

О. Ю. Софина, к. т. н., доц.; А. В. Лозун

РОЗРОБКА ТА ДОСЛІДЖЕННЯ МОДИФІКОВАНОГО АЛГОРИТМУ ФРАКТАЛЬНОГО КОДУВАННЯ

У статті запропоновано модифікацію алгоритму фрактального кодування шляхом використання дискретного косинусного перетворення, вейвлет-перетворення Хаара й паралельних обчислень. Розроблений алгоритм має на меті зменшення обчислювальної складності алгоритму фрактального кодування й підвищення швидкодії процесу стиснення зображень.

Ключові слова: стиснення зображень, фрактальне кодування, дискретне косинусне перетворення, вейвлет-перетворення, паралельні обчислення.

Вступ

Сьогодні завдяки значному поширенню електронно-обчислювальної техніки зберігання й обробка даних різноманітного типу здійснюється переважно в цифровому вигляді. Усе більшої популярності набувають мультимедійні типи даних: відео-, аудіозаписи й цифрові зображення. Виходячи з вищеприведеного, однією з найактуальніших проблем сучасних інформаційних технологій є розробка ефективних методів компресії мультимедійних даних, зокрема графічної інформації.

До мультимедійної інформації найчастіше застосовують методи стиснення із втратами. Такі застосування є справедливими, оскільки для мультимедійних об'єктів, зазвичай, можна відмовитися від зберігання певних їхніх особливостей (наприклад, дрібних деталей на зображенні), що дозволить підвищити ступінь стиснення. Але існують поширені алгоритми стиснення без втрат, такі як: FLAC для звукових файлів або PNG для цифрових зображень. За використання цих форматів варто враховувати, що їх проектували як універсальні для свого типу даних – і в результаті вони виявились непридатними для врахування особливостей конкретного файлу, що підлягає стисненню. Як наслідок відбувалось значне погіршення коефіцієнта стиснення порівняно з аналогічними алгоритмами компресії з втратами.

Для цифрових зображень як класу мультимедійних даних найпоширенішим форматом стиснення з втратами є JPEG. Він набув значного використання завдяки поширенню цифрових фотоапаратів, сканерів тощо. З урахуванням кількості представлених у форматі JPEG зображень стають очевидними значні втрати, пов'язані зі зберіганням, передачею і обробкою, можливо, неоптимально (за якістю і ступенем компресії) стисненої інформації.

Отже, набувають актуальності дослідження методів стиснення, які ґрунтуються на інших представленнях зображень, до яких належить фрактальне кодування, особливістю якого є властивість самоподібності зображення. Проте сучасні методи фрактального стиснення зображень потребують суттєвого розвитку з одночасним урахуванням багатьох критеріїв (зокрема, швидкості роботи, ступеня стиснення, якості під час декомпресії) для того, щоб їх можна було розглядати в якості реальної альтернативи JPEG для багатьох класів зображень, які використовують у науково-технічній і повсякденній сферах життєдіяльності людини.

Розв'язанню проблем і завдань фрактального кодування присвячено велику кількість досліджень вітчизняних (Д. С. Ватолін, В. В. Сергєєв, В. А. Сойфер, В. В. Александров, Н. Д. Горський) і зарубіжних (М. Барнслі, А. Жакен, Ю. Фішер, Д. Заупе) учених [1 – 3]. Однак і досі є куди рухатися в цьому напрямку.

Метою цієї статті є побудова модифікованого алгоритму фрактального кодування, що дозволить підвищити швидкодію процесу стиснення зображень, а також досягнути прийняттого коефіцієнта стиснення шляхом поєднання відомих технологій ДКП, вейвлет-перетворення і паралельних обчислень.

Основними завданнями, поставленими для досягнення мети, є розробка та дослідження

оптимального способу розбиття зображення на доменні й рангові блоки, розробка індексної й дескрипторної методики зіставлення доменних і рангових блоків та поєднання розроблених методик для утворення цілісного алгоритму.

