

МЕТОД ОПТИМІЗОВАНОГО ФОРМУВАННЯ МАСОК ПРИ КОДУВАННІ ІНФОРМАЦІЇ В ПАРАЛЕЛЬНО-ІЄРАРХІЧНОМУ ПЕРЕТВОРЕННІ

Проаналізовано особливості структурно-функціональної організації мережної моделі паралельно-ієрархічного перетворення інформаційних середовищ, а також відповідних методів формування масок. На основі проведених досліджень запропоновано метод оптимізованого формування масок при кодуванні інформації в паралельно-ієрархічному перетворенні та розроблено програмний комплекс, в якому підвищено ефективність прямого та зворотного паралельно-ієрархічного перетворення.

Вступ

Вирішення проблеми швидкого перетворення великих масивів інформації для ефективного забезпечення її запису, збереження, обробки і зчитування пов'язано зі створенням швидкодійних пристроїв кодування і декодування. Швидкодія процесу кодування-декодування масиву цифрових даних залежить, насамперед, від реалізованого алгоритму числової обробки.

Сучасний рівень розвитку схемотехніки багатоканальних цифрових пристроїв кодування-декодування великих масивів інформації у значній мірі забезпечується послідовними алгоритмами числової обробки, що пов'язано з невиправданими часовими витратами, що витрачаються на послідовний у часі процес кодування-декодування [1—4].

Найбільшої актуальності вказана проблема набуває в галузі кодування зображень, де природним способом доцільно використовувати паралельне кодування. На відміну від широко поширених типів кодування відеоінформації, наприклад різницевої, кодово-імпульсної, дельта-модуляції, в основу яких покладено принцип послідовного кодування різницевої інформації, у цій роботі пропонується використовувати паралельно-пірамідальний принцип обробки розподілених в просторово-часовій області результатів кодування масиву даних, що приводить до істотного підвищення алгоритмічної швидкодії, ущільнення масивів даних та забезпечення природної форми подання сигналів (зображень) [5].

Метою роботи є розвиток теоретичних основ паралельно-ієрархічного перетворення інформації шляхом розробки методу оптимізованого формування масок при кодуванні інформаційних масивів для підвищення швидкодії зворотного паралельно-ієрархічного перетворення зображень.

Постановка задачі дослідження. Аналіз відомих підходів до перетворення інформаційного поля

В науковій літературі виділяють три основних типи сприйняття та подальшого перетворення інформаційних полів [5, 6]:

1. Повне сприйняття характеризується тим, що або саме інформаційне поле (ІП) використовується в якості внутрішнього представлення інформації або спосіб його перетворення такий, що дозволяє однозначно відновлювати вихідне поле.

2. При проблемно-орієнтованому сприйнятті використовується не взаємно-однозначне, а багато-однозначне відображення ІП у поле сприйняття, тобто інформація, аналізована системою, є як би «укрупненим зліпком» реальної ситуації.

3. Рефлексорні сприйняття характеризуються тим, що заздалегідь задана процедура перетворення ІП у кінцевий набір ознак.

В цьому контексті, найбільш актуальним і трудомістким є пошук структур вторинних (перетворених) даних для проблемно-орієнтованих систем сприйняття, оскільки в системах повного сприйняття така структура цілком визначається самим ІП, а в системах із рефлексорним сприйняттям — розв'язуваною задачею. У цьому зв'язку здійснимо аналіз існуючих підходів до представлення ІП для організації даних у системах другого типу. Тут виділяють чотири основні схеми [1, 5—8]:

1) пряме представлення (організоване на основі перегляду лише локальних областей ПП і виявляється придатним для виділення первинних ознак, але не для ефективного одержання узагальненої інформації, опису і пошуків об'єктів);

2) представлення на основі ортогональних перетворень (наприклад, Фур'є, Адамара, Карунена–Лоева та ін., які описують ПП набором коефіцієнтів розкладання за деяким базисом, — в переважній більшості виявляються інформативними при аналізі лише тих класів ПП, що можна розглядати як «стаціонарні» сигнали);

3) структурне представлення (при автоматичному створенні опису виникають ті ж труднощі, що і при використанні прямого представлення);

4) представлення на основі нейроподібних методів перетворення [1, 5, 7]. Цей клас методів, є предметом досліджень. До цих методів кодування можна віднести: пірамідальне кодування, анізотропне нестаціонарне кодування з прогнозуванням, кодування на основі нарощування областей і кодування на основі розкладання за напрямками, кращі з яких спроможні забезпечити значення коефіцієнту ущільнення до 70 [1, 3—5].

