

УДК 681.3.05

МАЙДАНЮК В. П., ЛІЩУК О. О., КОРОЛЬ Д. С.

АСПЕКТИ ОПТИМІЗАЦІЇ ШВИДКОСТІ ФРАКТАЛЬНОГО УЩІЛЬНЕННЯ ЗОБРАЖЕНЬ

*Вінницький національний технічний університет,
21021, Хмельницьке шосе, 95, м. Вінниця, Україна*

Анотація. Проаналізовано фрактальний метод стиснення зображень. Проведено аналіз варіантів оптимізації та підвищення швидкодії фрактального ущільнення зображень. На основі досліджень розроблено швидкодійний алгоритм фрактального ущільнення та його програмна реалізація.

Ключові слова: фрактальне ущільнення зображень, системи ітеруючих функцій, класифікація блоків.

Анотация. Проанализирован фрактальный метод сжатия изображений. Проведен анализ вариантов оптимизации и повышения быстродействия фрактального сжатия изображений. На основе исследований разработан быстродействующий алгоритм фрактального сжатия и его программная реализация.

Ключевые слова: фрактальное уплотнения изображений, системы итерующих функций, классификация блоков.

Abstract. The fractal image compression method is analyzed. The analysis of variants of optimization and increase of speed of fractal coding of images is carried out. On the basis of researches, a high-performance fractal sealing algorithm and its software implementation are developed.

Keywords: fractal image densification, systems of deprivation functions, classification of blocks.

ВСТУП

Зображення, які представлені в цифровій формі, необхідно зберігати на носіях та передавати каналами зв'язку. Для економії пам'яті та більш ефективного використання ресурсів системи створюють спеціальні алгоритми кодування [1-4]. Зображення – це особливий вид даних, який має надлишковість в двох вимірах, що дає додаткові можливості для ущільнення [4]. Одним із перспективних методів ущільнення зображень є фрактальний метод [4]. Фрактальне кодування - це математичний процес для кодування растрів, які містять реальне зображення, в сукупність математичних даних, що описують фрактальні властивості зображення. Цей вид кодування заснований на тому, що усі природні і більшість штучних об'єктів містять надмірну інформацію у вигляді однакових блоків зображення, що повторюються. Вони отримали назву фракталів. Фрактал - це структура, яка складається з подібних форм і малюнків, що зустрічаються в різних розмірах.

АКТУАЛЬНІСТЬ

Алгоритм фрактального ущільнення відомий тим, що в деяких випадках дозволяє отримати дуже високі коефіцієнти ущільнення (найкращі приклади - до 1000 разів при прийнятній візуальній якості) для реальних фотографій природних об'єктів, що неможливо для інших алгоритмів ущільнення зображень з втратами.

Однак алгоритм не отримав масштабного розповсюдження через свою складність, яка полягає в тому, що для знаходження відповідних доменних блоків вимагається повний перебір. Оскільки при цьому переборі кожен раз повинні порівнюватися два масиви, дана операція виходить досить тривалою. Наприклад, для ущільнення зображення з розмірами 512x512 при глибині 8 біт на піксель потрібно близько 2 год при програмній реалізації фрактального ущільнення мовою C++. Зрозуміло, що при використанні таких популярних кросплатформених мов як C# або Java, які виконуються на віртуальній машині, це значення буде значно більшим.

Отже, задача підвищення швидкості ущільнення зображень фрактальним методом є актуальною дотепер, при покращенні показників швидкодії алгоритм фрактального ущільнення може стати одним з найбільш ефективних алгоритмів ущільнення зображень.

Метою роботи є підвищення швидкості процесу ущільнення зображень фрактальним методом без помітної втрати якості відновлених зображень.

Основні задачі:

Проаналізувати аспекти та особливості застосування фрактального методу ущільнення зображень.

Оцінити переваги та недоліки алгоритму фрактального ущільнення.

Запропонувати варіанти оптимізації та підвищення швидкодії ущільнення зображень з використанням фрактального методу.

АНАЛІЗ ФРАКТАЛЬНОГО МЕТОДУ УЩІЛЬНЕННЯ ЗОБРАЖЕНЬ

З фізичної точки зору фрактальне кодування ґрунтується на твердженні, що зображення містить афінну надлишковість. Математична модель, яка використовується при фрактальному стисненні зображень, називається системами ітеруєчих функцій (Iterated Function Systems – IFS). Системи ітеруєчих функцій містять набір стискальних перетворень w_i , які можливо задати так [4]:

$$W(S) = \bigcup_{i=1}^n w_i(S) \quad (1)$$

де S – зображення; w_i – набір стискальних перетворень.

