

Державний заклад
«ПІВДЕННОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені К. Д. УШИНСЬКОГО»



ОДЕСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ імені І. І. МЕЧНИКОВА

**ДВАДЦЯТЬ ДРУГА ВСЕУКРАЇНСЬКА КОНФЕРЕНЦІЯ
СТУДЕНТІВ І МОЛОДИХ НАУКОВЦІВ**

**ІНФОРМАТИКА, ІНФОРМАЦІЙНІ
СИСТЕМИ ТА ТЕХНОЛОГІЇ**

25 квітня 2025 р.

Одеса – 2025

Інформатика, інформаційні системи та технології: тези доповідей двадцять другої всеукраїнської конференції студентів і молодих науковців. Одеса, 25 квітня 2025 р. - Одеса, 2025. – 315 с.

Друкується за рішенням Вченої Ради
Університету Ушинського
(протокол № 16 від 29.05.2025 р.)

Організатори конференції продовжують традицію обміну досвідом у сфері освіти та використання інформаційних технологій. У конференції приймають участь студенти, аспіранти та молоді науковці вищих навчальних закладів України.

Тематика конференції охоплює наступне коло питань: сучасні інформаційні технології; інтелектуальні системи; методика викладання інформатики; інформаційні технології в освіті; психолого-педагогічне забезпечення інформатизації навчальної діяльності; дистанційна освіта і глобальні телекомунікаційні мережі; математичне моделювання й інформаційні технології; інформатизація системи керування освітою; інформаційні технології в менеджменті.

Наукові керівники:

завідувачка кафедри прикладної математики та інформатики
фізико-математичного факультету Університету Ушинського, д. т. н., проф. Т. Л. Мазурок,
завідувач кафедри математичного забезпечення комп'ютерних систем факультету математики, фізики та
інформаційних технологій ОНУ імені І. І. Мечникова, д. т. н., проф. Є. В. Малахов

Оргкомітет:

Голова:

Ректор Університету Ушинського,
д. і. наук, доц. А. В. Красножон

Заступники голови:

Проректор з наукової роботи Університету Ушинського, д. політ. н., проф. Г.В. Музиченко
Декан факультету математики, фізики та інформаційних технологій ОНУ імені І. І. Мечникова,
д. ф-м. н., проф. Ю. А. Ніцук

Члени оргкомітету:

д. т. н., проф.	Є. В. Малахов	д. т. н., проф.	Т. Л. Мазурок
д. т. н., проф.	Ю. О. Гунченко	к. п. н., доц.	А. О. Яновський
ст. викладач	І. М. Лісіцина	викладач	О. Я. Рубанська
ст. викладач	Н. Ф. Трубіна	к. ф.-м. н.	О. П. Бойко
ст. викладач	В. А. Корабльов	PhD, associated prof. (Poland)	A. Rychlik

© Навчально-науковий інститут природничо-математичних наук, інформатики та менеджменту Державного закладу «Південноукраїнський національний педагогічний університет імені К. Д. Ушинського», кафедра прикладної математики та інформатики, 2025

© Факультет математики, фізики та інформаційних технологій Одеського національного університету імені І. І. Мечникова, кафедра математичного забезпечення комп'ютерних систем, 2025

