

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

УДК 681.324:621.383.8

А. А. Яровий, канд. техн. наук, доц.

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕРЕЖНОЇ МОДЕЛІ ПАРАЛЕЛЬНО- ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

Розглянуто теоретичні обґрунтування мережного методу паралельно-ієрархічного перетворення на рівні моделей. Здійснено імітаційне моделювання нейроподібних паралельно-ієрархічних систем на основі технологій CPU і GPGPU та їх варіантний аналіз з подальшою розробкою програмного забезпечення для задач оброблення та порівняння зображень.

Вступ

В роботі розглядаються основні положення нейробіологічної інтерпретації паралельно-ієрархічних систем, принципи функціонування яких містять високий ступінь паралелізму обробки інформації на основі просторово-часової взаємодії конвергентних (інтегруючих) та дивергентних (відокремлюючих) структур. Мережне перетворення, покладене в основу функціонування паралельно-ієрархічних систем, є одним із шляхів реалізації паралелізму і дозволяє формалізувати процедури паралельної взаємодії багаторівневої організації обчислень у часі на різних рівнях ієрархії [1].

Актуальність цих досліджень найхарактерніша для задач високопродуктивних обчислень над великих масивів інформації різної природи, у яких необхідно здійснювати складну обробку й фільтрацію сигналів, наприклад, розпакування та обробка стислих відеоданих з великою роздільною здатністю, динамічна маршрутизація інформаційних потоків у реальному часі, прогнозування динамічних швидкозмінних даних та оперативна реакція і прийняття рішень у реальному часі, що вимагає застосування досить продуктивних інтелектуальних обчислювальних систем. Все вищесказане окреслює різноманітні галузі застосування мережного методу паралельно-ієрархічного перетворення, на основі якого розробляються обчислювальні структури паралельної пам'яті; цифрових систем приймання-передавання інформації; цифрових пристроїв ущільнення інформації; систем попередньої обробки та порівняння багатоградіційних зображень, у тому числі кореляційного порівняння; кодування; формування ознак для розпізнавання, обробки біомедичної інформації тощо [1, 2].

Метою роботи є імітаційне моделювання нейроподібних паралельно-ієрархічних систем на основі технологій CPU і GPGPU з подальшою розробкою програмного забезпечення для емуляції паралельних та паралельно-ієрархічних обчислень у вирішенні задач цифрової обробки інформації, зокрема, розпізнавання та порівняння плямових зображень лазерного променя.

Відповідно до мети, вирішенню підлягали такі задачі:

1. Проведення математичного та імітаційного моделювання прямого паралельно-ієрархічного перетворення в контексті дослідження особливостей його мережної моделі.
2. Розробка алгоритмів та програмної бібліотеки для виконання паралельно-ієрархічного перетворення на основі центрального процесора (CPU) як базової апаратної платформи.
3. Розробка алгоритмів та програмної бібліотеки для виконання паралельно-ієрархічного перетворення на основі графічного відеоадаптера (GPU) як базової апаратної платформи.
4. Розробка та застосування паралельно-ієрархічної мережі для вирішення актуальної прикладної задачі обробки та порівняння плямових зображень лазерних пучків.

Постановка проблеми. Теоретичні обґрунтування та аналіз нейроподібного мережного методу паралельно-ієрархічного перетворення на рівні моделей

