

# КІБЕРНЕТИКА та КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

*З метою розробки швидкодійного спрощеного кодека зображень проведено аналіз впливу відомих алгоритмів компресії зображень та інших параметрів на швидкодію. Обґрунтовано актуальність і доцільність розробки швидкодійного спрощеного кодека зображень для області Інтернету речей з метою збільшення рівня автономності пристроїв IoT, зменшення вартості побудови та поширення інфраструктури IoT. Розглянуті підходи та алгоритми для розробки швидкодійного спрощеного кодека зображень можуть бути застосовані для подальшої розробки швидкодійного спрощеного кодека відео.*

**Ключові слова:** обчислювальна складність, швидкі перетворення, ефективність обчислень, прогресивна передача даних, алгоритми інтра-прогнозування, спрощений кодек зображень, IoT.

© Я.В. Луц, В.К. Луц, 2021

УДК 004.9, 004.62, 004.627, 519.72

DOI:10.34229/2707-451X.21.1.6

Я.В. ЛУЦ, В.К. ЛУЦ

## ПРО РОЗРОБКУ ШВИДКОДІЮЧОГО СПРОЩЕНОГО КОДЕКА ЗОБРАЖЕНЬ

**Вступ.** Актуальність підвищення швидкодії кодеків зображень та відео обумовлена розвитком індустрії Інтернету речей (IoT), значним збільшенням кількості відеоданих і зображень, що генеруються та передаються (у тому числі – в реальному часі, наприклад, відео конференції та віртуальна реальність).

Еволюція алгоритмів компресії зображень та відео поки що йшла лише в одному напрямку – збільшення коефіцієнта компресії даних з метою економії необхідної для збереження пам'яті та зменшення навантаження на канали зв'язку.

Але досягнення в області збільшення обсягів накопичувачів даних та підвищення швидкості передачі даних у каналах зв'язку при повільному зростанні потужності процесорів робить актуальною задачу зменшення обчислювальної складності алгоритмів компресії даних, незважаючи на деяке очікуване збільшення розміру графічних файлів.

Дана задача може бути актуальна не тільки для IoT, де використовуються найменш потужні (енергоєфективні) мікропроцесори, а і для дорогих та високопродуктивних процесорів для обробки зображень та відео високої роздільної здатності (4K, 8K) зі збільшеним діапазоном яскравості – з метою зменшення вимог до обчислювальних ресурсів, що буде сприяти більш широкому розповсюдженню нових стандартів внаслідок застосування більш дешевих процесорів.

Інший важливий фактор – це зміна співвідношення операцій кодування й декодування графічних даних. Раніше кодування графічних даних було менш поширеним явищем, тому більш висока складність кодування відносно декодування (на порядок) не відіграла особливої ролі. Але, внаслідок широкого розповсюдження пристроїв для первинного кодування зображень та відео, набирає важливості задача зменшення обчислювальної складності кодування, особливо для області IoT, де саме кодування зображень та відео це основний процес.

### **Коефіцієнт енергоефективності алгоритмів кодування графічних даних**

З метою оцінки доцільності застосування тих чи інших алгоритмів з точки зору зменшення обчислювальної складності можна ввести коефіцієнт енергоефективності, що визначається співвідношенням між обчислювальною складністю виконання алгоритмів та їх внеском у кінцевий результат. Наприклад, якщо оригінальне зображення стискається в 30 раз, при цьому ентропійне кодування стискає у два рази, а алгоритми стиснення на основі перетворення з втратами, відповідно, в 15 разів – то при рівній обчислювальній складності цих двох етапів компресії зображення ентропійне кодування буде мати коефіцієнт енергоефективності в 7,5 разів менший.

Запропонований коефіцієнт енергоефективності алгоритмів обчислюється приблизно, оскільки обчислювальна складність алгоритмів залежить від конкретних умов застосування, особливостей програмної реалізації, наявності апаратного прискорення, конфігурації апаратної платформи та інше, тому оцінки енергоефективності різних алгоритмів можуть значно відрізнятись.

### **Огляд існуючих графічних форматів та алгоритмів компресії**

Основні сучасні графічні формати зображень, що найчастіше використовуються, такі: JPEG, PNG, GIF [1, 2] і WebP [3]. Також відомо такі формати: HEIF/HEIC, BPG (обидва на основі відеокодека H.265/HEVC [4]), AV1 Still Image File Format (AVIF) [5] – на основі відео-кодека AV1, FLIF – кодек зображень без втрат, а лише на основі передбачення і ентропійного кодування без використання перетворень; ці формати відрізняються значною обчислювальною складністю і нині не знайшли широкого розповсюдження.