Розбиття зображення на доменні і рангові блоки

Зображення розбивають на рангові блоки однакового розміру r і на всі можливі доменні блоки розміром $2r$. Цю множину доменів називають основною. Далі формують додаткову множину доменів таким чином: для кожного з «основних» доменів будують сім трансформацій, а саме: повороти блока на 90° , 180° і 270° , відображення блока відносно вертикальної осі симетрії і повороти отриманого відображеного блока також на 90° , 180° і 270° . У результаті поєднання основної і додаткової множин доменів формують розширену множину доменних блоків. Для кожного рангового блока проводять пошук серед розширеної множини доменів. Для цього кожному ранговому і доменному блока присвоюють індекс і дескриптор.

Наведений спосіб розбиття зображення на доменні і рангові блоки є оптимальним, оскільки кожен блок може бути однозначно ідентифікований його порядковим номером (за попередньо заданого фіксованого способу нумерації): для зберігання номера блока необхідна менша кількість біт, ніж, наприклад, для координат його верхнього кута. Ще однією перевагою є відсутність необхідності вказувати розмір кожного блока в стисненому файлі: достатньо один раз у заголовку файла вказати розмір рангового блока, а також коефіцієнт пропорційності для визначення розміру доменного блока.

Розрахунок індексів для доменних і рангових блоків

Із метою обчислення індексу для кожного блока b обчислюють матрицю B його двовимірного дискретного косинусного перетворення (ДКП). Виходячи з аналізу роботи [4], пропонуємо такий алгоритм розрахунку індексу з використанням отриманих ДКП-матриць:

– розглядають «верхній лівий кут» матриці B , яка відповідає поточному блока b (під матриця $\hat{B} = \{B_{ij}\}_{i,j=1,3}$).

– згідно з порядком вибору елементів для обчислення індексу, наведеного на рис. 1, оцінюють елементи цієї підматриці з 1-го до t -го, де t – обрана розмірність індексу, $t \in [1, 8]$. Якщо поточний елемент менший за нуль, то відповідному розряду індексу у двійковому представленні, починаючи зліва, присвоюють значення 0, якщо більший за нуль або дорівнює нулю – значення 1.

	1	5		
2	4	6		
3	7	8		

Рис. 1. Порядок вибору елементів для обчислення індексу

Застосовуючи до блока b повороти чи відображення, правила зміни ДКП-матриці B задають такими рівняннями:

$$\begin{cases} F'_{ij} = (-1)^{i+1} F_{ij}, f'_{ij} = f_{N-i,j}, \\ F'_{ij} = (-1)^{j+1} F_{ij}, f'_{ij} = f_{i,N-j}, \\ F'_{ij} = F_{ij}, f'_{ij} = f_{i,j}, \end{cases} \quad (1)$$

де $i, j = \overline{1, N}$.

Звідси випливає, що ДКП-матриці (відповідно, і індекси) «додаткових» доменів можна знайти досить просто на основі попередньо обчислених ДКП-матриць «основних» доменів.

Розрахунок дескрипторів для доменних і рангових блоків

Для обчислення дескрипторів використовують двовимірне дискретне вейвлет-перетворення Хаара [5]. Пропонуємо такий алгоритм формування дескриптора для блока b розміром $p \times p$ пікселів, де $p=2q$, $q \in \mathbb{N}$:

- обчислюють коефіцієнт e_1 :

$$e_1 = \frac{1}{p^2} \sum_{i=1}^p \sum_{j=1}^p b_{ij}. \quad (2)$$

- блок розбивають на чотири рівних підблоки b_2, b_3, b_4 і b_5 .
- послідовно обчислюють коефіцієнти

$$\begin{cases} e_2 = e_1 - \frac{1}{q^2} \sum_{i=1}^q \sum_{j=1}^q b_{ij} \\ e_3 = e_1 - \frac{1}{q^2} \sum_{i=1}^q \sum_{j=q+1}^q b_{ij} \\ e_4 = e_1 - \frac{1}{q^2} \sum_{i=q+1}^q \sum_{j=1}^q b_{ij} \\ e_5 = e_1 - \frac{1}{q^2} \sum_{i=q+1}^q \sum_{j=q+1}^q b_{ij} \end{cases} \quad (3)$$

що відповідають під блокам b_2, b_3, b_4 і b_5 .

- формують вектор $e = [e_2 e_3 e_4 e_5]$, який і буде шуканим дескриптором.

Як і обчислення індексу, формування дескриптора для додаткового домену може бути спрощено, а саме: замість розрахунку за вищенаведеним алгоритмом можна здійснити просту перестановку компонент дескриптора основного домену відповідно до конкретної трансформації блока.

