

УДК 621.398:007

А.А. ЯРОВИЙ

ПАРАЛЕЛЬНО-ІЄРАРХІЧНЕ ПЕРЕТВОРЕННЯ ІНФОРМАЦІЙНИХ СЕРЕДОВИЩ НА ОСНОВІ МАСКОВОГО МЕТОДУ

*Вінницький національний технічний університет
95, Хмельницьке шосе, Вінниця, 21021, Україна,
Тел.: +380 (432) 580019, E-mail: axa@vinnitsa.com*

Анотація. В проведених дослідженнях здійснено аналіз методів, математичних моделей та особливостей структурно-функціональної організації мережної моделі паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу. На основі проведених досліджень розроблено програмний комплекс призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами, а також зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для їх відновлення.

Анотация. В проведенных исследованиях совершен анализ методов, математических моделей и особенностей структурно-функциональной организации сетевой модели параллельно-иерархического преобразования информационных сред на основе масочного метода. На основании проведенных исследований разработан программный комплекс, который предназначен для реализации прямого параллельно-иерархического преобразования с оптимизацией формирования масок над информационными массивами, а также обратного параллельно-иерархического преобразования на основе оптимизированного масочного метода для их восстановления.

Abstract. The analysis of methods, mathematical models and features of structural-functional organization of the network model of parallel-hierarchical transformation on the base of masked method is carry out in researches. The software package, which applicative for realization of direct parallel-hierarchical transformation with optimization of mask generation at data arrays, as well as inverse parallel-hierarchical transformation on the base of optimized masked method for data recovery is developed on the base of scientific researches.

Ключові слова: паралельні обчислення, паралельно-ієрархічне перетворення, системи штучного інтелекту, обробка зображень, розпізнавання образів.

ВСТУП

В усіх без винятку реальних задачах, для розв'язання яких узагалі є сенс використовувати комп'ютерні системи, є присутнім природний внутрішній паралелізм, тобто можливість у тій або іншій формі розпаралелити дії, пов'язані з перетворенням даних. Кожен конкретний обчислювальний пристрій або система своїми апаратними і програмними засобами підтримує ті або інші види паралелізму обробки даних. Це справедливо і для комп'ютерних систем із класичною послідовною архітектурою, у якої вже присутні деякі риси паралелізму [1].

Паралельні завдання можуть виконуватися в одно- або багатопроцесорному середовищі. У однопроцесорному середовищі паралельні завдання існують в один і той же час та виконуються протягом одного й того самого інтервалу часу за рахунок контекстного перемикання. У багатопроцесорному середовищі, якщо достатня кількість процесорів, паралельні завдання можуть виконуватися в одні і ті ж моменти часу протягом одного й того ж періоду.

Методи паралельного програмування дозволяють розподілити роботу програми між двома (або більше) процесорами в рамках одного фізичного або одного віртуального комп'ютера. Основний фактор, що впливає на ступінь прийнятності для паралелізму того чи іншого інтервалу часу, визначається конкретним програмним додатком [2-4].

Мета технологій паралелізму – забезпечити умови, що дозволяють комп'ютерними програмами здійснити більший обсяг роботи за той же період часу. Тому проектування програм має орієнтуватися не на виконання однієї задачі в деякий проміжок часу, а на одночасне виконання декількох задач, на які

попередньо повинна бути розбита програма. Можливі ситуації, коли метою є не виконання більшого обсягу роботи протягом того ж інтервалу часу, а спрощення вирішення задачі з точки зору програмування. Іноді є сенс представляти вирішення проблеми як множину паралельно виконуваних завдань [2-4].

Встановити повну узгоджену відповідність між потенційним паралелізмом, закладеним у задачі, і його реалізацією в реальному обчислювальному середовищі є бажаною, але поки що важко досяжною задачею. Адже для цього прийшлося б формувати власне обчислювальне середовище (власну систему) під кожну задачу. Таке рішення є можливим, але використовується лише в спеціальних додатках [1, 2].

При аналізі паралелізму, реалізованого в обчислювальних пристроях, виділяють декілька рівнів. Це рівень завдань, алгоритмічний, командний і арифметичний рівні.

Очевидно, що з погляду вимог до обчислювального середовища, що реалізує принцип паралельно-ієрархічного перетворення, інтерес представляють лише ті рівні і форми паралелізму, для яких при їхній реалізації необхідна взаємодія операційних пристроїв (спецпроцесорів, пристроїв передачі даних, пристроїв обробки великих масивів інформації) з аналізованим обчислювальним середовищем [1, 5].

