Рівненський державний гуманітарний університет

ВОЛИНСЬКИЙ МАТЕМАТИЧНИЙ ВІСНИК

СЕРІЯ ПРИКЛАДНА МАТЕМАТИКА

Збірник наукових праць

Випуск 11 (20)

Рівне-2014

“Волинський математичний вісник. Серія прикладна математика” публікує результати досліджень з математичного моделювання і обчислювальних методів та суміжної проблематики в галузі математики, інформатики, механіки. Розрахований на наукових працівників, викладачів ВНЗ, аспірантів та студентів старших курсів.

“Волынский математический вестник. Серия прикладная математика”.
The “Volyn Mathematical Bulletin. Applied Mathematics Series”.

Редакційна колегія

Барановський С.В.
Бейко І.В.
Бомба А.Я. *(головний редактор)*
Булавацький В.М.
Власюк А.П.
Войтович М.М.
Гаращенко Ф.Г.
Гарбарчук В.І.
Джунь Й.В.
Каштан С.С.
Климюк Ю.Є. *(технічний секретар)*
Кратко М.І.
Кузьменко А.П.
Кундрат М.М.
Ляшенко І.М.
Мартинюк П.М. *(науковий секретар)*
Недашковський М.О.
Новаковскі А.
Новіков О.М.
Петрівський Я.Б.
Пригорницький Д.О.
Присяжнюк І.М.
Савула Я.Г.
Свідзинський А.В.
Скопецький В.В. *(консультант)*
Сяський А.О.
Турбал Ю.В.
Чикрій А.О.
Янчук П.С.
Ярощак С.В. *(науковий секретар)*

Видається у Рівненському державному гуманітарному університеті при сприянні Інституту кібернетики ім. В. М. Глушкова НАН України, Інституту прикладних проблем механіки і математики ім. Я. С. Підстригача НАН України, навчальних закладів та наукових товариств Волинського регіону. Друкується за ухвалою Вченої ради РДГУ (протокол № 10 від 28 травня 2015 р.).

Адреса редакції: 33028, Україна, м. Рівне, вул. Остафова, 31,
Рівненський державний гуманітарний університет,
кафедра інформатики та прикладної математики, редакція ВМВ.
Тел.: +380362260444. E-mail: vmvspm@ukr.net.

ISBN 966-7281-02-8
© РДГУ, 2014

У цьому випуску друкуються статті, підготовлені на основі матеріалів Всеукраїнської наукової конференції «Сучасні проблеми математичного моделювання та обчислювальних методів» (19 – 22 лютого 2015 року, Рівне).

ОРГАНІЗАТОРИ

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Національний університет водного господарства та природокористування
Міжнародний економіко-гуманітарний університет ім. акад. С. Дем'янчука
Харківський національний університет ім. В.Н. Каразіна
Київський національний університет ім. Т. Шевченка
Інститут кібернетики ім. В.М. Глушкова НАН України
Інститут прикладних проблем механіки і математики ім. Я.С. Підстригача НАН України
Наукове товариство ім. Т. Шевченка
Інститут механіки і прикладної інформатики
Університет Казимира Великого в Бидгощі (Польща)

ПРОГРАМНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

- Сергієнко І.В. – співголова
- Кушнір Р.М. – співголова
- Ляшко С.І. – співголова
- Бомба А.Я. – заступник голови
- Власюк А.П.
- Базилевич Р.П.
- Буй Д.Б.
- Булавацький В.М.
- Воробель Р.А.
- Гандель Ю.В.
- Гаращенко Ф.Г.
- Грищенко О.Ю.
- Кузьменко А.П.
- Куссуль Н.М.
- Мартинюк П.М.
- Міщенко В.О.
- Наконечний О.Г.
- Турбал Ю.В.
- Чапля Є.Я.
- Чикрій А.О.
- Цєшко М.
- Кубік Ю.

Зміст

ПАМ`ЯТІ АКАДЕМІКА В. С. МИХАЛЕВИЧА *6*

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

Бомба А. Я., Крока Л. Л. *Числові методи квазіконформних відображень розв'язання задач ідентифікації коефіцієнта електричної провідності за даними томографії прикладених потенціалів* *24*

Вовк О. М. *Термопружний стан термочутливих півпросторів за дії миттєвого джерела тепла на межі контакту* *34*

Герус А. В., Мартинюк П. М., Чеховська Н. В. *Чисельне розв'язання одновимірної задачі взаємозв'язаних процесів фільтраційної консолідації та контактної суфозії ґрунтів* *45*

Журавчак Л. М., Крук О. С. *Математичне моделювання процесів теплопровідності у кусково-однорідних середовищах з нелінійною поведінкою матеріалів зон* *55*

Каплун А. В., Крива Н. Р., Цимбалюк Л. І. *Двоперіодична задача про пластичне відшаровування жорсткого волокна ромбічного перерізу* *68*

Кот В. В. *Математичне моделювання контактної взаємодії ортотропної пластинки з круговим отвором і круглого диска через систему двох тонких пружних ребер* *76*

