

УДК 004.92

<https://doi.org/10.17721/1812-5409.2022/4.9>

Пікульський Р. М.<sup>1</sup>, аспірант,  
Кулябко П. П.<sup>2</sup>, к.ф.-м.н., доц.

R. M. Pikulsky<sup>1</sup>,  
P. P. Kuliabko<sup>2</sup>, Dr. Sci.

**Стан та перспективи подальших  
досліджень сфери обчислення  
глобального освітлення у реальному  
часі**

**Real-time global illumination computation  
status and further research prospects**

<sup>1,2</sup> Київський національний університет імені  
Тараса Шевченка, 83000, м. Київ, пр-т.  
Глушкова 4д,  
e-mail: <sup>1</sup>rostyslav.pikulsky@gmail.com,  
<sup>2</sup>kpp1@ukr.net

<sup>1,2</sup> Taras Shevchenko National University of Kyiv,  
83000, Kyiv, Glushkova st., 4d,  
e-mail: <sup>1</sup>rostyslav.pikulsky@gmail.com,  
<sup>2</sup>kpp1@ukr.net

*У статті розглядається така проблема сучасної комп'ютерної графіки як обчислення глобального освітлення в реальному часі. Глобальне освітлення є невід'ємною частиною фотореалістичного рендерінгу, але його підрахунок потребує доволі об'ємних обчислень. Через це на даний момент якісне глобальне освітлення існує лише у неінтерактивних рендерах (наприклад, у мультиплікаційних фільмах), а у реальному часі (наприклад, комп'ютерному моделюванні або симуляціях, комп'ютерних іграх) зазвичай використовуються певні наближення, які хоч і надають зображенню певну природність, але все одно мають вкрай помітні неточності. Але останнім часом дана тема набуває все більшого розвитку за рахунок удосконалення відеопроцесорів. Крім значного підвищення їх швидкодії та збільшення кількості ядер досить велику роль грає поява апаратного прискорення трасування променів. В даній роботі проводиться теоретичне дослідження проблеми глобального освітлення, наводяться існуючі підходи та розробки для вирішення даної проблеми та аналізуються перспективи подальших досліджень та розробки нових методів обчислення глобального освітлення в реальному часі з урахуванням новітніх апаратних можливостей обчислювальної техніки.*

*Ключові слова: комп'ютерна графіка, фотореалістичний рендерінг, рендерінг у реальному часі, глобальне освітлення, трасування променів, метод стохастичного трасування променів.*

*Currently, computer graphics is a very important part of computer science. Graphics-related developments have been used in many different situations, for example, in animated and cinema movie productions, in computer graphics applications, modeling, and simulation systems, for different visualizations in medicine, mathematics, physics, etc. One of the main problems of computer graphics is the task of transforming the information of some imaginary scene and its observer into a photorealistic image of this scene for them. Solving this problem is very important, but right now obtaining a good quality result is possible only in a non-interactive scenario (for example, in animated films), while in real-time (for example, in computer modeling or simulations, in computer games) it is usually necessary to use some approximate algorithms. Although these algorithms are often able to provide a natural-looking result, they still have plenty of very noticeable inaccuracies. However, this topic is gaining more and more development recently due to the improvement of graphics processors. In addition to a significant increase in computation speed and the number of cores, the appearance of ray tracing hardware acceleration plays a large role.*

*Global illumination computation is an inseparable part of photorealistic image generation. This paper is focused on solving this problem in real-time, which means developing a system capable of generating such images at a speed sufficient for the resulting sequence to be perceived by a person as a smooth animation. We give the theoretical information required for understanding this problem and describe existing methods and algorithms for solving it with their advantages and disadvantages. Also based on an overview of the topic's current state, we analyze further research prospects and directions for improving existing and developing new methods of real-time global illumination calculation, while considering compute power and technologies of the latest graphics hardware.*

*Keywords: computer graphics, photorealistic rendering, real-time rendering, global illumination, ray tracing, stochastic ray tracing method.*

## Вступ

На сьогоднішній день комп'ютерна графіка – дуже важлива область комп'ютерних наук. Рендерінг фотореалістичних зображень є вкрай необхідною частиною багатьох програмних пакетів, починаючи з програм моделювання та комп'ютерних симуляцій і закінчуючи комп'ютерними іграми. При цьому часто дуже важливо вміти генерувати фотореалістичне зображення динамічного середовища в інтерактивному режимі – щоб користувач міг споглядати анімовану демонстраційну сцену з різних позицій та напрямків.

