УДК 004.04

О.В. Шпортько
Рівненський державний гуманітарний університет,
кафедра економічної кібернетики

АНАЛІЗ ЕФЕКТИВНОСТІ МОДИФІКАЦІЙ АЛГОРИТМІВ ГРАФІЧНОГО ФОРМАТУ PNG

Описано принципи алгоритмів графічного формату PNG для виконання попередніх перетворень та стиснення без втрат даних зображень. Досліджено вплив різних варіантів комбінацій модифікацій цих алгоритмів на ефективність компресії. Сумісне застосування всіх розглянутих модифікацій дає змогу, наприклад, додатково зменшити коефіцієнти стиснення зображень набору ACT у форматі PNG в середньому на 8.26 %.

Ключові слова: безвтратне стиснення зображень, динамічні коди Хафмана, формат графічних файлів PNG.

Principles of algorithms of graphic format PNG are described for implementation of previous transformations and compression without losses of images. The influence of different variants of combinations of modifications of these algorithms on efficiency of compression was explored. As experiment shows, applications of all considered modifications allow additional decrease of aspects of the compression of the set ACT in the format of PNG on average by 8.26 %, for example.

Key words: lossless compression of images, dynamic Huffman's codes, format of graphic files of PNG.

Вступ

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 стиснення зображень не втрачає своєї актуальності протягом останніх десятиліть і, ймовірно, не втратить у найближчому майбутньому.

Сьогодні растровий графічний формат PNG [1] є одним з найефективніших та найуживаніших для збереження зображень без втрат. В мережі Інтернет, наприклад, нараховується більше 18 млн. сторінок, що містять зображення у цьому форматі, щороку кількість таких сторінок збільшується на понад 1 млн. Цьому сприяють насамперед прийнятні коефіцієнти стиснення (відношення розмірів стиснутого до нестиснутого файлів зображення, надалі – КС) та висока швидкість декодування. Вдосконаленням формату PNG дозволить додатково покращити ці показники. Саме тому дослідження ефективності окремих модифікацій даного формату та їх комбінацій **є на сьогодні актуальним завданням**.

Аналіз останніх досліджень і публікацій. Шляхи підвищення ефективності стиснення у форматі PNG

Стиснення зображень у форматі PNG (як і компресія без втрат у більшості випадків [1, с. 642]) складається з двох послідовних достатньо незалежних операцій: переходу до альтернативного подання зображення (відображення), під час якого зменшують *міжелементну* надлишковість, та поелементного кодування отриманих даних для ліквідації *кодової* надлишковості. Для зменшення міжелементної надлишковості у цьому форматі застосовуються предиктори [3], які збільшують кодову надлишковість, та словниковий алгоритм LZ77 [4], а для

ліквідації кодової надлишковості – коди Хафмана [5]. Розглянемо принципи обробки даних зображень у форматі PNG детальніше.

Контекстно-залежний алгоритм LZ77. Описуючи словникові алгоритми, фіксовану кількість попередніх закодованих неподільних елементів (літералів) вхідного потоку називають *словником*, наступних незакодованих – *буфером*, а їх поєднання – *ковзаючим вікном*. Алгоритм LZ77 базується на заміні у вихідному потоці послідовності чергових літералів буфера посиланням на аналогічну послідовність літералів, що починається у словнику, у вигляді пари чисел <*довжина; зміщення від закінчення словника*>. У випадку відсутності аналогічної послідовності літералів у словнику, 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. Наприклад, потік кодів байтів пікселів *3 4 6 0 3 4 6 2 3 5 1* в закодованому вигляді можна записати так: *3 4 6 0 <3; 4> 2 3 5 1.* Очевидно, що для досягнення стиснення довжина збіжної послідовності має перевищувати 2. Оскільки більші зміщення кодуються, як правило, більшою кількістю бітів, то збіжні послідовності шукаються у словнику з кінця справа наліво.

Під час декодування кодів, отриманих за алгоритмом 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.

