

ІНФОРМАТИКА, ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА АВТОМАТИЗАЦІЯ

УДК 004.6

DOI <https://doi.org/10.32838/2663-5941/2021.5/10>

Анищенко О.С.

Національний авіаційний університет

СИНТЕЗ ШВИДКОДІЮЧОГО АЛГОРИТМУ ФРАКТАЛЬНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ

У статті розкрито синтез швидкодіючого алгоритму фрактального стиснення зображень. Наголошено, що фрактальне стиснення можна розділити на такі два етапи: розбиття зображення на безліч рангових і доменних блоків (які можуть перекривати один одного) та застосування перетворень для кожної пари «доменний-ранговий блок» (геометричне, яке відображає доменний блок у рангу, та афінне, що змінює значення яскравості доменного блоку до максимальної відповідності значенням яскравості рангового блоку). Підкреслено, що від схеми розбиття, використаної на першому етапі, залежить якість стиснення. Чим більше доменних блоків, тим більшим є шанс знайти найбільш подібний ранговий блок. Визначено математичний складник алгоритму, описано принципи встановлення метрики відстані. Схематично представлено здійснення взаємодії доменів і діапазонів на прикладі реального зображення. Обґрунтовано, що крім безпосереднього перетворення значень пікселів, доменний блок також може бути підданий загальному масштабуванню (зменшенню розмірів до розмірів рангового блоку, наприклад, інтерполяцією або простим проріджуванням), повороту та іншим афінним перетворенням. Зазначено, що класифікація доменних і рангових блоків призначена для зменшення перебору блоків, і, як наслідок, прискорення алгоритму стиснення зображень. Кожен доменний блок класифікується до початку кодування. Під час підбору потенційний ранговий блок також класифікується і порівнюється тільки з доменами відповідного класу (або декількох близьких класів). Наголошено, що математично блок пікселів становить єдине ціле, а саме вектор позиції в абстрактному просторі позицій, де кожна окрема точка представляє інший блок. Коли до цього простору застосовується метрика відстані, взаємне розташування двох векторів визначає їхню близькість. Швидкодіючий алгоритм фрактального стиснення зображення використовує r -дерево, втілення якого запропоновано на зображенні підданому стисненню.

Ключові слова: швидкодіючий алгоритм, синтез, фрактал, стиснення, зображення, оптимізація, компресія, архівація.

Постановка проблеми. Незважаючи на незлічену кількість альтернатив, найпопулярнішими методами стиснення зображення є методи на основі блоків, де зображення перед обробкою розбивається на сітку із блоків, що не перекриваються. Хоча кожен блок кодується окремо, пікселі всередині блоку обробляються разом. Отже, перевагою є подібність (за кольором та інтенсивністю) між сусідніми пікселями. Існують такі дві основні категорії методів на основі блоків: перша перетворює доменні кодери (такі, як JPEG, яка спочатку переводить пікселі у блоку в альтернативну форму, у котрій нерелевантну інформацію можна легко відкинути, наприклад, за допомогою дискретної косинусної трансформації); друга – кодери просторових доменів (такі, як векторне кванту-

вання), що аналізують пікселі зразу в пошуках ефективного представлення. Обидві категорії інтенсивно досліджувались у 1980-х роках і, як наслідок, нині знаходяться у різних комерційних додатках. Незважаючи на цей успіх, методи трансформації стиснення зображення домену і просторового домену розвиваються в нових напрямках. Зокрема, значну увагу привернули дві нові математичні сутності: вейвлети та фрактали [1, 2]. Хвилі нагадують віконні синусоїдали (в одній загальній формі), а фрактали можна використовувати способом, подібним до векторного квантування. На додаток до низьких бітрейтів [3] дві властивості стиснених фракталами зображень роблять їх використання у відео-додатках актуальним. По-перше, це швидка декомпресія навіть

на низькошвидкісних ПК. По-друге – це можливість розгорнути вікно відео або збільшити його частину без непривабливих артефактів «сходів». На жаль, навіть за допомогою спеціальної апаратної допомоги етап стиснення може бути надто повільним. Кілька існуючих алгоритмів є продуктивними в режимі реального часу.

