УДК 681.3:004.93

А. А. ЯРОВИЙ, О. О. КУЛИК, І. Р. АРСЕНЮК

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ПАРАЛЕЛЬНОГО ОБРОБЛЕННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ТЕХНОЛОГІЙ OPENMP ТА NVIDIA CUDA

Вінницький національний технічний університет
21021, Хмельницьке шосе, 95, Вінниця, Україна
Тел.: +380 (432) 598243, e-mail: axa@vinnitsa.com

Анотація. В проведених дослідженнях здійснено аналіз особливостей реалізації обробки зображень на основі технологій створення багатопоточних додатків OpenMP та гетерогенних додатків NVIDIA CUDA. За результатами аналізу здійснено комп'ютерне моделювання процесу матричних паралельних обчислень та проаналізовано отримані показники швидкодії.

Аннотация. В проведенных исследованиях осуществлен анализ особенностей реализации обработки изображений на основе технологий создания многопоточных приложений OpenMP и гетерогенных приложений NVIDIA CUDA. По результатам анализа проведено компьютерное моделирование процесса матричних параллельных вычислений и проанализировано полученные показатели быстродействия.

Abstract. The analysis of the implementation details for realization of image processing on the base of technologies for developing of multithreaded applications OpenMP and heterogeneous applications NVIDIA CUDA is carried out in the research. According to the analysis the computer modeling of the matrix parallel computing and analysis of the performance indicators are carried out.

Ключові слова: паралельні обчислення, гетерогенні обчислення, багатопоточність, обробка зображень, GPGPU, CUDA, Multi-GPU Programming, OpenMP.

ВСТУП

У теперішній час одним із актуальних напрямів досліджень у галузі інформаційних технологій є створення методів та інтелектуальних систем обробки надвеликих обсягів інформації з метою відбору інформативних параметрів для їх подальшої класифікації та здійснення аналізу. Такі методи і системи знаходять застосування в задачах комп'ютерного зору, прогнозування погоди та змін клімату, моделювання складних фізичних процесів, розпізнавання зображень, синтезу мови людини та ін. Це, в свою чергу, сприяє розвитку цілого ряду технологій паралельних і розподілених обчислень, починаючи з багатопоточних додатків закінчуючи обчислювальними кластерами суперкомп'ютерного типу [1—4].

З цієї точки зору значний інтерес представляють задачі, пов'язані з системами комп'ютерного зору. Більшість таких систем розраховані на роботу в реальному часі (системи розпізнавання облич, автоматизованого керування транспортними засобами, тощо), що висуває високі вимоги до швидкодії їх функціонування. Окрім того, структура таких задач передбачає роботу з відеорядом, тобто великим набором динамічних зображень. При чому, кожне таке зображення може оброблятися, як правило, незалежно від інших, що дозволяє в повній мірі використати переваги паралельних підходів. Одним із прикладів такого підходу є паралельно-ієрархічне перетворення зображень, методика організації якого дозволяє ефективну реалізацію в паралельних обчислювальних системах різного типу.

Варто відзначити, що роботи в напрямку паралельних та гетерогенних обчислень проводять й великі компанії, наприклад, NVIDIA Corporation [5]. Окрім того, натепер наявна велика кількість засобів для досягнення паралельності: починаючи від GPGPU-платформ (NVDIA CUDA, AMD FireStream, OpenSL) закінчуючи відкритими стандартами та бібліотеками для певних мов програмування (OpenMP, Cilk, TBB та ін.). Велика кількість альтернатив дозволяє обирати найбільш ефективний засіб для кожної конкретної задачі, або навіть об'єднувати їх з метою досягнення ще більшого рівня розпаралелювання задля ще більшого приросту швидкодії.

Публікація містить результати досліджень, проведених при грантовій підтримці Державного фонду фундаментальних досліджень за конкурсним проектом Ф61/199-2015 «Методологія побудови високопродуктивних інтелектуалізованих паралельно-ієрархічних систем на основі сучасних мережевих обчислювальних комплексів з гетерогенною архітектурою».

МЕТА І ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою дослідження є підвищення швидкодії паралельного оброблення зображень за допомогою технологій OpenMP та NVIDIA CUDA, а також подальша реалізація на їх основі методів паралельно-ієрархічного перетворення.