Досліджуваний метод [1, 3] пірамідальної обробки даних полягає в тому, що починаючи з вихідних елементів інформаційного поля $x_0(i, j)$, які підлягають кодуванню, визначається елемент інформаційного поля із мінімальним розміром $x_1(i, j)$. В термінах обробки зображень — це не що інше як «низькочастотна версія зображення». Ця низькочастотна версія зображення з частотою зрізу f_1 може розглядатися як прогнозування для $x_0(i, j)$. Далі описана вище операція ітеративно повторюється. Тобто, з $x_1(i, j)$ вибирається елемент інформаційного поля з мінімальним розміром $x_2(i, j)$. Після n ітерацій утворюється послідовність поточних інформаційних полів, обумовлених виразом $x_{i-1}(i, j) — x_1(i, j)$. На кожній ітерації розмірність поточних інформаційних полів скорочується в кількість разів, що дорівнює n_i/n_{i+1} , де n_i і n_{i+1} — відповідно, кількість ненульових елементів інформаційного поля. Якщо ці поточні інформаційні поля розглядати як поставлені одне на інше, то в результаті утвориться пірамідальна структура даних [1, 2]. Пірамідальна структура обробки зображення на будь-якому рівні цієї структури являє собою результат згортки двох гауссоподібних функцій із вихідним зображенням. Зображення гарної якості у разі пірамідального кодування утворюється з середнім коефіцієнтом ущільнення біля 10. Якщо встановити вищі коефіцієнти ущільнення (до 70), то отримуємо значні перекручування від послідовних децимацій [1, 4]. Дані перекручування можна усунути за рахунок збільшення обсягів обчислень. Таким чином, інформаційне поле у разі пірамідального кодування описується упорядкованою послідовністю його елементів, що розташовуються звичайно одне над іншим. За такого подання інформаційне поле розбивається на рівні блоки, потім процедура розбивки повторюється для кожного блоку доки його розмір не стане рівним розміру елемента початкових інформаційних полів. Регулярність пірамідальних структур визначає їх зручну реалізацію й ефективне використання — з одного боку, а з іншого боку — ієрархія описів різноманітного ступеня спільності сприяє контекстно-незалежній структуризації даних. Ця структуризація може бути основою для моделювання індуктивних і дедуктивних процесів людського сприйняття [1].

Математична модель пірамідального розкладання множини $\mu = \{ a_i \}$, $i = \overline{1, n}$ має вигляд [1, 2]:

$$\sum_{i=1}^n a_i = \sum_{j=1}^R \left(n - \sum_{k=0}^{j-1} n_k \right) (a^j - a^{j-1}), \quad (1)$$

де $a_i \neq 0$, R — розмірність множини; S ($S = 1, 2, 3 \dots$) — непусті множини елементів, що задають інформацію.

З однакових елементів сформуємо підмножини, елементи однієї підмножини позначимо через a^k , $k = \overline{1, R}$; n_k — кількість елементів у k -й підмножині (тобто, кратність числа a_k); a^j — довільний елемент множини $\{a^k\}$, обраний на j -му кроці, $j = \overline{1, R}$; $a^0 = 0$; $n_0 = 0$. Аналізуючи вираз (1) пірамідальної обробки числової інформації можна зробити висновок про те, що у процесі обробки з кожним кроком кількість чисел зменшується (якщо множини, отримані після кожного кроку, поставити послідовно одна на іншу, то утворений ними тривимірний контур буде мати форму піраміди). Перетворенням множини μ в множину μ^1 , що задається моделлю (1), є оператор перетворення G [1, 2, 5]

$$G(\mu) = \mu^1. \quad (2)$$

Якщо для вихідних S масивів застосуємо оператор перетворення G , що задається виразом (1), то для кожного масиву отримуємо свій порядковий розклад [1, 2]:

$$\mu_1^1 = \bigcup_{i=1}^{R_1^1} a_{1i}^i, \quad \mu_2^1 = \bigcup_{i=1}^{R_2^1} a_{2i}^i, \quad \dots, \quad \mu_s^1 = \bigcup_{i=1}^{R_s^1} a_{si}^i, \quad (3)$$

де μ_s^1 — множина з номером S на першому рівні, тоді для k -го рівня множина з номером 1 запи-

сується μ_j^k , відповідно R_S^1 — кількість елементів у S множині на першому рівні, R_i^k — кількість елементів у S множині на k -му рівні.

Якщо об'єднати отримані елементи (3) у матрицю M_1 , то формується матриця прямого розкладу на першому рівні [1, 2, 6]

$$M_1 = \bigcup_{s=1}^S \left(\bigcup_{i=1}^{R_S^1} a_{si}^1 \right). \quad (4)$$

Якщо матрицю M_1 переписати у такий спосіб, щоб згрупувати її елементи не за рядками, а за стовпчиками, то сформується нова матриця [1]:

$$M_1^T = \bigcup_{i=1}^{R_S^1} \left(\bigcup_{s=1}^S a_{si}^1 \right), \quad (5)$$

тобто матрицю M_1 — транспонуємо. А позначивши через T оператор транспонування матриці, тобто, перехід від (4) до (5), маємо $T(M_1) = M_1^T$.

