

УДК 004.31

ОПТИМІЗАЦІЯ АЛГОРИТМУ ПЛИТКОВОГО РЕНДЕРІНГУ І ТІНЬОВИХ КАРТ

Донець Д. С.

Науковий керівник – ст. викл. Новіков Ю. С.

Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна

e-mail: dmytro.donets@nure.ua

This work is devoted to improving on existing methods of generating shadows performance-wise and enhancing visual quality of the output image created by real time 3D graphics engines (such as game engines). In this example the implementation is based on one of the most robust and powerful game engines, Unreal Engine 5. This example will illustrate minor benefits which are done by the algorithm tailored for specific project's needs, which are useful in case of not very high total polygon amount per scene. The visual output might not be the desired result in all cases, but it might be applied in many scenarios.

За основу даного методу було взято метод рендерінгу плиткової відкладеної тіні та використання тіньових карт. Цей метод включає 5 етапів виконання, а саме підготовку, заповнення, зваження, складання та перенесення.

Під час підготовки на процесорі обчислюється новий вид камери, зріз якої обмежений найменшим полем зору між межею персонажа та межею світла. Отримана додаткова тіньова карта займає 2048x2048 пікселів, але кожна тінь у результаті зменшує роздільну здатність на основі оцінки покриття екрану світлом. Також очищуються попередні тіньові карти на відеокарті на початку кожної відкладеної тіні.

У заповненні алгоритм уникає рендерінгу меша персонажів більше ніж один раз, перенаправляючи виклики рисування від стандартного буферу до додаткової тіньової карти. Після цього зважується ієрархічний буфер глибини та генерується карта тіней з міпмапою, зберігаючи найближчу відстань до світла у кожних 2×2 пікселях.

На етапі складання на процесорі буде створено об'єм тіні персонажа, простягаючи промені від джерела світла до позицій кожного обмеження, доки не буде досягнуто радіус світла. Після проєкції в екранних координатах алгоритм знаходить мінімальну та максимальну позиції для запуску обчислювальних робочих одиниць, обмежених задіяними тіньовими плитками. У обчислювальному шейдері кожна робоча одиниця шукає відповідний світловий індекс у своєму записі буфера фінального списку світла. Якщо його не знайдено або позначено як повністю затінений, то алгоритм припиняє роботу. В іншому випадку оцінюється видимість світла кожного пікселя, і якщо вся плитка тіні повністю

освітлена, алгоритм зупиняється на цьому. Коли виявляється часткова тінь, гарантовано, що виділено слот видимості та збережено нову зменшену видимість світла.

У перенесенні кожна високоякісна тінь обробляється окремо з повторним використанням спільної цілі візуалізації результату алгоритму. Це має перевагу в тому, що вартість обчислень є єдиним обмежуючим фактором, оскільки накладні витрати пам'яті фіксовані. Однак вміст тіньової карти відкидається між кожною парою символ/світло, що робить його недоступним для прямого рендерінга. Оскільки персонажі з високоякісними тінями не малюються в них, кожен результат алгоритму переноситься назад у пов'язану стандартну тіньову карту. Це досягається шляхом проектування квадрата у тіньовій карті у ближній площині в стандартні координати тіньової карти світла. Один стандартний піксель тіньової карти містить багато пікселів результату алгоритму, і зчитування кожного з них було б непомірно дорогим. Ця проблема з високою пропускнуою здатністю вирішується за допомогою версії алгоритму із міпмапою, де кожен піксель вибирає рівень міпмапи, якому потрібні дві вибірки для покриття найменшого розміру XY області. Потім виконується повторне зчитування значень результату алгоритму з міпмапою, доки не буде покрито всю область. Наприклад, якщо піксель проектується в область розміром 12 x 34 пікселі у результаті алгоритму, то буде обрано 3 рівень міпмапи.

На таблиці 1 відображено результати оптимізації у порівнянні з тінями, вбудованими в UE5. Такі тіні потребують лише на 55% часу більше, ніж без тіней і на 7% менше часу за Epic тіні, при цьому не втрачають у їх якості у випадку проектів з малою кількістю полігонів на одній сцені.

Таблиця 1 – Часу рендерінгу одного кадру алгоритму та без нього

	Час (мс)	Різниця
UE5 без тіней	3.29	
UE5 з Epic тінями	5.48	+66%
UE5 з тінями алгоритму	5.12	+55%

У результаті даної роботи було створено оптимізаційний алгоритм для рендерінгу кадрів у 3D додатках реального часу і протиставлено результат ігровому рушію UE5.

Список використаних джерел:

1. Bart W. Cull that cone! Improved cone/spotlight visibility tests for tiled and clustered lighting. Apr. 2017. url: <https://bartwronski.com/2017/04/13/cull-that-cone> (дата звернення: 23.02.2024).

2. Nathan R. Depth Precision Visualized. NVidia. July 2015. url: <https://developer.nvidia.com/content/depth-precision-visualized> (дата звернення: 19.02.2024).