

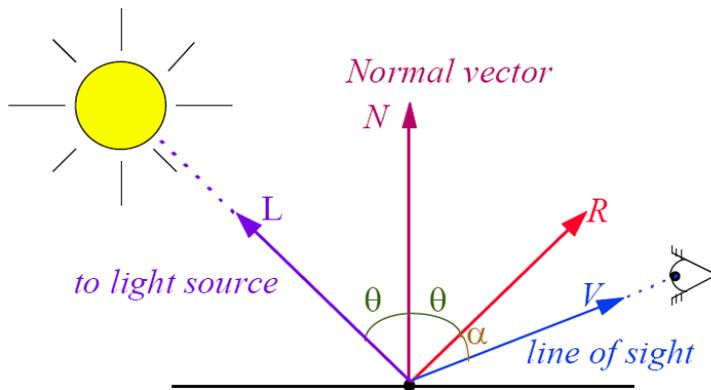
Графические ускорители

OpenGL

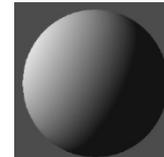
Direct3D

Графический ускоритель

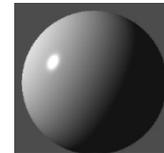
- Полигональный рендеринг примитивов (растрирование треугольников)
- Локальная модель освещенности



Ambient



Diffuse



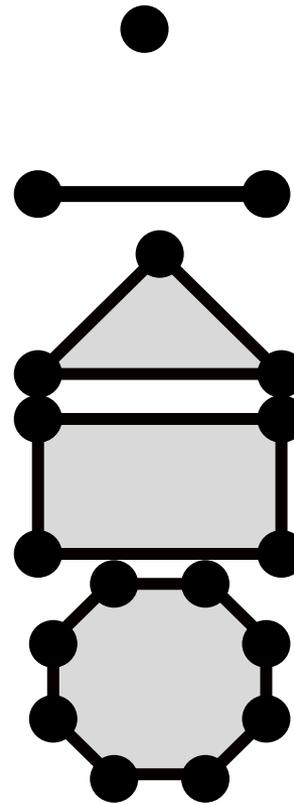
Specular

Модель освещенности

- Расчет освещенности ведется в каждой вершине независимо.
- Нет теней.
- Нет рекурсивной трассировки.
- Нет учета преломления луча на границе сред для полупрозрачных поверхностей.

Примитивы

- Точка
- Отрезок
- Треугольник
- Четырехугольник
- Многоугольник



Современные GPU рисуют 1-10 млн. примитивов за кадр

Свойства вершин

- Координаты (x, y, z)
- Цвет (R, G, B, A), материал
- Нормаль (n_x, n_y, n_z)
- Текстурные координаты (u, v)
- ...

Параметры вершин в OpenGL

	Имя регистра	Параметры	Значение по умолчанию	Пояснение
	LIGHT_ON	bool	false	Разрешен (true) или запрещен расчет освещенности
	Сторона грани	face	GL_FRONT GL_BACK	
1.	Нормаль	(x, y, z)	(0, 0, 1)	
2.	RGBA цвет	(R, G, B, A)	(1, 1, 1, 1)	Цвет вершины при отсутствии расчетов освещенности
3.	Координаты	(x, y, z, w)	(0, 0, 0, 1)	
4.	Текстурные координаты	(s, t, q, r)	???	

