

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 файла, 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. Сергеев, 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  $90^\circ$ ,  $180^\circ$  і  $270^\circ$ , відображення блока відносно вертикальної осі симетрії і повороти отриманого відображеного блока також на  $90^\circ$ ,  $180^\circ$  і  $270^\circ$ . 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  $\tau_1$ , 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.

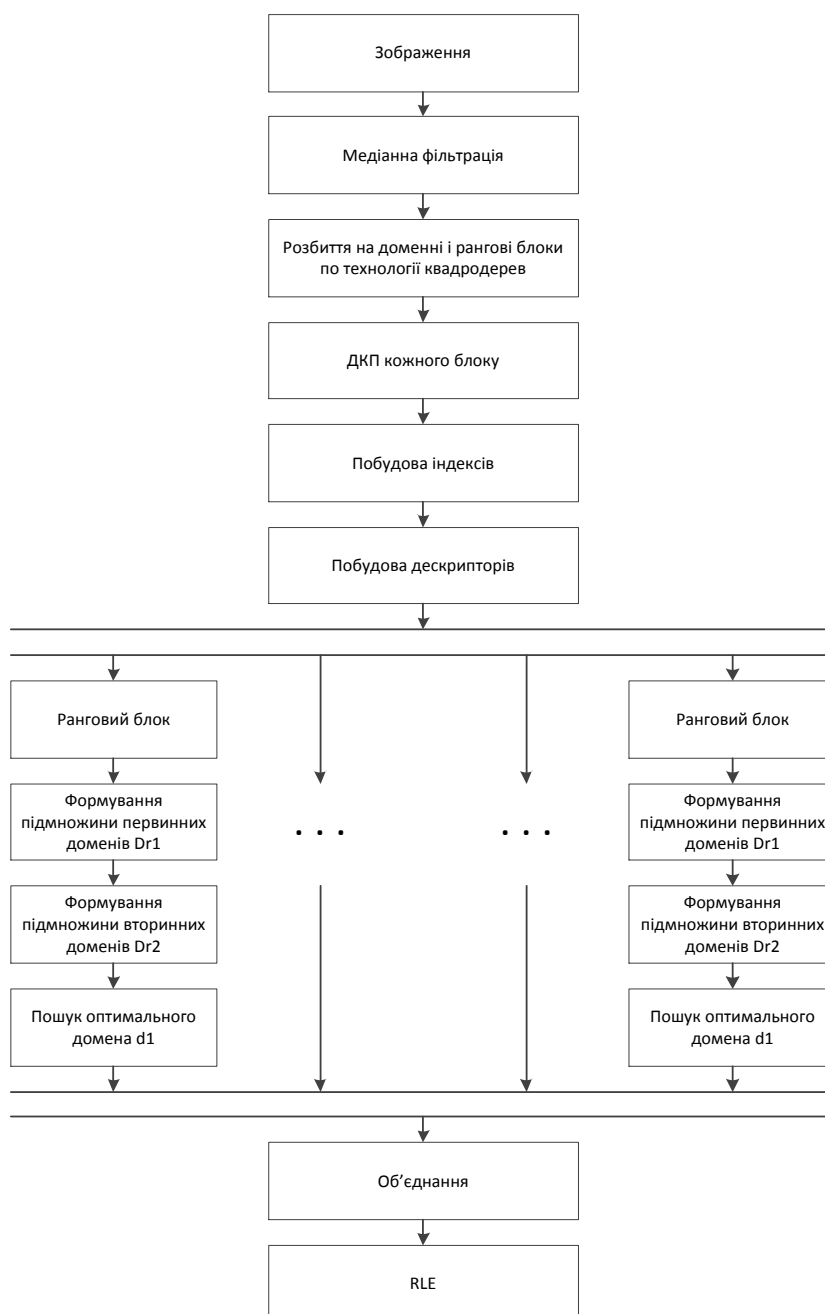


Рис. 2. Алгоритм пошуку оптимальних доменів

### 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):

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

Маємо таку умову:

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

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:

$$Q = s^2 \sum_{i=1}^N D_i^2 + No^2 + \sum_{i=1}^N R_i^2 - 2s \sum_{i=1}^N R_i D_i + 2so \sum_{i=1}^N D_i - 2o \sum_{i=1}^N R_i \quad (10)$$

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.

