

**О. Н. Романюк, д-р техн. наук, проф.; О. Л. Бобко;  
А. В. Снігур, канд. техн. наук, доц.**

## **МЕТРИЧНЕ ОЦІНЮВАННЯ РОЗПАРАЛЕЛЕННЯ РЕНДЕРИНГУ**

*Рендеринг сучасних зображень вимагає значних обчислювальних ресурсів, особливо під час обробки сцен із високою деталізацією, складною геометрією та реалістичним освітленням. У зв'язку з цим актуальним стає використання паралельних обчислювальних методів, які дозволяють суттєво скоротити час візуалізації, підвищити частоту оновлення кадрів (FPS), зменшити енергоспоживання та забезпечити стабільну продуктивність, зокрема у віртуальній і доповненій реальності. У статті проаналізовано основні способи реалізації розпаралелення: за окремими кадрами, об'єктами, фрагментами зображення (тайлами) та на рівні виконання шейдерів. Висвітлено особливості апаратної підтримки таких методів у сучасних графічних процесорах, включаючи SIMD- і SIMT-архітектури.*

*У результаті проведеного дослідження визначено комплекс показників для оцінювання якості паралельного виконання рендерингу. Запропоновано використовувати низку метрик: частоту кадрів, коефіцієнт прискорення, ефективність масштабування (з урахуванням законів Амдала та Густавсона), ступінь завантаження обчислювальних одиниць, баланс розподілу навантаження між потоками, ефективність управління відеопам'яттю та рівень енергоефективності системи. Для кожного з критеріїв наведено математичні формули та пояснення методів їх визначення, зокрема розрахунок коефіцієнтів нерівномірності навантаження, витрат енергії залежно від якості паралелізації, впливу синхронізації потоків і доступу до спільної пам'яті.*

*Практична цінність роботи полягає в тому, що запропоновані критерії можуть бути використані для моніторингу продуктивності графічних систем, автоматизованого профілювання рендерерів, вибору оптимальних алгоритмів паралельної обробки та оцінювання ефективності нових апаратних або програмних технологій. Зазначено, що обмеження масштабування обумовлені як особливостями архітектури обчислювальних систем, так і наявністю синхронізаційних затримок. Матеріал статті становить інтерес для фахівців у сфері комп'ютерної графіки, розробників програмного забезпечення для візуалізації, дослідників у галузі цифрової симуляції та обробки зображень у реальному часі.*

**Ключові слова:** розпаралелення рендерингу, комп'ютерна графіка, частота кадрів, масштабованість, баланс навантаження, критерії оцінки продуктивності.

### **Вступ**

Рендеринг [1] тривимірних графічних сцен є надзвичайно трудомісткою процедурою, оскільки передбачає складні математичні обчислення для кожного пікселя зображення з урахуванням властивостей світла, матеріалів, геометрії об'єктів та їх взаємодії. Сучасні зображення часто мають дуже високу роздільну здатність, де кожен піксель вимагає окремого обчислення, що в сукупності створює потребу у виконанні мільярдів операцій на кадр. У зв'язку з цим виникає об'єктивна потреба в розпаралелюванні [1 – 8] обчислень, яке дозволяє значно зменшити час обробки зображення. Рендеринг природно схильний до розпаралелювання, оскільки обчислення кольору кожного пікселя можуть здійснюватися незалежно один від одного. Це дозволяє використовувати сучасні графічні процесори [6], які містять тисячі обчислювальних ядер, здатних одночасно обробляти велику кількість пікселів або променів. Завдяки цьому досягається суттєве прискорення візуалізації, знижується енергоспоживання та відкриваються можливості для рендеринга в реальному часі. Без розпаралелювання рендеринг високоякісних сцен стає надто повільним і неефективним для практичного використання в сучасних системах комп'ютерної графіки.

Необхідність розробки критеріїв якості розпаралелення рендерингу зумовлена потребою об'єктивно оцінювати ефективність та доцільність використання різних паралельних підходів у побудові графічних зображень. Критерії дозволяють порівнювати продуктивність різних реалізацій, виявляти втрати обчислювальної потужності через нерівномірне навантаження, затримки синхронізації чи надлишковий обмін даними між потоками. Наприклад, один із ключових параметрів – це масштабованість, яка показує, як ефективно зростає продуктивність за збільшення числа обчислювальних одиниць. Іншим важливим критерієм є балансування навантаження: поганий розподіл задач призводить до простоїв частини системи та нераціонального використання ресурсів. Також важливо враховувати ефективність пам'яті, зокрема затрати на кешування та обмін між потоками, що може істотно впливати на швидкість рендерингу. Розробка таких критеріїв є необхідною умовою для оптимізації систем рендерингу як у реальному часі. Вони забезпечують основу для автоматичного профілювання, адаптивного вибору стратегії розпаралелення, а також об'єктивної оцінки нових апаратних і програмних технологій. Таким чином, критерії якості розпаралелення є інструментом не лише для аналітики, але й для підвищення продуктивності та якості зображення в графічних системах різного рівня складності.

