



KAPITEL 10 / CHAPTER 10¹⁰

METHODS FOR IMPROVING THE EFFICIENCY OF IMAGE OPTIMIZATION IN 3D GRAPHICS

DOI: 10.30890/2709-2313.2025-41-07-019

Вступ

3D-графіка сьогодні є невід'ємною частиною багатьох галузей, включаючи ігрову індустрію, кіно, віртуальну реальність та архітектурне моделювання. Однак, зростаючі вимоги до якості зображень та швидкості рендерингу створюють нові виклики для розробників та дизайнерів. Одним із ключових аспектів досягнення високої продуктивності є оптимізація зображень 3D-графіки.

3D-графіка має величезний вплив на багато галузей сучасного світу, включаючи комп'ютерні ігри, віртуальну реальність (VR), доповнену реальність (AR), кінематограф та архітектурне моделювання. Оптимізація 3D-зображень має вирішальне значення для досягнення реалістичних сцен без втрати продуктивності.

3D-графіка активно застосовується в архітектурному моделюванні, медицині та наукових дослідженнях, де візуалізація має ключове значення. Для цих сфер важливо забезпечити максимально реалістичні зображення без втрати якості під час оптимізації. Сучасні алгоритми оптимізації, зокрема методи стиснення текстур і полігонів, дозволяють досягати цього балансу. Це значно покращує користувацький досвід і підвищує ефективність роботи фахівців у різних галузях.

Штучний інтелект і машинне навчання також відіграють важливу роль у вдосконаленні технологій оптимізації 3D-графіки. Наприклад, нейромережі здатні автоматично визначати найменш важливі елементи зображення та оптимізувати їх без впливу на загальну якість. Це забезпечує нові можливості для створення графічного контенту високої якості з мінімальними витратами

¹⁰ Authors: Kobzev Igor, Petrov Konstantin, Hrabovskyi Yevhen, Horelov Yurii, Kosenko Viktor

Number of characters: 29712

Author's sheets: 0,74



ресурсів. Технології оптимізації зображень 3D-графіки мають великий потенціал для розвитку, що робить їх важливим напрямом у сучасних дослідженнях і розробках.

Отже, актуальність вдосконалення технології оптимізації зображень 3D-графіки обумовлена зростаючими вимогами до якості візуалізації та швидкості обробки даних у багатьох сферах.

В роботах [1-2] пропонуються підходи щодо створення адаптивного інтерфейсу веб-базованих інструментальних засобів оптимізації графічних зображень. Опис конкретних алгоритмів, на основі яких можуть бути спроектовані певні елементи технології оптимізації 3D-графіки, представлений в дослідженнях [3, 4]. У працях [5, 6] надаються рекомендації щодо використання 3D-графіки для створення інтерфейсу веб-додатків та інструментальних засобів у реальному часі з метою навігації у віртуальному середовищі. Дослідження [7] містить методичні рекомендації щодо обґрунтування інноваційної стратегії розвитку інформаційних технологій як основи для подальшого вдосконалення технології оптимізації 3D-графіки. Практичні рекомендації щодо використання Workflow, які можуть бути використані для оптимізації 3D-графічних зображень наведено роботі [8].

10.1. Оптимізація у 3D-графіці

Оптимізація у 3D-графіці є ключовим аспектом розробки будь-якого додатку, що використовує тривимірні моделі, оскільки вона безпосередньо впливає на продуктивність і користувацький досвід. Висока кількість полігонів, текстур та складні світлові ефекти можуть створювати значне навантаження на графічний процесор (GPU), що призводить до зниження частоти кадрів та затримок. Для забезпечення плавного відображення сцени, особливо у реальному часі, необхідно оптимізувати об'єкти так, щоб вони виглядали реалістично, але при цьому не перевантажували апаратне забезпечення.

