

УДК 004.046

*Д. П. Кучеров*, д-р техн. наук, старш. наук. співроб.  
Національний авіаційний університет  
orcid.org/0000-0002-4334-4175  
e-mail: d\_kuchеров@ukr.net;

*К. О. Моргун*, аспірант.  
Національний авіаційний університет  
orcid.org/0000-0002-0955-4750  
e-mail: mrsammamaj@gmail.com;

*Л. С. Онікієнко*, науковий співроб.  
Центральний науково-дослідний інститут озброєння та військової техніки Збройних Сил України  
orcid.org/0000-0002-2312-583X  
e-mail: 9225foxtrot@gmail.com

## ЗАСОБИ КЕРУВАННЯ ПАРАЛЕЛЬНИМИ ОБЧИСЛЕННЯМИ В ЗАВДАННЯХ КОМП'ЮТЕРНОЇ ГРАФІКИ

### Вступ

Великі обсяги обчислень, що зараз виникають при завданнях лінгвістичного аналізу, обробленні великих даних, у задачах комп'ютерної графіки, вимагають застосування програмних та апаратних засобів, які здатні їх проводити за обмежений час. На жаль, можливості підвищення показників апаратних засобів, що протягом тривалого часу відбувалося за рахунок підвищення частоти та щільності упакування логічних елементів на кристалі, вичерпалися. Настав час розвитку підходів, які основані на багато-процесорній побудові комп'ютерних систем, а з боку програмних засобів набувають розвитку принципи паралельного програмування.

### Аналіз останніх досліджень і публікацій

Обидва підходи, тобто багато-процесорна архітектура комп'ютерів та паралельне програмування, спираються на паралельну структуру системи, що дає поштовх до розвитку теорії паралельних обчислень, яку пов'язують з оптимізацією комп'ютерних розрахунків та створенням розподілених обчислювальних систем [1–3]. Певні напрацювання в комп'ютерній техніці вже використовують при обробленні тривимірних зображень, моделюванні взаємодії елементарних частинок у фізиці, аналізу та прогнозуванні розвитку хаотичних явищ, таких як погода та багато інших. Зацікавленість до такого роду обчислень пояснюється не стільки прагненням пізнання людством невідомих явищ і

виявленням їх закономірностей, скільки можливістю проведення таких розрахунків завдяки зрослому можливостям обчислювальної техніки, а також потребою оброблення великих масивів інформації.

Під паралельними обчисленнями розуміється спосіб організації комп'ютерних обчислень, за якого розробляються програми, створюють обчислювальні процеси, що виконуються одночасно [2].

Область наукових досліджень не обмежується програмними розробками, а й включає технічну складову, що містить набір комп'ютерів або сукупність процесорів, пам'яті і засоби комунікації між ними. Окремим випадком паралельних обчислень, який знайшов практичне втілення на практиці, є процесор з декількома ядрами в ноутбуку, багато-процесорний сервер або, наприклад, кластер (суперкомп'ютер).

Розрізняють два способи паралельної обробки даних — це власне паралельність обчислень і конвеєрне оброблення даних [1].

Основні ідеї реалізації даних підходів інтуїтивно зрозумілі, якщо паралельність передбачає роздільне виконання однотипної операції декількома обчислювачами з різними наборами даних.

Прикладами такого підходу є набір касових апаратів у супермаркеті, багатоканальне оброблення в техніці, алгебричні операції з елементами багатовимірних векторів у математиці. Основним недоліком цього підходу є однотипність.

На основі цих підходів розроблені готові програмні рішення для однопроцесорних систем мовами C/C++, які дозволяють застосувати інтерфейс OpenMP для проведення паралельних обчислень з указанням відповідних блоків функцій, які будуть їх проводити за умови підключення спеціальних директив компілятора. Більш того, існують засоби автоматичного розпаралелювання обчислень, наприклад, система автоматичного розпаралелювання ORC для мови C. Проблемою розроблення таких систем є аналіз залежності даних, який традиційно вирішується побудовою відповідного графа. Для розв'язання цієї задачі запропонована концепція «відкладених обчислень», згідно з якою перед обчисленнями розміщуються відповідні дані.

Очевидним підходом до паралельних обчислень є розбиття повної задачі на більш простіші задачі так, щоб усе час розрахунку був рівномірно поділений між окремими процесорами. Крім того, повинно бути введено специфічні властивості, які забезпечують виконання таких операцій. Оскільки кількість процесорів для виконання поставленого завдання є обмеженою, то відповідно й кількість обчислювальних потоків повинна бути обмеженою. Якщо обчислювальний потік обслуговує один процесор, то кількість обчислювальних потоків задається кількістю процесорів у комп'ютерній системі.