Аналіз останніх досліджень і публікацій. Початок розробки алгоритму фрактального стиснення зображення покладено можливістю застосування теорії систем ітераційних функцій у сфері стиснення зображень, дослідженої Майклом Барнслі та Аланом Слоуном [4]. Наступними кроками було вдосконалення та розвиток сфер застосування цього напрямку. Зокрема, Р. А. Зубко [5] провів огляд і стислий аналіз методів оптимізації фрактального кодування зображень. Автор запропонував варіанти суттєвого покращення часової ефективності стиснення фрактальним алгоритмом, навіть характеристики найперспективніших варіантів для практичного використання. Аналіз методів стиснення зображень на основі дискретно-косинусного перетворення запропонували Л. А. Клименко та А. Ю. Гордієнко [6]. У статті проводиться аналіз методів стиснення динамічних зображень, які набули найбільшого практичного застосування, – JPEG, MPEG, ДКП; розглядається дискретно-косинусне перетворення як найефективніший метод стиснення надмірної інформації.

Фрактальні методи стиснення зображень описали А. І. Вахіль та І. М. Кузьменко [7]. Ю.А. Щербакова та М.В. Скіцка [8] розкрили ефективність стиснення зображень залежно від попередньої обробки. У [9] авторами доведено, що здійснення фрактального стискання дозволяє застосовувати фрактальну компресію та декомпресію до графічних зображень, що розширює межі застосування фрактальної графіки не тільки до графічних об'єктів, але і до геометрії живої та неживої природи. Із зарубіжних авторів варто відзначити таких дослідників, як Galabov Miroslav [10], Menassel Rafik [11, 12], Menassel Rafik & Nini Brahim & Tahar Mekhaznia [13, 14], Khedgaonkar, Roshni [15], Menassel Rafik & Gaba Idriss & Titi Khalil [16], Garg Preedhi & Gupta Richa & Tyagi Rajesh [17], Agrawal Anshu & Chauhan Pushpraj [18], Amin Sobia & Gupta Richa & Mehrotra Deepti [19], Diep Do & Novinsky Nikolay [20] та інших. Проте, враховуючи описані наукові здобутки за цією темою, питання синтезу швидкодіючого алгоритму фрактального стиснення зображень залишається відкритим і потребує детального опрацювання.

Постановка завдання. Метою роботи є розкриття синтезу швидкодіючого алгоритму фрактального стиснення зображень.

Викладення основного матеріалу дослідження. Досліджуючи процес стиснення фрактального зображення, варто наголосити на фракталах, що лежать в основі цього процесу. Першочерговим фактором є приналежність фракталів. Вони належать не до комплексної площини (множини Мандельброта, множини Джулії), а до систем із ітераційними функціями (W – це сукупність афінних перетворень) із додатковою властивістю бути контрактними.

$$W = \bigcup_{i=1}^n w_i, w_i : R^3 \rightarrow R^3 \quad (1)$$

де w_i – афінні перетворення.

Під час перетворення w_i будь-які дві точки площини зображення наближаються одна до одної як просторово, так і у відтінках сірого або у кольорі.

$$d_{скв.ном}(w_i(u), w_i(v)) < s_{i,v} d_{скв.ном}(u, v), 0 \leq s_{i,v} < 1 \quad (2)$$

$$d_{евк.від}(w_i(u), w_i(v)) < s_{i,v} d_{евк.від}(u, v), 0 \leq s_{i,v} < 1 \quad (3)$$

де u, v – будь-які дві точки зображення.