### Аналіз літератури

Методи розпаралелення рендерингу [1, 10 – 13] передбачають розподіл обчислювального процесу з метою скорочення часу формування зображення. Один із найуживаніших способів це поділ зображення на окремі ділянки, такі як пікселі, рядки або блоки (тайли), кожен з яких обробляється окремо. Такий метод особливо ефективний при використанні графічних процесорів, де кожен фрагмент може обчислюватися паралельно численними потоками. Ще один метод – розділення сцени на об'єкти або їх групи, які незалежно рендеряться різними обчислювальними одиницями. Це доцільно для складних сцен, але вимагає грамотного розподілу навантаження. У трасуванні променів широко застосовується паралельна обробка променів, коли кожен промінь або їх набір обробляється окремо, що забезпечує високу ефективність на багатоядерних процесорах і сучасних GPU [6]. У великих проєктах, такі як комп'ютерна анімація чи візуальні ефекти, використовують розподілене рендеринг-середовище: обчислення розподіляються між кількома комп'ютерами або серверами, які обробляють частини сцени чи окремі кадри. Паралельна обробка також реалізується на рівні окремих завдань: одночасно виконується обрахунок освітлення, анімацій, шейдерів тощо. У випадку графічних процесорів реалізується масова паралельна обробка за принципом SIMD чи SIMT [3, 4], коли велика кількість даних опрацьовується однотипними операціями. У застосуваннях для віртуальної або доповненої реальності використовують одночасне рендерення з різних позицій огляду, наприклад, для обох очей у VR. Всі описані методики спрямовані на підвищення ефективності обчислень і пришвидшення процесу побудови зображень, однак супроводжуються труднощами, пов'язаними з синхронізацією, балансом ресурсів та передачею даних. У сучасних рендерерах часто поєднують кілька підходів для досягнення оптимального результату з точки зору швидкості та якості графіки.

Чисто апаратні методи розпаралелення рендерингу ґрунтуються на можливостях обчислювального обладнання автоматично виконувати велику кількість однотипних операцій паралельно без необхідності втручання програміста. Одним із таких методів є SIMD (Single Instruction, Multiple Data), що реалізується в сучасних центральних процесорах, де одна команда одночасно обробляє кілька елементів даних. У графічних процесорах, натомість, використовується концепція SIMT, за якої множина потоків виконує однакову інструкцію над різними наборами даних, що дозволяє досягти високої ефективності під час рендерингу. У самій архітектурі GPU [3, 10, 12] закладено паралельність завдяки наявності великої кількості поточкових мультипроцесорів та спеціалізованих шейдерних блоків, таких

як вершинні, піксельні, геометричні та тесселяційні шейдери. Кожен із них функціонує незалежно, обробляючи різні аспекти сцени. У сучасних відеокартах, таких як серія NVIDIA RTX, реалізовано спеціалізовані апаратні блоки – ядра для трасування променів, які забезпечують обчислення перетинів променів з об'єктами сцени апаратним шляхом, без використання універсальних обчислювальних ресурсів. Крім того, у складі графічних процесорів є тензорні ядра, що забезпечують масово-паралельну обробку матричних операцій, які використовуються в задачах штучного інтелекту, наприклад, для підвищення якості зображення за допомогою нейромереж. У спеціалізованих системах для вбудованих або промислових застосувань можуть використовуватися FPGA або ASIC – програмовані або фіксовані апаратні рішення, в яких логіка паралельної обробки реалізується безпосередньо на рівні електронних схем. Такі підходи дозволяють досягти мінімальної затримки та високої продуктивності, особливо у системах реального часу. Усі ці методи забезпечують апаратне розпаралелення обчислень у рендерингу і не вимагають ручного керування розподілом задач, оскільки побудовані на низькорівневій оптимізації та архітектурному масштабуванні.

### **Критерії оцінювання якості розпаралелення рендерингу**

Оцінка продуктивності рендерингу [1, 10, 13] при розпаралеленні найчастіше виконується за показником кадрів за секунду (FPS – frames per second), що є індикатором швидкості візуалізації сцени. Чим більше FPS, тим плавніше і швидше відбувається рендеринг. Під час розпаралелення рендерингу очікується збільшення FPS порівняно з послідовним виконанням завдяки розподілу обчислювального навантаження між кількома потоками або ядрами. Основними показниками ефективності в цьому процесі є прискорення і ефективність. Прискорення визначається як відношення FPS, досягнутого при паралельному рендерингу, до FPS послідовного рендерингу. Ефективність, своєю чергою, оцінюється як частка цього прискорення на кількість використовуваних обчислювальних одиниць, що показує, наскільки ефективно задіяно ресурси. Практичні вимірювання проводяться як для CPU-, так і для GPU-рендерингу, з урахуванням типу сцени, складності обчислень і методів. У високореалістичних системах рендерингу FPS може зростати сублінійно за збільшення кількості потоків, оскільки зростають витрати на синхронізацію.

