

УДК 621.3:681.34

О. Ю. СОФИНА, А. В. ЛОЗУН

Вінницький національний технічний університет, місто Вінниця

**МЕТОД СТИСНЕННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ПАРАЛЕЛЬНОГО АЛГОРИТМУ JPEG**

**Анотація.** Розроблено паралельний алгоритм стиснення зображень JPEG, який реалізовано у вигляді програмного додатку. Додаток розрахований на одну багатопроцесорну ЕОМ і на систему ЕОМ, об'єднаних в кластер. Разом з паралельною версією алгоритму стиснення реалізована й послідовна версія. Завдяки цьому можна запускати програму також лише на однопроцесорній системі. Після завершення роботи алгоритму стиснення є можливість побачити усі необхідні результати, а саме кількість процесів, коефіцієнт стиснення, час виконання й час передачі зображення. Розроблений програмний додаток може бути використаний при вирішенні практичних задач, пов'язаних з архівацією зображень, з метою забезпечення ефективної обробки зображення з високою швидкістю.

**Ключові слова:** алгоритм, стиснення зображень, JPEG, розпаралелювання обчислень.

**Аннотация.** Разработано параллельный алгоритм сжатия изображений JPEG, который реализован в виде программного приложения. Приложение рассчитано на одну многопроцессорную ЭВМ и на систему ЭВМ, объединенных в кластер. Вместе с параллельной версией алгоритма сжатия реализована и последовательная версия. Благодаря этому можно запускать программу также только на однопроцессорной системе. После завершения работы алгоритма сжатия есть возможность увидеть все необходимые результаты, а именно количество процессов, коэффициент сжатия, время выполнения и время передачи изображения. Разработанное программное приложение может быть использовано для решения практических задач, связанных с архивацией изображений, с целью обеспечения эффективной обработки изображения с высоким быстродействием.

**Ключевые слова:** алгоритм, сжатие изображений, JPEG, распараллеливание вычислений.

**Abstract.** Parallel algorithm for image compression JPEG, which is implemented as a software application, is developed. The application is designed for a multiprocessor computer system and computer combined into a cluster. Along with a parallel version of compression algorithm a sequential version is implemented. This allows you to run the program in single-processor system too. After completion of the compression algorithm, one can see all the needed results, such as the number of processes, compression ratio, the performance and the transmission time for images. The developed software application can be used in solving of practical problems related to the archiving of images, in order to ensure efficient image processing with high speed.

**Key words:** algorithm, image compression, JPEG, parallelizing computations.

**Вступ**

У зв'язку з широким розповсюдженням інформації виникають проблеми, пов'язані з її зберіганням та обробкою. Розвиток сучасних інформаційних систем і мереж привів до широкого використання цифрових зображень. У наш час багато галузей техніки, які мають відношення до отримання, обробки, зберігання і передачі інформації, значною мірою орієнтуються на розвиток систем, в яких інформація представлена у вигляді цифрових зображень. Такі зображення при зберіганні займають великі обсяги пам'яті. Так, растрове зображення розміром 1024 на 1024 пікселів з глибиною кольору 24 біти займає 3 Мб. Зрозуміло, що зберігання і передача великої кількості зображень у такому вигляді є досить трудомісткою задачею [1,2]. Стиснення зображень важливе для підвищення ефективності використання комунікаційних й інформаційно-обчислювальних ресурсів різноманітних систем. В даній статті акцент робиться на такій характеристиці процесу стиснення як швидкість. Було проаналізовано чимало алгоритмів стиснення зображень [3]. На основі проаналізованих даних, виходячи з простоти обчислень і реалізації, для подальшого дослідження було обрано алгоритм JPEG [4].

Об'єктом дослідження є процес обробки зображень, а предметом – методи стиснення інформації.

