

УДК 514.18

РЕАЛІЗАЦІЯ АЛГОРИТМУ ФРАКТАЛЬНОГО СТИСНЕННЯ ГРАФІЧНОГО ЗОБРАЖЕННЯ

Залевська О.В., к.т.н.,

o.zalevska@kpi.ua, ORCID: 0000-0002-3163-1695

Яблонський П.М., к.т.н.,

ypn@ukr.net, ORCID: 0000-0002-1971-5140

Сидоренко Ю.В., к.т.н.

sulico5@gmail.com, ORCID: 0000-0002-1953-0410

Фіногенов О.Д., к.т.н.,

fenyaad@gmail.com, ORCID: 0000-0002-1708-5632

Ситник А.Ю

sytnik.akim@gmail.com, ORCID: 0000-0001-8085-2163

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Однією з сфер застосування фрактальної графіки є фрактальне стиснення графічних зображень. Фрактальне стиснення забезпечує такі переваги, в порівнянні з іншими видами стискання, як незалежність від роздільної здатності зображення, швидке розпакування формату, відновлення повного зображення за його частиною та малий об'єм результуючих файлів, можливість моделювання реальних об'єктів живої та неживої природи. Практичні реалізації фрактального компресора пропонують різні рівні стиснення. Нижні рівні мають більш спокійні критерії пошуку, щоб скоротити час обробки, але з втратою деталей. Вищі рівні дають дуже добрі деталі, але обробка кожного зображення займає багато часу. Також до переваг фрактального методу стиснення відносять перевагу у швидкості декомпресії, зробивши більшу частину обчислень на етапі стиснення, забезпечуючи такий самий або кращий ступінь стиснення. Ці переваги означають, що фрактальне стиснення зображень добре підходить для додатків, що вимагають швидкого доступу до високоякісних зображень. Однак фрактальне стиснення зображення не позбавлене обмежень. До них належить тривалий крок стиснення, що не дозволяє використовувати його в додатках, де важливо мати можливість розсилати стиснені зображення з мінімальними затримками, наприклад, пряму трансляцію відео через мережу, відеоконференції.

В зв'язку зі складністю алгоритму та його непередбачуваність фрактальне стиснення, на практиці, застосовується досить рідко в порівнянні з стисненням jpeg та іншими. В роботі запропоновано реалізацію даного методу за допомогою мови програмування Python та середовищем розробки PyChart від компанії Get Brains. Наведено діаграму

класів додатку, описано архітектуру та діаграму прецедентів, блок схему алгоритму роботи системи та розглянуто головні функції реалізації алгоритмів компресії та декомпресії зображень. За наведеними даними було створено програмний застосунок фрактальної компресії та декомпресії графічного зображення, що дозволяє його використання при менших навантаженнях на операційну систему комп'ютера та потребує простіших характеристик системи в порівнянні з jpeg та іншими відомими алгоритмами стискування зображень.

Ключові слова: фрактальне стиснення, компресія зображень, декомпресія, фрактальна графіка, геометрія зображення, комп'ютерна графіка.

Постановка проблеми. Зберігання даних пов'язано з великими об'ємами інформації яку вони в собі містять. Для забезпечення можливості зменшення об'єму даних без втрати їх характеристик призводить до потреби в розробці програмного застосунку, який зміг би зменшити такі об'єми без втрати початкових якісних характеристик. Для цього необхідно дослідити існуючі формати зберігання та створення графічних даних, відтворити та удосконалити зберігання графічної інформації у текстовому файлі та розробити програмний застосунок для фрактальної компресії та декомпресії зображення.