В процесі послідовного застосування оператора G до множин μ_S^{1T} формується на другому рівні ($k = 2$) новий масив μ_2 [1, 6]:

$$\mu_2 = \bigcup_{R^1=1}^{R^1} \mu_{R^1}^2 = \bigcup_{R^1=1}^{R^1} \left(\bigcup_{i=1}^{R_{R^1}^2} a_{R^1 i}^2 \right). \quad (6)$$

Починаючи з другого рівня, утворення матриць $M_2^T, M_3^T, \dots, M_k^T$ відбувається не тільки за допомогою операції транспонування, як у (5), а й за діагоналями, і, отримані множини, є діагональними. Матрицю M_2^T можна отримати з M_2 таким же чином як і M_1^T і M_1 , виключивши перший елемент a_{11}^2 — хвостовий елемент мережі. Елемент a_{11}^2 — є першим результатом у S перетворенні вихідних масивів. На кожному рівні, починаючи з другого, утворюються по одному елементу типу a_{11}^k , де k — номер рівня ($k = 2, 3, \dots$), тому що в перших стовпцях матриць M_2, M_3, \dots, M_k знаходиться тільки по одному елементу. Матриця M_2^T переходить на третій рівень і є основою для побудови матриці M_3 і т. д., поки в матриці M_k (на якомусь k -му рівні) не залишиться один елемент, тобто, $T(S(M_k)) = M_k^T$ не містить жодного елемента. Послідовне застосування трьох операторів G, S, T формує функціонал Φ , тобто $\Phi(M) = T[S(G(M))]$, де S — оператор зсуву рядка на величину меншу номера даного рядка на одиницю і виключення першого стовпця матриці M_k в результат розкладу. Мережний метод прямого паралельно-ієрархічного перетворення полягає в послідовному застосуванні до початкових множин $\bigcup_{s=1}^S \mu_s$ по одному разу операторів перетворення G і транспонування T , а потім $(k-1)$ раз функціонала Φ [1, 2, 5]:

$$\Phi \left[T \left(G \left(\bigcup_{s=1}^S \left(\bigcup_{i=1}^n a_i \right) \right) \right) \right] = \bigcup_{t=2}^k a_{11}^t, \quad (7)$$

де a_{11}^k — вихідна інформація (хвостові елементи мережі) прямого паралельно-ієрархічного перетворення.

Імітаційне моделювання та програмна реалізація мережної моделі прямого паралельно-ієрархічного перетворення

Для демонстрації моделі вигляду (7) розглянемо числовий приклад прямого паралельно-ієрархічного перетворення з використанням, наприклад, G — перетворення, інформація якого задана у вигляді числових множин μ_1, μ_2, μ_3 [1]:

$$\mu_1 = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 7 \end{pmatrix}; \mu_2 = \begin{pmatrix} 1 \\ 4 \\ 9 \\ 10 \end{pmatrix}; \mu_3 = \begin{pmatrix} 3 \\ 15 \\ 20 \end{pmatrix}.$$

Застосовуючи модель вигляду (1), отримаємо:

$$G(\mu_1) = (8 \ 3 \ 4 \ 2); \quad G(\mu_2) = (4 \ 9 \ 10 \ 1); \quad G(\mu_3) = (9 \ 24 \ 5).$$

З отриманих результатів побудуємо матриці

$$M_1 = \begin{pmatrix} 8 & 3 & 4 & 2 \\ 4 & 9 & 10 & 1 \\ 9 & 24 & 5 & X \end{pmatrix};$$

$$M_2 = \begin{pmatrix} 12 & 8 & 1 & X & X \\ X & 9 & 12 & 15 & X \\ X & X & 12 & 2 & 5 \\ X & X & X & 2 & 1 \end{pmatrix} \Rightarrow a_{11}^1 = 12;$$

$$M_3 = \begin{pmatrix} 16 & 1 & X & X & X \\ X & 3 & 22 & X & X \\ X & X & 6 & 13 & X \\ X & X & X & 2 & 4 \end{pmatrix} \Rightarrow a_{11}^2 = 16;$$

$$M_4 = \begin{pmatrix} 2 & 2 & X & X \\ X & 12 & 16 & X \\ X & X & 4 & 11 \\ X & X & X & 4 \end{pmatrix} \Rightarrow a_{11}^3 = 2;$$

