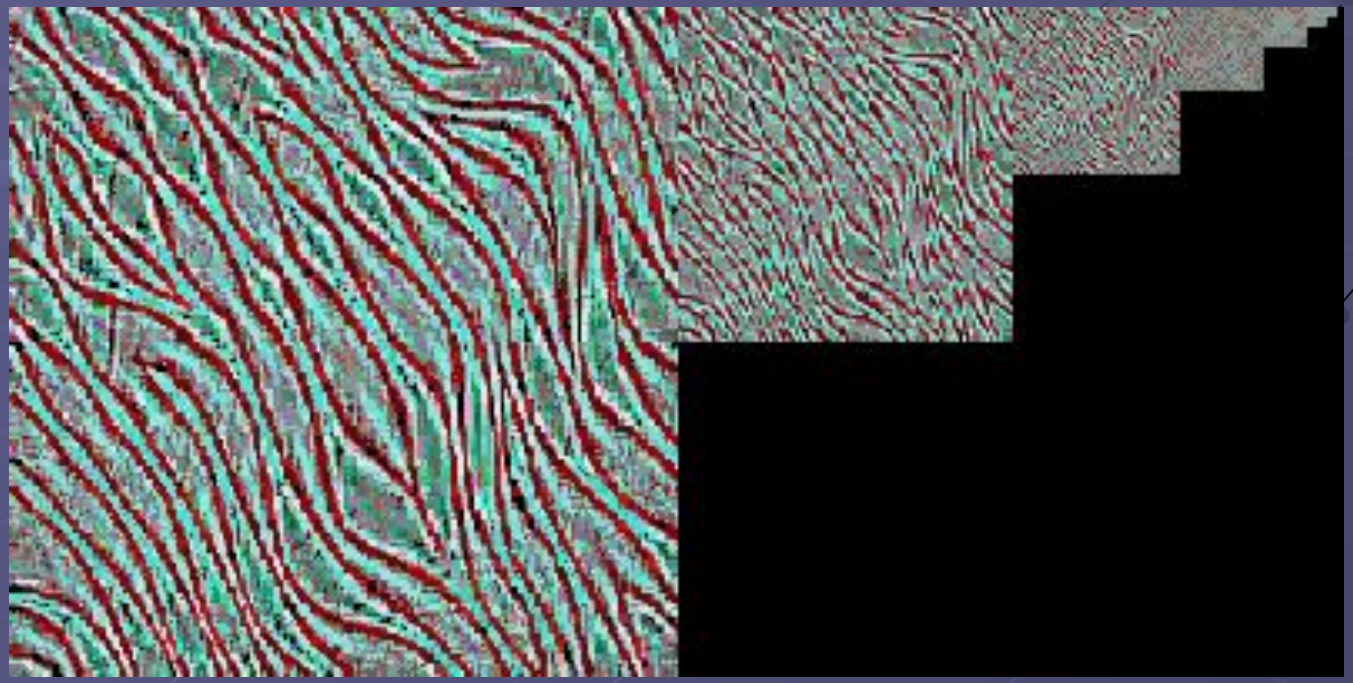
Билинейная фильтрация это техника устранения искажений изображения (фильтрация), таких, как "блочности" текстур при их увеличении. При медленном вращении или движении объекта (приближение/удаление) могут быть заметны "перескакивания" пикселов с одного места на другое, т.е. появляется блочность. Во избежании этого эффекта применяют билинейную фильтрацию, при использовании которой для определения цвета каждого пикселя берется взвешенное среднее значение цвета четырех смежных текселей и в результате определяется цвет накладываемой текстуры. Результирующий цвет пикселя определяется после осуществления трех операций смешивания: сначала смешиваются цвета двух пар текселей, а потом смешиваются два полученных цвета. Впрочем, существует целый класс артефактов визуализации, появляющихся в результате наложения текстур и известный под названием "depth aliasing" (депт-алиасинг, ошибки определения глубины сцены, другое название Z-aliasing), от которых билинейная фильтрация не избавляет и не может избавить. Ошибки "depth aliasing" возникаю в результате того факта, что объекты более отдаленные от точки наблюдения, выглядят более маленькими на экране. Если объект двигается и удаляется от точки наблюдения, текстурное изображение, наложенное на уменьшившийся в размерах объект становится все более и более сжатым. В конечном счете, текстурное изображение, наложенное на объект, становится настолько сжатым, что появляются ошибки визуализации..