Метрика відстані d змінюється залежно від теорії чи практики використання. Для перших метрикою вибору є відстань Гауссдорфа (на практиці $d_{евк.від}$ – це евклідова відстань, а $d_{скв.ном}$ – середньоквадратична помилка). Оскільки теоретик прагне спростити математичні докази, практик прагне досягти обчислювальної ефективності. Незалежно від цього, якщо сукупність афінних перетворень загалом є скорочувальною, то, застосовуючи до будь-якого вхідного зображення μ , система ітераційних функцій має таку властивість:

$$\mu_\infty = \lim_{n \rightarrow \infty} (W^n(\mu)) \text{ є унікальним атрактором, і} \quad (4)$$

$$\mu_\infty = W^n(\mu) \text{ – це зображення з нерухомою точкою} \quad (5)$$

Тобто початок стиснення зображення варто орієнтувати на неодноразові скорочувальні перетворення W після певної точки, причому будь-який процес стабілізується до унікального зображення атрактора μ_∞ незалежно від ∞ . Оскільки ми отримаємо подібне зображення, засноване на нескінченно малих об'єктах, це зображення вважатиметься фрактальним. Здатність системи ітераційних функцій створювати велику кількість фрактальних зображень, таких як Трикутник Серпінського, описано в [21]. Більш складним є зворотне перетворення (відновлення): з огляду на довільне цифрове зображення потрібно спробувати знайти систему ітераційних функцій, яка, оцінюючись ітерацією, дає інше схоже зображення за зовнішнім виглядом до оригіналу. Іншими словами, варто знайти такий набір перетворень $\{w_i\}$, що

$$d_{скв.ном}(\mu_{поч}, \mu_\infty) \quad (6)$$

Ця проблема відома як «обернена задача», яка у загальному вигляді залишається невирішеною. Проте у 1992 році Арно Жаккін представив обмежену версію оберненої задачі, на якій базується все сучасне програмне забезпечення для стиснення фрактальних зображень.

У найпростішій формі зображення, що підлягає стисненню, розбивають на два масштаби, один із них удвічі більший за інший (наприклад, на блоки розміром 8x8 та 4x4). Перший масштаб зображення виконується доменними блоками, менший масштаб – діапазонами. Блоки діапазонів не перекриваються, кожен із них містить піксель зображення. Блоки домену можуть перекриватись і кожен не повинен містити піксель, як показано на рис. 1.

Ліворуч зображено блоки домену, які відображаються на блоках діапазону, зображеного праворуч. У наведеному прикладі на рис. 1 міститься 25 перетворень, пронумерованих у порядку сканування.

Метою процесу стиснення є пошук афінного відображення w_i форми

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (7)$$

для кожного блоку діапазону, оскільки під час застосування до якогось (поки що не визначеного) блоку домену значення $d_{скв.лом}$ є мінімальним. У рівнянні (7) точка (x, y) є пікселем домену зі значенням градацій сірого z , а (x', y') – пікселем діапазону з новим значенням z' . Коефіцієнти e_i і f_i переводять доменний блок у положення блоку

діапазонів, оскільки його розмір зменшується удвічі. Коефіцієнт s_i масштабує значення яскравості (подібно до контрастності на телевізорі), тоді як коефіцієнт o_i введено для зміщення яскравості (подібно до регулятора яскравості). Отриманий фрактальний код є послідовністю кортежів (e_i, f_i, o_i, s_i) , по одному для кожного блоку діапазону. Із цими обмеженнями таку схему можна охарактеризувати як половину геометричної яскравості першого порядку, блоку локального домену, методу блокування розділеного діапазону. Зауважимо, що $s_{i,g}$ (3) зафіксовано на 1/2, тому для зручності " v " у $s_{i,v}$ з (2) опускається.

Суть процесу стиснення полягає у сполученні кожного блоку діапазону із блоком домену таким чином, щоб різниця між ними під час афінного перетворення, визначеного рівнянням (7), була мінімальною.