Аналіз останніх досліджень та публікацій Зображення можуть бути відтворені за допомогою невеликої кількості вихідних даних. Метод системи ітеративних функцій використовує коефіцієнти афінних перетворень для побудови складних фракталів[1]. Коефіцієнти зберігаються у текстовому файлі. Використовуючи коефіцієнти відновлення вхідне зображення потребує задіяння більших об'ємів пам'яті комп'ютера та більшу пропускну здатність каналів зв'язку[2]. Наприклад, ілюстрація розміром 640x480 точок займає близько 1 Мбайт, такий самий розмір має весь текст книги у 300 сторінок[3]. Створення електронних бібліотек і мультимедійних енциклопедій значно ускладнюється без використання програм компресорів з великим коефіцієнтом стискування та декомпресорів [4].

Кожне зображення володіє властивістю самоподібності. Встановлення осередків такої самоподібності та зберігання лише її коефіцієнту дозволяє стискати зображення в відношенні 10 000:1 в залежності від кількості самоподібних ділянок та початково розміру зображення [5]. Цей процес не залежить від роздільної здатності вихідного зображення. Практичні реалізації фрактального компресора пропонують різні рівні стиснення [6]. Нижні рівні мають більш спокійні критерії пошуку, щоб скоротити час обробки, але з втратою деталей. Вищі рівні дають дуже добрі деталі, але обробка кожного зображення займає багато часу. У роботі [7] детально показано, як алгоритм під назвою Iгри

Хаосу використовується для побудови системи ітерованих функцій за допомогою листа папороті.

Застосування стискування зображення з використанням алгоритмів фрактальної геометрії зберігає зображення в надмалому об'ємі без втрати роздільної здатності.

Формування цілей статті. Надати необхідний апарат для створення програмної реалізації алгоритму фрактальної компресії та декомпресії зображень.

Основна частина. Для створення даного програмного застосунку було використано мову програмування Python. Порівняно з іншими сучасними мовами програмування, таких як Java або C, Python досягає значно кращих результатів за короткий час. Програмний код Python значно коротший за код більшістю інших мов, що дає більше можливостей для аналізу і контролю його якості у короткі терміни.

Окрім того для даної мови створено досить велику кількість бібліотек, що дають можливість адаптування готових рішень для власних цілей, що в свою чергу дає можливість значно прискорити і спростити розробку у таких частинах роботи як створення інтерфейсу, імпортування та експортування зображення. PyCharm від компанії Get Brains є зручним середовищем розробки, що надає можливості аналізу коду, автоматичного рефакторингу, швидкої навігації, інтеграцію із Git.

Інтерфейс програмного продукту було розроблено за допомогою бібліотеки PyQt. Ця бібліотека пропонує просту і зрозумілу систему для створення інтерфейсів з реалізацією різних типів вікон для конкретних задач. Однією з ключових переваг даної бібліотеки є наявність простого застосунку, що забезпечує можливість розробки інтерфейсу мануально.

На рис. 1 наведено діаграму класів програмного застосунку. Як можна побачити із діаграми класів дана програма є досить лаконічною з погляду на архітектуру. Клас Mainwindow є класом інтерфейсу головного вікна, що відкривається при запуску програми. Одне з полів даного класу є об'єктом BrowseDialog, який по суті теж є інтерфейсом, але цей об'єкт відповідає за вікно відкриття файлу і вибору шляху вивантаження. Головний клас системи – клас Img – це клас в якому відбувається вся внутрішня обробка зображення. Конструктор цього класу створює об'єкт відповідно до переданого в нього параметру "type". Цей параметр позначає як саме буде відбуватися обробка, чи це буде компресія, чи декомпресія значення параметру може бути "cmpg" або "dcmpg" відповідно. В залежності від типу обробки відбувається компресія чи декомпресія вибраного файлу зображенню, яким відповідають процедури compress() або ж decompress(). Які в свою чергу будуть викликати за необхідністю внутрішні функції класу.