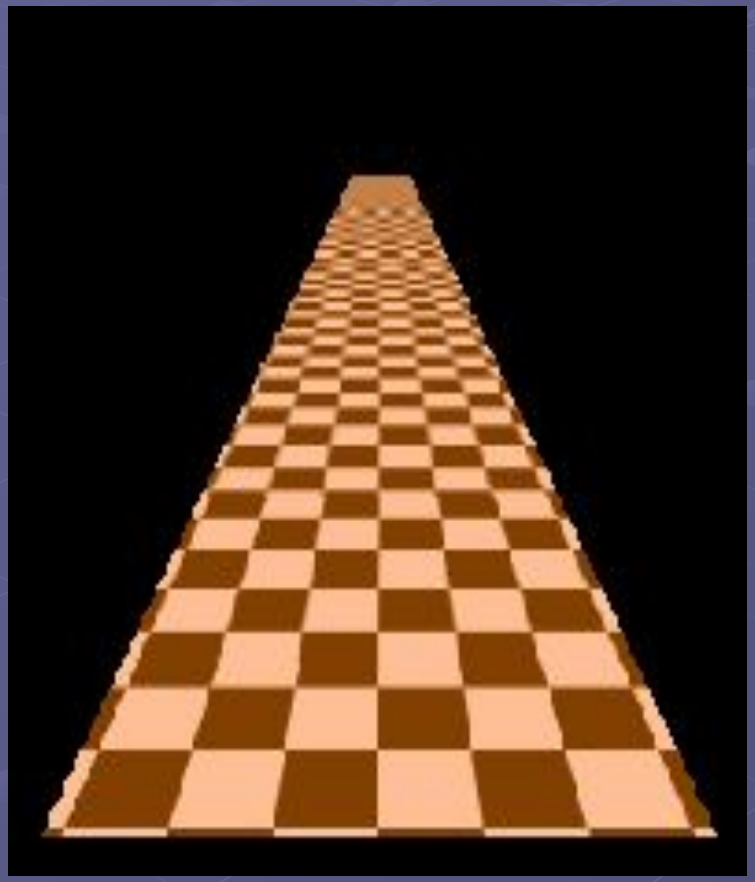


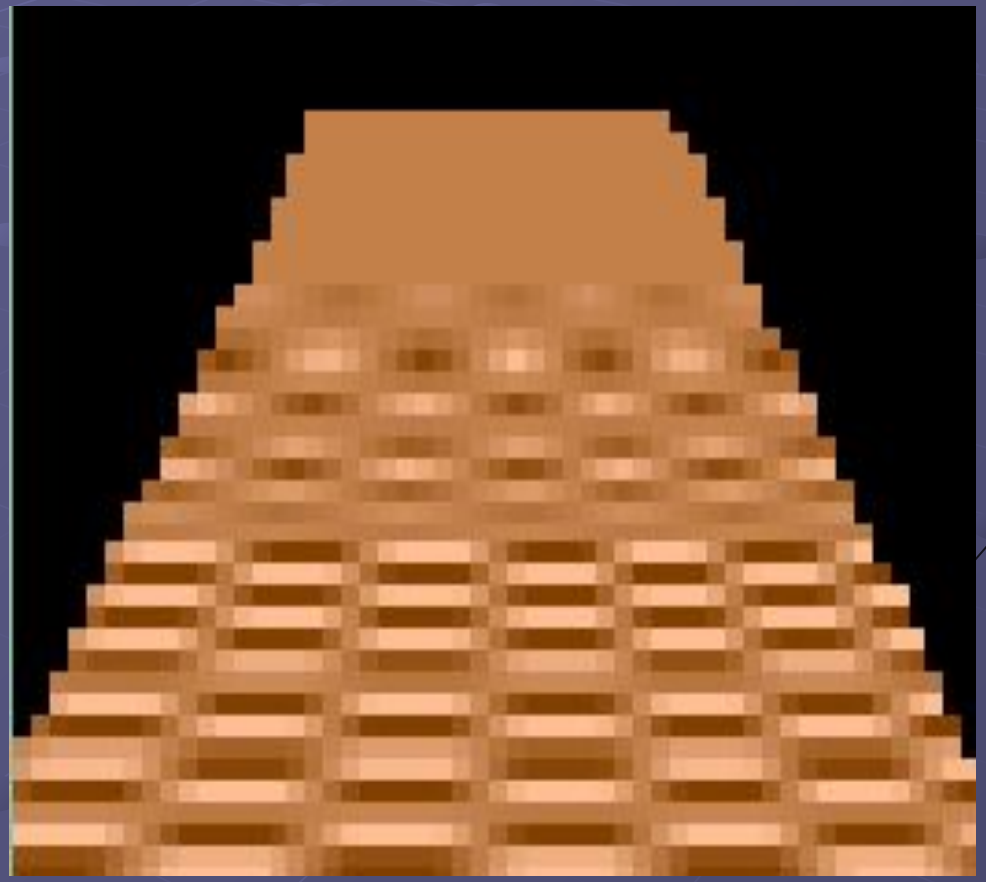
Mip-mapping

Если говорить кратко, то mip-mapping - наложение текстур, имеющих разную степень или уровень детализации, когда в зависимости от расстояния до точки наблюдения выбирается текстура с необходимой детализацией.
Mip-текстура (mip-map) состоит из набора заранее отфильтрованных и масштабированных изображений. В изображении, связанном с уровнем mip-map, пиксель представляется в виде среднего четырех пикселей из предыдущего уровня с более высоким разрешением. Отсюда, изображение связанное с каждым уровнем mip-текстуры в четыре раза меньше по размеру предыдущего mip-map уровня.
Степень или уровень детализации - Level of Detail или просто LOD, используются для определения, какой mip-map уровень (или какую степень детализации) следует выбрать для наложения текстуры на объект.

