

ПРИКЛАДНА МАТЕМАТИКА, ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА

УДК 004.043

<https://doi.org/10.31713/vt4202111>

Шпортко О. В., к.т.н., доцент (ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука», м. Рівне, ITShportko@ukr.net), **Шпортко Л. В., викладач** (Рівненський фаховий коледж економіки та бізнесу, м. Рівне, LChportko@gmail.com), **Бомба А. Я., д.т.н., професор** (Національний університет водного господарства та природокористування, м. Рівне, abomba@ukr.net)

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ СЛОВНИКОВИХ МЕТОДІВ КОМПРЕСІЇ ДЛЯ ПРОГРЕСУЮЧОГО ІЄРАРХІЧНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ БЕЗ ВТРАТ

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 однакових послідовностей по найближчих опрацьованих раніше пікселях. Подано результати застосування запропонованого алгоритму для стиснення зображень набору АСТ. Показано, що застосування наведеної модифікації алгоритму LZ77 дає змогу, наприклад, додатково зменшити коефіцієнти стиснення дискретно-тонових зображень цього набору в середньому на 0.59 брб.

Ключові слова: прогресуюче стиснення зображень; стиснення без втрат; словникові методи компресії; модифікації алгоритму LZ77.

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 – поспіль зліва направо. Як наслідок, вивести стиснуте зображення у цих форматах можливо лише після завершення декодування, 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 його 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 всього зображення.

З іншого боку, загальновідомо, що зменшення КС дискретно-тонових зображень в декілька разів під час послідовного обходу можливе за рахунок застосування словникових алгоритмів [3, С. 81–125], які, як правило, використовуються на одному з етапів стиснення зображень без втрат. Отже, **пристосування словникових алгори-**

тмів до прогресуючого ієрархічного стиснення зображень без втрат, що є метою цього дослідження, є на сьогодні актуальним завданням.

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 етапі контекстно-незалежне кодування усуває надлишковості між переважаючими значеннями яскравостей компонентів. Контекстно-залежне кодування може зменшувати КС в декілька разів за рахунок таких самих фрагментів, але однакові 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], модифікуємо його для прогресуючого ієрархічного стиснення зображень без втрат та дослідимо ефективність запропонованої модифікації.

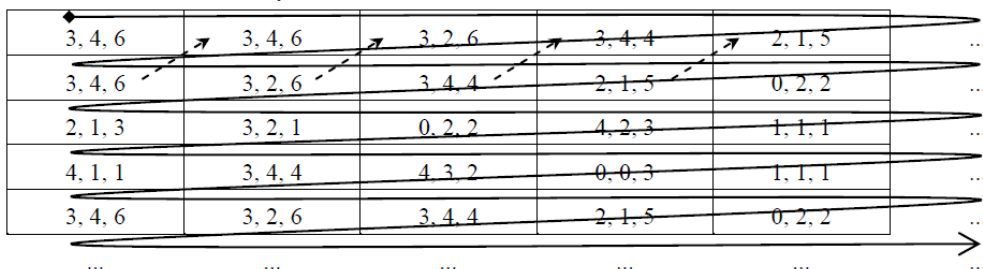
Серед контекстно-залежних алгоритмів у форматах графічних файлів найчастіше використовується словниковий алгоритм LZ77, оскільки він забезпечує найшвидше декодування [7, С. 82]. Описуючи словникові алгоритми, фіксовану кількість попередніх закодованих неподільних елементів (літералів) вхідного потоку називають *словником*, а наступних незакодованих – *буфером*. 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.

Алгоритм LZ77 (в контексті формату словникової компресії DEFLATE [8]) базується на заміні в процесі кодування 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.

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.

На сьогодні більшість зображень зберігаються в колірній моделі RGB з точністю дискретизації 8 бітів. Тобто колір кожного пікселя задається трьома байтами, які послідовно містять яскравості його червоної, зеленої та синьої компоненти. Черговість послідовного обходу яскравостей компонентів пікселів в цій колірній моделі схематично показана на рис. 1.