Присяжнюк О. В. *Сингулярно збурені процеси багатокомпонентного конвективно-дифузійного масопереносу в нанопористих середовищах з врахуванням терморежиму* *87*

Шупіков О. М., Сметанкіна Н. В., Угрімов С. В., Долгополова Н. В., Свєт Є. В. *Моделювання термопружного стану багатошарового оскління під впливом нестаціонарних температурних полів* *100*

Ярощак С. В. *Математичне моделювання процесу розробки родовищ нафти з використанням термо-гравітаційного дренажу* *117*

ОБЧИСЛЮВАЛЬНІ МЕТОДИ

Бричак А. А., Ванин В. А. *Применение метода конечных элемен-*

тов с разрывными базисными функциями к решению задач газовой динамики *129*

Грицько Б. Є. *Чисельно-аналітичний підхід до розв'язування задач теорії потенціалу в кусково-однорідних об'єктах з урахуванням умов неідеального контакту* *143*

Сапужак О. Я., Сироєжко О. В. *Використання методу граничних елементів у задачах електророзвідки постійним струмом* *156*

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Бахрушин В. Є., Ігнахіна М. О., Назаренко О. О. *Перевірка адекватності нелінійних регресійних моделей засобами мови R* *167*

Буй Д. Б., Пузікова А. В. *Аксіоматика багатозначних залежностей табличних баз даних: коректність, повнота, критерій повноти* *175*

Варламова Н. В., Гахов А. В., Мищенко В. О. *Успешность программной диверсности в разработке систем и персональные характеристики разработчика: концептуальная модель* *194*

Шпортько О. В. *Використання альтернативних стиснутих блоків для кожного блоку даних у форматі PNG* *213*

Jarynowski A. *Which activation function of cooperation describes human behavior?* *229*

УДК 004.043

Шпортько О. В.

ВИКОРИСТАННЯ ДВОХ АЛЬТЕРНАТИВНИХ СТИСНУТИХ БЛОКІВ ДЛЯ КОЖНОГО БЛОКУ ДАНИХ У ФОРМАТІ PNG

Анотація. *Пропонується алгоритм генерування двох альтернативних стиснутих блоків, ітеративного зменшення їх розмірів та вибору найкоротшого стиснутого блоку серед альтернативних для покращення компресії зображень у форматі PNG. Як показують експерименти, реалізація цього алгоритму дозволяє покращити коефіцієнт стиснення переважної більшості зображень на 2 – 6 %.*

Ключові слова: *безвтратне стиснення зображень, статичні предиктори, алгоритм LZ77, кодування Хафмана, растровий формат PNG.*

Вступ. Формат графічних файлів PNG [1] був створений 1 жовтня 1996 року для ефективного збереження растрових зображень без втрат після того, як компанія Unisys почала вимагати плату за використання формату GIF. На сьогодні дизайнери та розробники Web-сайтів найчастіше зберігають фотореалістичні зображення у форматі JPEG, а дискретнотонові – у форматі PNG. Крім цього, формат PNG найчастіше використовується для зберігання зображень, де втрати неприпустимі (наприклад, для зберігання рентгенівських знімків). Саме тому цей формат підтримується більшістю сучасних програм для перегляду і створення зображень. Проблема підвищення ефективності стиснення зображень у форматі PNG є актуальною на сьогодні і буде актуальною в найближчому майбутньому, оскільки навіть наближення до її розв'язання дозволяє зменшити розміри відповідних файлів, що, в свою чергу, дає змогу підвищити ефективність використання дискового простору та прискорити їх завантаження з мережі. У цій статті описується один зі способів часткового розв'язання зазначеної проблеми.

Принципи стиснення зображень у форматі PNG. Будь-яке стиснення даних можливе за рахунок зменшення надлишковостей. Чим більше

© О. В. Шпортько, 2015

видів надлишковостей виявляє і опрацьовує компресор і чим краще він ці надлишковості усуває – тим краще він зможе стиснути націлені на таку обробку дані. Формат PNG орієнтований на зменшення надлишковостей чотирьох типів: між сусідніми пікселами, що мають, як правило, близькі кольори; між однаковими фрагментами зображення; між однаковими змінами кольорів; між переважаючими кольорами пікселів зображення.

Піксели зображення записуються у PNG-файли найчастіше по рядках зверху вниз, а у кожному рядку – послідовно зліва направо (як символи у текстах), формуючи тим самим вхідний потік. Перед кодуванням ці піксели можуть бути попередньо оброблені предикторами, принцип дії яких розглядається наприкінці цього розділу. У PNG-файлах, що використовуються на сьогодні, стиснуті дані зберігаються у відокремлених блоках згідно формату словникового стиснення DEFLATE [2]. У блоках стиснутих даних містяться результати застосування до вхідного потоку алгоритму LZH [3], згідно з яким результати контекстно-залежного словникового алгоритму LZ77 [4] стискаються контекстно-незалежними кодами Хафмана [3].