Навіть якщо обмежуватися квадратними блоками, обчислення є величезними. Зображення 512x512 містить загалом 255 025 різних блоків домену 8x8, зокрема вісім ізометричних симетрій (чотири обертання, чотири відбиття) збільшують цю загальну суму до 2040200. Існує $(512/4) 2 = 16,384$ блоків діапазону, що робить загалом 33 426 636 800 можливих пар для тестування. Задля зменшення навантаження можна спробувати здійснити легкий пошук, наприклад, шляхом обмеження не перекривання блоків домену; у цьому випадку є всього 536 870 912 варіантів. Фактично пул доменів скоротився у 64 рази. Пошук все ще триває $O(n^2)$, але принаймні обчислення скоротилось із кількох днів до кількох годин. Недоліком є

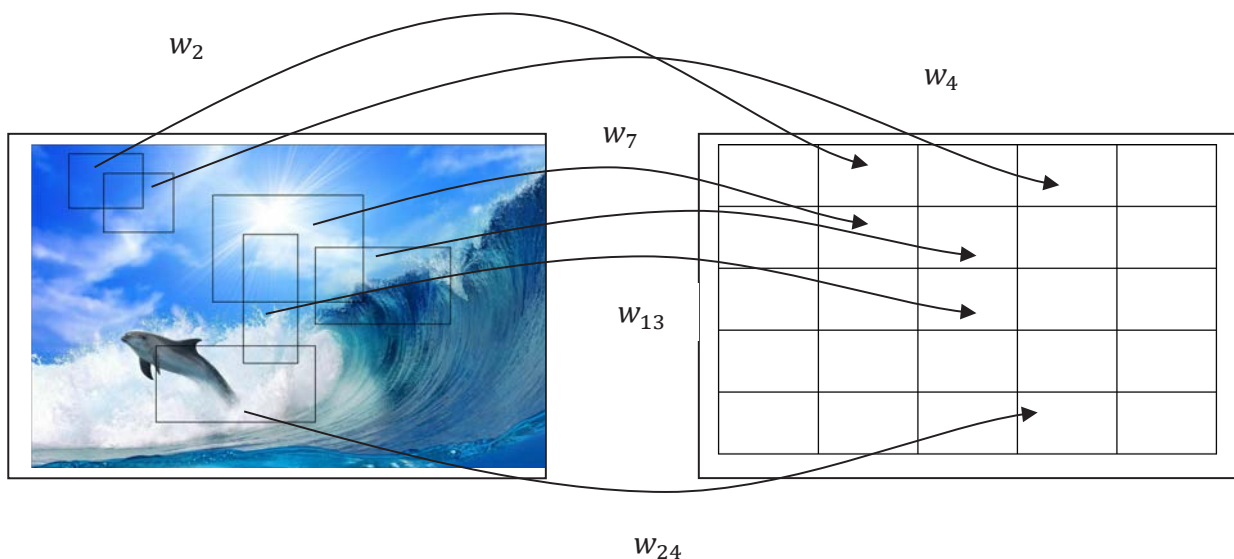


Рис. 1. Схема здійснення взаємодії доменів і діапазонів

те, що страждає якість зображення, оскільки найкращі пари можуть пропускатися.

Очевидним способом досягнення подальшого збільшення швидкості є обмеження пошуку невеликою областю відносно поточного блоку діапазонів. Якщо, скажімо, розглядаються лише 64 найближчі доменні блоки, тоді кількість можливостей зменшується до 1048 576. У крайньому випадку від пошуку можна взагалі відмовитися, варто просто з'єднати блоки діапазону із найближчим доменним блоком, навіть не враховуючи вісім симетрій. Такий підхід передбачає нижню межу обчислень. Часова складність становить $O(n)$ із малим постійним коефіцієнтом. Але знову ж таки якість страждає, цього разу різко, оскільки вона спирається на зображення, що мають основу подібної форми.

Більш складні алгоритми використовують категоризацію. Кожен блок пулу домену класифікується як гладкий, текстурований або такий, що містить край. Кожен блок діапазонів також класифікується. Тоді під час пошуку відповідної пари шукають лише відповідну категорію. Виведення формули обчислювальної складності є складним, оскільки розподіл блоків домену на різні категорії залежить від вмісту зображення. Тим не менш такий підхід може скоротити час пошуку із кількох годин до декількох хвилин. Це сучасний рівень техніки фрактального кодування зображень.