Відповідно до теореми Банаха, існує певний клас відображень, які називаються стискальними і для них справедливо таке твердження: якщо до якогось зображення f_0 ми почнемо багаторазово застосовувати відображення W таким чином, що:

$$f_1 = W(f_0), f_i = W(f_{i-1}), \quad (2)$$

то при i , що прямує до нескінченності, ми отримаємо таке саме зображення незалежно від того, яке зображення ми взяли за f_0 :

$$f = \lim_{i \rightarrow \infty} f_i \quad (3)$$

Зображення f називається нерухомою точкою перетворення W або атрактором.

В якості перетворень w_i використовуються афінні відображення:

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & S_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ O_i \end{bmatrix} \quad (4)$$

де a_i, b_i, c_i, d_i – афінні коефіцієнти деформації, стиснення, обертання; dx, dy – коефіцієнти переміщення; x, y – координати точки, що перетворюється; z – її інтенсивність. Параметр S_i керує контрастністю, а O_i – яскравістю зображення. Знаючи коефіцієнти цих перетворень, ми можемо відновити початкове зображення.

Алгоритм фрактального кодування зображень можна описати так. Процес ущільнення починається з того, що зображення спочатку розділяється на блоки, що не перекриваються (рангові області), а потім на домені блоки, які можуть взаємно перекриватися, як це показано на рис. 1 [2].

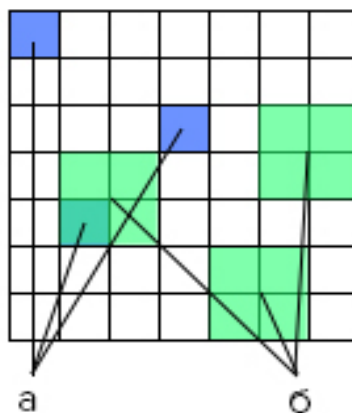


Рис. 1. Відбір блоків на зображенні: а – рангових, б – доменних

Домені повинні включати характерні фрагменти, які надалі використовуються для побудови декодованого зображення. Після цього починається кодування зображення шляхом підбору для кожної рангової області найбільш відповідного домена, за допомогою якого розподіл яскравості в ранговій

області може бути апроксимований розподілом яскравості в домені. Для того, щоб отримати найкращу апроксимацію, домені піддаються афінним перетворенням, в результаті яких відбувається не лише їх геометрична деформація, але і зміни контрасту та яскравості. Якщо розподілом яскравості в перетвореному домені не вдається досягти задовільної апроксимації розподілу яскравості в ранговій області, рангова область ділиться на чотири частини і процес повторюється. Якість необхідної апроксимації задається у вигляді допустимого значення середнього квадрата помилки апроксимації (середнього квадрата невідповідності). Номери доменів, використаних при кодуванні кожної рангової області, а також коефіцієнти афінних перетворень записуються у файл. Файл ущільненого зображення містить заголовок з інформацією про розташування рангових областей і доменів, а також таблицю ефективно упакованих афінних коефіцієнтів для кожної рангової області.

Одна з можливих схем кодування зображень фрактальним методом, запропонована Арно Жакеном (Arnaud Jacquin), містить такі етапи [4]:

– Зображення розділяється на області, що примикають одна до одної розміром $N \times N$ (рангові області).

– Задається набір доменних областей. Доменні області можуть перекриватись, вони не повинні обов'язково закривати всю поверхню зображення. Розміри доменних областей звичайно вибирають $2N \times 2N$.

– Для кожної рангової області підбирається доменна область, яка після афінних перетворень найбільш точно апроксимує рангову область. На практиці застосовується вісім варіантів відображення одного квадрата в інший з використанням афінних перетворень. Це повороти зображення на кути 0, 90, 180, 270 градусів відносно його центра і перетворення симетрії відносно ортогональних осей, які проходять через центр фрагменту перпендикулярно його сторонам.