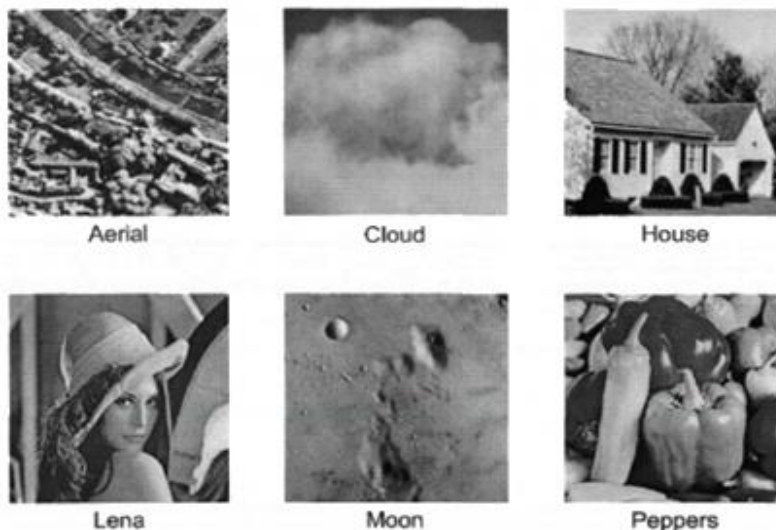


Рис. 3. Тестові зображення

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,  $\sigma$  – 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

$N$	$t$	$\sigma$	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

$N$	$t$	$\sigma$	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 вигравш, а на зображеннях розміром 240 пікселів – уже 70 – 75-кратний вигравш.

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.

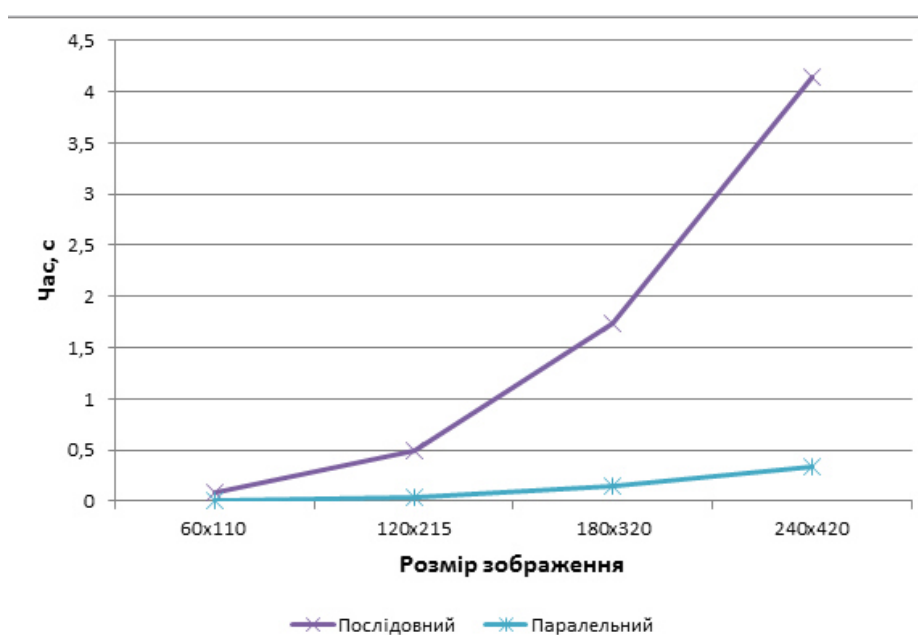


Рис. 4. Залежність часу виконання від розміру зображення

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](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.