Відповідно до мети підлягають вирішенню такі задачі:

1. Аналіз методів паралельного оброблення зображень з використанням технології створення багатопоточних додатків OpenMP.

2. Аналіз методів паралельного оброблення зображень з використанням технології гетерогенних обчислень NVIDIA CUDA.

3. Здійснення комп'ютерного моделювання на основі обраних програмних засобів.

4. Оцінювання отриманих результатів, зокрема, в контексті підвищення швидкодії оброблення зображень.

АНАЛІЗ МЕТОДІВ ПАРАЛЕЛЬНОГО ОБРОБЛЕННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ OPENMP ТА NVIDIA CUDA

З програмної точки зору робота з зображеннями, як правило, зводиться до роботи з масивами даних, у вигляді яких представляються зображення. Ще однією важливою особливістю оброблення зображень є те, що кожне зображення може оброблятися окремо. Зважаючи на те, що опрацюванню підлягає досить велика кількість зображень (зазвичай, декілька сотень, тисяч або більше), доцільним є паралельне оброблення зображень, тобто оброблення декількох зображень одночасно. Таким чином, стає можливим отримання значного приросту швидкодії без необхідності вносити значні зміни в алгоритм обробки.

Програмно такий підхід може бути реалізованим за допомогою таких технологій створення багатопоточних додатків, як OpenMP. OpenMP є інтерфейсом прикладного програмування для створення багатопоточних додатків, призначених в основному для паралельних обчислювальних систем зі спільною пам'яттю та складається з набору директив для компіляторів і бібліотек спеціальних функцій. OpenMP дозволяє легко і швидко створювати багатопотокові програми на алгоритмічних мовах Fortran і C / C ++, при чому директиви OpenMP аналогічні директивам препроцесора для мови C/C++ і є аналогом коментарів у алгоритмічній мові Fortran [6, 7].

З практичної точки зору OpenMP дозволяє створювати багатопоточні додатки на рівні центрального процесора (CPU). Робота багатопоточної програми починається з ініціалізації та виконання головного потоку (процесу), який у міру необхідності створює і виконує паралельні потоки, передаючи їм необхідні дані. Паралельні потоки з однієї паралельної області програми можуть виконуватися як незалежно один від одного, так і з пересиланням та отриманням повідомлень від інших паралельних потоків. Остання обставина ускладнює розробку програми, оскільки в цьому випадку програмісту доводиться займатися плануванням, організацією і синхронізацією посилання повідомлень між паралельними потоками. Як правило, доцільно виділяти так звані області розпаралелювання, в яких можна організувати виконання незалежних паралельних потоків [6, 7]. Для обміну даними між паралельними процесами (потоками) в OpenMP використовуються загальні змінні. При зверненні до загальних змінних в різних паралельних потоках можливе виникнення конфліктних ситуацій при доступі до даних. Для запобігання конфліктів можливо використати процедуру синхронізації (synchronization). З цієї точки зору задача оброблення зображень у системах комп'ютерного зору є добре адаптованою до реалізації на основі OpenMP, оскільки в ній можлива відсутність взаємодії між потоками.

Виконання паралельних потоків в паралельній області програми починається з їх ініціалізації. Вона полягає у створенні дескрипторів породжуваних потоків і копіюванні всіх даних з області даних головного потоку в області даних створюваних паралельних потоків. Після завершення виконання паралельних потоків управління програмою знову передається головному потоку [6, 7].

Варто відзначити, що приріст швидкодії від застосування OpenMP багато в чому залежить від технічних характеристик центрального процесора (CPU), а особливо — від кількості ядер. Окрім того, при даному підході час обробки одного зображення фактично не змінюється, а зміна вхідних даних в більшості випадків не впливає на ефективність застосування OpenMP, що може бути як перевагою, так і недоліком, в залежності від конкретної задачі [6, 7].

