Компьютерная графика. Современные технологии компьютерной графики и рендеринга

02.04.02 ФИИТ

Разработка мобильных приложений и компьютерных игр

2025-2026

Лекция 0: Введение.
От пикселя к конвейеру:
философия рендеринга в реальном времени

Впечатление от игры или приложения на 80% формируется тем,
что пользователь видит на экране

Компьютерная графика

- это не просто «рисование картинок»
- это наука о синтезе, обработке и визуализации изображений и данных с помощью компьютера

Актуальность для ФИИТ: Где это используется?

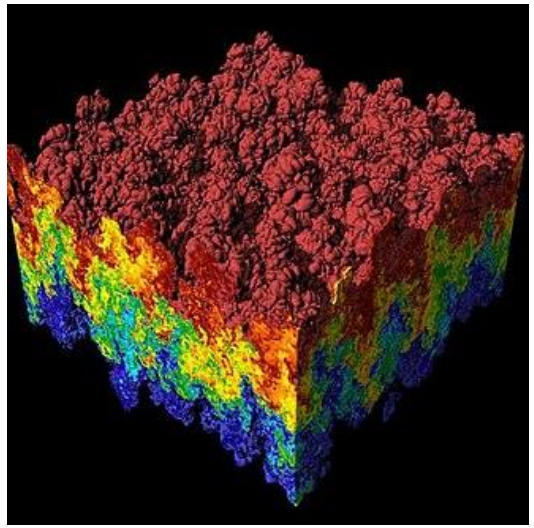
- Конечно — игры, кино, дизайн
- Научная визуализация: Рендеринг данных моделирования (климат, физика частиц, биология)
- Виртуальная и дополненная реальность (VR/AR)
- CV (Computer Vision): Обратная задача — анализ изображений. Графика синтезирует, vision анализирует
- Машинное обучение: Генерация синтетических данных, стиль-трансфер, работа с GANs

Научная визуализация

Рендеринг данных моделирования (климат, инженерные решения, физика частиц, биология и т.п.)

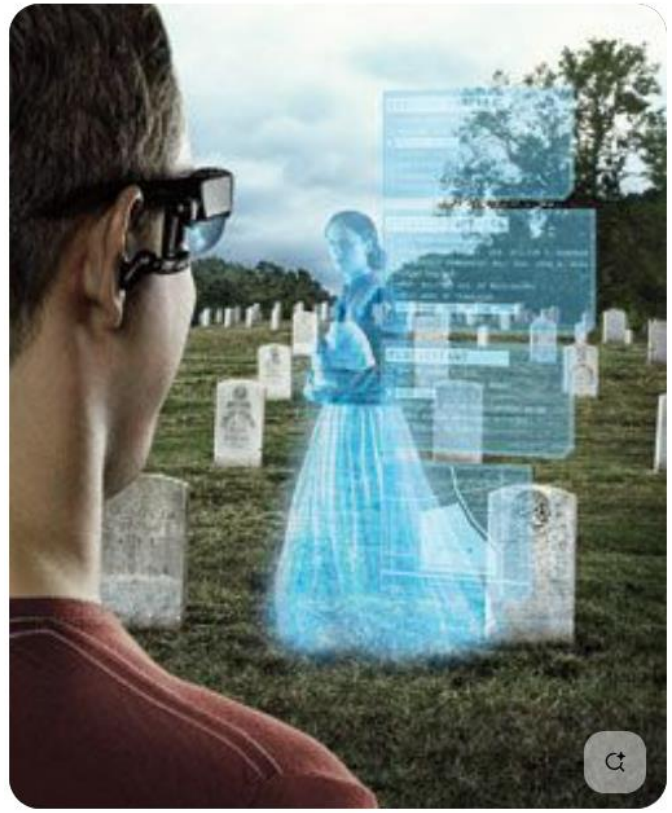
Рендеринг местности

Научная визуализация потока жидкости: поверхностные волны в воде



Научная визуализация моделирования неустойчивости Рэлея — Тейлора, вызванной смешением двух жидкостей

Виртуальная и дополненная реальность (VR/AR)

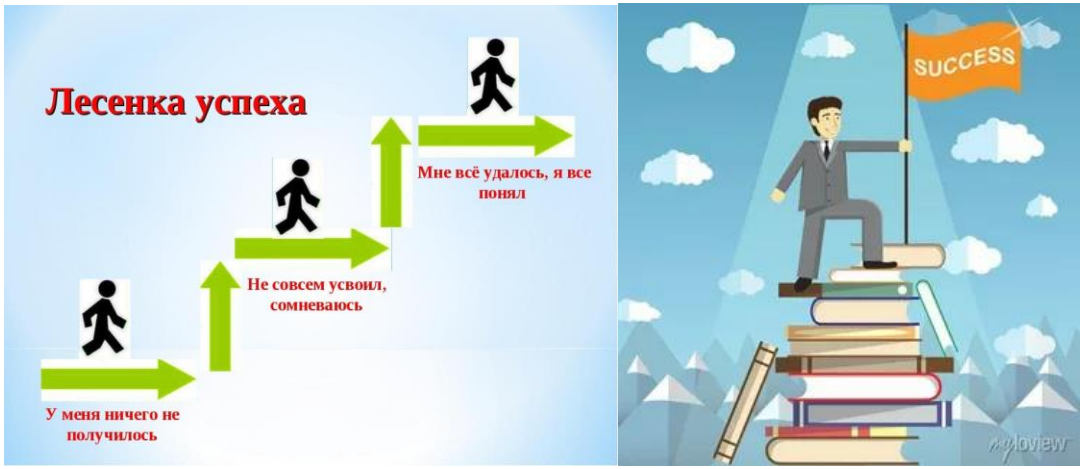


Машинное обучение: Генерация синтетических данных, стиль-трансфер, работа с GANs

Проблема разнородности и стратегия курса

Модуль 0 — Выравнивающий.
Его цель — дать всем общий язык и понимание рендеринга.
Для опытных — это систематизация.
Для новичков — интенсив.

Основные модули.
Здесь мы будем двигаться вместе, но возможно с дифференцированными заданиями

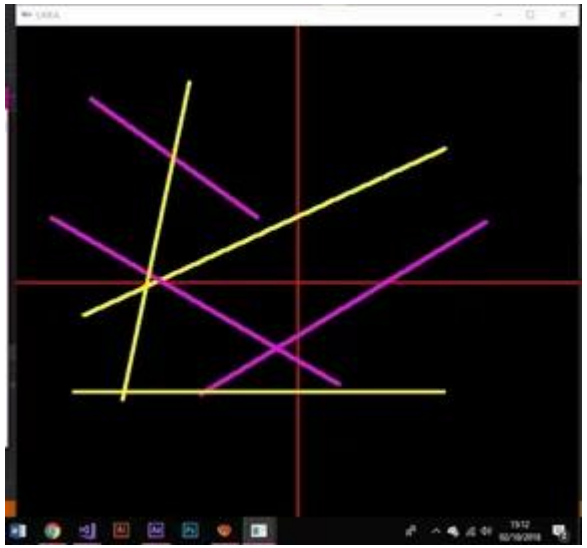


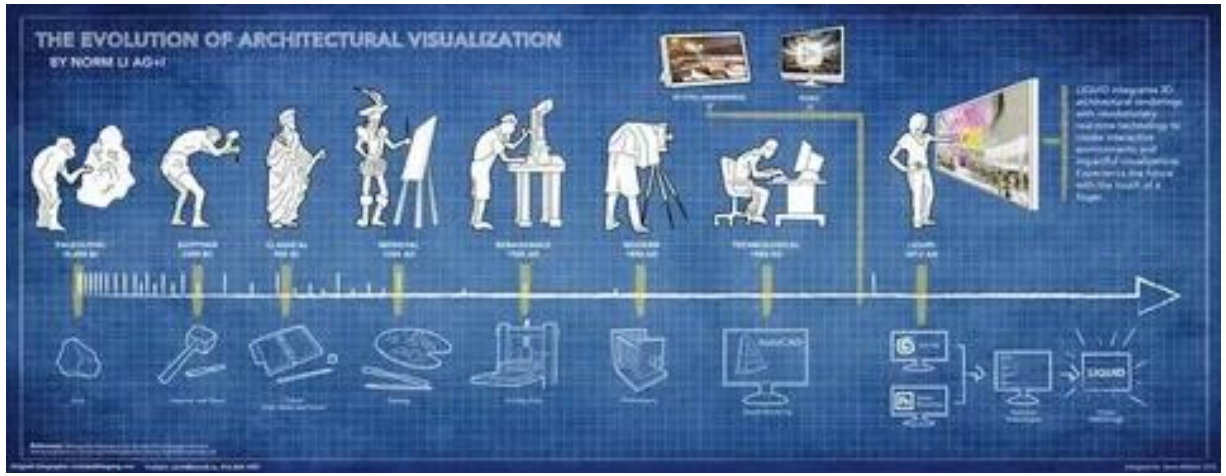
Ваша оценка будет зависеть не от того, что вы знали вчера, а от того, как вы сможете применять полученные знания в курсовом проекте — в создании работающего графического прототипа