ПРИНЦИПИ ПОБУДОВИ МОДЕЛІ ФОРМУВАННЯ ОПТИМАЛЬНИХ МАРШРУТІВ ДОСТАВКИ ТОВАРІВ	208
Бойко Д. С.	208
КОМПЛЕКСНА СИСТЕМА БЕЗПЕКИ З ЛАЗЕРНИМИ СЕНСОРАМИ ТА TELEGRAM-ІНТЕГРАЦІЄЮ	211
Соценко М. В., Каменєва А. В.	211
ГЕНЕРАТОР СЛІВ У СЕРЕДОВИЩАХ СИМУЛЯЦІЇ ЕЛЕКТРИЧНИХ СХЕМ.....	213
Руссу Я. С., Мирошніченко А. А., Шугайло Ю. Б.	213
ПРОБЛЕМИ ПРОГРАМНОГО ПРИСКОРЕННЯ ТРАСУВАННЯ ПРОМЕНІВ	215
Сокур О. М., Петрушина Т. І.	215
ІНТЕГРАЦІЯ ОСВІТНІХ МАТЕРІАЛІВ І ЗАСОБІВ КОМУНІКАЦІЇ В ЄДИНУ НАВЧАЛЬНУ СИСТЕМУ	219
Сокол Д. С.	219
ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ ЧАТ-БОТА ТА ТРАДИЦІЙНИХ КАНАЛІВ ТЕХНІЧНОЇ ПІДТРИМКИ ПРОГРАМНИХ ПРОДУКТІВ.....	221
Ярмошевич А. І.	221
ВИКОРИСТАННЯ НАВЧАННЯ З ПІДКРІПЛЕННЯМ ДЛЯ НАВІГАЦІЇ ДРОНІВ ЗА ДОПОМОГОЮ СЕНСОРНИХ ДАНИХ	222
Рябов Д. А., Пенко В. Г.	222
РОЗРОБКА МОДУЛЯ «ОБЛІК РЕЗУЛЬТАТІВ ВИКОНАННЯ ДОРОЖНЬО-БУДІВЕЛЬНИХ ПРОЕКТІВ» ІС ПРИВАТНОЇ КОНТРОЛЮЮЧОЇ ФІРМИ.....	224
Наконечний В. В.	224
ПОБУДОВА ГЕОМЕТРИЧНИХ ФІГУР СТЕРЕОМЕТРІЇ ЗА ДОПОМОГОЮ СЕРЕДОВИЩА GEOGEBRA.....	226
Ленько Б. П., Шаповалова Н. В.	226
РОЛЬ НЕЙРОННИХ МЕРЕЖ У СИСТЕМАХ РОЗПІЗНАВАННЯ ЕМОЦІЙ.....	228
Іванова І. В., Рудніченко М. Д.	228
ПРОЦЕС НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ.....	230
Іванова І. В., Рудніченко М. Д.	230
АНАЛІЗ АБСТРАКТНОГО СИНТАКСИЧНОГО ДЕРЕВА ДЛЯ ВИЯВЛЕННЯ СХОЖОСТІ КОДУ	231
Лебеденко Б. А., Антоненко О. С.	231
ПРОГНОЗУВАННЯ ПОГОДИ З ВИКОРИСТАННЯМ ІoT ТЕХНОЛОГІЇ.....	233
Чернова О. Ю., Антоненко О. С.	233
ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ ОБРОБКИ ДАНИХ У ДІЯЛЬНОСТІ ЗАКЛАДІВ СЕРЕДНЬОЇ ОСВІТИ	235
Цвяшко В. Ю.	235
ВИКОРИСТАННЯ ЧАТ-БОТІВ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ У НАВЧАННІ ІНФОРМАТИКИ	237
Міронова А. Ю., Савченко М. О., Коновалов С. М.	237

ПРОБЛЕМИ ПРОГРАМНОГО ПРИСКОРЕННЯ ТРАСУВАННЯ ПРОМЕНІВ

Сокур О. М., Петрушина Т. І.

Одеський національний університет імені І.І. Мечникова

Анотація. У тезах розглядаються проблеми та ключові алгоритми програмного прискорення процесу трасування променів.

Ключові слова: трасування променів, трасування шляху, комп'ютерна графіка, ray tracing, path tracing, BVH, CGI, photorealistic rendering algorithms.

Вимоги до генерації якісних фотореалістичних зображень сьогодні високі. Для отримання максимально наближеного до реальності комп'ютерного зображення використовуються алгоритм під назвою трасування променів (ray tracing) [1]. Цей підхід дозволяє моделювати фізичне розповсюдження світла в середі за допомогою обчислення просування променя, який є математичною моделлю фотону. Трасування променів дозволяє враховувати такі ефекти як фізично точні тіні, віддзеркалення, переломлення та розсіювання променів світла. Головна проблема, що виникає при програмній реалізації трасування променів, є досі очевидною і полягає в кількості необхідних променів для отримання достатньої інформації про середовище. Так, наприклад, звичайна 100 ватна лампа накаливання випромінює близько 10^{19} фотонів в секунду. Звичайно обчислення такої кількості променів є майже неможливим і точно не може бути виконано у реальному чи близькому до реального часі на сучасних комп'ютерах, але для отримання більш менш якісного зображення все одно доводиться відстежувати сотні мільйонів променів, і це тільки для єдиного зображення, тоді як для генерації фільмів, анімацій чи відеоігор це значення значно більше.