Невід'ємною частиною фотореалістичності зображення є глобальне освітлення. Це метод освітлення, коли крім освітлення безпосередньо з джерел світла враховується також світло, відбите або розсіяне певними поверхнями. Для того, щоб оцінити важливість глобального освітлення, можна уявити світлу кімнату з вікном, у яке світить сонце. Якщо враховувати лише пряме світло, то освітленою у кімнаті буде лише невелика частина поверхні, що не закрита від сонця стінами або стелею, а всі інші поверхні будуть абсолютно чорними; натомість, при глобальному освітленні завдяки відбиванню світла від стін уся поверхня кімнати буде відбивати певну частину світла у точку спостереження (як це і буває в реальному світі).

Таким чином, розробка методів глобального освітлення у реальному часі є досить актуальною.

## 1. Теоретичні відомості

Глобальне освітлення [15] – це загальна назва для технік освітлення, які враховують не тільки пряме світло від джерела, але і відбите і розсіяне світло від різних поверхонь і протяжних джерел світла.

Для того, щоб обчислити освітленість точки при спостереженні її з певного напрямку, використовується так зване рівняння рендерінгу:

$$L(\omega_0) = L_0(\omega_0) + \int_{\Omega} f(\omega_0, \omega) L_i(\omega) (\omega \cdot n) d\omega,$$

де  $\omega_0$  – напрям променя від точки до спостерігача,  $L_0(\omega_0)$  – освітлення, що випромінює точка у напрямку  $\omega_0$ ,  $\Omega$  – півсфера, орієнтована вздовж нормалі поверхні  $n$  у точці,  $f(\omega_0, \omega)$  – двонаправлена функція відбивної здатності (bidirectional reflectance distribution function, BRDF),  $L_i(\omega)$  – освітлення, що надходить до

точки з напрямку  $\omega$ ,  $\cdot$  – скалярний добуток векторів.

Двонаправлена функція відбивної здатності – це функція чотирьох дійсних змінних, яка визначає, як світло відбивається на непрозорій поверхні. Функція приймає вихідний вектор  $\omega_0$  та вхідний вектор  $\omega$ , взяті в системі координат, де нормаль поверхні  $n$  лежить вздовж осі  $Z$  і повертає відношення відбитого світла, що виходить уздовж  $\omega_0$ , до опромінення поверхні з напрямку  $\omega$ . Кожен напрямок сам по собі параметризується азимутальним кутом  $\varphi$  та зенітним кутом  $\theta$ , тому BRDF в цілому є функцією 4 змінних. BRDF має одиниці  $sr^{-1}$ , при цьомустерадіан ( $sr$ ) є одиницею виміру тілесного кута.

Зазвичай принцип відбивання світла від поверхні можна поділити на два типи: дифузне та дзеркальне. Дифузне відбивання – це ситуація, коли світло, що попадає на поверхню, практично рівномірно відбивається у напрямках, що належать напівсфері, орієнтованій вздовж нормалі поверхні. Натомість у випадку дзеркального відбиття світло в основному відбивається у напрямку, дзеркальному до напрямку падіння. Практично всі непрозорі поверхні відбивають світло за такими законами або певним чином комбінують ці типи відбивання, і цей факт використовується при дизайні алгоритмів обчислення глобального освітлення.