Основні алгоритми, що застосовуються для компресії зображень з втратами, такі: блочні перетворення, вейвлети, прорідження (відкидання) даних.

Основні алгоритми, що застосовуються для компресії зображень без втрат, це ентропійне кодування і передбачення (значень пікселів та/або коефіцієнтів перетворення). Більшість кодеків зображень використовують комплексний підхід, комбінуючи алгоритми з втратами та без втрат. Найчастіше компресія зображень складається з трьох етапів:

- передбачення значень пікселів (стиснення без втрат);
- блочне кодування за допомогою перетворень (стиснення з втратами);
- ентропійне кодування (стиснення без втрат).

Алгоритми передбачення на підставі попередніх даних прогнозують значення наступних, після чого обчислюються та кодуються різниці між прогнозованими та реальними значеннями, які зазвичай менші, ніж абсолютні значення. У WebP використовується 10 алгоритмів передбачення [3], у HEIF, кодека зображень на основі H.265, вже 35 алгоритмів [4], а у кодека VVC (Versatile video coding) навіть 67 алгоритмів передбачення. Задача створення оптимального кодека зображень – нетривіальна, має різні рішення, і в цьому напрямку тривають дослідження.

### **Оцінка перспектив підвищення швидкодії кодека зображень**

**Етап передбачення значень пікселів.** Для підвищення швидкодії кодека зображень бажано використовувати якнайменшу кількість алгоритмів передбачення, оскільки крім обчислювальної складності виконання алгоритмів передбачення, необхідно враховувати обчислювальну складність алгоритму оцінки та вибору (зазвичай для цього обчислюють співвідношення компресія/шум). Крім того, при наявності двох і більше алгоритмів передбачення потрібно зберігати додаткову службу інформацію про вибір алгоритму.

Зауважимо, що застосування алгоритмів передбачення зменшує ефективність використання перетворень на наступному етапі компресії даних – оскільки абсолютні значення пікселів мають між собою високу кореляцію і краще стискаються, ніж низькорельовані залишки (різниці), які залишаються в результаті застосування алгоритмів передбачення значень пікселів.

Враховуючи ці зауваження, зробимо висновок. З метою збільшення швидкодії кодеків зображень бажано розробити один універсальний алгоритм передбачення, щоб не застосовувати додаткові алгоритми для оцінки та вибору найкращого варіанту, оскільки зменшення рівня компресії на даному етапі частково компенсується на наступному етапі (на якому застосовуються перетворення).

Але задача вибору одного універсального алгоритму передбачення – складна, оскільки для різних зображень (чи областей зображень) краще підходять різні алгоритми.

**Етап використання перетворень.** Це основний етап роботи кодека зображень, на якому можна корегувати відразу декілька параметрів з метою збільшення швидкодії або пошуку оптимального співвідношення складність/якість. Відповідно, можна розробити адаптивний підхід, який використовує ті чи інші алгоритми в залежності від додаткових умов чи вибору користувача. Наприклад, вибір між одним алгоритмом блочного перетворення і декількома. У найпростішому варіанті, що реалізовано в JPG, використовується лише одне фіксоване перетворення – дискретне косинусне перетворення (ДКП) розмірності 8. В HEIF (на основі H.265) адаптивно використовуються 4 цілочисельних косинусних перетворення (ЦКП), розміром від 4x4 до 32x32 [4], і для блоків розмірності 4 додатково до ЦКП застосовується цілочисельне синусне перетворення (ЦСП), причому в різних режимах, вибір яких також складає додаткову обчислювальну складність, оскільки для кожного рядка та стовпця застосовують всі можливі комбінації з двох перетворень ДКП і дискретного синусного перетворення (ДСП), до яких потім застосовують досить складні алгоритми обчислення RDO, rate-distortion optimization (а виграш від застосування ДСП складає приблизно лише 1 %). Згідно статистики при дослідженні H.265/HEIF [4], основні перетворення – це ЦКП 8x8 і 16x16, а на перетворення розмірності 4 та 32 припадає значно менше випадків застосування. Отже, можна зробити висновок, що для збільшення швидкодії застосовувати лише одне перетворення (розмірності 8 або 16), і адаптивно, за певних умов, використовувати два перетворення, 8x8 та 16x16, для незначного покращення компресії даних.