**Актуальність**

У сучасному світі зображення є невід'ємною складовою мультимедійної інформації, що найчастіше створюється, накопичується і зберігається на цифрових носіях та передається каналами зв'язку. Компресія відповідних файлів дає змогу пропорційно підвищити швидкість обміну інформацією по мережі та зменшити обсяги використання дискового простору. Тому проблема підвищення ефективності й швидкості стиснення зображень не втрачає своєї актуальності протягом останніх десятиліть і, ймовірно, не втратить у найближчому майбутньому.

**Мета**

Метою даного дослідження є підвищення швидкості процесу архівації зображень за допомогою паралельного алгоритму стиснення JPEG.

**Задачі**

1. Розробити паралельний алгоритм стиснення зображень JPEG.
2. Реалізувати розроблений алгоритм у вигляді програмного продукту.
3. Реалізувати також послідовний алгоритм стиснення з метою порівняння й оцінки результатів стиснення вихідного зображення для обох алгоритмів.

**Розв'язання задач**

Рішення щодо розпаралелювання обчислень було прийняте внаслідок аналізу літератури щодо ефективності паралельних обчислень [5]. Вони дають змогу суттєво підвищити швидкість обробки зображень. Для вирішення поставлених задач спочатку реалізується відомий алгоритм стиснення JPEG.

Існують стандартні етапи роботи алгоритму JPEG наведені на рис. 1.

Одним з ключових етапів роботи алгоритму є дискретне косинусне перетворення, за допомогою якого реалізується перехід у спектральний простір.

Формули дискретного косинусного перетворення (1) – (3) наведені нижче.

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[ \frac{(2x+1)i\pi}{2N} \right] \cos \left[ \frac{(2y+1)j\pi}{2N} \right], \quad (1)$$

$$f(x, y) = \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(i)C(j) DCT(i, j) \cos \left[ \frac{(2x+1)i\pi}{2N} \right] \cos \left[ \frac{(2y+1)j\pi}{2N} \right], \quad (2)$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0, \\ 1, & x > 0. \end{cases} \quad (3)$$

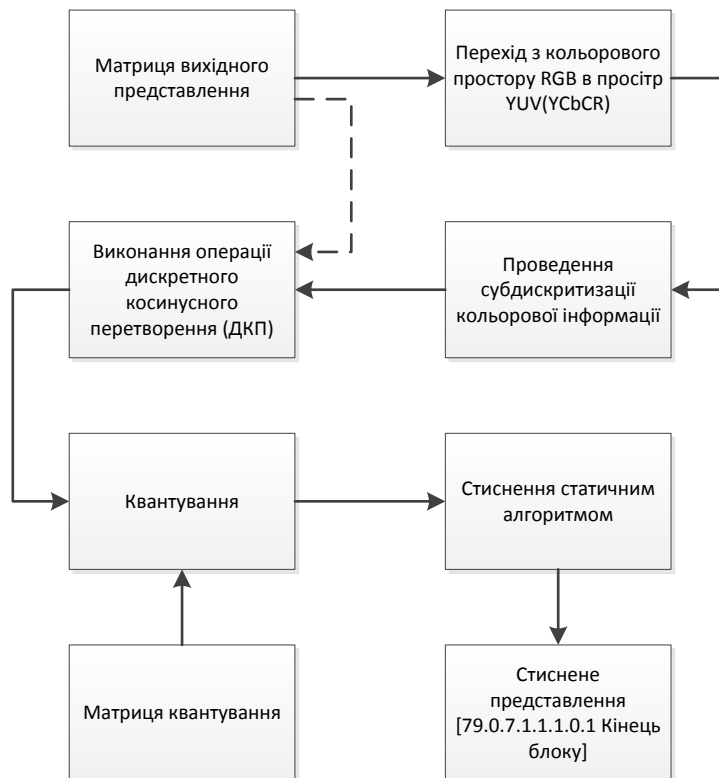


Рисунок 1 – Схема етапів роботи JPEG

Далі з метою покращення швидкодії процесу стиснення виконується розпаралелювання обчислень.

