Процесс построения изображения

Быковских Дмитрий Александрович

13.09.2025

Процесс построения изображения

Быковских Дмитрий Александрович

13.09.2025

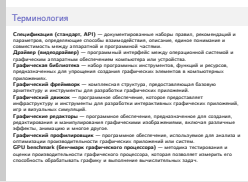
Содержание

- Программные средства
- Процесс построения изображения
- Модель графического конвейера

Содержание
- Программные средства
- Процесс построения изображения
- Модель графического конвейера

Терминология

Спецификация (стандарт, API) — документированные наборы правил, рекомендаций и параметров, определяющие способы взаимодействия, описание, единое понимание и совместимость между аппаратной и программной частями.
Драйвер (видеодрайвер) — программный интерфейс между операционной системой и графическим аппаратным обеспечением компьютера или устройства.
Графическая библиотека — набор программных инструментов, функций и ресурсов, предназначенных для упрощения создания графических элементов в компьютерных приложениях.
Графический фреймворк — комплексная структура, предоставляющая базовую архитектуру и инструменты для разработки графических приложений.
Графический движок — программное обеспечение, которое предоставляет инфраструктуру и инструменты для разработки интерактивных графических приложений, игр и визуальных симуляций.
Графические редакторы — программное обеспечение, предназначенное для создания, редактирования и манипулирования графическими изображениями, включая различные эффекты, анимацию и многое другое.
Графический профилировщик — программное обеспечение, используемое для анализа и оптимизации производительности графических приложений или систем.
GPU benchmark (бенчмарк графического процессора) — методика тестирования и оценки производительности графического процессора, которая позволяет измерить его способность обрабатывать графику и выполнение вычислительных задач.


Терминология

Спецификация (стандарт, API) — документированные наборы правил, рекомендаций и параметров, определяющие способы взаимодействия, описание, единое понимание и совместимость между аппаратной и программной частями.
Драйвер (видеодрайвер) — программный интерфейс между операционной системой и графическим аппаратным обеспечением компьютера или устройства.
Графическая библиотека — набор программных инструментов, функций и ресурсов, предназначенных для упрощения создания графических элементов в компьютерных приложениях.
Графический фреймворк — комплексная структура, предоставляющая базовую архитектуру и инструменты для разработки графических приложений.
Графический движок — программное обеспечение, которое предоставляет инфраструктуру и инструменты для разработки интерактивных графических приложений, игр и визуальных симуляций.
Графические редакторы — программное обеспечение, предназначенное для создания, редактирования и манипулирования графическими изображениями, включая различные эффекты, анимацию и многое другое.
Графический профилировщик — программное обеспечение, используемое для анализа и оптимизации производительности графических приложений или систем.
GPU benchmark (бенчмарк графического процессора) — методика тестирования и оценки производительности графического процессора, которая позволяет измерить его способность обрабатывать графику и выполнение вычислительных задач.

Графическое API (Application Programming Interface) — набор интерфейсов и функций, предоставляемых операционной системой или программной средой, который позволяет программам взаимодействовать с аппаратным обеспечением для рендеринга графики. Графические API позволяют разработчикам управлять отображением 2D- и 3D-графики на экране, взаимодействовать с графическим процессором (GPU), управлять ресурсами (например, текстурами, шейдерами) и другими элементами графики.

Спецификации (API) и графические библиотеки

OpenGL (1992, Silicon Graphics)
Открытая кросс-платформенная спецификация для работы с 2D и 3D графикой
Mesa (1995, Brian Paul)
Свободная реализация графических API OpenGL (позже Vulkan и др.) с открытым исходным кодом
DirectX (1995, Microsoft)
Пакет графических API для работы с играми и мультимедийными приложениями на платформе Windows
WebGL (2011, Khronos Group)
Графический API для веб-браузеров
Mantle (2013, AMD)
Спецификация низкоуровневого API
Vulkan (2016, Khronos Group)
Низкоуровневый (требует явного управления памятью и ресурсами) высокопроизводительный кроссплатформенный API для работы с 2D и 3D графикой

