

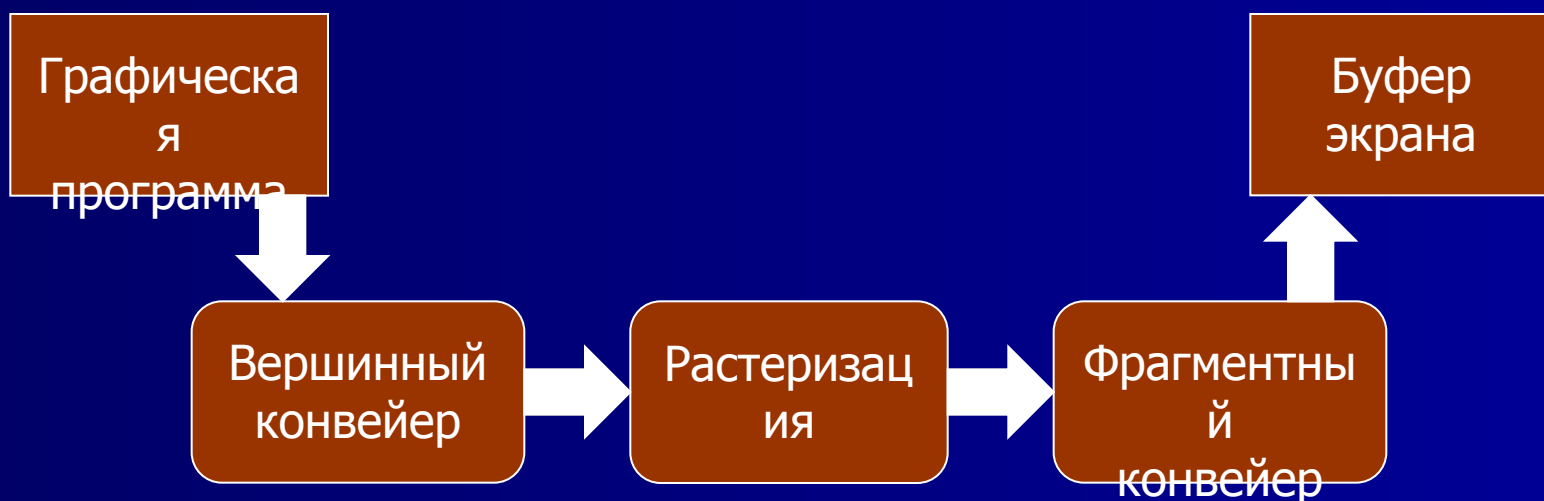


Графический конвейер Преобразования геометрии

Александр Шубин



Введение в проблему



- Цель: преобразовать трёхмерные описания объектов в двухмерную картинку
- Средство: задать строгий порядок преобразований



Вершинный конвейер

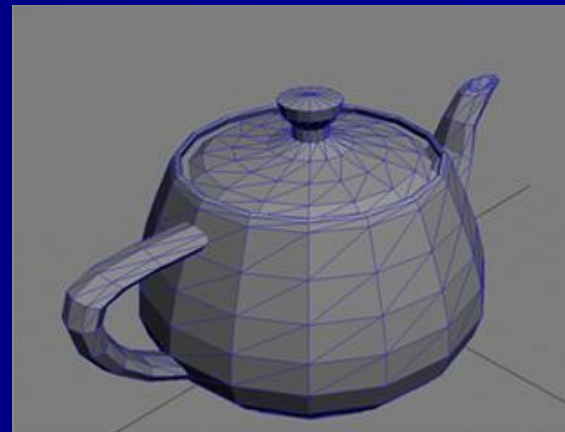


- работает с вершинами
- переходим от трёхмерных предметных координат к двумерным экранным



Вершинные операции

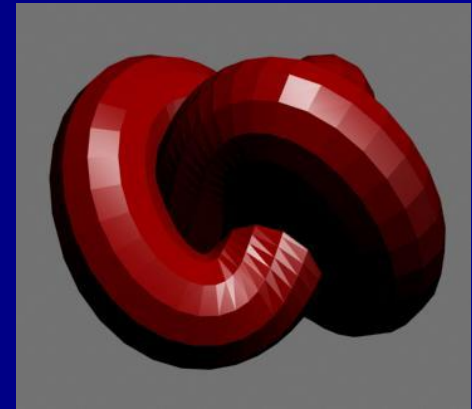
- координаты вершин преобразуются по видовым матрицам и матрицам проекции
- преобразуются нормали и текстурные координаты
- накладывается вычисление освещения