Для організації паралельної обробки даних необхідно забезпечити передачу даних між вузлами розподіленої системи. Для того, щоб паралельна програма могла працювати у різних за конфігураціями мережах, а також і на паралельних комп’ютерах, необхідний стандарт, що реалізує конфігураційно-залежні операції. В першу чергу маються на увазі операції передачі даних між вузлами. З цією метою було використано стандарт MPI (Message Passing Interface) [6,7].

При реалізації алгоритму паралельного стиснення JPEG вихідне зображення розбивається на рядки висотою 16 пікселів і довжиною, яка визначається шириною зображення. Кількість таких рядків

висота зображення в пікселях

16 пікселів

визначатиметься як процесами для передачі згідно з формулою (4):

. Загальна кількість рядків ділиться між

$$kilSTRONProc(i) = \begin{cases} \frac{\text{кільк\_рядків}}{n} + 1, \text{ якщо } i \leq \text{кільк\_рядків} \% n; \\ \frac{\text{кільк\_рядків}}{n}, \text{ у інших випадках,} \end{cases} \quad (4)$$

де  $kilSTRONProc(i)$  – кількість рядків, які передаються процесу з номером  $i$  при загальній кількості процесів  $n$ .

Після визначення кількості рядків, які отримає кожен процес, відбувається їх передача спочатку процесу 1 у кількості  $kilSTRONProc(1)$ , потім процесу 2 у кількості  $kilSTRONProc(2)$ , починаючи з рядка з номером  $kilSTRONProc(1) + 1$ , і так далі всім іншим процесам згідно з даним алгоритмом. Завдяки такій схемі розподілу даних досягається рівномірність розподілу навантаження по процесорах й повна їх незалежність відносно один одного. Цим визначаються високі показники швидкодії й ефективності обчислень.

Схема алгоритму роботи програми наведена на рис. 2.

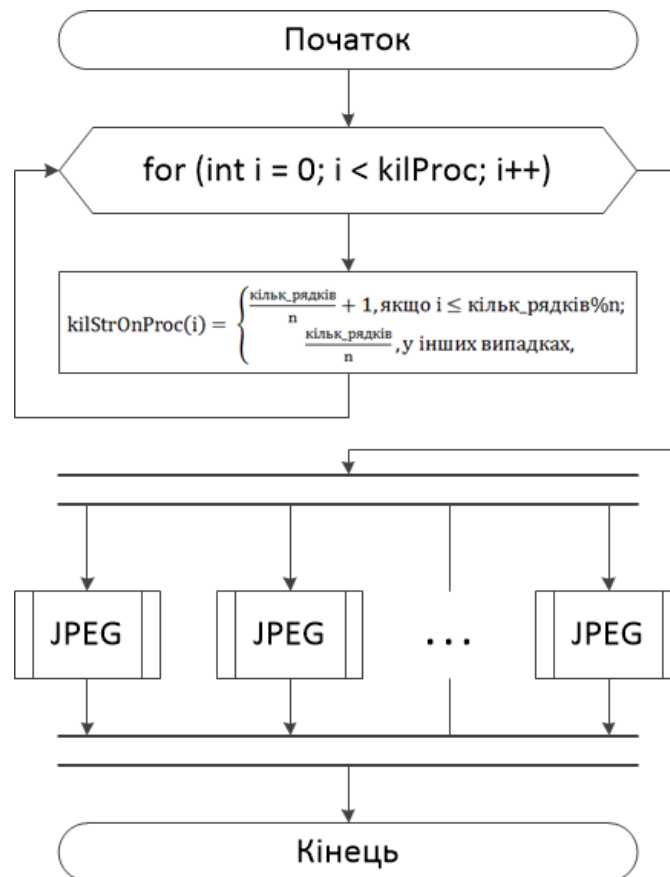


Рисунок 2 – Схема роботи програми

Головний процес здійснює розподіл даних згідно з вищеописаним алгоритмом. При запуску на багатопроцесорній системі створюється вказана кількість екземплярів програми. Кожен екземпляр визначає свій порядковий номер  $i$  в залежності від цього номеру й кількості процесорів або процесів (в ідеальному варіанті на один процесор припадає 1 процес) виконує частину алгоритму. Процес з номером 0 називається головним й відповідає за введення/виведення інформації, розбиття вихідного зображення та розсилання його блоків іншим процесам.