Спецификации (API) и графические библиотеки
OpenGL (1992, Silicon Graphics)
Открытая кросс-платформенная спецификация для работы с 2D и 3D графикой
Mesa (1995, Brian Paul)
Свободная реализация графических API OpenGL (позже Vulkan и др.) с открытым исходным кодом
DirectX (1995, Microsoft)
Пакет графических API для работы с играми и мультимедийными приложениями на платформе Windows
WebGL (2011, Khronos Group)
Графический API для веб-браузеров
Mantle (2013, AMD)
Спецификация низкоуровневого API
Vulkan (2016, Khronos Group)
Низкоуровневый (требует явного управления памятью и ресурсами) высокопроизводительный кроссплатформенный API для работы с 2D и 3D графикой

MESA

MESA — открытый проект, который предоставляет реализацию различных графических API (таких как OpenGL, OpenGL ES, Vulkan, OpenCL и др.) для Linux и других операционных систем с открытым исходным кодом.
MESA поддерживает реализацию OpenGL для устройств, разработанных Intel, AMD, NVIDIA, Qualcomm и др., включая виртуальные графические процессоры VMware и VirGL. Также есть несколько программных рендереров: Softpipe (драйвер Gallium) и LLVMpipe (высокоскоростной растеризатор на основе LLVM/JIT).

MESA

MESA — открытый проект, который предоставляет реализацию различных графических API (таких как OpenGL, OpenGL ES, Vulkan, OpenCL и др.) для Linux и других операционных систем с открытым исходным кодом.
MESA поддерживает реализацию OpenGL для устройств, разработанных Intel, AMD, NVIDIA, Qualcomm и др., включая виртуальные графические процессоры VMware и VirGL. Также есть несколько программных рендереров: Softpipe (драйвер Gallium) и LLVMpipe (высокоскоростной растеризатор на основе LLVM/JIT).

Проект начал разрабатывать Брайн Пол в 1993. Основная цель заключалась в реализации OpenGL, подобной IRIS GL.
Первая версия вышла в 1995 с разрешения SGI. Названа в честь языка программирования MESA.
https://docs.mesa3d.org/history.html

Многоуровневая модель игровой системы

Игровой или графический движок

Составляет надстройку над графической библиотекой и включает:

- Модуль анимации (скелетная анимация, кинематика, блендеры анимаций).
- Модуль физики (динамика твёрдых тел, жидкости, столкновения, мягкие тела).
- Система игрового ИИ (поведение NPC, навигация, боты, скрипты).
- Аудиосистема (звук в реальном времени, 3D-пространство, микширование).
- Система I/O и скриптов (управление событиями, привязка скриптов, GUI).
- Сетевая подсистема (сетевая синхронизация, клиент-сервер, P2P).

Графическая библиотека (API)

- Низкоуровневый интерфейс для работы с GPU: создание контекста, буферов, текстур, шейдеров, конвейера рендеринга.
- Управление ресурсами (ресурсный менеджер), загрузка и компиляция шейдеров.
- Математические утилиты (векторы, матрицы) и простые обёртки для загрузки изображений и моделей.

Многоуровневая модель игровой системы
Игровой или графический движок
Составляет надстройку над графической библиотекой и включает:
Модуль анимации (скелетная анимация, кинематика, блендеры анимаций).
Модуль физики (динамика твёрдых тел, жидкости, столкновения, мягкие тела).
Система игрового ИИ (поведение NPC, навигация, боты, скрипты).
Аудиосистема (звук в реальном времени, 3D-пространство, микширование).
Система I/O и скриптов (управление событиями, привязка скриптов, GUI).
Сетевая подсистема (сетевая синхронизация, клиент-сервер, P2P).
Графическая библиотека (API)
Низкоуровневый интерфейс для работы с GPU: создание контекста, буферов, текстур, шейдеров, конвейера рендеринга.
Управление ресурсами (ресурсный менеджер), загрузка и компиляция шейдеров.
Математические утилиты (векторы, матрицы) и простые обёртки для загрузки изображений и моделей.