3, 4, 6	3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	...
3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	0, 2, 2	...
2, 1, 3	3, 2, 1	0, 2, 2	4, 2, 3	1, 1, 1	...
4, 1, 1	3, 4, 4	4, 3, 2	0, 0, 3	1, 1, 1	...
3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	0, 2, 2	...
...

Рис. 1. Яскравості компонентів пікселів верхнього лівого кута умовного RGB-зображення та черговість їх послідовного обходу (позначена суцільною стрілкою)

Саме за цією черговою формують вхідний потік для стиснення зображень алгоритмом LZ77 в процесі послідовного обходу. Наприклад, під час послідовного обходу яскравостей компонентів перших чотирьох пікселів з рис. 1 формують потік 3, 4, 6, 3, 4, 6, 3, 2, 6, 3, 4, 4. Цей потік в закодованому алгоритмом LZ77 вигляді запишеться як 3, 4, 6, <4, 3>, 2, <3, 6>, 4.

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 (на рис. 1 позначені пунктирними стрілками) 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 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.

Послідовність обходу пікселів для реалізації прогресуючого ієрархічного стиснення зображень без втрат та його вплив на ефективність 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 було 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 у верхньому лівому куті, по рядках зверху вниз, а у кожному рядку – підряд зліва направо з кроком $h_1 = 2^k$, де

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

height – кількість рядків, *width* – кількість стовпців пікселів зображення (рис. 2, а). 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 ($l = \overline{2, k+1}$) 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. 2, 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. 2, в). На рис. 2 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 опрацьовані раніше і тому не опрацьовуються на черговому проході шару, виділені курсивом.

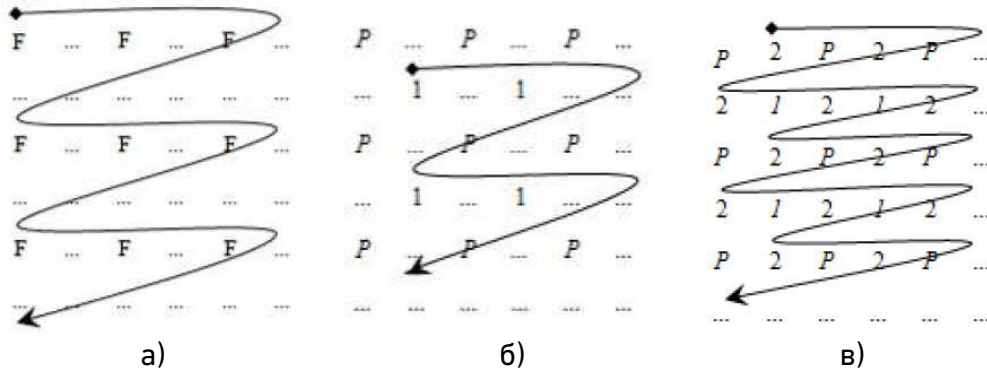


Рис. 2. Черговість обходу пікселів в процесі прогресуючого ієрархічного опрацювання: а) пікселів першого шару; б) пікселів першого проходу чергового шару; в) пікселів другого проходу чергового шару

Послідовно розміщені дані всіх проходів формують вхідний потік для кодування. Дані наступного шару збільшують кількість опрацьованих пікселів приблизно в 4 рази. Тому, як вже наголошувалося раніше, зупиняти процес декодування, використовуючи цю схему одходу, можливо вже після заповнення області виводу, не опрацьовуючи коди до кінця. Але 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].

З іншого боку, обхід пікселів зображення в декілька проходів збільшує коефіцієнти стиснення алгоритму LZ77 відносно послідовного обходу, адже в процесі прогресуючого обходу для незакодованих яскравостей кожного чергового пікселя буфера відповідні яскравості суміжних пікселів або ще не будуть входити в словник, або будуть мати в ньому великі зміщення, які кодуються значною кількістю бітів. Зокрема, на першому шарі пікселі обходяться з кроком $h_1 = 2^k$ (див. рис. 2, а), отже яскравості суміжних пікселів взагалі не будуть входити в словник. Існування в зображенні великих однакових фрагментів, віддалених саме на цей крок, мало ймовірно, тому на першому шарі алгоритм LZ77 неефективний. З кожним наступним шаром після другого кроку обходу пікселів зменшується вдвічі і тому ймовірність появи однакових послідовностей в буфері та словнику зростає, але для чергового пікселя буфера яскравості суміжних пікселів зображення або будуть опрацьовані на наступних шарах, і тому не входять в словник, або вже опрацьовані на попередніх шарах чи проходах, а отже мають в словнику великі зміщення.