Описуючи словникові алгоритми, фіксовану кількість попередніх закодованих елементів вхідного потоку називають словником, а наступних незакодованих – буфером. Алгоритм LZ77 базується на заміні у вихідному потоці послідовності чергових елементів буфера посиланням на аналогічну послідовність елементів словника у вигляді пари чисел *<довжина; зміщення від закінчення словника>*. У випадку відсутності аналогічної послідовності елементів у словнику, перший елемент буфера переноситься у вихідний потік без змін. Після цього закодовані елементи переносяться з початку буфера в кінець словника і кодування продовжується аналогічно аж до закінчення елементів вхідного потоку. Наприклад, потік кодів біт пікселів *36 38 35 35 36 38 35 36 36 38 35 38 36 38 35 35 28* в закодованому вигляді записується як *36 38 35 35 <3; 4> 36 <3; 4> 38 <4; 12> 28*. Алгоритм LZ77 використовується у форматі PNG насамперед для зменшення надлишковостей між однаковими фрагментами зображення.

Під час декодування кодів алгоритму LZ77 окремі елементи 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 елементів. Природно, що алгоритм декодування має розрізняти окремі елементи та групи *<довжина; зміщення>*. Згідно з алгоритмом LZH, у форматі DEFLATE з цією метою довжини замін та окремі елементи кодуються разом числами в межах [0; 285]. При цьому числа з діапазону [0; 255] відповідають кодам окремих елементів, 256 позначає закінчення блоку, а числа з діапазону [257; 285] вказують на базові значення довжин. Після базових значень довжин міститься визначена форматом кількість біт, що разом з базовим значенням однозначно визначає довжину заміни. Зміщення зберігається після відповідної довжини аналогічно – у вигляді базового значення та додаткових біт. Базове значення зміщення знаходиться в межах [0; 29]. У форматі DEFLATE максимальне значення довжини закодованої послідовності може сягати 285, а зміщення – 32768.

Ідея використання кодів Хафмана, що застосовуються для кодування окремих елементів і базових значень довжин та базових значень зміщень після виконання алгоритму LZ77, полягає у заміні чисел з більшою частотою кодами меншої кількості біт, ніж для чисел з меншою частотою. Коди Хафмана для елементів/довжин та зміщень визначаються для кожного блоку стиснутих даних окремо, що сприяє покращенню стиснення. Автономне кодування Хафмана в форматі PNG насамперед зменшує надлишковості між переважаючими кольорами пікселів зображення.

Згідно теореми Шенона, 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 величини

$$H = -\sum_i p(s_i) \times \log_2 p(s_i), \qquad (1)$$

яку ще називають *ентропією джерела* [3]. Ентропія джерела зменшується при збільшенні нерівномірності розподілу ймовірностей між елементами.

Зменшити ентропію джерела при обробці зображень у форматі PNG намагаються за допомогою предикторів. *Предиктор* – це функція, що прогнозує (моделює) значення чергового елемента, використовуючи значення відомих суміжних елементів. В процесі використанні 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

$$\Delta_{ij} = C_{ij} - predict_{ij}, \qquad (2)$$

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

Оскільки сусідні піксели зображення мають, як правило, близькі кольори і тому близькі значення відповідних компонентів, то часто значення предиктора буде співпадати зі значенням чергового елемента, найчастіше – буде близьким до цього значення і рідко значно відрізнятиметься від нього. Тобто більшість значень Δ_{ij} будуть близькими до 0. Такий перерозподіл частот значень (а, отже, і ймовірностей) значно підвищує нерівномірність розподілу [5] і тому зменшує ентропію джерела (згідно (1)), а, отже, і довжину закодованої послідовності. Таким чином, предиктори використовуються насамперед для зменшення надлишковостей між сусідніми пікселами зображення, що мають близькі кольори. Крім цього, застосування алгоритму LZ77 та кодування Хафмана до результатів дії предикторів дозволяє зменшити надлишковості між однаковими змінами кольорів.

У блоках стиснутих даних формату PNG даним кожного рядка передує окремий байт, що визначає предиктор, який застосовується до

компонент всіх його пікселів. На сьогодні форматом передбачено п'ять можливих значень цього байта [1], що визначає чотири різні предиктори: 0 – дані рядка не обробляються предикторами; 1 – предиктор рівний значенню відповідної компоненти зліва; 2 – предиктор рівний значенню компоненти зверху; 3 – предиктор рівний середньому арифметичному значень зліва та зверху; 4 – предиктор Піфа, що прогнозує значення у напрямку найменшого приросту. Опис предикторів формату PNG, міститься в [1], класифікація предикторів наведена в [5].

Алгоритм генерування, зменшення розмірів та вибору найкоротшого серед альтернативних стиснутих блоків. Для виявлення типів стиснутих блоків проаналізуємо різні варіанти розподілів частот їх елементів/довжин у форматі PNG на прикладі перших 64 Кб зображення Lena.bmp. Значення компонент пікселів цього блоку мають слабо виражену нерівномірність розподілу (рис. 1), адже найбільша частота серед окремих елементів рівна 1650 (2.5% від загальної кількості).

