

Пристосування словникових методів компресії до прогресуючого ієрархічного стиснення зображень без втрат

<https://doi.org/10.31713/MCIT.2021.47>

Шпортко Олександр Володимирович
Кафедра інформаційних систем та обчислювальних методів

ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янука»
м. Рівне, Україна
ITShportko@ukr.net

Бомба Андрій Ярославович
Кафедра комп'ютерних наук та прикладної математики

Національний університет водного господарства та природокористування
м. Рівне, Україна
abomba@ukr.net

Шпортко Леся Василівна

Циклова комісія інформатики та комп'ютерної техніки
ДВНЗ «Рівненський фаховий коледж економіки та бізнесу»
м. Рівне, Україна
LChportko@gmail.com

Abstract — Обґрунтована доцільність та наведений спосіб обходу пікселів для реалізації прогресуючого ієрархічного стиснення зображень без втрат. Запропонована модифікація алгоритму словникової компресії LZ77 для підвищення ефективності такого стиснення за допомогою додаткового пошуку співпадаючих послідовностей по найближчих опрацьованих раніше пікселях

Keywords — прогресуюче стиснення зображень; стиснення без втрат; словникові методи компресії; модифікації алгоритму LZ77.

I. ВСТУП

У сучасному світі зображення є невід'ємною складовою мультимедійної інформації, яка найчастіше створюється, накопичується і зберігається на цифрових носіях та передається каналами зв'язку [1]. Компресія відповідних файлів дає змогу пропорційно підвищити швидкість обміну інформацією по мережі та зменшити обсяги використання дискового простору. Всі графічні формати за принципом стиснення зображень поділяють на два основні класи: з втратами та без втрат. І якщо для переважної більшості алгоритмів компресії зображень з втратами можна забезпечити потрібний коефіцієнт стиснення (відношення розмірів стиснутого до нестиснутого файлів зображення, надалі – КС) за рахунок погіршення якості, то рівень стиснення зображень без втрат залежить, власне, лише від перепадів кольорів їх пікселів та самого алгоритму стиснення, не регулюється програмно і становить в середньому тільки 30–70%. На сьогодні дизайнери та розробники Web-сайтів найчастіше зберігають фотореалістичні зображення у форматі JPEG, а дискретно-тонові і ті, де втрати неприпустимі, – у

форматі PNG. Опрацювання яскравостей пікселів зображень у популярних графічних форматах, які виконують стиснення без втрат (у тому числі, і у форматі PNG [2, с. 249–317]), найчастіше здійснюється послідовно по рядках зверху вниз, а у кожному рядку – посліпль зліва направо. Як наслідок, вивести стиснуте зображення у цих форматах можливо лише після декодування всіх пікселів, а декомпресія знімків чи малюнків з мільйонами пікселів при такому способі обходу може тривати декілька секунд незалежно від розміру області чи роздільної здатності пристрою виводу.

Поряд з цим, для прискорення виводу великих зображень у форматах компресії з втратами вже розроблені графічні формати, які застосовують прогресуюче (поступальне) ієрархічне опрацювання пікселів [3, с. 176] (наприклад, вейвлети [4]). В процесі застосування цього способу опрацювання зображення пікселі обходять пошарово, збільшуючи щоразу роздільну здатність (прогресуюча складова). При цьому в процесі послідовної обробки даних чергового шару використовують дані попередніх шарів (ієрархічна складова). Зображення з пікселів чергового шару фактично є зменшеною у декілька разів (найчастіше – у чотири) копією зображення з пікселів наступного шару, а останній шар співпадає з вхідним зображенням. Тому під час прогресуючого ієрархічного декодування деталі зображення проявляються поступово. Зупинити таке декодування можливо вже після декомпресії шару з кількістю пікселів, не меншою від області виводу по кожній з осей, не очікуючи відтворення всіх пікселів зображення. З іншого боку загальновідомо, що словникові алгоритми дають змогу зменшити

КС дискретно-тонових зображень в декілька разів [3]. Отже, пристосування словникових алгоритмів до прогресуючого ієрархічного стиснення зображень без втрат, що є метою цього дослідження, є на сьогодні актуальним завданням.

II. АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ.

ПРИНЦИПИ СТИСНЕННЯ ЗОБРАЖЕНЬ КОНТЕКСТНО-ЗАЛЕЖНИМ АЛГОРИТМОМ LZ77

Як відомо, стиснення зображень без втрат у графічних форматах найчастіше відбувається в три етапи: на першому яскравості компонентів пікселів перетворюються за допомогою предикторів [5], які не стискають зображення, але збільшують нерівномірність розподілу яскравостей і тому підвищують ефективність третього етапу; на другому контекстно-залежне кодування зменшує надлишковості між подібними чи однаковими фрагментами; на третьому контекстно-незалежне кодування усуває надлишковості між переважаючими значеннями яскравостей компонентів. Контекстно-залежне кодування може зменшувати КС в декілька разів за рахунок однакових фрагментів, але такі фрагменти рідко трапляються у фотореалістичних зображеннях, тому єдиним універсальним етапом стиснення зображень без втрат є контекстно-незалежне кодування. За нашими підрахунками, застосування контекстно-незалежного алгоритму (наприклад, кодування Хафмана чи арифметичного кодування [2, 3, 5]) після використання предикторів без контекстно-залежного алгоритму в середньому забезпечує КС на рівні лише 45 %. Тому у цій статті розглянемо механізм дії популярного контекстно-залежного словникового алгоритму LZ77 [6] та модифікуємо його для прогресуючого ієрархічного стиснення зображень без втрат.

Серед контекстно-залежних алгоритмів у форматах графічних файлів найчастіше використовується словниковий алгоритм LZ77, оскільки він забезпечує найшвидше декодування [7, с. 82]. Описуючи словникові алгоритми, фіксовану кількість попередніх закодованих неподільних елементів (літералів) вхідного потоку називають *словником*, а наступних незакодованих – *буфером*. Алгоритм LZ77 (в контексті формату словникової компресії DEFLATE [8]) базується на заміні у вихідному потоці послідовності чергових літералів буфера посиланням на аналогічну послідовність літералів словника у вигляді пари чисел <довжина, зміщення від закінчення словника>. У випадку відсутності аналогічної послідовності літералів у словнику, довшої двох літералів, перший літерал буфера переноситься у вихідний потік без змін. Після цього закодовані літерали переносяться з початку буфера в кінець словника і кодування продовжується аналогічно аж до закінчення літералів вхідного потоку. Співпадаюча послідовність може виходити за межі словника в область буфера, але має розпочинатися в словнику. Максимальні розміри словника та буфера встановлюються конкретними реалізаціями алгоритму. Сукупність словника з закодованими

символами та буфера з незакодованими ще називають *ковзним вікном*, оскільки вони весь час синхронно переміщуються по елементах потоку.

Під час декодування кодів алгоритму LZ77 окремі літерали копіюються у вихідний потік без змін. Пари ж <довжина; зміщення> декодуються шляхом послідовного копіювання з кінця вихідного потоку за вказаним зміщенням в кінець вихідного потоку необхідної кількості літералів.

Згідно алгоритму LZ77, співпадаючі послідовності максимальної довжини шукають у словнику з кінця справа наліво, оскільки однакові фрагменти даних найчастіше зустрічаються недалеко. Як наслідок – менші зміщення у вихідному потоці трапляються частіше від більших, і тому кодуються у форматі словникової компресії DEFLATE [8] меншою кількістю бітів.

