

Наумовський А. Ю., студент 1 курсу
(магістратура) спеціальності «Інженерія
програмного забезпечення» ВНТУ
Майданюк В. П., к.т.н., доцент кафедри
програмного забезпечення

КОНТЕКСТНЕ МОДЕЛЮВАННЯ ТА АРИФМЕТИЧНЕ КОДУВАННЯ ПРИ УЩІЛЬНЕННІ ЗОБРАЖЕНЬ

Вінницький національний технічний університет, Україна

Анотація

Запропоновано засіб ущільнення зображень без втрат із показниками ущільнення, які перевершують інші формати ущільнення зображень без втрат, зокрема, такі як PNG, WebP, BPG або GIF.

Ключові слова: ущільнення даних, зображення, кодування зображень, ущільнення без втрат.

Abstract

The method image lossless compression with compression rates, which outperforming other formats like PNG, WebP, BPG or GIF was proposed

Keywords: data compression, image, image encoding, lossless compression.

Вступ

Багато трафіку та пам'яті на носіях інформації може бути даремно витрачено через використання неоптимального формату файлу для зберігання зображень. Наприклад, для деякого зображення може знадобитися близько 46 МБ в неущільненому форматі. Якщо користувач достатньо добре поінформований, то він помітить, що це малюнок ліній з кількома різними кольорами, тому він збереже зображення як GIF-файл (1106 KB) або файл PNG (804 KB), або, можливо, як WebP без втрат файл (787 KB). У іншому випадку користувач може зберегти зображення як файл PNG24 (1,5 MB), або набагато гірше, як JPEG з втратами (5,4 MB) або JPEG2000 без втрат (12 MB).

Формати файлів JPEG [1], WebP або BPG чудові для фотографій, але для інших типів зображень артефакти ущільнення є небажаними. Формати зображень, такі як GIF, PNG [2] та WebP без втрат, добре працюють для лінійних малюнків, але вони менш підходять для великих фотографій. На практиці використовуються і потрібні принаймні два формати зображень: JPEG та PNG.

LIF (Lossless Image Format) не просто покращує коефіцієнт ущільнення, але й робить стару дихотомію застарілою. Ми сподіваємось, що формат зображень LIF, запропонований у цій роботі, може покращити сучасний стан, оскільки він добре працює як для фотографій, так і для лінійних малюнків, і для всього "між ними".

Результати дослідження

Арифметичне кодування [3], також відоме як кодування діапазону, є формою ентропійного кодування на основі моделі ймовірності кодованих бітів. Натхненні кодеком FFV1 [4], ми використовуємо варіант контекстно-адаптивного бінарного арифметичного кодування. Ми назвали метод ентропійного кодування «метаадаптивним арифметичним кодуванням» майже нульового цілого числа, оскільки він метаадаптивний, тобто сама контекстна модель адаптована до даних.

Контекстно-адаптивне бінарне арифметичне кодування

У КАБАК модель ймовірності є адаптивною. Спочатку починаємо з довільним розподілом ймовірностей (наприклад, 50% для кожного біта). Після прийому блоку бітів (64 біти) модель оновлюється. Мета полягає в тому, щоб дізнатись про фактичний розподіл ймовірностей попередніх значень бітів, щоб краще прогнозувати майбутнє. Якщо є додаткова інформація про контекст, то можемо використовувати різну ймовірність у кожному контексті. Кореляція між контекстом і бітами призводить до більш точного оцінювання ймовірності та кращого ущільнення.

Для кодування цілих чисел використовується подання експоненти мантиси з різним контекстом

для кожної бітової позиції. Розроблена бінарizzaція заснована на моделях кодека FFV1 з деякими вдосконаленнями. Зокрема, запропоновано нове подання ущільнених даних, відмінністю якого є можливість запису даних в дерево рішень, що дозволило підвищити коефіцієнт ущільнення та швидкість кодування та декодування. Крім того, ми використовуємо різні контексти для кожного блоку бітів.