$$M_5 = \begin{pmatrix} 4 & 10 & X & X \\ X & 8 & 12 & X \\ X & X & 8 & 7 \end{pmatrix} \Rightarrow a_{11}^4 = 4;$$

$$M_6 = \begin{pmatrix} 16 & 2 & X \\ X & 16 & 4 \\ X & X & 7 \end{pmatrix} \Rightarrow a_{11}^5 = 16;$$

$$M_7 = \begin{pmatrix} 4 & 14 & X \\ X & 8 & 3 \end{pmatrix} \Rightarrow a_{11}^6 = 4;$$

$$M_8 = \begin{pmatrix} 16 & 6 \\ X & 3 \end{pmatrix} \Rightarrow a_{11}^7 = 16;$$

$$M_9 = (6 \ 3) \Rightarrow a_{11}^8 = 6;$$

$$M_{10} = (3) \Rightarrow a_{11}^9 = 3.$$

Результат прямого паралельно-ієрархічного перетворення

$$\prod_{t=2}^{10} \left[T \left(G \left(\bigcup_{i=1}^3 \mu_i \right) \right) \right] = (12, 16, 2, 4, 16, 4, 16, 6, 3) \cdot \sum_{i=1}^{10} \mu_i = 79.$$

Для цього числового прикладу основна властивість паралельно-ієрархічної мережі, яка використовується для процедури її навчання [1, 2]:

$$\sum_{i=1}^3 \mu_i = \sum_{i=1}^{11} a_i = 12 + 16 + 2 + 4 + 16 + 4 + 16 + 6 + 3 = 79.$$

Необхідно відмітити, що на поточному рівні t мережної обробки сума хвостових елементів на $(t-1)$ -му і $(t-2)$ -му рівнях плюс сума всіх елементів, крім хвостового t -го рівня, дорівнює сумі всіх хвостових елементів k рівнів, тобто, $a_{11}^{t-1} + a_{11}^{t-2} + \sum_{i=2}^N a_i^t = \sum_{t=1}^k a_{11}^t$, де N — кількість елементів на t -му рівні. Ця властивість мережних обчислень подібна одній з основних властивостей чисел Фібоначчі, прикладні аспекти застосування яких викладені в роботі [7].

З розглянутих тверджень і прикладу очевидно, що модель вигляду (7) цілком описує мережне паралельно-ієрархічне перетворення.

Розглянута математична модель прямого паралельно-ієрархічного перетворення, в порівнянні з відомими числовими методами перетворення (наприклад, розкладання в математичні ряди), простими операціями, типу додавання, забезпечують складну функціональну обробку сигналів у реальному часі, а також однозначність і оборотність із гарною збіжністю обчислювального процесу. Причому, мережний метод обробки одночасно приводить до швидкого ущільнення вхідних масивів інформації.

З метою вирішення поставлених проблем в роботі досліджуються та аналізуються деякі з основних технологій апаратної та програмної реалізації штучних нейроподібних паралельно-ієрархічних систем, зокрема, мультимедійні центральні процесори (CPU) та альтернативні сучасні апаратні засоби (зокрема GPU), а також наведено попередні результати їх імітаційного моделювання та програмної емуляції [8—10].

В ході наукових досліджень реалізовано програмну бібліотеку (на мові програмування C#) для виконання паралельно-ієрархічних обчислень у двох варіантах: на CPU (центральний процесор, рис. 1) та GPU (графічний відеоадаптер, рис. 2), а також створено набір демонстраційних додатків, що дають змогу оцінити достовірність роботи програмного продукту та показують ефективність застосування в прикладних задачах обробки та порівняння зображень (на прикладі обробки плямових зображень лазерного пучка, що використовується у системах профілювання лазерних променів).

