ТЕХНОЛОГИИ РЕНДЕРИНГА В РЕАЛЬНОМ ВРЕМЕНИ

Голенчук М.В. Дроздович В.И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Рябычина О.П. – к-т техн. наук, доцент

Аннотация. Рассмотрены особенности использования различных технологий рендеринга в реальном времени. Обоснована перспективность их использования в процессе обучения, рассмотрены их различия.

Рендеринг – это процесс преобразования трехмерных моделей или сцен в двумерные изображения на экране. Рендеринг в реальном времени требует от системы обрабатывать большие объемы данных за короткое время, обычно не менее 30 кадров в секунду. Это достигается за счет использования специализированных алгоритмов и аппаратного обеспечения [1].

Основные алгоритмы рендеринга:

Трассировка лучей (Ray Tracing) – это передовая технология 3D-графики, основанная на принципе отражения света. При рассмотрении окружающего мира, световые лучи отражаются от объектов и попадают в экран монитора, позволяя воспринять картину происходящего. Этот процесс включает такие явления, как отражение, преломление и поглощение, которые влияют на траекторию лучей [2].

Используя эти принципы, трассировка лучей моделирует поведение света в виртуальной среде, создавая высокодетализированное освещение и визуальные эффекты. Алгоритм анализирует точки соприкосновения лучей с объектами, учитывает свойства поверхностей и рассчитывает, как свет будет отражаться, рассеиваться, менять цвет и отбрасывать тени.

Одним из главных преимуществ этой технологии является независимость её сложности от геометрической сложности сцены. Также она позволяет одновременно обрабатывать множество лучей, распределяя их обработку между разными вычислительными ядрами.

Благодаря этому подходу трассировка лучей обеспечивает значительно более качественное изображение по сравнению с традиционной растеризацией. Для расчёта цвета в точке можно предоставить следующую упрощённую формулу:

$$Color(x, y) = \sum_{i=0}^n Light_i(x, y) \times Material(x, y) \quad (1),$$

где n – количество источников света, $Color(x, y)$ – показатель цвета в точке с координатами x и y , $Light_i(x, y)$ – показатель освещения в точке с координатами x и y , $Material(x, y)$ – показатель отображающий свойства материала в точке с координатами x и y .

В зависимости от полноты описания она может быть увеличена и изменена, учитывая диффузные свойства материала, спекулярные свойства, цветовую модель, функцию затенения по закону Ламберта и функцию затенения по модели Блинна-Фонга.

Растеризация – это процесс, лежащий в основе трёхмерной графики, где 3D-объекты, состоящие из множества точек и соединённых линий (многоугольников), преобразуются в 2D-проекцию на сетке пикселей экрана. Это превращение векторных объектов в пиксельное изображение позволяет компьютеру интерпретировать цвет, яркость и другие параметры каждого пикселя, что упрощает отображение изображения на цифровых устройствах.

Для этого метода используются математические алгоритмы, которые рассчитывают расположение пикселей на сетке и интерполируют цвета между ними. Растеризация включает преобразование координат вершин примитивов из системы координат модели в систему координат экрана. Обычно это происходит в вершинном шейдере, но при необходимости также могут применяться шейдеры тесселяции и геометрические шейдеры.

После этого происходит разложение примитивов на фрагменты, для которых выполняются пиксельные шейдеры, определяющие цвет каждого фрагмента. Этот процесс заканчивается отображением цветов в кадровый буфер.

Растеризация позволяет эффективно визуализировать изображения и изменять их, особенно при работе с текстурами и эффектами, такими как затенение. Однако данный подход не позволяет точно просчитывать освещение с физической точки зрения.

Для решения этой проблемы применяются приближительные модели освещения, такие как модель Фонга, модель Блинна-Фонга, а также такие техники, как карты теней, кубические карты отражений и глобальное освещение экранного пространства. Эти методы работают только с информацией в кадровом и историческом буфере, что ограничивает их точность при обработке объектов, находящихся за пределами прямой видимости.