Крім забезпечення продуктивності, оптимізація має значення для зниження



вимог до обчислювальних ресурсів, таких як оперативна пам'ять та пам'ять відеокарти. Це дозволяє досягти кращої сумісності з різними платформами, зокрема мобільними пристроями, де обмежені ресурси. За допомогою методів стиснення текстур, зменшення рівня деталізації (LOD) та спрощення полігональних моделей можна підтримувати високий рівень візуальної якості навіть на слабших пристроях. Це робить оптимізацію критично важливою для розробників, які прагнуть досягти балансу між якістю зображень та доступністю для широкої аудиторії.

Оптимізація також сприяє зниженню часу рендерингу, що є особливо важливим у галузях, де потрібна висока швидкість обробки даних, таких як архітектурне моделювання, симуляції та віртуальна реальність (VR). Зменшуючи час, необхідний для візуалізації складних сцен, оптимізація дозволяє зосередитись на покращенні деталізації та інтерактивності, створюючи більш реалістичний і захоплюючий досвід для користувачів. Таким чином, оптимізація у 3D-графіці забезпечує продуктивність, доступність та якість, що робить її невід'ємною частиною процесу розробки візуального контенту.

Оптимізація зображень у 3D-графіці полягає у зменшенні навантаження на апаратні ресурси, забезпечуючи при цьому максимальну якість зображення. Це особливо важливо для реального часу, де кожна затримка може негативно вплинути на сприйняття користувачем. Основними аспектами, які підлягають оптимізації, є полігони, текстури, освітлення та тіні, а також рендеринг на основі шейдерів.

Оптимізація полігонів, або «LOD» (Level of Detail), полягає у зменшенні кількості полігонів в об'єктах, що знаходяться на великій відстані від камери. Це дозволяє значно знизити навантаження на графічний процесор (GPU), що особливо важливо для ігор та додатків з відкритим світом. Сучасні методи використовують алгоритми динамічного зміщення, які автоматично адаптують рівень деталізації в залежності від положення камери.

Текстури займають значну частину пам'яті, що використовується під час рендерингу сцени. Використання методів компресії, таких як DXT (S3TC), ASTC



або новітні формати на основі машинного навчання, дозволяє знизити розмір текстур без втрати якості. Крім того, використання текстурних атласів дозволяє уникнути додаткових завантажень та перемикань текстур, що прискорює рендеринг.

Освітлення та шейдери є важливими компонентами 3D-сцен, які впливають на кінцеву якість зображення. Традиційне освітлення часто замінюється на підходи, що використовують глобальне освітлення та фізично коректні методи рендерингу (PBR). Використання таких технік, як baking освітлення або Light Probes, дозволяє зберегти якість світлових ефектів при мінімальних затратах на обчислення. Оптимізація шейдерів полягає у спрощенні складних операцій та використанні комп'ютерного буфера (G-buffer) для обчислення освітлення у реальному часі.

Новітні підходи до рендерингу, такі як трасування променів у реальному часі та рендеринг на основі вокселів, дозволяють значно покращити якість зображень, але потребують значних обчислювальних ресурсів. Для зниження навантаження на систему застосовуються гібридні підходи, що поєднують традиційний рендеринг та трасування променів. Також активно використовуються алгоритми штучного інтелекту, такі як DLSS (Deep Learning Super Sampling) від NVIDIA, які дозволяють рендерити зображення у нижчій роздільній здатності та збільшувати їх без значної втрати якості.

Машинне навчання та нейронні мережі стають все більш популярними для оптимізації 3D-зображень. Вони дозволяють автоматизувати процеси, які раніше вимагали значної участі людини, наприклад, створення LOD-моделей, генерацію нормалізованих карт та передбачення складних світлових ефектів. Завдяки машинному навчанню можна досягти оптимізації якості зображень без значного збільшення навантаження на апаратне забезпечення.

Хоча сучасні методи оптимізації значно покращили якість та продуктивність 3D-зображень, існують ще багато викликів. Розробка нових алгоритмів та інтеграція машинного навчання в процес рендерингу вимагають значних обчислювальних ресурсів та часу на навчання моделей. Однак, з



розвитком хмарних обчислень та потужніших графічних процесорів, з'являються нові можливості для подальшого вдосконалення процесу оптимізації.