Идеальная структура курса

Вход: Ваши знания + Модуль 0 (Фундамент)
Этап 1: Современные API — контроль и производительность
Этап 2: Продвинутый рендеринг (PBR, тени, пост-эффекты) — качество изображения
Этап 3: Мобильная/кроссплатформенная графика — оптимизация и ограничения
Этап 4: Гибридный рендеринг (Ray Tracing, Compute Shaders) — будущее и R&D
Этап 5: Архитектура в движке — системное мышление
Выход: Курсовой проект (технодемка) + навык чтения статей Siggraph/GDC

От линий к шейдерам





Эволюция

Векторные дисплеи (70-е)

Растровые дисплеи (80-е)

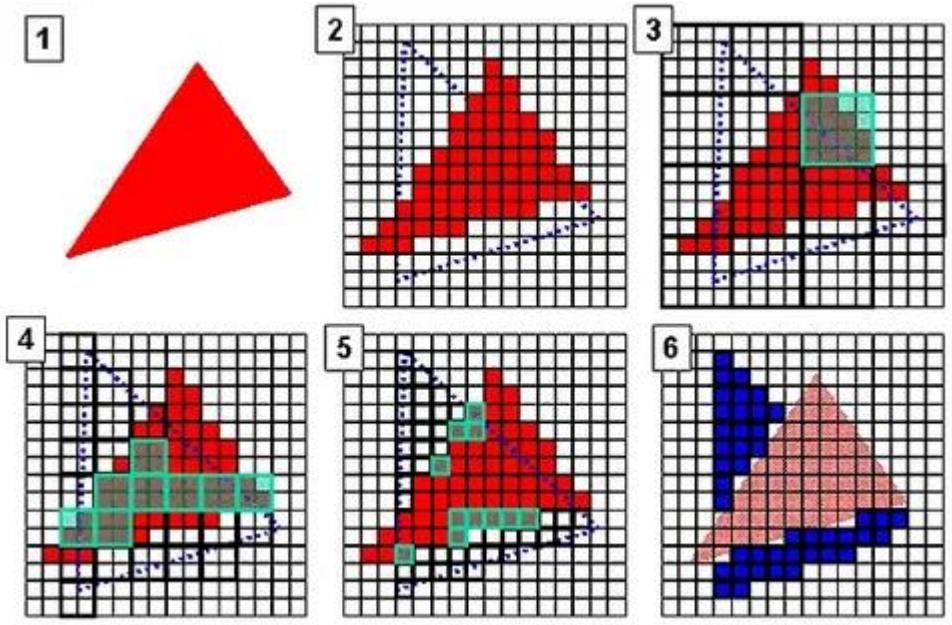
Фиксированный конвейер (90-е — нач. 2000-х)

Векторные дисплеи (70-е)

- Осциллографы
- Spacewars!
- Рисование линиями
- Проблема: нельзя закрасить область

Растровые дисплеи (80-е)

- Матрица пикселей
- Появление растеризации
- Проблема: как быстро закрасить полигон?
- Алгоритмы Брезенхема, заливки



1970-1980

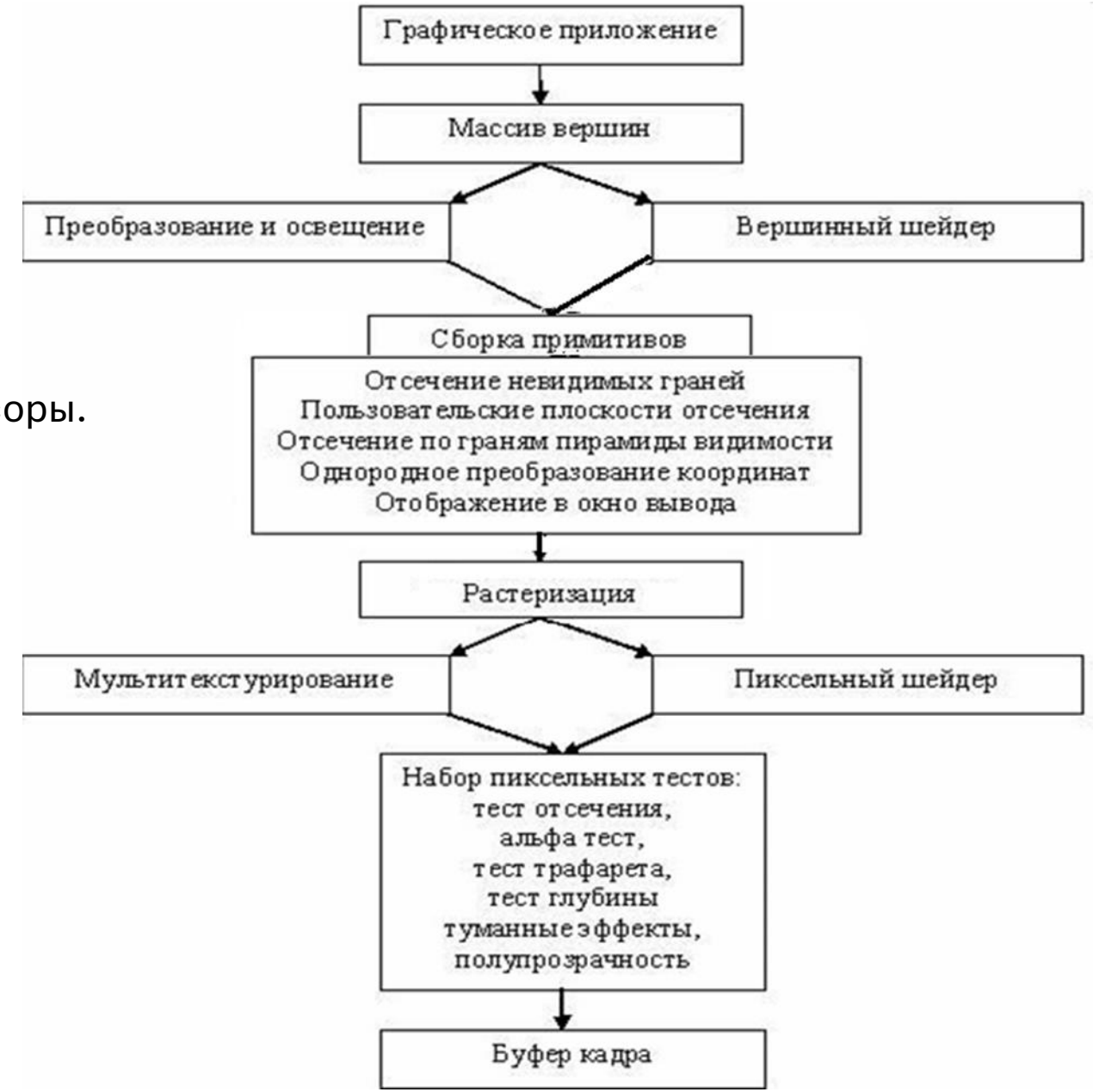
Первые растровые дисплеи

Фиксированный конвейер (90-е — нач. 2000-х)

- OpenGL 1.x, DirectX 7.
- Появление GPU как ускорителя фиксированных операций.
  - Аналогия: Фотоаппарат-мыльница. Есть кнопки «портрет», «пейзаж».
    Внутри — жесткая логика.
  - Что было фиксировано: Преобразования (вершины -> экран),
    освещение по Гуро/Фонгу, текстурные координаты.
  - Главная проблема: Негибкость.
    Хочешь мультяшный стиль (cel-shading) или воду? Нельзя.