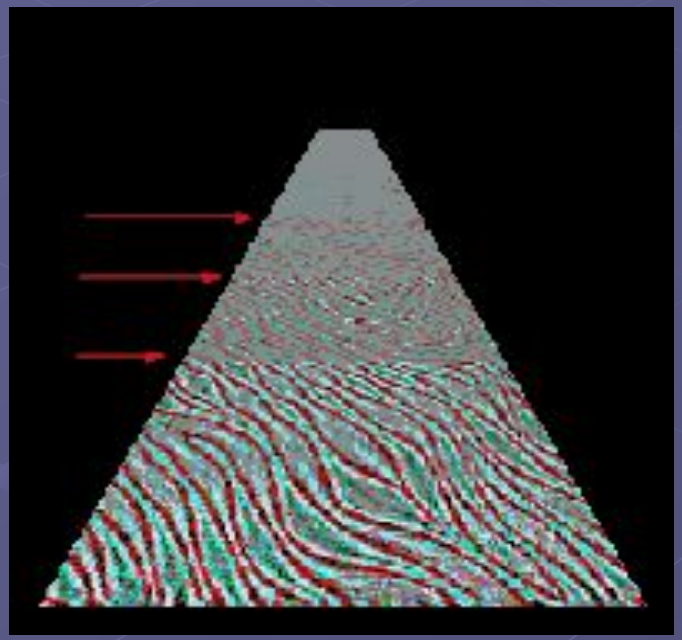


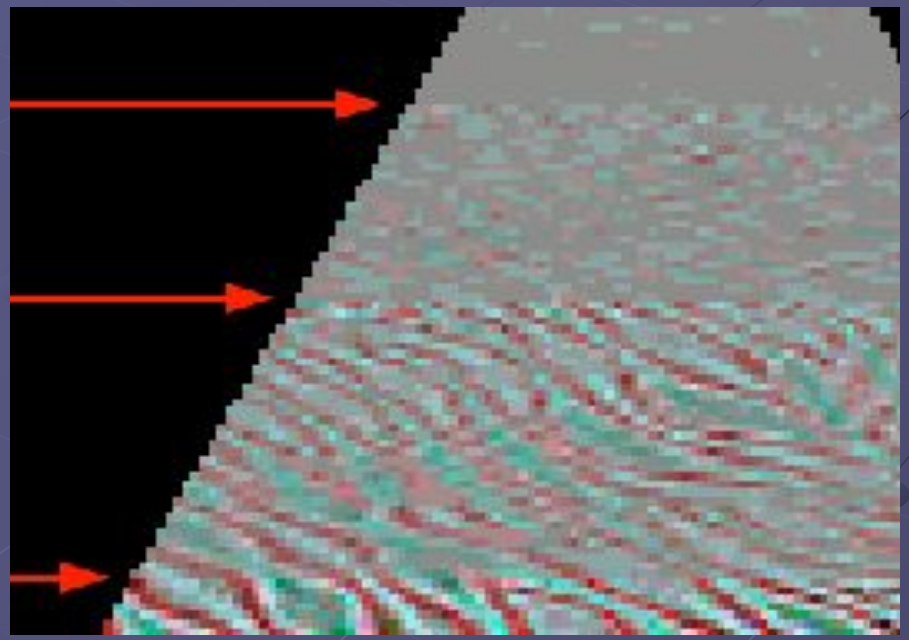
Ниже представлен прямоугольник, использовавшийся в первом примере, только теперь наложение текстур произведено с использованием техники mip-mapping. Обратите внимание, что ошибки визуализации "depth aliasing" исчезли и на расстоянии стороны прямоугольника выглядят размытыми, как и должно быть.





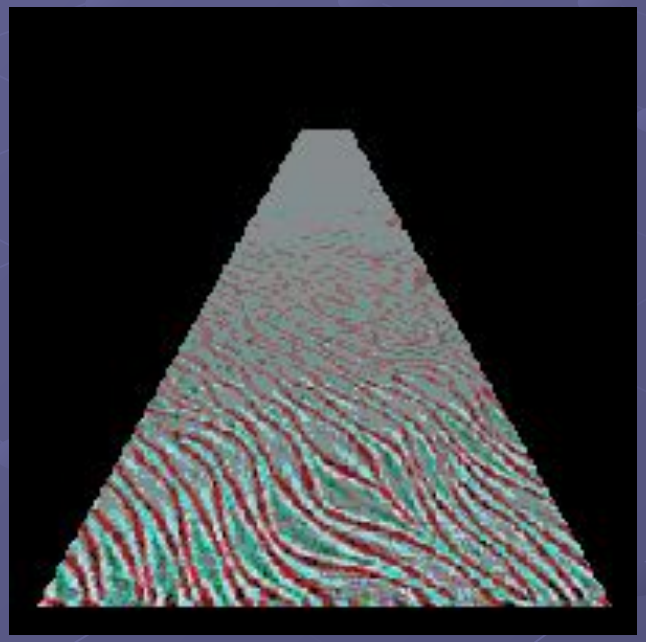
В то время, как mip-текстурирование решает проблему ошибок "depth-aliasing", его использование может стать причиной появления других артефактов. Для борьбы с этими артефактами используются различные техники фильтрации. При удалении объекта все дальше от точки наблюдения, происходит переход от низкого mip-map уровня (соответствующего изображению с высокой детализацией) к высокому mip-map уровню (соответствующего изображению с высокой степенью фильтрации и низкой детализацией). В момент нахождения объекта в переходном состоянии от одного mip-map уровня к другому, появляется особый тип ошибок визуализации, известных под названием "mip-banding" (мип-бендинг) -- полосатость или слоеность, т.е. явно различимые границы перехода от одного mip-map уровня к другому.

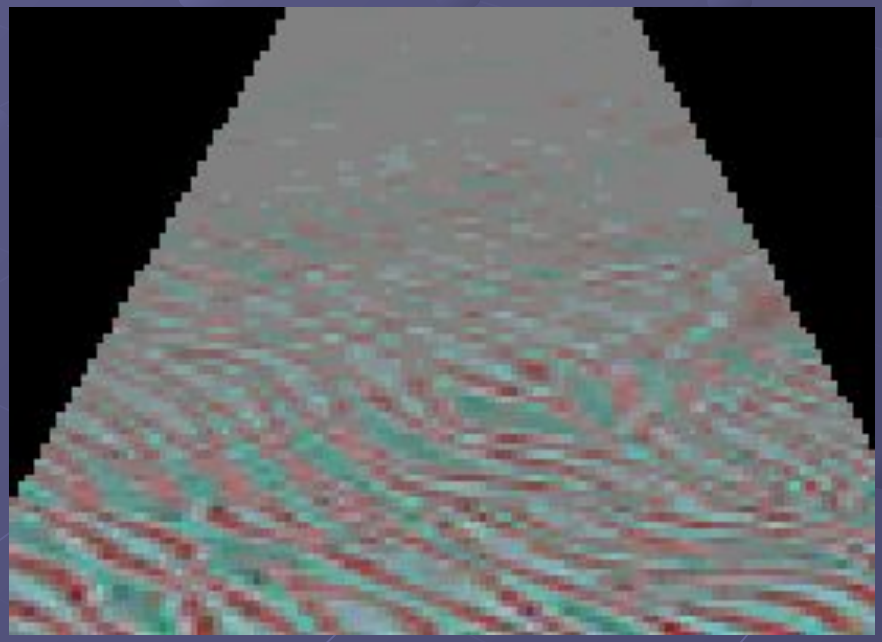




Трилинейная фильтрация

Трилинейная фильтрация (trilinear filtering) представляет собой технику, которая удаляет артефакты "mip-banding", возникающие при использовании mip-текстурирования. При трилинейной фильтрации для определения цвета пикселя берется среднее значение цвета восьми текселей, по четыре из двух соседних текстур и в результате семи операций смешивания определяется цвет пикселя. При использовании трилинейной фильтрации возможен вывод на экран текстурированного объекта с плавно выполненными переходами от одного mip уровня к следующему, что достигается за счет определения LOD путем интерполяции двух соседних mip-map уровней. Таким образом решая большинство проблем, связанных с mip-текстурированием и ошибками из-за неправильного расчета глубины сцены