Досліджуваний у цій роботі метод пірамідальної обробки даних [5, 7, 9] відноситься до нейроподібних методів перетворення і полягає в тому, що починаючи з вихідних елементів ПП $x_0(i, j)$, які підлягають кодуванню, визначається елемент ПП із мінімальним розміром $x_1(i, j)$. В термінах обробки зображень — це не що інше як «низькочастотна версія зображення». Вона з частотою зрізу f_1 може розглядатися як точний прогноз для $x_0(i, j)$. Далі вищеописана операція ітеративним чином повторюється. Тобто з $x_1(i, j)$ вибирається елемент ПП із мінімальним розміром $x_2(i, j)$. Після n ітерацій утворюється послідовність поточних ПП, обумовлених виразом: $x_{i-1}(i, j) — x_1(i, j)$ [5, 7, 9].

На кожній ітерації розмірність поточних ПП скорочується в кількість раз, що рівна $\frac{n_i}{n_{i+1}}$, де n_i і n_{i+1} — відповідно кількість ненульових елементів ПП. Якщо ці поточні ПП розглядати як поставлені одне на інше, то в результаті утвориться пірамідальна структура даних [5, 9]. Пірамідальна структура обробки зображення на будь-якому одному рівні такої структури являє собою результат згортки двох гаусоподібних функцій із вихідним зображенням. Зображення гарної якості при пірамідальному кодуванні утворюється при середньому коефіцієнті ущільнення біля 10 [5, 7, 9].

Опис методу оптимізованого формування масок при кодуванні інформації в паралельно-ієрархічному перетворенні

Досліджуваний спосіб паралельного перетворення великих масивів інформації розглянемо за допомогою мережного алгоритму, основні властивості якого — паралелізм та ієрархія, синхронність і детермінованість [7, 9, 10]. Паралельно-ієрархічна (ПП) мережа складається з множини кінцевих множин Ω , множини елементів A і умовно розбита на ряд рівнів.

Розглядаючи усі H початкових множин першого рівня, при виборі елемента $a_j^1(t_j)$, $i = \{1, 2, \dots, h\}$; $j = \{1, 3, 5, \dots\}$ з кожної множини, формуються нові множини, де i — порядковий номер початкової множини, t_j — такт, у якому вибраний елемент. При першому виборі елементів із H вхідних множин у такті t_1 формується нова множина $M_1^2(t_1) = \{a_1^1(t_1), a_2^1(t_1), a_3^1(t_1), \dots, a_h^1(t_1)\}$. На другому кроці перетворення t_3 з початкових множин формується ще одна множина з H елементів: $M_2^2(t_3) = \{a_1^1(t_3), a_2^1(t_3), \dots, a_h^1(t_3)\}$. Таке перетворення проводиться допоки усі вихідні множини не стануть нульовими [5, 7, 9].

Множини $M_1^2(t_1); M_2^2(t_3); \dots; M_i^2(t_j)$ є початковими множинами другого рівня. Їх також будемо перетворювати за мережним алгоритмом до повної збіжності. Тоді перетворення H вихідних множин, що проводиться на першому рівні перетворення, елементами яких є елементи першого рівня, визначимо як другий рівень [5, 7].

При перетворенні множин другого рівня формуються елементи, які сформують початкові мно-

жини для третього рівня і так далі до k -го рівня, чий елементи вже не створюють нову множину. Всі перетворення відбуваються за тактами t_i , $i = 1, 2, 3, \dots$. У кожному такті для будь-якого рівня відбувається або вибір елементів із множин за F^* критерієм, або G перетворення (різновид Q^* -перетворення) множин відповідно до раніш обраних елементів, що свідчить про властивість синхронності мережі [5, 9].