– Точність апроксимації F визначається за допомогою середньоквадратичного критерію:

$$F = \sum_{i,j} (Sd_{ij} + O_{ij} - r_{ij})^2, \quad (5)$$

де d_{ij} – значення, отримані в результаті усереднення по фрагментах з розмірами 2×2 елементів доменної області, що приводить її розмір до розміру рангової області; r_{ij} – значення елементів рангової області. Зміщення O_{ij} може бути як константою, так і описуватись поліномами першого, другого, третього порядків.

Приврівнявши до нуля часткові похідні від виразу по S і O :

$$\frac{\partial F}{\partial S} = 0, \quad \frac{\partial F}{\partial O} = 0, \quad (6)$$

знайдемо значення S і O , при яких досягається мінімум виразу:

$$O = \frac{1}{n^2} \left(\sum_{i,j} r_{ij} - S \sum_{i,j} d_{ij} \right) \quad (7)$$

$$S = \frac{n^2 \sum_{i,j} r_{ij} d_{ij} - \sum_{i,j} r_{ij} \sum_{i,j} d_{ij}}{n^2 \sum_{i,j} d_{ij}^2 - \left(\sum_{i,j} d_{ij} \right)^2} \quad (8)$$

Доменні блоки звичайно вибирають з кроком $n/2$ при $n=4$. У вихідний файл записуються такі параметри:

координати доменної області з найменшим значенням F_{min} ;

значення для O і S , отримані згідно з формулами (7, 8);

номер афінного перетворення.

Алгоритм декодування полягає в тому, що беруться два екземпляри одного і того ж зображення A і B , розподіл яскравості в яких неважливий. На цих зображеннях виділяються області, межі яких співпадають з межами рангових областей і доменів, а потім, використовуючи відомі значення афінних коефіцієнтів, по доменах, виділених на зображенні B , знаходяться розподіли яскравості в рангових областях зображення A . Після цього зображення A і B міняються місцями і операція повторюється. Можна показати, що при багаторазовому повторенні цієї операції розподіл яскравості в зображеннях A і B наближаються до розподілу яскравості в початковому зображенні. Звернемо увагу на те, що алгоритми стиснення і декомпресії асиметричні. Слід також відмітити, що процес стиснення вимагає набагато більше часу, ніж процес декомпресії. Декодування стисненого зображення носить ітераційний характер і складається з таких етапів:

– Створюються два зображення однакового розміру А і Б. Розмір цих зображень не обов'язково рівний розміру початкового зображення, початковий малюнок областей А і В будь-який

– Зображення Б розбивається на рангові області так як на першій стадії процесу стиснення. Для кожної рангової області зображення Б виконується афінне перетворення відповідної доменної області зображення А і результат поміщається в Б

– Виконуються операції ідентичні попередньому пункту, тільки зображення А і Б міняються місцями.

– Багатократно повторюються другий і третій кроки до тих пір поки зображення А і Б не стануть нерозрізненими.

Основним недоліком фрактального методу є низька швидкість кодування, яка пов'язана з тим, що для отримання високої якості зображення для кожного рангового блока необхідно виконати перебір всіх доменних блоків, і для кожного доменного блока необхідно виконати не менше восьми афінних перетворень[5].

В цифровій обробці сигналів швидкодія методу оцінюється кількістю арифметичних операцій, необхідних для виконання перетворення. Оцінимо, наприклад, число операцій множення при кодуванні зображень фрактальним методом (вважаючи, що $S=1$), оскільки для їх виконання витрачається найбільше часу. Нехай $M \times N$ розміри зображення і $M=N=n^k$, де n - розмір сторін рангового блока. Нехай крок вибору доменних блоків $n/2$. Для порівняння даного рангового блока з доменним блоком необхідно виконати $8n^2$ операцій множення. Тоді загальна кількість операцій множення для пошуку відповідного доменного блока така[4]:

$$M = 8n^2 * (\text{кількість доменних блоків}),$$

Кількість доменних блоків при кроці вибору $n/2$ складе:

$$\left(\frac{n^k - 2n}{n/2} + 1\right) * \left(\frac{n^k - 2n}{n/2} + 1\right),$$

Після підстановки кінцева формула має такий вигляд:

$$M = 8(4n^{k+1}(n^{k-1} - 3) + 9n^2). \quad (9)$$

Отже, для підвищення швидкодії потрібно зменшити загальну кількість арифметичних операцій.