Математично блок пікселів складає єдине ціле, зокрема вектор позиції в абстрактному просторі позицій, де кожна окрема точка представляє інший блок. Коли до цього простору застосовується метрика відстані, взаємне розташування двох векторів визначає їхню близькість. Для практичного здійснення потрібна багатовимірна структура даних, здатна зберігати та індексувати вектори позицій.

Нині наявні багато можливих способів перетворення блоку пікселів у вектор позиції: можна обчислити коефіцієнти дискретного косинусного перетворення, або коефіцієнти Карунена-Лоєва, або коефіцієнти текстури, або розширення Тейлора, або моменти інтенсивності, або інший набір функцій. Швидкодійний алгоритм фрактального стиснення зображень використовує найпростіший підхід: він зразу використовує значення пікселів. Отже, блок діапазону 4×4 стає 16-розмірним вектором положення. Однак слід враховувати такі два ускладнення:

1) блоки домену 8×8 мають бути зменшені (шляхом усереднення комірок 2×2 пікселів) для порівняння блоків діапазонів;

2) оскільки фрактальні коди складаються з афінних перетворень, шукаємо доменний блок, який найбільше відповідає цьому блоку діапазонів за афінного перетворення. Отже, пікселі у блоці слід спочатку нормалізувати до фіксованого середнього значення і дисперсії. Тобто, враховуючи 16 розмірний кортеж значень інтенсивності:

$\bar{z} = (z_1, z_2, z_3, \dots, z_{16})$ взято з піксельного блоку

$$\begin{bmatrix} z_1 & z_2 & z_3 & z_4 \\ z_5 & z_6 & z_7 & z_8 \\ z_9 & z_{10} & z_{11} & z_{12} \\ z_{13} & z_{14} & z_{15} & z_{16} \end{bmatrix}_{i,j} \quad (8)$$

вони перетворюються так, що

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z'_i = c_1$$

$$var = \frac{1}{n} \sum_{i=1}^n (z'_i - \bar{z})^2 = c_2 \quad (9)$$

де $n = 16$ і z'_i – нормовані значення пікселів; c_1 , і c_2 – константи.

Важливість цього кроку можна побачити, переглянувши рівняння (7). Оскільки афінне перетворення змінює середнє значення (через зміщення o_j) і дисперсію піксельного блоку (через шкалу s_i), дійсне порівняння вимагає нормалізації усього блоку після того, як вектори поблизу у просторі позицій представляють піксельні блоки, подібні відповідно до рівняння (3) до деякого афінного перетворення.

Далі алгоритм вимагає двох процедур: першу – для вставлення векторів положення, відповідних блокам домену, другу – для вилучення елемента, найближчого до цього блоку діапазону.

Швидкодійний алгоритм фрактального стиснення зображення використовує r -дерево. R -дерево (і його близький родич – r^* -дерево) – це структура даних, здатна ефективно індексувати багатовимірний простір. R -дерево не дуже відоме, але його можна вважати продовженням більш знайомого B -дерева (рис. 2).

Повні описи наведені у [2, 6]. На рис. 3 показано, що алгоритм працює у двох вимірах: « g » у r -дереві означає «прямокутник».

Верхній рівень $R1$ містить три прямокутники другого рівня, кожен із яких може містити до п'яти точок даних, x позначає місце пошуку. У прикладі, наведеному на рис. 3, показано одинадцять векторів положення, позначених від a до k , які є доменними блоками. Дерево r індексує ці вектори як вкладену множину прямокутників. X позначає розташування певного блоку діапазонів у цьому просторі позицій. За знаходження найближчого доменного блоку алгоритм пошуку звужується від