В результаті вищевказаних перетворень з'являються окремі елементи, що не входять у жодну множину. Це відбувається тому, що такий елемент у цьому такті для свого рівня обраний як єдиний, не бере участі у подальшій обробці масивів та є її результатом. Такі елементи називаються хвостовими або елементами, що формують результат [5, 7, 9, 11]. При G перетворенні множин, можуть відзначатися наприклад рівні елементи. Виділимо інформацію про всі рівні елементи і їх розташування в множині. З цією метою для кожного G перетворення множини необхідно поставити у відповідність двійковий код, у якому «одиниці» знаходяться в тих розрядах, у яких позиціях множини знаходяться рівні елементи. Всі інші розряди коду, що відповідають іншим елементам множини, заповнюються «нулями». Під тінговою маскою (надалі просто маскою) G перетворення множини розуміють двійковий код, розрядність якого дорівнює потужності множини, а «одиниці» знаходяться в тих розрядах коду, що відповідають місцезнаходженню відмічених на цьому кроці елементів множини. Такі маски формуються для всіх проміжних і нульових множин у всіх гілках і рівнях [5, 7, 11].

Маски формуються в процесі всього перетворення масиву до його повної збіжності на всіх рівнях і для всіх гілок. Маски необхідні для процесу декодування і містять інформацію про те, на якій позиції (позиціях) у масиві повинний знаходитися обраний елемент. Оскільки маски є результатом перетворення масиву, то їх для будь-якого масиву повинно бути не більше, ніж елементів у масиві [5, 7, 9, 11]:

$$f_j^v \leq \omega_j^v, \quad (1)$$

де f_j^v — кількість масок, сформованих при перетворенні j -го масиву на v -му рівні; ω_j^v — розмірність початкового j -го масиву на v -му рівні.

Фактично масок стільки ж, скільки обраних елементів.

Нехай m — розмірність масиву, r — кількість однакових елементів та l — кількість груп з однаковими елементами. Тоді кількість масок при перетворенні такого масиву дорівнює:

$$f = m - r + l. \quad (2)$$

Маски кожного масиву володіють трьома властивостями [5, 7, 9]:

1. Маска на будь-якому кроці перетворення масиву має не менше однієї «одиниці»:

$$F_j^v(t_i) \neq 0. \quad (3)$$

2. «Одиниця» у кожному розряді маски по всіх масках у процесі обробки масиву зустрічається тільки один раз:

$$F_j^v(t_i) \wedge F_j^v(t_k), \quad \text{де } k \neq i; \quad i = \{\omega, \omega + 2, \dots, z - 2, z\}; \quad k = \{\omega, \omega + 2, \dots, z\}. \quad (4)$$

3. Диз'юнкція всіх масок масиву, якщо масив не містить нульових елементів, дорівнює коду з «одиницями» у всіх розрядах:

$$F_j^v(t_\omega) \vee F_j^v(t_{\omega+2}) \vee \dots \vee F_j^v(t_{z-2}) \vee F_j^v(t_z) = 2^n - 1, \quad (5)$$

де n — розрядність маски.

Можна виділити такі методи формування масок при кодуванні інформації в паралельно-ієрархічному (ПІ) перетворенні [5]:

1. Формування масок за означенням (1). Маски масиву подають як двійкові слова, розмірність яких дорівнює розмірності масиву. Цей метод формування масок можна використовувати в будь-яких алгоритмах паралельно-ієрархічного перетворення. Недоліком такого методу є громіздкість представлення масок, але алгоритм формування масок при цьому дуже простий.

2. Стековий метод формування масок, — полягає в тому, що масками є початкові адреси (номера позицій) підмножин з однаковими елементами або безпосередньо адреса елемента, для

якого формується маска. Цей метод дозволяє скоротити обсяги представлення масок, але потребує додаткових перетворень при кодуванні і декодуванні масиву.