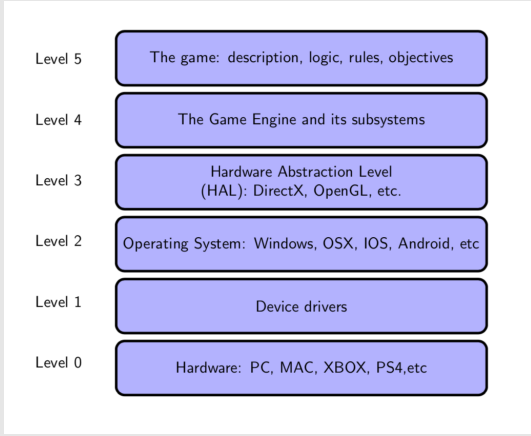


Рис. 1: Иерархическая структура игровой системы

Распределение вычислений между CPU и GPU

CPU (Central Processing Unit) и GPU (Graphical Processor Unit)

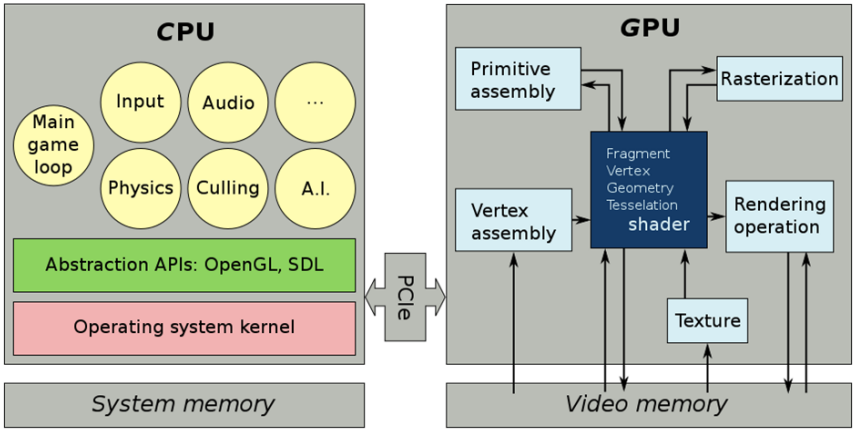


Рис. 2: Принципиальная схема распределения вычислений

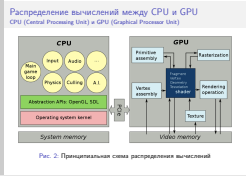


Рис. 3: Ghost of Tsushima (2024, Призрак Цусимы)

Растровая графика

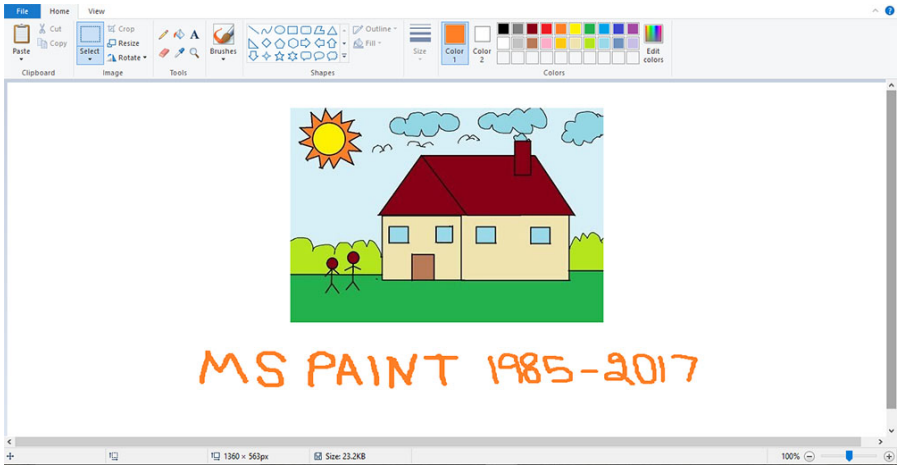
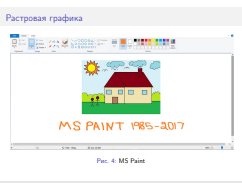


Рис. 4: MS Paint



Растровая графика (bitmap или pixel-based) представляет изображение в виде массива пикселей. Каждый пиксель хранит информацию о цвете, расположенные в определенном порядке формируют изображение.

Качество изображения зависит от количества пикселей. Чем выше разрешение (количество пикселей на единицу площади), тем детализированнее изображение.