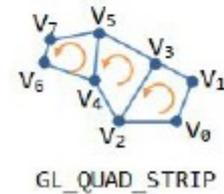
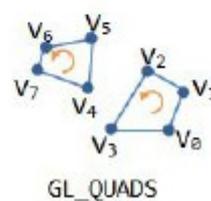
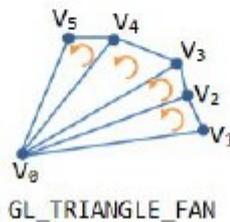
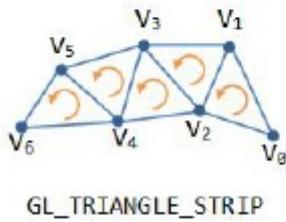
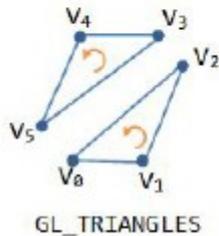
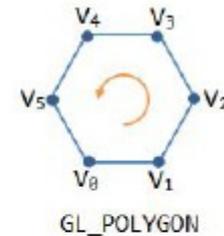
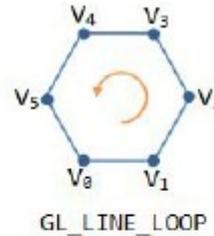
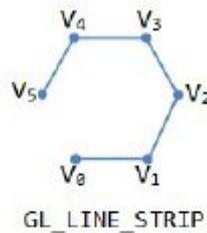
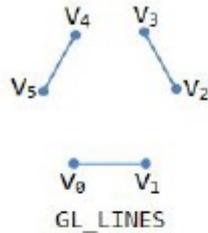
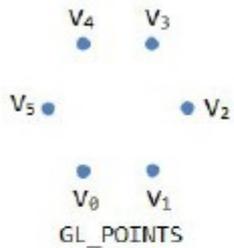
Параметры вершин в OpenGL

	Имя регистра	Параметры	Значение по умолчанию	Пояснение
5.	GL_AMBIENT	(R, G, B, A)	(0.2, 0.2, 0.2, 1.0)	Реакция на рассеянный цвет, если LIGHT_ON
6.	GL_DIFFUSE	(R, G, B, A)	(0.8, 0.8, 0.8, 1.0)	Цвет диффузного отражения, если LIGHT_ON
7.	GL_SPECULAR	(R, G, B, A)	(0, 0, 0, 1)	Цвет зеркального отражения, если LIGHT_ON
8.	GL_SHININESS	nPower	0	Степень зеркальности по Фонгу, от 0 до 128
9.	GL_EMISSION	(R, G, B, A)	(0, 0, 0, 1)	Цвет эмиссии

Задание примитивов

	DirectX	OpenGL
Point list	D3DPT_POINTLIST	GL_POINTS
Line list	D3DPT_LINELIST	GL_LINES
Line strip	D3DPT_LINESTRIP	GL_LINE_STRIP
Line loop		GL_LINE_LOOP
Polygon		GL_POLYGON
Triangle list	D3DPT_TRIANGLELIST	GL_TRIANGLES
Triangle strip	D3DPT_TRIANGLESTRIP	GL_TRIANGLE_STRIP
Triangle fan	D3DPT_TRIANGLEFAN	GL_TRIANGLE_FAN
Quad list		GL_QUADS
Quad strip		GL_QUAD_STRIP