**O. Sofina, Cand. Sc. (Eng), Assist. Prof.; A. Lozun**

## **DEVELOPMENT AND STUDY OF THE MODIFIED ALGORITHM OF FRACTAL CODING**

*The paper suggests modified algorithm of fractal coding using discrete cosine conversion, Haar wavelet – transform and parallel computations. The aim of the developed algorithm is to decrease computational complexity of fractal coding algorithm and increase the rate of image compression.*

**Key words:** *image compression, fractal coding, discrete cosine conversion, wavelet – transform, parallel computations.*

### **Introduction**

Nowadays as a result of wide spreading of electronic – computing equipment storage and processing of various types of data is realized in digital form.

Multimedia types of data – video – audiorecordings and digital images become more and more popular. One of the most actual problems of modern information technologies is the development of efficient methods of multimedia data compression, particularly – graphic information. Proceeding from the above- mentioned, one of the most urgent problems of modern information technologies is the development of efficient methods of multimedia data compression, particularly, graphic information.

Most frequency compression methods with losses are applied to multimedia information. Such approaches are valid as in case of multimedia objects it is possible to give up from saving their certain peculiarities (for instance, tiny elements on the image), that enables to increase the degree of compression. But there exist widely spread algorithms of the compression without losses, such as FLAC for audiofiles or PNG for digital images. Using these formats it is worth taking into account that they were designed as universal for certain type of data and, as a result they turned out to be useless for taking into consideration the peculiarities of the specific file to be compressed. As a consequence, considerable worsening of compression factor as compared with the analogous compression algorithms with losses, occurred.

For digital images, as the class of multimedia data, the most widely spread compression format with losses is JPEG. It became widely used as a result of digital cameras, scanner, etc. Taking into account the amount of images, presented in JPEG format, losses, connected with storage, transfer and processing of, probably, not optimally (by the quality and compression degree) compressed information, become evident.

Hence, studies, dealing with compression methods, become actual. They are based, on different image representation, namely, fractal coding, characteristic feature of this method is the feature of image self similarity. However, the existing methods of fractal compression of images require considerable development, taking into account numerous criteria (in particular, operation speed compression degree, decompression quality) to be considerable as real alternative to JPEG for numerous classes of images, being used in scientific, engineering and every day spheres of human activity.

A great deal of research was performed by native (D. C. Vatolin, V. V. Sergeev, V. V. Suifer, V. V. Aleksandrov, N. I. Gurskiy) and foreign (M. Barnsli, A. Jasquin, Y. Fisher, D. Zauper) scientists [1 – 3]. But, still there is much room for further research in this direction.

The aim of the paper is the construction of the modified algorithm of fractal coding, that will increase the speed of image compression process and achieve the acceptable compression factor, combining the already known DCT technologies, wavelet transform and parallel computations.

Main problems, to be solved for achieving this aim are development and study of optimal methods of image division into domain and ranking blocks, development of index and descriptor techniques of domain and ranking blocks matching and combining of the developed techniques for creation of the integral algorithm.

### Division of the image into domain and ranking blocks

Image is divided into ranking blocks of the same size  $r$  and into all possible domain flock of  $2r$  dimension. The given set of domains is called basic one. Further, additional set of domains is formed in the following manner: for each of “basic” domains seven transforms are built, namely: turns of the block at  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ , mapping of the block relatively the vertical axis of symmetry and turns of the obtained mapped block also at  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ . As a result of combination of the basic and additional sets of domains the expanded set of domain blocks is formed. For each ranking block the search among expanded set of domains is carried out. For this purpose each ranking and domain block is given index and descriptor.

This method of image division into domain and ranking blocks is optimal because each block may be definitely identified by its consecutive number (in case of preliminary set fixed method of numeration): for saving the number of block, fewer number of bits is needed, for instance, than for the coordinates of its upper angle. One more advantage is that there is no need to indicate the size of each block in the compressed file : it is sufficient to indicate only once the size of ranking block, and also the coefficient of proportionality for determination of domain block size.