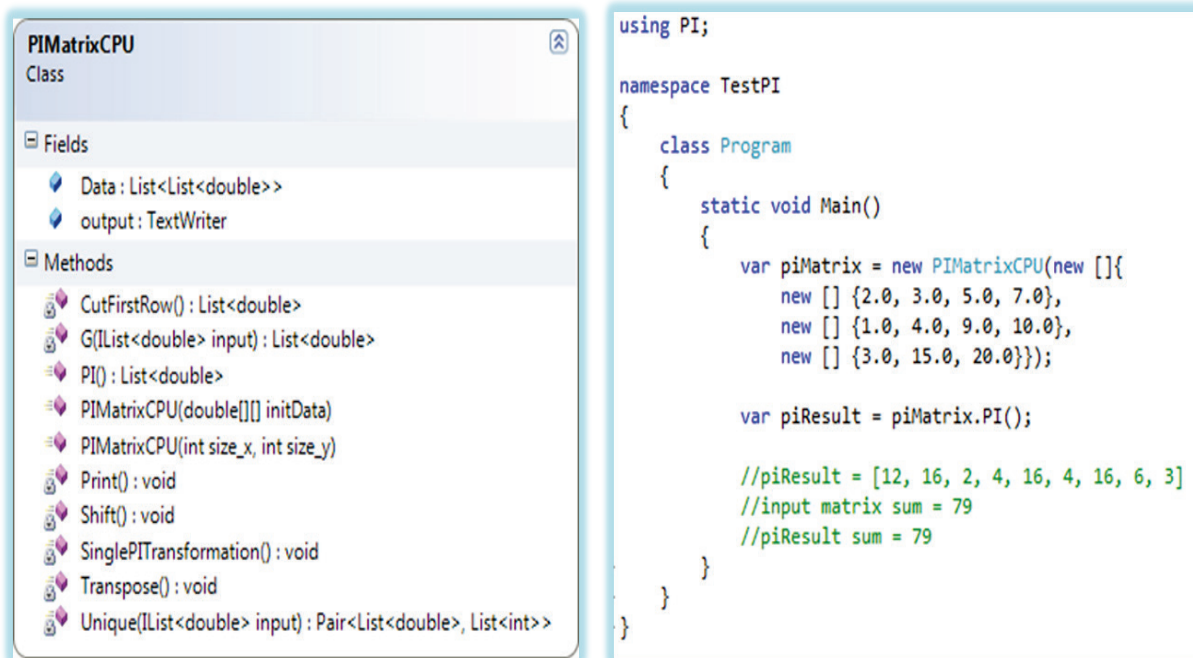


Рис. 1. Структура програмної бібліотеки PIMatrix (CPU)

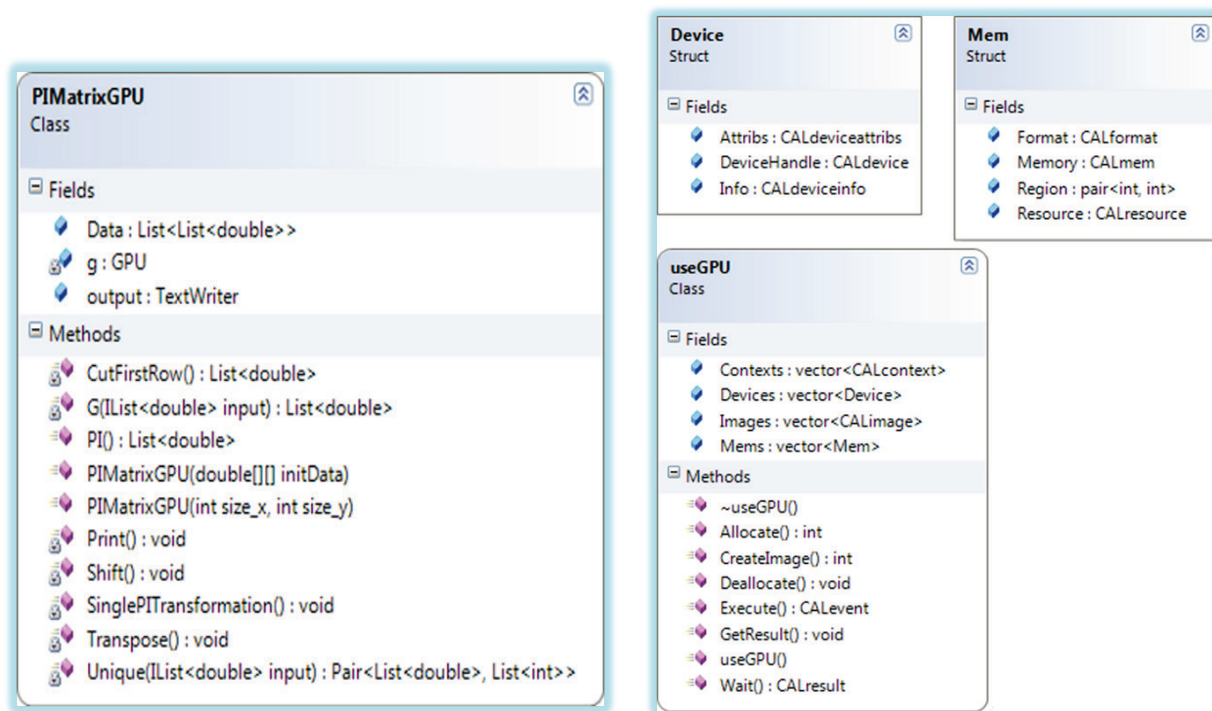


Рис. 2. Структура програмної бібліотеки PIMatrix (GPU)

