Создание продукта высокого качества и стоимости требует [redacted]
масштабируемости и параллелизма.

На заводах: → ~~[illegible]~~ конвейер за счёт разделения на простые задачи.

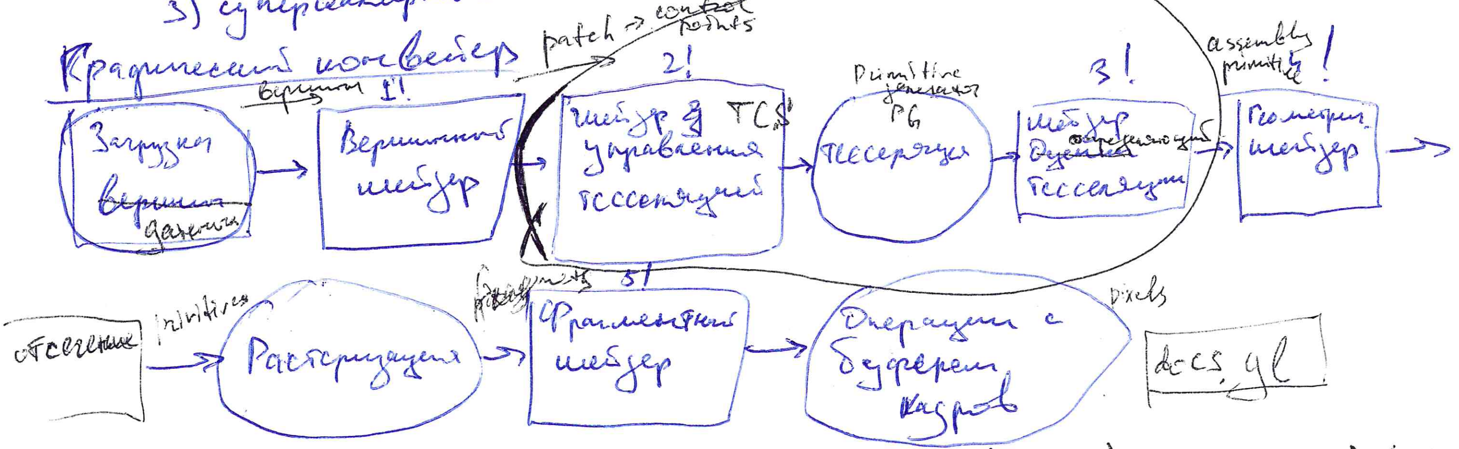
Цель OpenGL — предоставить абстрактный уровень м/у приложением
и графической подсистемой, состоящей из процессоров.

Графический процессор состоит из маленьких программируемых процессоров, наз. шейдеры
наз-ваемых шейдерные ядра, к-ые запускают мини программы,
Ядра очень ограничены по возможностям, т.е. нет:

1) предсказаний ветвления
2) выполнение вне порядка
3) суперскалярные вычисления

Barycentric coordinate

Графический конвейер



Корни OpenGL — Silicon Graphics, Inc. (SGI) and IRIS GL

GL ⇒ Graphic ~~library~~ version 1/1992

SGI → the Architectural Review Board (ARB) which include
Compaq, DEC, IBM, Intel and Microsoft.

Versions and Publication Dates

p. 7.

создан → V. 1.0		1992	+two profiles
	V. 2.0	2004	↗ core profiles (faster)
тесселяция ↓	V 3.0	2008 → fork	↘ compatibility profile
V. 4.0	V. 3.3.	March 2010	

92 year → basic filled polygons or sprite rendering capabilities.

fundamental rendering unit is primitive (points, lines, triangles)

rendering means rasterization

as primitive consists of vertexs (a simple point)

Graphics pipeline is broken down 2 major parts:

1) the front end or (primitive assembly)

2) the back end things after rasterizer depth and stencil testing трафарет
fragment shading, blending and updating the output image.

VBO Vertex Buffer Object (Объект вершинного буфера)

средство OpenGL позволяющее загружать определенные данные в память GPU.

Этапы: (процесс)

0) создание массива с данными
1) ~~до~~ создать ссылку (VBO ~~index~~ id)
2) связать с данными (массивом)
3) выделение памяти, загрузка данных в память.

примечание: выбор типа памяти, выбор типа буфера.

4) можно удалить лишнее.

1) координаты
2) цвета
3) нормали.

V.X ; V.Y. VZ
V[i].x;
V[i].y;
V[i].z

VAO Vertex Array Object. (Объект вершинного массива)

Объект который хранит данные о модели.
Таких моделей вы можете создавать целую кучу.

Хранит вызовы

VBO0	vertexposit.
VBO1	V. colours
VBO2	normal vectors V.s
	texture coordin.