3. Формування масок у логіко-часових кодах (ЛЧК) [5, 7, 12, 13]. Тут кожному елементу відповідає свій відрізок часу (квант часу) — ЛЧК. Якщо при кодуванні масиву зустрічається елемент, рівний обраному, то відповідний квант часу заповнюється імпульсом, у протилежному випадку його немає. Процес кодування масок проводиться паралельно для всіх різноманітних елементів масиву. Недоліком такого методу є необхідність додаткових перетворень при кодуванні і декодуванні масиву.

4. Досліджуваний у цій роботі метод оптимізованого формування масок при кодуванні інформації в III перетворенні. Він оснований на оптимізації явно надлишкового першого методу формування масок. Надмірність подання масок, розрядність яких дорівнює розмірності масиву, полягає в тому, що ті розряди масок, у яких утримується «одиниця», у всіх наступних масках заповнюються «нулями». Такі розряди з наступних масок можна виключити, що є основою запропонованого методу. При кодуванні масиву в цьому випадку кожна наступна маска має розмірність меншу за попередню на величину, що рівна кількості «одиниць» у попередній масці.

Проілюструємо це на прикладі кодування масиву даних, який складається з восьми елементів [5, 14]:

Номер елемента	Масив	Маски
1	3	1 0 → 0 → 1 1
2	3	1 1 → 0 → 0
3	10	0 0 → 1
4	3	1 0
5	3	1
6	4	0
7	12	0
8	7	0
Обрані елементи		3 4 7 10 12

Рис. 1. Приклад кодування масиву даних методом оптимізованого формування масок в паралельно-ієрархічному перетворенні

У цьому прикладі для збереження масок необхідна пам'ять обсягом 18 біт, у той час як для збереження повних масок (відповідно до першого методу формування масок) потрібно було б: $5 \times 8 = 40$ біт. Причому, останню маску можна взагалі не зберігати, тому що вона складається з усіх «одиниць». При декодуванні в передостанній масці на місці «одиниць» необхідно встановлювати передостанній, а на місці «нулів» — останній обраний елемент [5, 14].

Програмна реалізація методу оптимізованого формування масок при кодуванні інформації в III перетворенні та аналіз результатів

Розроблено програмний комплекс для реалізації прямого та зворотного III перетворення інформаційних середовищ, що містить два окремих програмних продукти [14]:

1. Програмний засіб для реалізації прямого III перетворення з оптимізацією формування масок. 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 III перетворення (рис. 2).

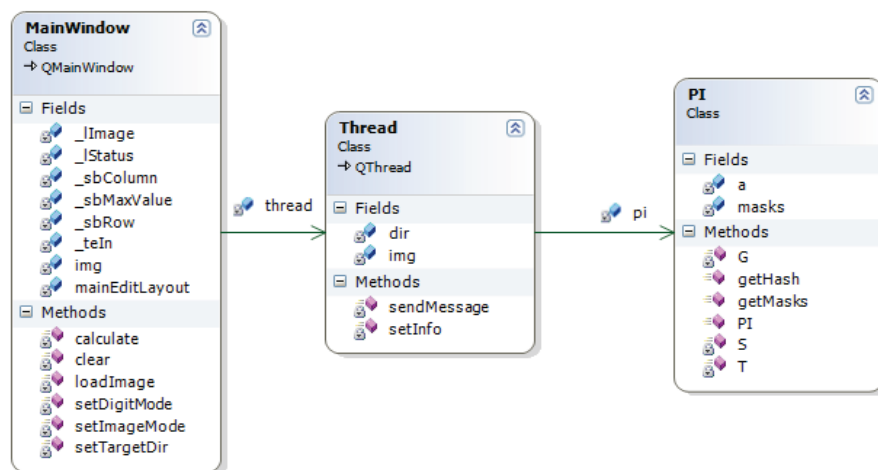


Рис. 2. Класова діаграма програмної бібліотеки для реалізації прямого ПІ перетворення з оптимізацією формування масок

2. Програмний засіб для реалізації зворотного ПІ перетворення на основі оптимізованого маскового методу для відновлення перетворених (за методом прямого ПІ 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 швидкодію декодування (рис. 3).