**Мета статті** — аналіз можливостей програмних засобів для керування відеоприскорювачами при створенні власних графічних додатків.

#### **Постановка завдання**

Розглядаються засоби паралельних обчислень, які дозволяють прискорити проведення обчислень в умовах великого обсягу даних на прикладі завдання рендерінгу, що виникає за необхідності створення фотореалістичних зображень.

Для вирішення цього завдання ставиться завдання проаналізувати можливості побудови апаратних та програмних засобів для забезпечення виконання головного завдання та визначити можливості щодо економії обчислювальних ресурсів під час застосування наявних засобів паралельних обчислень та

визначити напрями, які дозволять вирішити його в реальному масштабі часу.

#### **Апаратні технології паралельних обчислень**

Потік команд при паралельних обчисленнях являє собою послідовність мікрооперацій таких, як зсув, додавання, вирівнювання, нормалізація та ін., то стає необхідним виконання цих операцій окремими пристроями і відправлення результату наступному обчислювачу, здійснюючи підготовчі дії до прийому наступного набору даних. Блок мікрооперацій має назву *ступені конвеєра*. В межах одного блоку операції виконуються з вектором даних. Векторно-конвеєрний принцип побудови процесорів передбачає використання ряду функціональних блоків конвеєрного типу, у яких виконується набір векторних команд. Векторна організація команд дозволяє оперувати масивами незалежних даних, що призводить до ефективного завантаження доступних конвеєрів.

Відповідно до праці [4] такий спосіб обчислення даних застосовують у багатопроцесорних системах, обчислювальних кластерах, векторних, VLIW, суперскалярних процесорах і процесорах під конкретне завдання.

Багатопроцесорні системи мають відповідно до назви багато однотипних процесорів та можуть мати спільну або розподілену пам'ять. Обчислювальний елемент — процесор (мікропроцесор), з них побудовані персональні комп'ютери та робочі станції. Якщо пам'ять процесорів спільна, то кількість процесорів, що мають доступ до загальної пам'яті, з чисто технічних причин, не можна зробити великим. Виходом з цієї ситуації є застосування розподіленої або виділення локальної пам'яті кожному процесору. Головною перевагою такої архітектури є масштабованість, коли необхідна продуктивність реалізується шляхом додавання процесорів. Вимоги до системи визначається тільки обмеженнями на споживану потужність або вартість.

Обчислювальні кластери складаються з різномісних комп'ютерних систем, кожна з них має власну архітектуру та продуктивність, і утворюють окремий вузол. Вузли зв'язуються між собою єдиним комунікативним середовищем.

Особливістю реалізації паралельної обробки є те, що оброблювана інформація розподіляється між вузлами відносно великими порціями, інтервали часу, необхідні для передачі інформації між вузлами, на кілька порядків перевищують час передачі в межах одного вузла.

Ряд кластерів утворюють обчислювальну мережу (GRID), окремі кластери мають різне географічне розташування, дисципліни використання, належність. Цей напрям є комбінацією попередніх. З декількох процесорів і загальної для них пам'яті формується обчислювальний вузол. Якщо отриманої обчислювальної потужності недостатньо, то кілька вузлів об'єднуються високошвидкісними каналами. Унікальні можливості по продуктивності обчислювальної мережі представляє мережу Інтернет. Вона являє єдиний потужний комп'ютер, що складається з комп'ютерів. Процес організації обчислень в системі, побудованої за цим принципом, називають *метакомп'ютингом*, а комп'ютер — *метакомп'ютером*.

Основні властивості метакомп'ютера величезні ресурси, розподілена структура, динамічне управління конфігурацією, неоднорідність.

Векторні процесори виконують один потік команд за векторними інструкціями. Одна така інструкція являє собою певну арифметичну операцію для всіх процесорів, але з різними елементами вектору даних. Висока вартість таких комп'ютерів обмежує їх застосування.

Технологія VLIW-процесорів передбачає інструкції з довгими словами, що являють собою команди, які виконуються паралельно, але використовують різні дані. Паралелізм досягається на етапі компіляції або безпосередньо програмістом на асемблері. При цьому потік інструкцій один, але інструкція складається з набору команд, що оброблюються одночасно.

У суперскалярних процесорах розпаралелювання операцій відбувається на етапі виконання, що здійснюється спеціальним блоком аналізу. Такі типи архітектури компанія *Intel* застосовує процесорах класу *Pentium*, тому персональні комп'ютери та багатопроцесорні системи розробляють за цим принципом.

Процесори під конкретне завдання мають обмежене застосування із-за їх складності проектування та обмеження використання. Найчастіше всього в них використовуються апаратна реалізація проєктованих алгоритмів. Для розроблення цих процесорів використовують спеціальну мову опису апаратури — VHDL, Verilog, SystemC, JHDL та ін.

