

СРАВНЕНИЕ ГРАФИЧЕСКОГО РЕНДЕРИНГА НА CPU, CUDA И RT-ЯДРАХ

В статье описывается понятие графического рендеринга, его техник, проводится сравнение рендера на CPU, CUDA и RT-ядрах.

ВВЕДЕНИЕ

С развитием технологий визуализации появляются всё более продвинутые способы рендеринга — от классического CPU до ускоренного CUDA и специализированных RT-ядер. Однако оборудование, способное использовать все эти подходы, остаётся дорогостоящим и не всегда доступным. В таких условиях особенно актуальным становится вопрос: действительно ли современные методы, такие как CUDA и RT-рендеринг, дают ощутимые преимущества по сравнению с CPU, и в каких задачах это оправдано? Сравнение этих трёх подходов помогает понять, насколько оправдана инвестиция в ту или иную технологию и в каких сценариях каждый метод показывает наилучший результат.

I. РЕНДЕРИНГ И ВИДЫ ТЕХНОЛОГИЙ, КОТОРЫЕ УЧАСТВУЮТ В ОБРАБОТКЕ СЦЕН

Рендеринг (визуализация) — это преобразование 3D-сцены в 2D-изображение с учётом параметров: геометрии объектов, освещения, камеры и материалов [1].

CPU-ядра — это мощные вычислительные единицы, выполняющие сложные логические, арифметические и управленческие операции. В рендеринге они используются для высокоточного оффлайн-рендеринга, где важна точность, например, в рендерах Arnold, Corona, V-Ray CPU, LuxCoreRender. Каждый пиксель или луч может обрабатываться отдельным потоком, что позволяет добиться фотореалистичного результата [2].

CUDA (Compute Unified Device Architecture) — это проприетарная архитектура параллельных вычислений от NVIDIA, позволяющая использовать ядра GPU для выполнения общих вычислений, в том числе задач рендеринга. В отличие от CPU, CUDA-ядра ориентированы на массивные параллельные потоки, где каждая задача, например, расчет цвета пикселя может выполняться независимо от других [3].

RT-ядра (Ray Tracing Cores) — это специализированные аппаратные блоки внутри GPU, например, NVIDIA RTX-серии, предназначенные исключительно для трассировки лучей. В отличие от универсальных CUDA-ядер, RT-ядра реализуют аппаратную поддержку основных операций трассировки — пересечений лучей с объектами сцены и построения иерархий ускорения [4].

II. РАБОТА РЕНДЕРИНГА НА CPU, CUDA И RT ЯДРАХ НА АППАРАТНОМ УРОВНЕ

Рендеринг на CPU использует последовательную или умеренно параллельную обработку, так как CPU ориентирован на выполнение универсальных инструкций. Несмотря на меньшее число ядер по сравнению с GPU, каждое из них мощнее и подходит для сложных задач, включая ветвления, работу с памятью и математические вычисления. Это делает CPU оптимальным для оффлайн-рендеринга в архитектуре и визуальных эффектах. Рендереры на CPU (Arnold, Corona, V-Ray CPU) обрабатывают сцену (геометрию, материалы, свет, камеры) в ОЗУ, затем ядра CPU выполняют трассировку лучей и расчёт освещённости для пикселей или тайлов. Эффективность достигается за счёт кэшей L1–L3 и переиспользования данных, особенно в глобальном освещении [2].

CUDA-рендеринг применяется в Octane Render, Redshift, Iray для трассировки лучей, освещения, теней, отражений и постобработки. CUDA задействует все ресурсы GPU: регистры, ALU, кэши и специализированные блоки. Это обеспечивает высокую производительность без потери качества, особенно в режиме реального времени. Алгоритм работы:

- хост (CPU) подготавливает данные сцены и отправляет их в память GPU;
- Запускаются CUDA-ядра, в которых каждый поток обрабатывает пиксель/треугольник/луч;
- CUDA-ядра выполняют операции: арифметика с векторами, пересечения лучей и примитивов, шейдинг, вычисление освещённости;
- результаты возвращаются в CPU или сохраняются на GPU для дальнейшей постобработки. В этом процессе CPU выступает только как координатор, а вычисления выполняются CUDA-ядрами. Также используются текстурные блоки и кэши для ускорения доступа к данным [5]. RT-ядра занимают исключительно трассировкой, освобождая CUDA-ядра для других задач. Это повышает общую эффективность. Алгоритм работы следующий:
- CPU готовит сцену и передаёт на GPU;
- CUDA-ядра запускают рендер-ядро и распределяют задачи;
- RT-ядра берут на себя трассировку: вычисляют пересечения лучей с геометрией, возвращают результат в CUDA/шейдерные блоки;
- CUDA-ядра и тензорные ядра выполняют шейдинг, постобработку, Denoising и так далее;
- результат отсылается обратно в память и отображается [6].

