УДК 621.391

В.П. МАЙДАНЮК , О.В. КИРИЧЕНКО

УЩІЛЬНЕННЯ ДАНИХ БЕЗ ВТРАТ НА ОСНОВІ ПЕРЕТВОРЕНЬ

Вінницький національний технічний університет,
Хмельницьке шосе, 95, Вінниця, 21021, Україна,
тел.: (0432) 43-78-80, E-Mail: maydan2000@mail.ru

Анотація. Мета використання перетворень в ущільненні даних – перетворення потоку вхідних подій до вигляду, що дозволяє використовувати простіші і ефективніші моделі. Фактично, вони перетворюють одні види надмірності в інші, простіше модельовані. До таких перетворень відносять і перетворення BWT (Burrows-Wheeler Transform), яке розглядається в даній роботі.

Аннотация. Цель использования преобразований в сжатии данных – преобразование потока входных событий к виду, который позволяет использовать более простые и более эффективные модели. Фактически, они превращают одни виды избыточности в другие, проще моделируемые. До таких преобразований относят и преобразование BWT (Burrows-Wheeler Transform), которое рассматривается в данной работе.

Abstract. A purpose of the use of transformations in the compression of information is transformation of stream of events of entrances to the kind, that allows to use more simple and more effective models. Actually, they convert one types of surplus in other, simpler designed. Before such transformations take transformation of BWT (Burrows-Wheeler Transform), which is examined in this work.

Ключові слова: перетворення BWT, ущільнення даних.

ВСТУП

Ущільнення зображень з втратами включає використання методів ущільнення без втрат на останньому етапі, який і виконує власне ущільнення і від якого в значній мірі залежить загальний коефіцієнт ущільнення зображення [1]. Однак, з появою методу арифметичного кодування проблема генерації коду була фактично вирішена. З тих пір з метою підвищення коефіцієнту ущільнення основна увага стала приділятися питанням, пов'язаним з моделюванням. Нові підходи опираються на парадигму ущільнення за допомогою універсального моделювання і кодування, запропоновану Ріссаненом і Ленгдоном [2-4].

В світлі концепції універсального моделювання і кодування заслуговують на увагу методи ущільнення без втрат на основі перетворень. Мета використання перетворень в ущільненні даних – перетворення потоку вхідних подій до вигляду, що дозволяє використовувати простіші і ефективніші моделі. Фактично, вони перетворюють одні види надмірності в інші, простіше модельовані. Тобто, перетворення дозволяє представляти оброблювану інформацію в особливій формі, ідеально відповідній для подальшого ефективного кодування. Незвичність підходу полягає в наявності фактично двох етапів моделювання: перший етап – це робота перетворення, направлена на отримання «зручного» інформаційного представлення, а другий – побудова допоміжної моделі, на основі якої буде закодовано дане представлення [2].

До таких перетворень відносять перетворення MTF (Move To Front) та перетворення BWT (Burrows-Wheeler Transform) [3]. Однак, якщо MTF давно використовується при ущільненні як в якості перетворення так і в якості самостійного методу ущільнення, то по-перше перетворення BWT може використовуватись тільки в якості перетворення, а по-друге за рахунок використання перетворення BWT сумісно з MTF можна досягнути значних коефіцієнтів ущільнення, особливо високочастотних компонент зображення. Перетворення BWT застосовується для перетворення ланцюжкової надмірності в надмірність повторення подій. Спочатку вхідний потік подій циклічно зсувається на одну позицію і записується під початковим вхідним потоком стільки раз, скільки подій у вхідному потоці. Отримана

квадратна матриця сортується по рядках зліва направо. Доведено, що для відновлення початкового потоку подій достатньо останнього стовпця матриці (так званого префіксного стовпця) і номера рядка початкового потоку подій після сортування. Префіксний стовпець володіє великою надмірністю повторення подій і локальною надмірністю розподілу імовірності.

Однак, виконання зворотного перетворення BWT вимагає значних затрат пам'яті, особливо при великих об'ємах вхідного блоку даних. Для швидкого зворотного перетворення додатково до власне даних потрібний вектор зворотного перетворення, що є масивом чисел, розмір якого рівний числу символів в блоці. В роботі запропоновано новий алгоритм реалізації прямого і зворотного BWT перетворення, який ґрунтується на зберіганні в пам'яті лише чотирьох стовпців початкової матриці.

РОЗРОБКА АЛГОРИТМУ ВИКОНАННЯ ПРЯМОГО І ЗВОРОТНОГО BWT-ПЕРЕТВОРЕННЯ

Де-факто описувати BWT стало прийнято за допомогою прикладу перетворення рядка символів "абракадабра". Далі потрібно з рядка даних створити матрицю всіх можливих його циклічних перестановок. Першим рядком матриці буде початкова послідовність, другим рядком - вона ж, зсунута на один символ вліво, і т.д. Таким чином, отримаємо матрицю, зображену на рис. 1.