Обмеженість максимальної потужності окремого центрального процесора призводить до того, що при розв'язанні ряду задач доводиться шукати альтернативні рішення. Одним з таких рішень є GPGPU — технологія використання графічного процесора відеоадаптера для виконання загальних (неграфічних) обчислень. Однією з головних переваг GPU є велика кількість ядер (близько 2500 в сучасних відеоадаптерах), що дозволяє в повній мірі реалізувати переваги паралелізму. Теоретична потужність сучасних відеоадаптерів вже набагато вища, ніж у відповідних CPU [5, 8].

Природно, далеко не на всіх задачах вдається отримати високу продуктивність навіть на одному GPU. Тут можна виділити спільні системні вимоги для задач, що потребують реалізації обчислень загального призначення на GPU [9—11]: висока обчислювальна потужність; велика розмірність обчислюваного масиву; можливість незалежного обчислення цільових елементів на кожному з проходів; масиви даних, що використовуються при обчисленнях, повинні повністю поміщатися у відеопам'ять, тощо.

На даний момент технологія GPGPU реалізована в різноманітних програмно-апаратних архітектурах, серед яких варто відзначити NVIDIA CUDA, OpenCL та AMD FireStream. Найперспективнішою з них є NVIDIA CUDA, оскільки в порівнянні з аналогами має велику кількість переваг [12]:

— інтеграція в єдиний простір з іншими технологіями від NVIDIA, наприклад, NVIDIA SLI,

— підтримка Multi-GPU Programming,

— високий рівень інноватівності в порівнянні з аналогами,

— порівняно висока ефективність транзакцій між пам’яттю CPU та GPU,

— наявність розподіленої пам’яті (shared memory), що дозволяє значно пришвидшити роботу з пам’яттю,

— наявність єдиної віртуальної пам’яті (починаючи з CUDA 4.0) та єдиної пам’яті (починаючи з CUDA 6.0).

До недоліків CUDA варто віднести підтримку лише відеоадаптерів від NVIDIA та неповну підтримку стандарту C, що в свою чергу накладає певні обмеження.

Обробка зображень і робота з матрицями традиційно вважаються одними з найкращих задач для застосування GPGPU-обчислень, оскільки дуже велика кількість незалежних операцій з низкою обчислювальною складність дозволяє максимально повно використати усі обчислювальні ядра відеоадаптера. Втім, варто відзначити, що приріст швидкодії буде відрізнятись в залежності від розмірності вхідного зображення та алгоритму його обробки. Це обумовлено тим, що копіювання з пам’яті CPU у пам'ять GPU і навпаки є досить повільним процесом (вузьким місцем), і тому, наприклад, у випадку невеликих зображень може статися так, що витрати на переміщення даних будуть перевищувати виграш у часі за рахунок більш швидких обчислень.

Наступним логічним кроком при обробленні зображень за допомогою GPGPU технологій є спроба одночасного використання потужностей декількох відеоадаптерів. Вирішення цієї задачі значно спрощується тим, що платформа NVIDIA CUDA вже містись в собі засоби для реалізації подібних обчислень, а саме концепцію Multi-GPU Programming.

Multi-GPU Programming — набір засобів, спрямованих на одночасне використання декількох відеоадаптерів при виконанні CUDA-обчислень. Такий підхід забезпечує підвищення швидкодії та, з певними обмеженнями, збільшення розміру доступної для обчислень пам’яті. Окрім того, варто відзначити високу масштабованість даної технології. За потреби, обчислювальна потужність Multi-GPU системи може бути збільшена шляхом підключення додаткових відеоадаптерів, які можливо розмістити як в межах однієї комп’ютерної системи, так і в межах декількох, об’єднаних в обчислювальну мережу [13—16].

Зважаючи на вищевказане, можна виділити два варіанти реалізації технології Multi-GPU Programming: як на основі окремої комп’ютерної системи з декількома GPU; так і на основі обчислювальної мережі з декількох комп’ютерних систем.

Кожен з цих підходів має свої переваги та недоліки, але головним критерієм при виборі все ж залишається наявність потрібних апаратних засобів та відповідного програмного забезпечення. В даному дослідженні головну увагу приділено, зважаючи на наявні програмні та апаратні ресурси, саме першому варіанту, а саме обчисленням в рамках одного комп’ютера з двома відеоадаптерами. Але, оскільки обчислювальне навантаження може бути розподілене між двома відеоадаптерами різними способами, постає питання про вибір найбільш доцільного підходу.