Зупинимося більш детально на системних вимогах для розробки засобів організації високопродуктивних паралельно-ієрархічних обчислювальних процесів [1]. Такі системи повинні представляти в максимально стиснутому і кодованому вигляді як елементарні такі частини даних, які одночасно мають і кількісну, і якісну, і просторово-часову оцінку. Для побудови вказаних засобів на алгоритмічному і структурному рівнях повинно бути задане правило перетворення групи множин елементів даних. Для можливості вірогідного опису таких систем і виявлення взаємозв'язків їх елементів повинна бути задана розмірність множин, закон розподілу ймовірностей, Q^* –перетворення та F^* – критерій [6, 7]. На алгоритмічному рівні в них повинні використовуватися прості для обчислення операції – порівняння і зсуву та забезпечувати ущільнення вхідних даних у сполученні з можливістю організації швидкого паралельно-ієрархічного перетворення. Ріст їх функціональних можливостей повинен випереджати ріст їх складності, яка характеризує системи такого типу як багатофункціональні. В таких засобах повинні забезпечуватися паралельно-ієрархічний доступ до даних за рахунок паралельно-ієрархічної організації її алгоритмічної структури. Паралельно-ієрархічне перетворення повинно об'єднати властивості просторово-частотного і структурного перетворень. Хвостові елементи мережі паралельно-ієрархічного перетворення повинні служити основними елементами інформаційного середовища і забезпечувати адекватний та ущільнений опис початкових даних. При реалізації перетворення повинен забезпечуватися паралельно-ієрархічний доступ до даних, який дозволяє сполучити ієрархію і паралелізм обробки даних. Завдяки цьому можлива зміна внутрішньої структури її елементів, висока живучість, надійність, можливість перебудовуваності залежно від зовнішніх умов, пристосованість структури при зміні функцій системи. При реалізації системи повинна забезпечуватися її функціональна цілісність, тобто виконуються взаємопогоджувані дії всіх елементів для функціонування системи як єдиного цілого при досягненні поставленої мети. Цим вимогам повною мірою відповідає запропонована в роботі паралельно-ієрархічна мережева обчислювальна система.

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є розвиток теоретичних основ паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу та реалізації вдосконалених методів, алгоритмів та програмно-апаратних засобів для ефективного багатоетапного сприйняття, збереження, ущільнення і розпізнавання зображень.

МАТЕМАТИЧНИЙ ОПИС ПАРАЛЕЛЬНО-ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ ІНФОРМАЦІЙНИХ СЕРЕДОВИЩ

Розглянемо мережний метод паралельно-ієрархічного перетворення на рівні моделей [1, 5-8]. Нехай є S ($S = 1, 2, 3 \dots$) непустих множин елементів, що задають інформацію. Кількість елементів множини називається його довжиною (позначення L_μ – довжина множини μ). Кількість різноманітних елементів множини називається розмірністю даної множини (позначення R_μ) [1, 5, 6].

Розглянемо математичну модель паралельного розкладання множини $\mu = \{a_i\}, i = \overline{1, n}$.

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

де $a_i \neq 0$, R – розмірність даної множини. З однакових елементів формуються підмножини, елементи однієї підмножини позначаються через a^k , $k = \overline{1, R}$, n_k – кількість елементів у k -ій підмножині (тобто кратність числа a_k), a^j – довільний елемент множини $\{a^k\}$, обраний на j -му кроці, $j = \overline{1, R}$, $a^0 = 0$, $n_0 = 0$ [1, 6].

В попередніх роботах доведено, що модель (1) відповідає алгоритму паралельного перетворення масиву інформації [1, 5, 6]. На першому кроці алгоритму з даної множини чисел $t_1 = \{a_i\}$, $i = \overline{1, n}$, $a_i \neq 0$ вибирається довільне число й множиться на n , яке відповідає першій складовій правій частини (1). Для аналізу наступних кроків алгоритму введемо в розгляд множини чисел $z_2, \dots, z_j, z_{j+1}, \dots, z_R$, які утворюються за таким правилом [1, 6]: елементами множини z_{j+1} є ненульові числа, що беруть участь у $(j+1)$ -му кроці алгоритму. Це різниця кожного з елементів множини z_j з довільним числом, обраним із цієї множини на j -му кроці алгоритму.

