

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ШОСТКИНСЬКИЙ ІНСТИТУТ СУМСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ
ШОСТКИНСЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМЕНІ ІВАНА КОЖЕДУБА СУМДУ
Управління освіти Шосткинської міської ради
Виконавчий комітет Шосткинської міської ради

**МАТЕРІАЛИ
ВСЕУКРАЇНСЬКОЇ
НАУКОВО-МЕТОДИЧНОЇ
КОНФЕРЕНЦІЇ**

**24 КВІТНЯ
2025
ШОСТКА**

**ШОСТКИНСЬКИЙ ІНСТИТУТ
СУМСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ**

**“ОСВІТА, НАУКА ТА ВИРОБНИЦТВО:
РОЗВИТОК ТА ПЕРСПЕКТИВИ”**

РІДНУ
АЛЬМА - МАТЕР!!!
 $b' = r \log_a b$



Міністерство освіти і науки України
Сумський державний університет
Шосткинський інститут Сумського державного університету
Шосткинський фаховий коледж імені Івана Кожедуба
Управління освіти Шосткинської міської ради
Виконавчий комітет Шосткинської міської ради

ОСВІТА, НАУКА ТА ВИРОБНИЦТВО: РОЗВИТОК ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ
X Всеукраїнської
науково-методичної конференції
(Шостка, 24 квітня 2025 року)



Суми
Сумський державний університет
2025

УДК 372.862

Редакційна колегія:

відповідальний редактор – к.т.н., доцент Р.В. Закусило;
заступник відповідального редактора – к.е.н, завідувач кафедри
Ю.М. Мануйлович.

члени редакційної колегії:

к.х.н, ст.викладач С.В. Тимофіїв; к.філ.н, доцент Н.Ю. Бондар;
к.пед.н., доцент Ю.М. Мар'їнських; к.е.н, ст.викладач І.В. Новикова;
к.т.н., завідувач кафедри Г.М. Худолей; к.е.н, доцент О.М. Тур
к.е.н, ст.викладач І.В. Вареник;

ОСВІТА, НАУКА ТА ВИРОБНИЦТВО: РОЗВИТОК ТА
ПЕРСПЕКТИВИ: матеріали X Всеукраїнської науково-методичної
конференції, м. Шостка, 24 квітня 2025 року. – Суми : Сумський
державний університет, 2025. – 298 с.

Збірник містить матеріали X Всеукраїнської науково-методичної
конференції «Освіта, наука та виробництво: розвиток та перспективи»,
що проводиться на базі Шосткинського інституту Сумського
державного університету. Тематика поданих матеріалів охоплює
широке коло питань, присвячених актуальним проблемам сучасної
освіти, науки та виробництва.

Видання корисне, викладачам, аспірантам і студентам вищих
навчальних закладів, науковим співробітникам, працівникам хімічної
промисловості, фахівцям інформаційних технологій виробництва,
вчителям загальноосвітніх шкіл.

© Шосткинський інститут
Сумського державного університету, 2025
© Сумський державний університет, 2025

ОСОБЛИВОСТІ ПРОГРАМНО-АПАРАТНОЇ РЕАЛІЗАЦІЇ РЕНДЕРИНГУ

Романюк О.Н., Черняк О.І., Дембіцький О.М.
Вінницький національний технічний університет
rom8591@gmail.com

Програмно-апаратна реалізація рендерингу [1-6] полягає у комплексному поєднанні програмних алгоритмів обробки графіки з апаратними засобами, що забезпечують високу продуктивність і реалістичність виводу зображення. Рендеринг – це процес генерації зображення з тривимірної сцени за допомогою обчислень, і його ефективна реалізація потребує узгодженої роботи процесора (CPU), графічного процесора (GPU), оперативної пам'яті, а також програмного забезпечення, зокрема графічних API (наприклад, OpenGL, DirectX, Vulkan) та рушіїв (як-от Unity, Unreal Engine).

На рівні апаратного забезпечення ключову роль відіграє GPU, який здатен паралельно обробляти тисячі шейдерних операцій, що дозволяє рендерити зображення в реальному часі з високою деталізацією. У сучасних відеокартах реалізовані апаратні блоки для растеризації, обчислення освітлення, затінення, обробки текстур і глибини. Спеціалізовані компоненти, такі як RT Cores (для трасування променів) і Tensor Cores (для AI-покращення зображення), стали стандартом у новітніх GPU, зокрема від NVIDIA (архітектура Turing, Ampere, Ada Lovelace).

З програмної точки зору реалізація рендерингу охоплює побудову сцени, розміщення камер і джерел світла, призначення матеріалів і текстур, налаштування шейдерів і визначення порядку обробки об'єктів. Алгоритми рендерингу можуть бути різними: растеризація (найбільш поширена у реальному часі), трасування променів (для фотореалістичної графіки), гібридні підходи (поєднання обох) та методи глобального освітлення. Важливим є також використання буферів кадру, глибини, тіней, нормалей та постобробки, що реалізується засобами GPU-прискорення.