ОПТИМІЗАЦІЯ ФРАКТАЛЬНОГО АЛГОРИТМУ УЩІЛЬНЕННЯ ЗОБРАЖЕНЬ

Для підвищення швидкодії та ефективності фрактального кодування зображень використовують ряд методів оптимізації. Найпростіший і спосіб фрактального кодування є перевірка кожного доменного блока і виконання обчислень згідно виразів (5), (7), (8). Такий спосіб називається повним пошуком або повним перебором. При кодуванні зображень природного походження можна підвищити швидкодію кодування, прийнявши $S=1$, оскільки враховуючи статистику зображень завжди знайдеться доменний блок, який апроксимує заданий ранговий блок з необхідною точністю. Тоді з виразів (5), (7) одержимо [4]:

$$F = \sum_{i,j} (d_{ij} + O_{ij} - r_{ij})^2, \quad (10)$$

$$O = \frac{1}{n^2} \left(\sum_{i,j} r_{ij} - \sum_{i,j} d_{ij} \right). \quad (11)$$

Контрастність декодованого зображення може бути відновлена іншими методами. Таке спрощення дозволяє знизити кількість арифметичних операцій на 60 % і відповідно підвищити швидкість ущільнення. Найбільш відомі методи підвищення швидкодії кодування зображень фрактальним методом такі [4-6]:

1. Пошук доменних блоків, для яких F не перевищує заданого значення.
2. Локальний та сублокальний пошук.
3. Ізометричне передбачення.
4. Класифікація доменних і рангових блоків, ранговий порівнюється з доменними блоками того ж самого класу.

При програмній реалізації фрактального ущільнення зображень наряду з застосуванням спеціальних методів підвищення швидкості кодування значного ефекту можна досягнути за рахунок оптимізації роботи з масивами, оскільки швидкісні характеристики оперативної пам'яті значно уступають швидкості роботи процесора.

По-перше афінні перетворення можна виконувати лише з ранговими блоками. Тоді при черговому скануванні зображення афінні перетворення виконуються лише один раз перед початком

сканування із збереженням перетворень. З великою імовірністю можна стверджувати, що уже після обробки першого доменного блока всі перетворення рангового блока будуть переписані в кеш-пам'ять процесора, що значно підвищить швидкість звертання до них.

По-друге значним резервом для підвищення швидкості кодування є використання одновимірних масивів, оскільки при цьому виконуються операції лише з одним індексом. Перетворення двовимірних рангових і доменних блоків зображення в одновимірні не потребує значних додаткових обчислювальних затрат.

Ще одним важливим методом підвищення швидкості кодування є позбавлення від циклів у програмі при обчисленні середньо-квадратичного відхилення рангового і доменних блоків. При цьому у виразах прописуються всі квадрати різниць. Наприклад при розмірі рангового блока 8x8 необхідно в тексті програми прописати 8 виразів по 64 доданки кожний. Хоча це незручно, але виграш у швидкості кодування може становити 1,5 – 2 рази.

Підвищення швидкості кодування можна досягнути також за рахунок переривання обчислення середньоквадратичного відхилення чергового доменного блока від рангового за умови перевищення його поточного значення попереднього мінімального середньо – квадратичного відхилення доменного блока від рангового [7].

Класифікація доменних блоків передбачає попередній відбір доменних блоків для кожного рангового блока. Для попереднього відбору для кожного рангового та доменного блока можна обчислити коефіцієнти деякого двовимірного перетворення і за їх значенням виконати класифікацію цих блоків. Найбільш економічним є перетворення Уолша-Адамара, оскільки для його обчислення достатньо лише операцій додавання [2]. Для класифікації доменних і рангових блоків використовується лише декілька значень коефіцієнтів перетворення (рис. 2), що зменшує кількість обчислень та підвищує швидкість кодування.

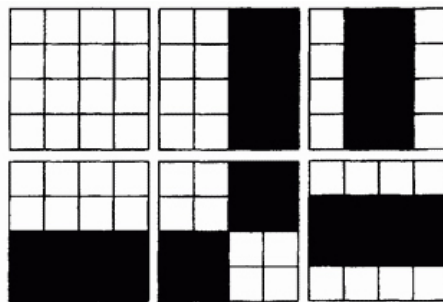


Рис.2. Базисні функції перетворення Уолша-Адамара

Перед відбором доменного блока для рангового вираховуємо середнє відхилення заданого рангового блока з кожним доменним, за формулою:

$$SQ = \sum_i (w_i - w_d)^2 \quad (12),$$

де w_i – результат перетворення Уолша-Адамара за i -тим базисним зображенням рангового блока, w_d – результат перетворення Уолша-Адамара за i -тим базисним зображенням доменного блока.