Утворимо по вказаному правилу множини z_2 [1, 6].

$$z_2 = \{a_i - a_1\}, \quad i = \overline{1, n - n_1},$$

де n_1 – кратність числа a^1 , тому z_2 складається з $(n - n_1)$ ненульових елементів. З множини z_2 вибирається довільний елемент і позначається як $(a^2 - a^1)$ [1,6]. У результаті виконання другого кроку алгоритму елемент $(a^2 - a^1)$ множиться на число елементів множини z_2 , тобто на число $(n - n_1)$. Отриманий добуток $(n - n_1)(a^2 - a^1)$ відповідає другій складовій правій частини (1).

Якщо в моделі (1) здійснити заміну знака Σ на об'єднання \cup , то результатом розкладання множини μ є об'єднання елементів (множина розкладання μ^1) [1, 5, 6].

$$\mu = \bigcup_{i=1}^n a_i = \bigcup_{j=1}^R \left(n - \sum_{k=0}^{j-1} n_k \right) (a^j - a^{j-1}) \quad (2)$$

Перетворення множини μ в множини μ^1 , яка задається моделлю (2), називається оператором перетворення G (різновид Q^* -перетворення) [1, 5, 6], тобто

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

Якщо для вихідних S масивів застосуємо оператор перетворення G , що задається виразом (2), то для кожного масиву одержимо своє порядкове розкладання [1, 5, 6]:

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

де μ_s^1 – множина під номером S на першому рівні, тоді для k -го рівня множина під номером 1 запишеться μ_1^k , відповідно R_s^1 – кількість елементів у S множині на першому рівні, R_1^k – кількість елементів у S множині на k -му рівні. Об'єднаємо отримані елементи (3) у матрицю M_1 , яка називається матрицею прямого розкладання на першому рівні [1, 5, 6]:

$$M_1 = \begin{pmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1R_1^1}^1 X & \dots & X \\ a_{21}^1 & a_{22}^1 & \dots & \dots & \dots & a_{2R_2^1}^1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{S1}^1 & a_{S2}^1 & \dots & a_{SR_S^1}^1 X & \dots & X \end{pmatrix}, \quad \text{або} \quad M_1 = \bigcup_{S=1}^S \left(\bigcup_{i=1}^{R_S^1} a_{si}^1 \right). \quad (4)$$

Для зручності запису вакантні місця в матрицях заповнюються позначенням "X". Не порушуючи відповідності, для наочності, другий рядок у (4) має максимальну довжину. Очевидно, що таку довжину може мати будь-який рядок.

Матрицю M_1 перепишемо, згрупувавши її елементи не за рядками, а за стовпчиками і таким способом сформуємо нову матрицю [1, 5, 6]:

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

тобто матрицю M_1 – транспонуємо. Позначимо через T оператор транспонування матриці, тоді [1, 5, 6]:

$$T(M_1) = M_1^T.$$

Оскільки R_S^1 – приймає різноманітні значення для кожної множини, то кількість множин у μ_1^T буде дорівнювати максимальному з чисел R_S^1 , тобто $\max \{R_1^1, R_2^1, \dots, R_S^1\} = R^1$. Таким чином, через R^1 позначається максимальна довжина множини з усіх множин розкладання на першому рівні [1, 6].

Матриця M_1^T складається відповідно з множин μ_S^1 , які переходять на наступний рівень і є основою для побудови матриці M_2 .

Послідовне застосування трьох операторів G, S, T називається функціоналом Φ [1, 5-7], тобто $\Phi(M) = T [S (G (M))]$, або

$$\Phi \left[T \left(G \left(\bigcup_{S=1}^S \mu_S \right) \right) \right] = \bigcup_{t=2}^k a_{11}^t, \quad (5)$$

де μ_S – початкові множини ($S = 1, 2, 3, \dots$), a_{11}^t – елементи розкладання початкових множин, одержувані по одному на кожному рівні, починаючи з другого.

Модель (5) можна переписати в іншому вигляді [1, 5-7]:

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

Таким чином, 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 . На кожному рівні паралельно-ієрархічного перетворення утворюються по одному елементу a_{11}^k розкладання S – вихідних множин, де a_{11}^k – вихідна інформація прямого паралельно-ієрархічного перетворення. Цією інформацією є діагональні множини з одного діагонального кратного загального елемента [1, 5-8].