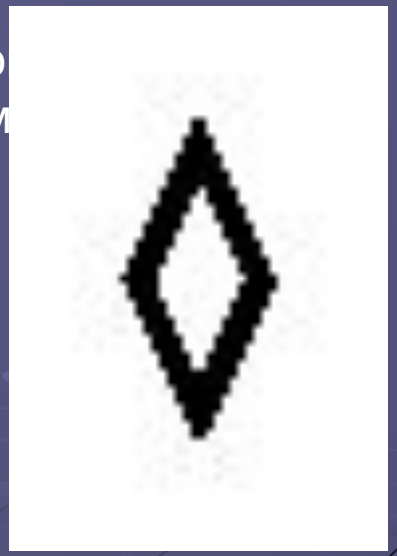




Environment Map-Bump Mapping

Технология, являющаяся дальнейшим развитием Bump Mapping. В этом случае, помимо базовой текстуры объекта, применяется еще две текстуры:
1. Текстура, являющаяся отрендеренным вариантом трехмерной сцены вокруг объекта (environment map).
2. Текстура - карта рельефа (bump map).
Самостоятельно и совместно с Procedural Texturing данная технология позволяет получить такие натуральные эффекты, как отражение, отражение в кривом зеркале, дрожание поверхностей, искажение изображения, вызываемое водой и теплым воздухом, трансформация искажений по шумовым алгоритмам, имитация туч на небе и др.

Antialiasing (anti-aliasing)



Удаление ступенчатости отображения линий и контуров, вызванно недостаточной разрешающей способностью экрана, многократным увеличением растрового изображения или ошибками при расчете 3D сцен.

Суть антиалиасинга в создании плавного цветового перехода на границах объекта, что и создает эффект сглаженности линий.

Сегодня существуют два метода реализации антиалиасинга – суперэмплинг(Supersampling) и мультисэмплинг(Multisampling).

Суперсэмплинг стал самой первой и простой, а в то же время и самой ресурсоемкой реализацией антиалиасинга. Каждый кадр просчитывается при разрешении, в несколько раз превышающим номинальное. Затем полученное изображение сжимается до номинального разрешения.

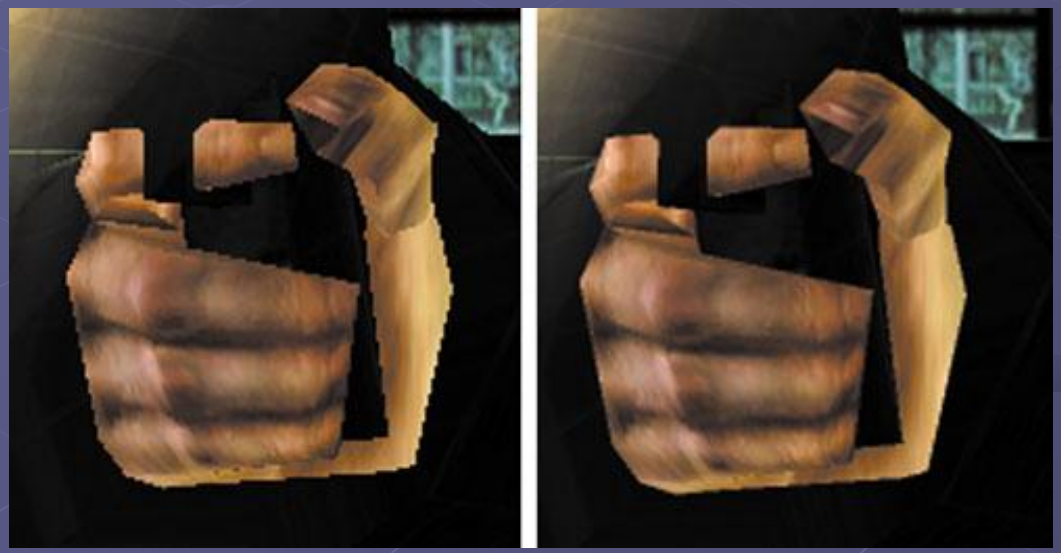
Суть мультисэмплинга заключается в обработке только границ объектов, что намного уменьшает объем нагрузки GPU и видеопамяти. Эта интеллектуальная технология антиалиасинга получила повсеместное распространение и используется всеми современными 3D-ускорителями.

Антиалиасинг, также как и анизотропия, бывает нескольких уровне – 2X, 4X, 6X, и чем выше уровень FSAA(Full Screen Antialiasing – полноэкранное сглаживание – другое название антиалиасинга), тем выше получается качество результирующего изображения соответственно.

Анизотропная фильтрация (Anisotropic Filtering, AF)

Билинейная и трилинейная фильтрация не совсем корректно рассчитывает цвет текселя - а точнее рассчитывает его неверно (отталкиваясь от законов нашего зрения) для наклонных плоскостей. Использование анизотропной фильтрации дополняет текущие режимы фильтрации, позволяя регулировать угол анизотропии. Чем больше угол - тем выше качество и реалистичность самой фильтрации, тем выше, в то же время, затраты на обсчет.