Основна мета оптимізації зображень у 3D-графіці – зменшити навантаження на графічний процесор (GPU) та центральний процесор (CPU), зберігаючи при цьому максимальну якість зображення. Це дозволяє досягти плавності анімації, уникнути затримок та забезпечити високу якість графіки в режимі реального часу.

Полігональна модель є основою для представлення 3D-об'єктів. Полігони зазвичай представлені трикутниками, які об'єднуються для створення поверхонь. Оптимізація полягає у зменшенні кількості полігонів, не втрачаючи при цьому деталізації об'єктів.

Формула площі трикутника за координатами трьох вершин повинна мати такий вигляд (1):

$$S = \frac{1}{2} |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)| \dots \dots \dots (1)$$

де $(x_1; y_1)$, $(x_2; y_2)$ та (x_3, y_3) - координати вершин трикутника.

Для оптимізації використовуються алгоритми зменшення кількості полігонів (mesh simplification), такі як Quadric Error Metrics (QEM), які обчислюють помилку між початковою та спрощеною моделями.

Текстури використовуються для додання деталізації 3D-об'єктам. Зменшення розмірів текстур є важливим етапом оптимізації, оскільки це дозволяє зменшити обсяг використання відеопам'яті (VRAM).

Компресія та оптимізація текстур є важливими елементами у 3D-графіці, що безпосередньо впливають на продуктивність рендерингу та використання пам'яті. Текстури представляють собою двовимірні зображення, що накладаються на поверхню 3D-об'єктів для надання їм більш реалістичного вигляду. Вони використовуються для передачі таких характеристик, як колір, блискучість, нормалі та карти відображення. Оскільки текстури часто мають високу роздільну здатність, їх обсяг може значно перевищувати обсяг геометрії самої моделі. Це створює високі вимоги до відеопам'яті (VRAM), особливо при роботі з великими сценами або в реальному часі. Звідси виникає необхідність



компресії текстур, що дозволяє зменшити їх розмір без значної втрати якості.

Компресія текстур здійснюється за допомогою спеціальних алгоритмів. Поширені методи компресії включають DXT (S3TC) та ASTC, які використовують спеціалізовані алгоритми стиснення для збереження кольорових даних з мінімальною втратою якості.

Наприклад, якщо початковий розмір текстури становить 16 МБ, а стиснутий – 4 МБ, коефіцієнт стиснення дорівнює 4, що означає зменшення розміру у чотири рази. Це суттєво знижує використання пам'яті та дозволяє зберегти більше текстур у відеопам'яті, покращуючи швидкість рендерингу.

Оптимізація текстур також включає використання текстурних атласів — великих зображень, які поєднують кілька текстур в одне зображення. Це дозволяє зменшити кількість перемикачів текстур під час рендерингу, що позитивно впливає на продуктивність, особливо у великих сценах. Важливим аспектом є правильне розміщення текстур на атласі та мінімізація порожнього простору, що можна визначити за допомогою формули ефективності використання текстурного атласу:

Формула ефективності використання текстурного атласу (2):

$$E_{atlas} = \frac{\sum_{i=1}^n A_{texture\ i}}{A_{atlas}} \quad (2)$$

де E_{atlas} – ефективність атласу, $A_{texture\ i}$ – площа i текстури, A_{atlas} – площа всього текстурного атласу.

Висока ефективність означає, що більше простору атласу використовується для корисної інформації, що мінімізує втрати ресурсів.

Компресія та оптимізація текстур тісно пов'язані з математичним моделюванням полігонів та важливістю оптимізації у 3D-графіці. Оскільки стиснення текстур дозволяє зменшити навантаження на пам'ять, це дає змогу вивільнити ресурси для обробки більш складної геометрії. Оптимізовані полігони і текстури працюють разом для досягнення балансу між якістю та продуктивністю, що є критичним для додатків з високими вимогами до реального часу, таких як ігри та віртуальна реальність (VR). Техніки компресії



допомагають зберегти деталізацію та забезпечити швидкий доступ до даних, що дозволяє досягти реалістичного відображення складних сцен без перевантаження графічного процесора та зберегти високу частоту кадрів.