Хотя растеризация значительно быстрее и эффективнее трассировки лучей, она не может воспроизводить эффекты, связанные с физическим поведением света, например реалистичные отражения и прозрачность.

Этот компромисс между скоростью и качеством делает растеризацию популярной в системах автоматизированного проектирования, графических приложениях и компьютерных играх, где нужна интерактивная графика.

Формула для растеризации, преобразующая трехмерные координаты в двумерные пиксели на экране, включает в себя несколько шагов, включая проекционное преобразование и перспективное деление. Простая формула для перспективного преобразования в трехмерном пространстве может быть представлена следующим образом:

$$x' = \frac{x}{1+rz}, y' = \frac{y}{1+rz} \quad (2),$$

где x' и y' — координаты проектированной точки на плоскости $z = 0$, x и y — исходные координаты точки в трехмерном пространстве, z — координата точки вдоль оси z , r — параметр перспективы, определяемый расстоянием до центра проекции.

Эта формула показывает, как координаты точки меняются при перспективном преобразовании, в зависимости от координаты z и параметра перспективы. Эта формула используется для создания иллюзии глубины в двумерных изображениях, что особенно важно в компьютерной графике и фотографии.

Для достижения высокоскоростного рендеринга используются различные математические техники:

Интерполяция: используется для плавной анимации между кадрами. Пример уравнения для интерполяции двух точек траектории:

$$P_t = P_1 + t(P_2 - P_1) \quad (3),$$

где $t \in [0; 1]$, P_1 и P_2 — точки, между которыми происходит интерполяция, параметр t задает положение интерполированного значения между ними [3].

Локальное пространство: применяется для оптимизации расчетов освещения.

Локальное пространство определяется как система координат относительно поверхности объекта:

$$TBN = \{ T \& B \& N \}^T \quad (4),$$

где T , B , N представляют собой вектора: касательный (T), бинормальный (B) и нормальный (N), которые вместе образуют базис, описывающий ориентацию поверхности в данной точке [4].

Графические процессоры (GPU) играют решающую роль при выполнении задач по рендерингу благодаря своей параллельной архитектуре:

Используют шейдера: программы на языках HLSL или GLSL для GPU.

Поддерживают трассировку лучей: NVIDIA RTX серия поддерживает эту функцию напрямую через аппаратное обеспечение.

Технологии рендеринга в реальном времени являются важнейшим элементом современных интерактивных систем. Они требуют применения сложных математических моделей вместе с мощным аппаратным обеспечением для достижения высокого качества изображений без потери производительности.

В будущем развитие этих технологий будет направлено на дальнейшее улучшение качества графики за счет новых алгоритмов обработки данных на GPU, что позволит еще больше приблизить пользовательский опыт к идеалу полноценной иммерсивности во всех видах интерактивных медиа.

Список использованных источников:

1. *What is real time rendering? A quick & comprehensive overview* [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://applet3d.com/3d-rendering/what-is-real-time-rendering/>
2. *How 3D Game Rendering Works, A Deeper Dive: Rasterization and Ray Tracing* [Электронный ресурс]. — Электронные данные. — Режим доступа: [How 3D Game Rendering Works, A Deeper Dive: Rasterization and Ray Tracing | TechSpot](https://www.techspot.com/resources/2/articles/how-3d-game-rendering-works-a-deeper-dive-rasterization-and-ray-tracing/)
3. *Линейная интерполяция и кривая Безье* [Электронный ресурс]. — Электронные данные. — Режим доступа: [Линейная интерполяция и кривая Безье | Блог Евгения Жирнова](https://www.evgenezhirnov.com/ru/linear-interpolation-and-bezier-curve/)
4. *Электронный учебник по компьютерной графике* [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://stratum.ac.ru/education/textbooks/kgrafic/lection04.html?ysclid=m8n1gi7vpz38110306>