Узагальнений алгоритм пошуку оптимальних доменів

1. Розглядають перший ранговий блок r_1 .

2. Формують підмножину D_{r_1} розширеної множини D доменів шляхом вибірки всіх доменів із значенням індексу, що повністю або частково збігається, методом ієрархічного пошуку (домени, що належать D_{r_1} , названі такими, що мають «первинну подібність» із ранговим блоком, який розглядають, або просто «первинними»).

3. Дескриптор блока r_1 по черзі порівнюють із дескрипторами кожного з доменів, що належать D_{r_1} .

4. Формують підмножину D_1 множини D_{r_1} шляхом вибору певної кількості доменів, дескриптори яких максимально наближені до дескриптора рангового блока r_1 в сенсі евклідової відстані (домени, що належать D_1 називаються такими, що володіють «вторинною подібністю» з ранговим блоком, який розглядається, або просто «вторинними»).

5. Серед знайдених «вторинних» доменів шукають домен d_1 , а також відповідні парі $r_1 - d_1$ коефіцієнти яскравості та контрастності s_1 і o_1 , виходячи з умов мінімізації СКВ: $d_1 = \operatorname{argmin}(\text{СКВ}(r_1, \tilde{d}_1))$, де $\tilde{d}_1 = s\tilde{d}_1 + oE$, \tilde{d}_1 – стиснений до розміру блока r_1 блок d_1 , а E – одинична матриця розміру блока r_1 . Доменний блок, для якого буде виконуватися наведена умова мінімізації СКВ, називається оптимальним або найкращим.

6. Якщо знайдений на попередньому кроці домен d_1 належить «основній» множині

доменів, фіксують його порядковий номер v_1 , якщо ж «додатковий» – фіксують номер трансформації τ_1 , шляхом якого він був отриманий з «початкового» домену, а також номер цього «початкового» домену \tilde{v}_1 .

7. Розглядають другий ранговий блок r_2 , для якого повторюють кроки 2 – 6, а потім ранговий блок r_3 і так далі, доки не буде знайдено оптимальний домен d_N для останнього рангового блока r_{N_r} , де N_r – кількість рангових блоків.

На рис. 2 наведена схема запропонованого алгоритму пошуку оптимальних доменів, яка враховує розпаралелення обчислень на рівні рангових блоків, оскільки для кожного з них необхідно виконувати певну кількість незалежних операцій.