Спершу необхідно розглянути основний принцип роботи алгоритму. Тривимірна сцена моделюється певними примітивами, частіше за все трикутниками. Аналогічно до того як виконується фото та відео зйомка, задача полягає в тому щоб відстежити якомога більше промінів від джерела світла до камери та зафіксувати накоплене значення світла, що потрапило до певного пікселя віртуальної камери. Як правило, цей шлях може складатися з багатьох відбитків та переломлень від різних поверхонь що належать примітивам сцени. Деякі промені можуть вийти за межі сцени чи повністю поглинутись поверхнею. При кожному відбитті промінь може втрачати інтенсивність та набувати певного кольору.

Базовою оптимізацією в даному алгоритмі є визначення напрямку, в якому відстежуються промені. З точки зору правильності вирішення задачі це не має значення, оскільки шлях променя не залежить від зміни напрямку, але, при практичній реалізації, є сенс відстежувати тільки ті промені, що точно попадуть до камери, тобто починати від камери у зворотному до розповсюдження напрямі. Це

справедливо, якщо поверхня джерела (або джерел) світла більша за поверхню сенсора віртуальної камери. Це пояснюється тим, що вірогідність того, що промінь з джерела потрапить до маленької камери менша за вірогідність того що промінь, що вийшов з камери, потрапить до великого джерела, а отже більша кількість променів будуть застосовані при обчисленні і, відповідно, кінцеве зображення буде отримано швидше.

Отже, створивши промінь, що виходить з камери, виконується пошук його перетину з поверхнею. Після цього необхідно оцінити як ця точка або фрагмент поверхні освітлений. Важливим аспектом розповсюдження світла є те як промінь розсіюється в різні напрями при потраплянні на поверхню (за виключенням ідеальної дзеркальної поверхні). Математично це описується функцією BRDF (Bidirectional reflectance distribution function), що визначає розподіл імовірностей відбиття променя в залежності від напрямку потрапляння та відбиття світла. Математично задача знаходження освітлення певної точки поверхні вирішується інтегруванням по всім можливим напрямкам [1]. Це означає, що необхідно згенерувати нескінчену кількість променів щоб визначити всі можливі напрямки, що неможливо на практиці, тому інтегрування вирішується наближено за допомогою метода Монте-Карла. Тобто, з первинного променя в точці перетину з поверхнею необхідно згенерувати декілька вторинних променів, як правило, у випадковому напрямку, з інтенсивністю, що відповідає значенню функції розподілу у цьому напрямку. Кожний з вторинних променів також може потрапити на певну поверхню, де процес повторюється рекурсивно. На практиці рекурсія переривається після декількох відбитків тому що кількість світла що буде отримана з променів глибокої рекурсії буде незначної та нею можна знехтувати (за виключенням дзеркальних поверхонь для яких можна навмисно додати рекурсивні кроки). Цей різновид алгоритму трасування променів також називається трасування шляху (path tracing) та був вперше представлений ще у 1986 році Джеймсом Каджія [2].

На цьому етапі можна використовувати певні алгоритми, що дозволяють більш ефективно виконувати трасування променів і полягають в тому, що замість випадкового напрямку променів обраються такі, що з більшою вірогідністю потрапляють до джерела світла [3]. Цей підхід дещо зменшує точність та справжність освітлення, так як може отримати недостатньо інформації зі слабо освітлених напрямів, хоча різниця не є суттєвою те не може бути визначена на перший погляд без детального аналізу.