Сборка примитивов

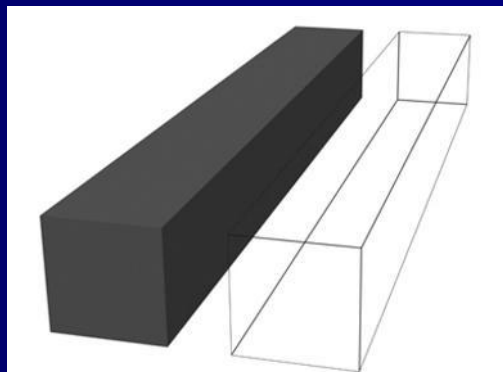
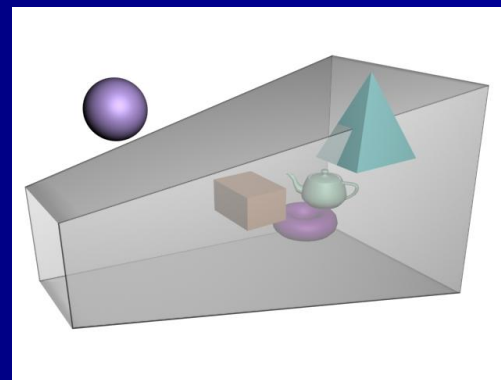
- формирование примитивов
(точки, линии,
многоугольники)
- необходимый этап,
т.к. дальше идёт работа
уже с наборами точек



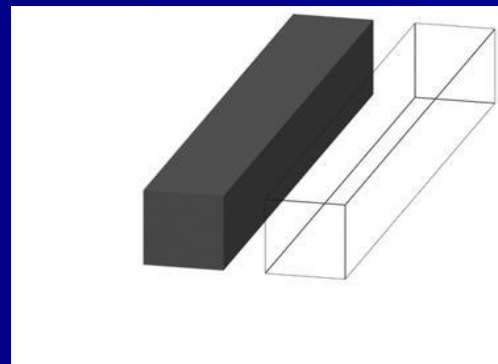


Обработка примитивов

- отсечение по плоскостям
отсечения и
отображаемому объёму
- расчёт перспективы



с перспективой

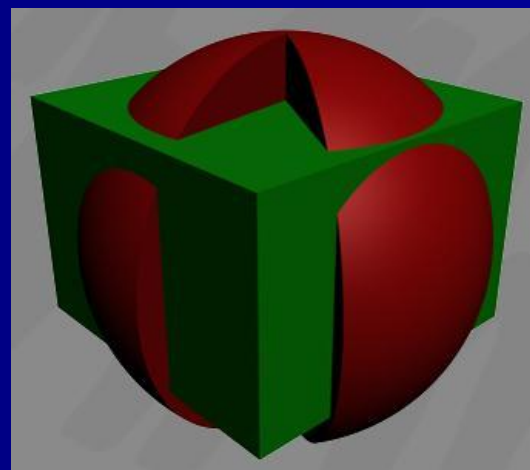


без перспективы



Обработка примитивов

- приведение к оконным (экранным) координатам
- проверка расположен ли примитив на переднем плане (отбраковка)





Растеризация

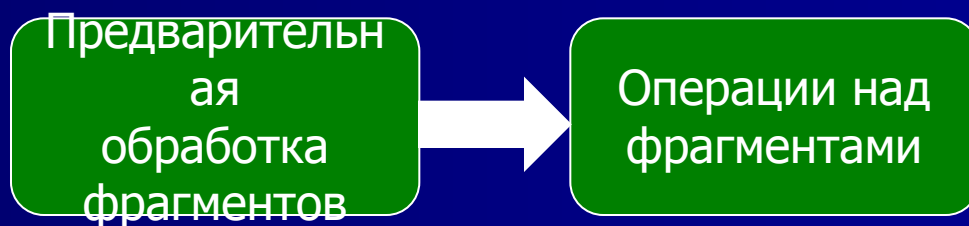


- на входе: обработанные вершины (после вершинного конвейера)
- на выходе: фрагменты, на которые разбиваются примитивы