БАЗОВІ АСПЕКТИ СТРУКТУРНО-ФУНКЦІОНАЛЬНОЇ ОРГАНІЗАЦІЇ МЕРЕЖНОЇ МОДЕЛІ ПАРАЛЕЛЬНО-ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ ІНФОРМАЦІЙНИХ СЕРЕДОВИЩ

В попередніх дослідженнях [1, 7] показано, що структурну модель організації паралельно-ієрархічної мережевої системи можна описати за допомогою шести функціональних множин, що дозволяє формалізувати структурну організацію її просторових зв'язків.

Для аналізу структурно-функціональної організації паралельно-ієрархічної мережевої системи спочатку розглянемо базові поняття.

Згорткою елементів із координатами $a_{1i}(t_j)$ і $a_{2i}(t_j)$ відповідно p_{1j} -ої і $p_{1(j+1)}$ -ої гілок j -ого рівня є елементи з координатами $a_{(i+1)1}(t_{j+1})$ і $a_{(i+1)2}(t_{j+2})$ p_{1j} -ої гілки $(j+1)$ -ого рівня [1, 6, 7]. Усіченою згорткою елементів із координатами $a_{1i}(t_j)$ і $a_{2i}(t_j)$ відповідно p_{1j} -ої і $p_{1(j+1)}$ -ої гілок j -ого рівня є елемент із координатою $a_{(i+1)1}(t_{j+1})$ p_{1j} -ої гілки $(j+1)$ -ого рівня [1, 6, 7].

Розгорткою елементів із координатами $a_{(i+1)1}(t_{j+1})$ і $a_{(i+1)2}(t_{j+2})$ p_{1j} -ої гілки $(j+1)$ -ої рівня є елементи з координатами $a_{1i}(t_j)$ і $a_{2i}(t_j)$ відповідно p_{1j} -ої і $p_{1(j+1)}$ -ої гілок j -ого рівня [1, 6, 7]. Усіченою розгорткою елемента з координатою $a_{(i+1)1}(t_{j+1})$ p_{1j} -ої гілки $(j+1)$ -ого рівня є елементи з

координатами $a_{1i}(t_j)$ і $a_{2i}(t_j)$ відповідно p_{1j} -ої і $p_{1(j+1)}$ -ої гілок j -ого рівня [1, 6, 7]. Аналогічно описуються поняття згортки, розгортки й усіченої розгортки для багатьох змінних.

Поточною згортокою елементів a_1 і a_2 є згортка даних елементів у будь-який момент часу t_j [1, 6, 7]. Попередньою згортокою елементів мережі стосовно поточної згортки є така згортка мережі, що знаходиться на однакову кількість згорток від хвостового елемента стосовно поточної згортки.

Наприклад, при кодуванні поточної згортки паралельно-ієрархічного перетворення необхідно зробити аналіз маскової функції попередньої згортки. Якщо маскова функція даної згортки дорівнює одиниці, то поточна згортка складається з одного елемента гілки, вміст якої переноситься на елемент сусідньої згортки і реалізується операція усіченої згортки [1, 6, 7].

Для демонстрації прямого й зворотного паралельно-ієрархічного перетворення необхідно розглянути поняття повної маскової мережі $C^{(m)}(i, j)$, що утвориться з основної $C_1^{(m)}(i, j)$ і хвостових маскових мереж $C_2^{(m)}(i, j)$ [1, 6, 7]. Також, як і у випадку базисної мережі, для $C^{(m)}(i, j)$ і $C_1^{(m)}(i, j)$, $C_2^{(m)}(i, j)$ справедливе співвідношення, яке відображає послідовний процес переходу інформаційного потоку даних із попереднього рівня паралельно-ієрархічної мережі на наступний [1, 5-7]:

$$M(k) = \bigcup_{j=1}^k M \Big] \frac{k}{j} [,$$

де $\Big] \frac{k}{j} [$ вказує на розмірність множини M на k -ому рівні, $k = \overline{1, n^2}$, n – кількість вихідних множин, $|k| \leq n^2$.

Причому елементи маскової мережі є бінарними, тобто $C^{(m)}(i, j) \in \{0, 1\}$ і для них справедлива така система [1, 5-7]:

$$C^{(m)}(i, j) = \begin{cases} 1, & \text{якщо } a_1 = a_2 = a_m = \dots = a_{M(\cdot)} \\ 0, & \text{в іншому випадку} \end{cases} \quad a_1 = a_2 = \dots = a_m = \dots = a_{M(\cdot)},$$

де $a_{M(\cdot)}$ – останній елемент множини $M(\cdot)$. По суті $C^{(m)}(i, j)$ визначає значення маскової функції згорток елементів $a_1(f_j), a_2(f_j), \dots$.