Кожний процес отримує блок даних, що являє собою частину вихідного зображення. Процес розглядає свій блок даних як окреме зображення й застосовує до нього алгоритм стиснення JPEG.

Паралельна програма стиснення зображень алгоритмом JPEG написана на мові C++ в середовищі Microsoft Visual Studio 2010 з використанням пакету MPICH 3.0.

Інтерфейс програмного додатку наведений на рис. 3.

Засобами Windows у програмі реалізована функція відкриття будь-якого растрового зображення, яка дає змогу вивести на екран, стиснути й дослідити необхідний графічний файл.

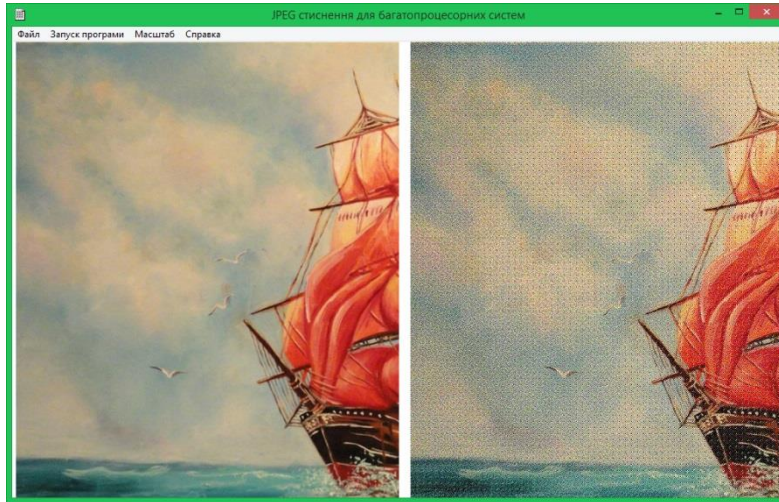


Рисунок 3 – Вікно програми: ліва частина – вихідне зображення, права – оброблене зображення

У лівій половині вікна виводиться вихідний графічний файл. У меню «Запуск програми» користувач має змогу вибрати вид стиснення. Після проведення обчислень в правій частині вікна відображається результат стиснення вихідного зображення й подальшого розпакування. Завдяки цьому можна візуально оцінити втрати. З метою більш детальної оцінки втрат зображення можна масштабувати.

Відлагодження й тестування паралельної програми проводилось з використанням програмно-апаратної платформи на базі процесору Intel Core i5-2400 з робочою частотою 3.1 ГГц, який містить у собі 4 обчислювальних ядра з встановленим пакетом MPICH 3.0. Також для перевірки роботи програми застосовувалась програмно-апаратна платформа на базі процесору AMD Phenom II N870 з частотою 2,3 ГГц. В результаті запуску програми на другій платформі була отримана залежність часу виконання від розміру зображення, наведена на рис. 4.

Для дослідження були вибрані 5 точкових зображень різних розмірів:

- 1.bmp – розмір 1920×1200, обсяг 6,59 Мб;
- 2.bmp – розмір 2560×1600, обсяг 11,7 Мб;
- 3.bmp – розмір 3072×2304, обсяг 20,2 Мб;
- 4.bmp – розмір 2975×4463, обсяг 37,9 Мб.