### Calculation of indexes for domain and ranking blocks

In order to calculate index for each block  $b$ , matrix  $B$  of its two – dimensional discrete cosine transform (DCT) is calculated. Proceeding from the analysis of [4] the following algorithm of index calculation using the obtained DCT matrices is suggested:

- “Upper left angle” of matrix  $B$ , that corresponds to current block  $b$  (submatrix  $\hat{B} = \{B_{ij}\}_{i,j=1,3}$ ) is considered;
- In accordance with the order of elements selection for calculation of the index, given in Fig. 1, elements of these submatix from the 1<sup>st</sup> to  $t^{\text{th}}$  where  $t$  – selected dimensionality of the index  $t \in [1, 8]$  are evaluated. If current element is less than zero, then corresponding digit of the index in binary representation, starting from the left, value 0 is assigned, if greater or equal zero – value 1.

	1	5		
2	4	6		
3	7	8		

Fig. 1. Order of elements selection for index calculation

When turns or mapping change rule is applied to block  $b$ , DCT matrices  $B$  are set by such equations:

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

where  $i, j = \overline{1, N}$ .

It follows that DCT-matrices (and, correspondingly, indices) of “additional” domains may be found rather simply, on the base of pre-calculated DCT – matrices of “basic” domains.

### Calculation of the descriptors for domain and ranking blocks

For calculation of the descriptors two-dimensional Haar wavelet-transform is used [5]. The following algorithm of descriptor formation (for block  $b$ , of  $p \times p$  pixels dimension, where  $p=2q$ ,  $q \in \mathbb{N}$ ) is suggested:

- coefficient  $e_1$  is calculated;

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

- block is divided into four equals subblocks  $b_2, b_3, b_4$  and  $b_5$ .
- we calculated the coefficients in succession

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

that correspond to blocks  $b_2, b_3, b_4$  and  $b_5$ .

As the index calculation, description formation for additional domain may be simplified, namely: instead of calculation, using the above-mentioned algorithm  $e = [e_2 e_3 e_4 e_5]$ , simple rearrangement of descriptor's components of the basic domain may be performed in accordance with the specific transformation of the block.

### Generalizing algorithm of optimal domains search

1. The first ranking block  $r_1$  is considered.

2. Subset  $D_{r_1}$  of the expanded set  $D$  of the domain is formed by means of sampling of all domains with completely or partially coinciding index value, applying the method of hierarchy search (domains, that belong to  $D_{r_1}$ , are called those, possessing “primary similarity” with ranking block, being considered or simply “primary”).

3. Descriptor of  $r_1$  block is compared by turns, with the descriptors of each of domains, belonging to  $D_{r_1}$ .

4. Subset  $D_1$  of the set  $D_{r_1}$ , is formed by means of selection of certain number of domains, the descriptors of which possess maximum degree approximation to ranking block  $r_1$  in the sense of Euclidian distance (domains, that belong to  $D_1$  are called those, possessing “secondary similarity” with ranking block, being considered, or simply “secondary”).

5. Among the found “secondary” domains domain  $d_1$ , is being searched and brightness and contrast coefficients  $s_1$  and  $o_1$ , corresponding to the pair  $r_1 - d_1$  proceeding from MSD minimization conditions:  $d_1 = \operatorname{argmin}(\operatorname{CKB}(r_1, \tilde{d}_1))$ , where  $\tilde{d}_1 = s\tilde{d}_1 + oE$ ,  $\tilde{d}_1$  - block  $d_1$ , compressed to the size of block  $r_1$ , and  $E$  – single matrix of block  $r_1$  size. Domain block, for which the given MSD minimization condition will be applied, is called optimal or the best.

6. If the domain  $d_1$  found at the previous step belongs to “basic” set of domains, then its consecutive number  $v_1$ , is fixed, if this domain to the “additional” set – number of transformation  $\tau_1$ , is fixed, by means of which it was obtained from the “initial” domain and the number of the given “initial” domain  $\tilde{v}_1$ .

7. The second ranking block  $r_2$  is considered for this block steps 2 - 6 are repeated, then ranking

block  $r_3$  is considered and so on, until optimal domain  $d_N$  is found for the last ranking block  $r_{N_r}$ , where  $N_r$  – number of ranking blocks.

Fig. 2 shows the scheme of the suggested algorithm of optimal domains search, that takes into account paralleling of computations on the level of ranking blocks, as for each of them a certain amount of independent operations must be performed.