Практичний інтерес становить використання наступної маскової функції згорток [1, 5-7]:

$$C^{(m)}(i, j) = \begin{cases} 1, & \text{якщо } a_1 = a_2 = r \\ 0, & \text{в іншому випадку} \end{cases}, \quad \text{де } r = 0, 1, 2, \dots$$

Використовуючи над подвійними згортками операцію усіченої згортки та її маскову функцію, а також розглянуту структурно-функціональну модель базисної мережі, можна в компактному (ущільненому) вигляді представляти вхідний потік інформації.

Кодування на основі паралельно-ієрархічного перетворення (тобто, пряме паралельно-ієрархічне перетворення) реалізується за допомогою вказаних нижче загальних правил [1, 6, 7].

1. При кодуванні поточної згортки паралельно-ієрархічної мережі відбувається аналіз маскової функції попередньої згортки. Якщо дана функція $C^{(m)}(i, j) = 1$, то маскова функція поточної згортки не запам'ятовується. У протилежному випадку, коли $C^{(m)}(i, j) = 0$, маскова функція поточної згортки фіксується. При цьому, якщо $C^{(m)}(i, j) = 1$, то вміст елемента поточної згортки попереднього рівня (меншого) передається на елемент наступного рівня (більшого) із такою адресою, вміст якого відмінний від нуля. У іншому випадку, якщо маскова функція попередньої згортки $C^{(m)}(i, j) = 0$, то вміст елемента попереднього рівня (меншого) передається на елемент наступного рівня (більшого) з однойменною адресою.

2. Відбувається запам'ятовування вмісту хвостових елементів паралельно-ієрархічної мережі.

3. Маршрутизація вмісту вхідних елементів відбувається з одного рівня (меншого) на інший рівень (більший) до свого хвостового елемента.

4. Якщо маскова функція попередньої згортки $C^{(m)}(i, j) = 1$, то поточна згортка формується з одного елемента, який утворить усічену згортку на наступному рівні. У протилежному випадку, коли

$C^{(m)}(i, j) = 0$, поточна згортка формується з двох елементів, які утворюють згортку на наступному рівні паралельно-ієрархічної мережі.

5. Якщо попередня згортка формується з одного елемента, тобто усіченої згортки, то маскова функція поточної згортки не запам'ятовується.

6. Якщо елементи поточних згорток корелюють (збігаються) з одним із варіантів елементів попередніх згорток, маскові функції яких $C^{(m)}(i, j) = 0$, то маскові функції таких поточних згорток також не запам'ятовуються.

Загальні правила при реалізації декодування (тобто, зворотного паралельно-ієрархічного перетворення) такі [1, 6, 7]:

1. Декодування здійснюється з вмісту кожного хвостового елемента мережі, починаючи з останнього.

2. Маршрутизація вмісту хвостового елемента відбувається з попереднього рівня (меншого) на наступний рівень (більший).

3. Якщо маскова функція попередньої згортки елементів, що утворюють при розгортці елемент наступної згортки дорівнює одиниці, тобто вміст цих же елементів при кодуванні збігається за величиною, то вміст елемента попереднього рівня передається на елемент наступного рівня, вміст якого відмінний від нуля. У протилежному випадку, маскова функція попередньої згортки елементів дорівнює нулю. Тобто, якщо вміст цих елементів при кодуванні не збігається за величиною, то вміст елемента попереднього рівня передається на елемент наступного рівня з однойменною адресою.

Зазначені загальні для паралельно-ієрархічного перетворення правила кодування і декодування застосовуються в алгоритмах паралельно-ієрархічних обчислювальних структур [1, 5-9].

ПРИКЛАДНІ АСПЕКТИ РЕАЛІЗАЦІЇ ПАРАЛЕЛЬНО-ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ ІНФОРМАЦІЙНИХ СЕРЕДОВИЩ НА ОСНОВІ МАСКОВОГО МЕТОДУ

Розроблений програмний комплекс паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами (у вигляді двовимірної матриці даних довільної розмірності або зображення), а також зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для їх відновлення [10].

Спочатку розглянемо програмний модуль, що призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок (рис. 1). Особливістю реалізованого алгоритму є застосування оптимізованого алгоритму формування масок, що зменшує обсяг пам'яті, необхідний для їх збереження у порівнянні із неоптимізованим алгоритмом, а також застосування множення мінімального елемента на потужність в операторі перетворення G , який є одним із етапів паралельно-ієрархічного перетворення.