Уровень анизотропной фильтрации определяется числом текселей, которые обрабатываются при вычислении конечного пикселя. Современные графические решения позволяют выставить уровень фильтрации в драйвере. Самыми распространёнными уровнями фильтрации являются 2x (16 текселей), 4x (32 текселя), 8x (64 текселя) и 16x (128 текселей). Очевидно, что при повышении уровня анизотропной фильтрации нагрузка на полосу пропускания памяти также увеличивается, а это неминуемо сказывается на производительности



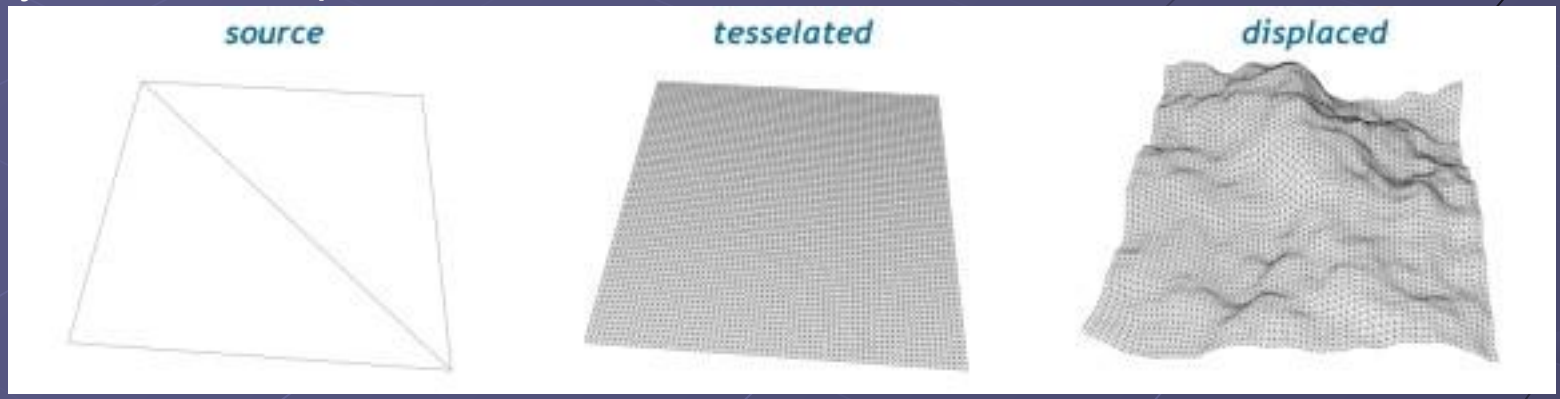
Procedural Textures (Процедурные Текстуры)

Процедурные текстуры - это текстуры, описываемые математическими формулами. Такие текстуры не занимают в видеопамяти места, они создаются пиксельным шейдером "на лету", каждый их элемент (тексель) получается в результате исполнения соответствующих команд шейдера. Наиболее часто встречающиеся процедурные текстуры: разные виды шума (например, fractal noise), дерево, вода, лава, дым, мрамор, огонь и т.п., то есть те, которые сравнительно просто можно описать математически. Процедурные текстуры также позволяют использовать анимированные текстуры при помощи всего лишь небольшой модификации математических формул. Например, облака, сделанные подобным образом, выглядят вполне прилично и в динамике и в статике

Преимущества процедурных текстур также включают в себя неограниченный уровень детализации каждой текстуры, пикселизации просто не будет, текстура как бы всегда генерируется под необходимый для ее отображения размер. Большой интерес представляет и анимированный Normal Mapping, с его помощью можно сделать волны на воде, без применения предпросчитанных анимированных текстур.

Displacement Mapping

Наложение карт смещения (Displacement Mapping) является методом добавления деталей к трехмерным объектам. 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. Наглядно создание ландшафта при помощи наложения карт смещения представлено на картинке. Исходными были 4 вершины и 2 полигона, в итоге получился полноценный кусок ландшафта.



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.

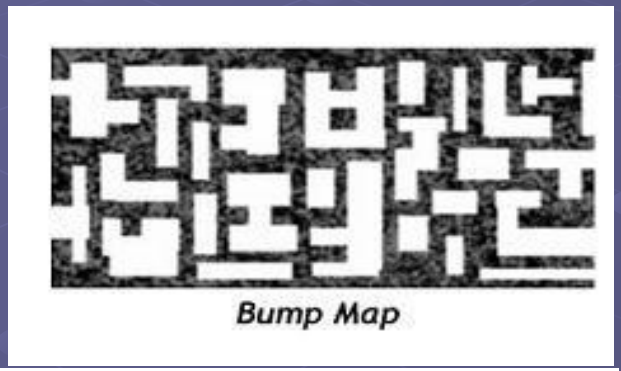
Другое преимущество в том, что применение карт смещения превращает сложные полигональные трехмерные сетки в несколько двухмерных текстур, которые проще поддаются обработке. Например, для организации Level of Detail можно использовать наложение карт смещения.

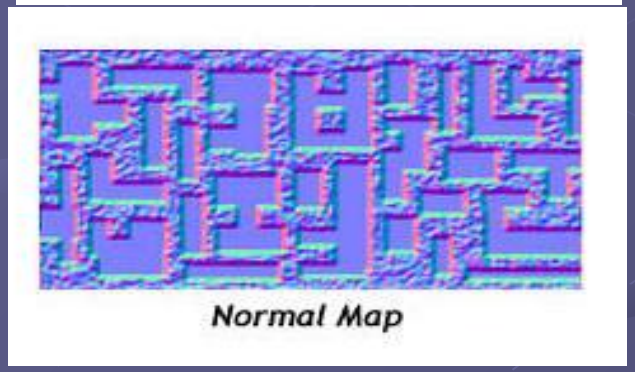
Но у карт смещения есть и некоторые ограничения. Например, гладкие объекты, не содержащие большого количества тонких деталей, будут лучше представлены в виде стандартных полигональных сеток или иных поверхностей более высокого уровня, вроде кривых Безье. С другой стороны, очень сложные модели, такие как деревья или растения, также нелегко представить картами смещения.