Мова реалізації програмного комплексу — «С++». Функції програмної бібліотеки, після перекompіляції, коректно працюють з різними операційними системами: MS Windows, GNU/Linux, Mac OS.

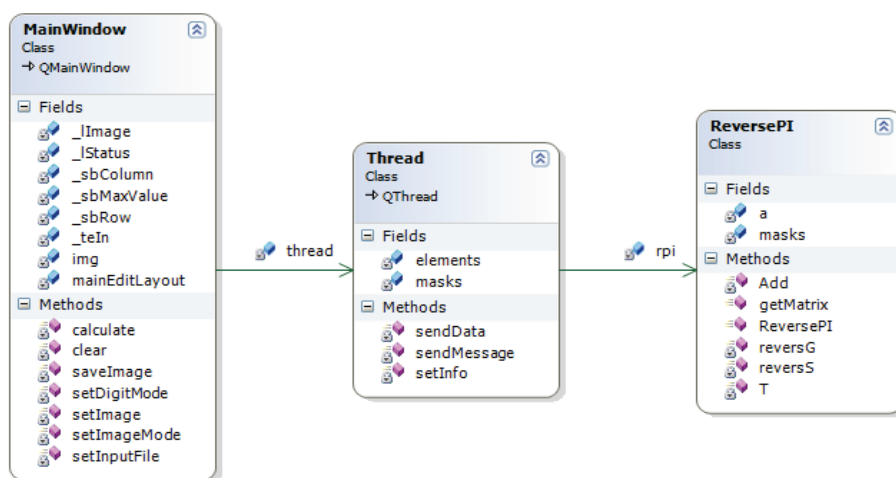


Рис. 3. Класова діаграма програмної бібліотеки для реалізації зворотного ПІ перетворення на основі оптимізованого маскового методу

Реалізація прямого ПІ перетворення з оптимізацією формування масок містить такі основні етапи:

1. Завантаження інформаційного масиву — у вигляді зображення або двовимірної матриці даних заданої користувачем розмірності.

2. Виконання прямого ПІ перетворення з оптимізацією формування масок над даними/зображенням (метод PI на рис. 2) — послідовне застосування трьох операторів $\Phi(M) = T [S (G (M))]$:

- 2.1 Транспонування (метод *T* на рис. 2);
 - 2.2 *G*-перетворення (метод *G* на рис. 2);
 - 2.3 Зсув (метод *S* на рис. 2); а також запам'ятовування одновимірної матриці хвостових елементів перетвореного інформаційного масиву (метод *getHash* на рис. 2).
3. Формування одновимірної матриці оптимізованих масок (метод *getMasks* на рис. 2).

4. Ведення файлу-протоколу.

Реалізація зворотного ПІ перетворення на основі оптимізованого маскового методу містить такі основні етапи:

1. Завантаження масиву хвостових елементів — у вигляді одновимірного набору даних, отриманих при кодуванні за методом прямого ПІ перетворення з оптимізацією формування масок;
2. Завантаження масиву оптимізованих масок — у вигляді одновимірного набору даних (з 0 та 1), отриманих при кодуванні за методом прямого ПІ перетворення з оптимізацією формування масок;
3. Виконання зворотного ПІ перетворення на основі оптимізованого маскового методу над заданими даними (метод ReversePI на рис. 3);
4. Відновлення початкового масиву даних — у вигляді двовимірної матриці даних або зображення (метод getMatrix на рис. 3).

Розроблений програмний засіб дозволяє завантажувати зображення, або ж задати розміри матриці для прямого та зворотного ПІ перетворення та заповнити задану матрицю власноруч або за допомогою генератора псевдовипадкових чисел. Після чого для заданого масиву даних виконується ПІ перетворення, виводяться значення хвостових елементів, їх сума та сума вхідної матриці (означені суми, згідно теорії ПІ перетворення, повинні збігатись), а також час обробки та узагальнений розмір масок.

Запропонований програмний комплекс для реалізації прямого та зворотного ПІ перетворення інформаційних середовищ протестовано на різноманітних наборах даних — у вигляді двовимірної матриці даних різноманітної розмірності, а також плямових зображень профілю лазерного променя різноманітної розмірності (рис. 4).

