

## Ray Tracing, як найбільш актуальна технологія рендеру

Олексій Копча, студент. Денис Лукенів, студент. Катерина Литвиненко, студентка. Світлана Цюцюра, д.т.н., професор, завідувач кафедри ІТ

Київський національний університет будівництва і архітектури, Київ, Україна

### АНОТАЦІЯ

Цей документ розповідає особливості технології Ray Tracing, засоби використання та різницю у порівнянні з використанням лише растеризації.

Ключові слова: растеризація, Ray Tracing, Path Tracing, комп'ютерна графіка, відеоігри.

### 1. ВСТУП

Графіка відеоігор розвивалася протягом багатьох років від ранньої появи пікселів і векторної графіки в 70-х роках до моменту, коли вони перестали бути пізнаваними. Залишилося одне – растеризація. Це рішення, яке виграло в поодиночці. Renderman від Pixar, заснований практично на всіх методах комп'ютерної графіки в реальному часі і сформував велику частину офлайн-рендеринга. Коли ми перейшли на 3D-рендеринг, тепер у нас є 3 осі для глибини або висоти. Це значно ускладнило цей процес, але растеризація була збережена на цьому етапі.

Очевидно, що в нього були внесені поліпшення, включаючи поділ зображення на фрагменти або плитки, щоб кожен піксель міг обчислюватися і зафарбовувати паралельно через ядра графічного процесора. Приблизно так малюється стандартна рамка цим методом.

### 2. АНАЛІЗ ІСНУЮЧИХ ПРОЦЕСІВ РАСТЕРИЗАЦІЯ

Растеризація - це процес проєкції 3D-моделей на 2D-площину для відображення на екрані комп'ютера. Однак, цей процес часто виконується апаратним забезпеченням з фіксованими функціями в графічному конвеєрі (рис. 1).



Рисунок 1. Загальний конвеєр рендеринга графіки OpenGL.

Цей конвеєр реалізований OpenGL, графічним драйвером, графічним обладнанням. Основна логіка для цього конвеєра об'єктно-орієнтована, що вимагає обходу об'єктів, перетворення і подальшого малювання кожного з них, якщо z-буфер не повідомляє, що він не попереду.

З цього конвеєра ми бачимо, що будь-яке зображення, яке ви бачите на екрані комп'ютера, починається як растрове або векторне зображення. Растрове зображення складається з набору затінених пікселів.

Головна перевага растеризації – її швидкість, особливо в порівнянні з трасуванням променів. Графічний процесор скаже грі створити тривимірне зображення з невеликих форм, найчастіше трикутників. Ці трикутники перетворюються в окремі пікселі, а потім пропускаються

через шейдер для створення зображення. Його можна використовувати як рендеринг в реальному часі.

Однак, растеризація – це просто процес обчислення зіставлення від геометрії сцени до пікселів і не наказує конкретний спосіб обчислення кольору цих пікселів. Таким чином, він не може враховувати затінення, особливо фізичний світ, і не може обіцяти отримати фотореалістичний результат. Це велике обмеження растеризації.

### 3. ТЕХНОЛОГІЯ ТРИВИМІРНОЇ ГРАФІКИ - RAY TRACING

Ray Tracing (трасування променів) – це технологія тривимірної графіки, яку дуже просто пояснити, просто оглянувшись навколо. Ми бачимо відбиті від об'єктів промені світла, які ми спостерігаємо нашими очима. Коли ми дивимось на предмети реального світу, ми бачимо світло відбите від них, що є променем світла. Промінь світла – це лінія, вздовж якої переноситься світлова енергія, промені починають свій рух від джерела світла, а після взаємодії з об'єктами навколишнього світу. Серед таких взаємодій можна виділити: відбивання, переломлення, поглинання. Всі вони впливають на траєкторію променів світла, котрі попадають на сітківку нашого ока і ми отримуємо картину того, що відбувається біля нас. Тобто, можна сказати, що ми бачимо відбите світло.

Знаючи це, можна розрахувати траєкторію та імітувати це світло в тривимірній графіці, що і є Ray Tracing.

Трасування променів є технологією рендеринга тривимірної графіки, де використовується ось цей принцип і дає змогу створювати неймовірно реалістичне освітлення, що практично схоже на реальне. Алгоритм бере до уваги де саме промінь світла доторкається до об'єкта, враховує властивості поверхні і вираховує як промінь буде розсіюватись, тобто, де буде відбиватись, де змінить колір, а де відкине тінь.

В переваги трасування входить те, що складність метода мало залежить від геометричної складності сцени, а розрахунки пролягають і можна легко трасувати декілька променів одночасно, розділяючи поверхність екрана на зони для їх трасування на різних обчислювальних ядрах.

Важливо, що цей імітує реальне розповсюдження променів світла, отримуючи картинку більш високої якості, у порівнянні з растеризуванням. У растеризації є явні недоліки. Наприклад, об'єкт, що не входить в сцену, не буде вимальовуватись на GPU, але він може відкидати видимі тінь і повинен відобразитись в поверхні, що відбиває

картинку (дзеркало), а оптимізація растеризації його відкинула і не приймала до уваги.

Такі ефекти як відбивання, переломлення і тіні досить складні для якісної реалізації при растеризації, являється результатом роботи трасування променів. Візьмемо, для прикладу, відображення – це всього одна з областей, в яких метод трасування променів помітно кращий растеризації.

Розрахунок відображень в екранному просторі дозволяє отримати більш-менш схожі відображення при деяких умовах, але при апаратному прискоренні GPU з використанням растеризації. А при трасуванні променів відображення завжди ідеальне без необхідності в додаткових складних алгоритмах. Ще одною важливою перевагою трасування є вивід відображення частин одного і того ж об'єкту один на одному, що важко зробити за допомогою растеризації.