Для роботи з програмною бібліотекою необхідно підключити її програмну збірку до проекту. Після чого потрібно створити об'єкт класу «PIMatrix», конструктор якого приймає або розмірність вхідної матриці (тоді, за замовчуванням після створення об'єкта, вона буде заповнена значенням «Double.NaN (Not A Number)»), або список із одновимірних масивів типу «double» — власне, сама вхідна матриця. Після чого необхідно викликати метод «PIMatrix.PI()», який повертає список із чисел типу «double» — набір хвостових елементів для вхідної матриці. Для модифікації також доступне поле «PIMatrix.Data» — матриця, для якої буде виконуватись паралельно-ієрархічне-перетворення (тип «List<List<double>>»).

Перший демонстраційний додаток (рис. 3) дозволяє задати розміри матриці для паралельно-ієрархічного перетворення та заповнити її власноруч або за допомогою генератора псевдовипадкових чисел. Після чого для заданої матриці виконується паралельно-ієрархічне перетворення, виводяться значення хвостових елементів, їх сума та сума вхідної матриці (означені суми, згідно теорії паралельно-ієрархічного перетворення, повинні збігатись) [1, 2, 6].

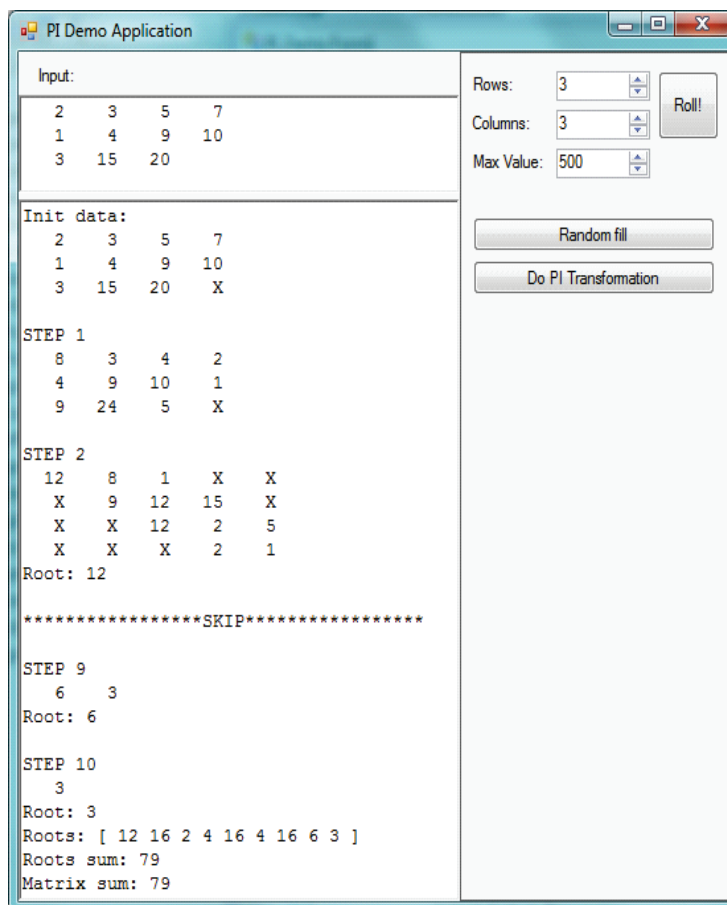


Рис. 3. Екранна форма програми для імітаційного моделювання прямого паралельно-ієрархічного перетворення