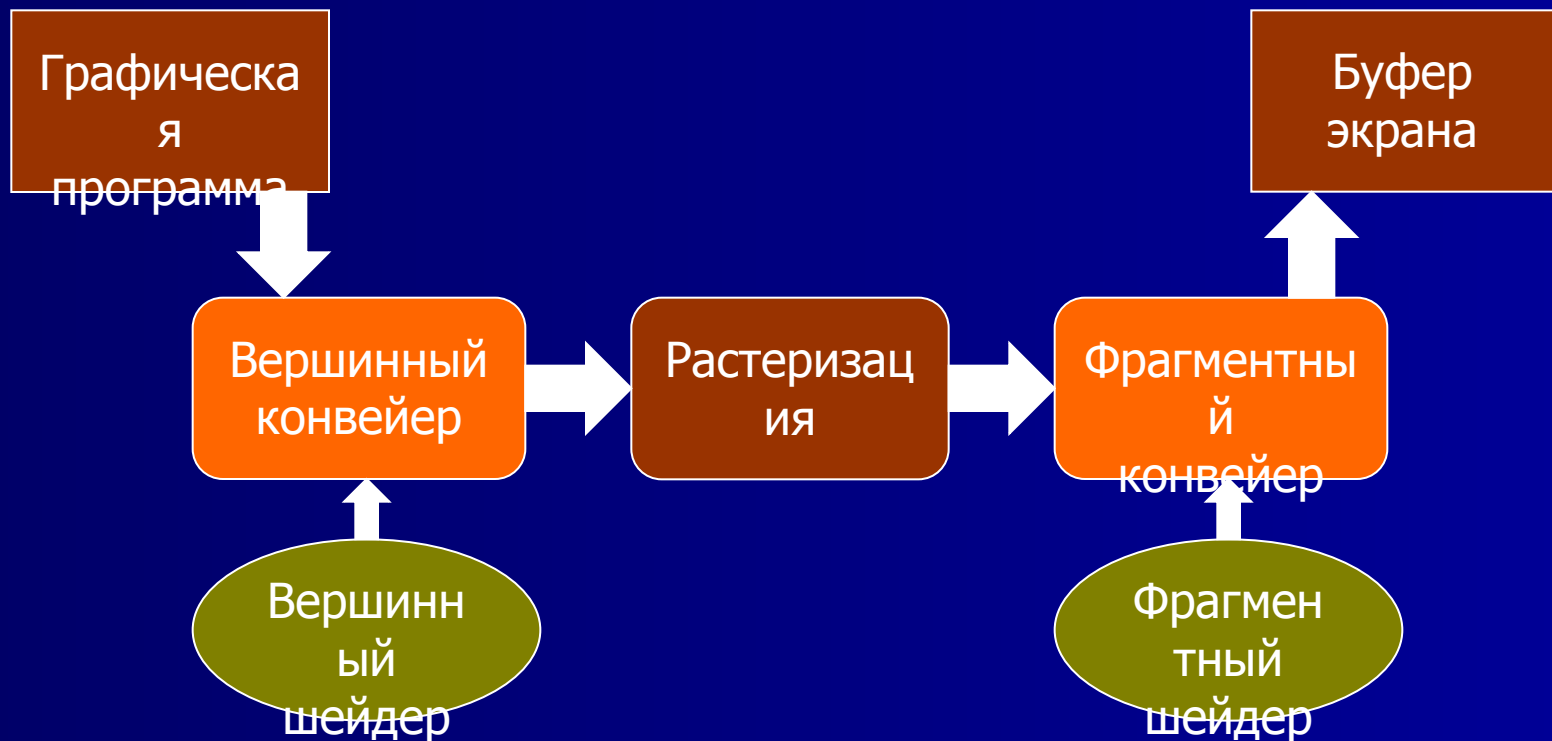
Фрагментный конвейер

Фрагментный конвейер



- ▣ Обработка фрагментов: текстурирование, дымка, сложение цветов...
- ▣ Операции: проверка прозрачности, глубины, отсечение по буферу трафарета...