Революция: Программируемый конвейер

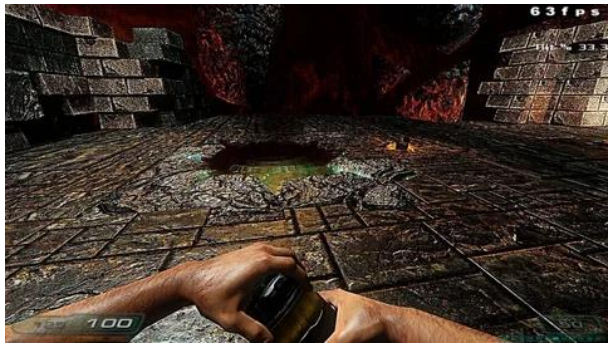
В начале 2000-х происходит переворот.
Вместо жесткой логики в GPU встраивают программируемые блоки — шейдерные процессоры.



Шейдер — это маленькая программа, которую вы пишете, и которая выполняется для каждой вершины (вершинный шейдер) или для каждого пикселя (фрагментный шейдер) параллельно на тысячах ядер GPU

Или тесселяционный, или геометрический, или вычислительный

Рождение современной графики



Doom 3 (2004) — одна из первых игр, где шейдеры массово использовались для per-pixel lighting и сложных материалов.

GPU как суперкомпьютер: переход к GPGPU

GPGPU (англ. **General-Purpose computing on Graphics Processing Units**) — **вычисления общего назначения на графических процессорах**.

Это техника использования графического процессора (GPU), предназначенного для компьютерной графики, для выполнения математических вычислений, которые обычно проводит центральный процессор (CPU).

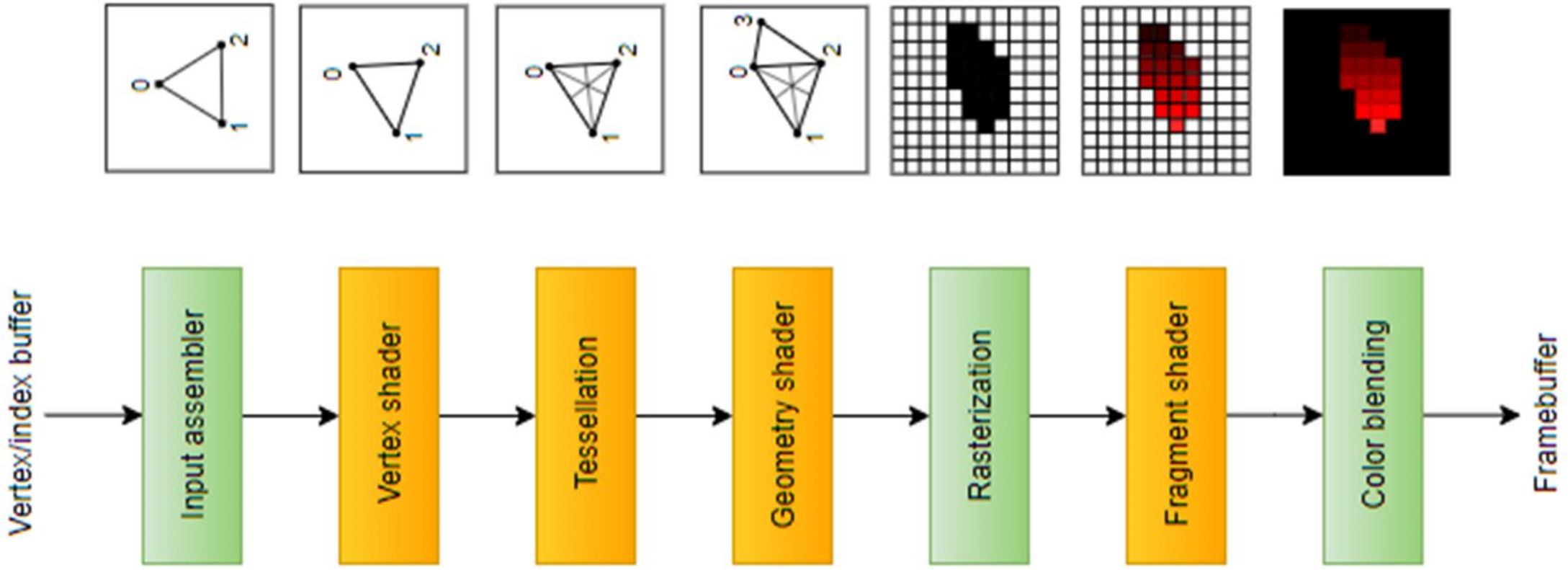
Оказалось, что архитектура GPU — тысячи простых ядер, идеально подходит не только для цвета пикселей.
Они могут считать что угодно, что можно распараллелить.

Могут считать что угодно

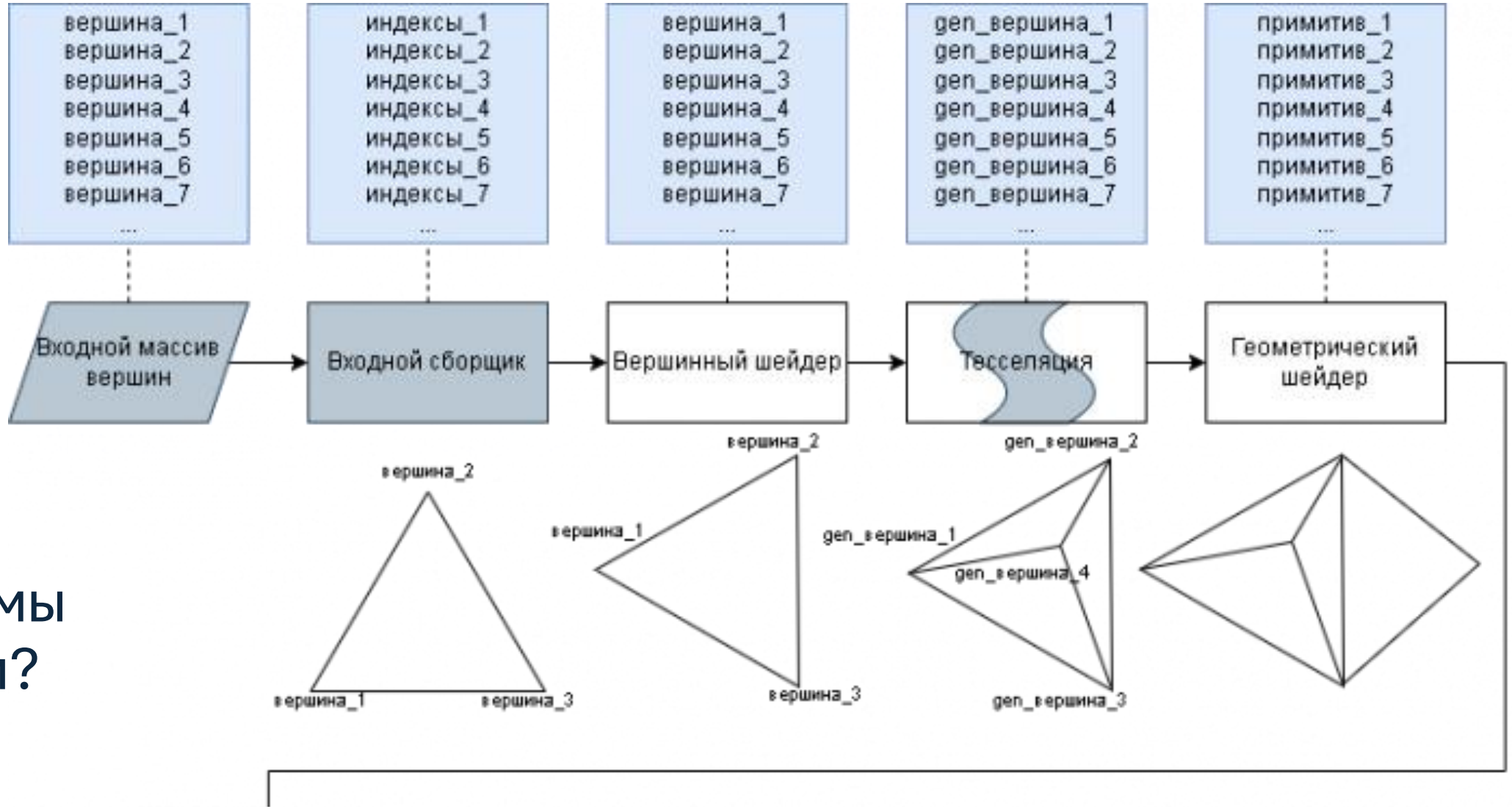
- Физика: Расчет столкновений тысяч частиц
- ИИ: Матричные умножения в нейронных сетях (именно поэтому для майнинга и ML используют видеокарты)
- Научные расчеты: Моделирование климата, белков

Современный GPU — это массово-параллельная вычислительная платформа.
Графика — лишь одно из его приложений, хотя и основное.
В нашем курсе мы, возможно, коснемся этого через вычислительные шейдеры (Compute Shaders).