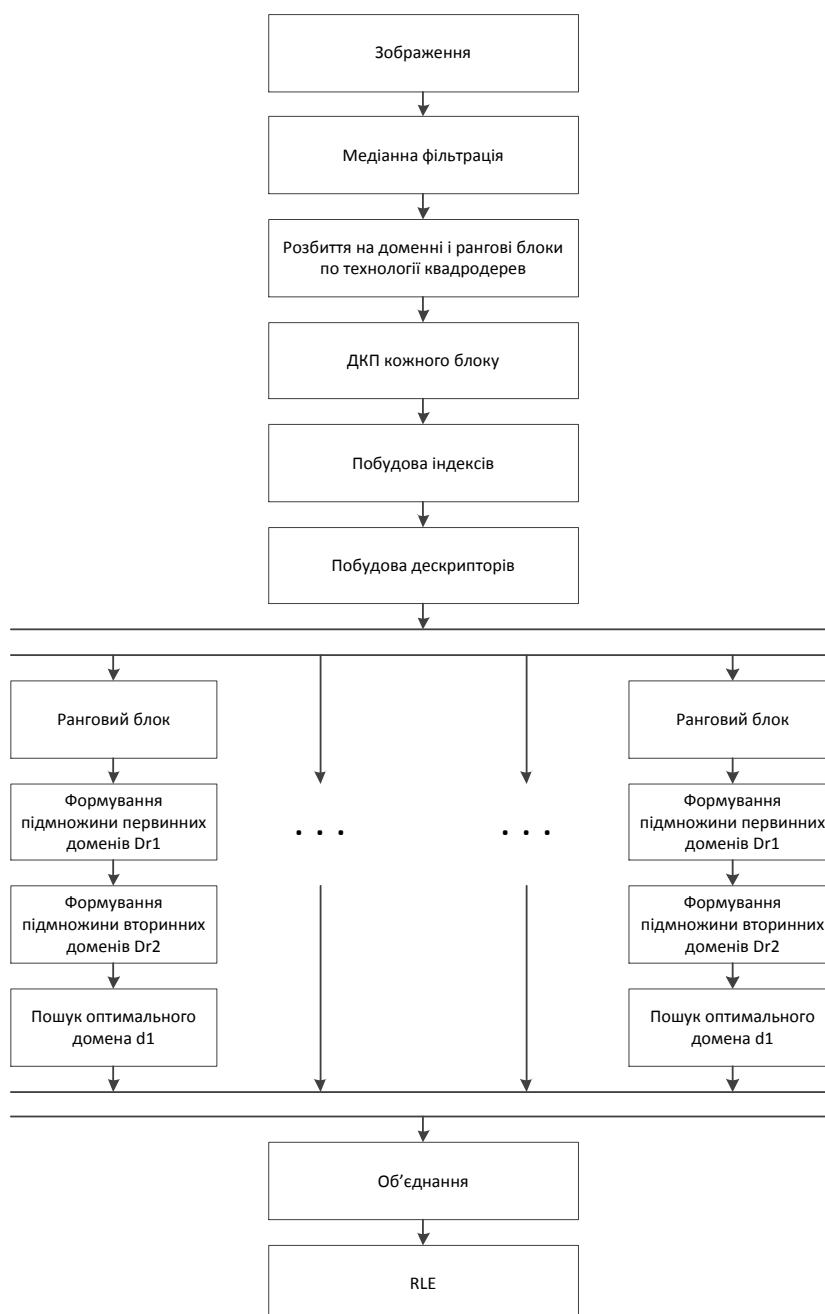


Рис. 2. Алгоритм пошуку оптимальних доменів

Модифікація формули обчислення середньоквадратичного відхилення

Модифікація формули обчислення СКВ роблять таким чином, щоб зменшити обчислювальне навантаження на систему.

Загальна формула перетворення значень пікселів доменного блока виглядає так:

$$D_i^* = sD_i + o, \quad (4)$$

де D_i^* і D_i – перетворений і початковий i -ий доменний блок відповідно; s – коефіцієнт зміни контрасту; o – коефіцієнт зсуву за яскравістю.

Для оцінки розбіжності (відстані) між перетвореним доменним та поточним ранговим блоками необхідно ввести відповідну метрику. Зазвичай використовують функцію середньоквадратичного відхилення (СКВ):

$$Q = \sum_{i=1}^N (D_i^* - R_i)^2 = \sum_{i=1}^N ((sD_i + o) - R_i)^2, \quad (5)$$

де R_i – i -ий ранговий блок; D_i^* і D_i – перетворений і початковий i -ий доменний блок відповідно; N – кількість пікселів у ранговому блоці.

Очевидно, що чим менша відстань між блоками, тим більше вони подібні.

Коефіцієнти s і o можна знайти з формули (5), узявши часткові похідні за цими змінними. Розкриємо квадрат у виразі (5):

$$Q = \sum_{i=1}^N s^2 D_i^2 + 2s o D_i + o^2 - 2s R_i D_i - 2o R_i + R_i^2. \quad (6)$$

Маємо таку умову:

$$\begin{cases} \frac{\partial Q}{\partial s} = \sum_{i=1}^N (2s D_i^2 + 2D_i o - 2R_i D_i) = s \sum_{i=1}^N D_i^2 + o \sum_{i=1}^N D_i - \sum_{i=1}^N R_i D_i = 0 \\ \frac{\partial Q}{\partial o} = \frac{1}{N} \sum_{i=1}^N R_i - \frac{1}{N} s \sum_{i=1}^N D_i \end{cases} \quad (7)$$

Виразимо зсув за яскравістю:

$$o = \frac{1}{N} \sum_{i=1}^N R_i - \frac{1}{N} s \sum_{i=1}^N D_i \quad (8)$$

Підставимо (8) у рівняння частинної похідної по s (7) і отримаємо такі формули для знаходження коефіцієнтів:

$$\begin{cases} s = \frac{N \sum_{i=1}^N R_i D_i - \sum_{i=1}^N R_i \sum_{i=1}^N D_i}{\sum_{i=1}^N D_i^2 - (\sum_{i=1}^N D_i)^2} \\ o = \frac{1}{N} (\sum_{i=1}^N R_i - s \sum_{i=1}^N D_i) \end{cases} \quad (9)$$