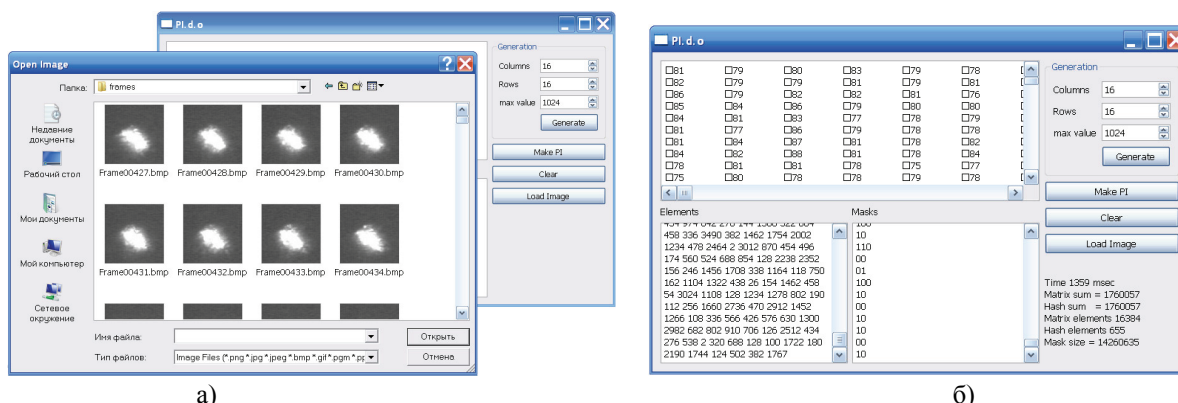


Рис. 1. Екранні форми програмного модуля прямого паралельно-ієрархічного перетворення з оптимізацією формування масок: а) завантаження плямового зображення профілю лазерного променя [11]; б) реалізація прямого перетворення

Для роботи із програмним модулем потрібно:

- 1) додати вихідні коди до проекту;
- 2) створити екземпляр класу `vector<vector<int>>`;

- 3) створити екземпляр класу PI_Optimised;
- 4) у створеному екземплярі класу PI_Optimised потрібно викликати метод getHash (vector<vector<int> > &a), якому передати створений екземпляр класу vector<vector<int> >, метод повертає хвостові елементи у вигляді одновимірного вектора класу vector<int>.
- 5) у екземплярі класу PI_Optimised викликати метод getOptimisedMasks(), метод повертає об'єкт класу QList<String>, який разом з вектором хвостових елементів є унікальною (неповторною для інших об'єктів) характеристикою початкового інформаційного масиву.

Програмний модуль реалізує такі функції:

- 1) завантаження інформаційного масиву – у вигляді зображення або двовимірної матриці даних заданої користувачем розмірності (рис. 1, а);
- 2) виконання прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над даними (зображенням) (рис. 1, б);
- 3) формування одновимірної матриці оптимізованих масок та одновимірної матриці хвостових елементів перетвореного інформаційного масиву (рис. 1, б);
- 4) ведення файлу-протоколу.

Розглянемо програмний модуль, що призначений для реалізації зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для відновлення перетворених (за методом прямого паралельно-ієрархічного перетворення з оптимізацією формування масок) інформаційних середовищ (у вигляді двовимірної матриці даних або зображення). Особливістю реалізованого алгоритму є модифікація процесу декодування на основі оптимізованого алгоритму роботи з масками, що підвищує швидкість декодування (рис. 2).

Маскою H вектора даних μ називають одновимірний масив, потужність якого дорівнює потужності вектора даних μ , елементи маски утворюються за таким правилом:

$$H[i] = \begin{cases} 0, \mu[i] \neq 0; \\ 1, \mu[i] = 0; \end{cases} (i = \overline{1 \dots n}),$$

де n – потужність μ .

Маски використовують при зворотному паралельно-ієрархічному перетворенні. Вони формуються при виконанні оператора перетворення G над вектором даних (рядком).

Процес G -перетворення рядка з утворенням масок складається з таких основних етапів:

1. Присвоюємо $i=1$.
2. Запам'ятовуємо маску H_i рядка μ_i .
3. Видаляємо всі нулі з μ_i , утворюючи μ_i' .
4. Знаходимо мінімальний елемент рядка $\mu_i' - m$.
5. Знайдений мінімальний елемент m множимо на потужність μ_i' та утворюємо i -тий елемент в рядку з результатом G -перетворення.
6. Якщо m дорівнює кожному елементу μ_i' , тоді кінець алгоритму.
7. Утворюємо рядок μ_{i+1} з μ_i' , шляхом віднімання від кожного елемента μ_i' значення m .
8. Якщо $i=i+1$, то переходимо до пункту 1.