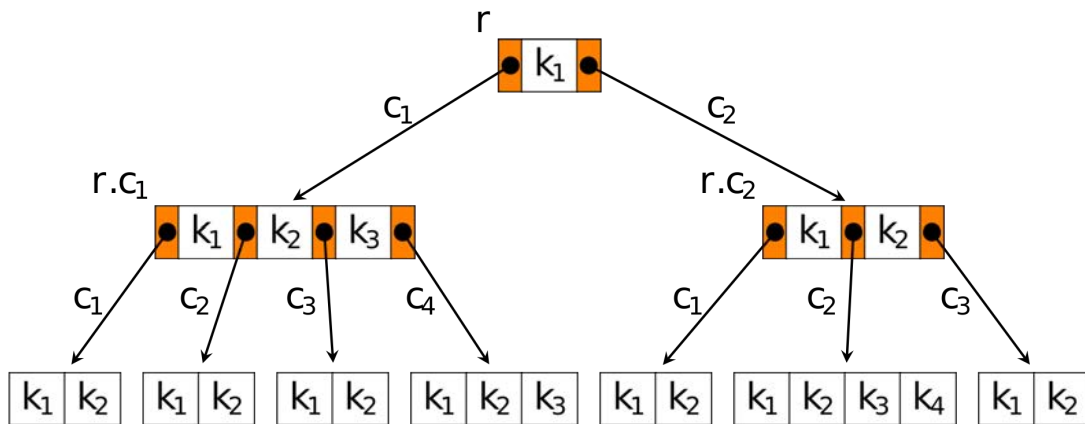


Рис. 2. Б-дерево

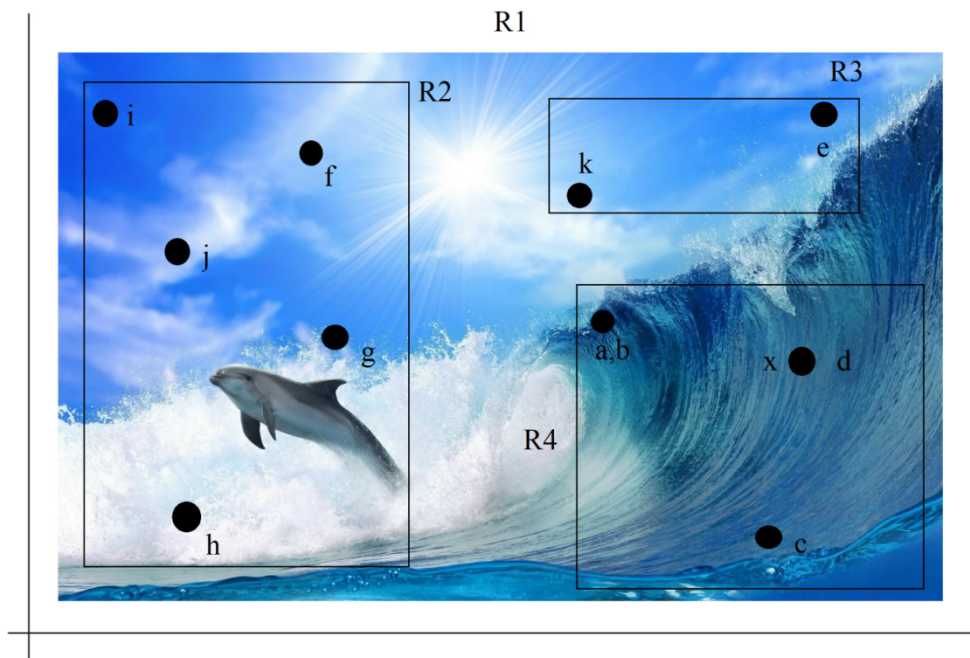


Рис. 3. Втілення алгоритму у двох вимірах

R1 до R4, перевіряє кожен елемент у R4 і повертає d як найближчу відповідність. На відміну від інших алгоритмів, які досліджують лише блоки домену, фізично близькі до певного блоку діапазонів, швидкодіючий алгоритм фрактального стиснення зображення досліджує структурно близькі блоки. Використовувана метрика відстані – це вибір реалізації. Використання багатовимірного індексу на просторі позицій дає дві переваги. Перша – це швидкість. Ураховуючи вкладений розподіл векторів позицій для визначення найближчого збігу, потрібно перевірити лише кілька елементів. Середній час пошуку становить приблизно $\propto n \log_m(n)$, де n – кількість вставлених

блоків домену, а m обмежує кількість елементів, що містяться в одному прямокутнику. Коли $m=1$, або коли m дуже велике, пошук зводиться до лінійного часу. Отже, слід бути обережним у виборі коефіцієнта розгалуження r -дерева. З експериментів значення 16 було задовільним для цієї програми. Наявність єдиного простору пошуку – друга перевага. Помилкова класифікація усувається, оскільки між категоріями немає жорстких кордонів, що призводить до меншої кількості бідних пар, хоча приріст швидкості обов'язково враховується з погляду на якість зображення.