У контексті задачі оброблення зображень можливо виділити два принципово відмінних варіанти: оброблення кожного окремого зображення за допомогою двох відеоадаптерів шляхом розподілення навантаження між ними або виділення кожному відеоадаптеру свого власного зображення для обробки, тобто робота в два потоки. Перший варіант в більшості випадків є недоцільним для реалізації в зв’язку з необхідністю постійного обміну даними між відеоадаптерами, що виникає в силу особливостей типових алгоритмів обробки зображень.

Робота в два потоки за своє структурою подібна до структури багатопоточних програм, але зберігає в собі усі переваги GPGPU обчислень. Оскільки, в більшості випадків зображення можуть оброблятися окремо один від одного, то при такій організації обчислень майже відсутня потреба в обміні даними між відеоадаптерами, що позитивно відзначається на швидкодії.

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ПАРАЛЕЛЬНОГО ОБРОБЛЕННЯ ЗОБРАЖЕНЬ

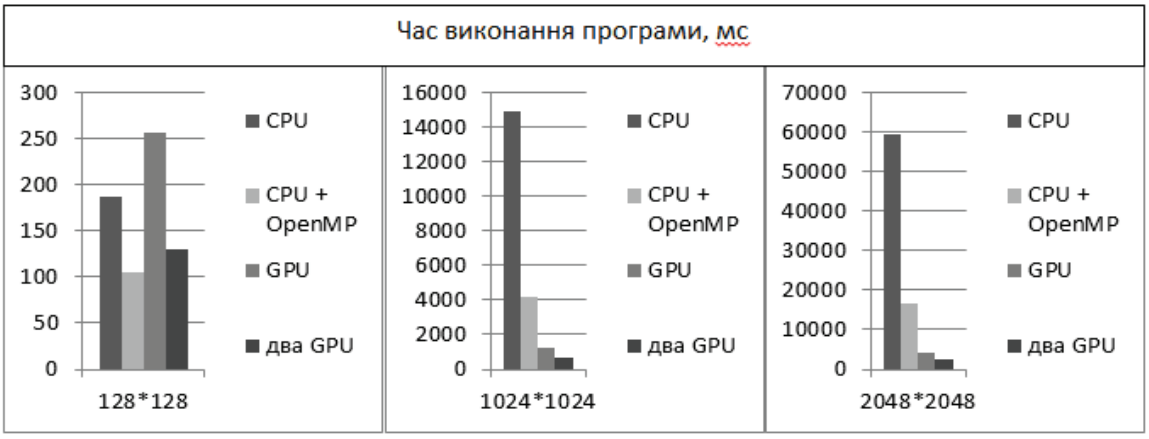
З метою перевірки вищенаведених тверджень та оцінки приросту швидкодії було здійснено комп'ютерне моделювання процесу паралельних обчислень. В якості задачі було обрано роботу з масивами чисел на основі базових арифметичних операцій, оскільки така задача подібна до типових задач оброблення зображень за математичними засобами та формами представлення інформації, але набагато легша в реалізації та перевірці результатів.

Тестову програму реалізовано на мові програмування C++ в середовищі Microsoft Visual Studio з використанням NVDIA CUDA та OpenMP. Початковими даними є набір зі 100 зображень, представлених у вигляді 200 числових масивів однакової розмірності, при чому кожному зображенню відповідає два вхідних масиви. В ході виконання програми кожна пара масивів додається між собою, а отриманий результат обробляється на основі багатократного виконання елементарних арифметичних операцій (додавання, віднімання, тощо). Це зроблено для підвищення навантаження на графічний та центральний процесори з метою підвищення точності результатів при збереженні простого режиму роботи з пам'яттю. Розроблена програма працює в чотирьох режимах: виконання на CPU, виконання на CPU з OpenMP, виконання на GPU, виконання на двох GPU. Такий підхід дозволяє наочно побачити різницю в часі виконання між обраними методами і оцінити отриманий приріст швидкодії. Для оцінювання впливу зміни розмірності вхідних зображень на швидкодію програми, було проведено тестування з масивами наступної розмірності: 2^7, 2^{10} та 2^{11}, що співвідносяться з зображеннями розмірністю 128×128, 1024×1024 та 2048×2048 пікселів відповідно. З метою зменшення похибки експеримент був проведений 10 разів. Отримані результати виглядають наступним чином (табл. 1). Графічне представлення результатів комп'ютерного моделювання наведено на рис. 1, зокрема, рис. 1(a) – гістограма для зображень розмірності 128×128 пікселів (CPU — 0,18 с; CPU + OpenMP — 0,1 с; GPU — 0,25 с; два GPU — 0,13 с); рис. 1(b) — гістограма для зображень розмірності 1024×1024 пікселів (CPU — 14,8 с; CPU + OpenMP – 4,1 с; GPU —1,2 с; два GPU — 0,6 с); рис. 1(c) — гістограма для зображень розмірності 2048×2048 пікселів (CPU — 59,5 с; CPU + OpenMP — 16,4 с; GPU — 4,2 с; два GPU — 2,2 с).