Освітлення значно впливає на якість відображення 3D-сцен. Реалістичне освітлення досягається за допомогою глобального освітлення (Global Illumination, GI), яке враховує відбиття світла від різних поверхонь. Фізично коректний рендеринг (Physically-Based Rendering, PBR) дозволяє моделювати матеріали, що реагують на світло максимально наближено до реальних.

Освітлення та шейдери є ключовими елементами у створенні реалістичних зображень у 3D-графіці. Вони відповідають за взаємодію світла з поверхнями об'єктів, визначаючи, як будуть виглядати кольори, тіні, відблиски та інші світлові ефекти. Освітлення базується на фізичних принципах, таких як відбиття та заломлення світла, що дозволяє створювати візуально правдоподібні сцени. При цьому, шейдери використовуються для програмування цих ефектів на графічному процесорі (GPU), забезпечуючи деталізовану обробку освітлення для кожного пікселя або вертексу. Це дозволяє зберегти високий рівень якості зображення, навіть у складних сценах.

Освітлення має розраховуватися за допомогою моделі Ламберта для дифузного освітлення (3):

$$I_{diff} = I_{light} \cdot \max(0, n \cdot l) \quad (3)$$

де I_{diff} - інтенсивність дифузного світла, I_{light} – інтенсивність джерела світла, n – нормальний вектор до поверхні, l – вектор напрямку до джерела світла.

Для складних сцен використовується техніка попереднього розрахунку освітлення (light baking), що дозволяє знизити навантаження на систему під час виконання програми.

Освітлення та шейдери тісно пов'язані з оптимізацією полігонів і текстур, оскільки правильне використання шейдерів може значно знизити обчислювальну складність сцени, залишаючи її візуально привабливою. Завдяки застосуванню методів, таких як нормальні карти, розробники можуть зменшити кількість полігонів, не жертвуючи деталізацією, що є важливим для високої



продуктивності рендерингу. Одночасно, компресія текстур зберігає якість зображень, зменшуючи використання пам'яті, що дає змогу більш ефективно використовувати обчислювальні ресурси для складних ефектів освітлення. В кінцевому підсумку, оптимізація освітлення та текстур дозволяє створити більш реалістичні та продуктивні візуальні сцени, що є критично важливим для додатків з високими вимогами до якості графіки та швидкодії.

10.2. Вдосконалення технологій рендерингу

Вдосконалення технологій рендерингу є важливим етапом розвитку 3D-графіки, оскільки воно безпосередньо впливає на якість відображення сцен і продуктивність графічних додатків. З розвитком апаратного забезпечення та алгоритмів обробки зображень з'являються нові методи рендерингу, які дозволяють створювати більш реалістичні візуальні ефекти. Одним із таких методів є трасування променів у реальному часі (real-time ray tracing), який дає змогу точно симулювати поведінку світла, включаючи відбиття, заломлення і тіні. На відміну від традиційної растризації, яка відображає об'єкти за допомогою трикутників, трасування променів моделює шлях кожного світлового променя через сцену, що забезпечує високу реалістичність.

Трасування променів дозволяє моделювати взаємодію світла з поверхнями об'єктів, враховуючи відбиття та заломлення. Це дозволяє створювати реалістичні ефекти, такі як відблиски та рефракція.

Основне рівняння трасування променів (4):

$$P(t) = O + tD, \quad (4)$$

де $P(t)$ – точка на промені, O – початкова точка (камера), D – напрямок променя, t – параметр, що визначає відстань від початкової точки.

Для кожного пікселя на екрані обчислюється траєкторія променя, що проходить через сцену, та визначається його взаємодія з об'єктами, враховуючи відбиття та заломлення. Це дозволяє створювати реалістичні ефекти, такі як прозорі матеріали, дзеркальні поверхні та м'які тіні.