У реальному часі доцільно досягати стабільних 60 FPS і вище. Стабільність FPS гарантує плавність зображення, що позитивно впливає на візуальне сприйняття та комфорт користувача. Якщо частота кадрів нестабільна, зокрема з різкими падіннями, ускладнюється сприйняття руху. Це викликають втому очей або дезорієнтацію, особливо у відеоіграх або віртуальній реальності. Стабільний показник FPS покращує інтерактивність: у програмному середовищі з чутливим інтерфейсом або симуляцією затримки у відображенні реакцій на дії користувача призводять до втрати контролю. За стабільності FPS точніше реалізувати синхронізоване управління ресурсами. Стабільне навантаження на GPU або CPU дозволяє ефективніше використовувати обчислювальні потоки, планувати завантаження текстур і обробку шейдерів. Також стабільність FPS спрощує оптимізацію системи: за постійної частоти кадрів легше виявляти вузькі місця та прогнозувати продуктивність. Для розробників це дає можливість коректно налаштувати рівні деталізації, адаптивні алгоритми освітлення або динамічне згладжування без ризику перевищення бюджету кадру. У візуалізації медичних, інженерних чи наукових даних стабільний FPS також є фактором точності, оскільки стрибкоподібна зміна або артефакти при обробці 3D-сцен можуть вплинути на правильність інтерпретації. В таблиці 1 наведено оптимальні значення FPS для різних галузей.

Таблиця 1

**Оптимальні значення FPS для різних галузей**

| Застосування                            | Оптимальний FPS | Мінімально прийнятний FPS |
|-----------------------------------------|-----------------|---------------------------|
| Кіно та анімація                        | 24              | 24                        |
| Телебачення (Європа, PAL)               | 25              | 25                        |
| Телебачення (США, NTSC)                 | 30              | 30                        |
| Веб-відео                               | 30 – 60         | 24                        |
| Відеоігри (консолі)                     | 30 або 60       | 30                        |
| Відеоігри                               | 60 – 240        | 60                        |
| Віртуальна/доповнена реальність         | 90 – 120        | 72                        |
| 3D-моделювання                          | 30 – 60         | 24                        |
| Наукова візуалізація в реальному часі   | 60              | 30                        |
| Медичні візуалізації (CT/MRI, хірургія) | 60 – 90         | 30                        |

**Оптимальні значення FPS для різних галузей**

Більші значення FPS, потрібні там, де плавність руху, швидкість реакції та зменшення затримки мають важливе значення. Найбільш характерні галузі застосування таких високих частот оновлення охоплюють кіберспорт та шутери від першої особи, в яких FPS понад 120 або навіть 240 суттєво зменшує затримку між введенням і відображенням на екрані, підвищує точність прицілювання і дає гравцеві конкурентну перевагу в динамічних сценах. У віртуальній реальності та доповненій реальності мінімально необхідними вважаються 90 FPS, однак для комфорту та відсутності ривків бажані 120 і більше, оскільки низький FPS може спричинити дезорієнтацію або нудоту. В інтерактивних симуляторах і тренажерах, особливо медичних або авіаційних, де важливе точне візуальне відображення дій користувача, висока частота кадрів (90 FPS і більше) забезпечує коректний візуальний зворотний зв'язок. У реальному часі в наукових чи військових візуалізаціях, де потрібно точно відстежувати положення об'єктів або швидко реагувати на зміну даних, також необхідна висока частота оновлення, яка мінімізує затримку. Використання високочастотних панелей, таких як геймерські монітори із частотою 144 – 360 Гц, вимагає відповідно високого FPS для повного використання потенціалу дисплея, інакше плавність руху не буде помітною.

Розпаралелення рендерингу на графічних процесорах [3, 6, 12] суттєво впливає на продуктивність, зокрема на показник кадрів за секунду. У випадку рендерингу складної 3D-сцени з великою кількістю об'єктів, CPU із одним потоком забезпечує низьку швидкість – близько 15 FPS при тисячі об'єктів, яка стрімко зменшується до 0.1 FPS при 100 тисячах об'єктів через обмеження послідовної обробки. У той же час, використання GPU навіть початкового рівня (наприклад, GTX 1050 з близько 640 CUDA-ядер) дозволяє досягти до 60 FPS за невеликого навантаження, але продуктивність знижується до кількох FPS за значного зростання кількості об'єктів. Сучасні графічні процесори середнього рівня (RTX 3060 з понад 3500 CUDA-ядер) демонструють набагато вищі показники – близько 144 FPS за 1 тисячі об'єктів і 24 FPS за 100 тисяч. Топові GPU, такі як RTX 4090 з понад 16 тисячами CUDA-ядер, забезпечують масштабовану обробку з FPS, що може перевищувати 300 навіть за високої складності сцени. Це демонструє ефективність розпаралелення графічних задач, проте відповідно до закону Амдала максимальне прискорення обмежене непаралельними етапами, тому подальше зростання FPS можливе лише за мінімізації послідовних обчислень.

Масштабування [10 – 13] під час розпаралелення рендерингу – це процес збільшення продуктивності рендерингу шляхом ефективного використання додаткових обчислювальних ресурсів (процесорів, ядер, графічних процесорів або окремих комп'ютерів) для одночасного виконання частин завдання, що дозволяє скоротити час обробки зображення або відео без втрати якості. Масштабування не завжди діє лінійно. Тобто за постійного збільшення

потоків швидкість рендерингу не буде збільшуватись постійно.