Таблиця 1.

Результати комп'ютерного моделювання

Час роботи програми / Розмірність в обраному режимі / зображення	2^7 (128×128)	2^{10} (1024×1024)	2^{11} (2048×2048)
Час роботи програми при реалізації на CPU, мс	187	14882	59592
Час роботи програми при реалізації на CPU з використанням OpenMP, мс	105	4165	16489
Час роботи програми при реалізації на GPU, мс	256	1232	4290
Час роботи програми при реалізації на двох GPU, мс	131	655	2294



a) b) c)

Рис. 1. Результати комп'ютерного моделювання:

a) гістограма для зображень розмірності 128×128 пікселів; b) гістограма для зображень розмірності 1024×1024 пікселів; c) гістограма для зображень розмірності 2048×2048 пікселів

Здійснення відлагодження, оптимізації та перевірки результатів роботи паралельних програм зазвичай є набагато складнішим у порівнянні зі звичайними програмами. Через це виникає потреба у додаткових засобах, які б спрощували роботу програміста та дозволяли перевірити коректність роботи програми. Одним з таких засобів є програмне забезпечення NVIDIA Nsight. Nsight надає широкий спектр можливостей для профілювання та налагодження гетерогенних програм, дозволяє оптимізувати продуктивність програми, прослідкувати за поведінкою системи та всіх її операцій [17]. Саме використання Nsight для моніторингу виконання розробленого програмного продукту для моделювання на двох GPU підтвердило повноцінне використання двох графічних карт (рис. 2). Так, розділ CUDA Devices вказує на те, що в ході виконання програми були використані «device 0» та «device 1», тобто два з чотирьох наявних графічних процесорів, що відповідає поставленій задачі. В розділі CUDA Contexts відображається виклик двох kernel-функцій, кожна з яких була виконана на окремому GPU.

Результати роботи розробленого програмного продукту для моделювання вказують на значне збільшення часу обробки зображень на основі CPU при переході до зображень великої розмірності (2048×2048), що підтверджує доцільність пошуку альтернативних способів організації обчислень. Так, застосування OpenMP на основі 4-ядерного центрального процесора дозволило досягнути приросту швидкодії в 3,6 разів при обробленні великорозмірних зображень у порівнянні зі звичайним виконанням на CPU. При роботі з невеликими зображенням (розмірність 128×128 пікселів) приріст становить лише 178 %, що пояснюється відносно невеликою обчислювальною складністю таких зображень у порівнянні із накладними витратами на створення потоків. Також, варто відзначити, що починаючи із зображень розмірністю 1024×1024 пікселів відносний приріст швидкодії від OpenMP залишається сталим.