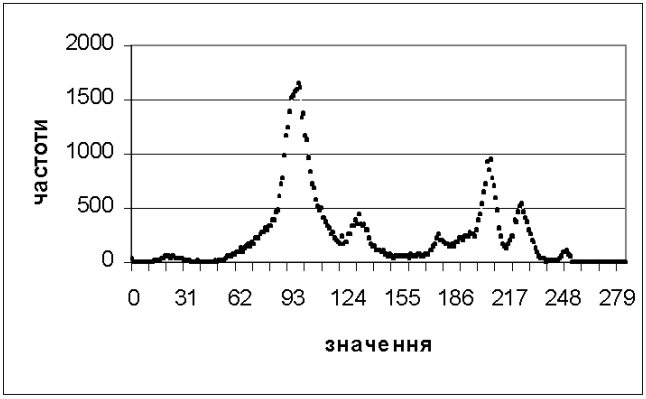

Рис. 1. Розподіл частот без застосування предикторів та замін

Ця нерівномірність значно посилюється після застосування предиктора Піфа (рис. 2), який дозволяє кардинально збільшити частоту нульового значення до 11405 (17.4%), збільшити частоти елементів навколо нуля до понад 4000 (6.1%, від'ємним результатам дії предикторів відповіда-

ють збільшені на 256 значення) та зменшити частоти всіх інших елементів.

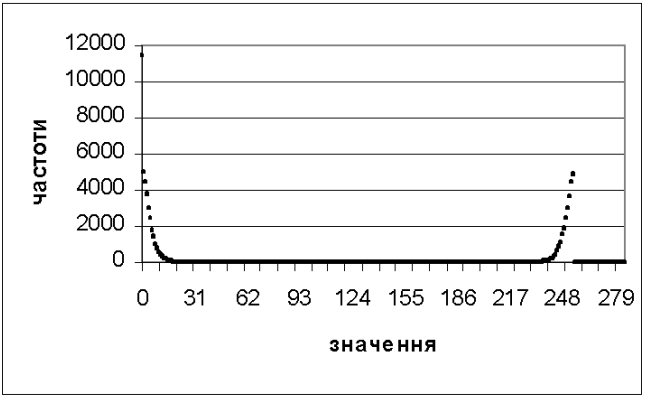

Рис. 2. Розподіл частот після застосування предиктора Піфа

Збільшує нерівномірність розподілу і застосування замін співпадаючих послідовностей (рис. 3).

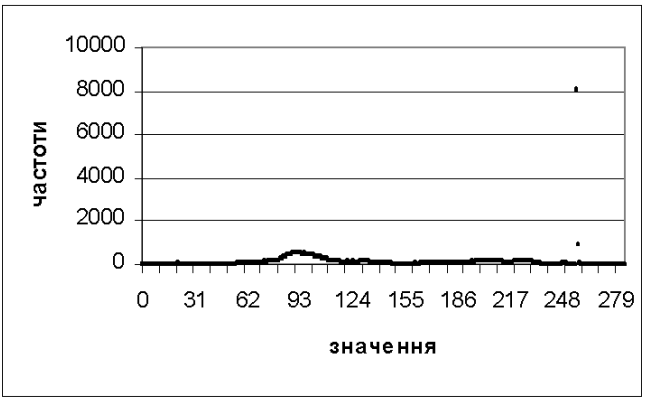

Рис. 3. Розподіл частот після застосування всіх замін LZ77

Але у фотореалістичних зображеннях, до яких належить і Lena.bmp, як правило, зустрічаються лише короткі повторення послідовностей елементів. Наприклад, у першому блоці цього зображення віднайдено 8016

повторень по 3 елементи (відповідає значенню 257) та 933 повторення по 4 елементи (відповідає значенню 258). При цьому короткі співпадаючі послідовності трапляються, як правило, серед елементів з найбільшою частотою, і тому застосування таких замін зменшує нерівномірність розподілу окремих елементів. Наприклад, частота елемента 96 зменшилася з 1650 до 513. Ще більше коротких співпадаючих послідовностей зустрічається у фрагментах зображення після застосування предикторів (рис. 4). Наприклад, після застосування предиктора Піфа кількість 3-елементних співпадаючих послідовностей зросла до 11066, а 4-елементних – до 3312. В той же час, зменшилася кількість найдовших співпадаючих послідовностей з 23 до 22, оскільки застосування предикторів реалізує вплив сусідніх елементів і здатне розбивати такі послідовності.