Закон Амдала [10, 12] визначає прискорення роботи програми за рахунок розпаралелення її частин і розраховується за формулою:

$$S = \frac{1}{(1-P) + \frac{P}{N}}$$

де:  $S$  – прискорення,  $P$  – частина програми, яку можна розпаралелити,  $(1 - P)$  – послідовна частина програми,  $N$  – кількість процесорів або потоків.

Закон Амдала [10, 12] у контексті розпаралелення рендерингу пояснює, наскільки можливо прискорити процес виведення зображення на екран за збільшення кількості обчислювальних ресурсів, зокрема ядер графічного процесора (GPU). Цей закон базується на припущенні, що лише певна частина рендерингового процесу може бути виконана паралельно, тоді як інша частина є строго послідовною. У випадку рендерингу добре розпаралелюються такі етапи: обробка пікселів у фрагментних шейдерах, побудова зображення на основі трасування променів, застосування текстур та обчислення локального освітлення для окремих об'єктів. Натомість слабо піддаються розпаралеленню або залишаються повністю послідовними такі задачі: оновлення сцени, підготовка буферів, синхронізація між CPU та GPU, фізичне моделювання або фінальна композитинг-обробка. Закон формалізується через математичну залежність, де загальне прискорення при використанні  $N$  потоків дорівнює оберненій величині від суми послідовної частини  $(1 - P)$  та паралельної частини, поділеної на  $N$ . Таким чином, навіть при використанні тисяч потоків (CUDA-ядер), продуктивність, наприклад у вигляді FPS, буде обмежена саме послідовною часткою алгоритму. Це означає, що теоретично максимальний приріст продуктивності досягається лише тоді, коли паралелізована частина є максимальною, а послідовна – зведена до мінімуму. Для прикладу, якщо лише 80 % рендерингу паралелізується, а 20 % виконується послідовно, то навіть за ідеально нескінченної кількості потоків загальне прискорення не перевищить п'ятикратного. Таким чином, реальне масштабування у рендерингу вимагає не тільки збільшення обчислювальної потужності GPU, а й реорганізації самого алгоритму рендерингу для зменшення послідовних етапів.

Якщо етап пост обробки на 70 % виконується паралельно і використовується за допомогою 10 потоків, тоді:

$$S = \frac{1}{(1-0.7) + \frac{0.7}{10}} = \frac{1}{0.37} \approx 2.7$$

Тобто, за використання 10 потоків прискорення під час пост-обробки відбудеться лише у 2.7 рази, що вказує на необхідність зменшення послідовної частини алгоритму для підвищення ефективності.

Закон Амдала, отже, є обмежуючим чинником ефективності паралельної графіки і відіграє ключову роль у проектуванні сучасних графічних рушіїв та оптимізації продуктивності під час створення реалістичного зображення.

Закон Густавсона-Барсиса [13] описує масштабованість паралельних обчислень, виходячи з припущення, що зі збільшенням кількості процесорів або потоків можна відповідно збільшувати обсяг задачі, зберігаючи час її виконання сталим. У контексті розпаралелення рендерингу це означає, що із зростанням кількості обчислювальних ресурсів можна не просто прискорити виведення зображення, а й значно підвищити його якість – збільшити роздільну здатність, деталізацію сцен, кількість об'єктів, складність освітлення чи ефектів –

при збереженні тієї ж частоти кадрів (FPS). Формально закон Густавсона виражається формулою:

$$S_{gust} = P - \alpha \cdot (P - 1)$$

де:  $P$  – кількість процесорів,  $\alpha$  – частка послідовного коду від 0 до 1.

Розрахуємо ефективність виконання за законом Густавсона при 70 % паралельного виконання та 10 потоках:

$$P = 10, \alpha = 1 - 0.7 = 0.9$$

$$S(10) = 10 - 0.9 \cdot (10 - 1)$$

$$S(10) = 10 - 0.9 \cdot 9$$

$$S(10) = 10 - 2.7 = 7.3$$

Тобто при 10 потоках і 70 % паралельних обчислень досягаємо прискорення у 7.3 рази.

На відміну від закону Амдала [10, 12], який обмежує максимальне прискорення через послідовні ділянки, закон Густавсона дозволяє зосередитися на розширенні задачі замість зменшення її часу виконання. У рендерингу це дозволяє реалізувати більш складні сцени, багатоканальну обробку зображень, високоточне глобальне освітлення та складну фізику – все це без зниження частоти кадрів на екрані. Таким чином, закон Густавсона краще описує сучасну парадигму масштабування в реальному часі, коли пріоритетом стає не тільки швидкість, а насамперед якість та складність обчислень, які GPU здатен обробити паралельно.

Збалансована система з паралельним виконанням використовує всі елементи системи рівномірно, тобто виконується рівний розподіл задач між потоками виконання. Рівномірний розподіл задач між потоками є важливим для усунення простою і забезпечення максимальної ефективності. Нерівномірне навантаження може суттєво знизити продуктивність системи. Баланс навантаження аналізується через моніторинг рівномірності розподілу задач між потоками виконання. Іноді необхідно розрахувати відсоток завантаження багатопотокової обчислювальної системи протягом певного часу, умовна формула для розрахунку завантаження [12]:

$$U_{GPU} = \frac{T_{active}}{T_{total}} + 100\%$$

де:  $U_{GPU}$  – відсоток завантаження GPU у відсотках,  $T_{active}$  – час, протягом якого ядра були зайняті виконанням,  $T_{total}$  – загальний час спостереження.