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. 3), а для забезпечення однозначності декодування коди зміщень в словнику збільшимо на 6.

На перших проходах чотири найближчі суміжні опрацьовані раніше пікселі віддалені діагонально (див. рис. 3, а). Це пікселі з попередніх шарів (див. рис. 2, б) і тому 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 найближчі опрацьовані пікселі того самого проходу відповідно по горизонталі та по вертикалі, але в словнику зміщення найближчих пікселів по вертикалі збільшене на кількість пікселів в рядку проходу.

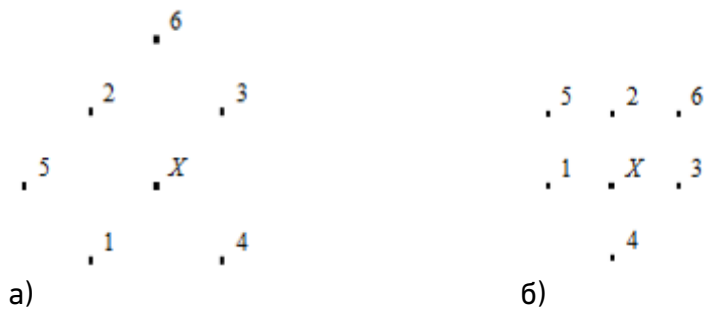


Рис. 3. Коди зміщень до яскравостей суміжних опрацьованих раніше пікселів на шарах, починаючи з другого: а) для першого проходу; б) для другого проходу

На других проходах чотири найближчих суміжних пікселі віддалені по горизонталі та по вертикалі (див. рис. 3, б). Це пікселі з попередніх шарів чи проходу (див. рис. 2, в). Вони розміщені ближче до чергового пікселя, ніж на першому проході, і тому від них починається більше однакових послідовностей. Пошук однакових послідовностей, починаючи з найближчих опрацьованих раніше пікселів попередніх шарів, дає змогу навіть віднаходити співпадаючі послідовності, які взагалі розкидані по різних шарах в словнику.

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 відносно пікселя *X* для зміщень 5 та 6 не кодуються, оскільки на черговому проході вони ще не опрацьовані.

Найістотніше пошук однакових послідовностей з найближчих опрацьованих раніше пікселів зменшує КС на останньому шарі, оскільки ці пікселі містяться в зображенні поряд з черговими пікселями *X* буфера і мають з ними найвищий рівень кореляції. Наприклад, під час першого проходу останнього шару для яскравостей компонентів пікселів зображення з рис. 1 однакова послідовність серед найближчих пікселів (рис. 4) буде віднайдена за зміщенням 3 (згідно кодів рис. 3, а). В словнику ця однакова послідовність взагалі розкидана по різних місцях. Пікселі на цьому проході обходяться з кроком 2 як в зображенні, так і в однаковій послідовності. Загальна кількість співпадаючих компонентів на рис. 4 рівна 6. Суміжні пікселі зліва від заштрихованих теж однакові, але в процесі прогресуючого ієрархічного стиснення вони обходяться на попередніх шарах, і тому не входять в

цю заміну, хоча під час послідовного обходу вони ввійшли б в одну заміну довжини 12 компонентів (на рис. 1 вона позначена пунктирними стрілками).

3, 4, 6	3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	...
3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	0, 2, 2	...
2, 1, 3	3, 2, 1	0, 2, 2	4, 2, 3	1, 1, 1	...
4, 1, 1	3, 4, 4	4, 3, 2	0, 0, 3	1, 1, 1	...
3, 4, 6	3, 2, 6	3, 4, 4	2, 1, 5	0, 2, 2	...
...