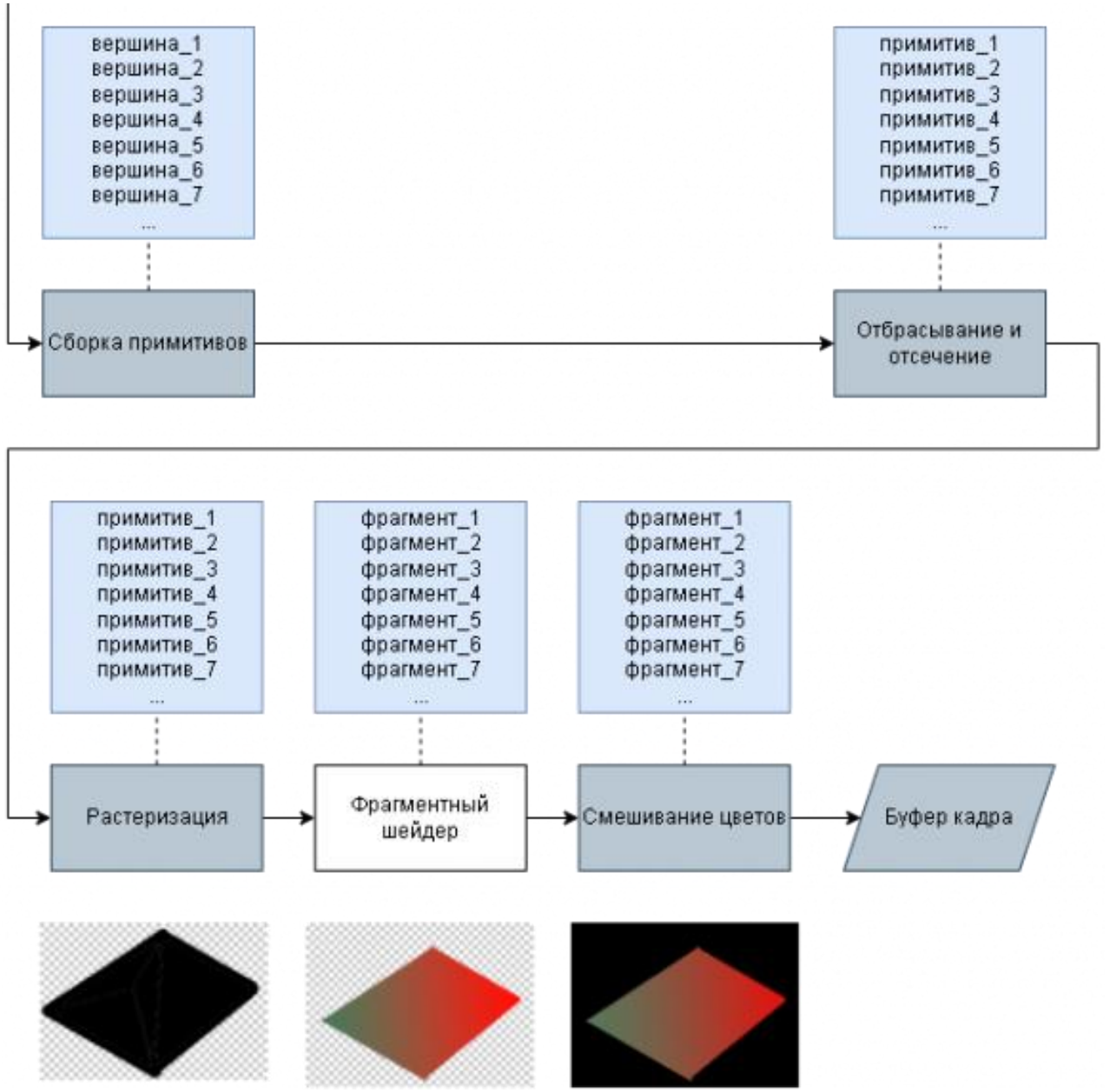
Анатомия конвейера рендеринга



А что мы видим?



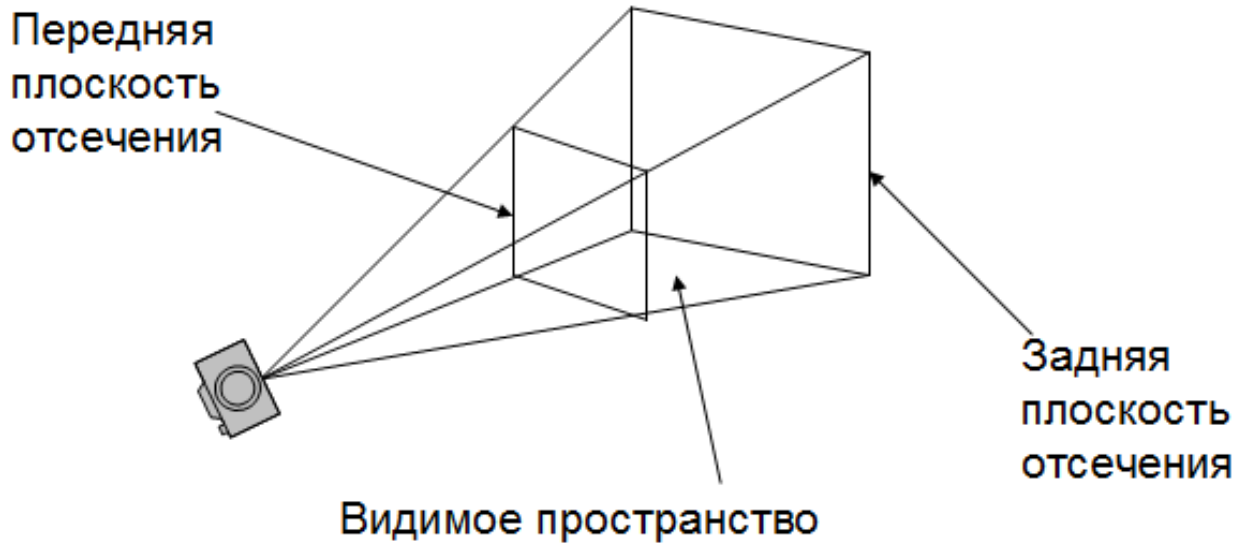
А что мы видим?



Повершинные операции (per-vertex operations)

- В течение этого этапа вершины преобразуются в примитивы
  - Различные матричные преобразования
  - Пространственные координаты проецируются из 3D в экранные
- Если используется текстурирование, координаты текстуры могут быть сгенерированы и изменены на этом шаге
- Если используется освещение, вычисления, связанные с ним, производятся с использованием:
  - трансформированных вершин,
  - нормалей поверхностей,
  - позиций источников света,
  - свойств материала,
  - а также другой информации, позволяющей вычислить цветовую величину

Сборка примитивов. Отсечение



- Отсечение — большая часть сборки примитивов. Это уничтожение частей геометрии, выпадающих за полупространство, определенное плоскостью
  - Отсечение точек просто отвергает или не отвергает вершину
  - Отсечение линий или полигонов может добавить дополнительные вершины в зависимости от того, как именно линия или полигон отсекаются

Сборка примитивов. Что дальше?