В наведеному тестовому прикладі здійснено реалізацію прямого ПІ перетворення з оптимізацією формування масок над кольоровим плямовим зображенням профілю лазерного променя (рис. 5), у форматі RGB, розмірністю 128×128 пікселів. А також — зворотне ПІ перетворення на основі оптимізованого маскового методу для його відновлення. Результати проведених експериментальних досліджень (і тестового прикладу також) переконливо демонструють підвищення ефективності ПІ перетворення (в контексті кодування-декодування інформації, і зображень, зокрема) за критеріями:

- 1) швидкодії обробки прямого ПІ перетворення: 984 мс — для неоптимізованого (рис. 6, графа «Time») та 782 мс — для оптимізованого метода (рис. 7, графа «Time»);
- 2) швидкодії обробки зворотного ПІ перетворення: 1250 мс — для неоптимізованого (рис. 8, графа «Time») та 422 мс — для оптимізованого метода (рис. 9, графа «Time»);
- 3) зменшення надмірності обсягів пам'яті при формуванні масивів масок: 25057800 біт — для неоптимізованого (рис. 6, графа «Mask size») та 10187072 біта — для оптимізованого метода (рис. 7, графа «Mask size»).

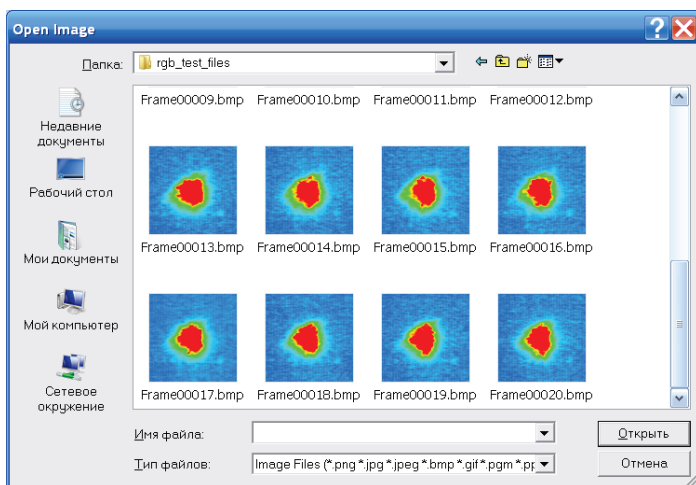


Рис. 4. Завантаження зображень профілю лазерного променя

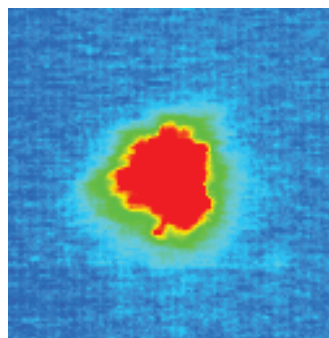


Рис. 5. Обране зображення профілю лазерного променя для тестового прикладу

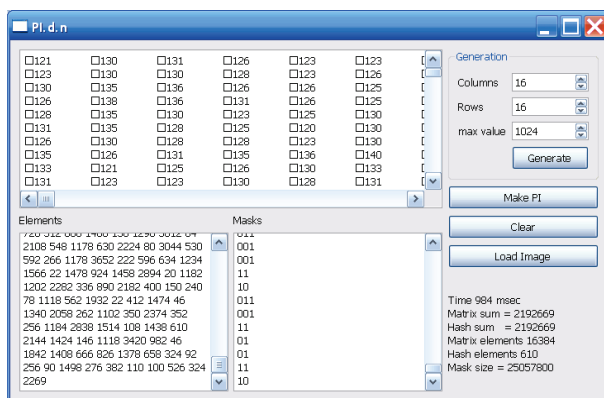


Рис. 6. Пряме 3D перетворення без оптимізації формування масок

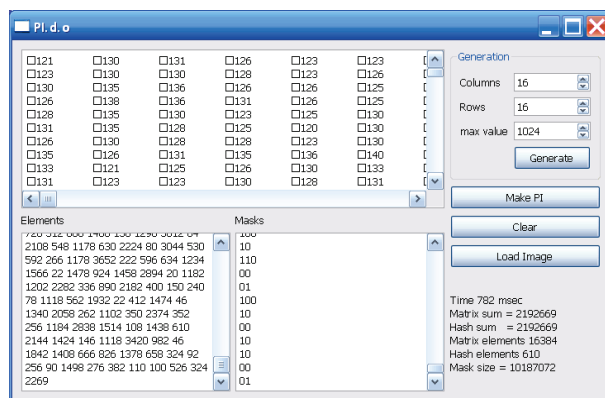


Рис. 7. Пряме 3D перетворення з оптимізацією формування масок

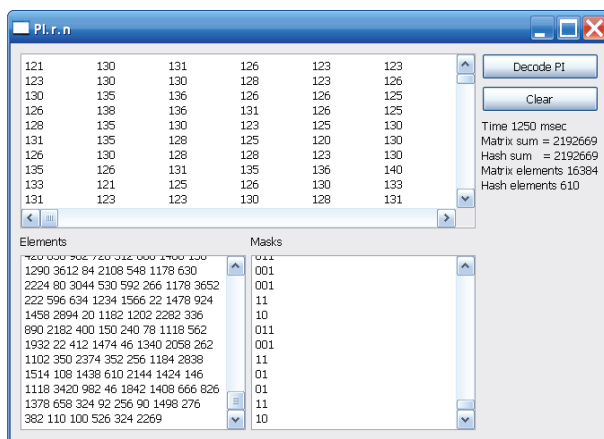


Рис. 8. Зворотнє 3D перетворення без оптимізації масок

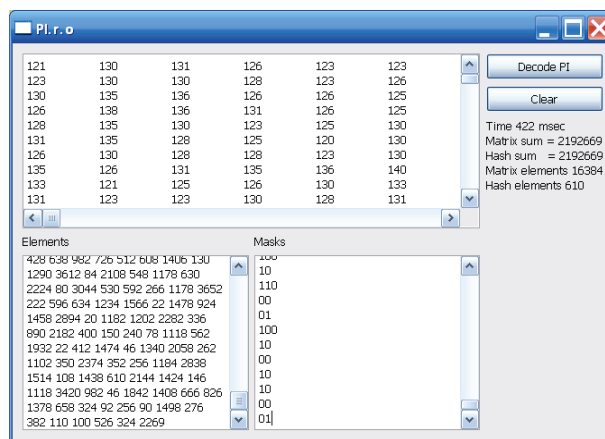