Задание примитивов

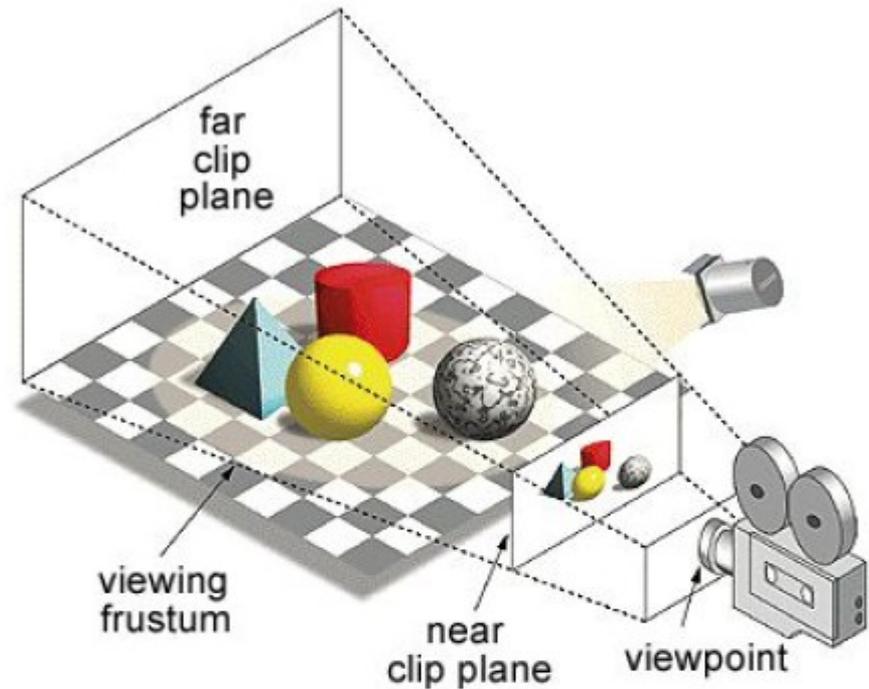


*Правило обхода
меняется !*



Сцена

- Объекты
(набор примитивов)
- Источники света
- Камера - это виртуальный объект (полный аналог видеокамеры из реального мира)



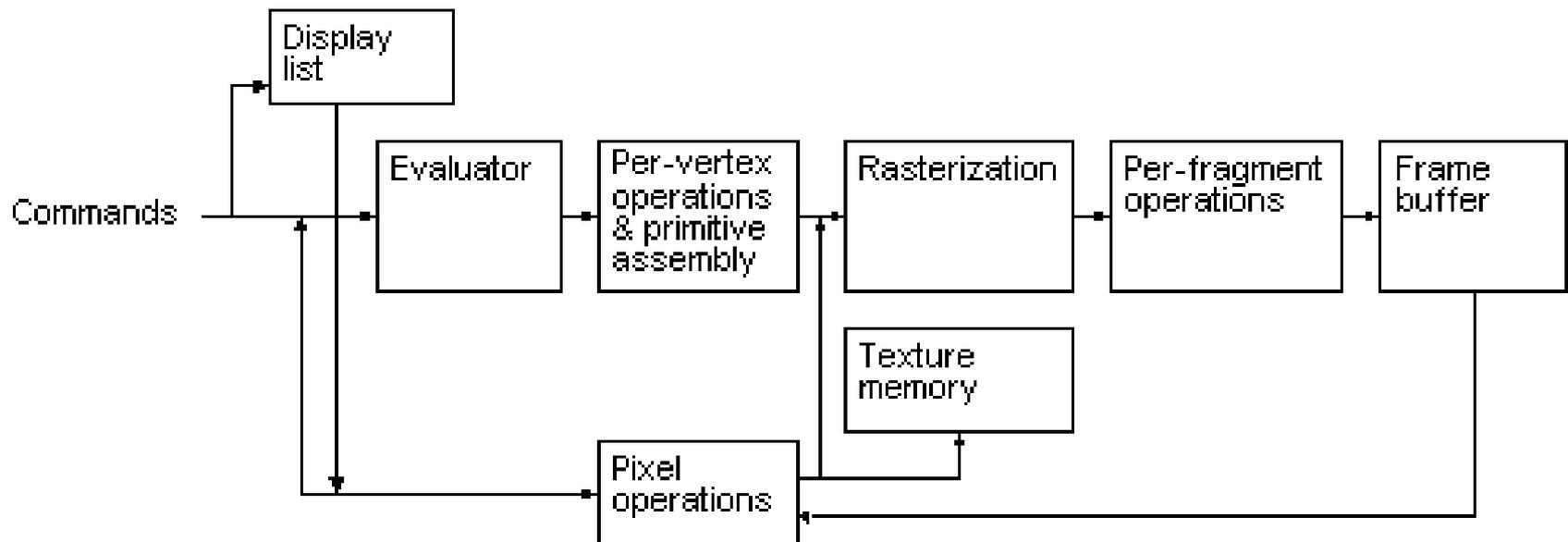
Сцена

- Мировая система координат.
- OpenGL не предоставляет средств для описания и моделирования сложных геометрических объектов.
- Источники света. Включен/выключен расчет освещенности.
- Имеется отдельный рассеянный свет.
- Туман.
- Примитивы: точки, отрезки (линии), грани. С каждой вершиной ассоциируются данные:
 - Координаты
 - Цвет, материал
 - Нормаль
 - Текстурные координаты