- Перспективное разделение
- Операции с портом просмотра (viewport)
- Если включён режим, то выполняется тест на лицевые грани
- Результатом этого этапа являются завершенные примитивы, т.е. трансформированные и отсеченные вершины со связанными цветом, глубиной и, иногда, координатами текстуры

Растеризация

- Растеризация – это процесс преобразования геометрических и пиксельных данных во фрагменты
- Каждый фрагмент соответствует пикселю в буфере кадра
- При развертке двух вершин в линию или вычислении внутренних пикселей полигона принимаются в расчет
  - шаблоны линий и полигонов
  - толщина линии
  - размер точек
  - модель заливки
  - вычисления связанные со сглаживанием

Операции над фрагментами

Могут изменить или даже выбросить некоторые фрагменты
Операции могут быть включены или выключены

- текстурирование
- вычисления тумана
- тест отреза (scissor test)
- альфа-тест
- тест трафарета (stencil test)
- тест буфера глубины

Если фрагмент не проходит один из включенных тестов, это может закончить его путь по конвейеру
Далее могут быть произведены наложение, смешивание цветов (dithering), логические операции и маскирование с помощью битовой маски

Наконец, фрагмент заносится в соответствующий буфер, где становится пикселем

Наложение текстуры

- Если используется несколько изображений текстур, разумно поместить их в объекты текстуры, чтобы можно было легко переключаться между ними
- Некоторые реализации OpenGL могут иметь дополнительные ресурсы для ускорения операций с текстурами. Например, может существовать специализированная быстрая текстурная память

Три кита: Ресурсы, Состояние, Команды

Ресурсы (Data): ***Что мы рисуем***	**Состояние (Pipeline State):** **Как мы это рисуем**	**Команды (Commands):** ***Что нужно сделать***
• **Буферы (Buffers):** • **Текстуры (Textures):** • **Сэмплеры (Samplers):**	• **Шейдерная программа (Shader Program)** • **Настройки растеризации:** • **Настройки тестов и смешивания:**	• **Draw Call:** • **Командный буфер (Command Buffer):**

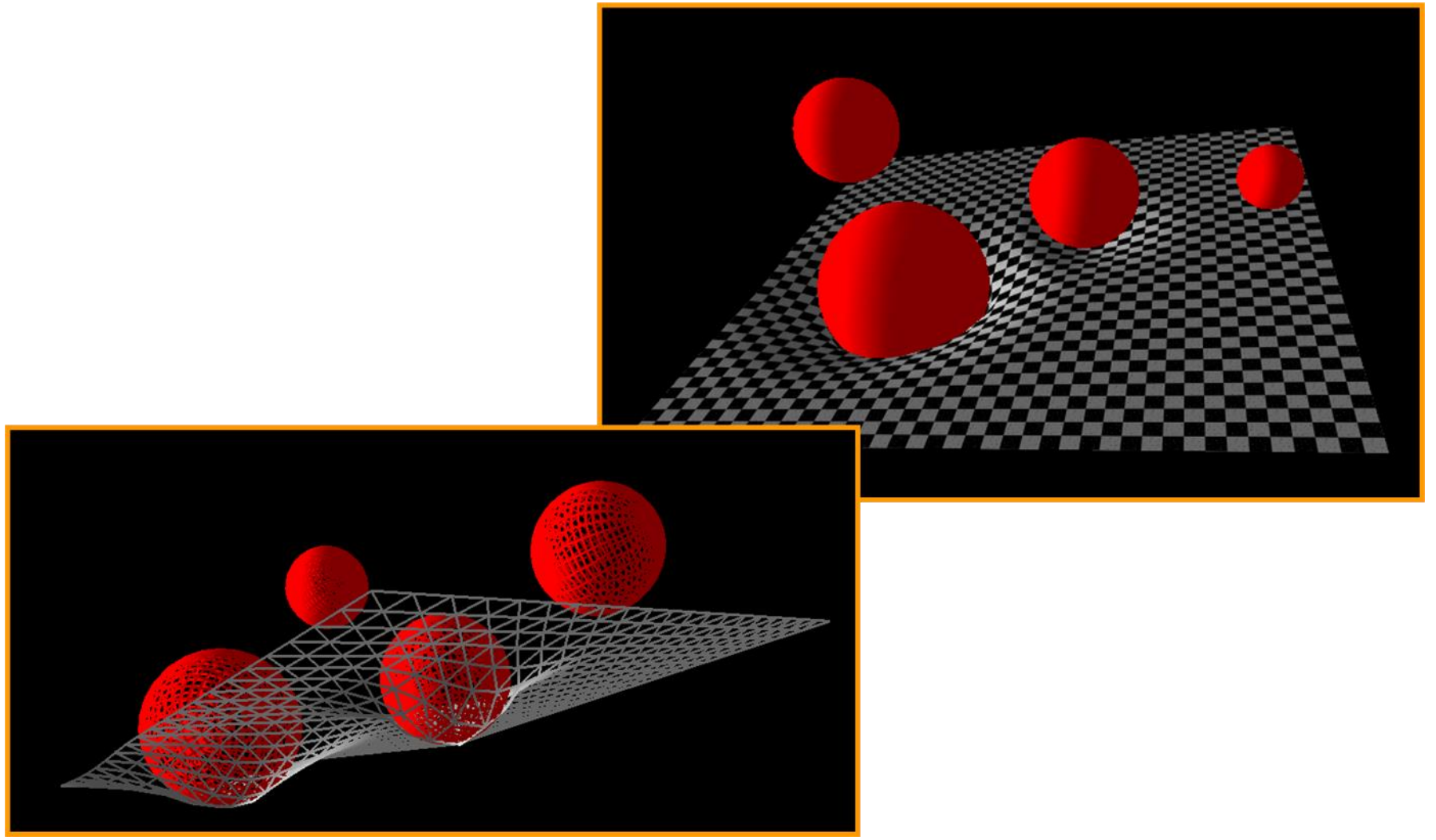
Вся графика — это процесс настройки состояния, привязки ресурсов и отправки draw call’ов.

Производительность упирается в то,
как часто вы меняете состояние и сколько draw call'ов отправляете. Это ключ к оптимизации.