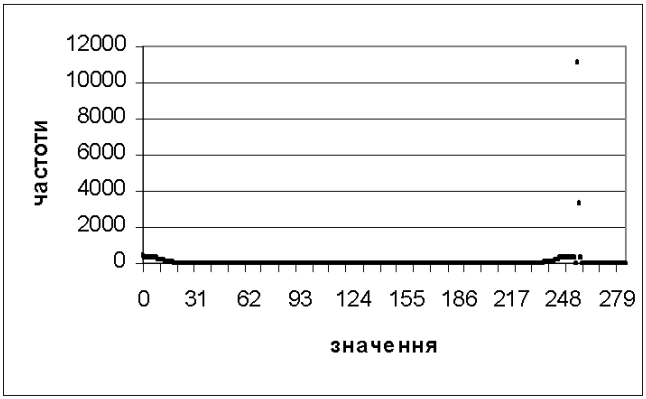

Рис. 4. Розподіл частот після застосування предиктора Піфа та всіх замін LZ77

Зовсім інші властивості мають розподіли елементів/довжин у випадку застосування лише довгих замін. Такі заміни, як правило, зменшують максимальні частоти елементів, але не порушують характеру нерівномірності розподілу. Наприклад, застосування замін від 9 елементів зменшило максимальну частоту розподілу з 1650 (рис. 1) до 1538 (рис. 5), а після застосування предиктора – з 11405 (рис. 2) до 5266

(рис. 6).

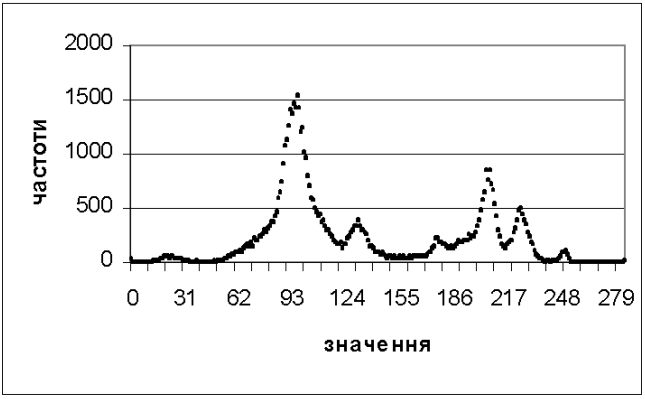

Рис. 5. Розподіл частот після застосування замін LZ77 від 9 елементів

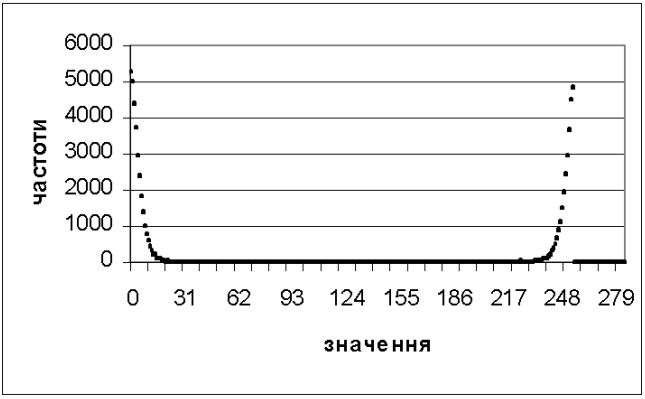

Рис. 6. Розподіл частот після застосування предиктора Піфа та замін LZ77 від 9 елементів

Заміни вважатимемо ефективними, якщо вони записуються не більшою кількість біт, ніж окремі елементи, які вони замінюють. Оскільки окремі елементи і базові значення довжин та зміщення у форматі DEFLATE записуються кодами Хафмана,. то заміна j довжини len_j, що виконується починаючи з елемента s_k за зміщення $offset_j$ ефективна лише тоді, коли

$$\sum_{i=0}^{len_j-1} l_{s_{k+i}} \geq l_{len_j} + d_{len_j} + \lambda_{offset_j} + \delta_{offset_j}, \qquad (3)$$

де l_m – довжина коду Хафмана елемента/довжини з кодом m, d_m – кількість додаткових біт для запису довжини заміни m, λ_m – довжина коду Хафмана зміщення з кодом m, δ_m – кількість додаткових біт для запису зміщення m. Але короткі заміни виявляються ефективними (рис. 3, 4), як правило, для стиснутих блоків, в яких інші короткі заміни теж враховуються (назвемо їх *орієнтованими на заміни*). Це пов'язано з тим, що короткі заміни суттєво зменшують частоти окремих елементів, а отже збільшують довжини їх кодів Хафмана. З іншого боку, короткі заміни виявляються, як правило, неефективними для стиснутих блоків, у яких інші короткі заміни теж не враховуються (рис. 5, 6), оскільки довгі заміни найчастіше не порушують характеру розподілу окремих елементів (назвемо такі блоки *орієнтованими на елементи*). Частоти окремих елементів у блоках, орієнтованих на елементи, завжди будуть не меншими частот цих же елементів у блоках, орієнтованих на заміни (рис. 7). Розподіли ж замін будуть розміщуватися навпаки.

Для окремого блоку даних коротшим може виявитися як стиснутий блок, орієнтований на заміни, так і стиснутий блок, орієнтований на елементи – в основному це залежить від характеру розподілу його окремих елементів: якщо блок даних має виражену нерівномірність, то для нього, як правило, коротшим виявляється блок стиснутих даних, орієнтований на елементи, при відсутності ж вираженої нерівномірності коротшим виявляється, як правило, блок, орієнтований на заміни. Крім цього, на довжини стиснутих блоків суттєво впливають зміщення між однаковими фрагментами блоку даних, адже більші зміщення кодуються більшою кількістю додаткових бітів.