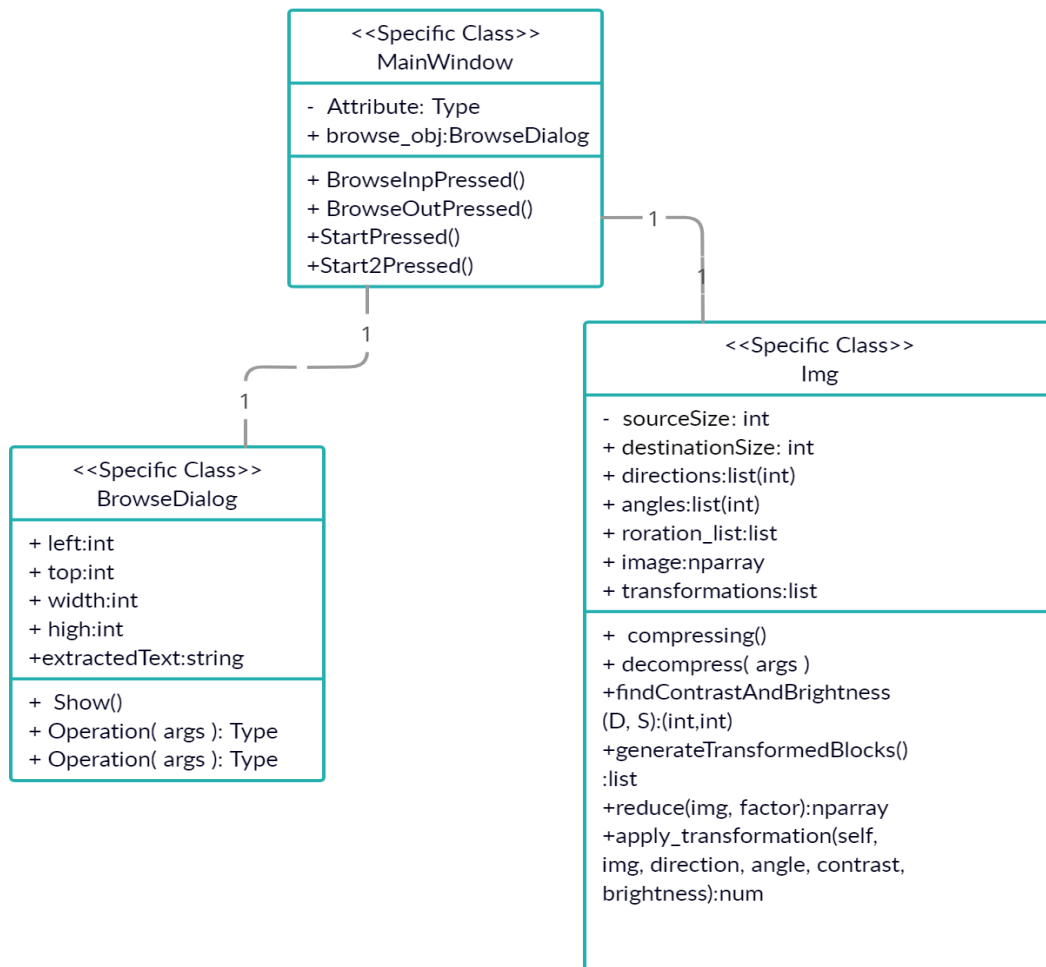


Рис. 1. Діаграма класів програмного застосунку для фрактальної компресії та декомпресії

Вся обробка файлів також зосереджена в класі `Img`. Конструктор цього класу сам витягає данні із файлу і формує новий файл відповідно до шляхів, що передаються йому від інтерфейсу `MainWindow`.

Наведемо головні функції алгоритму компресії даних. Функція `reduce` відповідає за зменшення зображення приводячи до середнього значення околиці. Функція `flip` відображає зображення дзеркально, якщо переданий в неї параметр `direction` дорівнює `-1`. Функція `rotate` повертає зображення на заданий кут. Для зберігання форми зображення кути повертання можуть бути лише `0`, `90`, `180` та `270` градусів. Повне перетворення зображення виконується у функції `aply_transformation`. Параметри `contrast`, `brightness`, `angle` відповідно позначають контрастність, яскравість та кут повертання.