Висновки і перспективи подальших досліджень. У роботі розкрито синтез швидкодіючого

алгоритму фрактального стиснення зображень. Стиснення фрактального зображення є спадком векторного квантування. Алгоритм пропонує конкурентний підхід до кодування нерухомих зображень, особливо за високих коефіцієнтів стиснення. Упровадження швидкодіючого алгоритму фрактального стиснення зображень дозво-

ляє встановити необхідну симетрію, що є можливим під час застосування на базі відео потоку реального часу. Перспективи подальших досліджень ґрунтуються на вдосконаленні розглянутого швидкодіючого алгоритму фрактального стиснення зображень за рахунок розширення спектру дії і зниження часу реалізації.

Список літератури:

1. Подчашинський Ю. О., Хаустович О. В. Дослідження методів фрактального стиснення відеозображень з вимірювальною інформацією, що передаються комп'ютерними мережами. *Вісник ЖДТУ. Серія «Технічні науки»*. 2018. № 1 (81). С.149–154. [https://doi.org/10.26642/tn-2018-1\(81\)-149-154](https://doi.org/10.26642/tn-2018-1(81)-149-154).
2. Подчашинський Ю. О., Чепюк Л. О. Порівняння фрактального і вейвлетного підходів до стиснення зображень. Тези доповідей XI Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2020 (ІКТ-2020)», (м. Житомир, 09 - 11 квітня 2020 р.). Житомир: Житомирська політехніка, 2020. С. 129-130.
3. Лукьянчиков С. Дослідження сучасних методів стиснення звукової інформації з втратами та без втрат. *Геометричне моделювання та інформаційні технології*. 2017. № 2. С. 62-67.
4. Зубко А.М. Інформаційно-комунікаційні технології як чинник розвитку професіоналізму педагогів у системі післядипломної освіти. *Інформаційні технології і засоби навчання*. 2020. Т. 77, № 3. С. 262-281.
5. Зубко Р. А. Дослідження можливості покращення часової ефективності фрактального стиснення зображення. *Вісник Університету «Україна»*. 2019. № 1 (22). С. 256-266.
6. Клименко Л. А., Гордієнко А. Ю. Аналіз методів стиснення зображень на основі дискретно-косинусного перетворення. *Інформаційно-керуючі системи на залізничному транспорті*. 2019. № 5. С. 24-31.
7. Вахіль А. І., Кузьменко І. М. Фрактальні методи стиснення зображень описали. URL: http://tef.kpi.ua/files/pdf/vahil_kuzmenko_1302541569.pdf (дата звернення: 25.08.2021).
8. Щербакова Ю.А., Скіцка М.В. Ефективність стиснення зображень в залежності від попередньої обробки. *Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: тези доповідей одинадцятої міжнародної науково-технічної конференції (8 – 9 квітня 2021 р.)*. 2021. Т. 2. С. 30.
9. Залевська О., Яблонський П., Сидоренко І., Феноген О. Реалізація алгоритму фрактального стиснення графічного зображення / *Сучасні проблеми моделювання*. 2021. С. 48-55.
10. Galabov M. *Fractal Image Compression*. 2003. doi: 10.1145/973620.973673.
11. Menassel R. *Optimization of Fractal Image Compression*. 2020. doi: 10.5772/intechopen.93051.
12. Menassel R. *Conception de métaheuristique d'optimisation pour la compression d'images*. 2018. URL: https://www.researchgate.net/publication/337210089_Conception_de_métaheuristique_d'optimisation_pour_la_compression_d'images (Last accessed: 17.03.2021).
13. Menassel R., Nini B., Tahar M. An improved fractal image compression using wolf pack algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*. 2017. Vol. 30. P. 1-11. doi: 10.1080/0952813X.2017.1409281.
14. Menassel R., Nini B., Mekhaznia T. *WolfPack algorithm for a fractal image compression*. 2016. P. 116-120. doi: 10.1145/3038884.3038903.
15. Khedgaonkar R. S., Kamble S. D. Application of Quadtree Partitioning in Fractal Image Compression using Error Based Approach. *IOSR Journal of Engineering (IOSRJEN)*. 2012. Vol. 2(1). P. 050-054. doi: 10.9790/3021-0215054.
16. Menassel R., Gaba I., Titi K. Introducing BAT inspired algorithm to improve fractal image compression. *International Journal of Computers and Applications*. 2020. Vol. 42(7). P. 697-704. doi: 10.1080/1206212X.2019.1638631.
17. Garg P., Gupta R., Tyagi R. K. Adaptive fractal image compression based on adaptive thresholding in DCT domain. In *Information Systems Design and Intelligent Applications* Springer, New Delhi. 2016. P. 31-40. doi: 10.1007/978-81-322-2755-7_4.
18. Agrawal A., Chauhan P. Design Image Compression for Fractal Image using Block Code Algorithm. *International Journal of Computer Sciences and Engineering*. 2018. Vol. 6. P. 451-455. doi: 10.26438/ijcse/v6i6.451455.
19. Amin S., Gupta R., Mehrotra D. Analytical Review on Image Compression Using Fractal Image Coding. In *Soft Computing: Theories and Applications*. 2018. P. 309-321. Springer, Singapore. doi: 10.1007/978-981-10-5699-4_30.
20. Novinsky N. Research of a New Algorithm for Acceleration of Fractal Images Compression. In *2019 International Conference on Engineering and Telecommunication (EnT) (IEEE)*. 2019. P. 1-3. doi: 10.1109/EnT47717.2019.9030573.