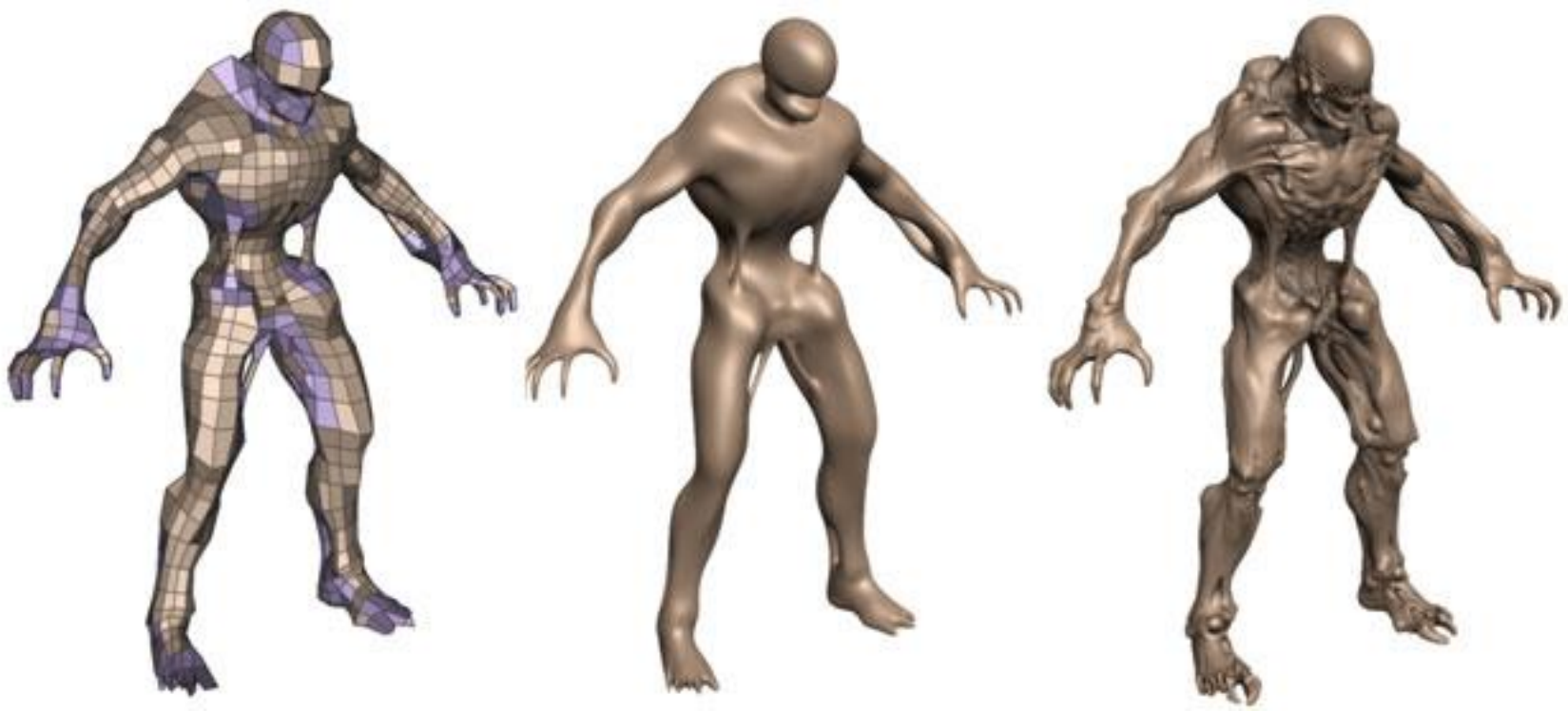
Конвейер как фабрика

- Сырье (Ресурсы): Доски (вершины), краска (текстуры), чертежи (шейдеры)
- Конвейерная линия (Pipeline): Станки (шейдерные процессоры), окраска (фрагментный шейдер), контроль качества (тесты)
- Настройка линии (State): Переключение с производства стульев на производство столов
- Заказ (Draw Call): Запусти линию, чтобы сделать 100 стульев

Примеры использования шейдеров



Применение тесселяции



Применение тесселяции к грубой модели (слева) позволяет создавать более гладкую модель (посередине). Использование карт смещения (справа) обеспечивает персонажам реалистичность кинематографического уровня.
© Kenneth Scott, id Software 2008

Скининг (skinning) — скриншот из Call of Duty 2

Matrix pallete skinning для скелетной анимации персонажей с большим количеством «костей».

Над вершинами каждого из персонажей поработал алгоритм скининга.

Причём, 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 + Displacement Mapping

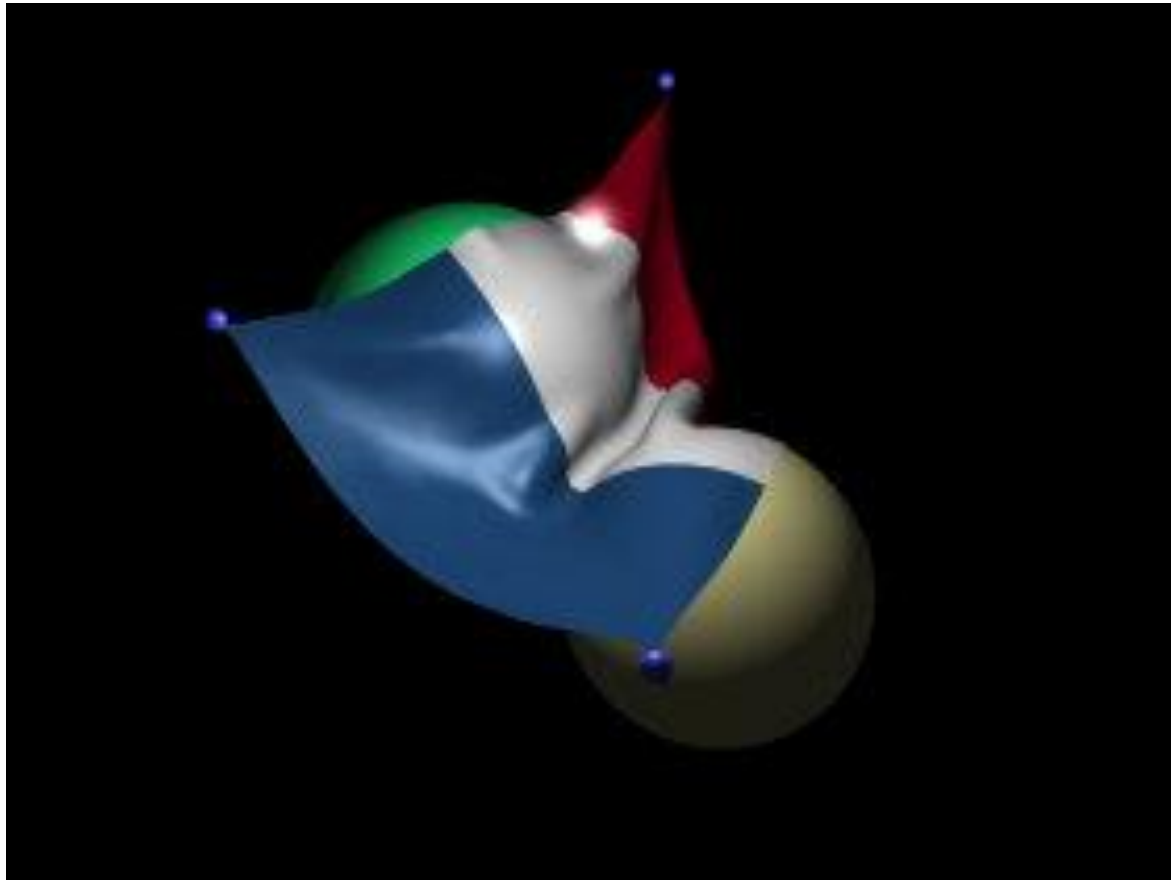
- Как самый явный и эффектный пример — создание реалистичных волн в динамике. Примеры в играх F.E.A.R. и Pacific Fighters. (Displacement Mapping в дополнение к Bump Mapping).

- 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

Анимация 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 материалов, подобных ткани