Контекстно-незалежне кодування Хафмана. Ідея використання динамічних контекстно-незалежних кодів Хафмана, що найчастіше застосовуються після виконання алгоритму LZ77 для стиснення літералів і базових значень довжин, а також для відокремленого стиснення базових значень зміщень, полягає у заміні чисел з більшою частотою (тут і надалі – абсолютною) кодами меншої кількості бітів, ніж для чисел з меншою частотою. Коди Хафмана для літералів/довжин та зміщень визначаються для кожного блоку стиснутих даних окремо, що сприяє покращенню стиснення.

Згідно з формулою Шеннона, елемент s_i з ймовірністю появи $p(s_i)$ найвигідніше кодувати $-\log_2 p(s_i)$ бітами. Тоді середня довжина коду елемента після застосування контекстно-незалежного алгоритму має наближатися до *ентропії джерела* [2]. Ентропія джерела зменшується із збільшенням нерівномірності розподілу ймовірностей між елементами. Середня довжина коду Хафмана збігається з ентропією джерела лише тоді, коли для всіх елементів s_i довжини їх оптимальних кодів $-\log_2 p(s_i)$ цілі. Чим більше $-\log_2 p(s_i)$ відхиляються від цілих чисел, тим більшою стає різниця між середньою довжиною коду Хафмана і ентропією джерела.

Різновиди предикторів формату PNG. Зменшити ентропію в процесі обробки зображень у форматі PNG намагаються за допомогою предикторів. *Предиктор* – це функція, що прогнозує (моделює) значення чергового елемента, використовуючи значення відомих суміжних елементів. Якщо піксел зображення характеризується декількома компонентами (наприклад, R, G, B), то предиктор кожної компоненти прогнозує значення згідно із відповідними компонентами суміжних пікселів. У процесі використання цієї технології обчислюють і надалі кодують відхилення чергової компоненти від прогнозованого предиктором значення. Тому у загальному випадку процес застосування предикторів до кожної компоненти піксела у вузлі (i, j) можна записати формулою

$$\Delta_{ij} = C_{ij} - predict_{ij} \ , \qquad (1)$$

де C_{ij} – значення компоненти до застосування предиктора, Δ_{ij} – значення компоненти після застосування предиктора, $predict_{ij}$ – значення предиктора, обчисленого для обраної компоненти.

У стиснутих блоках формату PNG даним кожного рядка передує окремий байт, що визначає предиктор, який застосовується до компонентів усіх його пікселів [1].

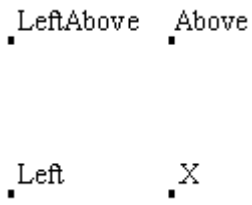

Рис. 1. Схема розміщення суміжних елементів для чергового елемента X

Для зручності опису предикторів цього формату введемо позначення яскравостей суміжних елементів для елемента *X* згідно схеми рис. 1. Тоді предиктори формату PNG мовою С записуються так:

```
char LeftPredict(char Left, char Above, char LeftAbove)
 {return Left;}
char AbovePredict(char Left, char Above, char LeftAbove)
 {return Above;}
char AveragePredict(char Left, char Above, char LeftAbove)
 {return (Left+Above)/2;}
char PaethPredict(char Left, char Above, char LeftAbove)
 {int pp=Left+Above-LeftAbove;
  int pa,pb,pc;
  pa=abs(pp-Left);
  pb=abs(pp-Above);
  pc=abs(pp-LeftAbove);
  if (pa<=pb && pa<=pc) return Left;
  else if (pb<=pc) return Above;
       else return LeftAbove;}
```

Оскільки суміжні пікселі зображення мають, як правило, близькі кольори і тому близькі значення відповідних компонентів, то часто значення предиктора збігатиметься зі значенням чергового елемента, найчастіше буде близьким до цього значення і рідко значно відрізнятиметься від нього. Тобто більшість відхилень яскравостей пікселів від значень, прогнозованих предикторами, будуть близькими до 0. Такий перерозподіл частот значень (а, отже, і ймовірностей) значно підвищує нерівномірність розподілу і тому зменшує ентропію джерела, отже, і довжину закодованої послідовності. Тобто, предиктори хоча й не виконують безпосереднє стиснення даних, але зменшують КС, насамперед контекстно-незалежного кодування Хафмана.