Перетворивши формулу (3), отримаємо вираз для знаходження відстані:

$$Q = s^2 \sum_{i=1}^N D_i^2 + N o^2 + \sum_{i=1}^N R_i^2 - 2s \sum_{i=1}^N R_i D_i + 2s o \sum_{i=1}^N D_i - 2o \sum_{i=1}^N R_i \quad (10)$$

Формули (6, 7) дозволяють спростити обчислювальне навантаження, оскільки суми $\sum_{i=1}^N R_i$, $\sum_{i=1}^N R_i^2$, $\sum_{i=1}^N D_i$, $\sum_{i=1}^N D_i^2$ можна обрахувати ще до початку перебору, коли вже сформовані множини рангових і доменних блоків. Тоді на етапі зіставлення необхідно обчислити лише суму $\sum_{i=1}^N R_i D_i$ і знайти коефіцієнти.

Експериментальні результати тестування розробленого методу

Проведемо спочатку дослідження класичного алгоритму фрактального стиснення. Для цього була сформована тестова вибірка, яка містить різні за своїм вмістом і структурою типи зображень, що трапляються найчастіше:

- наземні зображення сцен штучних об'єктів;
- наземні зображення природних об'єктів;

- аерокосмічні зображення;
- астрономічні зображення;
- зображення неструктурованих об'єктів, таких як: хмари або дим.

На рис. 3 наведені приклади зображень цих класів, які далі використовуватимуться з метою дослідження.

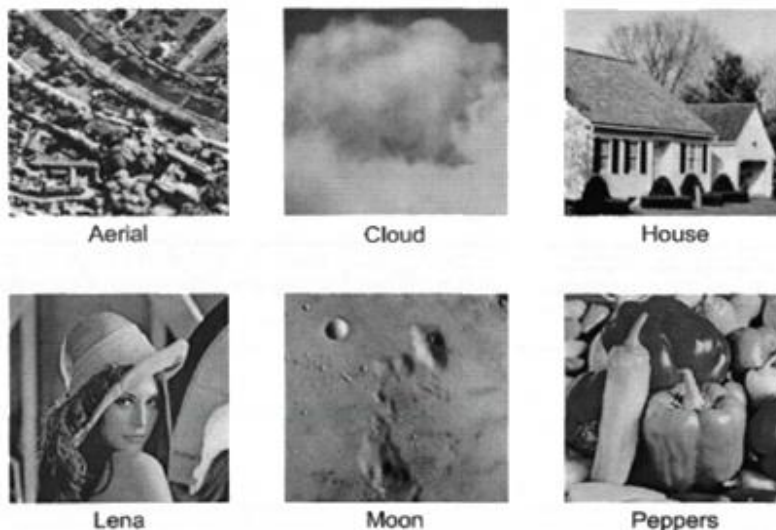


Рис. 3. Тестові зображення

Розглянемо показники ефективності, досягнені класичним алгоритмом фрактального стиснення. У табл. 1 наведені значення часу роботи кодера, середньоквадратичного відхилення відновленого зображення після декомпресії від вихідного, співвідношення сигнал-шум, а також компоненти критерію мінімальної довжини опису (МДО) і його підсумкового значення для зображень різного розміру – 60, 120, 180 і 240 пікселів, стиснених за фіксованого розміру рангового блока, що становить 6 пікселів. Дано означення критерію МДО. Принцип мінімальної довжини опису сформулюємо так: серед множини моделей необхідно обрати ту, яка дозволяє мінімізувати суму довжини опису моделі (у бітах) і довжину даних, описаних шляхом застосування цієї моделі (у бітах). Під моделлю розуміють певну інформаційну структуру, яка являє собою закодоване зображення. У якості довжини опису моделі будемо розглядати обсяг L_{img} стисненого зображення в бітах. Під довжиною даних, описаних шляхом застосування моделі, розуміють ту частину даних, яка не увійшла в саму модель. Іншими словами, це обсяг втрат L_{loss} унаслідок стиснення.