Частину кодеків використовують не просто блочні перетворення, а блочні перетворення з перекриттям блоків, для зменшення блочних артефактів – у результаті узгодження відновлених даних між собою (один і той же піксел входить у два блоки, і має два різні, хоча й близькі значення, на підставі яких формується одне усереднююче). При цьому блоки з перекриттям генерують більшу кількість коефіцієнтів, внаслідок більшої кількості блоків для того ж зображення. Отже, потрібно пропорційно збільшити кількість коефіцієнтів для перетворення без перекриття, щоб отримати й порівняти два зображення. В результаті отримаємо приблизно ту ж саму якість й для блочних перетворень без перекриття за рахунок збільшення кількості коефіцієнтів, а з позиції обчислювальної складності – більшу складність не тільки за рахунок збільшення кількості блоків перетворень, а й внаслідок необхідності узгодження значень пікселів. З іншого боку, кодек AVIF [5] використовує саме перетворення з перекриттям – тому це питання потребує додаткових досліджень. Але попередньо для швидкодіючого кодека обираємо блочне перетворення без перекриття.

Цікаво також дослідити застосування додаткового перетворення до коефіцієнтів, отриманих на першому рівні перетворення. Наприклад, такі кодеки, як WebP від компанії Гугл та JPEG XR, розроблені фірмою Майкрософт у 2006 році, використовують дворівневе перетворення (за аналогією з вейвлет-технологією), стискаючи отримані коефіцієнти косинусного перетворення за допомогою перетворень Адамара – Уолша (WebP) та Адамара (JPEG XR). Але особливість даних кодеків це використання перетворень першого рівня малої розмірності – 4x4. Відповідно, коефіцієнти сусідніх блоків будуть мати вищу кореляцію, ніж для перетворень більшої розмірності (8 та 16, не кажучи вже про 32). Саме тому в цьому випадку обґрунтоване застосування другого рівня перетворення, що дозволяє додатково зменшити кореляцію даних. Але при використанні перетворень більшої розмірності дворівневе перетворення буде мати менший рівень додаткового стиснення, і є недоцільним з точки зору обчислювальної складності. Відповідно, у швидкодіючому спрощеному кодеку зображень буде достатньо використовувати лише перетворення розмірності 8 та 16 (адаптивно).

Розглянемо питання доцільності використання матриці квантування коефіцієнтів перетворення. Не тільки JPG, але й більш нові кодеки зображень, наприклад, AVIF, використовують матрицю квантування замість одного значення параметра квантування коефіцієнтів QP. AVIF використовує вісім наборів параметрів квантування [5], які можна адаптивно вибирати для кожного блоку і макроблоку, що мають індивідуальні параметри для двох площин кольоровості і можуть використовувати просторове прогнозування.

Але JPG мав на меті отримати максимальну компресію, тому й було вибрано занадто великі коефіцієнти квантування, особливо для низьких частот (що й призвело до посилення блочних ефектів). Нині переважають протилежні тенденції, на збільшення якості зображень, тому для швидкодіючого кодека зображень достатньо використовувати один параметр квантування коефіцієнтів QP замість матриці.

Наступне питання – наявність швидких алгоритмів обчислення блочних перетворень. Наприклад, косинусні перетворення, що застосовані в стандарті H.265, мають лише часткову факторизацію. З метою збільшення швидкодії доцільно застосовувати косинусні перетворення, для яких розроблені швидкі алгоритми низької обчислювальної складності, як в [6, 7]. Також замість дискретних перетворень, які використовуються у WebP, доцільно застосовувати цілочисельні перетворення, оскільки це теж підвищує швидкодію кодека.

**Етап ентропійного кодування.** Після отримання коефіцієнтів перетворення і відкидання частини даних (шляхом ділення коефіцієнтів на параметр квантування і заокруглення до цілого) застосовують алгоритми ентропійного кодування, які дозволяють зменшити в 2 – 3 рази розміри отриманого файлу зображення. Алгоритми ентропійного кодування мають досить високу обчислювальну складність, але зменшити складність кодування даних на цьому етапі досить важко, хоча нові підходи розробляються, наприклад, заслуговує на подальші дослідження метод рекурсивного групового кодування [8], який не тільки має меншу обчислювальну складність, а й кращу якість, тобто більшу ступінь компресії файлу.