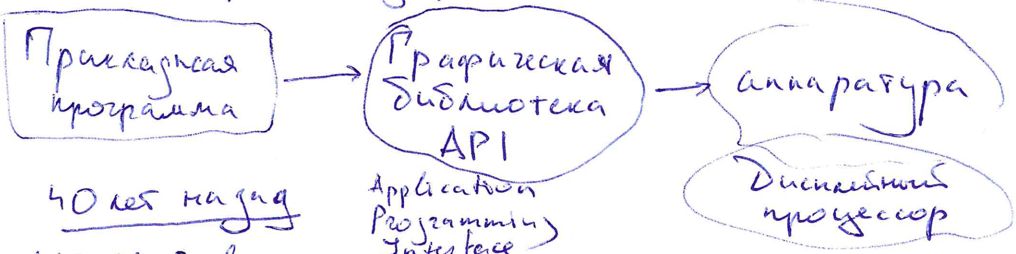
EBO · Element Buffer Object (Index или IBO)
буфер, к-ый хранит индексы (для порядка отрисовки вершин)

Эйнджел _ Интерактивная КГ

стр.44. 1-ая глава

Моделирование → $\overset{\text{закрашивание}}{\text{тонирование}}$

схема построения изображения

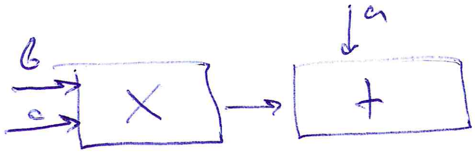


40 лет назад

идея: В связи с ростом производительности СБИС

↓

Конвейерный принцип обработки информации =>



$a+(b\times c)$

↓

Для геометрических данных ~ 10^6 вершин! ⇒ За счет чего достигается ускорение

Обработка **Вершины**

1. $\overset{\text{Модельное}}{\text{Геометрическое}}$ преобразование
2. Отсечение видовое преобразование
3. Проективное преобразовани (потеря данных!!!)
4. Растровое преобразование

Пиксели

локальной с.к. 1 OCS

мировой 1 WCS

2 VCS

3,4 CCS

Матрица комплексного преобразования

с.к. камеры

с.к. экрана

2. Отсечение [Clipping]

~~параллельное~~) прямоугольное

~~перспективное~~) угол зрения

4. Растровое преобразование

Преобразование описания 2-х мерных объектов в коды засветки пикселей в буфере кадра.

Этап выполняется аппаратно!

Представление ~~графической~~ геометрической инф-ии

Базовый набор
(точка, отрезок, многоугольник и набор ребер)

Гладкие кривые
(окружность, эллипс, кривые Безье)

Геометрические тела
параллелепипед, пирамида, эллипсоид.

Геометр. примитивы — 3д набор геом. фигур

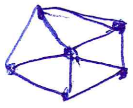
Проблемы: 1) ~~Не вез.~~ не на всех архитектурах аппаратно реализованы
2) мат. описание (аппроксимация) интерполяц / экстраполяц

Выход: 1) Моделить все треугольниками!!! почему?
2) конструктивная геометрия тел   объемные тела операции (∪, ∩)

Примитивы в OpenGL: → точки
отрезки ломанные, циклы
~~полиг~~ многоугольники (квадрат, треугол)
полосы (группы 3х чем их собирать вершин)
шрифты / сфера, цилиндр, конус

Требование: эффективность, надежность

Полигональные модели



1) вершины ·

2) отрезки 3) полилиния 4) полигон 5) полигональная поверхность от сетка

Преимущества ГИС, САПР
1) удобство масштабировани
2) небольшой объем данных
3) аппаратная поддержк

Недостат
4) выполнение топол. операций → построение сечен
5) апрокс. приводит к погрешности

Представление геометрической инф-ии

Куликов Б-Овчинникова (1)

Ур86 Алг. основы современной КГ

Геометрические примитивы -

- Базовый набор геометр. фигур, к-ый лежит в основе всех графич. построений м.б. аппаратно реализован. (отрезок, многоугольник и набор мтср)

В набор примитивов необходимо включать
гладкие кривые (окружности, эллипсы, кривые Безье),
классы поверхностей.
геометрические тела → параллелепипед, пирамиды, эллипсоиды.

проблемы:
не везде реализованы аппаратно
матем описания, средства манипулирования

Самое распростр. использование треугольников

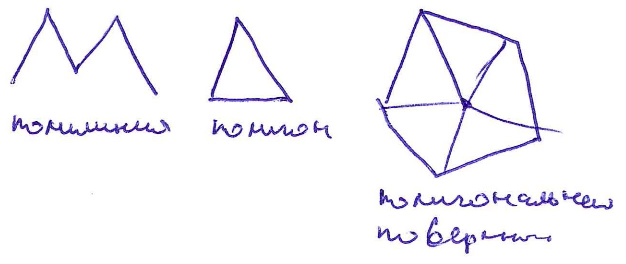
альтернатива ~~конструктивн~~ использование геометрич. тел.
объемные примитивы операции (·∪, ∩)

примитивы в OpenGL → точки (вершины)
отрезки.
ломанные
многоугольники (3-х, 4-х)
полосы (групп 3х или 4-х. с общими вершин.)
шрифты
сфера, цилиндр, конус?

Требования: эффективность
надежность

Полигональные модели

-1) вершины
ю) отрезки. → 1) полилинии
2) полигоны
3) полигональные поверхности (ПП)



преимущества использования (ПМ)
-) удобство масштабирования
.) небольшой объем
:) аппаратная поддержка многих операций

недостатки
построение сечений → выполнение топологических операций
аппроксимация приводит к значит. погрешности.