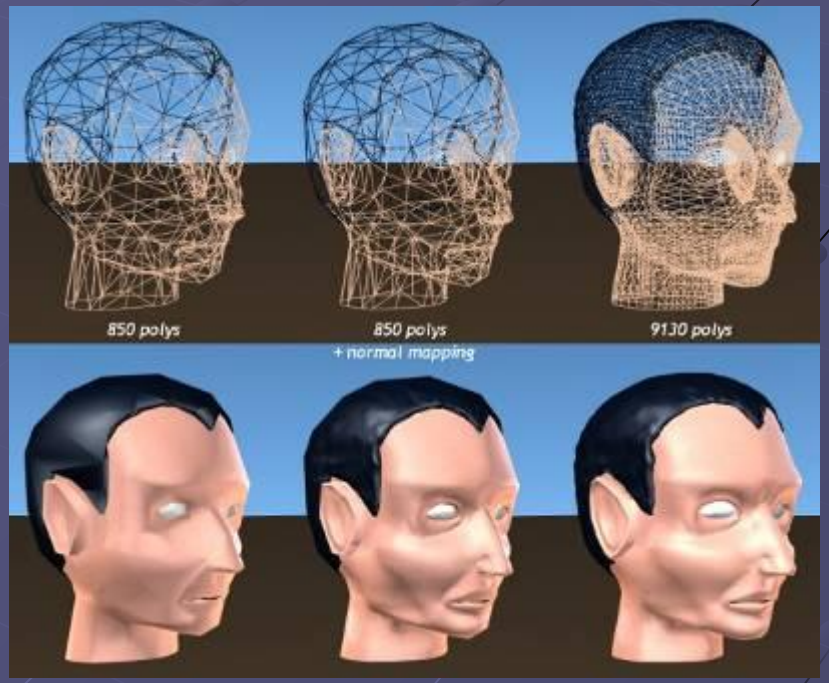
Normal Mapping

Нормалмаппинг - это улучшенная разновидность техники бампмаппинга, описанной ранее, расширенная ее версия.

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.





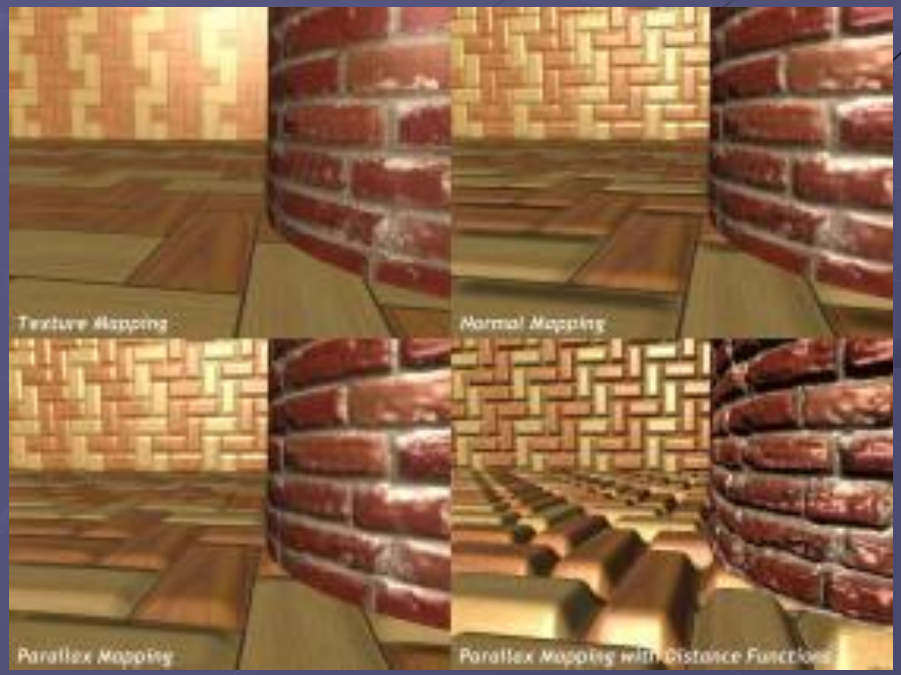


До нормалмаппинга были разработаны несколько таких методов, но низкополигональные модели даже с бампмаппингом получаются заметно хуже более сложных моделей. Нормалмаппинг хоть и имеет несколько недостатков (самый явный - так как модель остается низкополигональной, это легко видно по ее угловатым границам), 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.

Parallax Mapping/Offset Mapping

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.

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. Модифицированные методики получили несколько разных названий. 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.

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.

Postprocessing (Постобработка)

В широком смысле, постобработка - это все то, что происходит после основных действий по построению изображения. Иначе говоря, постобработка - это любые изменения изображения после его рендеринга. Постобработка представляет собой набор средств для создания специальных визуальных эффектов, и их создание производится уже после того, как основная работа по визуализации сцены выполнена, то есть, при создании эффектов постобработки используется готовое растровое изображение. Из эффектов постобработки в играх чаще всего используют Bloom, Motion Blur, Depth Of Field. Существует и множество других постэффектов: noise, flare, distortion, sepia и др

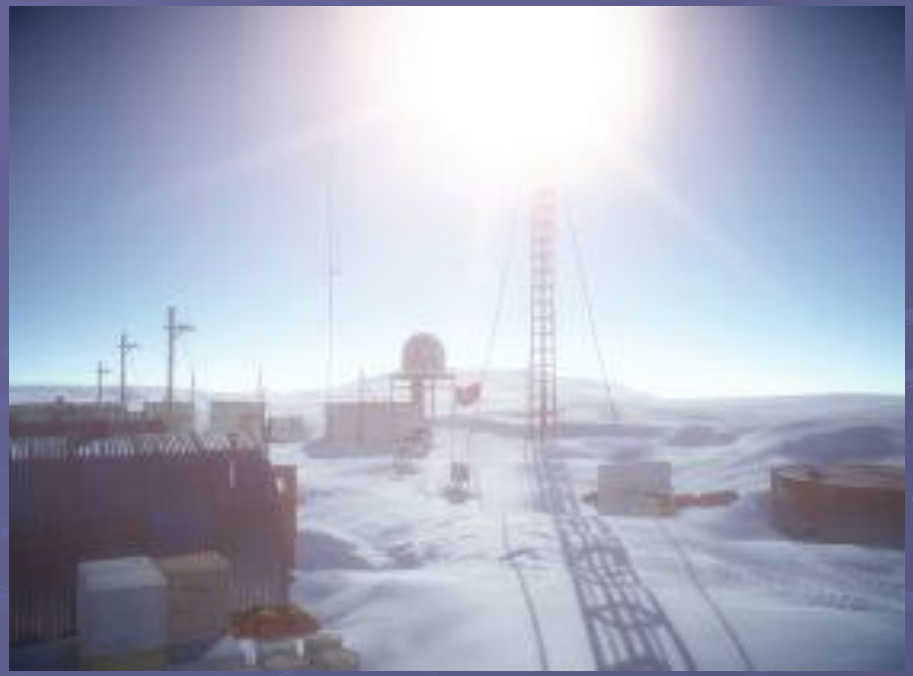
High Dynamic Range (HDR)