**Етап фільтрації зображень після декодування.** Після відновлення зображення застосовують алгоритми фільтрації, які покращують візуальну якість зображення, зменшуючи блочні артефакти. При умові високої якості збережених зображень (а саме ця тенденція спостерігається нині і буде ще більш поширеною в майбутньому) алгоритми фільтрації можуть бути застосовані адаптивно, варіативно (тобто використовувати складні алгоритми, або спрощені алгоритми фільтрації з меншою обчислювальною складністю, або не використовувати взагалі, оскільки спотворення для зображень високої якості незначні). Крім того, алгоритми фільтрації не зменшують похибок відновлення, різниці між оригінальним та відновленим зображенням, як за абсолютними значеннями, так і за критеріями оцінки похибок (середньоквадратична похибка та ін.). З метою перевірки цього твердження проведено наступний експеримент: оригінальне зображення було спочатку зменшено в 4 рази, потім – збільшено в 4 рази (для цього застосовувались прості алгоритми – для зменшення знаходили середнє значення блоку  $2 \times 2$ , що й становило значення пікселя зменшеного зображення, а для збільшення – значення пікселя меншого зображення приймалось як значення блоку пікселів  $2 \times 2$ ). Відповідно, були обчислені абсолютні різниці та відносна похибка між оригінальним зображенням та отриманим в результаті експерименту. Потім зменшене в 4 рази зображення збільшили (в 4 рази) за допомогою стандартної програми Microsoft Office Picture Manager, яка використовує різні алгоритми покращення зображення для підвищення якості. Для отриманого зображення також обчислили різниці та похибку відносно оригінального зображення. Очікувалось, що покращене зображення буде мати менші оцінки похибки відносно оригіналу, ніж просто збільшене. Але різниці взагалі не виявилось (з точністю до 0,1 %). Тобто візуально якість незначно покращилась, але абсолютні та відносні значення похибок були однаковими. Це може означати, що наближення значень пікселів відновленого зображення до оригіналу, яке дають алгоритми фільтрації, компенсуються похибками, що вносяться в зображення цими алгоритмами. При цьому зауважимо, що

розмитість на зображенні, яку генерують алгоритми фільтрації, в певній мірі заважає. Відповідно, відмова від застосування деблочних фільтрів є обґрунтованою не тільки з точки зору зменшення обчислювальної складності, але й з позицій кращого розпізнавання. З іншого боку, алгоритми фільтрації у майбутньому можуть бути покращеними, що призведе до збільшення чіткості контурів зображення, зменшення ефекту розмитості та зменшення похибки відносно оригіналу.

#### **Новий алгоритм прогресивної передачі даних для спрощеного кодека зображень**

Враховуючи надлишковість значної кількості зображень пропонується спосіб прогресивної передачі зображень, який передає оригінальне зображення лише за запитом.

Запропонований алгоритм прогресивної передачі передає спочатку ряд зменшених зображень оригіналу від найменшого до найбільшого з урахуванням налаштувань передачі; а оригінал передається тільки за вимогою користувача. Алгоритм прогресивної передачі зображень має зручність для користувачів, яким для попереднього огляду не потрібно в більшості випадків занадто великих параметрів якості та розміру зображення, і навпаки – у яких іноді виникає необхідність більш якісного зображення, що і пропонує алгоритм прогресивної передачі.

Іншими перевагами запропонованого способу прогресивної передачі зображень є технічні, а саме: зменшені версії зображення дублюють основну інформацію базового зображення, а отже, можуть бути використані для відновлення зображення при частковій втраті частини даних внаслідок завад у радіомережах, тобто виконують функції додаткового захисту даних; для алгоритмів розпізнавання образів використання зменшених зображень для перших ітерацій може значно зменшити обчислювальну складність та прискорити швидкодію.

Недоліком даного підходу можна назвати збільшення обсягів переданих даних у випадку, коли потрібно передавати оригінал зображення за запитом. В залежності від співвідношення кількості даних, де достатньо зменшених зображень, та даних, де робиться запит і передається оригінал зображення – ця особливість може бути як перевагою (передається менше даних, оскільки оригінали не потрібні, достатньо зменшених версій зображень), так і недоліком (передаються і оригінали, і зменшені варіанти зображень).

Іншим недоліком запропонованого алгоритму є умова зв'язування даних між собою. Відповідно, це може створювати певні складнощі, наприклад, необхідність збереження посилань, оригіналів зображень на той період часу, коли може знадобитись додаткова інформація за запитом. При реалізації алгоритму це потрібно враховувати.