У мобільних і вбудованих системах також застосовуються спеціалізовані графічні прискорювачі (наприклад, ARM Mali, Imagination PowerVR), які оптимізовані під обмеження споживання енергії. У галузях, де потрібна найвища точність (наприклад, медична візуалізація або CAD), використовується рендеринг на основі CPU або FPGA, що дозволяє реалізовувати спеціалізовані математичні моделі.

Таким чином, програмно-апаратна реалізація рендерингу є результатом тісної взаємодії між складними програмними структурами та високопродуктивними апаратними компонентами, що дозволяє

досягати як реалістичності, так і швидкодії у відображенні графічних сцен.

Подальший розвиток програмно-апаратної реалізації рендерингу тісно пов'язаний із впровадженням штучного інтелекту, машинного навчання та обчислень з плаваючою точкою високої точності. Такі технології, як Deep Learning Super Sampling від NVIDIA або Xe Super Sampling від Intel, використовують нейронні мережі для реконструкції високоякісного зображення на основі меншої кількості вхідних пікселів. Це дозволяє значно зменшити навантаження на GPU без втрати якості, оптимізуючи ресурси рендерингу в реальному часі.

Крім того, у програмній частині активно використовуються комп'ютерні шейдери (vertex, fragment, geometry, compute shaders), які дозволяють виконувати кастомізовану обробку графічних даних. Комп'ютерні шейдери, особливо у форматі general-purpose computing on graphics processing units, застосовуються не лише для візуалізації, а й для фізичного моделювання, обробки зображень, глибини поля тощо.

Також важливою є оптимізація рендерингу на архітектурному рівні. Паралелізм обчислень забезпечується через графові структури обробки сцен, такі як Bounding Volume Hierarchy, які дозволяють швидко відсікати невидимі об'єкти і таким чином зменшувати кількість операцій. Техніки Level of Detail дають змогу динамічно зменшувати складність геометрії об'єктів залежно від відстані до камери, зменшуючи навантаження без суттєвої втрати візуальної якості.

Вбудовані засоби діагностики, профілювання й оптимізації, як-от NVIDIA Nsight, AMD Radeon GPU Profiler або Intel GPA, дозволяють детально відстежувати час виконання окремих етапів рендерингу, виявляти «вузькі місця» в шейдерних програмах, аналізувати ефективність кешування текстур і використання буферів.

У сфері апаратної реалізації також активно використовуються FPGA для задач рендерингу з високою спеціалізацією. Це особливо актуально для вбудованих систем, авіоніки або військових візуалізаційних платформ, де потрібна жорстка детермінованість затримок, енергетична ефективність і безпека. У таких випадках алгоритми рендерингу кодуються безпосередньо на апаратному рівні у вигляді HDL (наприклад, Verilog чи VHDL), що дозволяє створити повністю апаратну pipeline обробки сцен.

Сучасна програмно-апаратна реалізація рендерингу є багаторівневою інтеграцією високопродуктивних обчислень, інтелектуальних алгоритмів, оптимізованих апаратних ресурсів і гнучких програмних засобів. Вона постійно еволюціонує, прагнучи досягти балансу між якістю, швидкістю, енергоспоживанням та

універсальністю для найширшого спектра застосувань – від ігор до наукових симуляцій та візуалізації великих даних.

Новітні тенденції у програмно-апаратній реалізації рендерингу включають використання хмарних обчислень і розподіленого рендерингу. Це дозволяє значно збільшити обчислювальні ресурси за рахунок масштабування, що особливо актуально для кіновиробництва, архітектурної візуалізації та інженерного моделювання. У таких системах великі сцени поділяються на фрагменти (кадри, плитки, області), які рендеряться паралельно на різних вузлах із подальшим збиранням результату. При цьому використовується спеціальне програмне забезпечення (наприклад, Blender Render Farm, Autodesk Backburner, Chaos Cloud), що координує розподіл задач, контроль черг і збереження проміжних результатів.

Ще одним важливим аспектом є використання графічних рушіїв з відкритим кодом і платформою для створення кастомізованих візуалізаційних середовищ. Наприклад, рушії Godot, OGRE, або технології WebGL дають змогу гнучко реалізовувати рендеринг як у настільних, так і у веб-орієнтованих додатках. У цьому контексті важливу роль відіграє оптимізація передачі даних між CPU і GPU: ефективне керування пам'яттю, асинхронна обробка команд, стиснення буферів і використання інтерфейсів типу Uniform Buffer Object (UBO) або Shader Storage Buffer Object (SSBO).