Векторная графика

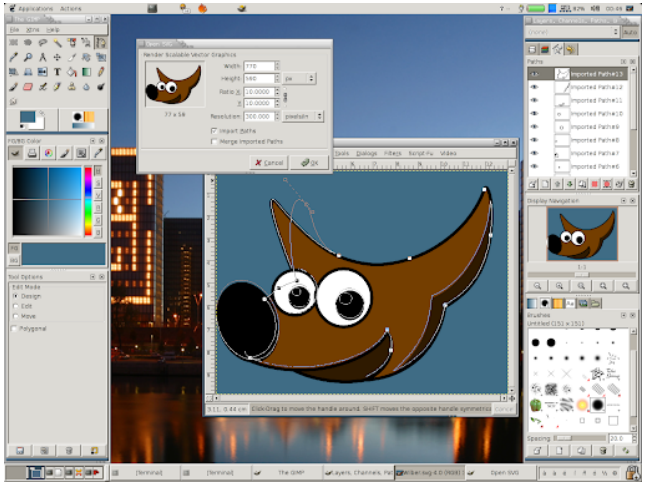


Рис. 5: GIMP

Векторная графика
Рис. 5: GIMP

Векторная графика представляет изображение, состоящее из примитивов и/или математических объектов (линии, точки, треугольники, кривые, поверхности и др.). Эти объекты описываются с помощью набора параметров и/или уравнений, что позволяет бесконечно масштабировать изображение без потери качества.

Воксельная графика

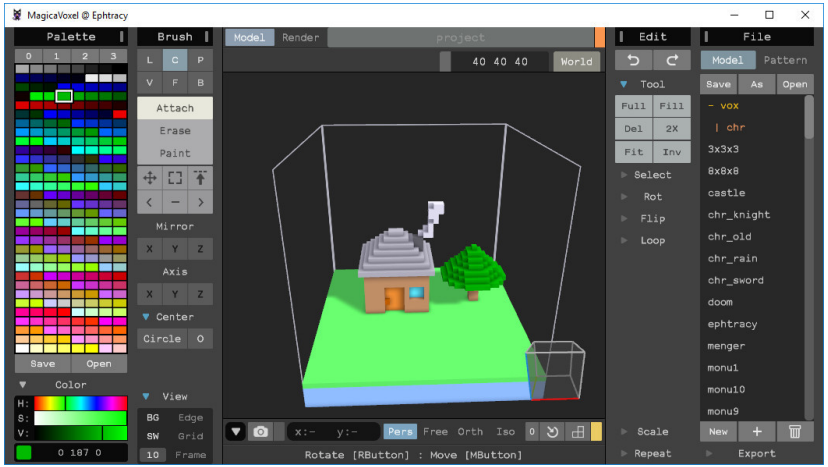
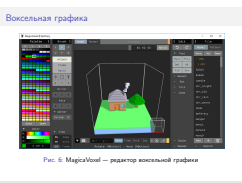


Рис. 6: MagicaVoxel — редактор воксельной графики



Воксельная графика — трехмерный аналог растровой графики, где изображение состоит из объёмных пикселей, называемых вокселями (volume elements, объемные элементы). Воксели представляют собой кубические блоки, которые формируют объёмные объекты в 3D-пространстве.

Gaussian Splatting (2023)

Рис. 7: Процесс построения изображения с помощью Gaussian Splatting

Gaussian Splatting (2023)

Рис. 7: Процесс построения изображения с помощью Gaussian Splatting

3D Gaussian (3D-гауссиан) — трехмерный объект ("эллипсоид"), описываемый следующей формулой:

$$G(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^3 \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right),$$

где μ — центр; $\Sigma = RSS^T R^T$ — ковариационная матрица (форма и ориентация); R — поворот; S — масштаб.

Splatting (Сплаттинг) — техника рендеринга, при которой 3D-примитивы проецируются на 2D-экран как "кляксы"с размытием.

Наложение полупрозрачных гауссианов (alpha blending) рассчитывается как:

$$c(r) = \sum_{i \in N} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j),$$

где r — луч взгляда; c_i — цвет i-го гауссиана рассчитывается с помощью сферических гармоник; $\sigma_i = \alpha_i \cdot G_i$; α_i — прозрачность i-го гауссиана; N — множество гауссианов в сцене.