Гібридні методи рендерингу поєднують трасування променів з традиційними методами растризації, що дозволяє зберегти продуктивність при рендерингу складних сцен. Наприклад, растризація може використовуватись для відображення основних об'єктів сцени, тоді як трасування променів застосовується лише для обчислення ефектів глобального освітлення, рефлексій або заломлення у вузьких областях сцени. Це дозволяє уникнути повного моделювання складних світлових взаємодій для кожного пікселя та обчислювати їх лише тоді, коли це справді потрібно.

Інший сучасний підхід до рендерингу — це методи на основі вокселів (Voxel-based rendering) та використання рендерингу з глибоким навчанням, як-от DLSS (Deep Learning Super Sampling). Вокселі, або об'ємні пікселі, представляють 3D-простір у вигляді регулярної сітки, що спрощує обчислення освітлення та колізій між об'єктами. Воксельний рендеринг може бути використаний для симуляції об'ємних ефектів, таких як туман або дим. У той же час, технології на основі машинного навчання, такі як DLSS, дозволяють підвищити роздільну здатність рендерингу, відтворюючи високу деталізацію зображень на основі вхідних кадрів нижчої роздільної здатності. Формула для оцінки збільшення роздільної здатності у DLSS виглядає так (5):

$$S_{output} = f(S_{low}, \theta), \quad (5)$$

де S_{output} — вихідне зображення високої роздільної якості, S_{low} — вхідне зображення з низькою роздільною здатністю, θ — параметри навченої моделі.

Це дозволяє зменшити навантаження на GPU, зберігаючи високу частоту кадрів та якість зображення.

Вдосконалення технологій рендерингу безпосередньо пов'язане з оптимізацією полігонів, текстур та освітлення, оскільки нові методи дозволяють ефективніше використовувати обчислювальні ресурси. Наприклад, завдяки трасуванню променів та гібридним методам, можна використовувати менш детальні полігональні сітки та оптимізовані текстури, зосереджуючи обчислення на освітленні та відображенні. Використання машинного навчання для покращення якості зображень дозволяє досягати високої роздільної здатності без



значних витрат на пам'ять і потужність. Таким чином, новітні технології рендерингу забезпечують максимальну якість зображень при мінімальних витратах ресурсів, що є критичним для додатків з високими вимогами до продуктивності, таких як відеоігри та віртуальна реальність.

10.3. Використання машинного навчання

Використання машинного навчання (ML) для оптимізації 3D-графіки стає все більш поширеним, оскільки воно дозволяє автоматизувати складні процеси та знижувати навантаження на апаратні ресурси. Зокрема, нейронні мережі допомагають у процесах підвищення роздільної здатності зображень, компресії текстур, оптимізації освітлення та автоматичного створення LOD (рівнів деталізації). Це дозволяє досягати високої продуктивності та реалістичності без значного збільшення кількості полігонів або пам'яті, необхідної для зберігання текстур. Наприклад, технологія Deep Learning Super Sampling (DLSS) від NVIDIA використовує нейронні мережі для реконструкції зображень високої роздільної здатності з низькороздільних вхідних кадрів.

Основне рівняння для навчання нейронної мережі, що використовується в DLSS, може бути представлене як мінімізація функції втрат (6):

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i, \theta))^2, \quad (6)$$

де $L(\theta)$ – функція витрат, y_i – реальні значення, $f(x_i, \theta)$ – передбачення нейронної мережі, x_i – вхідні данні, θ – параметри моделі.

Завдяки цьому підходу, модель навчається перетворювати зображення з низькою роздільною здатністю в більш деталізовані версії, що дозволяє зберігати високу частоту кадрів навіть на вимогливих сценах.

Машинне навчання також активно використовується для компресії текстур, що пов'язане з темою оптимізації текстур та полігонів. Наприклад, використання автоенкодерів дозволяє зменшити розмір текстур без втрати значної кількості візуальної інформації. Автоенкодер складається з двох частин: енкодера, що стискає вхідну текстуру до зменшеного представлення, та декодера, який



відновлює текстуру.

