УДК 004.627

СТИСНЕННЯ ДАНИХ

Ратникова Т.В.

Науковий керівник - доцент кафедри "Інформаційні технології проектування в машинобудуванні", канд.. техн.. наук

Павлишко А.В.

Стиснення даних - процедура перекодування даних, яка виробляється з метою зменшення їх обсягу. Застосовується для більш раціонального використання пристроїв зберігання та передачі даних.

Стиснення буває без втрат (коли можливо відновлення вихідних даних без спотворень) або з втратами (відновлення можливе з спотвореннями, несуттєвими з точки зору подальшого використання відновлених даних). Стиснення без втрат зазвичай використовується при обробці комп'ютерних програм і даних, рідше - для скорочення обсягу звуковий, фото-та відеоінформації. Стиснення з втратами застосовується для скорочення обсягу звуковий, фото-та відеоінформації, воно значно ефективніше стиснення без втрат.

Стиснення базується на усуненні надмірності інформації, що міститься у вихідних даних. Прикладом надмірності є повторення в тексті фрагментів (наприклад, слів природної або машинної мови). Подібна надмірність зазвичай усувається заміною повторюваних послідовностей більш коротким значенням (кодом). Інший вид надмірності пов'язаний з тим, що деякі значення в даних, які стискаються, зустрічаються частіше інших, при цьому можливо замінювати дані, що часто зустрічаються, більш короткими кодами, а рідкісні - більш довгими (ймовірнісний стиск).

Є 2 основні підходи до стиснення файлів невідомого формату.

• На кожному кроці алгоритму стиснення або наступний символ поміщується як є (зі спеціальним прапором позначає, що він не стиснутий), або вказуються кордони слова з

попереднього шматка, яке співпадає з наступними символами файлу. Розархівування стислих таким чином файлів виконується дуже швидко, тому ці алгоритми використовуються для створення програм, які розпаковуються самі.

• Для кожної послідовності в кожен момент часу збирається статистика її зустрічі у файлі. На основі цієї статистики обчислюється ймовірність значень для чергового символу. Після цього можна застосовувати той чи інший різновид статистичного кодування, наприклад, арифметичне кодування або кодування Хаффмана для заміни часто зустрічаючих послідовностей на більш короткі, а рідко зустрічаючих - на більш довгі.

Існує багато різних практичних методів стиснення без втрати інформації, які, як правило, мають різну ефективність для різних типів даних та різних обсягів. Однак, в основі цих методів лежать три теоретичні алгоритми:

• алгоритм RLE (Run Length Encoding);

• алгоритми групи KWE (KeyWord Encoding);

• алгоритм Хаффмана.

В основі алгоритму RLE лежить ідея виявлення повторюваних послідовностей даних і заміни їх більш простою структурою, в якій вказується код даних та коефіцієнт повторення. Коефіцієнт стискування, що характеризує ступінь стиснення, можна обчислити за формулою:

$$K = \left(V_0 / V_i\right) \times 100\%$$

де V_0 - об'єм пам'яті, необхідній для зберігання вихідній (результуючої) послідовності даних, V_i -вхідної послідовності даних. Чим менше значення коефіцієнта стиснення, тим ефективніше метод стиснення. Зрозуміло, що алгоритм RLE буде давати кращий ефект стиснення при більшій довжині послідовності даних, що повторюються. Більша ефективність алгоритму RLE досягається при стисненні графічних даних (особливо для однотонних зображень).

Нами, на підставі алгоритму RLE розроблено новий метод стиснення даних, який полягає в комбінації декількох послідовних різних алгоритмів стиснення. Своєрідний багаторівневий алгоритм стиснення.

Після кожного рівня стиснення відбувається перевірка і запис отриманого розміру файлів. Це робиться для того, що може виникнути ситуація, коли отриманий розмір файлу був мінімальним після першого рівня стиснення, а в подальшому більше. У такому випадку програма повернеться до мінімального розміру файлу.

Залежно від типу файлу програма вибирає послідовність і принцип алгоритмів стиснення. Якщо вихідний файл - файл растрового зображення, то визначається його кольорова модель (RGB або CMYK) і в залежності від цього першим проходить стандартний алгоритм RLE стиснення, але не по одному байту, а групами по три відразу (для колірної моделі RGB) або групами по чотири відразу (для колірної моделі CMYK). Якщо файл не растрова фотографія, то запускається стандартний алгоритм RLE який зчитує по одному байту і змінює послідовність однакових байт на їх кількість і значення.

Після цього отримані дані проглядаються послідовно ще одним алгоритмом RLE другого рівня стиснення. Справа в тому, що висока ймовірність того, що після першого алгоритму RLE в отриманій послідовності даних залишилося багато повторюваних послідовностей байт. Залежно від типу вихідного файлу вибирається кількість байт в групі повтору (по одному послідовно - для не растрового типу файлу, по 4 для RGB, по 5 для CMYK). При аналізі файлів після подвійного RLE стиснення були відзначені послідовності даних з часто повторюваними цифрами 1, 2, 3 і 4 (особливо 1) у тих місцях файла, де алгоритми RLE не дали стиснення через чергування різноманітних даних. Відповідно часто повторювані числа 1, 2, 3 та 4 можна прибрати спеціальним алгоритмом задавши прапори для цих чисел і кількість байт які ставляться до них. Це дасть можливість істотно зменшити отриманий файл і з 100% гарантією сказати, що отриманий файл буде завжди менше вихідного, а не навпаки, як це іноді трапляється.