Найбільшого пришвидшення можна досягти за допомоги так званих структур прискорення (acceleration structures). Структури прискорення — це структури даних, які використовуються в трасуванні променів для прискорення процесу пошуку

перетинів між променями та об'єктами в сцені. Без них підхід грубої сили перевірки кожного променя на кожен об'єкт був би надзвичайно повільним, навіть для помірно складних сцен, займаючи час лінійно залежний від кількості примітивів [1]. Застосування структур прискорення дозволяє ефективно шукати перетин променя з примітивами, не перевіряючи перетин з усіма примітивами, а лише з тими, що потенційно можуть знаходитись на шляху променя. Для цього примітиви сцени мають бути правильно впорядковані, в залежності від їх положення у просторі та конкретної структури прискорення. Використання структур прискорення аналогічно до того як пошук у відсортованому масиві може бути виконано значно швидше ніж у масиві з довільною структурою. Завдяки використанню структур прискорення трасування променів часто може виконуватися за $O(m \log n)$ часу, де m – кількість пікселів, а n – кількість об'єктів. Хоча створення цих структур зазвичай займає принаймні $O(n)$ часу. Необхідно враховувати, що деякі структури прискорення можуть мати негативний ефект для динамічної сцени, через необхідність переобчислення структури за незначної модифікації сцени.

Існує декілька типів структур прискорення, причому ієрархії обмежувальних об'ємів (BVH) і Kd-дерева є найпоширенішими в сучасних трасувальниках променів [4].

Bounding Volume Hierarchies (BVH) базуються на поділі примітивів на ієрархію непересічних наборів. Кожен вузол в ієрархії зберігає обмежувальний бокс примітивів у вузлах під ним, а самі примітиви зберігаються у листках [1].

Під час обходу променя, якщо промінь не перетинає межі вузла, усе піддерево під цим вузлом можна пропустити. Це дозволяє швидко відхиляти великі частини геометрії сцени.

Алгоритми побудови BVH можуть варіюватися від швидких до складних, балансуючи між швидкістю побудови та якістю BVH (що впливає на швидкість трасування променів)[5].

Kd-дерева базуються на адаптивному рекурсивному просторовому підрозділі, розбиваючи простір на ієрархію непересічних множин.

Хоча вони широко використовуються, вони можуть бути менш ефективними у будівництві та більш схильні до пропусків перехресть через помилки округлення порівняно з BVH. Незважаючи на це, багато трасувальників променів використовують kd-дерева [5].

Останньою, але не менш важливою частиною є програмна реалізація та її адаптація під апаратуру. По-перше, використання GPU (графічного процесора) є значно кращим варіантом за використання CPU (центрального процесора) так як GPU налічує в сотні разів більше ядер. Так, наприклад GeForce RTX 5090 має 21760

ядер [6] і хоча структурно вони значно простіші та менш потужні ніж ядра CPU, їх кількість значно обходить 8-16 ядер на типовому CPU і тому є вирішальною.

З іншої ж сторони використання GPU ускладнює програмну реалізацію. По-перше адаптація алгоритмів для паралельного обчислення сама по собі складна задача, яка може потребувати значної адаптації алгоритмів та, можливо, знаходження альтернативних. Інша проблема пов'язана з тим, що велика кількість ядер йде з певними архітектурними обмеженнями. Програмування на GPU досить сильно відрізняється від програмування звичайних програм. Необхідно враховувати, що ядра GPU поділені на групи, що також називаються warp або wavefront. Група складається приблизно з 32-64 паралельних потоків, в залежності від конкретної моделі GPU. В межах однієї групи використовується модель SIMD (Single Instruction/Multiple Data), тобто лише один потік інструкцій на групу паралельних ядер. Як результат, будь-які розгалуження (оператор `if` або цикли, що мають різну кількість ітерацій) імітуються і фактично виконуються всі гілки програми.

І хоча, на перший погляд, задача трасування променів здається досить легко розпаралелюється, враховуючи те, що обчислення кожного окремого променя логічно не залежить від обчислень інших променів, і достатньо було б просто виділити кожному з променів окремий потік, це не зовсім так у випадку реалізації на GPU. Проблема полягає в тому, що навіть промені які початково вийшли з однієї точки можуть проходити дуже різні шляхи та, відповідно, перевірятися на перетин з різними примітивами різних типів у різній послідовності, самі ж примітиви можуть мати різні матеріали, в залежності від яких, рекурсивно генерувати інші промені. Це лише найбільш значущі розгалуження, яких набагато більше. Тобто задача трасування променів дуже неоднорідна і не може бути ефективно запрограмована на GPU без розбиття на елементарні підзадачі, та їх групування за схожими властивостями, які потім вже можна виконувати паралельно. Групуючи подібні промені, вартість обходу структури прискорення та виконання операцій зі стеком можна значно зменшити і збільшити частоту попадання в кеш[7].