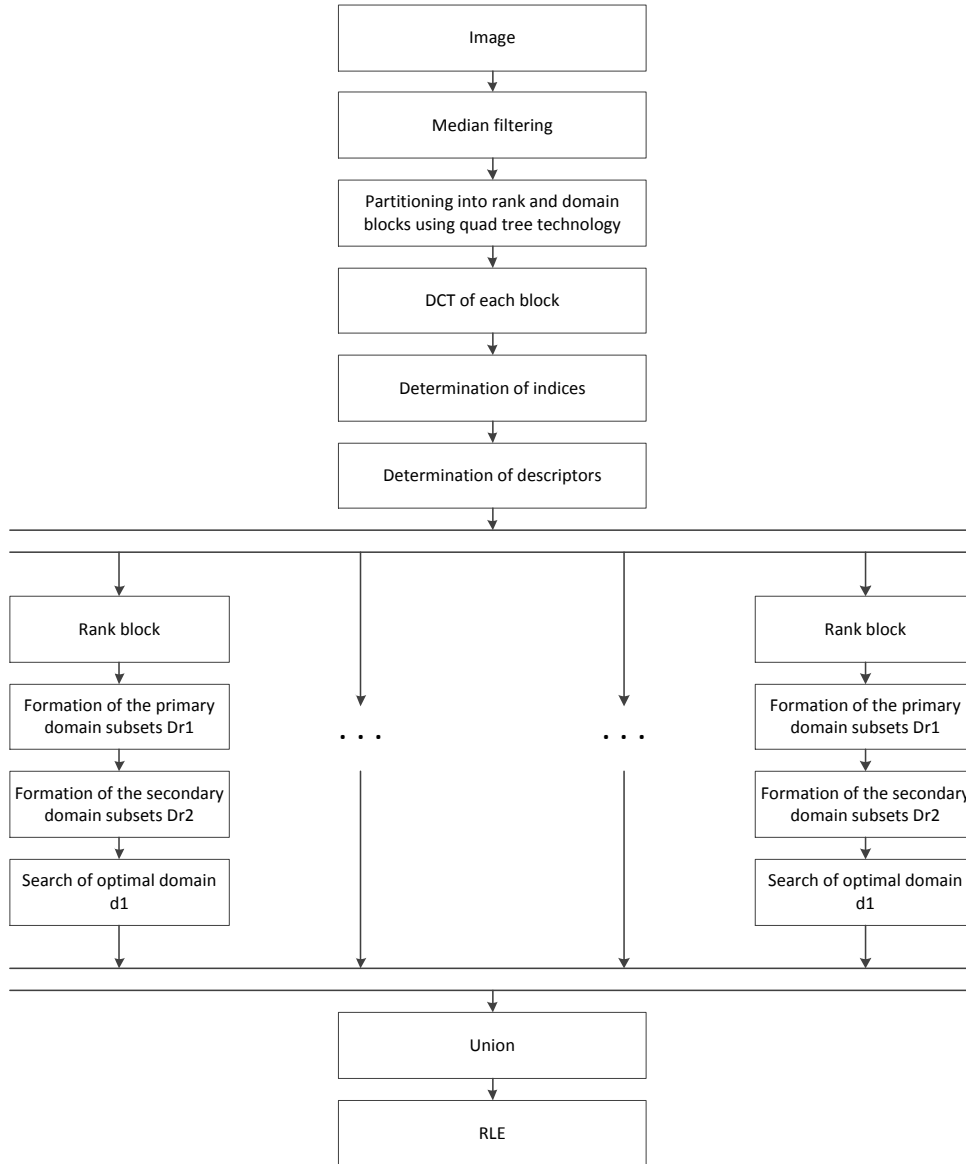


Fig. 2. Algorithm of optimal domains search

### Modification of standard deviation formula calculation

Modification of standard deviation formula calculation is carried out to decrease computation load on the system.

General formula of values transformation of domain block pixels has the following form:

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

where  $D_i^*$  and  $D_i$  transformed and initial  $i$ -<sup>th</sup> domain block, correspondingly;  $s$  – coefficient of contrast change;  $o$  – coefficient of brightness shift.