Відбирається задана кількість доменних блоків з найменшим відхиленням для поточного рангового блока та виконується стандартний алгоритм фрактального ущільнення з відібраними доменними блоками.

Для порівняння порахуємо необхідну кількість операцій для стиснення зображення в градаціях сірого розміром 512x512 пікселів при розмірі рангового блока 8 пікселів.

Для алгоритму запропонованим Жакемом:

Кількість ітерацій при проході циклу по всім ранговим блокам:

$$(512/8 * 512/8) = 4096$$

Кількість операцій при проході по всіх доменних блоках і його перетвореннях для 1 рангового:

$$(((512/16 * 512/16)*8)*8)*8 = 524288$$

Загальна кількість математичних операцій: 4096* 524288= 2147483648.

Для алгоритму з попереднім відбором, запропонованим у даній статті:

Кількість ітерацій при проході циклу по всім ранговим блокам:

$$(512/8 * 512/8) = 4096$$

Кількість операцій при попередньому відборі для 1 рангового блока (де k -кількість заданих перетворень, в цій статті $k=6$):

$$(512/16 * 512/16) * k = 1024 * 6 = 6144 * 4096 = 25165824$$

Кількість операцій для відбору найкращого, з попередньо відібраних, доменного блоку для рангового (де d -кількість попередньо відібраних доменних блоків для 1 рангового): $((d*8)*8) * 4096 = d*512*4096$.

Автори статті у наведених нижче прикладах приймали d як 10% від кількості усіх доменних блоків:

$$(512/16 * 512/16) * 0,1 * 512 * 4096 = 214748365.$$

$$\text{Загальна кількість операцій: } 25165824 + 214748365 = 239914189.$$

Отже при заданих умовах вдалося значно зменшити кількість арифметичних операцій.

Інший підхід передбачає класифікацію як доменних так і рангових блоків і фрактальні перетворення виконуються лише для одного рангового блоку із списку подібних. Кожен новий ранговий блок який створюється отримує ідентифікатор схожості (IC) який дорівнює «-1» що означає, що даний ранговий блок не схожий ні на один із вже отриманих.

Під час виконання розбиття зображення на рангові блоки необхідно кожен наступний ранговий блок порівняти з усіма попередніми які мають IC = «-1». Якщо різниця між кожним пікселем нового рангового блоку та попередніми ранговими блоками відсутня, тобто

$$r[i][j] - r_new[i][j] = 0, \quad (13)$$

то два рангових блоки є однаковими і в IC нового рангового блоку необхідно записати номер по порядку (ПП) рангового блоку з яким виконалось порівняння та було отримано $r[i][j] - r_new[i][j] = 0$. Наприклад, r має номер ПП = 0 і значення IC = -1, r_new має номер ПП=2, значення IC дорівнює номеру ПП блоку r , тобто IC = 0 (рис. 3).



Рис.3. Класифікація рангових блоків

Під час розбиття на доменні блоки необхідно порівняти кожен новий усереднений доменний блок з усіма унікальними доменами, які було отримано раніше. Порівняння виконується наступним чином. Необхідно знайти різницю між кожним пікселем з унікальних та новим за формулою

$$d[i][j]+O-d_new[i][j], \quad (14)$$

де O – коефіцієнт зміщення по яскравості між блоками. Якщо в результаті отримано значення яке по модулю не перевищує певного порогу то новий доменних блок не потрібно додавати у список унікальних, якщо ж хоча б один піксель перевищує встановлений поріг, то такий доменний блок необхідно вважати унікальним (значення різниці знаходиться в межах від -510 до +510, оскільки кожен піксель не перевищує значення 255 та зміщення по яскравості знаходиться в межах від -255 до +255).