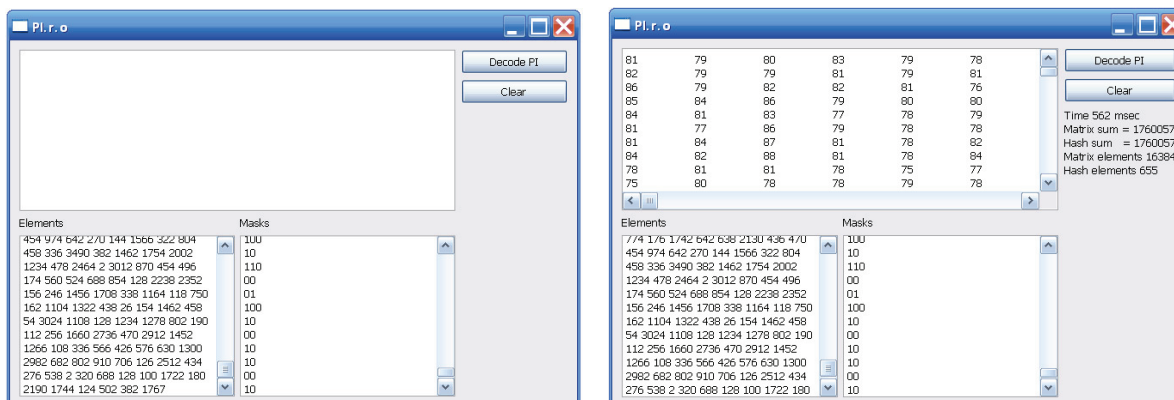


Рис. 2. Екранні форми програмного модуля зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу: а) завантаження масиву хвостових елементів та масиву оптимізованих масок; б) реалізація зворотного перетворення

Програмний модуль реалізує такі функції:

- 1) завантаження масиву хвостових елементів – у вигляді одновимірного набору даних, отриманих при кодуванні за методом прямого паралельно-ієрархічного перетворення з оптимізацією формування масок (рис. 2, а);
- 2) завантаження масиву оптимізованих масок – у вигляді одновимірного набору даних (з 0 та 1), отриманих при кодуванні за методом прямого паралельно-ієрархічного перетворення з оптимізацією формування масок (рис. 2, а);
- 3) виконання зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу над заданими користувачем даними (рис. 2, б);
- 4) відновлення початкового масиву даних (у вигляді двовимірної матриці даних або зображення) (рис. 2, б).

Мова реалізації програмного комплексу – "C++". Функції програмної бібліотеки, після перекомпіляції, коректно працюють з різними операційними системами: MS Windows, GNU/Linux, Mac OS. Для роботи програмного комплексу необхідна наявність встановленої на комп'ютері операційної системи MS Windows або GNU/Linux, або Mac OS; наявність скомпільованої під відповідну платформу програмної бібліотеки; тактова частота процесора не менше 1 GHz; наявність оперативної пам'яті обсягом від 200 Mb. Файл програмної бібліотеки займає до 20 Mb.

Запропонований програмний комплекс паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу було протестовано на різноманітних наборах даних (у вигляді двовимірної матриці даних різноманітної розмірності, а також зображень різноманітної розмірності). Було здійснено реалізацію прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами, а також зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для їх відновлення. Результати роботи комплексу співпали з результатами математичного та комп'ютерного моделювання, що свідчить про коректність та достовірність роботи програмного продукту [10].

ВИСНОВКИ

В ході проведених наукових досліджень було проаналізовано методи, математичні моделі та особливості структурно-функціональної організації мережної моделі паралельно-ієрархічного перетворення інформаційних середовищ на основі маскового методу. А саме, на рівні детального аналізу математичного опису прямого та зворотного паралельно-ієрархічного перетворення, формування системних вимог до розробки засобів організації високопродуктивних паралельно-ієрархічних обчислювальних процесів, системного аналізу структурної організації та програмно-апаратних особливостей організації паралельних обчислень в роботі набули подальшого розвитку теоретичні та методологічні обґрунтування структурно-функціональної організації паралельно-ієрархічної мережевої обчислювальної системи. Відзначено, що межах запропонованої системи є можливість здійснювати моделювання часового зсуву між гілками мережі, наявність якого приводить до можливості проектування високопродуктивних паралельно-ієрархічних та ієрарх-ієрархічних структур і обчислювальних пристроїв. Набув подальшого розвитку опис паралельного обчислювального процесу обробки інформації у вигляді багатоетапних процедур кореляційних взаємодій, що дозволяє синтезувати паралельно-ієрархічну мережеву обчислювальну систему для складної функціональної обробки інформаційних полів в динаміці, зокрема для задач ефективного багатоетапного сприйняття, збереження, ущільнення, обробки зображень (на прикладі вузькоспеціалізованої практично-прикладної задачі профілювання лазерних променів [11]) та розпізнавання образів.