Для підвищення ефективності стиснення зображень без втрат сьогодні розроблено модифікації формату PNG, які орієнтовані переважно на один з трьох його базових алгоритмів: з метою оптимізації застосування предикторів найчастіше використовують два методи: формування та переходу до різницевих кольорових моделей [6] і коригування значення предиктора [7]; коефіцієнти стиснення алгоритму LZ77 зменшують, використовуючи альтернативний алгоритм LZPR [3] в сукупності з алгоритмом мінімізації розміру стиснутих блоків [8]; показники ж компресії кодування Хафмана покращують за допомогою використання палітри [9]. У цій же статті ми дослідимо взаємовплив зазначених модифікацій формату PNG, що й є **метою дослідження.** Серед зазначених модифікацій наведемо принципи виконання розроблених нами алгоритмів LZPR та переходу до різницевих кольорових моделей, оскільки вони найсуттєвіше зменшують КС зображень.

Формування розкладу LZPR з використанням декількох ковзаючих вікон

Розглянемо результати застосування двох різних предикторів до згаданого вище потоку *3 4 6 0 3 4 6 2 3 5 1*. При застосуванні предиктора *LeftPredict* згідно (1) отримаємо потік *3 1 2 -*

6 3 1 2 -4 1 2 -4. Якщо закодувати цей потік згідно із алгоритмом LZ77, то отримаємо коди *3 1 2 -6 <3;4> -4 <3;3>.*

Нехай над заданим потоком *3 4 6 0 3 4 6 2 3 5 1* у попередньому рядку міститься потік *2 5 4 0 1 3 5 2 1 4 0.* Тоді при застосуванні предиктора *AbovePredict* згідно (1) до заданого потоку отримаємо потік *1 -1 2 0 2 1 1 0 2 1 1.* Якщо закодувати цей потік згідно із алгоритмом LZ77, то отримаємо коди *1 -1 2 0 2 1 1 <4;4>*. Як видно з отриманих кодів, застосування цього предиктора зруйнувало повторення в середині потоку, зате створило нове в кінці. І це не дивно, адже предиктори *LeftPredict* та *AbovePredict* дозволяють отримати відповідно горизонтальні та вертикальні **прирости яскравостей** елементів, тобто описують **зміну форми** зображених об'єктів у різних напрямках. Аналогічно, інші предиктори дозволяють виділити інші властивості зображених об'єктів. Однакові значення властивостей об'єктів можуть повторюватися в зображенні в різних місцях, що дозволить підвищити ефективність використання алгоритму LZ77. З іншого боку, застосування предикторів значно підвищило частоти перетворених елементів біля нуля, тим самим збільшивши нерівномірність розподілу, що дає змогу значно ефективніше використовувати контекстно-незалежні алгоритми стиснення.

Як зазначалося вище, формат PNG передбачає використання єдиного предиктора для цілого рядка. Але ж для різних фрагментів рядка оптимальними можуть виявитися різні предиктори. Тому в [3] нами запропоновано алгоритм LZPR (Ziv - Lempel predictor), який використовує оптимальні статичні предиктори для різних фрагментів рядка. Це дозволяє ефективніше використовувати переваги контекстно-залежного словникового алгоритму та формувати послідовності з меншою ентропією джерела перед застосуванням контекстно-незалежного ентропійного алгоритму.