For evaluation of the discrepancy (distance) between the transformed domain and the given

ranking blocks it is necessary to introduce the corresponding metric. As a rule, function of standard deviation is used:

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

where  $R_i$  –  $i$ -<sup>th</sup> ranking block;  $D_i^*$  and  $D_i$  – transformed and initial  $i$ -<sup>th</sup> domain block, correspondingly;  $N$  – number of pixels in ranking block.

It is obvious, that the less is the distance between blocks, the more they are similar.

Coefficients  $s$  and  $o$  can be found from the formula (5), taking partial derivatives by these variables. Let us remove the brackets in the expression (5):

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

We have the following condition:

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

We express the shift by brightness:

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

Let us substitute (8) in the equation of partial derivative by  $s$  (7) and we obtain the following formulas to find the coefficients:

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

Having transformed the formula (3), we obtain the expression to find the distance:

$$Q = s^2 \sum_{i=1}^N D_i^2 + No^2 + \sum_{i=1}^N R_i^2 - 2s \sum_{i=1}^N R_i D_i + 2so \sum_{i=1}^N D_i - 2o \sum_{i=1}^N R_i \quad (10)$$

Formulas (6, 7) allow to simplify the computation loading, because the sums  $\sum_{i=1}^N R_i$ ,  $\sum_{i=1}^N R_i^2$ ,  $\sum_{i=1}^N D_i$ ,  $\sum_{i=1}^N D_i^2$  can be calculated still before the exhaustive search, when the sets of ranking and domain blocks are formed. Then, on the stage of comparison it is necessary to calculate only the sum  $\sum_{i=1}^N R_i D_i$  and find the coefficients.

### Experimental results of developed method testing

First we perform the study of classical algorithm of fractal compression. For this purpose the testing sample was formed, it comprises the following, different by content and structure, types of images, met most frequently:

- land images of the scenes of artificial objects;
- land images of natural objects;
- aerospace images;
- astronomic images;
- images of non-structured objects, such as clouds or smoke.

Fig. 3 contains the examples of these classes of images that further will be used for investigation.

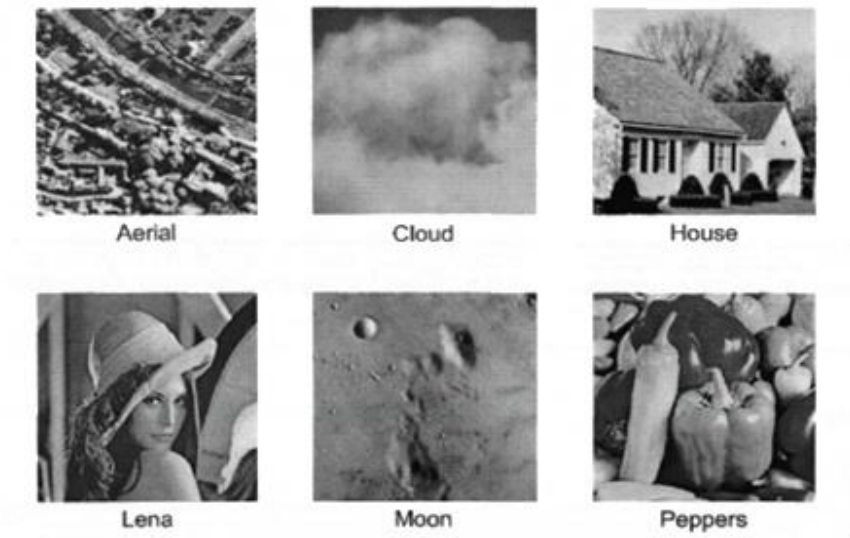


Fig. 3. Test images

Let us consider the indices of efficiency, obtained by means of classic algorithm of fractal compression. Table 1 contains the values of coder operation time, standard deviation of the restored image after decompression from initial image, signal/noise ratio and also the components of the criterion of minimum length of description (MLD) and its resulting value for images of various size - 60, 120, 180 and 240 pixels, compressed at the fixed size of ranking block, that is 6 pixels. We will define MLD criterion.