Цей показник визначає, яку частку часу система дійсно виконує обчислення, а не простоює через очікування ресурсів, синхронізаційні бар'єри чи нерівномірне навантаження між потоками. Формально відсоток завантаження обчислюється як відношення часу активного виконання задач до загального часу аналізу. Наприклад, якщо в системі з 8 ядрами кожне ядро працювало по 7,5 секунд протягом загального інтервалу в 10 секунд, то сумарний час активності становить 60 секунд. У такому випадку завантаження розраховується як  $(60 / 80) \times 100\% = 75\%$ . Це означає, що система була ефективно задіяна протягом 75 % часу. У випадку GPU, особливо при паралельному рендерингу, відсоток завантаження часто вимірюється за допомогою спеціальних інструментів, таких як nvidia-smi, що показують,

наскільки активно графічний процесор використовується для виконання обчислень. Високий рівень завантаження вказує на добре спроектовану схему розподілу навантаження між потоками та оптимальну взаємодію між обчисленнями і доступом до пам'яті. Натомість низький рівень завантаження може свідчити про існування вузьких місць у вигляді затримок при обміні даними, надмірного використання синхронізацій або неефективного планування задач. Тому аналіз завантаження є важливим під час розробки високопродуктивних систем рендерингу, де ефективне використання апаратних ресурсів прямо впливає на швидкість формування зображення.

Іноді необхідно визначити рівень нерівномірного навантаження процесора. Для процесу рендерингу рівень нерівномірного навантаження процесора відображає, наскільки ефективно розподілене обчислювальне навантаження між усіма потоками або ядрами. Якщо частина ядер простоює або завершує обчислення раніше за інші, загальний час виконання задачі зростає через необхідність чекати завершення найповільнішого потоку. Цей показник особливо важливий у задачах паралельного рендерингу, де обчислення кадру розбиваються між ядрами.

Для визначення рівня нерівномірного навантаження [12] використовується така формула:

$$L = 1 - \frac{\sum_{i=1}^p T_{i,active}}{P \cdot T_{max,active}}$$

де:  $L$  – коефіцієнт невідповідності навантаження,  $\sum_{i=1}^p T_{i,active}$  – сумарний активний час роботи всіх ядер або потоків,  $P$  – кількість потоків,  $T_{max,active}$  – максимальний активний час роботи найбільш навантаженого ядра.

Якщо  $L = 0$  це означає найбільш ефективне балансування навантаження. Чим ближче значення  $L$  до 1 тим менш ефективний розподіл навантаження.

У контексті графічного рендерингу нерівномірність виникає, коли деякі частини зображення є складнішими для обчислення – наприклад, вимагають глибшого трасування променів або більше світлових взаємодій. Щоб зменшити дисбаланс, застосовують динамічний розподіл задач, коли потоки отримують наступні блоки роботи лише після завершення попередніх, що дає змогу краще збалансувати навантаження.

Такі метрики як відсоток завантаження та коефіцієнт невідповідності завантаження можна використовувати під час тестування рендерингу у динаміці. Таким чином можна визначити чи впливає розмір сцени на швидкість (FPS) і одночасно бачити чи ефективно використовуються обчислювальні ресурси системи.

Під час паралельного рендерингу ефективне управління пам'яттю VRAM дозволяє уникати перевантажень і забезпечувати стабільну роботу за високих обчислювальних навантажень. Потенційно можлива ситуація коли потоки GPU читають чи пишуть дані у спільну пам'ять. Це може призвести до зменшення продуктивності. Для того щоб оцінити ефективність використання пам'яті використовуються спеціальні програмні засоби такі як NVIDIA Nsight Graphics для графічних карт від NVIDIA або AMD Radeon GPU Profiler відповідно для графічних карт виробництва компанії AMD. Наприклад, якщо аналіз показує високий рівень запису або читання пам'яті але при цьому завантаження процесорів GPU низьке, це може означати проблему під час спільного доступу до однієї ділянки пам'яті багатьма потоками. Це дозволяє виявити перевантаження пам'яті і забезпечити стабільну роботу навіть за умов високих обчислювальних навантажень.