**Висновки.** Обґрунтовано доцільність розробки швидкодіючого спрощеного кодека зображень, проведено аналіз впливу на швидкодію відомих алгоритмів кодування зображень та інших параметрів. Для підвищення швидкодії кодека зображень запропоновано зменшити кількість та складність застосованих алгоритмів, у тому числі – використовувати лише одне блочне цілочисельне перетворення зі швидкими алгоритмами обчислення низької складності. Запропоновано новий алгоритм прогресивної передачі зображень – передача зменшеного зображення з наступною передачею за вимогою оригінального зображення. Розглянуті підходи для розробки швидкодіючого спрощеного кодека зображень можуть бути застосовані для подальшої розробки швидкодіючого спрощеного відео кодека.

#### **Список літератури**

1. Сэлмон Д. Сжатие данных, изображений и звука. М.: Техносфера, 2004. 368 с.  
<https://www.elibrary.ru/item.asp?id=19582820>
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2012. 1104 с.  
<https://www.elibrary.ru/item.asp?id=21558366>
3. A new image format for the Web | WebP | Google Developers. URL: <https://developers.google.com/speed/webp/>
4. ITU-T Rec. H.265/ISO/IEC 23008-2: 2013. Information technology – High efficiency coding and media delivery in heterogeneous environments. Part 2: High efficiency Video Coding, 2013. <https://www.iso.org/standard/35424.html>

5. AV1 Image File Format (AVIF). URL: <https://aomediacodec.github.io/av1-avif/>
6. Гнатив Л.А. Целочисленные косинусные преобразования для высокоэффективного кодирования изображений и видео. *Кибернетика и системный анализ*. 2016. **52** (5). С. 161–176. <https://doi.org/10.1007/s10559-016-9881-7>
7. Гнатив Л.А. Целочисленные косинусные преобразования: методы построения новых быстрых преобразований порядка 8, 16 и их применение. *Кибернетика и системный анализ*. 2014. 50 (6). С. 104–121. <https://doi.org/10.1007/s10559-014-9682-9>
8. Kozhemiakina N., Lukin V.V., Ponomarenko N.N., Astola J., Egiazarian K.O. JPEG compression with recursive group coding. *Electronic Imaging*. 2016. 15. P. 1 – 6. <https://doi.org/10.2352/ISSN.2470-1173.2016.15.IPAS-196>

Одержано 09.02.2021

**Луц Ярослав Васильович**,  
магістрант Національного технічного університету України  
«КПІ ім. Ігоря Сікорського», Київ,  
[rorr-t@ukr.net](mailto:rorr-t@ukr.net)

**Луц Василь Костянтинівич**,  
молодший науковий співробітник  
Інституту кібернетики імені В.М. Глушкова НАН України, Київ.  
[ykluts@gmail.com](mailto:ykluts@gmail.com)

UDC 004.9, 004.62, 004.627, 519.72

Ya.V. Luts<sup>1</sup>, V.K. Luts<sup>2\*</sup>

## About the Development of a High-Speed Simplified Image Codec

<sup>1</sup> National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

<sup>2</sup> V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv

\* Correspondence: [ykluts@gmail.com](mailto:ykluts@gmail.com)

In order to develop a high-speed simplified image codec, an analysis of the influence of known image compression algorithms and other parameters on performance was done. The relevance and expediency of developing a high-speed simplified image codec for the Internet of Things in order to increase the level of autonomy of IoT devices, reduce the cost of construction and dissemination of IoT infrastructure were substantiated. The efficiency coefficient of image compression algorithms was introduced, which is determined by the ratio between the computational complexity of the algorithms and their contribution to the final result. Simplification and reduction of the number of algorithms for predicting pixel values were proposed and substantiated, because at this stage a significant number of computational operations is added by the procedure of comparing different prediction algorithms with each other. It is proposed to use only one block integer transformation with fast low complexity algorithms of calculating, which will significantly reduce the complexity of the block transformation stage, including due to the lack of high computational complexity of the algorithm for comparing the quality of block transformations. At the stage of entropy coding, it is also proposed to use simplified algorithms, because the contribution of this stage to the overall result in the general background is quite small, and the computational complexity is high (50 – 70 % of all calculations). A new algorithm for progressive image transfer was proposed - the transfer of a reduced image followed by the transfer of the original image on demand. The considered approaches and algorithms for the development of high-speed simplified image codec can be applied to further development of high-speed simplified video codec.

**Keywords:** computational complexity, fast transforms, computational efficiency, progressive data transfer, intra-prediction algorithms, simplified image codec, IoT.