Опишемо механізм дії алгоритму LZPR на основі "жадібного" [10] розкладу вхідної послідовності. Нехай послідовність елементів потоку $s_1 \ldots s_{j-1}$ вже закодована. На черговому кроці виконується пошук збіжної послідовності максимальної довжини не лише для незакодованих елементів буфера $s_j^0 s_{j+1}^0 \ldots$ у словнику $s_i^0 s_{i+1}^0 \ldots$ ($i<j$), але й для всіх елементів буферів $s_j^k s_{j+1}^k \ldots$ у відповідних словниках $s_i^k s_{i+1}^k \ldots$, де k вказує на номер застосованого предиктора (вхідна послідовність вважається результатом дії предиктора *NonePredict*). Тобто пошук збіжних послідовностей ведеться не лише серед елементів вхідного потоку, але й серед результатів дії всіх предикторів. Оптимальним для позиції j вважається предиктор l, що дозволяє віднайти збіжну послідовність для елементів буфера $s_j^l s_{j+1}^l \ldots$ максимальної довжини *len*. **Тобто оптимальним вважається предиктор, що дозволяє закодувати послідовність найбільшої довжини.** При виявленні збіжної послідовності $s_j \ldots s_{j+len-1}$ кодується трійкою чисел <*довжина; зміщення від закінчення закодованої частини потоку; номер застосованого предиктора*>, тобто <*len; j-i; l*>, і кодування продовжується з позиції *j+len.* Якщо жоден з предикторів не дозволяє віднайти збіжну послідовність, то в закодовані дані заноситься елемент s_j^m, де m – номер предиктора, що дозволяє забезпечити загалом найменшу ентропію джерела (для формату PNG – це *PaethPredict*), і кодування продовжується з позиції $j+1$.

Якщо декілька предикторів дозволяють закодувати послідовність максимальної довжини *len*, то за алгоритмом LZPR обирається той з них, який породжує найменше зміщення. Якщо ж породжені зміщення теж виявляються однакові, то обирається предиктор, простіший для розрахунку.

Застосуємо запропонований алгоритм LZPR до згаданого потоку *3 4 6 0 3 4 6 2 3 5 1* (див. рис. 2). Як і слід було чекати, кількість віднайдених збіжних послідовностей виявилася максимальною, що дає змогу використати переваги алгоритму LZ77. Крім того, зменшилася ентропія окремих закодованих символів, що дає змогу в подальшому ефективніше використати контекстно-незалежний алгоритм.

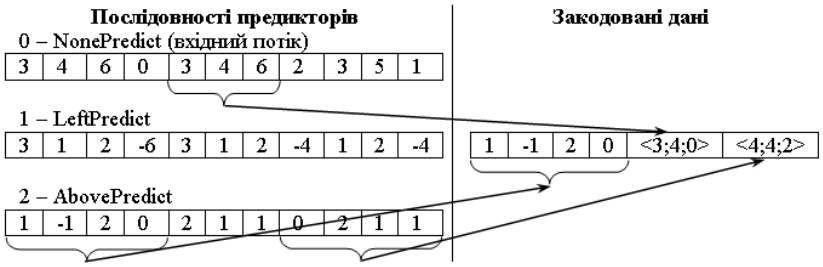

Рис. 2. Застосування алгоритму LZPR до потоку 3 4 6 0 3 4 6 2 3 5 1

Алгоритм формування та переходу до різницевих кольорових моделей

Як відомо, фіксовані альтернативні кольорові моделі використовуються в програмному забезпеченні для стиснення не перший рік (наприклад, в архіваторі WinRAR стиснення RGB зображень відбувається після переходу до моделі R-G, G, B-G). Але такі моделі формуються статично і не залежать від перепадів яскравостей пікселів окремого зображення. Тому нагадаємо адаптивний алгоритм [6] для формування та переходу до альтернативних різницевих кольорових моделей з метою покращення КС RGB-зображень за рахунок зменшення ентропії.