Таким чином, розроблені підходи мають бути використані для певних завдань, де вони будуть найбільш ефективні, тобто для вирішення завдань з великими обсягами даних, наявність реального масштабу часу або певного класу завдань. Але в світі розвивається і інший напрям, який орієнтований на розвиток обчислювальної здатності окремих процесорів, що дозволить ефективно використовувати обчислювальні ресурси, час та при цьому мати невелику вартість.

#### **Програмні технології паралельних обчислень**

Універсальними засобами щодо розроблення програмного забезпечення під час виконання паралельних обчислень є система програмування на основі передавання повідомлень MPI, технологія проектування систем із спільною пам'яттю OpenMP, використання директив паралельного оброблення, мови OpenCL, mpC.

Система програмування MPI використовується для програмування паралельних обчислень у багатопроцесорних системах, обчислювальних кластерах, скалярних процесорах, що мають розподілену пам'ять. При цьому програма є множиною процесів, що виконуються в заданому адресному просторі. Взаємодія між процесорами здійснюється шляхом обміну повідомлень.

Найбільш розповсюдженим засобом паралельного програмування процесорних засобів із спільною пам'яттю є технологія OpenMP. При використанні цієї технології код програми, що написаний для послідовної архітектури не змінюється, але розробник в директивах компілятора вказує які частини коду можуть бути використаними декількома процесорами.

У паралельних блоках можуть бути вказані локальні та загальні дані для всіх потоків. Препроцесор за директивами звертається до бібліотек, які реалізують паралельні обчис-

лення певними архітектурами. Технологія підтримується Visual C++. Спосіб реалізації паралелізму в технології OpenMP іноді називають «вилочним» чи «пульсуючим» та реалізується компанією *Intel* в процесорах, що підтримують технологію Hyper Threading. При програмуванні кластерів використовують гібридний підхід, згідно з яким взаємодія між вузлами здійснюється за допомогою MPI, а вирішення конкретних завдань вузлом за технологією OpenMP.

Технологія використання директив паралельного оброблення застосовує модель паралелізму за даними та обчисленнями. Дії програміста містять об'явлення розподілених масивів, вирівнювання масивів, опис різних типів віддалених даних; розподіл витків циклів поміж окремими процесорами; створення паралельних завдань; введення редуційних операцій. Програма може являти собою послідовну частину, у яку додаються спеціальні директиви препроцесора, що надають їй властивості паралелізму. Прикладом може бути DVM програмування [4].

Мова OpenCL використовується для паралельних обчислень у графічних процесорах. OpenCL забезпечує паралелізм на рівні інструкцій та на рівні даних та є здійсненням техніки GPGPU. Побудована на стандарті мови C99. Її ключовими відмінностями від C99 є відсутність підтримки вказівників на функції, рекурсії, бітові поля, масиви змінної довжини, стандартні файли заголовків; розширення мови для паралелізму: наявність векторних типів, синхронізації, функцій для Work-items/Work-Groups; кваліфікаторів типів пам'яті: `_global`, `_local`, `_constant`, `_private`, а також власного набору вбудованих функцій.

Мова mpC застосовується для здійснення обчислень на неоднорідних обчислювальних кластерах. Програма на mpC є множиною процесів взаємодіючих шляхом передавання повідомлень. Програміст не керує процесами та вузлами, він тільки задає обчислення, які повинні проводитися процесами, що складають програму. Паралелізм описується в термінах процесів та мереж, які можуть створюватися та знищуватися в ході її виконання. Врахування неоднорідності обчислювальної мережі дозволяє оцінити продуктивність кожного вузла та розподілити процеси між

ними так, щоб зменшити загальний час виконання програми. Розподіл відбувається за умови рівномірного завантаження вузлів. Програмісту дозволяється вказувати відносні обсяги обчислень мережі. Мова має засоби вимірювання завантаженості, що дозволяє ефективно використовувати кластери для рішення обчислювальних завдань.

Проведений аналіз існуючих програмних засобів вказує на існування різних засобів для виконання паралельних обчислень. Залежно від обсягів задіяних даних та процесів програміст має можливість обрати необхідний засіб, але їх можливості щодо розроблення програмного забезпечення на рівні інструкцій обмежуються використанням алгоритмів *peephole* оптимізації після генерації виконуваного коду. Наступний рівень розвитку програмних засобів слід пов'язувати зі створенням компіляторів для обчислювальних систем з розподіленими ресурсами.