Отже, хоча трасування променів є найкращим підходом до візуалізації фотореалістичних зображень, фільмів та відеоігор, складність алгоритму досить велика і, задля отримання результату у прийнятний проміжок часу, важливо розуміти та застосовувати розглянуті алгоритми. Також розуміння існуючих алгоритмів є ключовим для подальшого дослідження алгоритмів у галузі трасування променів, що залишається актуальним доки не досягнуто повноцінного трасування шляху в реальному часі.

Література

1. Matt Pharr, Wenzel Jakob, Greg Humphreys Physically Based Rendering, Fourth Edition. [Електронний ресурс] – Режим доступу: <https://pbr-book.org/4ed>
2. Kajiya, J. T. (1986). "The rendering equation". Proceedings of the 13th annual conference on Computer graphics and interactive techniques. ACM.
3. Justin F. Talbot "Importance Resampling for Global Illumination" – Режим доступу: <https://cdn.pharr.org/ReSTIR.pdf>
4. Daniel Meister, Jiří Bittner Performance Comparison of Bounding Volume Hierarchies for GPU – Режим доступу: <https://jcgt.org/published/0011/04/01/paper.pdf>
5. Daniel Meister, Carsten Benthin Survey on Bounding Volume Hierarchies – Режим доступу: for Ray Tracing https://www.researchgate.net/publication/352138447_A_Survey_on_Bounding_Volume_Hierarchies_for_Ray_Tracing
6. GeForce RTX 5090 [Електронний ресурс] – Режим доступу: <https://www.nvidia.com/en-us/geforce/graphics-cards/50-series/rtx-5090/>
7. Timo Aila, Samuli Laine Understanding the Efficiency of Ray Traversal on GPUs [Електронний ресурс] – Режим доступу: https://users.aalto.fi/~ailat1/publications/aila2012hpg_techrep.pdf

ІНТЕГРАЦІЯ ОСВІТНІХ МАТЕРІАЛІВ І ЗАСОБІВ КОМУНІКАЦІЇ В ЄДИНУ НАВЧАЛЬНУ СИСТЕМУ

Сокол Д. С

Національний університет «Одеська політехніка»

Ключові слова: цифрова освіта, навчальна платформа, відкритий вихідний код, дистанційне навчання.

У ХХІ столітті освіта зазнає стрімких змін під впливом цифрових технологій, що відкривають нові можливості для ефективного навчання, взаємодії та управління знаннями. В умовах глобалізації та дистанційної взаємодії виникає необхідність у створенні універсальних програмних рішень, які б об'єднували функціональність подання навчального контенту та засобів комунікації між учасниками освітнього процесу. Зокрема, інтеграція освітніх матеріалів і засобів комунікації в єдину навчальну систему є актуальним завданням для освітніх установ, що прагнуть модернізувати навчальний процес і зробити його більш гнучким, доступним та інклюзивним.

Метою представленого дослідження є проектування програмного забезпечення, яке дозволяє поєднати в одному середовищі мультимедійні освітні ресурси (тексти, відео, презентації, тести, інтерактивні завдання) та інструменти для синхронної й асинхронної комунікації (чати, відеозв'язок, коментарі тощо). Така система сприяє

Державний заклад
«ПІВДЕННОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
імені К. Д. УШИНСЬКОГО»



ОДЕСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ імені І. І. МЕЧНИКОВА

ДВАДЦЯТЬ ДРУГА ВСЕУКРАЇНСЬКА КОНФЕРЕНЦІЯ
СТУДЕНТІВ І МОЛОДИХ НАУКОВЦІВ

ІНФОРМАТИКА, ІНФОРМАЦІЙНІ
СИСТЕМИ ТА ТЕХНОЛОГІЇ

Збірник робіт

Збірник робіт надрукований в авторській редакції
без внесення суттєвих змін оргкомітетом

Підписано до друку 25.04.2025
Здано у виробництво 25.04.2025
Формат 60x84/16. Папір офсетний. Друк офсетний.
Тираж 50 примірників

Надруковано з готового оригінал-макета