Principle of minimal length of description will be formulated in the following way: among the set of models it is necessary to select the model, that enables to minimize the sum of model description length (in bits) and the length of data, described, applying this model (in bits). The term “model” implies certain information structure, representing coded image. As the length of model description we will consider volume  $L_{img}$  of the compressed image in bits.

Length of data, described by means of applying the model implies that part of data, that didn't enter in the model itself. In other words, this is the volume of the losses  $L_{loss}$  as a result of compression. Hence, MLD criterion can be calculated by the formula:

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

We will consider designations, used in Table 1. Here  $N$  – size of the image,  $t$  – time of computation in seconds,  $\sigma$  – standard deviation of the restored image from the original, PSNR – signal – noise ratio,  $L_{loss}$  – evaluation of the volume of information losses during compression in bits,  $L_{img}$  – evaluation of the length of the compressed file in bits,  $L$  – sum of  $L_{loss}$  values and  $L_{img}$  – minimal length of compressed image description. Minimization of this length is realized at all possible correspondences of domain and ranking blocks at their fixed dimensions.

Table 1

#### Results of testing of classic algorithm for images of different size

$N$	$t$	$\sigma$	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

Let us consider how high is the compression quality, using classic fractal algorithm. Rather good quality of the restored image is considered to be PSNR~30, then Table 3 contains evaluations of compression coefficients in case of image restoration with such quality by means of division of output number of bits in the image by the value of  $L_{img}$ . The given value corresponds to the closest to 30 value of PSNR in Table 1. Besides, Table 3 also contains compression coefficients for JPEG format in case of regulations, corresponding to PSNR~30.

Table 2

**Compression coefficients at PSNR~30**

Image	Compression coefficient	
	Fractal algorithm	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

It is worth mentioning, that when standard programming facilities are used, JPEG compression coefficient may be for worse as a result of nonoptimal regulation recording of large volume of additional information in the file.

As it is seen from the Table compression coefficients differ greatly for various images. Now we will test the developed modified algorithm of the fractal coding and compare the results. As the main drawback of classic algorithm is rapid growth of operation time while increase of image size, its practical usage is inexpedient. The developed methods of domain and ranking blocks correspondences search in fractal coding were described in the given paper. We will perform experimental testing of the developed algorithm and determine losses of image quality. As the growth of operation rate does not influence the size of the compressed file, it will be sufficient to consider the value of standard deviation and PSNR for the restored image.

Table 3 contains characteristics of the quality of the restored image, obtained after compression by the modified algorithm and also its operation time on the images of various size.

Table 3

## The results of modified algorithm testing

$N$	$t$	$\sigma$	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

As it is seen from Table 3, operation rate is not only higher, as compared with classic algorithm (30 times on the images of 120 pixels of size) but also has other dependence on image size. As a result on the images of 60 pixels of size, we obtain 8 multiple gain, and on the images of 240 pixels of size – we obtain 70 – 75 multiple gain.

Comparing Tables 1 and 3 it may be noted that quality criterion of minimal length of description (MLD) as a result of application of optimized algorithm worsens approximately by 4 %. For part of images, constant worsening (for instance, from 1.5% to 3.5% for Aerial) is observed at the increase of image size, where as for other images the reduction of worsening may take place. Thus, gain in operation rate as compared with classic algorithm grows when image size increases, where as the loss in compression coefficient does not almost depend on the size or does not depend at all.

In the given work the application of parallel computation is suggested. It should be noted that paralleling of computations was already applied for optimized algorithm of fractal coding, that was described above. In Fig. 3, there is a graph, that represents the results of comparison of serial and parallel algorithm. As the paralleling of computations does not depend on the structure of the image and also does not influence the quality of the restored image, four images of different size were chosen for testing. Hence, we will consider the dependence of compression time on the size of the image. Operation of parallel algorithm in the given case is realized for 16-processor system.