Ще одним прикладом явної переваги трасування є рендеринг прозорих об'єктів. За допомогою растеризування дуже складно імітувати ефекти прозорості, так як її розрахунок залежить від порядку рендеру і для цього доводиться сортувати прозорі полігони. А алгоритм трасування променів сам дозволяє намалювати будь-які ефекти прозорості з ідеальною якістю.

Також певною перевагою трасування над растеризацію є малювання тіней. При растеризації використовуються карти тіней (shadow maping), які також основані на растеризації, тільки рендеринг робиться з другої точки сцени з іншими параметрами В свою чергу, трасування дозволяє вирішити проблему тіней автоматично, без потреби додаткових алгоритмів і об'єктів пам'яті, і отримати реалістичні м'які тіні. 4]

#### 4. РІЗНИЦЯ МІЖ RAY TRACING ТА PATH TRACING

У чому Path Tracing відрізняється від звичайного Ray Tracing? Для цього потрібно подивитись у сутність проблеми з продуктивністю рейтрейсингу. Річ у тім, що коли надсилається 10 променів, вони відбиваються у 100 променів, тим часом ті 1000 променів відбиваються у 10000 і так далі. І якщо для рендеру кадру для фільму або мультфільму проблеми зі швидкістю рендера не є проблемою, то при оброчі в теперішньому часі (ігри, попередній перегляд результату рендерінгу) є проблемою. Звичайно коли ми говоримо про рейтрейсинг у теперішньому часі то ми розуміємо що накладаються обмеження щодо кількості променів, обмеження на кількість відскоків і тд. Якраз у такій ситуації Path Tracing показує найбільшу ефективність. Він використовує алгоритм по якому вибирає один випадковий промінь, по цьому променю дораховує інші промені й так зменшує навантаження на систему. Не зважаючи на те, що Path Tracing створений за для зменшення навантаження, програми які його використовують зараз є найбільш важкими. Вони накладають набагато менше обмежень на Path Tracing ніж на Ray Tracing, у наслідок чого менш прожерливий алгоритм споживає більше, але і видає набагато більш реалістичну картинку. Як приклад можна навести лише ігри: Quake 2, Minescraft RTX. Можна побачити що він використовується лише в дуже невимогливих іграх.

Як працює глобальне освітлення за допомогою Ray Tracing RTXGI? Якщо не вдаватися у великі деталі то воно модифікує технології Probes. З новими технологіями Probes

посилають промені, завдяки променям вони точно розуміють відстань до об'єктів, яке випромінювання зберігати та розуміють інформацію від променів інших Probes, також розуміючи інформацію про об'єкти поруч змінює світло, урахувавши матеріали з яких зроблені об'єкти та як це буде впливати на світло, як у наслідок виходить більш м'яким зображення, з правильними тінями, і правильним переходом від зовнішнього світу до звичайних приміщень. Як приклад можна використати гру Metro Exodus.

В цій грі можна побачити головну перевагу: правильне відбивання світла на інші об'єкти. Тобто, один об'єкт може впливати на забарвлення сусідніх, додає відображення свого кольору на інші поверхні. Це і є наслідком можливості зв'язку Probes один з одним. У випадку растеризованої графіки Probe1 не знає всю інформацію про Probe2, та не використовує для зміни освітлення матеріали поруч себе, як наслідок ми не можемо побачити відскоків світла від об'єктів на інші об'єкти як кінцевий результат.

Чи сильно впливають на результат різні методи по зменшенню навантаження на систему? Відносно сильно, річ у тому що чим менше ми відправляємо промені, та чим менше вони відбиваються, тим менше ми отримуємо інформації та через надзвичайно низьку кількість променів які посилаються через рейдер у реальному часі та можна побачити ефект шуму, але зараз є багато різних методів цей шум зменшити, вони називаються Denoiser. Зараз їх використовують на базі Sampling, Flutering Technique та машинного навчання, останнє з часом буда отримувати більше поширення у зв'язку з тим що усі карти які підтримують Ray Tracing у ріалтаймі мають блок для швидкої роботи зі штучним інтелектом та машинним навчанням.

#### 5. ВИСНОВКИ

Чи будуть у найближчому часі використовувати лише Ray Tracing замість звичної растеризованої графіки? Якщо ми кажемо про ріалтайм то така можливість з'явиться дуже не скоро, використання Ray Tracing global illumination, reflection, AO, Shades, дуже витратне навіть при використанні одного або декількох елементів зі списку, тому про повну заміну растеризованої графіки наразі не йдеться. У комп'ютерних іграх трасування променів в режимі реального часу з прийнятним рівнем швидкодії можлива за умови її апаратної підтримки відеокартою.

#### Список літератури

- [1] Растеризація. (2018, Вересень). У Вікіпедії. <https://en.wikipedia.org/wiki/Rasterisation>
- [2] Глобальне освітлення RTX. У NVidia Developer. <https://developer.nvidia.com/rtxgi>
- [3] Патрік Херн. (2019, Липень). Що таке трасування шляху і трасування променів? І чому вони покращують графіку? <https://clck.ru/RgHM4>
- [4] Олексій Берилло. (2018, Липень). У Канобу. <https://kanobu.ru/articles/revolyutsiya-v-grafike-hto-takoe-trassirovka-luchej-372475/>
- [5] Ален Гальван. (2020, Жовтень). Зниження шуму трасування променів. <https://clck.ru/Rgpp>