III. ПРАКТИЧЕСКОЕ СРАВНЕНИЕ РЕНДЕРИНГОВ

Для практического сравнения рендеринга использовалась программа Blender с загруженной в неё тяжёлой для обработки сцены с несколькими автомобилями. На моделях применены блестящие материалы на кузове, а на стёклах есть отражения. Свет в сцене выставлен так, чтобы отражения лучей затрагивали сразу обе машины и отражались между объектами. Тестирование для всех типов ядер производилось в одинаковых условиях с использованием рендера Cycles в разрешении 1920x1080 пикселей при количестве samples 1225. Для проведения тестов использовался персональный компьютер со следующими характеристиками:

- CPU — 13th Gen Intel(R) Core(TM) i7-13700KF;
- GPU — ASUS TUF Gaming GeForce RTX 4070 Ti 12GB;
- RAM — 32 GB DDR5 6400 МГц.

Итоги проведённого сравнения приведены в таблице 1.

Таблица 1 – Результаты сравнения рендерингов

| Тип ядер | Время рендеринга | Затраченная ОП |
|----------|------------------|----------------|
| CPU | 6 мин 28 сек | 174,32 Мб |
| CUDA | 37 сек | 626,42 Мб |
| RT | 21 сек | 550,61 Мб |

Как видно из результатов тестирования рендеринг на CPU самый долгий (в 10,5 раз дольше, чем на CUDA и RT ядрах), но при этом он затрачивает наименьшее количество оперативной памяти, чем другие рендеринги (3,6 раза меньше). Из всех методов рендеринга рендер на RT-ядрах оказался самым быстрым (в 18 раз быстрее CPU рендеринга и в 1,8 раза быстрее CUDA рендеринга) при этом он затрачивает меньше оперативной памяти чем CUDA рендеринг, но больше, чем CPU рендеринг (в 3,2 раза), однако скорость обработки явно окупается так как итоговый кадр выглядит одинаково на каждом виде рендеринга.

IV. ВЫВОДЫ

Результаты тестирования показали, что рендеринг на CPU занимает значительно больше времени, чем на CUDA и RT-ядрах, при минимальном использовании оперативной памяти. CUDA-рендеринг

Адамович Богдан Константинович, студент 3 курса кафедры вычислительных методов и программирования БГУИР, jemojem@yandex.ru

Научный руководитель: Кукин Дмитрий Петрович, Заведующий кафедры вычислительных методов и программирования БГУИР, kukin@bsuir.by.

обеспечивает заметный прирост скорости, но требует больше ресурсов, а наилучшую производительность продемонстрировали RT-ядра, сократив время рендера до минимума при умеренном потреблении памяти. Это подтверждает, что современные аппаратные ускорители действительно дают ощутимое преимущество в скорости и эффективности, особенно в ресурсоёмких сценах.

1. До конца загрузки осталось 19 часов: разбираем, что такое рендеринг? [Электронный ресурс]. – Режим доступа: <https://practicum.yandex.ru/blog/что-такое-рендеринг/> – Дата доступа: 10.04.2025.
2. Что такое ядро процессора и какие функции оно выполняет: подробное руководство [Электронный ресурс]. – Режим доступа: <https://mksegment.ru/a/что-такое-ядро-процессора-i-kakie-funkcii-ono-vypolnyaet?> – Дата доступа: 10.04.2025.
3. Best Hardware for GPU Rendering in Octane – Redshift – Vray [Электронный ресурс]. – Режим доступа: <https://www.cgdirector.com/best-hardware-for-gpu-rendering-in-octane-redshift-vray/> – Дата доступа: 10.04.2025.
4. Real-Time Ray Tracing [Электронный ресурс]. – Режим доступа: https://developer.nvidia.com/rtx/ray-tracing?sortBy=developer_learning_library%2FsortBy%2Ftitle%3Aasc – Дата доступа: 10.04.2025.
5. Ray Tracing on a GPU with CUDA – Comparative Study of Three Algorithms [Электронный ресурс]. – Режим доступа: <https://gowtham-maran.github.io/Ray%20Tracing%20on%20a%20GPU%20with%20CUDA.pdf> – Дата доступа: 10.04.2025.
6. RT-ядра (Ray Tracing Cores) [Электронный ресурс]. – Режим доступа: <https://dzen.ru/b/Z5mgYL0KJ3qyR-0p> – Дата доступа: 10.04.2025.