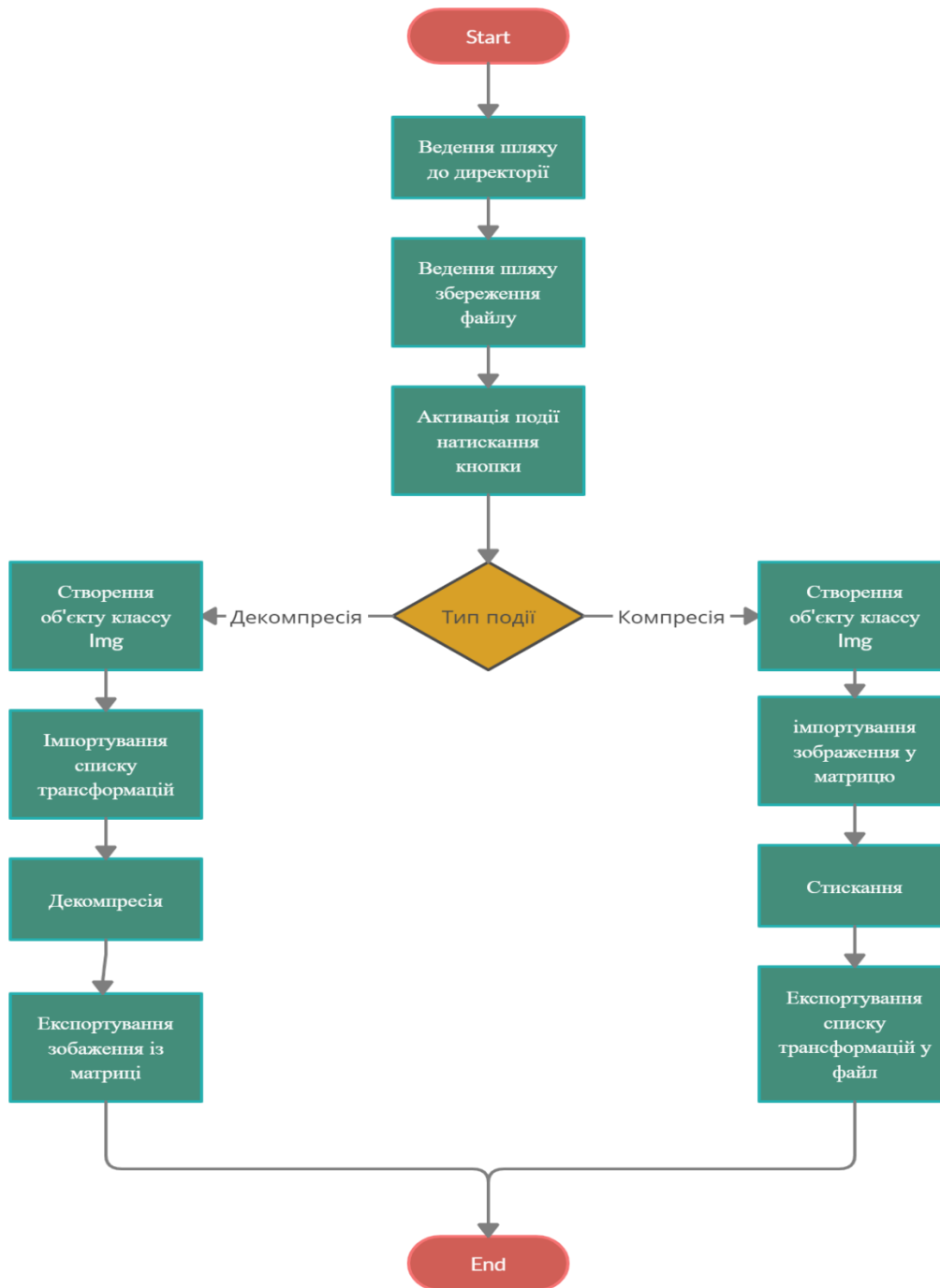


Рис. 2. Блок схема програмного застосунку для реалізації фрактального стиснення та відновлення зображення.