Рис. 9. Зворотнє 3D перетворення з оптимізацією масок

Також, результати роботи програмного комплексу збігаються з результатами математичного та комп'ютерного моделювання, що свідчить про коректність та достовірність роботи програмного продукту [14].

Результати роботи можуть знайти застосування у експериментальному дослідженні функціонування не лише 3D інтелектуальних обчислювальних систем, але й для подальших досліджень з розробки високопродуктивних ієрарх-ієрархічних мереж на основі оптико-електронної та оптичної елементної бази.

Висновки

Розглянуто основні положення побудови та організації обчислювальних пристроїв із пірамідалною обробкою та 3D перетворенням інформації, а також здійснено дослідження їх ефективності.

В ході досліджень досягнуто основну мету — підвищено ефективність прямого та зворотного 3D перетворення за рахунок підвищення швидкодії перетворення зображень, а також оптимізації надмірності подання масок, розрядність яких дорівнювала розмірності масиву даних, що обробляється.

Проаналізовано відомі способи формування та подання масок із застосуванням маскового методу зворотного 3D перетворення для обробки зображень (неоптимізований, в якому маски масиву подають двійкові слова, розмірність яких дорівнює розмірності масиву, а також стековий та на основі ЛЧК). На основі проведеного порівняльного аналізу в ході досліджень здійснено розробку методу подання масок для оптимізації зворотного 3D перетворення зображень, в якому при кодуванні масиву кожна наступна маска має розмірність менше за попередню на кількість «одиниць» у попередній масці, що приводить до суттєвого зменшення обсягів пам'яті, необхідної для збереження масок, а також містить властивості виявлення помилок кодування.