Навчання дерева

Ми пропонуємо динамічну структуру даних як контекстну модель. По суті це дерево рішень (фактично одне дерево на потік), створене під час кодування. На рисунку 1 показано приклад дерева. Кожен внутрішній (не листковий) вузол має умову: нерівність, яка порівнює одну із властивостей контексту із значенням блоку бітів. Дочірні вузли відповідають двом тестовим гілкам. Під час кодування кожен вузол містить один фактичний контекст (масив ймовірностей) та два віртуальних контексти на властивість. Під час декодування використовуються лише фактичні контексти. Для кожного кодованого значення дерево рішень переміщується, поки не буде досягнуто вузла листків. Спочатку фактичний контекст використовується для виведення значення контексту, а оцінка вартості (кількість бітів для ущільненого виводу) оновлюється. Для кожної з властивостей кожен вузол листка підтримує поточне середнє значення властивостей, що зустрічаються у цьому листкові; один віртуальний контекст використовується для значень контексту нижче середнього, інший використовується для більш високих середніх значень. Для кожної властивості відповідно підбираємо віртуальний контекст та оновлюємо його ймовірності та оцінку витрат.

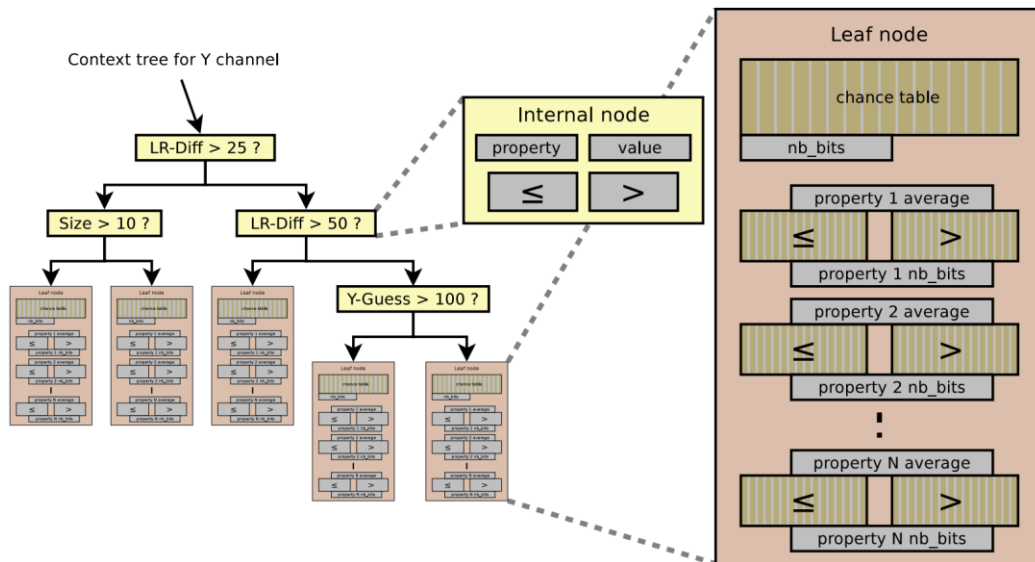


Рисунок 1 — Структура дерева

Ці оцінки витрат вказують, які властивості є найбільш значущими. Якщо властивість не має значення, то сума витрат для двох її віртуальних контекстів буде однаковою або більшою, ніж у фактичного контексту. Якщо все ж властивість є релевантною, то використання двох різних контекстів залежно від значення для цієї властивості призведе до кращого ущільнення. Порівнюємо вартість "найкращої" пари віртуальних контекстів у даному вузлі листка з вартістю фактичного контексту. Якщо різниця у вартості є меншою деякого фіксованого порогу, то вузол листка стає вузлом рішення. Рисунок 2 ілюструє це. Дерево, яке ми побудували, не обов'язково є оптимальним; майбутні кодери можуть використовувати інші алгоритми, оскільки структура дерева є частиною кодованого бітового потоку.