Реалізація алгоритму фрактального стискання:

1. Функція `generateTransformedBlocks` шукає всі можливі афінні перетворення доменних блоків
2. Для кожного окремого кінцевого блоку ми перевіряємо всі згенеровані доменні блоки, для кожного оптимізуємо контрастність і яскравість використовуючи у функції `findContrastAndBrightness` метод пошуку найменших квадратів з бібліотеки `numpy.linalg`.

3. Якщо протестоване перетворення є найкращим з перевірених воно зберігається. Перетворення зберігаються у список transformations, який після закінчення аналізу зберігається програмою у відповідному файлі вже вказаним методом бібліотеки pickle.

Алгоритм же декопресії починає роботу із фактично випадкового зображення і декілька разів застосовує стискаюче відображення. Кожне стискаюче відображення має свою унікальну кінцеву точку, тому алгоритм спрацює незалежно від вибраного початкового зображення.

Висновки. Реалізація фрактальне стискання надає можливість для застосування фрактальної компресії та декомпресії до графічних зображень. Це розширює межі застосування фрактальної графіки не тільки до графічних об'єктів, а й до геометрії живої та не живих природи.

Література

1. Anson L.F. Fractal Image Compression. *BYTE*, Oct. 1993, p. 195-202.
2. Barnsley M.F. Fractal Everywhere. 2nd ed., Academic Press, San Diego, 1993.
3. Barnsley M.F. Fractal Image Compression. Notices of the AMS, June 1996, p. 657-662.
4. Barnsley M.F., Demko S. Iterated function systems and the global construction of fractals. Proc. R. Soc. London, A 399 (1985), p. 243-275.
5. Barnsley M.F., Hurd L.P. Fractal Image Compression, AK Peters, Ltd., Wellesley, Massachusetts, 1992.
6. Ванін В.В., Залевська О.В. Опис стійких положень динамічних систем засобами фрактальної апроксимації. *Сучасні проблеми моделювання. Технічні науки*. Мелітополь, 2015. №. 4. С. 18-21
7. An Introduction to Fractal Image Compression. Literature Number: BPRA065. Texas Instruments Europe, October, 1997.

РЕАЛИЗАЦИЯ АЛГОРИТМА ФРАКТАЛЬНОГО СЖАТИЯ ГРАФИЧЕСКОГО ИЗОБРАЖЕНИЯ

Залевская О.В. Финогенов А.Д., Яблонский П.Н. Сидоренко Ю.В.,
Сытник А.Ю.

Одной из сфер применения фрактальной графики является фрактальное сжатие графических изображений. Фрактальное сжатие обеспечивает такие преимущества, по сравнению с другими видами сжатия, как независимость от разрешения изображения, быстрое распаковки формата, восстановление полного изображения с его частью и малый объем результирующих файлов, возможность моделирования реальных объектов живой и неживой природы. Практические реализации фрактального компрессора предлагают разные уровни сжатия. Нижние уровни имеют более спокойные критерии поиска, чтобы сократить время обработки, но с потерей деталей. Высокие уровни дают очень хорошие

детали, но обработка каждого изображения занимает много времени. Также к преимуществам фрактального метода сжатия относят преимущество в скорости декомпрессии, сделав большую часть вычислений на этапе сжатия, обеспечивая такой же или лучший уровень сжатия. Эти преимущества означают, что фрактальное сжатие изображений хорошо подходит для приложений, требующих быстрого доступа к высококачественным изображениям. Однако фрактальное сжатие изображения не лишено ограничений. К ним относится длительный шаг сжатия, не позволяющий использовать его в приложениях, где важно иметь возможность рассылать сжатые изображения с минимальными задержками, например, прямую трансляцию видео по сети, видеоконференции.