21. Щекань Н. П. Геометричні та алгебраїчні фрактальні методи в інформаційних технологіях обробки і аналізу потоків даних. *Моделювання та інформаційні системи в економіці* : зб. наук. пр. Київ : КНЕУ, 2018. Вип. 95. С. 205–219.

Anyshchenko O.S. SYNTHESIS OF HIGH-SPEED FRACTAL IMAGE COMPRESSION ALGORITHM

*The article reveals the synthesis of a high-speed fractal image compression algorithm. It is emphasized that fractal compression can be divided into two stages: splitting the image into many rank blocks and into many domain blocks (which can overlap each other); application of transformations for each domain-rank block: geometric, which reflects the domain block in the rank, and affine, which changes the brightness value of the domain block to the maximum correspondence to the brightness values of the rank block. It is emphasized that the quality of compression depends on the partitioning scheme used in the first stage. The more domain blocks, the better the chance of finding the most similar ranking block. The mathematical component of the algorithm is determined, the principles of establishing distance metrics are described. The implementation of the interaction of domains and ranges is schematically presented on the example of a real image. It is substantiated that in addition to direct conversion of pixel values, the domain block can also be subjected to general scaling (reduction in size to the size of the rank block, for example, interpolation or simple thinning), rotation and other affine transformation. It is noted that the classification of domain and rank blocks is designed to reduce the number of blocks, and, as a consequence, to speed up the image compression algorithm. Each domain block is classified before encoding. During the selection, the potential ranking block is also classified and compared only with the domains of the corresponding class (or several related classes). It is emphasized that mathematically a block of pixels forms a single whole, namely, a position vector in an abstract position space, where each individual point represents a different block. When a distance metric is applied to this space, the relative position of the two vectors determines their proximity. The high-speed fractal image compression algorithm uses an *r*-tree, the implementation of which is proposed in the image subjected to compression.*

Key words: *high-speed algorithm, synthesis, fractal, compression, image, optimization, compression, archiving.*