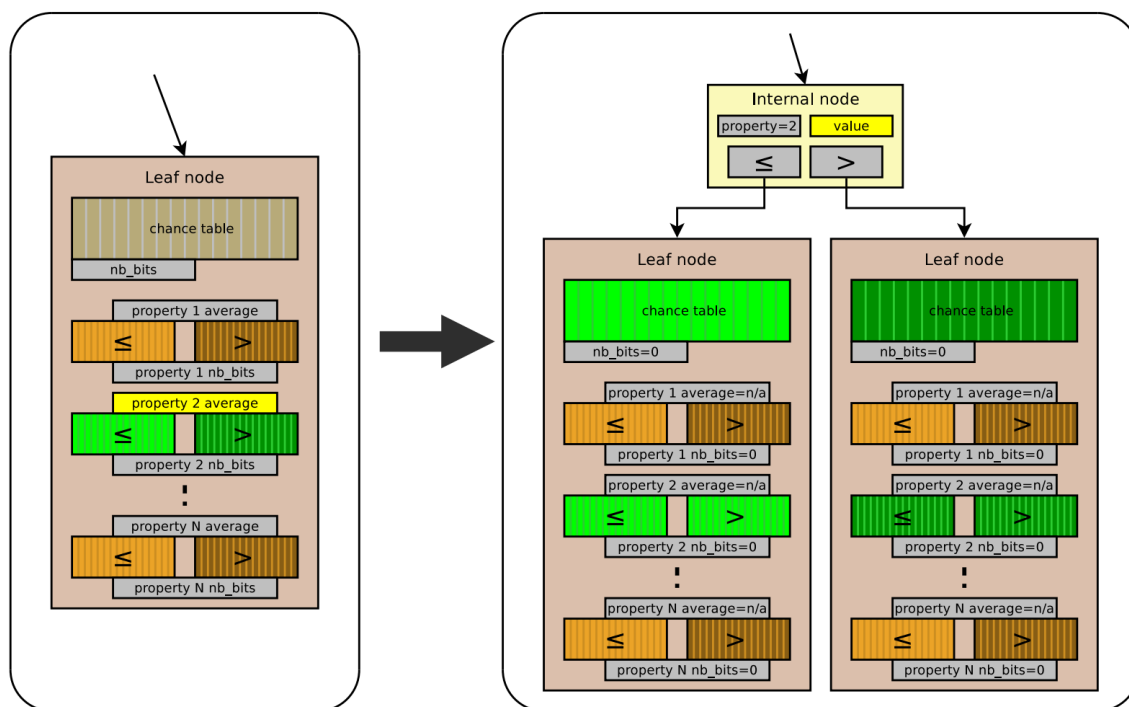


Рисунок 2 — Формування дерева: перший крок

Розроблене дерево має три основні переваги порівняно з використанням фіксованого контекстного масиву:

- 1) Не потрібно використовувати квантовані значення властивостей, тому можемо виділити майже-однакові значення властивостей;
- 2) властивості фактично використовуються лише в тому випадку, якщо вони сприяють кращому ущільненню конкретного зображення;
- 3) Контекстне дерево масштабується із зображенням: для великих, складних зображень буде використано більше контекстів, ніж для невеликих, простих зображень.

Висновки

LIF добре стискає різні види зображень, а не лише фотографії. Ми сподіваємось, що LIF може стати кроком у напрямку "універсального" формату зображення. Арифметичне кодування можна узагальнити до ущільнення загального призначення. Основна ідея полягає у використанні відносно простих дерев рішень, але будь-який тип класифікатора може бути використаний. Навчання не обов'язково має бути швидким - час кодування зазвичай є набагато менш важливим, ніж час декодування. Єдина вимога полягає в тому, що вивчений об'єкт (наприклад, дерево рішень) може зберігатися стисло і його можна швидко реконструювати під час декодування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Joint Photographic Experts Group (JPEG). JPEG 2000 standard, ISO/IEC 15444, ITU-T Recommendation T.800 [Електронний ресурс] / Joint Photographic Experts Group (JPEG) – Режим доступу до ресурсу: <http://www.jpeg.org/jpeg2000>.
2. Thomas V. Portable Network Graphics (PNG) specification, RFC 2083, ISO/IEC 15948:2004 [Електронний ресурс] / Boutell Thomas. – 2004. – Режим доступу до ресурсу: <http://www.libpng.org/pub/png/>.
3. Віттер Я. Арифметичне кодування для ущільнення даних / Я. Віттер, Р. Ніл, Дж. Клері. — Commun. ACM, 1987. — 540с.
4. Нейдермеєр М. Description of the FFV1 video codec [Електронний ресурс] / Міхаель Нейдермеєр – 2004. – Режим доступу до ресурсу: <http://www.ffmpeg.org/michael/ffv1.html>.