Виведемо спочатку формулу для розрахунку довжини ентропійного коду довільної послідовності елементів. Нехай кожне з значень елементів v_i зустрічається N_i разів у послідовності завдовжки $N = \sum_i N_i$. Тоді $p(v_i) = N_i / N$ і загальна довжина ентропійного коду, становитиме

$$L = N \times H = N \log_2(N) - \sum_i N_i \log_2(N_i). \qquad (2)$$

Замінювати компоненту кольорової моделі лінійною комбінацією з іншою компонентою доцільно лише тоді, коли після застосування обраного предиктора довжина ентропійного коду комбінації буде меншою за довжину ентропійного коду компоненти. Для забезпечення однозначності декодування в зображенні можна виконати **максимум дві** заміни значень різних компонентів різницями з іншими компонентами по всіх пікселах. Відповідно виникає задача **вибору під час попередньої обробки зображень серед можливих шести таких замін** (значень R_{ij} різницями R_{ij}-$k_{RG}G_{ij}$ або R_{ij}-$k_{RB}B_{ij}$, значень G_{ij} різницями G_{ij}-$k_{GR}R_{ij}$ або G_{ij}-$k_{GB}B_{ij}$ та значень B_{ij} різницями B_{ij}-$k_{BR}R_{ij}$ або B_{ij}-$k_{BG}G_{ij}$) **тих двох, які максимально зменшать довжину ентропійного коду результатів дії обраного предиктора**.

Розглянемо алгоритм формування різницевих кольорових моделей лише для k_{nm}=1, адже параметри таких моделей з цілими коефіцієнтами розраховуються найшвидше, оскільки не вимагають виконання операцій з дійсними числами та заокруглень. Ці ж причини дають змогу суттєво прискорити декодування. У цьому випадку необхідно оцінити доцільність заміни значень компоненти R_{ij} різницями R_{ij}-G_{ij} або R_{ij}-B_{ij}, значень G_{ij} різницями G_{ij}-R_{ij} або G_{ij}-B_{ij} та значень B_{ij} різницями B_{ij}-R_{ij} або B_{ij}-G_{ij}.

Запишемо досліджувані ентропійні довжини у вигляді матриці аналізу A:

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} L(\Delta(R)) & L(\Delta(R)-\Delta(G)) & L(\Delta(R)-\Delta(B)) \\ L(\Delta(G)-\Delta(R)) & L(\Delta(G)) & L(\Delta(G)-\Delta(B)) \\ L(\Delta(B)-\Delta(R)) & L(\Delta(B)-\Delta(G)) & L(\Delta(B)) \end{pmatrix}. \qquad (3)$$

Враховуючи (2), неважко пересвідчитись, що $a_{mn} = a_{nm}$, тобто для формування цієї матриці необхідно обчислити значення лише шести елементів. Нехай у межах кожного піксела компонента R має індекс 0, G – 1, B – 2. Бачимо, що різницева кольорова модель визначається максимум двома недіагональними елементами різних рядків матриці A, які серед елементів, менших за діагональні елементи своїх рядків, сумарно найбільше від них відхиляються (забезпечують максимальні

зменшення ентропійних довжин кодів). У випадку наявності таких елементів індекс рядка кожного з них визначає зменшувану компоненту, а індекс стовпця – компоненту, що від неї віднімається. Наприклад, вибір елемента a_{12} вказує, що в альтернативній кольоровій моделі для кожного піксела зображення необхідно значення компоненти *G* зменшити на значення компоненти *B*.

Покроково алгоритм формування та переходу до різницевої кольорової моделі з цілими коефіцієнтами перед стисненням зображення з використанням предикторів записується так:

1. Розрахувати для кожної компоненти всіх пікселів зображення результати дії обраного лінійного предиктора;

2. Обчислити ентропійні довжини кодів компонентів і різниць компонентів результатів дії обраного предиктора та зберегти їх в матриці *A* згідно з (3);

3. Занести значення 0 у змінні *index11*, *index12*, *index21* та *index22*, що визначають можливі різниці альтернативної кольорової моделі;

4. Визначити в матриці *A* два елементи, які не належать головній діагоналі, не симетричні відносно неї, менші за діагональні елементи своїх рядків, і сумарно найбільше від них відхиляються. Якщо такі елементи присутні, то занести в змінні *index11* та *index12* відповідно номер рядка та номер стовпця першого елемента, а в змінні *index21* та *index22* – номер рядка та номер стовпця другого елемента. Інакше визначити в матриці *A* недіагональний елемент, менший за діагональний елемент свого рядка. Якщо і такий елемент відсутній, то перейти до кроку 8. Інакше занести в змінну *index11* номер рядка цього елемента, а в змінну *index12* – номер його стовпця;