Найбільшою складністю у розв'язанні рівняння рендерінгу є обчислення інтегралу на півсфері. Якщо враховувати лише пряме освітлення, цей інтеграл можна перетворити на суму на множині джерел світла (бо освітленість з інших напрямків вважається рівною нулю), але при глобальному освітленні потрібно враховувати відбите та розсіяне світло, яке фактично може надходити з будь-якого напрямку. Отже, для кожного напрямку з точки з деякою дискретизацією ми маємо визначити поверхню, що випромінює світло на поточну точку в даному, обчислити значення підінтегральної функції, та просумувати всі результати. Ясна річ, це потребує великої кількості обчислень.

## 2. Трасування променів

Для обчислення глобального освітлення так чи інакше використовуються алгоритми трасування променів.

В широкому розумінні, трасування променів – це знаходження перетину заданого променя у тривимірному просторі з деяким об'єктом заданої сцени. Якщо промінь перетинається з об'єктами сцени у декількох місцях, зазвичай мається на увазі найближчий до вершини променя перетин. Під представленням сцени та її об'єктів мається на увазі повна інформація про розташування, форму та оптичні властивості поверхні об'єктів сцени. Формат такого представлення може бути досить різноманітним, і на його основі будуються конкретні алгоритми трасування променів. Разом із форматами точного представлення сцени, існують також формати наближеного представлення для яких неможливо провести точне трасування променів. Перевагами таких форматів зазвичай є зменшення обсягу необхідної пам'яті для зберігання сцени та/або збільшена швидкодія алгоритмів трасування променів – через це такі формати можуть стати в нагоді у випадках, коли їх точність є задовільною.

Класичним методом представлення форми поверхні є її апроксимація у вигляді сітки трикутників. Вона не є точним представленням, але точність можна регулювати шляхом збільшення або зменшення кількості трикутників, таким чином можна досягти задовільного компромісу. Сама сітка також може бути представлена по-різному; найпростішим представленням є звичайна неупорядкована множина трикутників, але існують також інші структури, зокрема такі, що добре пристосовані саме для трасування променів, наприклад, BSP-дерево (binary space partitioning tree), octree (octal tree), BVH (bounding volume hierarchy).

Крім сітки трикутників, існують також інші представлення поверхні, наприклад, сплайни, хмари точок (point clouds), воксельні структури.

Властивості поверхні зазвичай поділяються на рівномірні та нерівномірні. Рівномірні – це такі, що є доволі сталими для геометричних форм і їх не має сенсу розрізняти в межах одного об'єкту (в разі їх різноманітності у межах деякого об'єкту можна просто розділити об'єкт на декілька), вони можуть задаватися як мета-інформація об'єкта. Зазвичай до них відносять лише BRDF поверхні, хоча в деяких випадках певні нерівномірні властивості також можна вважати рівномірними в межах певної сцени. Нерівномірні – такі, що можуть змінюватися в межах об'єкта (наприклад, колір поверхні, гладкість тощо), їх формат залежить від формату представлення форми поверхні. Для сіток трикутників зазвичай використовуються текстури – двовимірні

карти/зображення, куди заносяться конкретні значення властивостей; вершини сітки при цьому містять спеціальні двовимірні координати, за якими відбувається семплінг значень з текстур.

### 3. Стохастичний алгоритм обчислення відбитого світла

Для створення фотореалістичного зображення необхідно на основі даних про положення та властивості камери для кожного пікселя зображення обчислити промінь камери (або множину променів), вздовж якого освітлюється даний піксель. Тоді значенням пікселя має бути енергетична яскравість (radiance) світла, що потрапляє у камеру вздовж даного променя або їх множини. Для її обчислення потрібно обчислити світло з усіх видимих поверхонь, що перетинає промінь (їх може бути декілька, якщо деякі з них напівпрозорі), що робиться обчисленням рівняння рендерінгу, де у свою чергу вимагається освітленість поточної точки простору вздовж променів напівсфери інтегрування і т. д.

Найпростішим алгоритмом обчислення якісного наближення освітленості точки з урахуванням глобального освітлення є метод стохастичного трасування променів [19, 22]. Він базується на обчисленні інтегралу рівняння рендерінгу методом Монте-Карло – це означає, що відбувається трасування множини випадкових променів з точки, обчислюються відбите світло, що надійшло з даних напрямків і обчислюється середній результат. Опишемо алгоритм трасування одного променя.