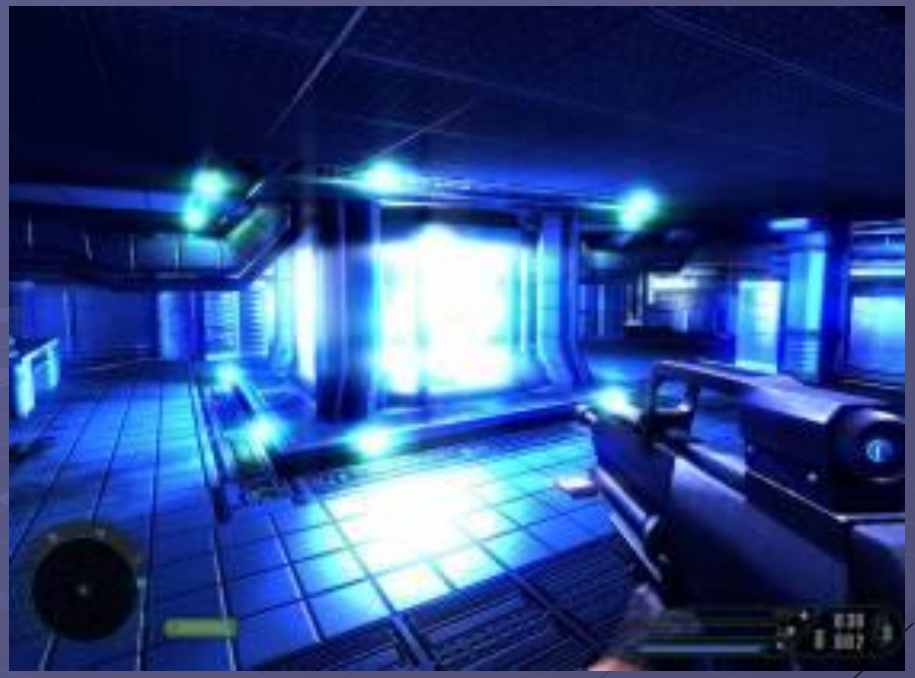
High Dynamic Range (HDR) в применении к 3D графике - это рендеринг в широком динамическом диапазоне. Суть HDR заключается в описании интенсивности и цвета реальными физическими величинами.

Так, динамический диапазон модели RGB составляет 256:1 или 100:1 cd/m2 (два порядка). Эта модель описания цвета и интенсивности общепринято называется Low Dynamic Range (LDR). Итак, динамического диапазона модели описания RGB недостаточно для представления изображений, которые человек способен видеть в реальности, эта модель значительно уменьшает возможные значения интенсивности света в верхней и нижней части диапазона.

HDR рендеринг позволяет изменять экспозицию уже после того, как мы отрендерили изображение. Дает возможность имитировать эффект адаптации человеческого зрения (перемещение из ярких открытых пространств в темные помещения и наоборот), позволяет выполнять физически правильное освещение, а также является унифицированным решением для применения эффектов постобработки (glare, flares, bloom, motion blur). Алгоритмы обработки изображения, цветокоррекцию, гамма-коррекцию, motion blur, bloom и другие методы постобработки качественней выполнять в HDR представлении.

В приложениях 3D рендеринга реального времени (играх, в основном) HDR рендеринг начали использовать не так давно, ведь это требует вычислений и поддержки render target в форматах с плавающей точкой, которые впервые стали доступны только на видеочипах с поддержкой DirectX 9.





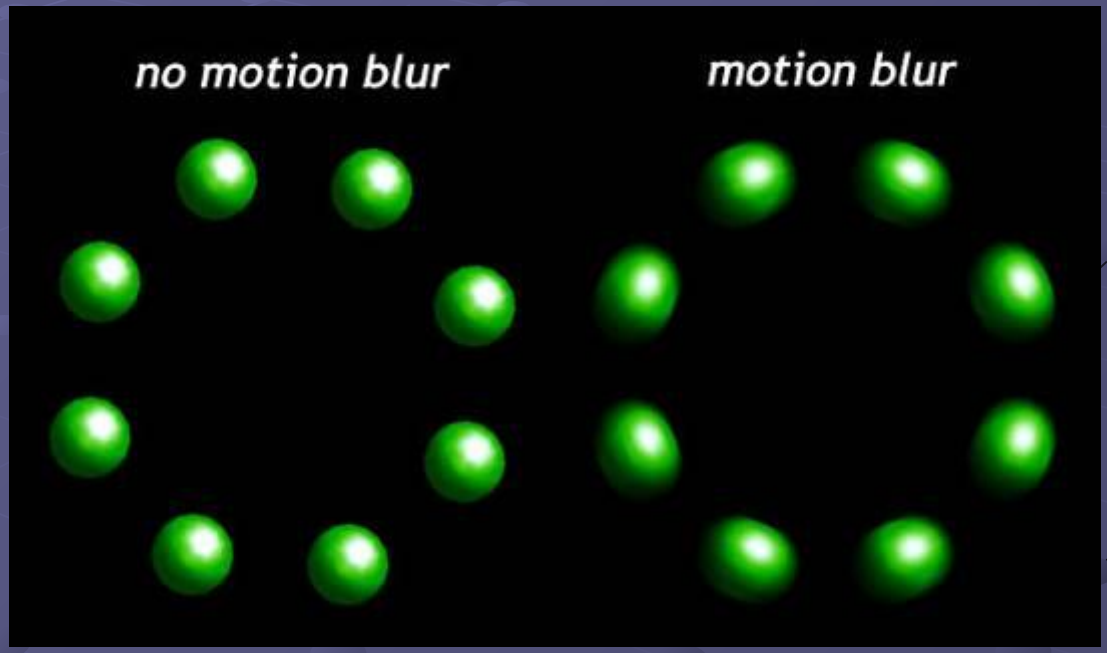

100
6


ATi

Motion Blur

Смазывание в движении (motion blur) происходит при фото- и киносъемке из-за движения объектов в кадре в течение времени экспозиции кадра, в то время, когда затвор объектива открыт

По изображению без смазывания нельзя даже сказать, движутся сферы или нет, в то время как motion blur дает четкое представление о скорости и направлении движения объектов. Отсутствие смазывания при движении служит и причиной того, почему движение в играх при 25-30 кадрах в секунду кажется дерганым, хотя кино и видео при этих же параметрах частоты кадров смотрится прекрасно.