Нейронні мережі можуть використовуватися для передбачення складних освітлювальних ефектів або генерації текстур високої якості з низькороздільних зразків.

Сучасні хмарні технології дозволяють виконувати складні обчислення для рендерингу та оптимізації на віддалених серверах. Це значно розширює можливості для розробників, надаючи доступ до потужних ресурсів без необхідності купівлі дорогого обладнання.

Формула розподілу завдань (7):

$$T_{total} = \frac{T_{task}}{N} + T_{overhead}, \quad (7)$$

де T_{total} – загальний час виконання, T_{task} – час виконання одного завдання, N – кількість серверів, $T_{overhead}$ – накладні витрати на розподіл завдань.

Розподілені обчислення дозволяють значно зменшити час рендерингу складних сцен.

Математичне моделювання полігонів є основою створення тривимірних об'єктів у 3D-графіці. Полігональні сітки складаються з великої кількості трикутників, які об'єднуються для формування поверхонь об'єктів. Кожен трикутник описується координатами своїх вершин у тривимірному просторі, що дозволяє формувати складні геометричні фігури. Математичні моделі, що описують ці сітки, використовують математику для представлення простору, обчислення нормалей, площ та об'єму об'єктів, що є необхідними для правильної взаємодії світла з поверхнею. Важливою складовою цього процесу є розуміння того, як обчислення впливають на продуктивність системи, адже складні моделі можуть значно уповільнювати рендеринг.

Оптимізація полігональних сіток прямо пов'язана з кількістю полігонів, що складають об'єкт. Чим більше полігонів має модель, тим детальнішою вона виглядає, але тим більші обчислювальні ресурси потрібні для її відображення. Тому, щоб знизити навантаження на графічний процесор і зменшити час рендерингу, розробники використовують техніки оптимізації, такі як зниження рівня деталізації (Level of Detail, LOD). Ці методи дозволяють автоматично



змінювати кількість полігонів залежно від відстані об'єкта до камери, залишаючи високодеталізовану модель тільки тоді, коли вона знаходиться близько до глядача. Математичне моделювання в цьому випадку включає обчислення кривих і спрощених геометрій, що підтримують візуальну цілісність при мінімальній кількості полігонів.

Крім LOD, ще одним методом є оптимізація сіток за допомогою алгоритмів спрощення, таких як Quadric Error Metrics (QEM). QEM дозволяє обчислити помилку, що виникає при видаленні або заміні вершини трикутника на іншу, і на основі цього видаляти малозначущі полігони, що не впливають на загальний вигляд об'єкта. Формула розрахунку квадратичної помилки дозволяє зменшити кількість полігонів, зберігаючи при цьому максимальну наближеність до початкової форми. Цей підхід особливо корисний для об'єктів, що мають велику кількість деталей, таких як персонажі у відеоіграх або архітектурні елементи, де важливо зберігати естетичну привабливість об'єкта.

10.4. Математичне моделювання полігонів

Математичне моделювання полігонів безпосередньо пов'язане з важливістю оптимізації у 3D-графіці, оскільки саме від правильного розрахунку та застосування геометричних моделей залежить ефективне використання ресурсів. Оптимізовані моделі дозволяють досягти високої якості зображення без значних втрат продуктивності, що особливо важливо для додатків з високими вимогами до реального часу, таких як відеоігри та VR/AR-додатки. Використання методів математичного моделювання у поєднанні з техніками оптимізації забезпечує баланс між деталізацією та швидкістю, створюючи максимально комфортний і реалістичний досвід для користувачів, знижуючи при цьому навантаження на апаратне забезпечення.

Оптимізація полігональних сіток прямо пов'язана з кількістю полігонів, що складають об'єкт. Чим більше полігонів має модель, тим детальнішою вона виглядає, але тим більші обчислювальні ресурси потрібні для її відображення.