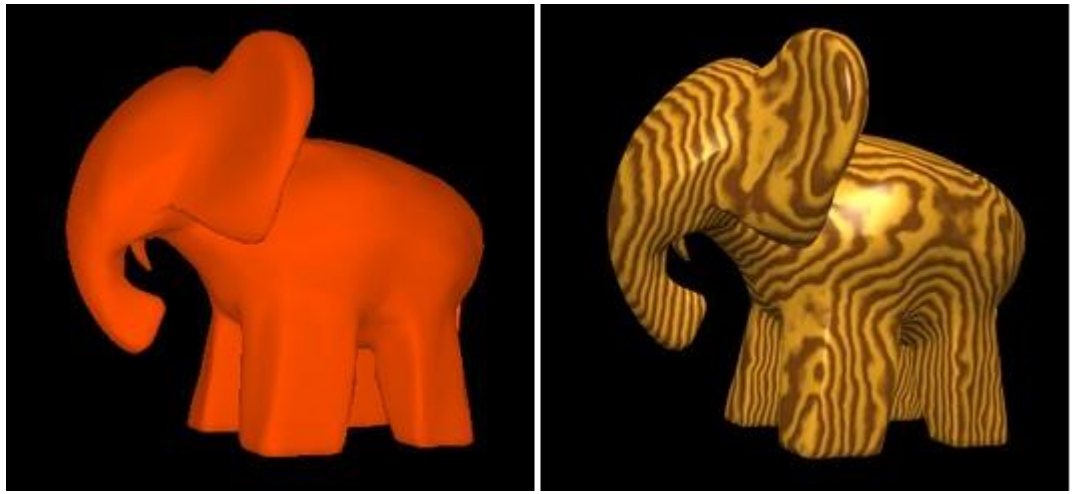
Toon shading/Cel shading

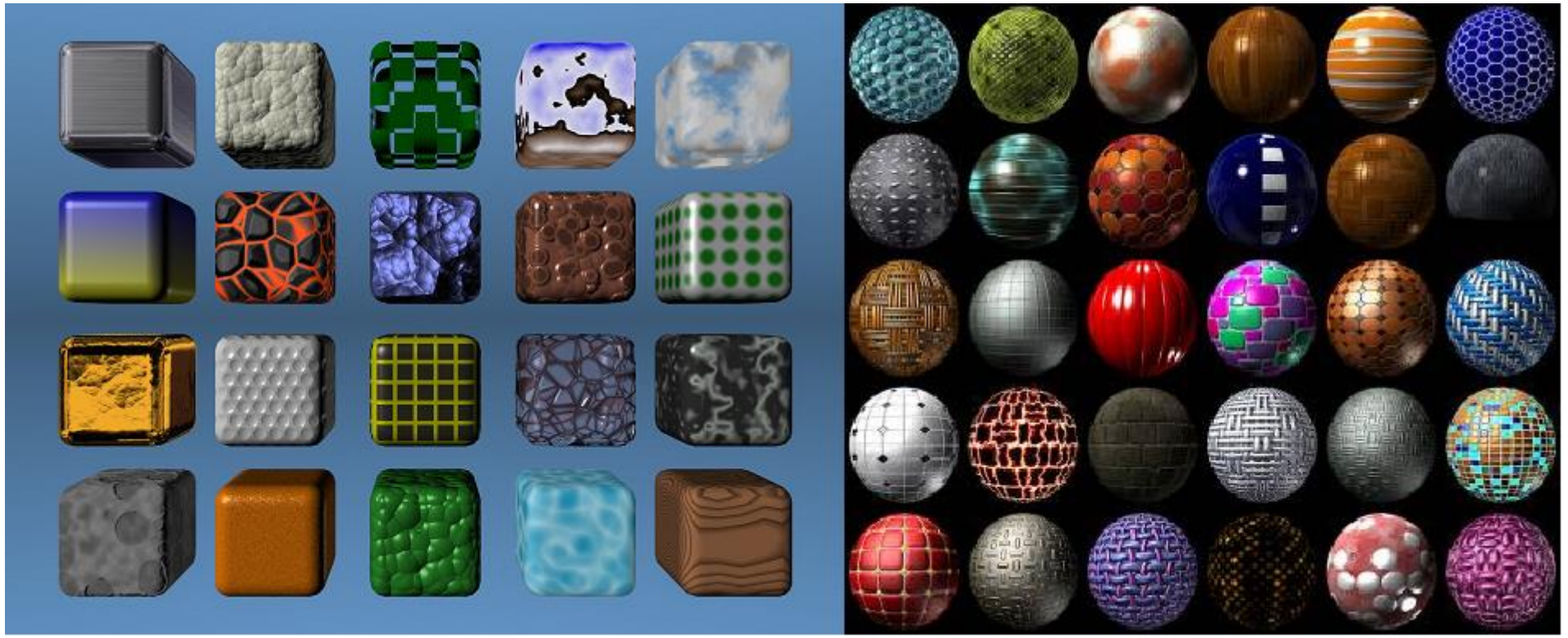
Мультитекстурирование

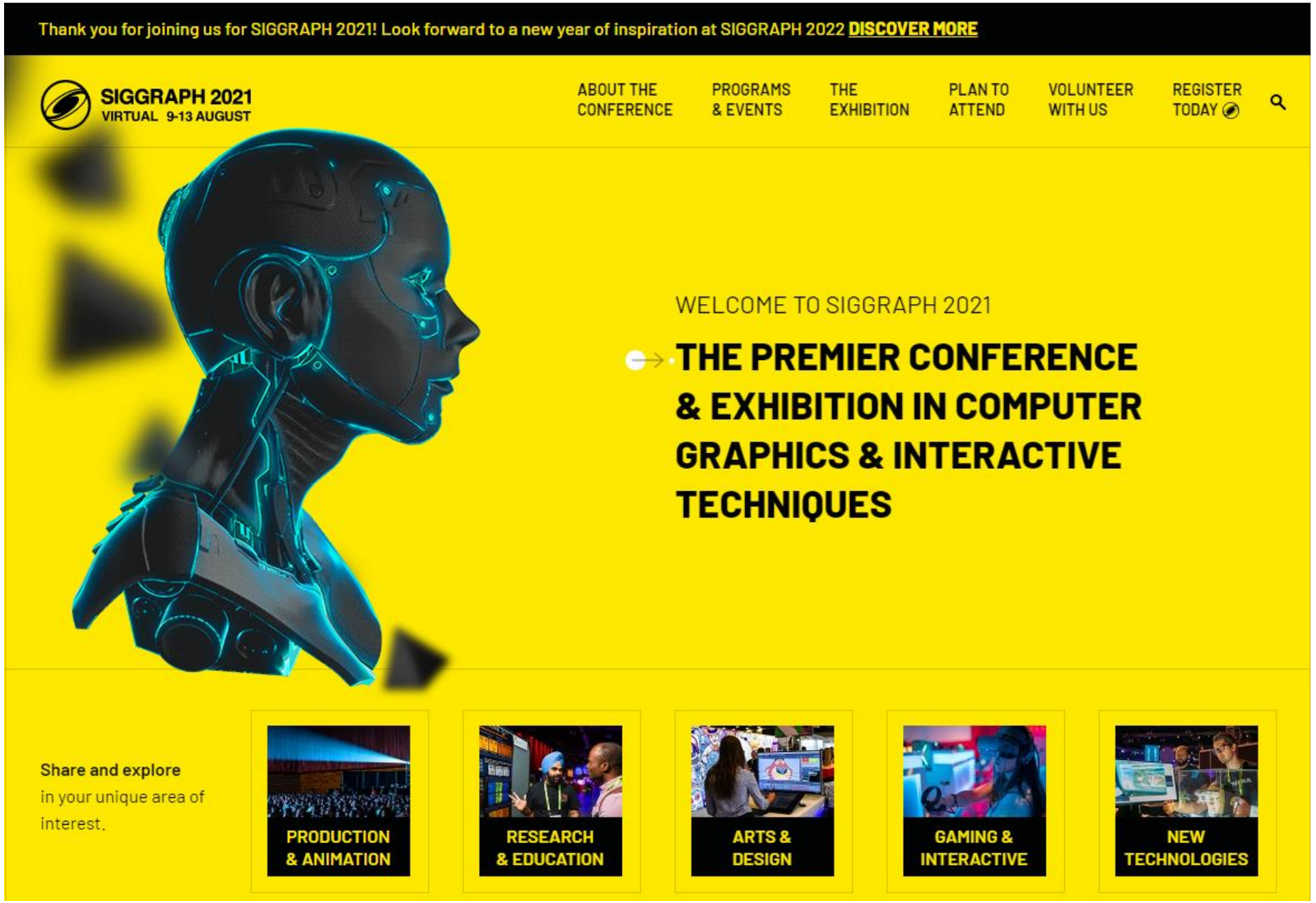
Несколько слоев текстур (colormap, detailmap, lightmap и т.д.).
Используется вообще во всех играх.