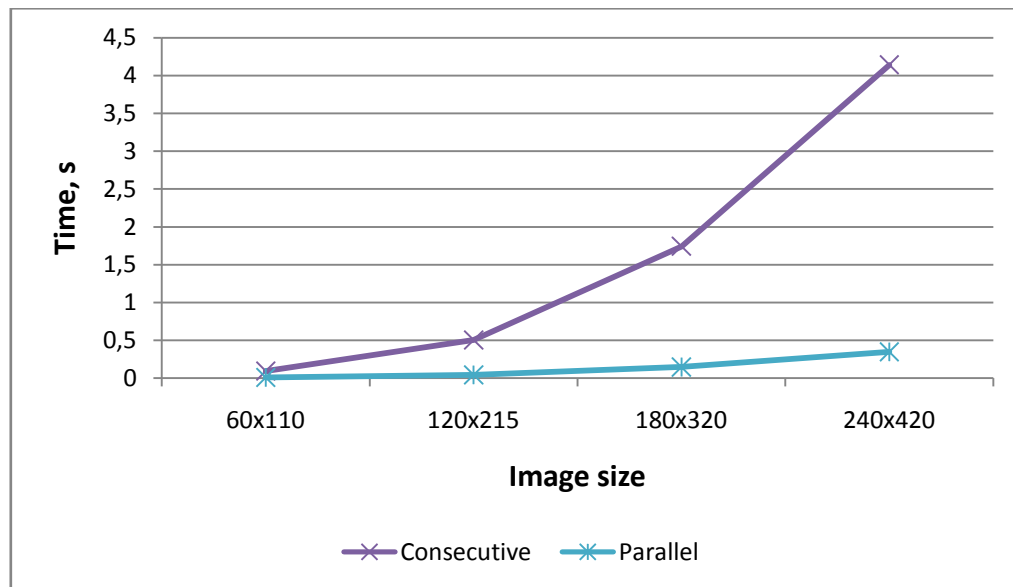


Fig. 4. Dependence of execution time on the size of the image

As it is seen, the rate of images processing applying parallel algorithm is far higher, than operation time of serial algorithm. It was found out that the usage of each additional processor will increase the operation rate by 60 – 75%.

### Conclusions

The developed algorithm of optimal domains search, that is modified algorithm of fractal coding, enables to increase image processing rate due to the usage of DCT, Haar wavelet transform and parallel computations and gives the possibility to decrease computation loading on the system.

As fractal coding refers to NP – complete problems, acceptable solution of such problems is achieved only when efficient subject-dependent heuristics are used.

Greater part of studies, dealing with fractal compression are dedicated to the development of similar types of heuristics.

For suboptimal realization of fractal compression it is necessary to limit exhaustive search of domain blocks, ranking blocks and corresponds between them and this was proposed to do in the given paper. Heuristics must cut off unpromising ways of image division into ranking blocks without performing further compression steps, and in case of fixed division of the image into ranking blocks – cut off unpromising subsets of domain blocks for certain ranking blocks.

In the given paper the formation of subjects – specific heuristics, was realized by means of the most widely used methods, namely DCT coefficients and wavelet –decomposition of the corresponding blocks. These transformations give the possibility to construct features, that insignificantly distort pixel- wise degree of blocks similarity.

Decrease of computational complexity is achieved by means of reducing pixel-wise comparisons, due to the usage of indices and descriptors. Pixel – wise comparisons are realized only in case of evaluation of correspondence of secondary domains to ranking one. Also, modified formula of standard deviation calculation influences the decrease computational load, because the value of sums and sums square of domain and ranking blocks may be calculated prior to the start of exhaustive search.

Thus, modified algorithm of fractal compression is suggested, its efficiency is confirmed experimentally.

### REFERENCES

1. Ватолин Д. С. Тенденции развития алгоритмов архивации графики / Д. С. Ватолин // Открытые системы. – Наукові праці ВНТУ, 2015, № 2

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 [Электронный ресурс]: – Режим доступа: [https://ru.wikipedia.org/wiki/Дискретное\\_вейвлет-преобразование](https://ru.wikipedia.org/wiki/Дискретное_вейвлет-преобразование). – Назва з екрану.

**Sofina Olga** – Cand. Sc. (Eng.), Assistant Professor with the Department of Automation and Instrumentation Technologies.

**Lozun Alla** – Master with the Department of Automation and Instrumentation Technologies.  
Vinnitsia National Technical University.