### Обчислення в графічному процесорі

Найбільш розповсюдженою сферою застосування паралельних обчислень є комп'ютерна графіка, фахівці з якої звичайно оперують із тривимірними поверхнями, основою яких є поверхня полігонів (багатокутників), яку потім освітлюють. Створену поверхню трансформують у потрібну форму шляхом певних маніпуляцій з координатами вершин, що здійснюється методами лінійної алгебри. Елементарною фігурою для побудови полігональної поверхні є трикутник завдяки унікальним властивостям щодо компактності та легкості перетворень. Недоліком такої поверхні є кількість трикутників під час побудови складної поверхні та необхідність запам'ятовування послідовності трансформацій. При цьому важливими питаннями стають опис фігури та її збереження в пам'яті комп'ютера, що призводить до великого обсягу обчислень.

Економити кількість записів при опису координати фігури трикутниками можна, якщо застосовувати техніку зв'язування спільних кутів. Це дозволяє описувати квадрат не шістьма точками, а чотирма, куб не 36, а 8 точками. Збільшення складності фігури збільшує і кількість трикутників, за допомогою яких вона створюється.

Загальні вершини зв'язуються за допомогою додаткової індексації. Для запису інформації про фігуру трикутниками потрібно вказувати кількість трикутників, порядковий номер індексу кожної вершини в масиві вершин та координати вершин. При цьому формується структура, яка включає перелічені дані. Зазначена структура впаковується в контейнери формату obj та fbx. Цей підхід використовується відомими засобами візуалізації такими як Mental Ray, Render Men та ін. Формат obj забезпечує зберігання тільки геометричної інформації про об'єкт, а саме, дані полігональної мережі та тип поверхні. Файли в форматі fbx використовуються для обміну даними між додатками сцени, формат також підтримує освітлення, матеріали, камери, анімацію, а також геометричні дані.

Додатковою інформацією, яка може зберігатися в файлі даних, це нормалі та текстура. Нормаль напрямлена перпендикулярно дотичної площини до поверхні, що розглядається. Нормаль є вектором, який не перетворюється афінним матричним перетворенням  $4 \times 4$ . Вона підлягає спеціальним перетворенням замість матричних обернень. Головною метою використання нормалей є затінення, рівень якого визначається орієнтацією поверхні, яка визначає кількість світла, відбитого в напрямку очей. При цьому важливу роль має кут між нормаллю та напрямком вхідного пучка світла та кут між нормаллю та напрямком на око, який називають також *кутом огляду*. Цей кут показує частину падаючого світла, що відбивається в бік глядача. Нормаль до багатокутника визначається лицевою гранню чи за його вершинами. Нормаль до поверхні обчислюється як векторний добуток пари ребр, що утворюють поверхню, ребра подаються векторами. Напрямок нормалі залежить від напрямку обходу вершин чи від рішення користувача, який напрямок обходу вершин слід обрати. Всі ці дані дозволяють утворити об'ємну фігуру у площині.

Текстура відтворює візуальні властивості поверхні чи об'єкта, а при їх побудові накладається на мережеву поверхню цифрового зображення шляхом макетування (покриття) відповідної грані 3D-моделі. В найпростішому випадку вона характеризується яскравіс-

тю та кольором матеріалу, що покриває об'єкт. Ці параметри визначаються на кінцевому етапі вирішення задачі визначення освітлення об'єкта шляхом вирішення інтегрального рівняння Фредгольма 2-го роду [6; 7; 11; 12]

$$L(x, \omega, \lambda, t) = \int_{\Omega} L_t(x, \omega_i, \lambda, t) d\omega_i, \quad (1)$$

де  $L(\cdot)$  — кількість світла, що діє на спостерігача із точки  $x$ , яку він бачить під напрямком  $\omega$  у діапазоні довжин хвиль видимого світла  $\lambda$  у момент часу  $t$ ;  $\Omega$  — освітлювана область;  $i$  — її окрема ділянка.

Для вирішення (1) застосовують метод Монте-Карло, який дозволяє інтеграл правої частини цього рівняння замінити виразом [6; 7]

$$I_x = \frac{\lambda_2 - \lambda_1}{N} \sum_{i=0}^{N-1} S(\lambda) x(\lambda). \quad (2)$$

У виразі (2)  $N$  — обсяг вибірки, яка утворюється кількістю променів, що приймаються обраною довжиною хвилі, що знаходиться в діапазоні  $\lambda \in [\lambda_1, \lambda_2]$ ;  $S(\lambda)$  — спектральна щільність сили світла;  $x(\lambda)$  — функція спостереження кольорових координат, значення якої дорівнюють силі відбитого світла.