1. Нехай маємо: точка  $p$  – та, для якої треба порахувати освітлення,  $n$  – нормаль у даній точці,  $\omega_0$  – напрям, у якому відбивається світло, величину якого потрібно обчислити. Також приймемо що  $L$  – поточний результат (початкове значення нульове),  $F$  – поточний BRDF-множник (початкове значення – 1).

2. Додаємо до результату величину випроміненого світла з точки  $p$ , враховуючи BRDF-множник (воно буде ненульовим у випадку якщо точка є джерелом світла):

$$L := L + F * L_0(\omega_0)$$

3. Стохастично обираємо напрям  $\omega$  з напівсфери, орієнтованої вздовж нормалі  $n$ .

4. Здійснюємо трасування променя  $(p, \omega)$  у сцені, знаходимо точку  $p'$  перетину; нехай  $n'$  – нормаль у цій точці.

5. Додаємо у BRDF-множник вплив BRDF даної точки:

$$F := F * f_p(\omega_0, \omega)(\omega \cdot n)$$

6. Переходимо до пункту 2, оновивши поточні дані:

$$p := p'; n := n'; \omega_0 := -\omega$$

Цикл продовжується допоки  $F$  не буде достатньо малим для того, щоб подальші обчислення не впливали на значущі цифри результату. Також часто вводять додаткові обмеження на кількість ітерацій для збільшення швидкодії.

Цей метод є дуже важливим, оскільки за умови достатньої кількості вибірок дає точний результат глобального освітлення. Це означає, що його можна використовувати як еталонний результат для оцінки якості та порівняння різних алгоритмів обчислення глобального освітлення. Зазвичай при вивченні деякого алгоритму вибираються деякі умови рендерінгу (сцена, динаміка об'єктів та камери тощо), які подаються на вхід цьому алгоритму та методу стохастичного трасування променів, що запускається з достатньо великою кількістю трасувань. Отримані результати порівнюють між собою певними метриками. Серед поширених метрик можна виділити наступні:

1. MSE (mean squared error, середньоквадратичне відхилення):

$$MSE(R, T) = \frac{1}{mn} \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} (R(i, j) - T(i, j))^2,$$

де  $m, n$  – розміри зображення,  $R, T$  – результуючі зображення двох алгоритмів відповідно.

2. PSNR (peak signal-to-noise ratio, пікове співвідношення сигналу до шуму):

$$PSNR(R, T) = 20 \log_{10} \left( \frac{M}{\sqrt{MSE(R, T)}} \right),$$

де  $M$  – максимальне значення серед усіх пікселів обох зображень.

Також даний алгоритм широко використовується для обчислення глобального освітлення за необхідності якнайточнішого результату – наприклад, у мультиплікації та кінематографічній графіці. На жаль, застосувати цей метод в умовах реального часу доволі важко – для отримання точного результату зазвичай потрібно обчислити дуже багато трасувань, а для цього потрібно дуже багато обчислювальних ресурсів.

#### 4. Методи обчислення глобального освітлення в реальному часі

На даний момент розроблено багато алгоритмів/структур даних обчислення глобального освітлення в реальному часі. Загалом їх можна поділити на дві групи: статичні передобчислені структури [4, 12-13, 16, 18] та методи наближеного динамічного обчислення [3, 6-7, 9-10].

Статичні передобчислені структури – спеціальні структури даних, що певним чином зберігають інформацію про глобальне освітлення або інформацію, яка призначена для швидкого обчислення глобального освітлення. Найпоширенішими є такі структури:

1. Карта освітлення (lightmap) [4, 13, 18] – це структура даних, при використанні якої освітленість поверхонь у віртуальній сцені попередньо обчислюється та зберігається у текстурах для подальшого використання. Карти освітлення складаються з люмелів (lumel, lumen element), по аналогії до текстелів (texel, texture element), елементів текстур. Менші люмелі дають карту освітлення з більш високою роздільною здатністю, забезпечуючи більшу деталізацію освітленості ціною зниження продуктивності та збільшення використання пам'яті. Таким чином, використовуючи цю техніку, дизайнерам рівнів та 3D-художникам часто доводиться йти на компроміс між продуктивністю та якістю; якщо одночасно використовувати достатньо багато карт освітлення з високою роздільною здатністю, їх об'єм може перевищити об'єм пам'яті відеокарти, що може знизити продуктивність програми або призвести до неможливості її виконання на даному апаратному забезпеченні. Значення люмелів карти освітлення можуть бути обчислені за допомогою певного алгоритму обчислення глобального освітлення; зазвичай використовується алгоритм стохастичного трасування променів. Карти освітлення добре підходять для збереження дифузного освітлення (освітлення дифузних поверхонь), оскільки воно не залежить від напрямку падіння світла, а є звичайною скалярною характеристикою точки поверхні. Але очевидно, що вони працюють лише у повністю статичних сценах – при будь-яких змінах у сцені карти освітлення втрачають релевантність.

2. Light probes [12] – це структура даних, що зберігає освітлення, що надходить у певну точку простору у сцені з усіх напрямків. Метод зберігання може бути різним; найчастіше використовуються кубічні текстур (так називають масив з шести текстур, кожна з яких відповідає одній з граней уявного куба з центром

у даній точці) або спеціальні апроксимації на кшталт сферичних гармонік (spherical harmonics) [1, 8, 17]. Текстури дозволяють зберігати результат з високою дискретизацією, тож добре підходять для створення чітких віддзеркалень, в той час як сферичні гармоніки менш точні, але мають властивості, що дозволяють використовувати їх для обчислення глобального освітлення дифузних поверхонь. Перевагою light probes перед картами освітлення є те, що вони прив'язані не до об'єктів сцени, а до певної точки простору, тож можуть використовуватися не тільки для статичних, а й для динамічних об'єктів. З іншого боку, дані структури є менш точними ніж карти освітлення та призводять до графічних артефактів (неточностей), які дуже складно приховати.

Дані методи дозволяють досягнути задовільного результату освітлення поверхні. Але очевидно, що вони мають серйозні недоліки. По-перше, вони є приблизними – для досягнення високої точності необхідно виділити значні обсяги пам'яті. По-друге, їх використання вимагає значної дизайнерської роботи – вони призводять до графічних артефактів, що зазвичай унеможлиблює їх автоматичне використання, натомість ускладнює роботу 3D-дизайнерам, які мусять вдаватися до різних хитрощів для того, щоб зробити їх невидимими до користувача. По-третє, вони не можуть працювати у повністю динамічних середовищах – для попереднього обчислення вмісту таких структур зазвичай використовується метод стохастичного трасування променів, і тому у випадку певних змін у середовищі швидко оновлення таких структур є практично неможливим.

Методи наближеного динамічного обчислення класифікувати складно – існує багато їх варіацій та реалізацій. Одним з найперших можна вважати воксельне трасування конусів (voxel cone tracing) [3, 6]. Цей метод полягає у побудові наближеної структури сцени у вигляді дерева вокселів та трасуванні конусів (тобто трасуванні одночасно усіх променів з певної точки, що утворюють з певним центральним променем кут, менший за заданий) за допомогою цього дерева. Воксель – це абстракція підмножини простору у вигляді паралелепіпеда, кожне ребро якого паралельне одній з координатних осей простору та перпендикулярне двом іншим. Суміжні вокселі (ті, що мають спільні грані) утворюють воксельні сітки – тривимірні масиви вокселів, що покривають паралелепіпедну підмножину простору; воксельним деревом зазвичай

називають множину воксельних сіток різної точності, що покривають паралелепіпеди різних розмірів, але зі спільним центром. Подібні структури можуть порівняно швидко оновлюватися згідно зі змінами у сцені та дозволяють здійснювати швидко наближене трасування, що і дає можливість застосовувати такий алгоритм у реальному часі для динамічних сцен. З іншого боку, його результат так само містить серйозні графічні артефакти, виправити які дуже нелегко. Також варто зазначити, що він має невисоку точність, а для її покращення потрібно підвищувати розмірність воксельних сіток, що призводить до значного збільшення необхідного об'єму пам'яті.