Отже, критерій МДО можна розрахувати за формулою:

$$L = L_{loss} + L_{img} \quad (11)$$

Розглянемо умовні позначення, які використовують у табл. 1. Тут N – розмір зображення, t – час обчислень у секундах, σ – середнє квадратичне відхилення (СКВ) відтвореного зображення від оригінального, PSNR – співвідношення сигнал-шум, L_{loss} – оцінка обсягу втрат інформації за стиснення в бітах, L_{img} – оцінка довжини стисненого файлу в бітах, L – сума значень L_{loss} і L_{img} – мінімальна довжина опису стисненого зображення. Мінімізацію цієї довжини здійснюють по всіх можливих відповідностях доменних і рангових блоків за їхніх фіксованих розмірів.

Таблиця 1

Результати тестування класичного алгоритму для зображень різного розміру

N	t	σ	PSNR	L_{img}	L_{loss}	L
Aerial						
60	0,82	32,365	16,685	3022	25233	28255
120	16,04	21,336	21,547	13014	92574	105588
180	91,10	13,840	25,307	30421	187285	217706
240	287,56	12,329	26,311	55485	322719	378204
Cloud						
60	0,82	3,343	35,133	3023	12913	15936
120	15,97	1,720	40,903	13014	38951	51965
180	87,24	1,146	44,429	30421	69578	99999
240	286,70	1,121	44,665	55485	124105	179590
House						
60	0,82	21,403	21,520	3023	21725	24748
120	16,03	13,743	25,367	13014	78847	91861
180	86,64	9,088	28,959	30421	159497	189918
240	301,86	8,765	29,138	55485	281237	336722
Lena						
60	0,82	14,308	24,121	3023	20531	23554
120	15,94	8,918	28,561	13014	69354	82368
180	86,44	6,216	31,731	30421	135884	166305
240	288,86	6,086	31,916	55485	238511	293996
Moon						
60	0,82	8,141	29,916	3023	16859	19882
120	16,04	5,685	32,652	13014	62723	75737
180	86,43	4,100	35,806	30421	127659	158080
240	298,76	4,414	35,196	55485	235437	290922
Peppers						
60	0,82	16,944	22,345	3023	21298	24321
120	15,97	9,946	27,757	13014	71099	84113
180	86,34	6,019	32,538	30421	134361	164782
240	284,15	5,334	32,767	55485	228856	284341

Розглянемо, наскільки високою є якість стиснення з використанням класичного фрактального алгоритму. Оскільки досить гарною якістю відновленого зображення зазвичай вважають величину PSNR~30, у табл. 2 наведені оцінки коефіцієнтів стиснення у разі відновлення зображення з такою якістю шляхом ділення вихідного числа біт у зображенні на значення L_{img} . Це значення відповідає найближчому до 30 значенню PSNR в табл. 1. Крім того в табл. 2 також наведені коефіцієнти стиснення для формату JPEG за налаштувань, які відповідають PSNR~30.

Таблиця 2

Коефіцієнти стиснення за PSNR~30

Зображення	Коефіцієнт стиснення	
	Фрактальний алгоритм	JPEG
Aerial	3,6	4,4
Cloud	59,4	47,3
House	8,2	8,0
Lena	14,7	15,8
Moon	33,3	24,2
Peppers	14,8	15,1

Варто зазначити, що за використання стандартних програмних засобів коефіцієнт

стиснення JPEG може бути набагато гіршим через неоптимальні налаштування й запис великого обсягу додаткової інформації у файл.

Як видно з таблиці, коефіцієнти стиснення суттєво відрізняються для різних зображень.

Протестуємо тепер розроблений модифікований алгоритм фрактального кодування і порівнюємо результати.

Оскільки основним недоліком класичного алгоритму є швидке зростання часу роботи із збільшенням розмірів зображень, очевидно є неперспективність його використання на практиці. У цій статті описані розроблені методи пошуку відповідностей доменних і рангових блоків у фрактальному кодуванні. Виконаємо експериментальну перевірку розробленого алгоритму й визначимо втрати якості зображення. Оскільки підвищення швидкодії не впливає на розміри стисненого файла, достатньо буде розглянути значення СКВ і PSNR для відновленого зображення.

У табл. 3 наведені характеристики якості відновленого зображення, отриманого після стиснення модифікованим алгоритмом, а також час його роботи на зображеннях різних розмірів.