Після розбиття зображення на доменні і рангові блоки виконується власне ущільнення фрактальним методом. Відмінністю є те що коли обирається ранговий блок, то перевіряється значення IC і якщо IC= -1, то виконується перебір унікальних доменів, щоб знайти найбільш схожий до даного рангового блоку. Якщо IC не дорівнює -1 то замість виконання перебору унікальних доменів, необхідно лише дізнатись який доменний блок було знайдено для рангового блоку R з номером ПП що дорівнює IC. Саме цей підхід і досліджується у подальшому.

МОДЕЛЮВАННЯ І РЕЗУЛЬТАТИ

Експериментально досліджувалася швидкість стиснення алгоритмом повного класичного перебору за схемою Жакена та швидкість стиснення швидкісним алгоритмом. Для моделювання ущільнення зображень фрактальним методом мовою програмування C++ з використанням середовища Qt Creator розроблено програмне забезпечення, яке дозволяє задавати розмір рангового блоку, крок вибору доменних блоків, поріг відхилення доменного і рангового блоків, а також формує допоміжну інформацію, зокрема, про розташування унікальних доменних блоків. Для роботи програми потрібна операційна система Windows 10 з встановленою платформою Microsoft.NET Framework 4.0. Головне вікно програми наведено на рис. 4.

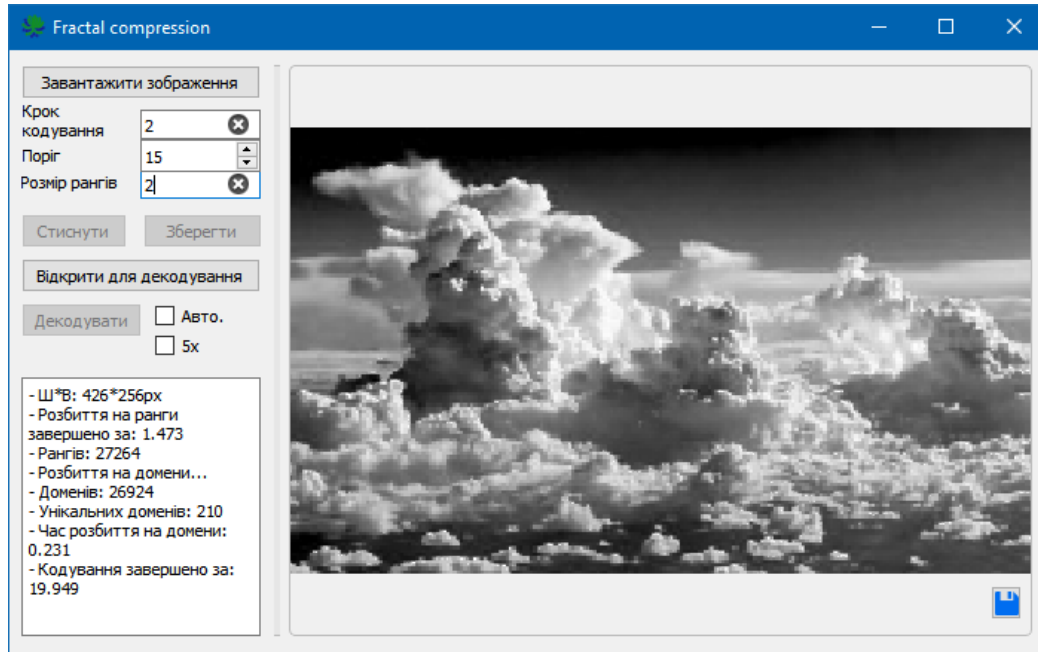


Рис.4. Головне вікно програми прискореного ущільнення, після виконання ущільнення



Рис. 5. Розташування унікальних доменних блоків

При проведенні експериментальних досліджень застосовуються наступні технічні засоби :

- Монітор - 15.6" (1366x768) LED
- Процесор - Intel® Core™ i7-2630QM

З рис. 4 можна зробити висновок, що серед 26924 доменних блоків було знайдено унікальних для даного зображення 210, що дало можливість виконати фрактальне ущільнення даного зображення за 19,949 сек. при прийнятній якості відновленого зображення. Розташування унікальних доменних блоків показує рис. 5. Для повного перебору всіх 26924 потрібно 2455,34 сек. Тобто досягнуто прискорення фрактального ущільнення в 123 рази.