Таким чином, можливості наявних методів обчислення глобального освітлення в реальному часі досить обмежені, хоч і застосовні у деяких умовах. Також всі вони мають обмежену точність та призводять до появи графічних дефектів.

## 5. Застосування стохастичного трасування променів у реальному часі

Важливим моментом у розвитку комп'ютерної графіки стала поява апаратної підтримки трасування променів у сучасних графічних адаптерах. А саме, до драйверів GPU було додано спеціальні програмні засоби побудови певних допоміжних структур для трасування променів, а до самих відеопроцесорів – спеціальні обчислювальні блоки, що прискорюють трасування променя по цим допоміжним структурам. Ця зміна відкриває багато нових можливостей у сфері комп'ютерної графіки, і одна з них – це застосування стохастичного трасування променів у реальному часі.

Станом на зараз апаратне прискорення трасування променів не дозволяє застосовувати класичний метод стохастичного трасування променів у інтерактивному режимі – незважаючи на значне прискорення операції трасування, обчислення глобального освітлення все одно потребує багато стохастичних семплів, обчислення яких займає значний час. Тому замість того, щоб обчислювати повне освітлення точки за один кадр, його обчислюють поступово, здійснюючи один або декілька семплів за кадр і генеруючи результуюче зображення для поточного кадру на основі результатів семплів поточного та минулих кадрів за допомогою спеціалізованих алгоритмів. Такі алгоритми зазвичай виконують дві основні задачі:

1. Комбінація результату стохастичних семплів поточного кадру з результатами

попередніх кадрів. Для цього алгоритм зазвичай передбачає збереження деяких значень з минулого кадру, які суміщаються з результатами поточного кадру і використовуються у наступному. Тут важливо врахувати зміни, які відбулися на сцені з минулого кадру, такі як зміщення/повороти камери та/або об'єктів сцени, певні деформації об'єктів та зміни джерел світла, оскільки такі зміни роблять деяку частину збережених результатів нерелевантною;

2. Видалення шуму (деноїзінг, denoising) – в умовах недостатньої кількості семплів отримане зображення буде містити шум та неточності – дана стадія видаляє шум та згладжує отримане зображення ціною втрати деяких деталей та появи інших неточностей у результаті.

Алгоритмів комбінації існує багато; часто розроблюються окремі алгоритми для рероекції та деноїзінгу, які потім комбінуються разом. Одним з найдосконаліших таких алгоритмів станом на зараз можна вважати алгоритм SVGF (spatiotemporal variance guided filtering) [20] з деякими модифікаціями, що зменшують дефекти від динамічних об'єктів [21].

Даний метод є одним з головних напрямків досліджень у сфері глобального освітлення та комп'ютерної графіки в цілому. Він має серйозні переваги – разом з потужним алгоритмом комбінації результатів обчислення стохастичних вибірок він дозволяє отримати досить якісний результат. Також дуже важливо те, що даний алгоритм може працювати у повністю динамічних сценах – оновлення допоміжних структур для трасування зазвичай не дуже дороге порівняно із самим трасуванням (враховуючи те, що зазвичай потрібно здійснити багато трасувань за кадр). На жаль, він також має недоліки. Основним є те, що результат алгоритму зазвичай сильно страждає від нестачі обрахованих вибірок. Якщо камера та середовище статичні, то через деяку кількість кадрів зображення зможе накопичити достатню кількість семплів, але за умов динамічного середовища інформація про освітлення буде часто втрачатися, що призводитиме до погіршення якості зображення. Цю проблему частково вирішують алгоритми видалення шуму, але в загальному випадку вони не можуть згенерувати ідеальний результат за умови відсутності необхідної інформації.

## 6. Перспективи подальших досліджень

На основі викладених відомостей та аналізу задачі глобального освітлення можна виділити

декілька основних перспективних напрямків подальших досліджень.