Программируемый конвейер



- Позволяет заменить фиксированную часть обработки вершин и фрагментов программируемой



Вершинный процессор

Вершинный конвейер

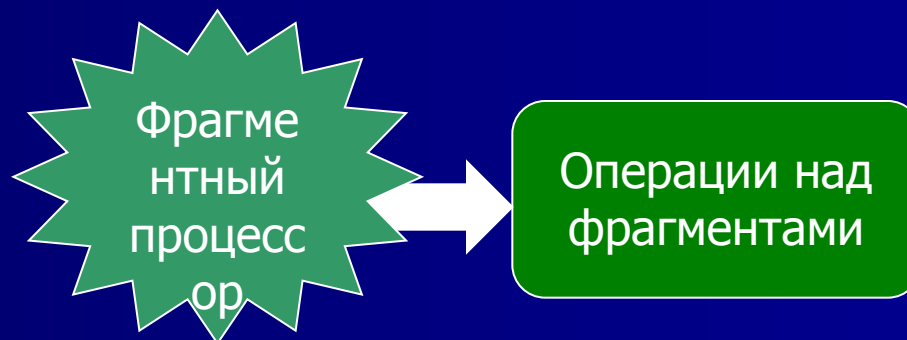


- позволяет программировать этап обработки вершин
- работает по отдельности с каждой вершиной
- должен полностью заменять фиксированную функциональность



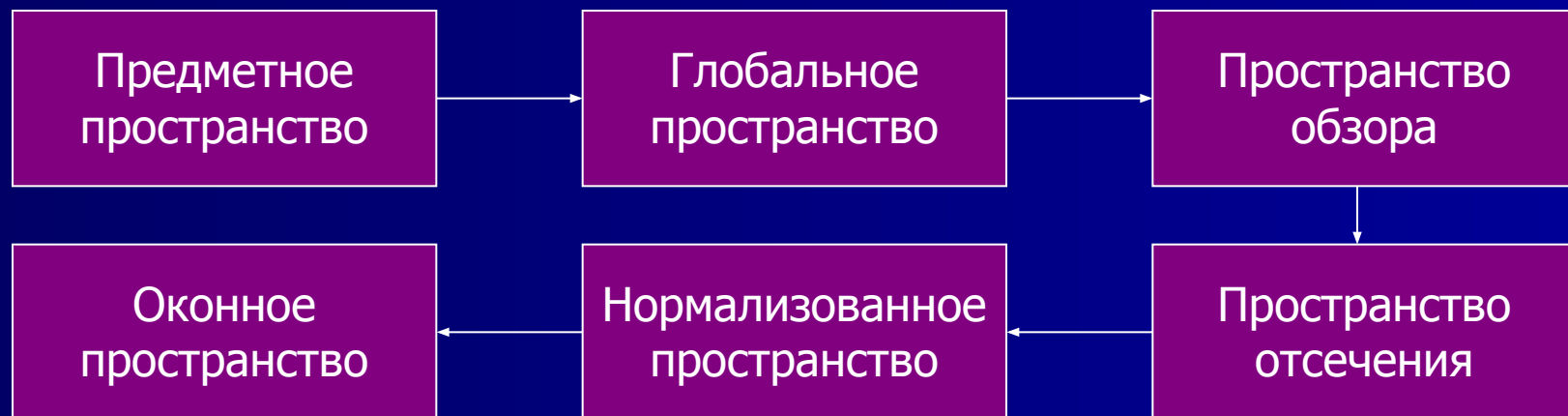
Фрагментный процессор

Фрагментный конвейер



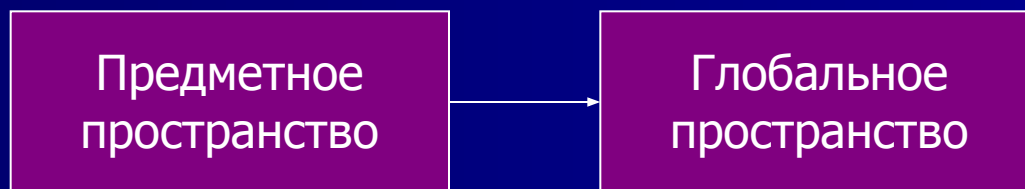
- позволяет программировать этап обработки фрагментов
- работает по отдельности с каждым фрагментом
- должен полностью заменять фиксированную функциональность

Преобразования координат



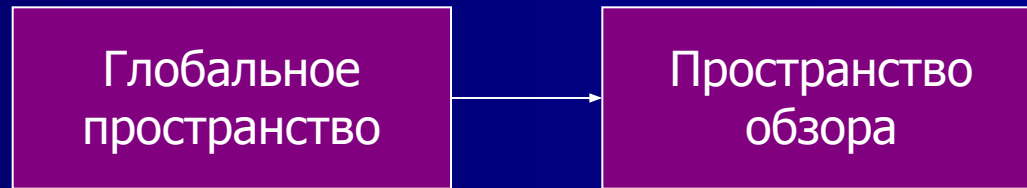
- каждое пространство координат имеет свои свойства
- большая гибкость