5. Якщо значення змінних *index11* та *index22* однакові, то переставити значення змінних *index11* з *index21* та *index12* з *index22*, тобто змінити черговість віднімання компонентів кольорової моделі;

6. Зменшити для кожного піксела зображення значення компоненти з індексом *index11* на значення компоненти з індексом *index12*;

7. Якщо значення змінних *index21* та *index22* різні, тобто друга різниця кольорової моделі визначена, то зменшити для кожного піксела зображення значення компоненти з індексом *index21* на значення компоненти з індексом *index22*;

8. Завершити формування та перехід до різницевої кольорової моделі і виконати стиснення перетвореного зображення з використанням предикторів.

Аналіз ефективності взаємовпливу модифікацій формату PNG

Проаналізуємо ефективність застосування згаданих модифікацій формату PNG та їх комбінацій для стиснення восьми файлів стандартного тестового набору ACT, що містить як синтезовані (№№ 1, 2, 7) так і фотореалістичні (№№ 3-6, 8) зображення. Завантажити їх можна, наприклад, з web-сторінки http://www.compression.ru/arctest/act/act-tif.htm. Результати тестування на комп'ютері з процесором AMD-K6, 300МГц, 128 Mb RAM наведено у табл. 1-3 для семи варіантів модифікацій даного формату чи їх комбінацій:

1. Попередній аналіз зображень та майже оптимальний розклад LZ77 без модифікацій формату PNG;

2. "Жадібний" розклад LZPR;

3. "Жадібний" розклад LZPR після застосування різницевих кольорових моделей;

4. "Жадібний" розклад LZPR після коригувань значень предиктора;

5. "Жадібний" розклад LZPR після застосування різницевих кольорових моделей та коригувань значень предиктора (у разі доцільності їх використання);

6. "Жадібний" розклад LZPR після застосування різницевих кольорових моделей, коригувань значень предиктора (у разі доцільності їх використання) та палітрування;

7. Майже оптимальний розклад LZPR після застосування різницевих кольорових моделей, коригувань значень предиктора (у разі доцільності їх використання) та палітрування.

Таблиця 1

Коефіцієнти стиснення файлів зображень набору ACT після застосування різних варіантів модифікацій формату PNG чи їх комбінацій, %

№ варіанта	Номер файла з набору ACT								Середній КС
	1	2	3	4	5	6	7	8	
1	20.85	6.24	60.21	51.60	53.71	64.61	6.56	57.33	40.14
2	20.13	5.69	60.47	47.27	51.24	58.63	6.01	54.73	38.02
3	20.13	5.69	58.78	42.15	46.42	51.34	6.01	47.18	34.71
4	20.42	5.72	58.13	46.75	49.80	58.54	6.01	53.51	37.36
5	20.13	5.69	56.18	40.85	43.95	50.39	6.01	45.10	33.54
6	17.42	5.66	55.53	38.77	42.65	46.40	6.08	43.97	32.06
7	17.37	5.25	55.53	38.42	42.52	46.40	5.67	43.89	31.88

Таблиця 2

Час кодування файлів зображень набору ACT програмами для різних варіантів модифікацій формату PNG чи їх комбінацій, с

№ варіанта	Номер файла з набору ACT								Середній час
	1	2	3	4	5	6	7	8	
1	67.40	166.42	25.48	64.26	35.21	37.79	66.73	49.60	64.11
2	11.10	22.74	7.86	17.24	10.27	15.77	9.67	14.44	13.64
3	12.08	24.94	8.24	17.74	10.55	17.09	10.49	15.05	14.52
4	11.69	23.95	8.13	17.68	10.55	16.21	10.21	14.88	14.16
5	13.96	28.01	8.89	18.73	11.21	18.18	11.81	16.04	15.85
6	24.06	28.45	16.43	27.19	18.95	25.59	11.97	26.47	22.39
7	79.26	249.47	15.71	36.96	20.54	27.52	99.25	28.84	69.70