Тому, щоб знизити навантаження на графічний процесор і зменшити час рендерингу, розробники використовують техніки оптимізації, такі як зниження рівня деталізації (Level of Detail, LOD). Ці методи дозволяють автоматично змінювати кількість полігонів залежно від відстані об'єкта до камери, залишаючи високодеталізовану модель тільки тоді, коли вона знаходиться близько до глядача.

Важливим аспектом таких алгоритмів є розрахунок видимої площі об'єкта (8):

$$A_{visible} = A_{model} \cdot \cos(\theta), \quad (8)$$

де A_{model} – площа об'єкта, θ – кут між нормаллю до поверхні об'єкта та напрямком до камери.

Математичне моделювання полігонів безпосередньо пов'язане з важливістю оптимізації у 3D-графіці, оскільки саме від правильного розрахунку та застосування геометричних моделей залежить ефективне використання ресурсів. Оптимізовані моделі дозволяють досягти високої якості зображення без значних втрат продуктивності, що особливо важливо для додатків з високими вимогами до реального часу, таких як відеоігри та VR/AR-додатки. Використання методів математичного моделювання, таких як розрахунок площі трикутників та використання метрик помилок, у поєднанні з техніками оптимізації забезпечує баланс між деталізацією та швидкістю, створюючи максимально комфортний і реалістичний досвід для користувачів, знижуючи при цьому навантаження на апаратне забезпечення.

10.5. Розподілені обчислення та хмарні технології

Розподілені обчислення та хмарні технології відіграють важливу роль у сучасній 3D-графіці, оскільки вони дозволяють виконувати складні обчислення для рендерингу та обробки великих обсягів даних на віддалених серверах, знижуючи вимоги до локального обладнання. Це особливо актуально для ресурсомістких завдань, таких як рендеринг фотореалістичних сцен з



трасуванням променів, симуляція глобального освітлення та обробка високороздільних текстур. Розподілені системи дозволяють розбити такі завдання на кілька вузлів, які паралельно виконують свої частини роботи, значно прискорюючи процес рендерингу. Це дає змогу досягти високої продуктивності та забезпечити масштабованість, що є важливим для великих студій анімації, ігрових компаній та архітектурних візуалізацій.

Хмарні технології, такі як Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure, дозволяють використовувати потужні графічні процесори для розподіленого рендерингу у віддаленому середовищі. Це дає змогу невеликим командам або індивідуальним розробникам мати доступ до потужних обчислювальних ресурсів, не інвестуючи у дороге обладнання. Наприклад, можна запускати симуляції фізично коректного рендерингу або виконувати трасування променів на віртуальних машинах із підтримкою GPU.

Високі обчислювальні вимоги методів трасування променів та машинного навчання можуть бути знижені завдяки розподіленим обчисленням, що дозволяє обробляти великі обсяги даних паралельно. Наприклад, створення LOD (рівнів деталізації) для великої кількості моделей або генерація високо деталізованих текстур може бути виконана на кількох серверах одночасно, що значно прискорює цей процес. Хмарні технології також дозволяють зберігати текстури у великій роздільній здатності та синхронізувати їх з локальними машинами, забезпечуючи оптимальний доступ до ресурсів під час рендерингу складних сцен. Це дозволяє розробникам досягти високої якості візуалізації, ефективно використовуючи доступні ресурси.



Висновки

Таким чином, оптимізація зображень у 3D-графіці є важливим аспектом сучасної цифрової індустрії. Використання передових методів, таких як компресія текстур, динамічна зміна рівнів деталізації, гібридні методи рендерингу та машинне навчання, дозволяє досягти високої якості зображень при зниженому навантаженні на апаратне забезпечення. Подальше вдосконалення цих технологій допоможе створювати ще більш реалістичні та захоплюючі візуальні світи для користувачів.

Науковим результатом дослідження є вдосконалена технологія оптимізації зображень 3D-графіки.

Практичним результатом роботи є рекомендації стосовно оптимізації зображень 3D-графіки.

Подальшим напрямком дослідження можу стати оцінка ефективності використання програмного інструментарію для оптимізації зображень 3D-графіки.