Преобразования координат



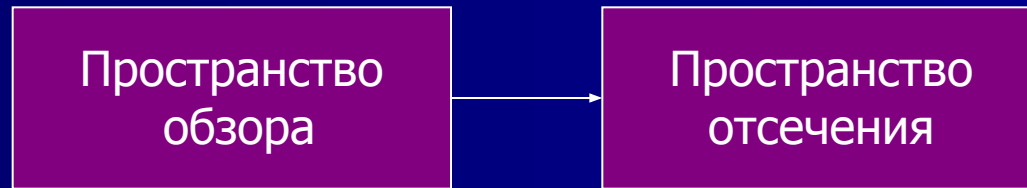
- переходим от одного предмета к нескольким
- необходимы общие единицы измерения
- расположение начала координат должно быть удобно для всей сцены

Преобразования координат



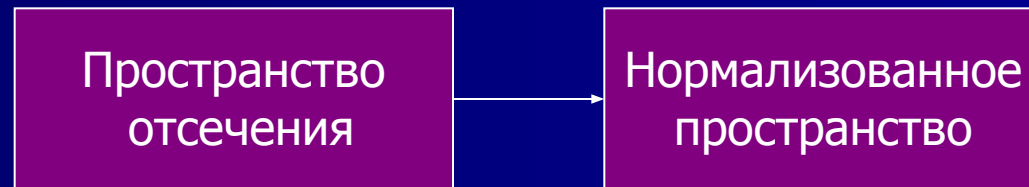
- учитываем параметры обзора (точка обзора, точка фокуса, направление верха)
- начало координат теперь в точке обзора
- в OpenGL сразу идёт переход из предметного пространства в пространство обзора (модельновидовая матрица)

Преобразования координат



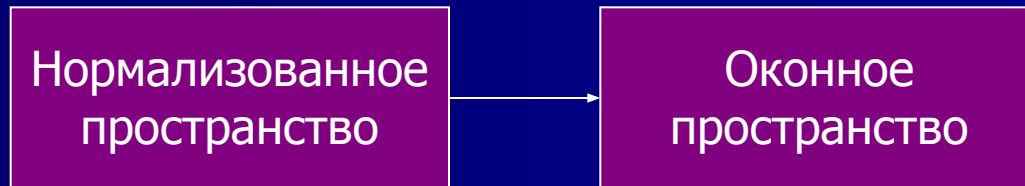
- исключаем примитивы, не входящие в область обзора
- область обзора определяют видимый объём и пользовательские плоскости отсечения
- в OpenGL видимый объём задаётся матрицей проекции

Преобразования координат



- проводится расчёт перспективы
- все графические примитивы помещаются в пространство между $(-1, -1, -1)$ и $(1, 1, 1)$
- промежуточное пространство на пути к окну

Преобразования координат



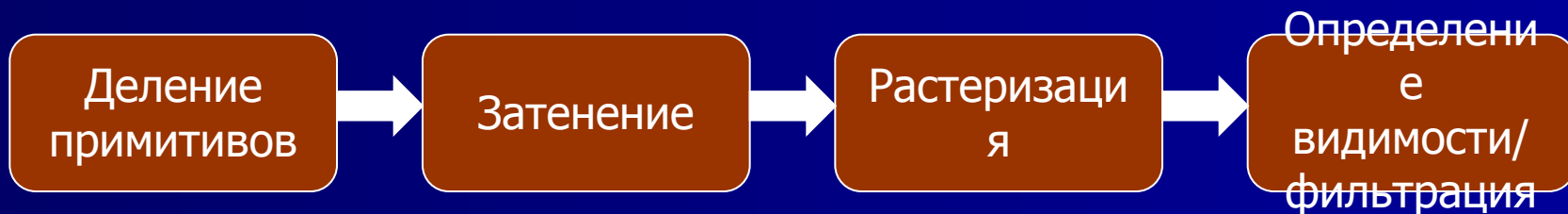
- переходим в оконные координаты
- растеризация происходит в оконных координатах





Другие конвейеры

Конвейер Reyes



- разработана Lusacfilm и Pixar для высококачественного рендеринга сложных сцен
- имеет четыре стадии



Другие конвейеры

Отличия Reyes от OpenGL

- операция текстурирования не нуждается в дополнительной фильтрации
- основной примитив – микрополигон (однородно покрашенный четырёхугольник)
- одна стадия затенения





**Спасибо за
внимание!**