Рис. 4. Однакова послідовність довжиною 6 компонентів за зміщенням 3 (заштрихована вертикально) для яскравостей пікселів першого проходу останнього шару (заштрихована горизонтально) умовного RGB-зображення

Отже, додатковий пошук однакових послідовностей, починаючи з найближчих опрацьованих раніше пікселів, або віднаходить такі послідовності, які взагалі розкидані по словнику і тим самим збільшує ймовірність знаходження цих послідовностей, або використовує менші зміщення, ніж під час пошуку лише по словнику, і тому зменшує КС зображень.

Результати експериментів. Проаналізуємо результати застосування запропонованої модифікації алгоритму словникової компресії LZ77 (табл. 1–3) на прикладі прогресуючого ієрархічного стиснення зображень стандартного тестового набору Archive Comparison Test (надалі АСТ). Цей набір містить як синтезовані (№№ 1 (з шумами), 2, 7), так і фотореалістичні (всі інші) зображення. Завантажити TIFF-версії цих зображень можна, наприклад, з <http://www.compression.ca/act/act-files.html> чи з <http://www.compression.ru/arctest/act/act-tif.htm>. Тестування проводилося на комп'ютері з процесором Intel Pentium 4 з тактовою частотою 3 GHz та RAM 4 Gb. В табл. 1-3 ефективність застосування алгоритму LZ77 з стисненням по словнику в процесі послідовного обходу пікселів показана на прикладі утиліти MinPNG [10], яка на сьогодні забезпечує мінімальні КС у форматі PNG.

Як і очікувалося, застосування алгоритму LZ77 суттєво зменшує КС переважно штучних дискретно-тонових зображень, оскільки вони містять багато однакових послідовностей пікселів. Застосування класичного алгоритму LZ77 в процесі послідовного обходу зменшує КС зображень набору АСТ в середньому на 0.55 brb (два перших ря-

дки табл. 1, по дискретно-тонових зображеннях таке зменшення становить 1.03 brb), сповільнює кодування в 5.83 рази, оскільки виконує пошук однакових послідовностей в словнику, але прискорює декодування на 16%, адже дає змогу отримати менші файли.

Таблиця 1

Ентропія яскравостей компонентів пікселів зображень набору АСТ після застосування різних варіантів алгоритму LZ77, brb

Варіант алгоритму LZ77	№ файла								Середня ентропія
	1	2	3	4	5	6	7	8	
MinPNG без використання LZ77	2.20	1.84	4.84	4.21	4.32	5.51	1.75	4.64	3.66
MinPNG з LZ77 по словнику	1.67	0.50	4.82	4.16	4.31	5.16	0.52	4.61	3.21
Ієрархічне стиснення без використання LZ77	2.75	1.70	4.69	3.98	4.18	5.42	1.60	4.41	3.59
Ієрархічне стиснення з LZ77 по словнику	2.52	0.87	4.67	3.94	4.16	5.25	0.93	4.40	3.34
Ієрархічне стиснення з LZ77 по найближчих опрацьованих пікселях	1.37	0.68	4.66	3.82	4.15	5.32	0.68	4.33	3.13
Ієрархічне стиснення з LZ77 по найближчих пікселях та по словнику	1.34	0.59	4.65	3.82	4.15	5.16	0.63	4.33	3.08

Таблиця 2

Час кодування зображень набору АСТ різними варіантами алгоритму LZ77, с

Варіант алгоритму LZ77	№ файла								Середній час
	1	2	3	4	5	6	7	8	
MinPNG без використання LZ77	1.81	2.37	0.56	0.95	0.56	1.04	0.93	0.98	1.15
MinPNG з LZ77 по словнику	7.07	16.16	3.12	6.86	3.89	4.43	6.57	5.47	6.70
Ієрархічне стиснення без використання LZ77	1.85	2.42	0.60	0.97	0.58	0.97	0.96	0.96	1.16
Ієрархічне стиснення з LZ77 по словнику	2.57	3.55	0.94	1.32	0.87	1.42	1.52	1.31	1.69
Ієрархічне стиснення з LZ77 по найближчих опрацьованих пікселях	2.03	2.73	1.01	1.35	1.00	1.49	1.21	1.36	1.52
Ієрархічне стиснення з LZ77 по найближчих пікселях та по словнику	2.36	3.71	1.16	1.7.5	1.57	1.77	1.56	1.69	1.95