Ефективність використання пам'яті під час розпаралелення рендерингу [10, 12] є одним із

чинників, що визначає загальну продуктивність графічної системи, особливо в умовах обробки складних сцен з великою кількістю об'єктів і текстур. При поділі рендерингового процесу на паралельні потоки кожне обчислювальне ядро або блок отримує частину завдання, і від того, наскільки раціонально організовано зберігання та доступ до даних, залежить як швидкодія, так і використання апаратних ресурсів. Неефективне використання пам'яті може призвести до дублювання текстур і геометричних даних, конфліктів доступу, кеш-промахів та перевантаження пропускну здатності підсистеми пам'яті. Одним з основних принципів є уникнення надмірного дублювання даних, що дозволяє економити як глобальну, так і локальну пам'ять, забезпечуючи обмін лише тими фрагментами, які необхідні для поточного обчислення. Важливу роль відіграє використання кешованої пам'яті та спільної (shared) пам'яті в межах поточних блоків GPU, що дозволяє мінімізувати затримки при зверненні до глобальної пам'яті. Балансування навантаження між потоками забезпечує рівномірний розподіл пам'яті, уникаючи ситуацій, коли одні потоки завантажені більше, ніж інші, що негативно впливає на масштабованість. Також важливо застосовувати алгоритмічну оптимізацію доступу до даних, зокрема структури даних, орієнтовані на кеш, наприклад, плиткову обробку або ефективну організацію буфера глибини. Розумне використання просторової та часової локальності дозволяє зменшити кількість кеш-промахів, а також скоротити загальний обсяг звернень до пам'яті. У сучасних GPU реалізовані механізми перекриття обчислень і доступу до пам'яті, що за умов правильної організації рендерингового процесу дозволяє максимально завантажити обчислювальні блоки без простоїв.

Енергоефективність є важливим для мобільних та портативних пристроїв. Ефективний алгоритм повинен мінімізувати використання ресурсів, щоб продовжити тривалість роботи пристрою від акумулятора. Енергоспоживання аналізується шляхом вимірювання кількості енергії, витраченої на виконання рендерингу. Це особливо актуально для мобільних та портативних пристроїв, де оптимізація енергоспоживання сприяє продовженню часу автономної роботи. Під час паралельного рендерингу надмірне енергоспоживання може бути викликане надмірною кількістю потоків. Тому необхідно щоб під час рендерингу було задіяно рівно стільки потоків скільки необхідно для оптимальної продуктивності з мінімальними енергозатратами. Визначити, що саме надмірна кількість потоків спричиняє підвищене енергоспоживання можна за допомогою профілювання застосовуючи спеціальні програми наведені вище.

Виробники процесорів, а також відео карт GPU, застосовують критерій TDP (thermal design power), що визначає затрату енергії під час операційного навантаження. Формула для визначення енергозатрат багато потокової програми [4, 11]:

$$P = (N_{processors}) \cdot (RWatts / Processors) \cdot (T_{hours})$$

де:  $P$  – споживання енергії (кВт/год),  $N_{processors}$  – кількість процесорів,  $R$  – thermal design power;  $T$  – час виконання програми.

Для того щоб визначити, чи впливає якість паралелізації на енергоспоживання застосуємо закон Амдала:

$$S = \frac{T_{serial}}{T_{parallel}} = \frac{1}{(1-p) + \frac{p}{N_{processors}}}$$

де:  $p$  – частка програми що виконується паралельно,  $N_{processors}$  – кількість процесорів.

Повністю паралелізована програма становить  $p = 1$ , що спрощує формулу визначення прискорення:  $S = N$ .

Формула визначення часу паралельного виконання набуває вигляду:

$$T_{parallel} = \frac{T_{serial}}{N}$$

Під час реалізації рендерингу присутні затримки на синхронізацію. Також, не всі алгоритми піддаються паралелізації повністю. Введемо додатковий коефіцієнт  $\eta$ :

$$T_{parallel} = \frac{T_{serial}}{\eta \cdot N}$$

де:  $\eta$  – якість паралелізації ( $0 < \eta \leq 1$ ),  $\eta = 1$  означає ідеальну паралелізацію,  $\eta < 1$  паралелізація не ідеальна.

Змінімо формулу споживання енергії так:

$$P = N \cdot R \cdot \frac{T_{serial}}{\eta \cdot N}$$

тоді:

$$P = \frac{R \cdot T_{serial}}{\eta}$$

Видно, що чим менше значення  $\eta$  тим більше значення  $P$ , тобто чим менший рівень паралелізації – тим більше затрат енергії.

Критерії ефективності використання пам'яті під час розпаралелення рендерингу відіграють важливу роль у забезпеченні високої продуктивності та масштабованості систем візуалізації. Ефективне використання пам'яті означає мінімізацію надлишкового копіювання, зменшення латентності доступу до даних, уникнення конфліктів у кеші та забезпечення злагодженої роботи потоків без зайвих блокувань. Основні критерії включають локальність пам'яті, коефіцієнт використання пам'яті, швидкість доступу до пам'яті, обсяг міжпроцесорного обміну та ефективність кешування.

У контексті розпаралелення рендерингу, кеш-промах – це ситуація, коли процесору або графічному процесору потрібно звернутися до даних, яких немає в кеші, і тому доводиться звертатися до повільнішої пам'яті (наприклад, до оперативної пам'яті чи відеопам'яті).

