

ПОРІВНЯННЯ МЕТОДІВ ВТОРИННОГО УЩІЛЬНЕННЯ АЛГОРИТМУ JPEG

Вовк О.В.

Національний технічний університет України «Київський політехнічний інститут», студент

Вступ

JPEG є одним з найпопулярніших алгоритмів ущільнення зображень з втратами. Даний алгоритм використовується для ущільнення повно-колірних зображень з плавними переходами кольорів. Етап вторинного ущільнення даного алгоритму може проводитись різними методами, які в тих чи інших ситуаціях можуть мати певні переваги.

I. Постановка задачі

Метою даної роботи є порівняння різних методів вторинного ущільнення алгоритму JPEG та визначення специфіки їх застосування.

II. Алгоритми ущільнення

JPEG можна поділити на кілька етапів:

- Підготовка.
- Дискретно косинусоїдальне перетворення.
- Квантування.
- Вторинне ущільнення.

Перші три етапи є однаковими для різних варіантів JPEG, але четвертий етап може відрізнятися в залежності від вибраного алгоритму. Є три основних алгоритми вторинного ущільнення:

1. RLE.
2. LZW з кодом змінної довжини.
3. Адаптивне кодування Хаффмана.

RLE — простий алгоритм ущільнення, який оперує серіями даних, де один і той же символ зустрічається кілька разів підряд. При кодуванні, рядок з однаковими символами, які являють собою серію, замінюється на рядок, який містить сам символ та кількість його повторень.

LZW — алгоритм Лампеля-Зіва-Велча. Це універсальний алгоритм ущільнення даних без втрат. При стисненні, даний алгоритм динамічно створює таблицю перетворення рядків в якій певним послідовностям символів ставиться у відповідність група біт певної фіксованої довжини. Таблиця ініціалізується усіма 1-символьними рядками (у разі 8-бітних символів - це 256 записів). В процесі кодування, алгоритм переглядає символ за символом, і зберігає кожен новий, унікальний 2-символьний рядок в таблицю у вигляді пари код/символ, де код посилається на відповідний перший символ. Після того як новий 2-символьний рядок збережений в таблиці, на вихід передається код першого символу. Коли на вході читається черговий символ, для нього за таблицею знаходиться рядок максимальної довжини який вже зустрічався, після чого в таблиці зберігається код цього рядка з наступним символом на вході. На вихід видається код цього рядка, а наступний символ використовується як початок наступного рядка.

Адаптивне кодування Хаффмана — алгоритм оптимального префіксного кодування алфавіту.

Побудова коду Хаффмана зводиться до побудови відповідного бінарного дерева за наступним алгоритмом:

1. Складається список кодованих символів, при цьому один символ розглядається як дерево, що складається з одного елемента, вагою, рівною частоті появи символу в тексті.
2. Зі списку вибирається два вузли з найменшою вагою.
3. Формується новий вузол з вагою, рівною сумі ваг обраних вузлів, і до нього приєднується два обраних вузла в якості дочірніх.
4. Щойно сформований вузол додається до списку.
5. Якщо в списку більше одного вузла, то повторюються пункти з другого по п'ятий.

Адаптивне ущільнення дозволяє не передавати модель повідомлення разом з ним самим і обмежитися одним проходом по повідомленню як при кодуванні, так і при декодуванні.

III. Порівняння алгоритмів

RLE. Очевидно, що даний алгоритм кодування ефективний для даних, що містять велику кількість серій, наприклад, для простих графічних зображень, таких як іконки, графічні малюнки.

Однак це кодування погано підходить для зображень з плавним переходом тонів, таких як фотографії.

LZW. Цей метод дозволяє досягти одну з найкращих ступенів ущільнення серед інших існуючих методів ущільнення графічних даних, при повній відсутності втрат. Перевагами даного алгоритму є одно-прохідність, відсутність потреби збереження до вихідного файлу таблиці рядків при декомпресії. Недоліком цього алгоритму є те, що не проводиться аналіз вхідних даних.

Адаптивне кодування Хаффмана. Алгоритм служить для отримання префіксних кодів - кодів змінної довжини, які дозволяють здійснювати економне кодування інформації. При економному кодуванні символічної інформації символи, що часто зустрічаються кодуються короткими кодами, а ті, що рідко зустрічаються — довгими. Таким чином, в наслідок аналізу вхідної інформації, у загальному випадку даний алгоритм даватиме найкращий результат.

Розглянемо роботу даних алгоритмів на прикладі ущільнення довільного зображення з розширенням 8x8 яке займає 64 байти. Результати роботи алгоритмів в табл.1.

Таблиця 1

Результати роботи алгоритмів ущільнення

Алгоритм	Розмір вихідного зображення (байт)	Розмір стисненого зображення (байт)	Процент від початкового зображення (%)
RLE	64	34	53.125
LZW	64	35	54.687
Адаптивне кодування Хаффмана	64	22	34.375

Як видно з таблиці, Адаптивне кодування Хаффмана дає найкращий результат. Стиснене зображення займає лише 34% від вихідного. RLE та LZW показали приблизно однаковий результат, але ефективність даних алгоритмів може варіюватися в залежності від вихідного зображення. Так алгоритм RLE буде ефективним лише у вузькому колі випадків і він не підходить для зображень з плавним переходом кольорів. Алгоритм LZW є більш універсальним, але в загальному випадку він поступається кодуванню Хаффмана, проте він швидший.

Для підвищення ступеню ущільнення перед проходом за алгоритмом адаптивного кодування Хаффмана можна провести попереднє ущільнення вхідного вектора за допомогою алгоритму RLE. Але це призведе до ускладнення процесу кодування/декодування та підвищить вимоги до обчислювальних потужностей необхідних для цього алгоритму. Даний підхід буде найбільш ефективним при ущільненні великих зображень з малою різноманітністю кольорів та відтінків.

Висновок

Таким чином, найбільш ефективним з точки зору ступеню ущільнення та обчислювальної складності є алгоритм адаптивного кодування Хаффмана. Проте, в окремих вузьких випадках інші представлені алгоритми можуть виявитися більш ефективними.

З навчальною та демонстраційною метою йде робота над створенням програмного продукту, який є частиною віртуальної мультимедійної лабораторії — демонстраційний модуль ущільнення зображень. Метою цього модуля є наочна демонстрація алгоритмів ущільнення зображень на прикладі алгоритму ущільнення JPEG. Оскільки алгоритм RLE не дає високих результатів ущільнення у більшості випадків – його широко не використовують в наш час стосовно зображень (фотографій). LZW також не став стандартом для JPEG через правові моменти та патенти на нього. Програмний продукт пишеться на JavaScript, що зменшує швидкодію, тому гібридний алгоритм не підходить через свою підвищену обчислювальну складність. Отже варто використовувати у якості алгоритму вторинного ущільнення алгоритм Хаффмана, який на даний момент являється стандартом алгоритму JPEG.

Список використаних джерел

1. Д. Сэлмон. Сжатие данных, изображения и звука. — М.: Техносфера, 2004. — 368 с.
2. Ватолин Д.С., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002.
3. G.K.Wallace "The JPEG still picture compression standard"//Communication of ACM. Volume 34.Number 4 April 1991.