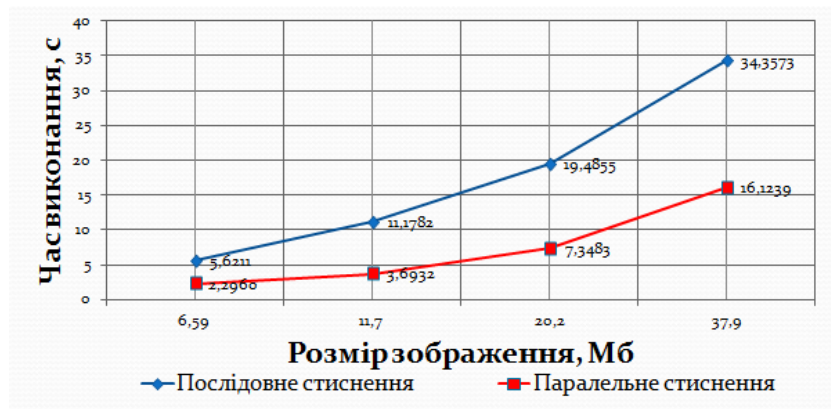


Рисунок 4 – Залежність часу виконання від розміру зображення

Отже, як видно з графіків зі збільшенням розміру зображення зростає час виконання стиснення, проте при використанні паралельного алгоритму час виконання програми для кожного із зображень приблизно в 2,5 рази менший, ніж час виконання при послідовному стисненні.

#### Висновки

1. Розроблено метод стиснення зображень на основі розпаралелюванні алгоритму JPEG.
2. На основі створеного алгоритму розроблено програмний додаток для реалізації паралельної обробки зображень.
3. Реалізована можливість використання програмного продукту як на одній багатопроцесорній машині, так і на багатьох ЕОМ, об'єднаних в кластер.
4. Проведено порівняльний аналіз, який доводить ефективність застосування паралельних обчислень для обробки цифрових зображень.
5. Закладено основу для модифікації розробленого алгоритму шляхом підвищення коефіцієнта стиснення і якості зображення застосуванням певних засобів фільтрації й фрактального алгоритму стиснення.

#### Список літератури

1. Претт У. Цифровая обработка изображений / Претт У. ; пер. с англ. Д. С. Лебедева – М. : Мир, 2002. – 792 с.
2. Двухэтапные методы и алгоритмы сжатия цифровых изображений на основе дискретных преобразований Уолша [Електронний ресурс]: – Режим доступу: <http://www.dissercat.com/content/dvukhetapnye-metody-i-algoritmy-szhatiya-tsifrovyykh-izobrazhenii-na-osnove-diskretnykh-preob>. – Назва з екрану.
3. Ватолин Д. С. Тенденции развития алгоритмов архивации графики / Д. С. Ватолин // Открытые системы. – 2010. – № 2. – С. 15-24.
4. John W. O'Brien. The JPEG Image Compression Algorithm / John W. O'Brien // APPM-3310 FINAL PROJECT. – 2005. - №4. – Р. 4-7.
5. Антонов А. С. Введения в параллельные вычисления : [учебно-справочное издание] / Антонов А. С. – М. : Научно-исследовательский вычислительный центр, 2002. – 69 с.
6. Message Passing Interface [Електронний ресурс]: – Режим доступу: [http://uk.wikipedia.org/wiki/Message\\_Passing\\_Interface](http://uk.wikipedia.org/wiki/Message_Passing_Interface). – Назва з екрану.
7. Шпаковский Г. И. Программирование для многопроцессорных систем в стандарте MPI / Г. И. Шпаковский, Н. В. Серикова. — Мн. : БГУ, 2002. –323 с.  
Стаття надійшла: 25.11.2014.

#### Інформація про авторів

**Софина Ольга Юрійвна** – к.т.н., доцент, доцент кафедри автоматики та інформаційно-вимірювальної техніки ВНТУ.

**Лозун Алла Василівна** – студент кафедри автоматики та інформаційно-вимірювальної техніки ВНТУ.