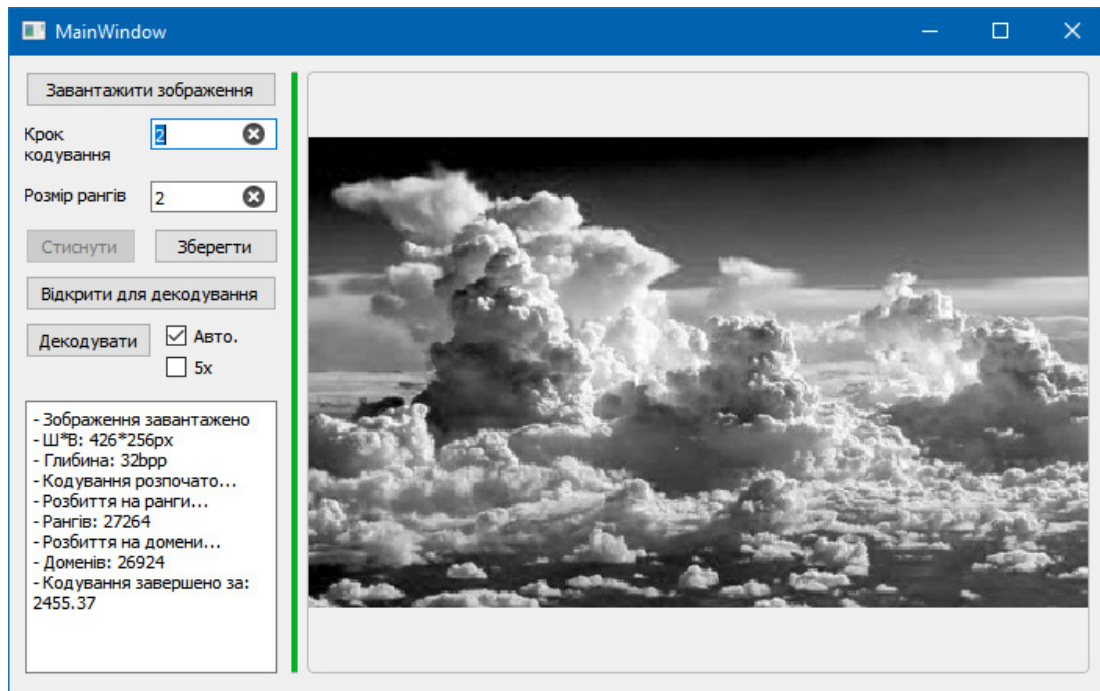


Рис.6. Головне вікно програми стиснення алгоритмом повного класичного перебору за схемою Жакена, після виконання ущільнення

Експерименти з різними типами зображень показали, що швидкість кодування зростає в 10-130 раз в залежності від складності зображення.

ВИСНОВКИ

Аналіз фрактального ущільнення зображень показав, що даний метод здатен забезпечити найкраще співвідношення ступеня ущільнення і якості відновленого зображення та має хороші перспективи для подальшого розвитку. Основним недоліком фрактального методу є низька швидкість ущільнення. Вона пов'язана з тим, що для отримання високої якості зображення, для кожного рангового блоку необхідно виконати перебір всіх доменних блоків. Для кожного доменного блоку необхідно виконати не менше восьми афінних перетворень. Ця проблема розв'язана тільки частково. Внаслідок відмічених недоліків цей метод застосовується на практиці порівняно рідко.

Запропоновано декілька оригінальних методів підвищення швидкості фрактального ущільнення. Зокрема, за рахунок попереднього відбору близьких доменних блоків за коефіцієнтами двовимірного ортогонального перетворення, що зменшує простір пошуку для кожного рангового блоку і відповідно підвищує швидкість фрактального ущільнення. Інший підхід передбачає класифікацію як доменних так і рангових блоків і фрактальні перетворення виконуються лише для одного рангового блоку із списку подібних. З використанням розробленого програмного забезпечення виконано натурне моделювання цього методу підвищення швидкості фрактального ущільнення зображень. Експерименти з різними типами зображень показали, що швидкість кодування зростає в 10-130 раз в залежності від складності зображення. На жаль, запропоновані методи не повністю вирішують проблему підвищення швидкості фрактального ущільнення зображень. Тому перед дослідниками відкриваються можливості покращення ефективності та пошуку інших способів оптимізації фрактального кодування.