Величина  $S(\lambda)$  обирається зі спеціальних таблиць для визначеної довжини хвилі світла. Якщо обрано кольорове подання об'єкту, тоді слід визначати (2) для кожної кольорової координати:

$$I_x = \frac{\lambda_{2r} - \lambda_{1r}}{N} \sum_{i=0}^{N-1} S(\lambda) x_r(\lambda); \quad (3)$$

$$I_y = \frac{\lambda_{2g} - \lambda_{1g}}{N} \sum_{i=0}^{N-1} S(\lambda) x_g(\lambda); \quad (4)$$

$$I_z = \frac{\lambda_{2b} - \lambda_{1b}}{N} \sum_{i=0}^{N-1} S(\lambda) x_b(\lambda), \quad (5)$$

де  $\lambda_{1r}$ ,  $\lambda_{2r}$ ,  $\lambda_{1g}$ ,  $\lambda_{2g}$ ,  $\lambda_{1b}$ ,  $\lambda_{2b}$  — границі довжин хвиль червоного ( $r$ ), зеленого ( $g$ ) та синього кольорів ( $b$ ).

Перехід від координатної системи  $X, Y, Z$  до простору RGB здійснюється звичайним лінійним перетворенням [6; 7]

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 0,41847 & -0,15865 & -0,08284 \\ -0,09117 & 0,25243 & 0,01571 \\ 0,00092 & -0,00255 & 0,17860 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (6)$$

Оскільки виведення зображення відбувається на екран монітора, що є нелінійним пристроєм, то виникає необхідність додатко-

вої гама-корекції над кожним значенням моделі RGB у формі [6, 13]

$$C_{RGB} = \begin{cases} 12,92C_{lim}, & \text{якщо } C_{lim} \leq 0,0031308; \\ (1+a)C_{lim}^{1/2,4} - a, & \text{якщо } C_{lim} > 0,0031308, \end{cases} \quad (7)$$

де  $a = 0,55$ .

При цьому отримується модель sRGB. Отримані з (6) значення приводяться до діапазону  $[0, 1]$  та далі перетворюються до діапазону  $[0, 255]$  шляхом множення на коефіцієнт 255, результат множення округлюється до цілих чисел.

### Моделювання обчислень

Основою підходу до моделювання інтегрування за методом Монте-Карло. Рішення нескладного завдання, що складається в обчисленні площі під ділянкою гармонічної кривої методом Монте-Карло, показано на рис. 1. Відповідно до завдання 2000 точок вільно розподіляються в межах прямокутника  $[0,4] \times [0, 6]$ , крива показана синім кольором, точки, що визначають площу позначені зеленим кольором, інші точки кидання в прямокутник показані червоним кольором.

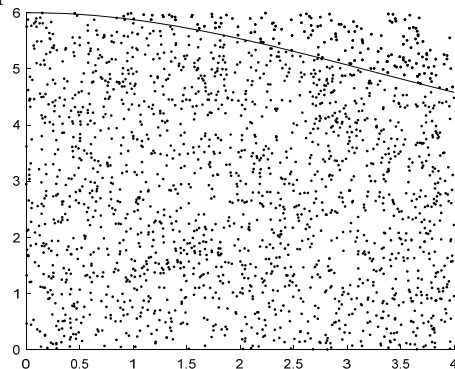


Рис. 1. Принцип знаходження інтегралу (площі кривої) методом Монте-Карло

Найбільш удалим підходом щодо розбиття повної задачі на простіші — є рівномірне поділення часу розрахунку між окремими процесорами. Крім того, повинно бути введені специфічні властивості, які забезпечують виконання таких операцій. До них відносять зернистість, за допомогою якої зображення поділяється на елементарні блоки. Розрізняють крупнозернисті та дрібнозернисті зображення, які відрізняються обсягом обчислень. При цьому зрозуміло, чим крупніше елементарні блоки, тим швидше відбувається оброблення. Явне чи неявне паралельне обчислення передбачає розроблення

шляхів паралельного обчислення або проведення розрахунків на багатопроцесорному комп'ютері. При цьому повинні бути враховані масштабованість, синхронізація та затримання процесів. Масштабованість надає можливість збільшення потужності (підключення додаткових процесорів) для вирішення початкового завдання. Синхронізація не припускає накладання обчислювальних процесів. Затримка процесів повинна забезпечити передавання даних за запитами [8].

Оскільки кількість процесорів для виконання поставленого завдання є обмеженим числом, то відповідно й кількість обчислювальних потоків повинна бути обмеженою. Якщо обчислювальний потік обслуговує один процесор, то кількість обчислювальних потоків задається кількістю процесорів в комп'ютерній системі. Таким чином й зображення повинне бути поділений на таку саму кількість частин. Різні підходи до ділення зображення на чотири частини показані на рис. 2.