Процесс построения изображения

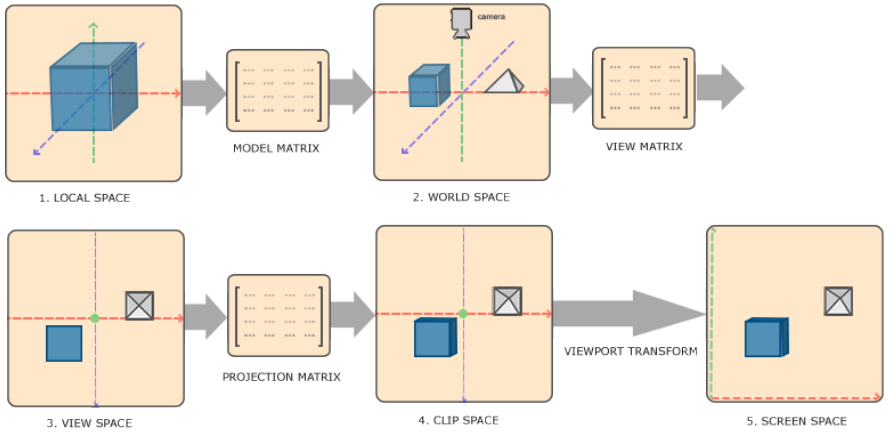
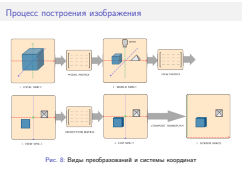


Рис. 8: Виды преобразований и системы координат



Локальные координаты (ЛК)
(координаты модели)

Геометрические преобразования

Мировые координаты (МК)
(координаты модели в сцене)

Преобразование наблюдения

Координаты наблюдения (КН)

Преобразование проецирования,
включая нормирование и отсечение

Нормированные координаты (НК)

Преобразование поля просмотра,
включая растеризацию

Координаты устройства (КУ)

Процесс построения изображения

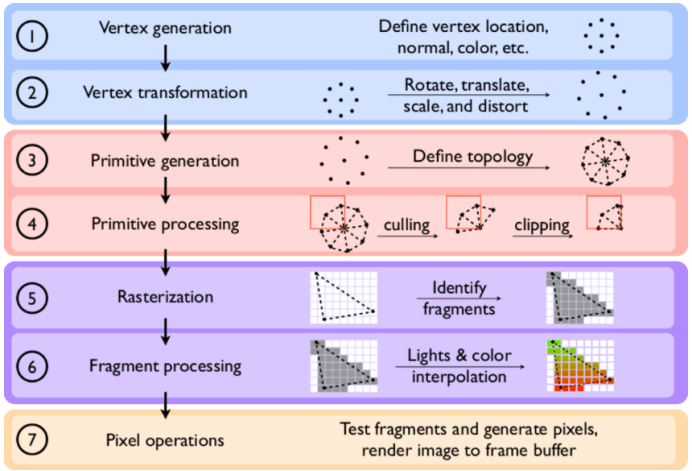


Рис. 9: Схема графического конвейера

Конвейер рисования в OpenGL

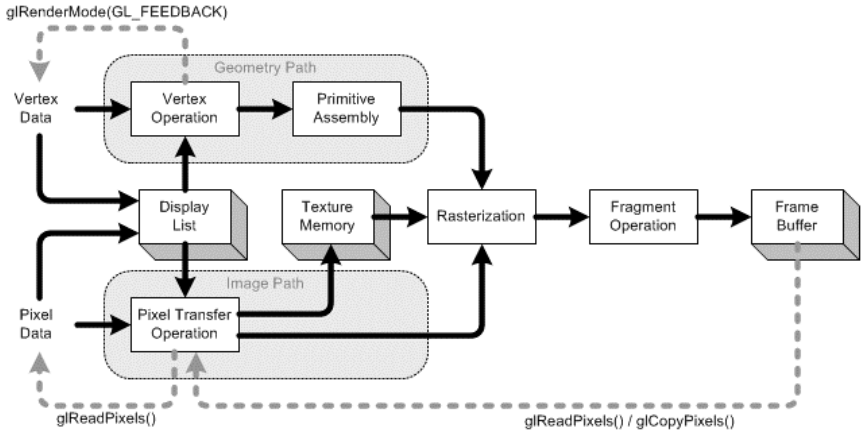
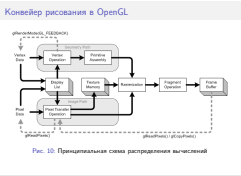