Сцена

- Каждая вершина вместе с ассоциированной информацией обрабатывается независимо и одинаково в порядке задания. Единственное исключение - случаи клиппирования.
- Способы отображения: точками, отрезками, сплошными гранями.
- Цвет и материал. Присваиваются только вершинам.
- Поверхности: односторонние и двусторонние.
- Закраска: гладкая (Гуро) или однородная (Flat). (A для отрезка или грани, когда все вершины имеют разные цвета, «главная» вершина).
- Дисплейные списки как пример сегментирования дисплейного файла и средства поддержки систем моделирования.
- Преобразования модельных координат.
- Камера неподвижна. Расположение камеры в сцене выполняется как подбор ракурса сцены перед камерой, т.е. через модельное преобразование.

Конвейерная обработка



Конвейер в OpenGL

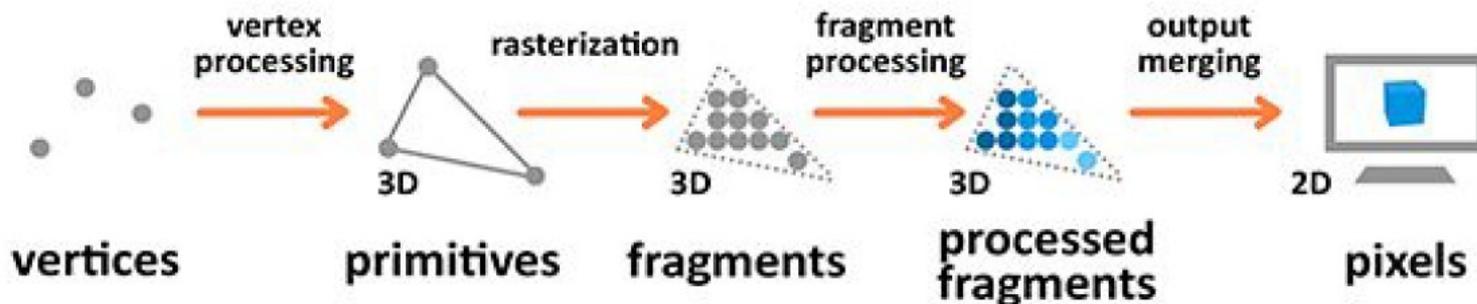
Конвейерная обработка

1. Формирование 3D-сцены
(анимация, выбор, детализация...)
2. Декомпозиция на примитивы
3. Модификация вершин
(вершинные шейдеры)
4. Обход примитивов
5. Рендеринг пикселов
(пиксельные шейдеры)
6. Формирование итогового изображения