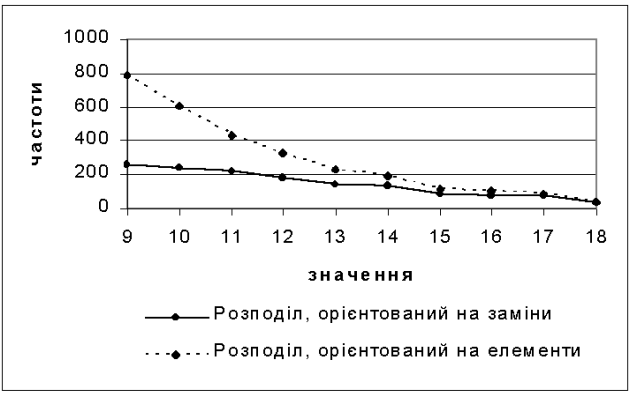

Рис. 7. Фрагмент альтернативних розподілів частот елементів перших 64 Кб зображення Lena.bmp

Для чергового блоку даних характер розподілу його елементів і, тим більше, зміщень наперед визначити неможливо, тому **для кожного блоку даних доцільно створити два альтернативних стиснутих блоки: орієнтований на заміни та орієнтований на елементи, зменшити їх довжини,** використовуючи (3), та **обрати найкоротший з них для зберігання даних** у форматі DEFLATE для PNG-файла (рис. 8).

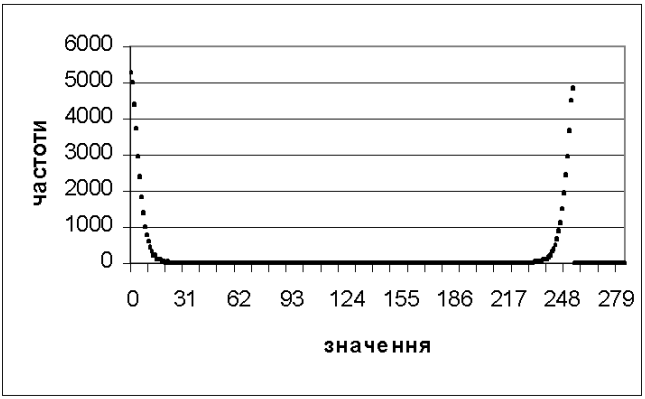

Рис. 8. Розподіл частот після застосування алгоритму вибору найкоротшого стиснутого блоку

У форматі DEFLATE мінімальна довжина кодів Хафмана рівна 1, максимальна – 15, максимальна кількість додаткових біт довжини складає 5, а зміщення – 13. Тому, згідно (3), для будь-яких розподілів ефективними будуть заміни довжиною від 48 елементів. Ефективність решти замін залежить не лише від їх елементів, а й загалом від розподілу стиснутого блоку.

Зменшити довжини стиснутих блоків, орієнтованих на заміни, можна за рахунок відкидання неефективних замін. Згідно (3), такими вважатимемо заміни, для яких

$$\sum_{i=0}^{len_j-1} l'_{s_{k+i}} < l'_{len_j} + d_{len_j} + \lambda'_{offset_j} + \delta_{offset_j}, \qquad (4)$$

де штрих над літерами вказує на розрахунок параметрів з блоку, орієнтованого на заміни.

Зменшити довжини стиснутих блоків, орієнтованих на елементи, можна за рахунок врахування ефективних замін. Згідно (3), такими вважатимемо заміни, для яких

$$\sum_{i=0}^{len_j-1} l''_{s_{k+i}} \geq l''_{len_j} + d_{len_j} + \lambda''_{offset_j} + \delta_{offset_j}, \qquad (5)$$

де два штрихи над літерами вказують на розрахунок параметрів з блоку, орієнтованого на елементи.

Заміни, ефективні в блоках, орієнтованих на елементи, виявляються ефективними і в блоках, орієнтованих на заміни. І навпаки: заміни, неефективні в блоках, орієнтованих на заміни, виявляються неефективними і в блоках, орієнтованих на елементи. Більше того, врахування ефективних замін в блоках, орієнтованих на елементи, зменшує частоти окремих елементів і, відповідно, може збільшити після перерахунку довжини їх кодів Хафмана. Тому серед замін, неефективних з попередніми кодами Хафмана, можуть виявитися ефективні з перерахованими такими кодами. І навпаки: вилучення неефективних замін в блоках, орієнтованих на заміни, збільшує частоти окремих елементів і, відповідно, може зменшити після перерахунку довжини їх кодів Хафмана,

тому серед замін, ефективних з попередніми кодами Хафмана, можуть виявитися неефективні з перерахованими такими кодами. Більшість замін виявляються, як правило, ефективними в блоках, орієнтованих на заміни і неефективними в блоках, орієнтованих на елементи. Після кожного перерахунку кодів розподіли частот альтернативних стиснутих блоків будуть зближуватися між собою.