Одним з найперспективніших та найпопулярніших станом на зараз напрямків є вдосконалення та розробка модифікацій методу стохастичного трасування променів. З появою апаратної підтримки трасування променів у графічних процесорах застосовність даного методу у реальному часі різко зросла, що разом з його перевагами (основною з яких є те, що математичне очікування результату алгоритму є точним значенням освітлення, тобто, за достатньої ефективної кількості вибірок алгоритм дає саме коректний результат в межах заданої точності, а не певне схильне до нерівномірних похибок наближення) робить його вкрай привабливим для якнайширшого впровадження. До того ж, для багатьох інших алгоритмів даний метод також необхідний для виконання певних попередніх обчислень, пришвидшення яких полегшить роботу користувачів бібліотеки графічних алгоритмів.

Одним з головних способів вдосконалення методу стохастичного трасування променів є застосування різних моделей вибірки за значимістю (importance sampling) [5]. Це загальна техніка для алгоритмів на основі методу Монте-Карло, що полягає у зменшенні дисперсії результату обчислюваної функції шляхом модифікації розподілу випадкових чисел, що використовується для вибірки аргументів функції. При цій зміні для збереження математичного очікування замість обчислення середнього значення функції обчислюється середня частка значення функції та значення функції щільності розподілу, що дозволяє зменшити дисперсію шляхом підбору розподілу таким чином, щоб функція його щільності була графічно «схожа» на обчислювану функцію. Зменшення дисперсії результату дозволяє зменшити необхідну кількість вибірок для отримання результату необхідної точності, отже, зменшується час, необхідний для даного обчислення.

У випадку алгоритму стохастичного трасування променів підбір розподілу для вибірки напрямків є доволі непростою, оскільки на результат функції впливає декілька чинників (енергетична яскравість світла, що надходить у точку з даного напрямку, значення BRDF), і їх вплив дуже складно оцінити, тож вдосконалення підбору даних розподілів є цікавим напрямком для дослідження. Для цього існує декілька технік комбінування та покращення розподілу, наприклад, multiple importance sampling (MIS),

filtered importance sampling (FIS), resampled importance sampling (RIS) тощо [5]. Однією з найновіших розробок у сфері покращення семплінгу є метод ReSTIR [2], що покращує вибірку вхідних параметрів для обчислення освітлення у конкретному пікселі результуючого зображення, використовуючи дані про аналогічну вибірку, виконану для сусідніх пікселів зображення та цього ж пікселя у попередніх кадрах. Даний метод також дозволяє значно зменшити дисперсію результату обчислення глобального освітлення, і його вдосконалення та застосування у різних ситуаціях є досить перспективним.

Іншим перспективним напрямком дослідження є вдосконалення алгоритмів деноїзінгу. Зазвичай використання методу стохастичного трасування променів на сучасних графічних адаптерах у реальному часі призводить до наявності шуму у результаті за рахунок недостачі кількості випадкових вибірок. Також є інші методи (наприклад, light propagation volumes), вихідний результат яких містить графічні артефакти, що потребують корекції. Тому в багатьох випадках алгоритми видалення шуму є невід'ємною частиною процедури рендерінгу, тож їх вдосконалення та розробка нових алгоритмів для специфічного використання є дуже важливими.

Ще одним важливим напрямком досліджень є вдосконалення алгоритмів трасування променів. Ця процедура є однією з фундаментальних складових будь-якого алгоритму глобального

освітлення і у багатьох випадках її виконання забирає найбільше часу. Тому доволі актуальним є вдосконалення процедури трасування променів, розробка різноманітних структур для її прискорення та різних способів представлення геометрії сцени, що дозволяють оптимізувати час її виконання та обсяг необхідної пам'яті.

Розробка та вдосконалення структур для динамічного обчислення глобального освітлення також залишається перспективною, особливо в контексті обчислення дифузного глобального освітлення. Продуктивність та об'єм оперативної пам'яті відеоадапторів зростають, і це дає простір для модифікації та збільшення точності таких методів, що розширює межі їх застосування. Особливо цікаво їх дослідження у поєднанні з апаратною підтримкою трасування променів у новітніх графічних картах.