В зображеннях однакові фрагменти чи фрагменти однакової структури також найчастіше трапляються недалеко, оскільки саме суміжні пікселі мають між собою найбільшу кореляцію [9, с. 675]. Але під час послідовного обходу менші зміщення мають однакові фрагменти, розміщені по горизонталі. Зміщення ж однакових фрагментів по вертикалі з кожним рядком збільшуються на кількість яскравостей пікселів в рядку. Якби пікселі зображень обходилися послідовно по стовпцях, то однакові фрагменти по вертикалі кодувалися б значно меншими зміщеннями, але тоді більші зміщення мали б однакові фрагменти по рядках. Тобто результати кодування зображення алгоритмом LZ77 можуть зазнати суттєвих змін після його повороту на 90°. У цій же статті ми покажемо, як підвищити ефективність застосування алгоритму словникової компресії LZ77 в процесі прогресуючого стиснення зображень без втрат за рахунок кодування зміщень до найближчих опрацьованих раніше пікселів як по горизонталі, так і по вертикалі меншими значеннями, ніж до решти пікселів.

III. ПОСЛІДОВНІСТЬ ОБХОДУ ПІКСЕЛІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРЕСУЮЧОГО ІЄРАРХІЧНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ БЕЗ ВТРАТ З ВИКОРИСТАННЯ АЛГОРИТМУ LZ77

Поступальне ієрархічне стиснення зображень дає змогу, з одного боку, прискорити декодування, а з іншого – врахувати в процесі стиснення значення попередніх опрацьованих елементів з чотирьох, а не лише з двох різних боків. Саме тому для досягнення мети дослідження нами розроблено дієву схему обходу пікселів та відповідні предиктори [5]. Зокрема, для прогресуючого ієрархічного обходу ми пропонуємо схему, за якою на першому шарі пікселі зображення опрацьовуються послідовно, починаючи з першого, по рядках зверху вниз, а у кожному рядку – підряд зліва направо з кроком $h_1 = 2^k$, де k визначається з

$$\text{умови } k = \left\lfloor \log_2 \left(\frac{\max(\min(\text{height}; \text{width}); 16) - 1}{15} \right) \right\rfloor,$$

height – кількість рядків, *width* – кількість стовпців

пікселів зображення (рис. 1а). Цей крок забезпечує опрацювання на першому шарі принаймні 16 пікселів по кожній з осей (як у піктограмах), якщо зображення має неменші розміри. На наступних шарах ($l = 2, k + 1$) проміжні пікселі зображення обробляються в два проходи: на першому послідовно опрацьовуються ті з них, які містяться на перетині діагоналей квадратів з вершинами у суміжних пікселях попередніх шарів з кроком $h_l = 2^{k+2-l}$ як по рядках, так і по стовпцях (див. рис. 1, б), а на другому необроблені пікселі послідовно обходяться між суміжними пікселями попередніх шарів і пікселями першого проходу з тим самим кроком по стовпцях і з удвічі зменшеним – по рядках (див. рис. 1, в).

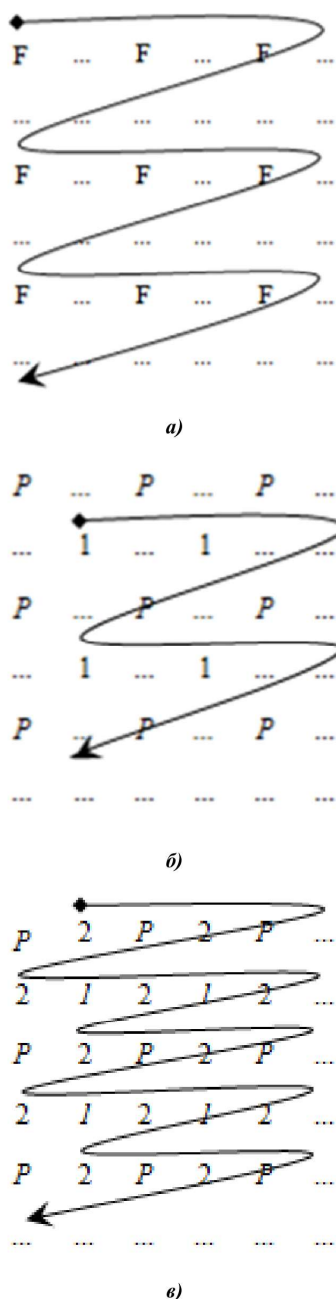


Рисунок 1. Черговість обходу пікселів в процесі прогресуючого ієрархічного опрацювання: а) пікселів першого шару; б) пікселів першого проходу чергового шару, в) пікселів другого проходу чергового шару

На рис. 1 символом F позначені пікселі першого шару, символом P – пікселі попередніх шарів, цифрою 1 – пікселі першого проходу чергового шару, цифрою 2 – пікселі другого проходу чергового шару. Пікселі, які не опрацьовуються на черговому шарі, виділені курсивом.