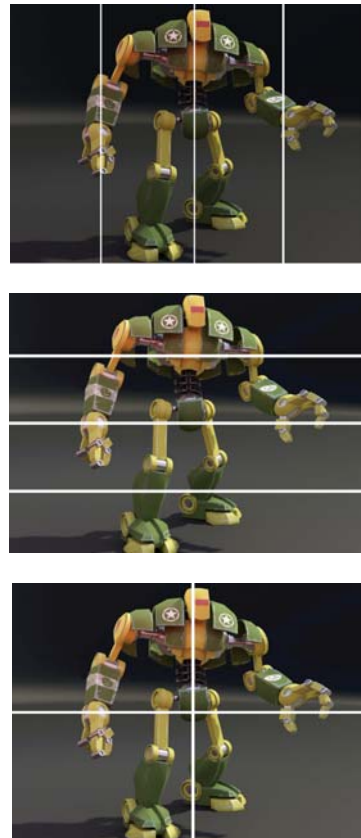


Рис. 2. Підходи до ділення зображень

Обчислювальна система будується за принципом клієнт-серверної системи. Вихідне зображення знаходиться на клієнтському процесорі, поділяється на частини, кількість

яких відповідає кількості процесорів у системі. Кожна частина зображення створює обчислювальний потік, який спрямовується для проведення розрахунків відповідних процесорів, які попередньо налагоджуються для проведення розрахунків. Після проведення розрахунків кожна частина повертається клієнтському процесору, де зображення збирається у повне.

Оцінка якості зображення проводиться за критеріями відношення сигнал/шум (PSNR) та середньоквадратичної похибки (MSE). Встановлюються граничні показники цих критеріїв  $PSNR > 20$  дБ та  $MSE < 0.001$ , які приймаються за показники якісного зображення. Якщо якість зображення не відповідає цим показникам, зображення повертається для оброблення. Зображення, що пройшло оброблення та задовольняє встановленим показникам якості повертається на клієнтський процесор для відтворення засобами відображення, дії серверних процесорів деактивуються.

З метою підвищення швидкості розрахунків кожна частина поділяється на менші за розміром та здійснюється розрахунок з більш дрібними частинами.

На рис. 3 зображено процес візуалізації зображення обчислювальними блоками ЦПП (процесор Intel Core i7 4770 - 4 ядра (8 потоків), 3.4GHz). Кожен з логічних блоків відображається у вигляді квадрату і переміщується в рамках зображення під час обчислень, що дає змогу відслідковувати прораховані області зображення, та фіксувати проміжні стадії (рис. 3) рендерингу. Обчислення відбувається в два етапи: прорахунок точок заключного збору зображення (Final gathering) та прорахунок фінального етапу. Заключний збір є методом оцінки глобальної освітленості для заданої точки зображення шляхом вибірки ряду напрямків над півкулею над цією точкою (такий набір семплів називається *заклучною точкою зібрання*), або шляхом усереднення ряду сусідніх точок (з метою оптимізації обчислень). У першому випадку орієнтація півкулі визначається нормаллю трикутника, на поверхні якого лежить точка. У прорахунках були використані такі налаштування фінальної збірки: початкова щільність точок — 0.8 на піксель; кіль-

кість променів на точку зображення — 250; інтерполяція — 30 вибірок.

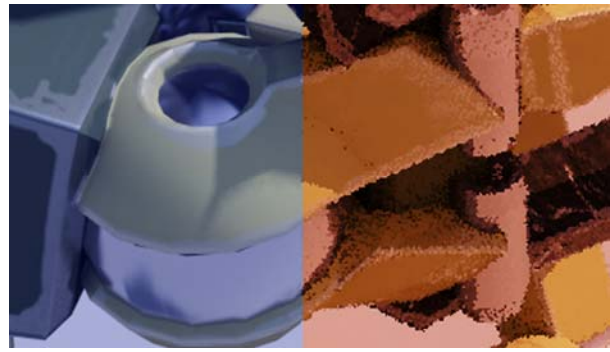


Рис. 3. Фінальне зображення (ліворуч) і проміжне (праворуч)

Природною проблемою, що виникає при розробленні паралельної системи, є оцінювання її продуктивності та ресурсоємності. Ефективність прискорення проведення обчислювальних операцій може бути оцінено за законом Амдала

$$S \leq \frac{1}{f + (1-f)/p},$$

у якому  $f$  — доля послідовних операцій в обчислювальному процесі ( $f = 0$  відповідає повністю паралельному процесу,  $f = 1$  повністю послідовному), а  $p$  — число зайнятих процесорів.

Попередній аналіз [8] показав що доля послідовних операцій в алгоритмі становить 36 %, тому очікуваний приріст ефективності під час розбиття основного процесу на 8 слід очікувати приріст ефективності в 2,27 раз. Практичне застосування дозволило встановити приріст продуктивності в 12 %. З метою досягнення більшої продуктивності від паралельної системи слід створювати спеціалізований процесор для виконання цього завдання.