Враховуючи те, що довжина стиснутого блоку складається з суми довжин закодованих елементів, довжин замін та зміщень, загальну довжину блоку, орієнтованого на заміни, будемо розраховувати за формулою

$$L' = \sum_{i=0}^{255} n_i' l_i' + l_{256}' + \sum_{i=257}^{285} n_i' \left(l_i' + d_i \right) + \sum_{i=0}^{29} \eta_i' \left(\lambda_i' + \delta_i \right), \qquad (6)$$

де n_i – частота елемента чи базової довжини *i*, η_i – частота базового зміщення *i*. Довжину блоку, орієнтованого на заміни, будемо розраховувати аналогічно:

$$L'' = \sum_{i=0}^{255} n_i'' l_i'' + l_{256}'' + \sum_{i=257}^{285} n_i'' \left(l_i'' + d_i \right) + \sum_{i=0}^{29} \eta_i'' \left(\lambda_i'' + \delta_i \right). \qquad (7)$$

Покроково алгоритм використання двох альтернативних стиснутих блоків для кожного блоку даних записується так:

1. Створити для блоку даних два альтернативних стиснутих блоки: блок, орієнтований на заміни – з застосуванням всіх можливих замін і блок, орієнтований на елементи – з застосуванням замін з довжиною від 48 елементів. Створити список замін, що ввійшли в перший і не ввійшли в другий стиснутий блок для подальшого аналізу. Визначити для елементів блоків довжини відповідних кодів Хафмана;
2. Вилучити неефективні заміни згідно (4) зі списку аналізованих та з блоку, орієнтованого на заміни;
3. Вилучити ефективні заміни згідно (5) зі списку аналізованих та врахувати їх в блоці, орієнтованому на елементи;

4. У випадку зменшення кількості замін для аналізу після виконання двох попередніх кроків перевизначити для елементів альтернативних блоків довжини кодів Хафмана та повернутися до пункту 2;
5. Обрати найкоротший стиснутий блок серед альтернативних згідно (6) та (7) і використати його для збереження стиснутих даних.

На практиці для прискорення виконання алгоритму обрати найкоротший стиснутий блок можна вже після першої ітерації і надалі зменшувати лише його розмір [6], але це ускладнює програмну реалізацію алгоритму.

Результати експериментів. Розглянемо на завершення результати застосування описаного алгоритму використання альтернативних блоків для стиснення восьми різнотипних 24-бітних зображень стандартного набору файлів ACT у форматі PNG (завантажити їх TIFF-версії можна, наприклад, з http: //compression.ca/act/act-files.html). Тестування проводилося за допомогою програми з CD [1], у яку були внесені такі модифікації: забезпечена можливість виходу з словника в буфер під час кодування повторів; реалізований вибір предиктора для рядка пікселів зображення на основі співставлення ефективності стиснення рядка без попередньої обробки з оцінкою нерівномірності розподілу після дії кожного предиктора; запроваджений аналіз кількостей додаткових біт при записі зміщень; розмір блоків даних збільшений до 64 Кб та відкинуті допоміжні текстові блоки.

Результати тестування наведено в табл. 1, 2, де показником компресії файлів обрано коефіцієнт стиснення, виражений у bpp. Дані тестування програми без застосування розглянутого алгоритму наведено в третьому, а з застосуванням – у четвертому стовпці. Крім цього, для порівняння ефективності стиснення у другому стовпці таблиць вказано результати тестування програми Microsoft Photo Editor 2000, яка не застосовує предиктори, а в п'ятому та шостому – результати популярної серед Web-дизайнерів програми OptiPng (http://www.optipng.sourceforge.net) відповідно з параметрами стандартного та максимального стиснення, яка генерує короткі PNG-файли за результатами перебору предикторів, розмірів стиснутих блоків та стратегій стиснення.

Таблиця 1

Коефіцієнти стиснення файлів зображень набору ACT у форматі PNG після застосування різних варіантів програм, bpp

Bmp-файл	Photo Editor 2000	Miano без алгор.	Miano з ал-гор.	OptiPng стан-дарт	OptiPng макс.
Clegg	9.06	5.69	5.03	5.37	5.32
Frymire	1.78	1.64	1.64	1.63	1.63
Lena	22.85	15.92	14.45	14.48	14.48
Monarch	18.36	13.18	12.61	12.51	12.49
Peppers	21.97	14.08	12.92	12.95	12.92
Sail	20.73	16.09	16.09	15.97	15.80
Serrano	1.84	1.72	1.72	1.70	1.70
Tulips	21.61	14.65	13.88	13.84	13.82
Середній КС	**14.77**	**10.37**	**9.79**	**9.81**	**9.77**

Таблиця 2

Час стиснення файлів зображень набору ACT у форматі PNG різними варіантами програм на комп'ютері з частотою 300 МГц, с