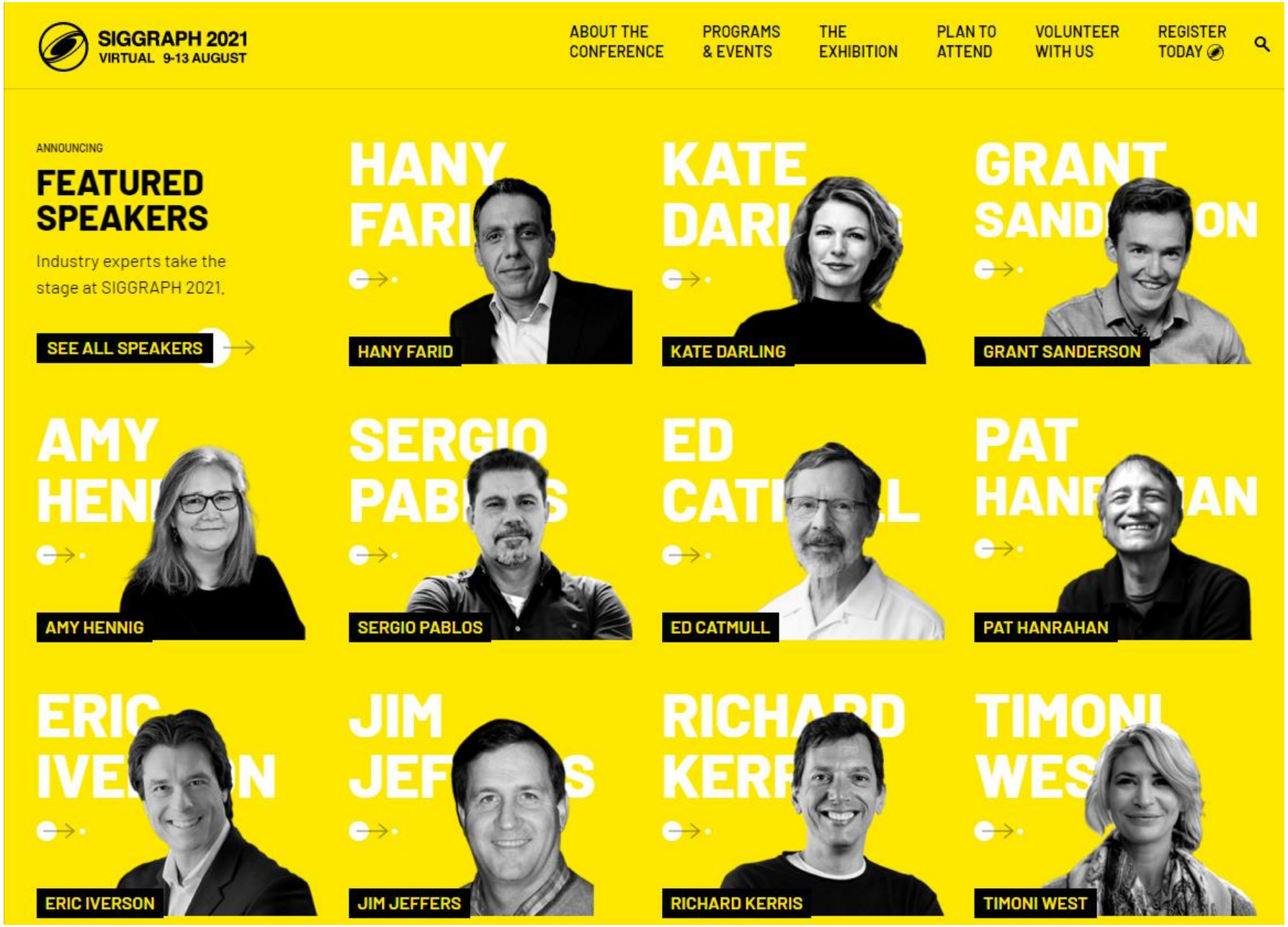
Постобработка кадра: Depth of Field

Процедурные текстуры





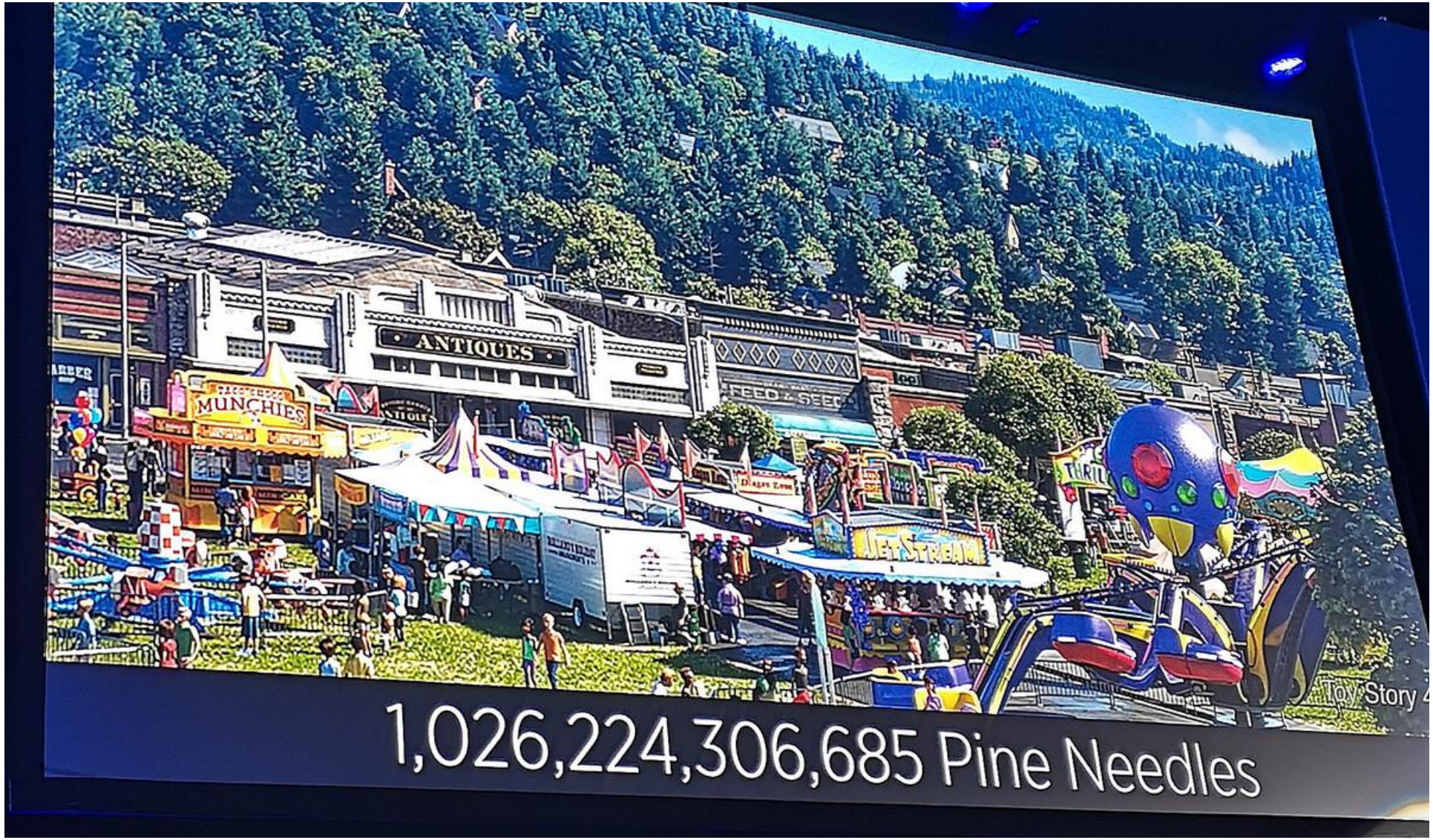

Thank you for joining us for SIGGRAPH 2021! Look forward to a new year of inspiration at SIGGRAPH 2022 DISCOVER MORE
SIGGRAPH 2021
VIRTUAL 9-13 AUGUST
ABOUT THE CONFERENCE
PROGRAMS & EVENTS
THE EXHIBITION
PLAN TO ATTEND
VOLUNTEER WITH US
REGISTER TODAY
WELCOME TO SIGGRAPH 2021
THE PREMIER CONFERENCE & EXHIBITION IN COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
Share and explore
in your unique area of interest.
PRODUCTION & ANIMATION
RESEARCH & EDUCATION
ARTS & DESIGN
GAMING & INTERACTIVE
NEW TECHNOLOGIES


SIGGRAPH 2021
VIRTUAL 9-13 AUGUST
ABOUT THE CONFERENCE
PROGRAMS & EVENTS
THE EXHIBITION
PLAN TO ATTEND
VOLUNTEER WITH US
REGISTER TODAY
ANNOUNCING
FEATURED SPEAKERS
Industry experts take the stage at SIGGRAPH 2021.
SEE ALL SPEAKERS
HANY FARID
KATE DARLING
GRANT SANDERSON
AMY HENNIG
SERGIO PABLOS
ED CATMULL
PAT HANRAHAN
ERIC IVERSON
JIM JEFFERS
RICHARD KERRIS
TIMONI WEST

В первой Корпорации Монстров у Салливана было 2 миллиона волос(кривых)

В четвертой Истории Игрушек счет пошел на миллиарды (листьев)

И даже на триллионы



И всякие другие интересные вещи

https://www.youtube.com/watch?v=Lu7zyqHMB8Q