Послідовно розміщені дані всіх проходів формують вхідний потік для кодування. Дані наступного шару збільшують кількість опрацьованих пікселів приблизно в 4 рази, тому зупиняти процес декодування можливо вже після заповнення області виводу, не опрацьовуючи коди до кінця. Запропонована послідовність обходу пікселів зображення дає змогу не лише прискорити декодування, коли розміри області виводу значно менші від розмірів зображення, а й застосовувати ієрархічні предиктори для прогнозування значення кожного елемента чергового пікселя [5].

IV. МОДИФІКАЦІЯ АЛГОРИТМУ LZ77 ДЛЯ ПРОГРЕСУЮЧОГО ІЄРАРХІЧНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ БЕЗ ВТРАТ

Для підвищення ефективності застосування алгоритму LZ77 в процесі прогресуючого ієрархічного обходу на всіх шарах, починаючи з другого, будемо виконувати пошук однакових послідовностей для елементів буфера не лише у словнику, а й починаючи з найближчих раніше опрацьованих пікселів з кроком чергового шару. Оскільки серед раніше опрацьованих пікселів черговий піксель X має найвищий рівень кореляції саме з найближчими серед них, то найменші 6 зміщень (з кодами від 1 до 6) закріпимо за найближчими опрацьованими раніше пікселями (рис. 2). Також, для забезпечення однозначності декодування, всі зміщення в словнику збільшимо на 6.

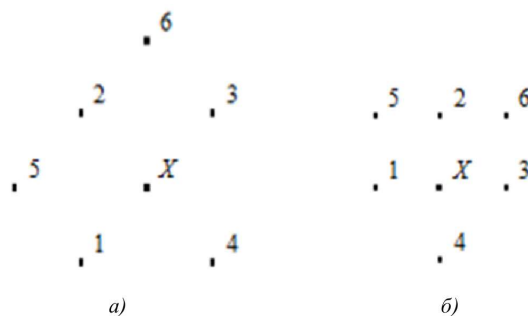


Рисунок 2. Коди зміщень до яскравостей суміжних опрацьованих раніше пікселів на шарах, починаючи з другого: а) для першого проходу; б) для другого проходу

На перших проходах чотири найближчих суміжних пікселі віддалені діагонально (див. рис. 2, а). Це пікселі з попередніх шарів (див. рис. 1, б) і в словнику вони мають великі зміщення, а однакові послідовності в зображенні, які з них починаються, взагалі можуть бути розкидані по всьому словнику. П'яте та шосте зміщення для першого проходу кодують опрацьовані пікселі того самого проходу, але в словнику вони мають більші зміщення.

На других проходах чотири найближчих суміжних пікселі віддалені по горизонталі та вертикалі (див. рис. 2, б). Це пікселі з попередніх шарів (див. рис. 1, в), вони розміщені ближче до чергового пікселя, ніж на першому проході, і тому від них починається більше однакових послідовностей. П'яте та шосте зміщення для другого проходу віддалені діагонально і кодують пікселі того самого проходу з попереднього рядка, але в словнику вони мають більші зміщення. Симетричні пікселі з наступного рядка не кодуються, оскільки відносно чергового пікселя вони ще не опрацьовані. Отже, пошук однакових послідовностей починаючи з найближчих опрацьованих раніше пікселів дає змогу або зменшити їх зміщення відносно словника або ж віднаходить такі послідовності, які взагалі розкидані по різних шарах в словнику. Найістотніше пошук однакових послідовностей з найближчих опрацьованих раніше пікселів зменшує КС на останньому шарі, оскільки такі пікселі містяться в зображенні поряд з пікселями буфера.