Кеш-промах виникає в той момент, коли процесор або графічний процесор намагається звернутися до даних, яких немає в кеші – спеціальній швидкій пам'яті, що призначена для зберігання часто використовуваних або ймовірних до повторного використання даних. У випадку кеш-промаху система змушена звертатися до повільнішої пам'яті, наприклад до оперативної або відеопам'яті, що призводить до затримок в обробці та зниження загальної продуктивності. У рендерингу це особливо важливо, оскільки графічні обчислення часто вимагають високої швидкодії під час доступу до текстур, буферів, координат пікселів та інших ресурсів. Під час розпаралелення задачі рендерингу, коли одночасно працює велика кількість потоків, зростає ймовірність непослідовного доступу до пам'яті, що збільшує кількість кеш-промахів. Це, у свою чергу, може призвести до зростання часу генерації кадру, зменшення кількості кадрів за секунду, зниження масштабованості паралельного алгоритму та підвищення енергоспоживання. Особливо значним вплив кеш-промахів має у високонавантажених GPU-системах, де доступ до пам'яті часто визначає вузьке місце обчислювального процесу. Щоб зменшити вплив кеш-промахів, розробники застосовують різноманітні оптимізації, наприклад забезпечення локальності даних, структурований доступ до масивів, використання тайлового рендерингу, кешування текстур і зменшення кількості

унікальних звернень до пам'яті. Таким чином, кеш-промахи є ключовим фактором, що впливає на ефективність розпаралеленого рендерингу, і потребують ретельного врахування під час проєктування систем високопродуктивної графіки.

Одним із важливих показників є локальність пам'яті, яка вимірює, наскільки ефективно потоки працюють із даними, розташованими просторово або часово близько. Висока локальність зменшує кількість кеш-промахів і підвищує швидкість доступу. Це оцінюється через співвідношення між кеш-промахами та загальною кількістю звернень до пам'яті [13]

$$L = 1 - \frac{M_{cache}}{A_{mem}}$$

де  $L$  – коефіцієнт локальності,  $M_{cache}$  – кількість кеш-промахів,  $A_{mem}$  – загальна кількість звернень до пам'яті.

Другим важливим критерієм є коефіцієнт використання пам'яті  $U_{mem}$ , який показує, яка частка виділеної пам'яті реально використовується для корисних обчислень [11]

$$U_{mem} = \frac{B_{useful}}{B_{total}}$$

де  $B_{useful}$  – обсяг даних, що безпосередньо впливають на результат рендерингу,  $B_{total}$  – загальний обсяг використаної пам'яті.

Ще один показник [11] обсяг міжпроцесорного обміну або перехресного трафіку  $T_{inter}$ , що має бути мінімальним для досягнення ефективної масштабованості. Це можна оцінити як [11]

$$T_{inter} = \sum_{i \neq j} D_{ij}$$

де  $D_{ij}$  – обсяг переданих даних від процесора  $i$  до  $j$ . Зменшення  $T_{inter}$  дозволяє уникнути затримок через шину пам'яті чи мережеву інфраструктуру.

Також важливо враховувати ефективність кешування. Наприклад, для GPU, важливим критерієм є здатність об'єднувати доступи до пам'яті з різних потоків у єдину транзакцію. Його можна описати як [11]

$$C_{eff} = \frac{T_{ideal}}{T_{actual}}$$

де  $T_{ideal}$  – ідеальний (мінімальний) час доступу до пам'яті за умов повного об'єднання звернень, а  $T_{actual}$  – реальний час доступу, що враховує фрагментацію.

Нарешті, узагальненим критерієм ефективності пам'яті може бути ефективність пам'яті у рендерингу  $E_{mem}$ , яка включає співвідношення швидкості виконання до обсягу використаної пам'яті:

$$E_{mem} = \frac{FPS}{B_{total}}$$

де FPS – кількість кадрів за секунду, а  $B_{total}$  – загальний обсяг оперативної пам'яті, використаної на кадр.

Таким чином, ефективне розпаралелення рендерингу повинно не лише масштабувати обчислення, але й оптимізувати доступ до пам'яті на всіх рівнях – від кешу до глобальної пам'яті GPU або міжвузлового доступу у кластері.

Оцінити збереження реалістичності під час розпаралелення рендерингу можна за допомогою об'єктивних і суб'єктивних метрик, які дозволяють визначити, наскільки результуюче зображення після розпаралеленого обчислення відповідає еталонному зображенню. До об'єктивних методів відносять метрики подібності між зображеннями, такі як PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), FSIM (Feature Similarity Index), а також спеціалізовані метрики для HDR-рендерингу або фізично обґрунтованої візуалізації (PBR). Наприклад, PSNR використовується для вимірювання відношення між максимально можливою потужністю сигналу і рівнем шуму, викликаного похибками візуалізації: чим вище значення PSNR, тим менше відхилення від еталону. SSIM аналізує зображення за локальними контрастами, структурою і яскравістю, що дозволяє краще оцінити збереження зорового сприйняття. Водночас, об'єктивні метрики не завжди повністю узгоджуються з людським сприйняттям реалістичності, тому їх доповнюють суб'єктивними методами: проводять опитування експертів або користувачів, які порівнюють результати рендерингу за шкалами якості, натуралістичності, передачі матеріалів, освітлення й тіней. У сфері рендерингу також важливо зберегти фізичну відповідність моделей відбиття світла, коректну роботу з глобальним освітленням і точність симуляції ефектів (туман, каустика, тіньові переходи), особливо якщо використовуються методи трасування променів у реальному часі. Таким чином, оцінювання збереження реалістичності в умовах розпаралелення рендерингу базується на поєднанні математичних метрик схожості зображень, аналізу фізичної достовірності візуалізації та людського сприйняття, що дозволяє комплексно виявити втрати якості або артефакти, спричинені некоректним розподілом обчислень.