Таблиця 3

Результати тестування модифікованого алгоритму

N	t	σ	PSNR	L_{img}	L_{loss}	L
Aerial						
60	0,09	35,010	16,120	3022	25626	28648
120	0,48	24,425	20,373	13014	95273	108287
180	1,68	16,261	23,906	30421	194581	225002
240	3,93	14,431	24,943	55485	335408	390893
Cloud						
60	0,09	4,021	33,532	3023	13659	16682
120	0,50	1,992	39,628	13014	41788	54802
180	1,74	1,312	43,251	30421	75539	105960
240	4,01	1,235	43,780	55485	131567	187052
House						
60	0,09	25,520	19,992	3023	22599	25622
120	0,47	17,643	23,198	13014	82894	95908
180	1,60	11,260	27,098	30421	167840	198261
240	4,14	15,148	24,522	55485	300273	355758
Lena						
60	0,09	17,702	22,158	3023	21563	24586
120	0,53	10,83	27,429	13014	73525	86539
180	1,66	7,389	30,374	30421	143374	173795
240	4,10	7,076	30,497	55485	248954	304439
Moon						
60	0,09	9,981	27,097	3023	17621	20644
120	0,45	6,564	31,785	13014	65029	78043
180	1,62	4,755	34,342	30421	133436	163857
240	4,05	5,000	33,909	55485	244204	299689
Peppers						
60	0,09	22,30	20,072	3023	22770	25793
120	0,48	12,073	26,181	13014	75027	88041
180	1,60	7,388	30,303	30421	142956	173377
240	4,04	6,416	31,671	55485	241339	296824

Як видно з табл. 3, швидкість роботи не лише вища порівняно з класичним алгоритмом (у 30 разів на зображеннях розміром 120 пікселів), а й має іншу залежність від розміру зображення. Унаслідок цього на зображеннях розміром 60 пікселів отримуємо 8-кратний вигреш, а на зображеннях розміром 240 пікселів – уже 70 – 75-кратний вигреш.

Шляхом порівняння таблиць 1 і 3 можна зробити висновок, що критерій якості

мінімальної довжини опису (МДО) в результаті використання оптимізованого алгоритму погіршується в середньому приблизно на 4%. При цьому для частини зображень спостерігаємо постійне зростання погіршення (наприклад, з 1,5% до 3,5% для Aerial) за збільшення розміру зображення, тоді як для інших зображень може відбуватись навіть зменшення погіршення. Отже, вигреш у швидкодії порівняно з класичним алгоритмом значно зростає за збільшення розміру зображення, тоді як програш у коефіцієнті стиснення від розміру майже не залежить або не залежить зовсім.

У цій роботі також було запропоновано застосування паралельних обчислень. Зазначимо, що розпаралелення обчислень застосовувалось уже до оптимізованого алгоритму фрактального кодування, який був описаний вище. На рис. 4 наведений графік, що відображає результати порівняння послідовного і паралельного алгоритму. Оскільки розпаралелення обчислень не залежить від структури зображення, а також ніяк не впливає на якість відтвореного зображення, для тестування було обрано 4 зображення різних розмірів. Отже, розглянемо залежність часу стиснення від розміру зображення. Робота паралельного алгоритму в цьому випадку реалізована для 16-процесорної системи.