Отже, додатковий пошук однакових послідовностей починаючи з найближчих опрацьованих раніше пікселів збільшує ймовірність знаходження таких послідовностей та використовує менші зміщення, ніж під час пошуку по словнику, і тому зменшує КС алгоритму LZ77 зокрема і зображень загалом.

V. ВИСНОВКИ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1. В нових версіях графічних форматів та нових форматах компресії зображень без втрат доцільно реалізувати прогресуюче ієрархічне стиснення, оскільки це дозволяє суттєво прискорити декодування, коли розміри області виводу менші від розмірів зображення.
2. Зменшення розмірів стиснутих прогресуючим ієрархічним способом зображень досягається в основному на останніх шарах, оскільки пікселі, які при цьому використовуються як словниковим алгоритмом, так і предикторами [5], мають з прогнозованим пікселем в середньому найвищий рівень кореляції відносно попередніх шарів.
3. Підвищити ефективність застосування класичного алгоритму LZ77 в процесі прогресуючого ієрархічного стиснення зображень без втрат можливо за рахунок додаткового пошуку однакових послідовностей

починаючи з найближчих опрацьованих раніше пікселів.

4. Словникові алгоритми суттєво зменшують КС насамперед штучних дискретно-тонових зображень [10], оскільки такі зображення містять багато однакових послідовностей яскравостей компонентів пікселів. Наприклад, як показали наші дослідження, по набору АСТ таке зменшення КС в середньому перевищило 1 bpp.

Надалі, з метою додаткового зменшення розмірів файлів стиснутих зображень без втрат і прискорення декодування в процесі прогресуючого ієрархічного стиснення, нами планується дослідити ефективність застосування різницевої кольорних моделей [11] до окремих фрагментів зображень.

ЛІТЕРАТУРА

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Introduction to Algorithms," Third Edition, vol. 1, Kiyv : Dialektika, 2020, 648 p., (In Ru).
- [2] Дж. Миано, Форматы и алгоритмы сжатия изображений в действии : учеб. пособ. М. : Триумф, 2003, 336 с., (Серия: Практика программирования).
- [3] Д. Сэлмон. Сжатие данных, изображений и звука. М. : Техносфера, 2006, 368 с. (Серия: Мир программирования: цифровая обработка сигналов).
- [4] С. Уэлстид. Фракталы и вейвлеты для сжатия изображений в действии : учеб. пособ. М. : Триумф, 2003, 320 с., ил.
- [5] A. Shportko and V. Postolati, "Development of Predictors to Increase the Efficiency of Progressive Hierarchic Context-Independent Compression of Images Without Losses," in Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021), April 22–23, 2021, Kharkiv, Ukraine, vol. 1, ceur-ws.org, pp. 1026–1038, URL: <http://ceur-ws.org/Vol-2870/paper77.pdf>.
- [6] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression" in IEEE Transactions on Information Theory, May 1977, vol. 23(3), pp. 337–343.
- [7] Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М. : ДИАЛОГ-МИФИ, 2003. 384 с.
- [8] P. Deutsch, "DEFLATE Compressed Data Format Specification version 1.3," RFC 1951, 1996, Alladin enterprises, May 1996, 15 p.
- [9] Э. Прэрт. Цифровая обработка изображений / пер. с англ. М. : Мир, 1982, кн. 2, 480 с., ил.
- [10] "MinPNG 1.0 – утиліта для мінімізації розміру файлів зображень у форматі PNG (True Color)," URL: <http://apserver.org.ua/peregl.php?d=view&tid=131>, (дата звертання: 26.09.2021).
- [11] О. В. Шпортко. Використання різницевої кольорових моделей для стиснення RGB-зображень без втрат. Відбір і обробка інформації, 2009. № 31 (107). С. 90–97.