Таблиця 3

Час декодування зображень набору АСТ, стиснутих різними варіантами алгоритму LZ77, с

Варіант алгоритму LZ77	К-ть шарів	№ файла								Середній час
		1	2	3	4	5	6	7	8	
MinPNG без використання LZ77	4, всі	0.59	1.34	0.31	0.56	0.30	0.68	0.39	0.34	0.56
MinPNG з LZ77 по словнику	4, всі	0.55	0.70	0.34	0.50	0.36	0.52	0.27	0.51	0.47
Ієрархічне стиснення без використання LZ77	4	0.09	0.03	0.02	0.04	0.03	0.03	0.05	0.05	0.04
	всі	0.81	1.43	0.32	0.49	0.40	0.48	0.50	0.59	0.63
Ієрархічне стиснення з LZ77 по словнику	4	0.06	0.02	0.02	0.03	0.02	0.04	0.02	0.04	0.03
	всі	0.71	1.16	0.30	0.46	0.42	0.49	0.33	0.56	0.56
Ієрархічне стиснення з LZ77 по найближчих пікселях	4	0.06	0.01	0.02	0.04	0.02	0.02	0.04	0.04	0.03
	всі	0.59	1.25	0.30	0.47	0.44	0.44	0.42	0.56	0.56
Ієрархічне стиснення з LZ77 по найближчих пікселях та по словнику	4	0.06	0.02	0.02	0.04	0.02	0.03	0.04	0.02	0.03
	всі	0.58	1.21	0.30	0.54	0.32	0.49	0.49	0.42	0.54

Цей самий класичний алгоритм в процесі прогресуючого ієрархічного стиснення зменшує КС в середньому лише на 0.25 brb (третій та четвертий рядки табл. 3) і сповільнює кодування на 45.7%, але прискорює декодування на 11%.

Пропонована у цій статті модифікація алгоритму LZ77 з пошуком однакових послідовностей лише серед найближчих раніше опрацьованих елементів в процесі ієрархічного стиснення дає кращі результати: після її застосування КС в середньому зменшився на 0.46 brb, кодування сповільнилося на 31%, а декодування прискорилося на тих самих 11%. Але таке покращення спостерігається не на всіх зображеннях. Наприклад, для зображення № 6 менші КС забезпечує алгоритм LZ77 з пошуком однакових послідовностей у словнику, оскільки воно містить багато однакових віддалених фрагментів.

На завершення проаналізуємо результати виконання пошуку послідовностей однакових елементів для буфера як серед найближчих опрацьованих раніше елементів, так і в словнику (останній рядок в табл. 1–3). По набору АСТ в середньому КС додатково зменшився, але лише на 0.05 brb. Зменшення КС від одночасного застосування цих двох підходів на величину сукупного зменшення від кожного з них не відбулося, адже більшість найдовших однакових послідовностей, які починаються з найближчих опрацьованих раніше елементів, також міститься і в словнику. Але у словнику такі однакові послі-

довності мають більші зміщеннями, і тому при сукупному використанні кодуються зміщеннями до найближчих опрацьованих елементів. Застосування дослідженої модифікації алгоритму LZ77 дало змогу додатково зменшити КС дискретно-тонових зображень набору АСТ в середньому на 0.59 brb (четвертий та шостий рядки табл. 1).

І головне: прогресуючий ієрархічний спосіб обходу пікселів дає змогу виконати декодування значно швидше від послідовного обходу, коли розміри області виводу значно менші розмірів зображень (наприклад, для заповнення області виводу 128 x 128 пікселів (4 шари) таке декодування виконується в середньому у 16–18 разів швидше, див. табл. 3), оскільки час ієрархічного декодування залежить від меншого серед розмірів області виводу та розмірів зображення (про це наголошувалося ще в [3, С. 175]), а послідовного – лише від розмірів зображення.