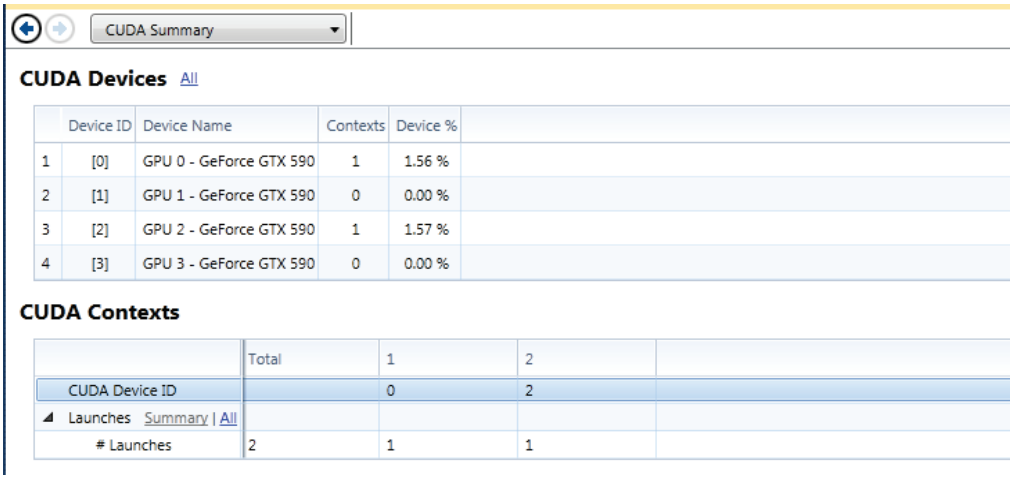


Рис. 2. Моніторинг роботи програмного продукту за допомогою NVIDIA Nsight

Реалізація оброблення зображень на основі одного відеоадаптера виявилась доцільною лише для зображень з розмірністю 1024×1024 та вище. Нижче цієї межі спостерігається втрата швидкодії до 23 %, що пояснюється великими накладними витратами на копіювання даних в порівнянні з виграшем від прискорених обчислень. Однак, при роботі із зображеннями розмірністю 1024×1024 пікселів швидкодія збільшується у 12 разів в порівнянні з CPU, а при розмірності 2048×2048 пікселів — у 14 раз, що підтверджує доцільність використання GPGPU обчислень при роботі із зображеннями великої і надвеликої розмірності.

Застосування додаткового відеоадаптера дозволило підвищити швидкодію в 1,9 разів в порівнянні з одним GPU. Приріст швидкодії в порівнянні з CPU при розмірності зображення 2048×2048 пікселів досяг 25 раз, що наочно демонструє переваги гетерогенних обчислень.

ВИСНОВКИ

Проведені дослідження показують доцільність і високу перспективність застосування технології паралельних обчислень для задач обробки зображень. За результатами експериментальних досліджень при роботі із зображеннями великої розмірності (2048×2048 пікселів) найбільш ефективним виявилось використання двох відеоадаптерів за допомогою технології NVIDIA CUDA, що дозволило досягти приросту швидкодії в 25 разів в порівнянні з реалізацією на CPU. До переваг такого підходу варто також віднести високу масштабованість, що дозволяє реалізувати необхідні обчислення на основі трьох, чотирьох або більше відеоадаптерів. Такі результати підтверджують перспективність подальшого

використання технологій гетерогенних обчислень для реалізації методів паралельного оброблення зображень, у тому числі паралельно-ієрархічного перетворення.

При роботі ж із зображеннями малої розмірності (128×128 пікселів) найбільш доцільним виявилось застосування технології створення багатопоточних додатків OpenMP, що привело до приросту швидкодії в 1,7 рази. При більших розмірностях зображень OpenMP показує гірші результати у порівнянні з технологіями гетерогенних обчислень, хоча і дозволяє збільшити швидкодію в 3,6 рази в порівнянні з CPU.

Отримані результати вказують на необхідність комплексного застосування існуючих технологій паралельних обчислень в залежності від умов конкретної задачі та наявних апаратних ресурсів. Варто відзначити перспективність комбінованого застосування різних технологій в універсальних системах, які працюють з різними обчислювальними алгоритмами та розмірностями зображень.