0 абракадабра
1 бракадабраа
2 ракадабрааб
3 акадабраабр
4 кадабраабра
5 адабраабрак
6 дабраабрака
7 абраабракад
8 браабракада
9 раабракадаб

Оскільки, дані поступають з файлу побайтно і якщо відомий розмір блока BWT-перетворення, то немає сенсу очікувати прийому всього блоку над яким виконується перетворення. Можна сформувати матрицю циклічних перестановок на етапі читання файлу. Прийнятий байт спочатку записується в порядку прийому в "0" рядок матриці (рис. 1), а потім записується в інші рядки матриці в позиції, які визначаються за наступним виразом:

Pos:=(rbwt –ja +ia) mod rbwt,

де rbwt – розмір блоку BWT – перетворення, ia – номер поточного стовпця, ja – номер рядка в який записується черговий байт.

Відсортуємо всі рядки даної матриці у відповідності з лексикографічним порядком символів. Вважатимемо, що один рядок повинен знаходитися в матриці вище за інший в тому випадку, якщо в найлівішій з позицій, починаючи з якої рядки відрізняються, в цьому рядку знаходиться символ лексикографічно менший, ніж у іншого рядка. Іншими словами, слід відсортувати символи спочатку по першому символу, потім рядки, у яких перші символи рівні, - по другому і т.д. (рис. 2). Тепер залишився останній крок - виписати символи останнього стовпця і запам'ятати номер початкового рядка серед відсортованих. Отже, "рдакрааааббб",2 - це результат, отриманий в результаті перетворення Барроуза - Уілера.

Розглянемо процес відновлення початкової матриці. Хай нам відомий тільки результат перетворення, тобто - "рдакрааааббб",2. Відсортуємо всі символи останнього стовпця (рис. 3) у відповідності з лексикографічним порядком. Очевидно, що в результаті такого сортування ми отримали перший стовпець початкової матриці. Оскільки останній стовпець відомий, додамо його в отриману матрицю (рис. 4).

0 аабракадабр
1 абраабракад
2 абракадабра - початковий рядок
3 адабраабрак
4 акадабраабр
5 браабракада
6 бракадабраа
7 дабраабрака
8 кадабраабра
9 раабракадаб
10 ракадабрааб

Рис. 2 - Матриця циклічних перестановок рядка "абракадабра" відсортована
зліва направо у відповідності з лексикографічним порядком символі

0 а
1 а
2 а
3 а
4 а
5 б
6 б
7 д
8 к
9 р
10 р

Рис. 3. Відсортовані символи початкового рядка

0 а...р
1 а...д
2 а...а
3 а...к
4 а...р
5 б...а
6 б...а
7 д...а
8 к...а
9 р...б
10 р...б

Рис. 4. Перший і останній стовпці матриці циклічних перестановок

Тепер самий час пригадати, що рядки матриці були отримані в результаті циклічного зсуву початкового рядка. Тобто, символи останнього і першого стовпців утворюють один з одним пари. І нам ніщо не може перешкодити відсортувати ці пари, оскільки обов'язково існують такі рядки в матриці, які починаються з цих пар. І ще допишемо в матрицю і відомий нам останній стовпець (рис. 5).

0	аа...р	
1	аб...д	
2	аб...а	
3	ад...к	
4	ак...р	
5	бр...а	
6	бр...а	
7	да...а	
8	ка...а	
9	ра...б	
10	ра...б	

Рис. 5. Перший, другий і останній стовпці матриці

Таким чином, два стовпці нам вже відомі. Легко помітити, що відсортовані пари разом з символами останнього стовпця складають трійки. Аналогічно відновлюється вся матриця. А на підставі записаного наперед номера початкового рядка в матриці - і сам початковий рядок (рис. 6).

0	ааб… р	аабр...р		аабракада.р	аабракадабр
1	абр….л	абра...д		абраабрак.д	абраабракад
2	абр… а	абра...а		абракадаб.а	абракадабра
3	ада…к	адаб...к		адабраабр.к	адабраабрак
4	ака… р	акад..р		акадабраа.р	акадабраабр
5	бра… а	браа...а	...	браабрака.а	браабракада
6	бра… а	брак...а		бракадабр.а	бракадабраа
7	даб… а	дабр...а		дабраабра.а	дабраабрака
8	кад… а	када...а		кадабрааб.а	кадабраабра
9	раа... б	рааб...б		раабракад.б	раабракадаб
10	рак... б	рака...б		ракадабра.б	ракадабрааб

Рис. 6. Процес визначення всіх стовпців матриці

Однак, такий підхід вимагає великих затрат пам’яті. Існує метод швидкого зворотного перетворення. Для швидкого зворотного перетворення додатково до власне даних потрібний вектор зворотного перетворення, що є масивом чисел, розмір якого рівний числу символів в блоці. Порядок отримання вектора зворотного перетворення пропонується таким. Після отримання початкового рядка - "рдакрааааабб",2, "рдакрааааабб" записується в три стовпці масиву, як показано на рис. 7.