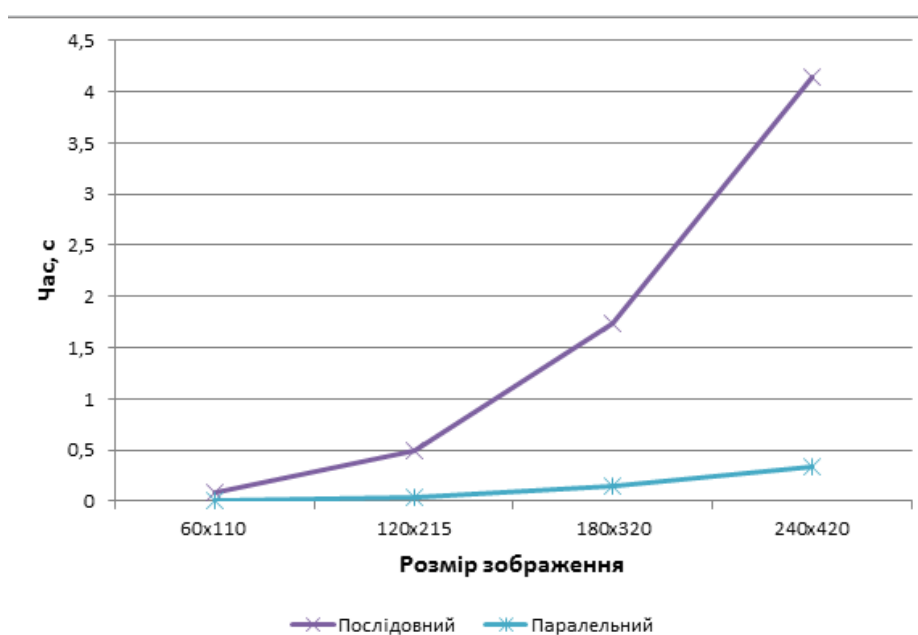


Рис. 4. Залежність часу виконання від розміру зображення

Як бачимо, швидкість обробки зображень за паралельним алгоритмом набагато вища, ніж час роботи послідовного алгоритму. Було встановлено, що використання кожного додаткового процесора призводить до підвищення швидкодії на 60 – 75%.

Висновки

Розроблений алгоритм пошуку оптимальних доменів, який безпосередньо являє собою модифікований алгоритм фрактального кодування, дозволяє підвищити швидкість обробки зображення за рахунок використання ДКП, вейвлет-перетворення Хаара і паралельних обчислень, а також дозволяє зменшити обчислювальне навантаження на систему.

Оскільки фрактальне кодування належить до NP-повних задач, прийнятний розв'язок таких задач досягається лише за використання ефективних предметно-залежних евристик. Більшість робіт із фрактального стиснення присвячені саме розробці подібного типу евристик.

Для субоптимального виконання фрактального стиснення необхідно обмежити перебір доменних блоків, рангових блоків і відповідностей між ними, що й було запропоновано в цій статті. Евристики мають відкидати неперспективні способи розбиття зображення на рангові блоки без виконання подальших кроків стиснення, а також за фіксованого розбиття зображення на рангові блоки – відкидати неперспективні підмножини доменних блоків для певного рангового блока. У цій роботі формування предметно-специфічних евристик здійснювали за допомогою найпоширеніших методів, якими наразі є коефіцієнти ДКП і вейвлет-розкладання відповідних блоків. Ці перетворення дозволяють побудувати ознаки, які несуттєво спотворюють попиксельну міру подібності блоків.

Зменшення обчислювальної складності досягають шляхом зменшення попиксельних порівнянь за рахунок використання індексів та дескрипторів. Попиксельні порівняння здійснюють лише за оцінки відповідності вторинних доменів ранговому. Також на зменшення обчислювального навантаження має вплив модифікована формула розрахунку середньоквадратичного відхилення, оскільки значення сум і квадратів сум доменних і рангових блоків можна обчислити ще до початку перебору.

Отже, запропоновано модифікований алгоритм фрактального стиснення, ефективність якого підтверджена експериментально.

СПИСОК ЛІТЕРАТУРИ

1. Ватолин Д. С. Тенденции развития алгоритмов архивации графики / Д. С. Ватолин // Открытые системы. – 2010. – № 2. – С. 15 – 24.
2. Fisher Y. Fractal image compression using iterated transform / Y. Fisher, E.W. Jacobs, R. D. Boss // Naval Ocean Systems Center: NOSC Technical Report. – 1991. – № 1408. – P. 1122 – 1128.
3. Jacquin A. Image coding based on a fractal theory of iterated contractive image transformations / A. Jacquin // IEEE Transactions on Image Processing. – 1992. – No 1. – P. 18 – 30.
4. Ватолин Д. С. Использование ДКП для ускорения фрактального сжатия изображений / Д. С. Ватолин // Программирование. – 1999. – № 3. – С. 51 – 57.
5. Дискретное вейвлет-преобразование [Електронний ресурс]: – Режим доступу: https://ru.wikipedia.org/wiki/Дискретное_вейвлет-преобразование. – Назва з екрану.

Софина Ольга Юрївна – к. т. н., доцент кафедри автоматичної та інформаційно-вимірювальної техніки.

Лозун Алла Василівна – магістрант кафедри автоматичної та інформаційно-вимірювальної техніки. Вінницький національний технічний університет.