СПИСОК ЛІТЕРАТУРИ

1. Katsushi Ikeuchi Computer Vision. A Reference Guide. / Katsushi Ikeuchi — Springer, 2014. — 898 p.
2. Richard Szeliski Computer Vision. Algorithms and Applications. / R. Szeliski — Springer, 2011. — 643 p.
3. Шапиро Л. Компьютерное зрение. / Шапиро Л., Стокман Дж. — М. : Бином. Лаборатория знаний, 2009. — 760 с.
4. A. Yarovyy Organization of High-Performance Parallel-Hierarchical Computing Processes for Classification of Laser Beam Images. / A. Yarovyy, L. Timchenko, N. Kokriatskaia, S. Nakonechna, M. Mateichuk — Development and application systems : Proceedings of the 12th International Conference on DAS-2014, May 15—17, 2014, Suceava, Romania — Suceava, Universitatea Stefan cel Mare Suceava, 2014 — p. 192—197.
5. NVIDIA — World Leaderin Visual Computing Technologies [Електронний ресурс] — Режим доступу : http://www.nvidia.ru/page/home.html.
6. Левин М. А. Параллельное программирование с использованием OpenMP [Електронний ресурс]. — Режим доступу : http://www.intuit.ru/studies/courses/1112/232/info.
7. The OpenMP® API specification for parallel programming [Електронний ресурс] — Режим доступу : http://openmp.org/wp.
8. M. Galloy CPU vs GPU performance [Електронний ресурс] / M. Galloy // Режим доступу : http://michaelgalloy.com/2013/06/11/cpu-vs-gpu-performance.html.
9. Гергель В. П. Высокопроизводительные вычисления для многоядерных многопроцессорных систем. / Гергель В. П. — Н. : ННГУ им. Н.И.Лобачевского, 2010. — 421 с.
10. Яровий А. А. Прикладні аспекти і перспективи побудови кластерів на основі GPU для реалізації паралельної та паралельно-ієрархічної обробки інформації / А. А. Яровий // Оптико-електронні інформаційно-енергетичні технології. — 2009. — № 2 (18). — С. 119—126.
11. Скрибцов П. В. Сравнение производительности графических ускорителей и центрального процессора при вычислениях для больших объемов обрабатываемых данных / Скрибцов П. В., Долгополов А. В. // Нейрокомпьютеры: разработка, применение — Москва, Издательство «Радиотехника», 2007. — № 9. — С. 421—425.
12. NVIDIA CUDA — Неграфические вычисления на графических процессорах [Електронний ресурс] — Режим доступу : http://www.ixbt.com/video3/cuda-1.shtml.
13. M. Colgrove Multi-GPU Programming Using CUDA Fortran, MPI, and GPUDirect [Електронний ресурс] — Режим доступу : https://www.pgroup.com/lit/articles/insider/v3n3a2.htm.
14. P. Micikevicius NVIDIA. Multi-GPU Programming. [Електронний ресурс] — Режим доступу : http://www.nvidia.com/docs/IO/116711/sc11-multi-gpu.pdf.
15. Перспективи застосування технології NVIDIA SLI для паралельно-ієрархічної обробки зображень / Яровий А. А., Кулик О. О. : Збірник тез доповідей VII Міжнародної науково-технічної конференції [Фотоніка ОДС-2015)], (Вінниця, 21—23 квітня 2015 р.) — Вінниця, ВНТУ, 2015. — с. 10.
16. Свідоцтво про реєстрацію авторського права на твір № 62201. Комп'ютерна програма «Комп'ютерна програма обробки зображень у GPU-орієнтованому апаратному забезпеченні на основі модифікованого для Multi-GPU Programming методу прямого паралельно-ієрархічного перетворення» / Яровий А. А., Кулик О. О., Матейчук М. С. Дата реєстрації Державною службою інтелектуальної власності України 23.10.2015.
17. NVIDIA Nsight [Електронний ресурс] — Режим доступу : http://www.nvidia.com/object/nsight.html.