Рис. 10: Принципиальная схема распределения вычислений



Упрощенная схема графического конвейера

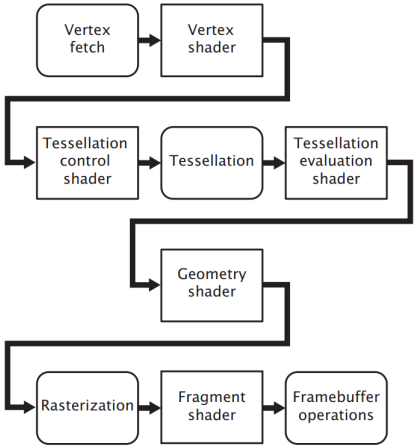


Рис. 11: Порядок вычисления шейдеров

Загрузка данных

Вершина (vertices)

Вершинный шейдер

Группа вершин (primitives/patches)

Шейдер управления тесселяцией

Тесселяция

Шейдер оценки тесселяции

Примитивы (primitives)

Геометрический шейдер

Примитивы (primitives)

Растеризация и интерполяция

Пиксели (fragments)

Пиксельный (фрагментный) шейдер

Пиксели (fragments)

Операции с буферами кадров

Пиксели (Pixels)

Рис. 12: DirectX 11 (2008): без тесселяции (слева) и с тесселяцией (справа)

Вершинный и пиксельный шейдеры:
DirectX 8.0, 2000; OpenGL 2.0, 2004
Геометрический шейдер:
DirectX 10, 2006; OpenGL 3.2, 2009
Шейдеры тесселяции:
DirectX 11, 2009; OpenGL 4.0, 2010

Упрощенная модель графического конвейера

OpenGL API

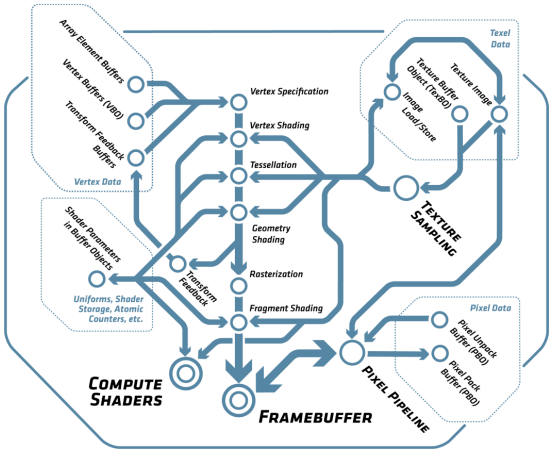


Рис. 13: Графический конвейер OpenGL (спецификация)

Упрощенная модель графического конвейера
OpenGL API

Рис. 13: Графический конвейер OpenGL (спецификация)

Упрощенная модель графического конвейера

Vulkan API

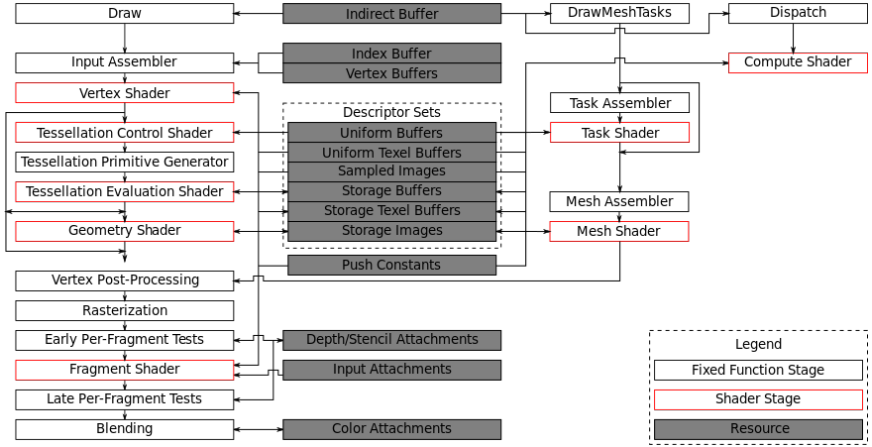


Рис. 14: Блок-схема конвейера Vulkan API