Таблиця 3

Час декодування файлів зображень набору ACT програмами для різних варіантів модифікацій формату PNG чи їх комбінацій, с

№ варіанта	Номер файла з набору ACT								Середній час
	1	2	3	4	5	6	7	8	
1	3.79	4.23	2.31	3.13	2.19	3.35	1.60	3.46	3.01
2	3.41	4.28	2.36	2.86	2.04	3.18	1.70	3.19	2.88
3	3.57	4.78	2.47	2.97	2.20	3.41	1.98	3.24	3.07
4	4.89	6.86	2.75	3.62	2.53	3.95	2.85	3.85	3.91
5	3.57	4.78	2.80	3.62	2.53	3.90	1.98	3.79	3.37
6, 7	3.24	4.73	2.47	3.40	2.36	3.51	1.97	3.52	3.15

Як свідчать дані цих таблиць, у середньому застосування для тестових зображень: контекстно-залежного алгоритму LZPR (варіант 2) у порівнянні з найкращим на сьогодні способом стиснення у стандарті PNG (варіант 1) зменшило КС на 2.12 %, прискорило кодування у 4.7 рази та декодування – на понад 4 %, що вказує на його ефективність; використання різницевих кольорових моделей (варіант 3) суттєво не вплинуло на КС синтезованих зображень та зменшило КС фотореалістичних зображень на 5.29 %, сповільнивши при цьому кодування на 6.5 % та декодування – на 7 %; коригування значень предиктора (варіант 4) підвищило КС синтезованих зображень на 0.11 %, зменшило КС фотореалістичних зображень на 1.12 %, уповільнило кодування на 2 % та декодування – аж на 36 %; застосування ж палітри (варіант 6) зменшило КС зображень на 1.48 % і, хоча й уповільнило кодування на 41 %, але прискорило декодування на 7 %. Застосування майже оптимального розкладу LZPR (варіант 7) замість “жадібного” дає змогу додатково зменшити

КС у середньому на 0.18 %, але сповільнює кодування більш ніж утричі, тому цей розклад доцільно застосовувати лише в процесі збереження остаточних варіантів зображень.

З іншого боку, як показали експерименти, застосування методів формування та переходу до різницевих кольорових моделей і коригування значень предикторів ефективне не для всіх RGB-зображень [11]. Доцільність застосування першого з цих методів регламентується алгоритмом його реалізації [6]. Другий метод ми пропонуємо використовувати лише тоді, коли він зменшує прогнозовану ентропійну довжину коду зображення. Такий додатковий аналіз доцільності застосування методу коригування значення предиктора (варіант 5) хоча й у середньому сповільнює кодування ще на 2.3 %, але забезпечує незростання КС для всіх зображень та прискорює декодування на 25 %. Додаткове зменшення КС фотореалістичних зображень внаслідок сукупного застосування двох методів підвищення ефективності використання предикторів на 0.76 % пояснюється зменшенням енергії по двох перетворених компонентах різницевої кольорової моделі, що дає змогу ефективніше використовувати для них метод коригування значення предиктора.

Як свідчать експерименти, підвищити КС зображень в трикомпонентних кольорових моделях можна не лише за рахунок декореляції даних окремих компонентів, а й за допомогою міжкомпонентної декореляції. Таку декореляцію слід виконувати так, щоб підсилити прояви властивостей зображення, що використовуються алгоритмами препроцесингу та безпосереднього стиснення обраного графічного формату. У випадку використання неадаптивних предикторів для стиснення зображень без втрат виконувати міжкомпонентну декореляцію доцільно за допомогою різницевих кольорових моделей. Ефективність застосування різницевих кольорових моделей для таких зображень майже завжди підвищується зі зменшенням частки унікальних кольорів. Формування різницевих кольорових моделей відображає процес пошуку кольорів, стосовно яких енергія приростів яскравостей є мінімальною. На практиці доцільно використовувати різницеві кольорові моделі з цілими коефіцієнтами, оскільки такі кольорові моделі хоча й не гарантують досягнення мінімальних КС для всіх зображень, проте забезпечують максимальну швидкість кодування і, головне, декодування, тому їх доцільно впровадити в наступні версії форматів, що використовують неадаптивні предиктори та не виконують міжкомпонентну декореляцію (PNG, JPEG-LS, BMF та інші), на рівні стандартів.