Для компенсации отсутствия смазывания в движении желательна или высокая частота кадров (60 кадров в секунду или выше) или использование методов дополнительной обработки изображения, для эмуляции эффекта motion blur. Это применяется и для улучшения плавности анимации и для эффекта фото- и кинореалистичности одновременно.

Возможные применения эффекта motion blur в играх: все гоночные игры (для создания эффекта очень высокой скорости движения и для применения при просмотре ТВ-образных повторов), спортивные игры (те же повторы, а в самой игре смазывание можно применять для очень быстро движущихся объектов, вроде мяча или шайбы), файтинги (быстрые движения холодного оружия, рук и ног), многие другие игры.

Bloom

Bloom - это один из кинематографических эффектов постобработки, при помощи которого наиболее яркие участки изображения делаются еще более яркими. Это эффект очень яркого света, проявляющийся в виде свечения вокруг ярких поверхностей, после применения bloom фильтра такие поверхности не просто получают дополнительную яркость, свет от них (ореол) частично воздействует и на более темные области, соседствующие с яркими поверхностями в кадре.
В 3D графике Bloom фильтр делается при помощи дополнительной постобработки - смешивания смазанного фильтром blur кадра (всего кадра или отдельных ярких его областей, фильтр обычно применяется несколько раз) и исходного кадра. Один из наиболее часто применяемых в играх и других приложениях реального времени алгоритм постфильтра bloom

Depth Of Field (DOF)

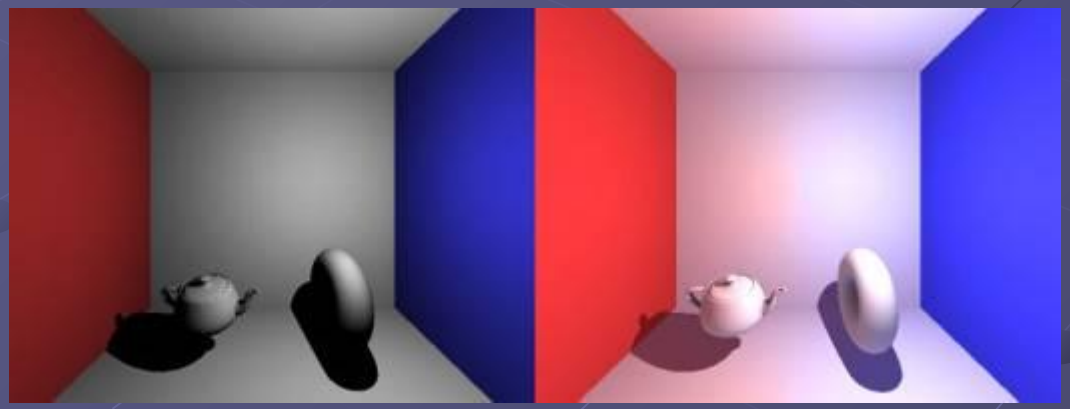
Depth of field (глубина резкости), вкратце, это размывание объектов в зависимости от их положения относительно фокуса камеры. В реальной жизни, на фотографиях и в кино мы видим одинаково четко не все объекты, это связано с особенностью строения глаза и устройства оптики фото- и киноаппаратов. У фото- и кинооптики есть определенное расстояние, объекты, расположенные на таком расстоянии от камеры находятся в фокусе и выглядят на картинке резкими, а более удаленные от камеры или приближенные к ней объекты выглядят, наоборот, размытыми, резкость снижается постепенно при увеличении или снижении расстояния.

Для достижения фото- и кинореалистичности приходится применять специальные алгоритмы, чтобы сделать для компьютерной графики нечто похожее. Эти техники симулируют эффект разного фокуса для объектов, находящихся на разном расстоянии

Одним из распространенных методов при рендеринге в реальном времени является смешивание оригинального кадра и его смазанной версии (несколько проходов blur фильтра) на основе данных о глубине для пикселей изображения.

Global Illumination

Реалистичное освещение сцены смоделировать сложно, каждый луч света в реальности многократно отражается и преломляется, число этих отражений не ограничено. А в 3D рендеринге число отражений сильно зависит от расчетных возможностей, любой расчет сцены является упрощенной физической моделью, а получаемое в итоге изображение лишь приближено к реалистичности.
Алгоритмы освещения можно разделить на две модели: прямое или локальное освещение и глобальное освещение (direct или local illumination и global illumination). Локальная модель освещения использует расчет прямой освещенности, свет от источников света до первого пересечения света с непрозрачной поверхностью, взаимодействие объектов между собой не учитывается
В глобальной модели освещения, global illumination, рассчитывается освещение с учетом влияния объектов друг на друга, учитываются многократные отражения и преломления лучей света от поверхностей объектов, каустика (caustics) и подповерхностное рассеивание (subsurface scattering).



Возможные применения эффекта motion blur в играх: все гоночные игры (для создания эффекта очень высокой скорости движения и для применения при просмотре ТВ-образных повторов), спортивные игры (те же повторы, а в самой игре смазывание можно применять для очень быстро движущихся объектов, вроде мяча или шайбы), файтинги (быстрые движения холодного оружия, рук и ног), многие другие игры.