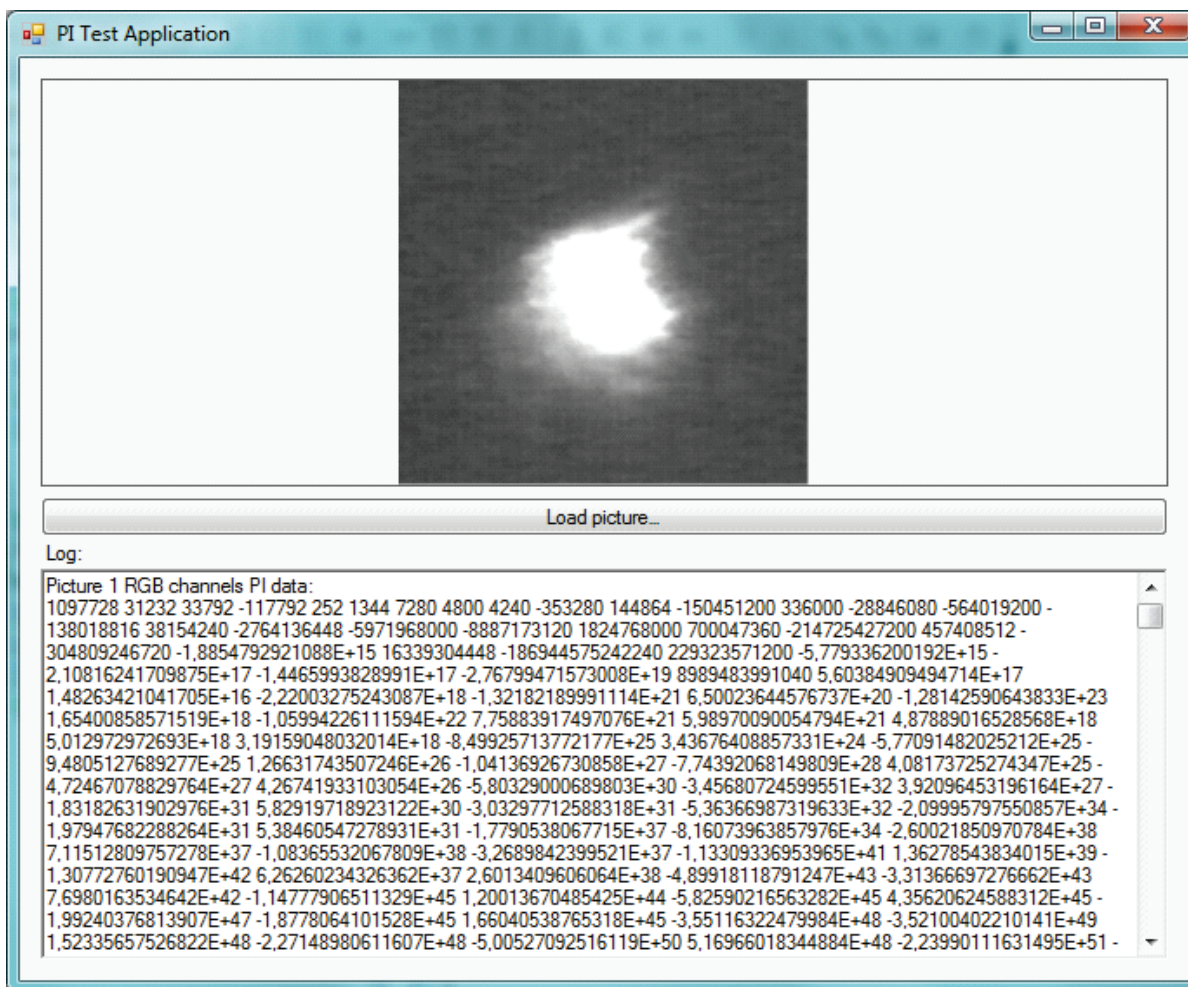


Рис. 4. Екранна форма програми для порівняння зображень на основі застосування паралельно-ієрархічного перетворення

Другий демонстраційний додаток (рис. 4) дозволяє завантажувати попарно зображення, над кожним із яких виконується паралельно-ієрархічне перетворення та на основі отриманого набору хвостових елементів виконується порівняння введених зображень.

Результати роботи можуть знайти застосування у експериментальному дослідженні функціонування не лише паралельно-ієрархічних систем, але й для подальших досліджень з розробки ієрархічних мереж на основі технологій GPGPU, а також на основі оптико-електронної та оптичної елементної бази.