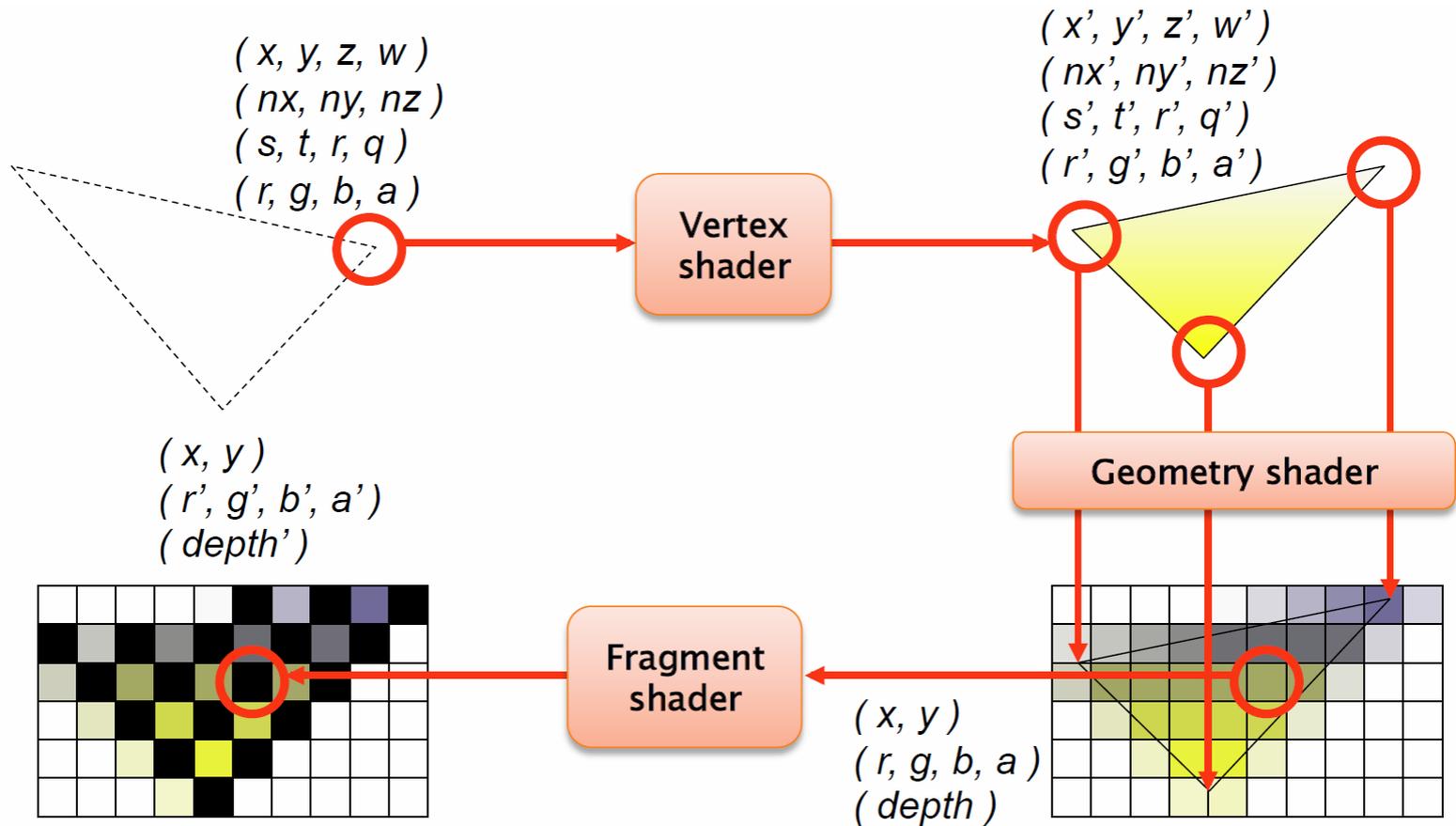
Конвейерная обработка

Обход и рендеринг примитивов:

- Умножение вершин на матрицы преобразований (включая видовое преобразование)
- Перебор всех «треугольников» по набору вершин
- Клипирование и растривание
- Текстурирование



Шейдеры



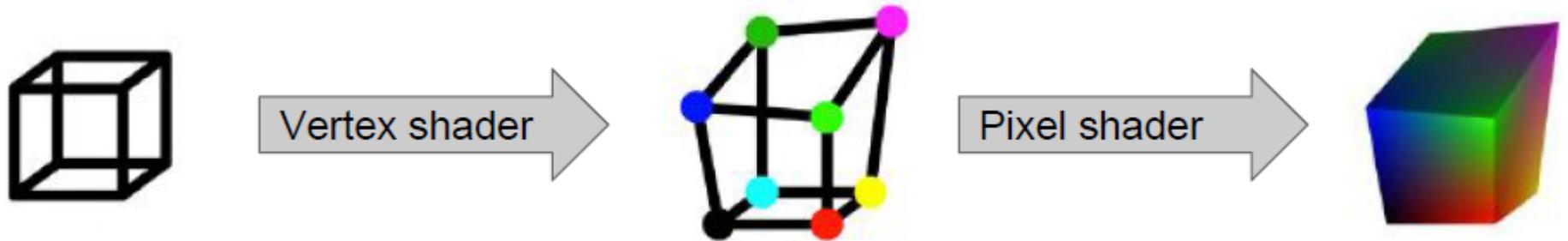
Вершинный шейдер



Небольшая программа, которая исполняется в GPU для каждой вершины:

- Изменяет свойства вершин – позиция, цвет, нормаль, текстурные координаты, ...
- Не создает новые и не удаляет существующие вершины

Пиксельный шейдер

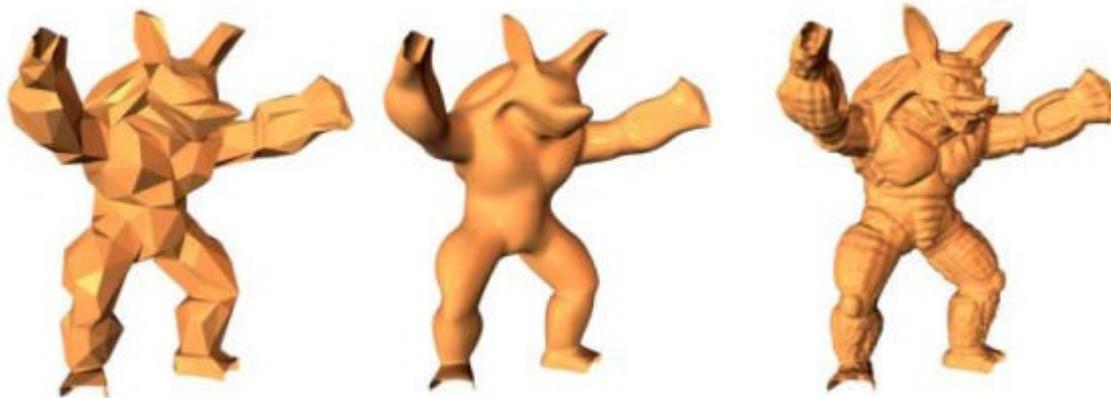


Небольшая программа, которая исполняется в GPU для каждого пикселя:

- обрабатывает данные, связанные с пикселями (например, цвет, глубина, текстурные координаты)
- мультитекстурирование

Геометрический шейдер

- Изменяет геометрию сцены (порождает новые примитивы), например N-patch

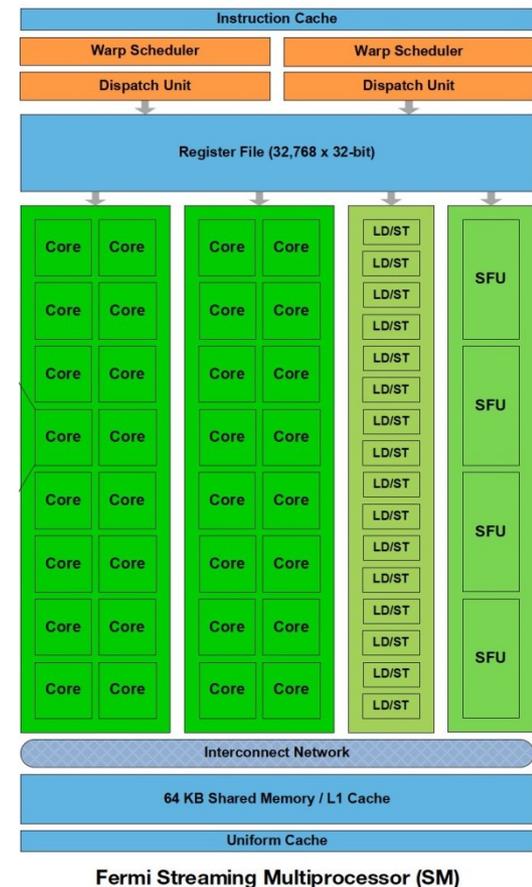


Copyright by www.Malbred.com 2005

Аппаратная реализация

- Много АЛУ
- Сотни локальных регистров
- Блочная «двумерная» память (медленная), но быстрый кэш

Идеально для параллельной обработки одинаковым кодом разных данных



Проблемы стандартных решений

Направление ветра (Davis Wireless Vantage Pro2™)