СПИСОК ЛІТЕРАТУРИ

1. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. / Ватолин Д., Ратушняк А., Смирнов М., Юкин В. – М.: ДИАЛОГ-МИФИ, 2002. – 384 с.
2. Сэломон Д. Сжатие данных, изображений и звука. / Д. Сэломон – М.: Техносфера, 2004. – 368 с.
3. Кожем'яко В.П. Аналіз та перспективи розвитку кодування зображень / В.П. Кожем'яко, В.П. Майданюк, К.М. Жуков - Вісник ВПІ, 1999, № 3. – 42-48с.
4. Майданюк В. П. Методи і засоби комп'ютерних інформаційних технологій. Кодування зображень. Навчальний посібник // Вінниця: ВДТУ, 2001. – 63 с.
5. Кожем'яко В.П., Майданюк В.П., Жуков К.М., Хамді Р.Р., Піка С.О. Фрактальне

стиснення зображень природного походження // Міжнародний науково-технічний журнал "Вимірювальна та обчислювальна техніка в технологічних процесах", Хмельницький, 1999, № 2, с. 50-54

6. Kozhemiako V.P., Maidanuk V.P., Pika S., Zhukov K.M. Speeding up of fractal image compression // Proceeding of SPIE, 2001, Vol. 4425, p. 9-16.
7. Майданюк В. П. Оптимізація роботи з масивами при програмній реалізації фрактального ущільнення зображень // Тези доповідей Четвертої Міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія» м. Вінниця, 28-30 травня 2014 року. - <http://epsi.vntu.edu.ua/uploads/2014/21566f3cb607ef54fad70ae3695246eabd53cd46.pdf>

REFERENCES

1. Metody szhatyya dannykh. Ustroystvo arkhyvatorov, szhatye yzobrazhenyy u vydeo. / Vatulyn D., Ratushnyak A., Smyrnov M., Yukyn V. – M.: DYALOH-MYFY, 2002. – 384 s.
2. Sélomon D. Szhatye dannykh, yzobrazhenyy u zvuka. / D. Sélomon – M.: Tekhnosfera, 2004. – 368 s.
3. Kozhem'yako V.P. Analiz ta perspektyvy rozvytku koduvannya zobrazhen' / V.P. Kozhem'yako, V.P. Maydanyuk, K.M. Zhukov - Visnyk VPI, 1999, № 3. – 42-48с.
4. Maydanyuk V. P. Metody i zasoby komp'yuternykh informatsiynykh tekhnolohiy. Koduvannya zobrazhen'. Navchal'nyy posibnyk // Vinnytsya: VDTU, 2001. – 63 s.
5. Kozhem'yako V.P., Maydanyuk V.P., Zhukov K.M., Khamdi R.R., Pika S.O. Fraktal'ne stysnennya zobrazhen' pryrodnoho pokhodzhennya // Mizhnarodnyy naukovo-tekhnichnyy zhurnal "Vymiryuval'na ta obchyslyuval'na tekhnika v tekhnolohichnykh protsesakh", Khmel'nyts'kyy, 1999, № 2, s. 50-54.
6. Kozhemiako V.P., Maidanuk V.P., Pika S., Zhukov K.M. Speeding up of fractal image compression // Proceeding of SPIE, 2001, Vol. 4425, p. 9-16.
7. Maydanyuk V. P. Optymizatsiya roboty z masyvamy pry prohramniy realizatsiyi fraktal'noho ushchil'nennya zobrazhen' // Tezy dopovidey Chetvertoyi Mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Informatsiyi tekhnolohiyi ta komp'yuterna inzheneriya» m. Vinnytsya, 28-30 travnya 2014 roku. - <http://epsi.vntu.edu.ua/uploads/2014/21566f3cb607ef54fad70ae3695246eabd53cd46.pdf>

Надійшла до редакції 19.09.2017 р.

МАЙДАНИУК ВОЛОДИМИР ПАВЛОВИЧ – к.т.н., доцент кафедри програмного забезпечення Вінницького національного технічного університету, м. Вінниця, Україна.

ЛІЩУК ОЛЕКСАНДР ОЛЕКСАНДРОВИЧ - аспірант кафедри програмного забезпечення Вінницького національного технічного університету, м. Вінниця, Україна.

КОРОЛЬ ДЕНИС СЕРГІЙОВИЧ - магістр кафедри програмного забезпечення Вінницького національного технічного університету, м. Вінниця, Україна.