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, оскільки такі зображення містять багато однакових суміжних послідовностей яскравостей компонентів пікселів. По набору АСТ таке зменшення КС в середньому перевищило 1 brb.

Надалі, з метою додаткового зменшення розмірів файлів стиснутих зображень без втрат і прискорення декодування в процесі прогресуючого ієрархічного стиснення, нами планується дослідити

ефективність застосування різницевих колірних моделей [11] до окремих фрагментів зображень.

1. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms, Third Edition. Kiyv : Dialektika, 2020. Vol. 1. 648 p. 2. Миано Дж. Форматы и алгоритмы сжатия изображений в действии : учеб. пособ. М. : Триумф, 2003. 336 с. 3. Сэлмон Д. Сжатие данных, изображений и звука. М. : Техносфера, 2006. 368 с. 4. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии : учеб. пособ. М. : Триумф, 2003. 320 с. 5. Shportko A., Postolatii V. Development of Predictors to Increase the Efficiency of Progressive Hierarchic Context-Independent Compression of Images Without Losses. *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021)* (Kharkiv, 22–23 apr. 2021). Vol. I. ceur-ws.org. P. 1026–1038. 6. Ziv J., Lempel A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*. May 1977. Vol. 23(3). P. 337–343. 7. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М. : ДИАЛОГ-МИФИ, 2003. 384 с. 8. Deutsch P. DEFLATE Compressed Data Format Specification version 1.3. RFC 1951. Alladin enterprises, 1996. 15 p. 9. Прэтт Э. Цифровая обработка изображений / пер. с англ. М. : Мир, 1982. Кн. 2. 480 с. 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 звернення: 26.12.2021). 11. Шпортко О. В. Використання різницевих кольорових моделей для стиснення RGB-зображень без втрат. *Відбір і обробка інформації*. 2009. № 31 (107). С. 90–97.

REFERENCES:

1. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms, Third Edition. Kiyv : Dialektika, 2020. Vol. 1. 648 p. 2. Miano Dj. Formaty i algoritmy sjatiya izobrajeniy v deystvii : ucheb. posob. M. : Triumph, 2003. 336 s. 3. Selomon D. Sjatie dannyih, izobrajeniy i zvuka. M. : Tehnosfera, 2006. 368 s. 4. Uelstid S. Fraktaly i veyvlety dlya sjatiya izobrajeniy v deystvii : ucheb. posob. M. : Triumph, 2003. 320 s. 5. Shportko A., Postolatii V. Development of Predictors to Increase the Efficiency of Progressive Hierarchic Context-Independent Compression of Images Without Losses. *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021)* (Kharkiv, 22–23 apr. 2021). Vol. I. ceur-ws.org. P. 1026–1038. 6. Ziv J., Lempel A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*. May 1977. Vol. 23(3). P. 337–343. 7. Vatolin D., Ratushnyak A., Smirnov M., Yukin V. Metody sjatiya dannyih.

Ustroystvo arhivatorov, sjatie izobrajeniy i video. M. : DIALOG-MIFI, 2003. 384 s. **8.** Deutsch P. DEFLATE Compressed Data Format Specification version 1.3. RFC 1951. Alladin enterprises, 1996. 15 p. **9.** Prett E. TSifrovaya obrabotka izobrajeniy / per. s angl. M. : Mir, 1982. Kn. 2. 480 s. **10.** MinPNG 1.0 – utylita dlia minimizatsii rozmiru failiv zobrazhen u formati PNG (True Color). URL: <http://apserver.org.ua/peregl.php?d=view&tid=131> (data zvernennia: 26.12.2021). **11.** Shportko O. V. Vykorystannia riznytsevykh kolorovykh modelei dlia stysnennia RGB-zobrazhen bez vtrat. *Vidbir i obrobka informatsii*. 2009. № 31 (107). S. 90–97.