Із розвитком віртуальної та доповненої реальності (VR/AR) виникають нові виклики до рендерингових систем. Зокрема, потрібно забезпечити наднизьку затримку (менше 20 мс) і високу частоту кадрів (90+ FPS) одночасно для обох очей. Для цього застосовуються адаптивні техніки, такі як foveated rendering (фовеальний рендеринг), коли деталізація зображення зменшується поза центральною областю зору користувача, що визначається за допомогою трекінгу очей.

Ще одним напрямом є програмно-апаратне забезпечення рендерингу в штучному середовищі – наприклад, для тренування автономних транспортних систем або роботів, які використовують віртуальні сцени для симуляції реального світу. Такі сцени генеруються з урахуванням фізично достовірного освітлення, матеріалів, глибини та динаміки, що потребує точного моделювання та великої обчислювальної потужності, яку забезпечують сучасні графічні сервери з кількома GPU та апаратними прискорювачами.

Важливо згадати про енергетичну ефективність як один із ключових параметрів сучасної апаратної реалізації. Апаратні платформи нового покоління, зокрема мобільні SoC, реалізують динамічне масштабування частоти та напруги, а також апаратне

вимикання невикористовуваних блоків GPU для економії енергії під час простою або рендерингу малонавантажених сцен.

Таким чином, програмно-апаратна реалізація рендерингу – це динамічна і міждисциплінарна область, що об'єднує досягнення комп'ютерної графіки, електроніки, інформатики та штучного інтелекту. Вона продовжує розвиватися у напрямку підвищення реалістичності, швидкодії, доступності та енергоефективності, відкриваючи нові горизонти візуального відтворення цифрового контенту.

Окрему увагу в програмно-апаратній реалізації рендерингу слід приділити інтерфейсам обміну даними між різними підсистемами обробки. Наприклад, для передачі геометрії, текстур, анімаційних даних та команд рендерингу від центрального процесора до графічного прискорювача використовуються шини типу PCI Express (PCIe), які підтримують високу пропускну здатність та низьку затримку. У високопродуктивних системах також реалізується прямий доступ GPU до оперативної пам'яті (технологія Unified Memory), що дозволяє уникнути дублювання даних у різних адресних просторах. Це особливо важливо у задачах з великими об'ємами вхідної інформації, як-от медичне сканування, візуалізація великих наукових масивів або цифрові близнюки.

З точки зору програмного забезпечення, середовище розробки графічних застосунків часто базується на рушіях з модульною архітектурою, які дозволяють компонувати елементи сцени у вигляді вузлів, шейдерів, анімаційних скриптів та ефектів постобробки. Інструменти візуального програмування, зокрема нодові редактори матеріалів та шейдерів, надають можливість створювати складні візуальні ефекти без необхідності глибокого знання коду. Це розширює доступність розробки та дозволяє поєднувати технічні та художні підходи в одному середовищі. Водночас для досягнення максимальної продуктивності та гнучкості найважливіші частини рендерингового конвеєра, як правило, реалізуються на низькорівневих мовах програмування (наприклад, GLSL, HLSL, SPIR-V), які компілюються у машинний код, оптимізований під конкретну архітектуру GPU.

На апаратному рівні сучасні графічні процесори мають складну мікроархітектуру, яка включає потокові мультипроцесори (SM – streaming multiprocessors), які містять обчислювальні ядра, шейдерні блоки, кеші, спеціалізовані функціональні блоки для інтерполяції, тесселяції, трасування променів та апаратного декодування відео. Управління цими ресурсами здійснюється за допомогою планувальників задач та драйверів, які динамічно розподіляють

навантаження, враховуючи пріоритети, черги команд та наявні ресурси. Підтримка паралельного виконання кількох графічних потоків та обчислювальних ядер дозволяє реалізовувати мультизадачність у рендерингу, наприклад, одночасну обробку декількох сцен або кадрів у реальному часі.

Перелік використаних джерел

1. Hearn, D., & Baker, M. P. (2010). *Computer Graphics with OpenGL* (4th ed.). Prentice Hall.

2. Pharr, M., & Humphreys, G. (2010). *Physically Based Rendering: From Theory to Implementation* (2nd ed.). Morgan Kaufmann.

Shreiner, D., Davis, R., & Molnar, S. (2013). *OpenGL Programming Guide* (8th ed.). Addison-Wesley.

3. Akenine-Möller, T., Haines, E., & Hoffman, N. (2018). *Real-Time Rendering* (4th ed.). CRC Press.

4. Романюк О. Н., Чорний А. В. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Вінниця: УНІВЕСУМ-Вінниця, 2006.

5. Романюк О. Н., Майданюк В. П., Бобко О. Л., Тітова Н. В., Романюк С. О. Використання GPU для розпаралелювання рендерингу. *Advanced Top Technology. Electron. Text. Data*, №3, 2024, с. 46–48.

6. Романюк О. Н., Романюк О. В., Чехмestрук Р. Ю. «Комп'ютерна графіка» (електронний навчальний посібник) Вінниця: ВНТУ, 2023, 147 с.