Наведено попередні результати імітаційного моделювання та програмної емуляції запропоно-

ваного методу подання масок для оптимізації зворотного ПІ перетворення зображень, а також створено набір демонстраційних додатків, що дають змогу оцінити достовірність роботи програмного продукту та показують ефективність застосування в прикладних задачах кодування, оброблення та порівняння зображень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Прэтт У. Цифровая обработка изображений : в 2-х т. / У. Прэтт. — М. : Мир, 1982. — Т. 1. — 310 с., Т. 2. — 790 с.
2. Гренандер У. Лекции по теории образов. Анализ образов / У. Гренандер. — М. : Мир, 1981. — 448 с.
3. Скляр Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр. — [изд. 2-е., испр.] ; [пер. с англ.] — М. : Издательский дом «Вильямс», 2003. — 1104 с.
4. Сергиенко А. Б. Цифровая обработка сигналов : учеб. пос. / А. Б. Сергиенко. [2-е изд.]. — СПб. : Питер, 2006. — 752 с.
5. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту : моног. / [В. П. Кожем'яко, Ю. Ф. Кутаєв, С. В. Свечніков та ін.] — Вінниця : УНІВЕРСУМ–Вінниця, 2003. — 324 с.
6. Александров В. В. Представление и обработка изображений: рекурсивный подход / В. В. Александров, Н. Д. Горский. — Л. : Наука, 1985. — 192 с.
7. Вступ в алгоритмічну теорію ієрархії і паралелізму нейроподібних обчислювальних середовищ та її застосування до перетворення зображень. Основи теорії пірамідально-сітвового перетворення зображень / [В. П. Кожем'яко, Л. І. Тимченко, Ю. Ф. Кутаєв, І. Д. Івасюк] — К. : УМК ВО, 1994. — 272 с.
8. СБИС для распознавания образов и обработки изображений / под ред. К. Фу. — М. : Мир, 1988. — 248 с.
9. Тимченко Л. И. Многоэтапная параллельно-иерархическая сеть как модель нейроподобной схемы вычислений / Л. И. Тимченко // Кибернетика и системный анализ. — 2000. — № 2. — С. 114—134.
10. Тимченко Л. И. Конвергентні та дивергентні процеси в реальних та штучних нейронних мережах / Л. І. Тимченко // Вісник Вінницького політехнічного інституту. — 1997. — № 1. — С. 5—10.
11. Яровий А. А. Методологічні особливості побудови паралельно-ієрархічних та ієрарх-ієрархічних мереж на основі кластерних систем з розподіленою обробкою інформації / А. А. Яровий // Оптико-електронні інформаційно-енергетичні технології. — 2010. — № 1 (19). — С. 69—79.
12. Образний відео-комп'ютер око-процесорного типу : моног. / [В. П. Кожем'яко, Г. Л. Лисенко, А. А. Яровий, А. В. Кожем'яко]. — Вінниця : Універсум–Вінниця, 2008. — 215 с.
13. Кожемяко В. П. Модели параллельно-иерархической обработки информации на основе аппарата логико-временных функций / В. П. Кожемяко, Л. И. Тимченко, А. А. Яровой // Информационные и компьютерные технологии, моделирование, управление : тезисы докл. (1—4 ноября 2010 г.). — Тбилиси : Изд-во ГТУ, 2010. — С. 231—232.
14. Розробка методу подання масок для оптимізації зворотного паралельно-ієрархічного перетворення зображень / Яровий А., Сугак І. // Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування : матеріали V Міжнарод. наук.-техн. конф. — Вінниця, 19—21 травня 2011 р. — Вінниця : ВНТУ, 2011 — 25 с.

Рекомендована кафедрою комп'ютерних наук

Стаття надійшла до редакції 8.11.11
Рекомендована до друку 29.11.11

Яровий Андрій Анатолійович — доцент кафедри комп'ютерних наук.

Кафедра комп'ютерних наук, Вінницький національний технічний університет, Вінниця