### Висновки

У статті представлено основні підходи до організації великих обсягів обчислень, до яких зараз інтенсивно застосовують паралельні обчислення, детально проаналізовано одне з таких завдань, яке пов'язано з моделюванням фотореалістичних зображень засобами комп'ютерної графіки.

Забезпечення фотореалістичності вирішується збільшенням часу рендерингу при розв'язанні рівняння глобального освітлення.

Для забезпечення зменшення часу на проведення рендерінгу вихідне зображення розбивається на фрагменти, кількість яких відповідає кількості обчислювальних процесів.

Організація обчислювальних процесів, яка включала підготовку даних, синхронізацію, оформлення результату, забезпечувалась моєю паралельного програмування.

Приріст у часі склав приблизно 12 %, що не протирічить закону Амдала.

#### Перспективи подальших досліджень

Подальші дослідження слід направити на розроблення та аналіз структури спеціалізованого процесора, який буде орієнтований спеціально для виконання цього завдання.

#### ЛІТЕРАТУРА

1. **Воеводин В. В.** Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. — СПб.: БХВ-Петербург, 2002. — 608 с.
2. Параллельные вычисления [Электронный ресурс] — Режим доступа: [https://ru.wikipedia.org/wiki/параллельные\\_вычисления](https://ru.wikipedia.org/wiki/параллельные_вычисления).

3. Введение в параллельные вычисления [Электронный ресурс] — Режим доступа: <http://habrahabr.ru/post/126930>.

4. **Долинский М.** Обзор аппаратных и программных средств реализации параллельной обработки / М. Долинский, А. Толкачев // Компоненты и технологии. — № 4. — 2006. [Электронный ресурс] — Режим доступа: [http://www.kit-e.ru/assets/files/pdf/2004\\_06\\_152.pdf](http://www.kit-e.ru/assets/files/pdf/2004_06_152.pdf)

5. **Моргун К. О.** 3D моделювання анімованих персонажей / К. О. Моргун, Д. П. Кучеров // Наукоємні технології. — № 2 (26). — 2015. — С. 133–140.

6. **Kuchеров D. P.** A visualization of 3D images by Back Ray Tracing: the Geometric Approach / D. P. Kuchеров, K. O. Morgun, T. I. Golenkovskaya // Electronic and Control systems. — No. 2(48). — 2016. — P. 140–144.

7. **Morgun K. O.** Calculation of integrals by Monte Carlo in the illumination problem of synthesized objects / D. P. Kuchеров, I. V. Ogirko, O. I. Ogirko, K. O. Morgun, T. I. Golenkovskaya // Science and education a new dimension. Natural and Technical Sciences. Vol. IV (11). — Is. 96. — 2016. — P. 42–47.

8. **Моргун К. О.** Інформаційна технологія рендерінгу 3D-об'єктів / К. О. Моргун // Наукоємні технології. — № 3 (35). — 2017. — С. 189–196, doi: 10.18372/2310-5461.35.11837. (укр)

**Кучеров Д. П., Моргун К.О., Онікієнко Л. С.**

#### ЗАСОБИ КЕРУВАННЯ ПАРАЛЕЛЬНИМИ ОБЧИСЛЕННЯМИ В ЗАВДАННЯХ КОМП'ЮТЕРНОЇ ГРАФІКИ

*У роботі розглянуто засоби забезпечення прискорення обчислень, які виникають при роботі з великими обсягами даних. Встановлено, що окрім комп'ютерної графіки, така потреба виникає в задачах не тільки для задач лінійної алгебри, де дані є структурованими, але й з неструктурованими даними. Відмічається, що природним підходом до рішення цієї проблеми є використання паралельних обчислень, що може бути виконано при багатопроцесорній архітектурі обчислювальних засобів та паралельному програмуванні. Відомі підходи, які спираються на апаратну реалізацію щодо вирішення конкретних завдань є витратними з боку ресурсів та в межах окремої організаційної структури є не придатними для вирішення розрахункових завдань різного типу. У той же час розвиваються інші підходи, які дозволяють підтримувати відомі процесорні структури та існуюче для них програмне забезпечення. Підхід оснований на застосуванні паралельної структури запропоновано в роботі для рішення задачі рендерінгу, в основі якої вирішується інтегральне рівняння освітленості методом Монте-Карло. Особливістю вирішення цього завдання є перехід від загального рівняння до трьох рівнянь, відповідних прийнятій кольорової моделі зображення, а саме моделі RGB. Рішення знаходиться в просторових координатах колірності, що вимагає подальшого переходу від просторових координат до реальних координат колірності. Крім цього, в кінцевому рішенні додатково відбувається корекція нелінійності пристрою відображення. Для застосування паралельних обчислень вирішення цього завдання в роботі пропонується розбиття вихідного зображення на частини відповідно до кількості процесорів, що приймають участь в розрахунках, та забезпечення їх синхронізації відповідними програмними засобами керування. Наводиться ряд прикладів застосування методу Монте-Карло для рішення наведених математичних задач, серед яких інтегрування функції та розвиток цього підходу для роботи із зображеннями. Наводяться також результати рішення цього завдання для створення фотореалістичного зображення, а також аналізується ефективність наданого рішення за формулою Амдала.*

**Ключові слова:** паралельні обчислення; засоби програмування; засоби обчислення; комп'ютерна графіка; Монте-Карло метод.