ррр	рра	0
ддд	дда	1
ааа	ааа	2
ккк	кка	3
ррр	рра	4
ааа	ааб	5
ааа	ааб	6
ааа	аад	7
ааа	аак	8
ббб	ббр	9

Другий стовпець матриці відсортуємо в лексикографічному порядку і допишемо четвертий стовпець, який буде містити номер даного рядка (рис. 8). Нульовий і перший стовпець залишаються незмінними.

Відсортуємо рядки матриці по першому і другому стовпці в лексикографічному порядку, нульовий стовпець матриці при цьому залишається незмінним. Результати наведено на рис. 9.

раа 2
даб 5
ааб 6
кад 7
рак 8
абр 9
абр 10
ада 1
ака 3
бра 0
бра 4

Рис. 9. Відсортовані рядки матриці

Останній стовпець чисел і є вектором зворотного перетворення. Тепер отримати початковий рядок зовсім просто. Насамперед візьмемо елемент вектора зворотного перетворення, відповідний номеру початкового рядка в матриці циклічних перестановок, T[2]=6. Інакше кажучи, як перший символ в початковому рядку слід узяти шостий символ з нульового стовпця "рдакрааааббб". Це символ "а". Далі T[6]=10. Це десятий символ з нульового стовпця "рдакрааааббб" - “б”. T[10]= 4 - “р”, T[4]=8 - “а”, T[8]=3 - “к”, T[3]= 7 - “а”, T[7]= 1 - “д”, T[1]= 5 - “а”, T[5]= 9 - “б”, T[9]= 0 - “р”, T[0]= 2 - “а”. В результаті отримаємо слово “абракадабра”, що і потрібно.

МОДЕЛЮВАННЯ І РЕЗУЛЬТАТИ

Тестування BWT-перетворення виконувалось з метою визначення ступеня ущільнення для файлів різних типів методом MTF, та іншими архіваторами до і після використанням перетворення BWT, а також для виявлення оптимального розміру блока для BWT-перетворення. Для тестування були вибрані файли розміром 0,1-1 МГбайти таких типів:

- *.doc
- *.txt
- *.mdb
- *.bmp (зображення).
- *.pdf
- *.exe

Результати тестування наведені в табл. 1 та табл. 2. Аналіз результатів наведених в табл. 1 показує, що застосування послідовно перетворення BWT, а потім MTF забезпечує кращі коефіцієнти ущільнення в порівнянні тільки з MTF. Коефіцієнт ущільнення зростає приблизно на 30 %. Однак, архіватор ZIP, краще ущільнює початкові файли у порівнянні з файлами після перетворення BWT.

Залежність коефіцієнта ущільнення від розміру блоку показує, що зменшення розміру блоку від 512 до 256 в чотири рази збільшує швидкодію прямого перетворення в той час як коефіцієнт ущільнення зменшується лише на 6 % (табл. 2). Тому прийнятними є розміри блоків від 128 до 256.

Зворотне перетворення за рахунок побудови вектора зворотного перетворення виконується значно швидше і залежить лише від розміру файлу. Для файлів розміром до 1 Мб зворотне перетворення виконується менш ніж за 10 сек.

Таблиця 1.

Результати тестування програми ущільнення даних при максимальному розмірі блоку BWT – перетворення в 512 байт

Тип файлу	До ущільнення, байт	Ущільнення методом MTF початкового файлу, байт	Ущільнення методом MTF файлу після BWT-перетворення, байт	Ущільнення архіватором ZIP початкового файлу і файлу після BWT, байт
*.doc	175616	113577	82508	43023 (61020)
*.txt	113090	94615	66645	36492 (60324)
*.mdb	774144	470773	353379	166466 (251144)
*.pdf	135640	222566	221160	126805 (130847)
*.exe	408576	449540	341603	211498 (279899)

Таблиця 2.
Залежність коефіцієнту ущільнення від розміру блоку

Розмір блоку BWT, байти	До eoskmytyyz, байт	Ущільнення методом MTF початкового файлу, байт	Ущільнення методом MTF файлу після BWT-перетворення, байт	Час виконання BWT, сек
512	175616	113577	82508	240
256	175616	113577	87210	60
128	175616	113577	92767	10
64	175616	113577	99871	7
32	175616	113577	110125	6
16	175616	113577	125383	6

СПИСОК ЛІТЕРАТУРИ

Майданюк В.П. Методи і засоби комп'ютерних інформаційних технологій. Кодування зображень. Вінниця: ВДТУ, 2001.– 63 с.

1. Балашов К.Ю. Сжатие информации: анализ методов и подходов. – Минск, 2000. – 42 с (Препринт / Ин-т техн. Кибернетики НАН Беларуси; № 6).
2. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384 с.

3. Семенюк В. В. Экономное кодирование дискретной информации. – СПб.: СПбГИТМО (ТУ), 2001. – 115 с.

Надійшла джо редакція 25.11.2008

МАЙДАНЮК В.П. – к.т.н., доцент, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, Україна.

КИРИЧЕНКО О.В. – пошукач кафедри АІВТ, Вінницький національний технічний університет, Вінниця, Україна.