SPISOK LITERATURI

1. Katsushi Ikeuchi Computer Vision. A Reference Guide./ Katsushi Ikeuchi — Springer, 2014. — 898 p.
2. Richard Szeliski Computer Vision. Algorithms and Applications./ R. Szeliski — Springer, 2011. — 643 p.
3. Shapiro L. Kompyuternoe zrenie./ Shapiro L., Stokman Dzh. — M.: Binom. Laboratoriya znaniy, 2009. — 760 s.
4. A. Yarovyy Organization of High-Performance Parallel-Hierarchical Computing Processes for Classification of Laser Beam Images. / A. Yarovyy, L. Timchenko, N. Kokriatskaia, S. Nakonechna, M. Mateichuk — Development and application systems : Proceedings of the 12th International Conference on DAS-2014, May 15-17, 2014, Suceava, Romania — Suceava, Universitatea Stefan cel Mare Suceava, 2014 — p. 192—197.
5. NVIDIA — World Leaderin Visual Computing Technologies [Electronic resource] — Mode of access: http://www.nvidia.ru/page/home.html.
6. Levin M. A. Parallelnoe programmirovanie s ispolzovaniem OpenMP [Electronic resource] — Mode of access: http://www.intuit.ru/studies/courses/1112/232/info.
7. The OpenMP® API specification for parallel programming [Electronic resource] — Mode of access: http://openmp.org/wp.
8. M. Galloy CPU vs GPU performance [Electronic resource] / M. Galloy // Mode of access: http://michaelgalloy.com/2013/06/11/cpu-vs-gpu-performance.html.
9. Gergel V. P. Vyisokoproizvoditelnyie vyichisleniya dlya mnogoyadernyih mnogoprotsessornyih sistem. / Gergel V. P. — N. : NNGU im. N.I.Lobachevskogo, 2010. — 421 s.
10. Yarovyi A. A. Prykladni aspekty i perspektyvy pobudovy klasteriv na osnovi GPU dlia realizatsii paralelnoi ta paralelno-iierarkhichnoi obrobky informatsii / A. A. Yarovyi // Optyko-elektronni informatsiino-enerhetychni tekhnolohii. — 2009. — №2 (18). — S. 119—126.
11. Skribtsov P. V. Sravnenie proizvoditelnosti graficheskih uskoriteley i tsentralnogo protsessora pri vyichisleniyah dlya bolshih ob'emov obrabatyivaemyih dannyih / Skribtsov P. V., Dolgopolov A. V. // Neyrokompyuteryi: razrabotka, primenenie — Moskva, Izdatelstvo «Radiotehnika», 2007. — # 9. — S. 421—425.
12. NVIDIA CUDA — Negraficheskie vyichisleniya na graficheskih protsessorah [Electronic resource] — Mode of access: http://www.ixbt.com/video3/cuda-1.shtml.
13. M. Colgrove Multi-GPU Programming Using CUDA Fortran, MPI, and GPUDirect [Electronic resource] — Mode of access: https://www.pgroup.com/lit/articles/insider/v3n3a2.htm.
14. P. Micikevicius NVIDIA. Multi-GPU Programming. [Electronic resource] — Mode of access: http://www.nvidia.com/docs/IO/116711/sc11-multi-gpu.pdf.
15. Perspektyvy zastosuvannia tekhnolohii NVIDIA SLI dlia paralelno-iierarkhichnoi obrobky zobrazhen / Yarovyi A. A., Kulyk O. O. : Zbirnyk tez dopovidei VII Mizhnarodnoi naukovo-tekhnichnoi konferentsii [Fotonika ODS-2015)], (Vinnytsia, 21—23 kvitnia 2015 r.) — Vinnytsia, VNTU, 2015. — s. 10.
16. Svidotstvo pro reiestratsiiu avtorskoho prava na tvir № 62201. Komp'iuterna prohrama «Komp'iuterna prohrama obrobky zobrazhen u GPU-oriientovanomu aparatnomu zabezpechenni na osnovi modyfikovanoho dlia Multi-GPU Programming metodu priamoho paralelno-iierarkhichnoho peretvorennia» / Yarovyi A. A., Kulyk O. O., Mateichuk M. S. Data reiestratsii Derzhavnoiu sluzhboiu intelektualnoi vlasnosti Ukrainy 23.10.2015.
17. NVIDIA Nsight [Electronic resource] — Mode of access: http://www.nvidia.com/object/nsight.html.

Надійшла до редакції 20.11.2015 р.

ЯРОВИЙ АНДРІЙ АНАТОЛІЙОВИЧ — д.т.н., професор, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, Україна.

КУЛИК ОЛЕКСАНДР ОЛЕКСАНДРОВИЧ — магістрант кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, Україна.

АРСЕНЮК ІГОР РОСТИСЛАВОВИЧ — к.т.н., доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, Україна.