**Kuchеров D. P., Morgun K. O., Onikienko L. S.**

## **TOOLS FOR PARALLEL COMPUTING IN COMPUTER GRAPHIC TASKS**

*The paper considers the means of ensuring acceleration of computations that arise when working with large volumes of data. It has been established that in addition to computer graphics, such a problem arises in problems not only for problems of linear algebra, where the data are structured, but also with unstructured data. It is noted that the natural approach to solving this problem is to use parallel computations, which can be performed in the multiprocessor architecture of computers and parallel programming. Known approaches based on hardware implementation to address specific tasks are costly from resources and, within a separate organizational structure, are not suitable for solving settlement tasks of different types. At the same time, other approaches are being developed to support well-known processor structures and existing software for them. The approach based on the application of a parallel structure is proposed in the paper for solving the rendering problem, which is based on the integral equation of illumination by the Monte Carlo method. The peculiarity of solving this problem is the transition from the general equation to the three equations corresponding to the adopted color model of the image, namely the RGB model. The solution is in the spatial coordinates of color, which requires a further transition from spatial coordinates to real coordinates of color. Apart from, in the final decision, the correction of the nonlinearity of the display device is additionally performed. To apply parallel calculations to solve this problem in the work, it is proposed to partition the original image into parts in accordance with the number of processors that participate in the calculations and ensure their synchronization with appropriate software management tools. A number of examples of the Monte Carlo method for solving the above mathematical problems are presented, among them the integration of the function and the development of this approach to work with images. The results of the solution of this problem are also given during the creation of a photorealistic image, as well as the analysis of the Amdahl's formula given solution.*

**Keywords:** parallel computing; programming tools; computing tools; computer graphics; Monte Carlo method.

**Кучеров Д. П., Моргун К. О., Анিকেенко Л. С.**

## **СРЕДСТВА УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМИ ВЫЧИСЛЕНИЯМИ В ЗАДАЧАХ КОМПЬЮТЕРНОЙ ГРАФИКИ**

*В работе рассмотрены средства обеспечения ускорения вычислений, которые возникают при работе с большими объемами данных. Установлено, что кроме компьютерной графики, такая потребность возникает в задачах не только для задач линейной алгебры, где данные структурированы, но и с неструктурированными данными. Отмечается, что естественным подходом к решению этой проблемы является использование параллельных вычислений, может быть выполнено при многопроцессорной архитектуре вычислительных средств и параллельном программировании. Известны подходы, опирающиеся на аппаратную реализацию по решению конкретных задач, являются затратными со стороны ресурсов и в пределах отдельной организационной структуры, является не пригодными для решения расчетных задач различного типа. В то же время развиваются другие подходы, которые позволяют поддерживать известные процессорные структуры и существующее для них программное обеспечение. Подход основан на применении параллельной структуры предложено в работе для решения задачи рендеринга, в основе которой решается интегральное уравнение освещенности методом Монте-Карло. Особенностью решения этой задачи является переход от общего уравнения до трех уравнений, соответствующих принятой цветной модели изображения, а именно модели RGB. Решение находится в пространственных координатах цветности, что требует дальнейшего перехода от пространственных координат к реальным координатам цветности. Кроме этого, в конечном решении дополнительно происходит коррекция нелинейности дисплея. Для применения параллельных вычислений решения этой задачи в работе предлагается разбиение исходного изображения на части в соответствии с количеством процессоров, участвующих в расчетах, и обеспечения их синхронизации соответствующими программными средствами управления. Приводится ряд примеров применения метода Монте-Карло для решения приведенных математических задач, среди которых интегрирования функции и развитие этого подхода для работы с изображениями. Приводятся также результаты решения этой задачи для создания фотореалистичного изображения, а также анализируется эффективность предоставленного решения по формуле Амдала.*

**Ключевые слова:** параллельные вычисления; средства программирования; средства вычислений; компьютерная графика; Монте-Карло метод.

Стаття надійшла до редакції 22.03.2018 р.  
Прийнято до друку 04.06.2018 р.  
Рецензент — д-р техн. наук, проф. Квасніков В. П.