В связи со сложностью алгоритма и его непредсказуемостью фрактальное сжатие, на практике применяется достаточно редко по сравнению с сжатием *jpeg* и другими. В работе предложено реализовать данный метод с помощью языка программирования *Python* и средой разработки *PyCharm* от компании *Get Brauns*. Приведены диаграмму классов приложения, описано архитектуру и диаграмму прецедентов, блок-схему алгоритма работы системы и рассмотрены основные функции реализации алгоритмов компрессии и декомпрессии изображений. По приведенным данным было создано программное приложение фрактальной компрессии и декомпрессии графического изображения, что позволяет его использование при меньших нагрузках на операционную систему компьютера и требует простых характеристик системы по сравнению с *jpeg* и другими известными алгоритмами сжатия изображений.

Ключевые слова: фрактальное сжатие, компрессия изображений, декомпрессия, фрактальная графика, геометрия изображения, компьютерная графика.

IMPLEMENTATION OF FRACTAL COMPRESSION OF GRAPHIC IMAGE ALGORITHM

Olha Zalevska, Oleksij Finogenov, Petro Yablonsky, Iuliia Sydorenko,
Akim Sytnyk

One of the areas of application of fractal graphics is fractal compression of graphic images. Fractal compression provides advantages over other types of compression, such as independence of image resolution, fast format unpacking, recovery of the full image and part of the resulting files, the ability to model real objects of living and inanimate nature. Practical implementations of the fractal compressor offer different levels of compression. Lower levels have calmer search criteria to reduce processing time, but with loss of detail. Higher levels give very good details, but processing each image takes a long time. Also, the

advantages of the fractal compression method include the advantage in the rate of decompression, making most of the calculations at the stage of compression, providing the same or better degree of compression. These benefits mean that fractal image compression is well-suited for applications that require quick access to high-quality images. However, fractal image compression is not without limitations. These include a long compression step, which does not allow its use in applications where it is important to be able to send compressed images with minimal delays, such as live video over the network, video conferencing.

Due to the complexity of the algorithm and its unpredictability, fractal compression is rarely used in practice compared to the compression of jpeg and others. The paper proposes the implementation of this method using the Python programming language and the PyCharm development environment from Get Brains. The diagram of classes of the application is given, the architecture and the diagram of precedents, the block diagram of algorithm of work of system are described and the main functions of realization of algorithms of compression and decompression of images are considered. According to the data, a software application of fractal compression and decompression of the graphic image was created, which allows its use at lower loads on the computer operating system and requires simpler system characteristics compared to jpeg and other known image compression algorithms.

Keywords: fractal compression, image compression, decompression, fractal graphics, image geometry, computer graphics.

Referenses

1. Anson, L.F. (1993) Fractal Image Compression. BYTE, Oct. 195-202. [in English]
2. Barnsley, M.F.(1993) Fractal Everywhere. 2nd ed., Academic Press, San Diego [in English]
3. Barnsley, M.F., (1996) Fractal Image Compression. Notices of the AMS, 657-662. [in English]
4. Barnsley, M.F., Demko, S. (1985) Iterated function systems and the global construction of fractals. Proc. R. Soc. London, A 399, 243-275. [in English]
5. Barnsley, M.F., Hurd, L.P. (1992) Fractal Image Compression, AK Peters, Ltd., Wellesley, Massachusetts. [in English]
6. Vanin V.V., Zalevs'ka O.V. (2015) Description of stable positions of dynamic systems by means of fractal approximation. *Modern problems of modeling*. Melitopol, 4. 18-21 [in Ukrainian]
7. An Introduction to Fractal Image Compression. (1997) Literature Number: BPRA065. Texas Instruments Europe, October, [in English]