Висновки

Запропоновано нейроподібну мережну модель паралельно-ієрархічної обробки інформації у вигляді багатоетапних процедур кореляційних взаємодій, що дає можливість вдосконалення структурно-функціональної організації паралельно-ієрархічних мереж.

Здійснено моделювання часового зсуву між гілками мережі на основі таких технологій програмно-апаратної реалізації як мультимедійні центральні процесори (CPU) та альтернативні сучасні апаратні засоби (GPU), що дозволило проаналізувати показники ефективності нейроподібної мережної моделі.

Розроблено алгоритми та програмно реалізовано мовою програмування C# спеціалізовані програмні бібліотеки для виконання паралельно-ієрархічних обчислень, що дозволило довести збіжність результатів імітаційного моделювання та програмної реалізації з результатами математичного моделювання.

Проведено експериментальні дослідження, в яких застосовано пряме паралельно-ієрархічне мережне перетворення для розв'язання задачі попереднього оброблення плямоподібних зображень лазерного променя, що дозволило підвищити показники швидкодії під час перетворення, ущіль-

нення та оброблення зображень.

СПИСОК ЛІТЕРАТУРИ

1. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : моногр. / [В. П. Кожем'яко, Ю. Ф. Кутаєв, С. В. Свечніков та ін.] — Вінниця : УНІВЕРСУМ-Вінниця, 2003. — 324 с.
2. Кожем'яко В. П. Паралельно-ієрархічні мережі як структурно-функціональний базис для побудови спеціалізованих моделей образного комп'ютера : моногр. / В. П. Кожем'яко, Л. І. Тимченко, А. А. Яровий. — Вінниця : Універсум-Вінниця, 2005. — 161 с.
3. Патент № 2013805 Российская Федерация. Способ параллельного сложения длительной группы временных интервалов / Кожемяко В. П., Кутаев Ю. Ф., Мартынюк Т. Б., Тимченко Л. И., Буда А. Г. ; заявл. 1995.
4. Мусман С. Достижения в области кодирования изображений / Мусман С., Пирш П., Граллерт Х.-Й. // ТИИЭР. — 1985. — Т. 73, № 4. — С. 23—54.
5. Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідално сільового перетворення зображень / [Кожем'яко В. П., Тимченко Л. І., Кутаєв Ю. Ф., Івасюк І. Д.]. — Київ : УМК ВО, 1994. — 272 с.
6. Яровий А. А. Нейроподібна мережна модель паралельно-ієрархічної обробки цифрової інформації для задач ідентифікації плямових зображень лазерних пучків / А. А. Яровий // Вісник Вінницького політехнічного інституту. — 2007. — № 5 (74). — С. 95—102.
7. Стахов А. П. Коды золотой пропорции. / А. П. Стахов— Москва : Радио и связь, 1984 — 151 с.
8. Яровой А. А. Прикладные аспекты программно-аппаратной реализации нейроподобных параллельно-иерархических систем / А. А. Яровой / Нейроинформатика-2009 : науч. сессия МИФИ — 2009, XI Всероссийская научно-техническая конференция (Москва, 27—30 января 2009 г.), [в 2 частях. Ч. 2.] : сб. науч. тр. — Москва : МИФИ, 2009. — С. 39—48.
9. Яровий А. А. Аналіз функціонування програмної моделі GPGPU в контексті організації паралельних обчислень в нейроподібних паралельно-ієрархічних системах / А. А. Яровий // Оптико-електронні інформаційно-енергетичні технології. — 2009. — № 1 (17). — С. 42—49.
10. Прикладна реалізація масштабних нейронних та нейроподібних паралельно-ієрархічних мереж на основі технологій GPGPU [Електронний ресурс] : [Електронне наукове фахове видання] / А. А. Яровий, Ю. С. Богомолов, К. Ю. Вознесенський // Наукові праці Вінницького національного технічного університету. — 2009. — № 2. — С. 1—8. — Режим доступу до журн. : http://www.nbu.gov.ua/e-journals/VNTU/2009_2_ua/2009-2.files/uk/09aayogt_ua.pdf.

Рекомендована кафедрою комп'ютерних наук

Надійшла до редакції 12.01.10
Рекомендована до друку 19.03.10

Яровий Андрій Анатолійович — доцент кафедри комп'ютерних наук.
Вінницький національний технічний університет