### Висновки

Метричне оцінювання паралельного рендерингу є невід'ємною частиною аналізу якості та продуктивності графічних систем. Воно дає змогу кількісно оцінити, наскільки ефективно працює розпаралелення в контексті реального часу, зберігаючи при цьому точність візуального результату. Основні метрики, такі як FPS, час рендерингу кадру, PSNR, SSIM, LPIPS і MSE, дозволяють детально оцінити вплив на зображення і системні ресурси. Важливим є також порівняння результатів до і після розпаралелення для виявлення потенційних візуальних артефактів чи втрати структури сцени. Метричне оцінювання допомагає встановити межу допустимих спрощень та оптимізацій у рендеринговому пайплайні. Окрім об'єктивних метрик, у поєднанні із суб'єктивною оцінкою воно формує повну картину впливу паралельної обробки на сприйняття користувачем. У процесі оптимізації алгоритмів ці показники дозволяють обґрунтовано обирати параметри розподілу задач між потоками або обчислювальними вузлами. Це також сприяє прогнозованості роботи рендеринг-систем у різних апаратних конфігураціях.

Таким чином, метричне оцінювання забезпечує надійний і формалізований підхід до аналізу ефективності та якості паралельного рендерингу.

### СПИСОК ЛІТЕРАТУРИ

1. Романюк О. Н., Романюк О. В., Чехмиструк Р. Ю. Комп'ютерна графіка. Вінниця : ВНТУ, 2023. 146 с.
2. Основні критерії оцінювання продуктивності та реалістичності систем кінцевої візуалізації // О. Н. Романюк та ін. Інформаційні технології і автоматизація : матеріали наук.-техн. конф. (м. Одеса, 21–22 жовт. 2021 р.). Одеса, 2021. С. 72–74.
3. Завальнюк Є. К., Романюк О. Н. Реалізація паралелізму потоків команд і даних графічних процесорів. *Сучасний стан та пріоритети модернізації науки... : матеріали міжнар. конф. (м. Рівне, 19 жовт. 2023 р.)*. Рівне : Ред.-видавн. центр МЕНУ, 2023. Ч. 3. С. 156–158.
4. Романюк О. Н., Романюк О. В., Романюк С. О. Методи розпаралелення рендерингу. *Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі: матеріали VII Міжнародної науково-практичної конференції. Ч.1. М-во освіти і науки України; Київ. нац. ун-т культури і мистецтв*. Київ : Видавничий центр КНУКіМ, 2022. С. 64–67

5. The analysis of subjective metrics and expert methods for image quality assessment / O. N. Romaniuk et al. *Intellectual Capital Is the Foundation of Innovative Development*. 2023. Vol. 3. P. 131–133.
6. Використання GPU для розпаралелювання рендерингу / О. Н. Романюк та ін. *Advanced Top Technology*. 2024. № 3. С. 46–48.
7. Романюк О. Н., Ляшенко Ю. Л., Гончарук О. П. Метод розпаралелення процедури зафарбовування в системах комп'ютерної графіки. *Наукові праці ДонНТУ*. 2010. Вип. 11 (164). С. 129–132.
8. Романюк О. Н., Бобко О. Л. Контейнеризація для розпаралелення рендерингу. *Теорія і практика сучасної науки та освіти: матеріали XIV Міжнародної науково-практичної конференції* м. Львів, 9-10 січня 2025 року. – Львів : Львівський науковий форум, 2025. С. 63–67.
9. Романюк О. Н., Бобко О. Л. Особливості програмного розпаралелення рендерингу. *Сучасний стан та перспективи розвитку науки, освіти і технологій: збірник тез доповідей міжнародної науково-практичної конференції* (Кременчук, 4 січня 2025 р.): у 2 ч. Кременчук: ЦФЕНД, 2025. Ч. 2. С. 67–68.
10. Real-Time Rendering: The Fourth Edition / T. Akenine-Möller et al. Natick (USA) : A K Peters/CRC Press, 2018. 1198 p.
11. Hwu W. W. Programming Massively Parallel Processors: A Hands-On Approach. 4th ed. Cambridge : MIT Press, 2022. 900 p.
12. Engel W. GPU Zen 3: Advanced Rendering Techniques on the GPU. Berlin : Black Cat Publishing, 2024. 500 p.
13. Grama A., Gupta A., Karypis G., Kumar V. Introduction to Parallel Computing. 2nd ed. Pearson Education, 2003. 936.

Стаття надійшла до редакції 16.06.2025.

Стаття пройшла рецензування 18.06.2025.

**Романюк Олександр Никифорович** – д-р техн. наук, професор, завідувач кафедри програмного забезпечення, e-mail: rom8591@gmail.com.

**Бобко Олексій Леонідович** – аспірант кафедри програмного забезпечення.

**Снігур Анатолій Васильович** – канд. техн. наук, доцент кафедри обчислювальної техніки. Вінницький національний технічний університет.