Комплексне застосування розроблених модифікацій формату PNG сумісно з алгоритмами коригування значень предиктора та мінімізації розміру стиснутих блоків дає змогу досягнути в середньому найменших сьогодні КС без втрат в основному за рахунок фотореалістичних зображень з високим рівнем унікальних кольорів, суттєво не впливаючи на час декодування, оскільки дані модифікації орієнтовані на підвищення ефективності різних способів кодування цього формату: різницеві кольорові моделі дають змогу зменшити ентропію після застосування предикторів, алгоритм LZPR вдосконалює механізм дії алгоритму LZ77, а палітрування використовується для покращення показників компресії HUFF. Саме це створює передумови для успішного сумісного застосування досліджених модифікацій в інших графічних форматах та архіваторах.

Висновки

1. Метод коригування значення предиктора варто застосовувати для стиснення резервних копій зображень у складі архіваторів (а не у графічних форматах) після алгоритму переходу до різницевих кольорових моделей, якщо таке використання зменшує прогнозовану ентропійну довжину коду, оскільки цей метод суттєво сповільнює декодування, хоча й зменшує КС фотореалістичних зображень.

2. Алгоритми переходу до різницевих кольорових моделей, LZPR та палітрування доцільно впровадити в наступні версії стандарту формату PNG, оскільки вони суттєво зменшують КС зображень, кардинально не впливаючи при цьому на час декодування.

1. Boutell T. PNG Specification. Version 1.0 / Boutell T., et. all // RFC 2083, Boutell. Com, inc. – March 1997. – 102 p. 2. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с. 3. Шпортько О. В. Оптимізація використання статичних предикторів

у процесі стиснення зображень без втрат / О. В. Шпортько // Відбір і обробка інформації. – 2008. – № 28 (104). – С. 82–89. 4. Ziv J. A universal algorithm for sequential data compression / Ziv J., Lempel A. // IEEE Transactions on Information Theory. – May 1977. – Vol. 23(3). – P. 337–343. 5. Huffman D. A Method for the Construction of Minimum Redundancy Codes / D. Huffman // Proceedings of the IRE. – Vol. 40(9). – P. 1098–1101. 6. Шпортько О. В. Використання різницевих кольорових моделей для стиснення RGB-зображень без втрат / О. В. Шпортько // Відбір і обробка інформації. – 2009. – № 31 (107). – С. 90–97. 7. Weinberger M. J. From LOCO-I to the JPEG-LS Standard / M. J. Weinberger, Seroussi G. // Hewlett-Packard Laboratories, Palo Alto, HPL-1999-3. – Jan 1999. – 19 p. 8. Шпортько О. В. Вибір найкоротшого з альтернативних стиснутих блоків динамічних кодів Хафмана у форматі PNG / О. В. Шпортько // Комп'ютинг. – 2009. – Т. 8, вип. 2. – С. 58–67. 9. Шпортько О. В. Використання палітри для групового статистичного кодування RGB-зображень без втрат / О. В. Шпортько // Відбір і обробка інформації. – 2009. – № 30 (106). – С. 125–132. 10. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : ДИАЛОГ-МИФИ, 2003. – С. 17–106. 11. Шпортько О. В. Аналіз взаємовпливу модифікацій формату PNG / О. В. Шпортько, Л. В. Шпортько // Волинський математичний вісник. Зб. наукових праць. - Рівне: РДГУ, 2012. – № 9 (18). – С. 182–188. – (Серія: Прикладна математика).