### Висновок

Обчислення глобального освітлення у реальному часі – дуже актуальна задача комп'ютерної графіки, яка станом на сьогодні не має однозначного якісного та стабільного розв'язання. Тому дослідження у цьому напрямку є дуже важливими. В даній статті було надано теоретичні відомості про проблему обчислення глобального освітлення, описано популярні підходи та розробки для вирішення даної задачі та проаналізовано найперспективніші напрямки подальших досліджень у цій сфері.

### References

1. Tomas Annen, T. , Kautz, J., Durand, F., and Seidel, H.-P.: Spherical Harmonic Gradients for Mid-Range Illumination, Proceedings of Eurographics Symposium on Rendering, June 2004.
2. Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. ACM Transactions on Graphics (Proceedings of SIGGRAPH), 39(4), July 2020.
3. Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin: Interactive Indirect Illumination Using Voxel Cone Tracing
4. Hao Chen. 2008. Lighting and Materials of Halo3. In Game Developers Conference
5. Colbert M., Premoz S., Fran G.: Importance sampling for production rendering. SIGGRAPH 2010 Course Notes (2010)
6. Crassin C.: Voxel Cone Tracing and Sparse Voxel Octree for Real-time Global Illumination. GTC 2013
7. Dachsbacher C., Stamminger M.: Reflective shadow maps. In Proceedings of the 2005 symposium on Interactive 3D graphics and games (New York, NY, USA, 2005), I3D'05, ACM, pp. 203–231.
8. Green, R.: Spherical Harmonic Lighting: The Gritty Details, 2003.
9. Kaplanyan A.: Light propagation volumes in CryEngine 3, 2009.
10. Kaplanyan A., Dachsbacher C.: Cascaded light propagation volumes for real-time indirect illumination, 2010.
11. McGuire, M., Luebke, D.: Hardware-accelerated global illumination by image space photon mapping. In HPG '09: Proceedings of the Conference on High Performance Graphics 2009, 77–89.

12. Morgan McGuire, Mike Mara, Derek Nowrouzezahrai, David Luebke: Real-Time Global Illumination using Precomputed Light Field Probes
13. Gary McTaggart. 2004. Half-Life 2 source shading. In Game Developers Conference.
14. Oat, C.: Irradiance volumes for real-time rendering. ShaderX 5: Advanced Rendering Techniques.
15. Pharr, M., Humphreys, G., Morgan Kaufmann: Physically Based Rendering, 2004.
16. Ramamoorthi, R., and Hanrahan, P.: An Efficient Representation for Irradiance Environment Maps, SIGGRAPH 2001, 497-500.
17. Sloan, P.-P.: Stupid spherical harmonics tricks. Presentation, Game Developer Conference (GDC2008), San Francisco, CA.
18. Peter-Pike Sloan and Ari Silvennoinen. 2018. Directional Lightmap Encoding Insights. In SIGGRAPH Asia 2018 Technical Briefs (SA '18 Technical Briefs ), December 4–7, 2018, Tokyo, Japan. ACM, New York, NY, USA, 3 pages.
19. Rupard, Jason, "Ray Tracing And Global Illumination" (2003)
20. Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination. In Proceedings of HPG '17, Los Angeles, CA, USA, July 28-30, 2017, 12 pages. DOI: 10.1145/3105762.3105770
21. Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient Estimation for Real-Time Adaptive Temporal Filtering. Proc. ACM Comput. Graph. Interact. Tech. 1, 2, Article 24 (August 2018), 16 pages.
22. Ingo Wald, Timothy J. Purcell, Jörg Schmittler, Carsten Benthin, Philipp Slusallek: Realtime Ray Tracing and its use for Interactive Global Illumination
23. Wang, R., Wang, R., Zhou, K., Pan, M., Bao, H.: An efficient gpu-based approach for interactive global illumination. ACM Transactions on Graphics (Proc. of SIGGRAPH 2009) 28, 3, 1–8.

Надійшла до редакції 11.12.2022