Shportko A. V., Candidate of Engineering (Ph.D.), Associate Professor (Academician Stepan Demianchuk International University of Economics Humanities, Rivne), **Shportko L. V., Lecturer** (Vocational College of Economics and Business, Rivne), **Bomba A. Ya., Doctor of Engineering, Professor** (National University of Water and Environmental Engineering, Rivne)

IMPROVING THE EFFICIENCY OF VOCABULARY COMPRESSION METHODS FOR PROGRESSIVE HIERARCHICAL COMPRESSION OF IMAGES WITHOUT LOSSES

The expediency and the method of layer-by-layer pixel travers for the implementation of progressive hierarchical compression of images without losses are substantiated. To bypass the images, a scheme is proposed, according to which the first layer processes at least 16 pixels on each of the axes (if any) with the same step on both rows and columns. In the next layers, the pixels are processed in two passes: the first one sequentially processes those contained at the intersection of diagonals of squares with vertices in adjacent pixels of previous layers, and the second – sequentially bypasses equally spaced pixels between adjacent pixels of the previous layers and first pass pixels. The input stream for encoding is formed from sequentially placed data of all layers and passes.

Three main typical stages of lossless image compression are described. The principles of input stream compression by the context-dependent algorithm LZ77 on the second of these stages are considered. A modification of the LZ77 dictionary compression algorithm is proposed. It allows to increase the efficiency of progressive hierar-

chical compression by additional search for identical sequences not only in the dictionary, but also starting from the nearest previously processed pixels. Additional search for identical sequences either finds sequences that are generally scattered in the dictionary and thus increases the probability of finding them or uses smaller offsets than during search only in the dictionary, and therefore reduces the compression ratio of images.

The results of the usage of the proposed algorithm for image compression of the ACT set are presented. It is shown that the application of the above modification of the LZ77 algorithm allows, for example, to further reduce the compression coefficient of discrete-tone images of this set by an average of 0.59 bpb.

According to the results of the study, the following conclusions were made:

1. In the new versions of graphic formats and new lossless image compression formats, it is advisable to implement progressive hierarchical compression, thus it can significantly speed up decoding when the size of the output area is smaller than the image size.

2. Reducing the size of images compressed in a progressive hierarchical way is mostly achieved on the last layers, because the pixels used by both the dictionary algorithm and predictors have the highest level of correlation with the predicted pixel comparing with the previous layers.

3. An increase of the efficiency of the classical algorithm LZ77 in the process of progressive hierarchical compression of lossless images is possible by additional search for the same sequences starting from the nearest previously processed pixels.

4. Vocabulary algorithms reduce the compression coefficient of primarily artificial discrete-tone images significantly, because such images contain many identical adjacent sequences of brightness of the pixel components (for example, of the ACT set compression coefficient reduction is more than 1 bpb).

***Keywords:* progressive image compression; lossless compression; dictionary compression methods; modifications of the LZ77 algorithm.**

Шпортько А. В., к.т.н., доцент (ЧВУЗ «Международный экономико-гуманитарный университет имени академика Степана Демьянчука», г. Ровно, ITShportko@ukr.net), **Шпортько Л. В., преподаватель** (Ровенский профессиональный колледж экономики и бизнеса, г. Ровно, LChportko@gmail.com), **Бомба А. Я., д.т.н., профессор** (Национальный университет водного хозяйства и природопользования, г. Ровно, abomba@ukr.net)

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ СЛОВАРНЫХ МЕТОДОВ КОМПРЕССИИ ДЛЯ ПРОГРЕССИРУЮЩЕГО ИЕРАРХИЧЕСКОГО СЖАТИЯ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРЬ

Обоснована целесообразность и приведен способ обхода пикселей для реализации прогрессирующего иерархического сжатия изображений без потерь. Предложена модификация алгоритма словарной компрессии LZ77 для повышения эффективности такого сжатия посредством дополнительного поиска одинаковых последовательностей по ближайшим ранее обработанным пикселям. Приведены результаты применения предложенного алгоритма для сжатия изображений набора АСТ. Показано, что применение предложенной модификации алгоритма LZ77 позволяет, например, дополнительно уменьшить коэффициенты сжатия дискретно-тоновых изображений этого набора в среднем на 0.59 bpb.

Ключевые слова: прогрессирующее сжатие изображений; сжатие без потерь; словарные способы компрессии; модификации метода LZ77.