На основі проведених досліджень, математичного та комп'ютерного моделювання розроблено програмний комплекс призначений для реалізації прямого паралельно-ієрархічного перетворення з оптимізацією формування масок над інформаційними масивами, а також зворотного паралельно-ієрархічного перетворення на основі оптимізованого маскового методу для їх відновлення.

СПИСОК ЛІТЕРАТУРИ

1. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : [Монографія.] / В.П. Кожем'яко, Ю.Ф. Кутаєв, С.В. Свечніков, Л.І. Тимченко, А.А. Яровий – Вінниця: УНІВЕРСУМ-Вінниця, 2003. – 324 с.
2. Хьюз К. Параллельное и распределенное программирование на C++. / Хьюз К., Хьюз Т. [Пер. с англ.] – М.: ИД «Вильямс», 2004. – 672 с.
3. Воеводин В.В. Параллельные вычисления. / В.В.Воеводин, Вл.В.Воеводин – СПб.: БХВ-Петербург,

2002. – 608 с.
4. Бройнль Т. Паралельне програмування: Навч. посібник / Бройнль Т. [Пер. з нім. В.А. Святного] – К.: Вища школа, 1997. – 358 с.
 5. В.П. Кожем'яко Паралельно-ієрархічні мережі як структурно-функціональний базис для побудови спеціалізованих моделей образного комп'ютера : [Монографія.] / В.П. Кожем'яко, Л.І. Тимченко, А.А. Яровий. – Вінниця: Універсум-Вінниця, 2005. – 161 с.
 6. Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідально сільового перетворення зображень. / Кожем'яко В.П., Тимченко Л.І., Кутаєв Ю.Ф., Івасюк І.Д. – К: УМК ВО, 1994. – 272 с.
 7. Тимченко Л.І. Паралельно-ієрархічне перетворення як системна модель ефективних засобів штучного інтелекту : автореф. дис. на здобуття наук ступеня доктора техн. наук : спец. 05.13.23 / Тимченко Леонід Іванович ; Державний науково-дослідний інститут інформаційної інфраструктури. – Львів, 2002. – 36 с.
 8. Тимченко Л.И. Многоэтапная параллельно-иерархическая сеть как модель нейроподобной схемы вычислений / Л.И. Тимченко // Кибернетика и системный анализ. – 2000. – №2. – С. 114-134.
 9. Методологічні та прикладні аспекти реалізації мережної моделі прямого та зворотного паралельно-ієрархічного перетворення. / Яровий А.А., Богомолів Ю.С., Сугак І.М. : матеріали Міжнародної науково-технічної конференції [Системний аналіз та інформаційні технології "SAIT 2011"], (Київ, 23–28 травня 2011 р.) – К.: ННК "ПСА" НТУУ "КПІ", 2011. – С. 518.
 10. Розробка методу подання масок для оптимізації зворотного паралельно-ієрархічного перетворення зображень. / Яровий А., Сугак І. : матеріали V Міжнародної науково-технічної конференції [Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування "СПРТП-2011"], (Вінниця, 19-21 травня 2011 р.) – Вінниця, ВНТУ, 2011 – с. 25.
 11. Теоретико-методологические и прикладные аспекты использования технологий визуализации для задач профилирования лазерных лучей / А.А. Яровой, А.М. Яровой : [Электронный журнал Национального исследовательского ядерного университета "МИФИ", Москва] // Научная визуализация. – 2010. – Том 2. – №3. – С. 50-72. – Режим доступа до журн.: <http://sv-journal.com/2010-3/04/index.html>.

Надійшла до редакції 24.12.2010 р.

ЯРОВИЙ А.А. – к.т.н., доцент, докторант кафедри комп'ютерних наук, Вінницький національний технічний університет, Вінниця, Україна.