Bmp-файл	Photo Editor 2000	Miano без алгор.	Miano з алго-ритмом	OptiPng стан-дарт	OptiPng макс.
Clegg	3	18	20	72	996
Frymire	4	23	23	78	684
Lena	2	7	9	14	289
Monarch	3	13	16	33	733
Peppers	2	8	10	16	408
Sail	3	11	12	21	486
Serrano	2	9	9	23	336
Tulips	3	12	15	26	562
Середній час	**2.75**	**12.63**	**14.25**	**35.38**	**561.75**

Як свідчать результати тестування, застосування описаного алгоритму зменшило коефіцієнт стиснення на 2 – 6% для 63% зображень,

які, як правило, мають виражену нерівномірність розподілу, та не вплинуло на компресію решти файлів, хоча й сповільнило виконання програми в середньому на 13%. Крім цього, реалізація розглянутого алгоритму дозволила наблизитися до коефіцієнта стиснення програми повного перебору, а для деяких зображень і перевищити його, витрачаючи при цьому для компресії в 1.55 – 49 разів менше часу.

З наведених результатів дослідження зробимо такі висновки:

1. Для підвищення ефективності стиснення даних у форматах, що послідовно використовують алгоритми декількох методів, слід враховувати взаємний вплив цих алгоритмів;
2. Розглянутий алгоритм дозволяє суттєво зменшити коефіцієнт стиснення більшості зображень за рахунок використання орієнтованого на елементи та орієнтованого на заміни альтернативних стиснутих блоків для кожного блоку даних;
3. Описаний алгоритм не вимагає модифікації декодера чи програм перегляду зображень і за рахунок зменшення розмірів файлів лише прискорює їх роботу. Саме тому він може бути ефективно використаний для збереження даних у стандартах, що використовують формат стиснення DEFLATE.

1. *Boutell T.* PNG Specification. Version 1.0 / Boutell T., et. all // RFC 2083, Boutell. Com, inc. – Mar. 1997. – 102 p.
2. *Миано Дж.* Форматы и алгоритмы сжатия изображений в действии : Учеб. пособ. / Дж. Миано. – М.: Триумф, 2003. – 336 с., ил. – (Серія: Практика программирования).
3. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ-МИФИ, 2003. – 384 с.
4. *Ziv J.* A universal algorithm for sequential data compression / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. – May 1977. – Vol. 23(3). – P. 337-343.
5. *Бредихин Д. Ю.* Сжатие графики без потерь качества [Електронний ресурс] / Д. Ю. Бредихин. – 2004. – Режим доступу: http://www.compression.ru/download/articles/i_lless/bredikhin_2004_lossless_image_compression_doc.rar. – Назва з екрана.

6. *Шпортько О. В.* Оптимізація блоків стиснутих даних у графічному форматі PNG / О. В. Шпортько // Вісник Національного університету "Львівська політехніка". – 2009. – № 653. – С. 223-231. – (Серія: Інформаційні системи та мережі).

Рівненський державний гуманітарний університет, Рівне
E-mail: chportko@yandex.ru
chportko@ukr.net

Надійшла 30.01.2015

Шпортько А. В.

ИССПОЛЬЗОВАНИЕ ДВУХ АЛЬТЕРНИТИВНЫХ СЖАТЫХ БЛОКОВ ДЛЯ КАЖДОГО БЛОКА ДАННЫХ В ФОРМАТЕ PNG

Аннотация. *Предлагается алгоритм генерирования двух альтернативных сжатых блоков, итеративного уменьшения их размеров и выбора кратчайшего сжатого блока среди альтернативных для улучшения компрессии изображений в формате PNG. Как показывают эксперименты, реализация этого алгоритма позволяет улучшить коэффициент сжатия подавляющего большинства изображений на 2 - 6 %.*

Ключевые слова: *сжатие изображений без потерь, статические предикторы, алгоритм LZ77, кодирование Хаффмана, растровый формат PNG.*

Shportko A. V.

THE USE OF TWO ALTERNATIVE COMPRESSED BLOCKS FOR EVERY BLOCK OF DATA IN THE FORMAT OF PNG

Annotation. *The algorithm of generating of two alternative compressed blocks reducing of their sizes and selection of the shortest compressed block among alternative for the improvement of compression of the images in the format of PNG is offered. As experiments show, realization of this algorithm allows to improve an aspect of compression of the majority of images on 2 - 6 %.*

Keywords: *compression of images without losses, static predictors, algorithm of LZ77, Hafman's encoding, raster format of PNG.*

Наукове видання

ВОЛИНСЬКИЙ МАТЕМАТИЧНИЙ ВІСНИК

серія прикладна математика

Збірник наукових праць

Випуск 11 (20)

Відповідальний за випуск
Ярощак С.В.

Підписано до друку 10.10.2013 р.
Папір офсет. Формат 60/84 1/16.
Ум. друк. арк. 12,7. Тираж 100. Зам. № 146/1.

Редакційно-видавничий відділ
Рівненського державного гуманітарного університету
Україна, м. Рівне, 33028, вул. С. Бандери, 12