Національна академія наук України
Інститут кібернетики імені В.М. Глушкова

Кваліфікаційна наукова
праця на правах рукопису

Луц Ярослав Васильович

УДК 004.627, 004.932

ДИСЕРТАЦІЯ

МЕТОДИ І АЛГОРИТМИ ОБРОБЛЕННЯ І КОДУВАННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ШВИДКИХ ЦІЛОЧИСЛОВИХ СИНУС-КОСИНУСНИХ ТА ДИСКРЕТНИХ МУЛЬТИВЕЙВЛЕТ ПЕРЕТВОРЕНЬ

113 – «Прикладна математика»
Галузь знань 11 – «Математика та статистика»
Подається на здобуття наукового ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Луц Я.В.

Науковий керівник Гнатів
Лев Олексійович,
к.т.н., с.н.с.

Київ – 2025

АНОТАЦІЯ

Луц Я.В. Методи і алгоритми оброблення і кодування зображень на основі швидких цілочислових синус-косинусних та дискретних мультивейвлет перетворень. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 113 «Прикладна математика». – Інститут кібернетики імені В.М. Глушкова Національної академії наук України, Київ. – 2025.

Об'єкт дослідження **–** підходи, методи і алгоритми кодування та оброблення, аналізу зображень з втратами.

Мета роботи **–** дослідження та розробка нових ідей, методів, алгоритмів оброблення, аналізу та кодування зображень з втратами.

Мета роботи обумовлює дослідження перспектив розвитку аналізу та кодування зображень, зокрема з урахуванням розвитку нейромереж та штучного інтелекту. Іншим напрямком роботи є розробка нових методів та алгоритмів, в тому числі нових ортогональних перетворень, серед яких найбільша увага приділяється цілочисловим однонормовим гібридним перетворенням на основі базових перетворень різних типів – синус-косинусним, модифікованим косинусним та іншим гібридним варіантам перетворень. Окрема увага приділяється спрощеним гібридним та мультигібридним ортогональним перетворенням, які ефективні для швидкодійного кодування з урахуванням застосування нейромереж для декодування, відновлення, генерації даних.

Основні отримані результати:

Запропоновано ряд нових підходів, методів та алгоритмів для збільшення енергоефективності кодування як традиційних кодеків, так і гібридних, з застосуванням нейромереж на додачу до традиційних методів, серед яких новий спрощений режим роздільного кодування з використанням єдиної комбінації косинусного і синусного перетворень та ряд нових цілочислових перетворень порядку 4, 8 та 16. Запропоновані перетворення можуть бути альтернативою синусному перетворенню типу VII для

енергоефективного кодування. Розроблена концепція мультигібридних ортогональних перетворень, яка доповнює гібридні перетворення. Запропоновано два методи побудови мультигібридних перетворень – на основі двох гібридних перетворень і на основі комбінації гібридного і відомого перетворень. Розроблено три мультигібридні цілочислові перетворення. Розроблено матричний метод побудови гібридних цілочислових спрощених синус–косинусних перетворень типу VII порядку 16 на основі двох перетворень порядку 8: синусного типу VII і косинусного типу II. Розроблено два цілочислових однонормових спрощених синус-косинусних перетворення низької обчислювальної складності, яка менше на 38,6% порівняно з відомим алгоритмом дискретного синусного перетворення типу VII порядку 16 з цілочисловою апроксимацією, який застосований у сучасному стандарті з відеокодування VVC (Versatile Video Coding). Розроблено швидкий алгоритм обчислення оберненого цілочислового косинус-синусного перетворення типу IV порядку 16 низької мультиплікативної складності. Розроблено метод побудови гібридного синус-косинусного перетворення типу II порядку 4, і запропоновано три нових цілочислових перетворення зі швидкими алгоритмами обчислення. Розроблено метод побудови мультигібридного спрощеного синус-косинусного перетворення типу II порядку 8, запропоновано два нових цілочислових перетворення і розроблено для них швидкі алгоритми низької обчислювальної складності без виконання операції множення. Запропоновано метод побудови цілочислових модифікованих косинусних і синусних перетворень типу II порядку 8, на основі якого розроблено цілочислові однонормові модифіковані синусні та косинусні перетворення типу II порядку 8 зі швидкими алгоритмами обчислення. Розглянуто архітектуру гібридного кодека з використанням нейромереж на стороні декодера, що має пріорітетом інформативність та мінімальність закодованих традиційними методами даних, оскільки використовує найбільш узагальнені вхідні дані та застосовує генеративні нейромережі для відновлення даних та підвищення візуальної якості. В рамках цього підходу запропоновано ряд нових методів та

алгоритмів кодування, зокрема, алгоритм виділення мультимасштабних контурів за критеріями масштабу, яскравості і зв'язності. Запропоновано спрощений алгоритм отримання контурів на основі суми вертикальних та горизонтальних перепадів, різниць. Запропоновано передавати традиційними методами в якості спеціалізованої інформації контури, спрощені палітри, реперні точки та фрагменти текстур для подальшого оброблення і наближеного відновлення на стороні декодера нейромережевими методами. Запропоновано нову міру для оцінки похибок в просторі кольорів на основі обчислення похибки пропорцій вертикальних та горизонтальних різниць. Розроблена комп'ютерна технологія кодування зображень на основі 2D швидких мультивейвлет-перетворень в базисі фрактальних ступінчастих функцій з мультивейвлет-пакетом розміру 8×8 для трьох рівнів розкладу.

Практичне значення результатів дослідження.

Застосування розроблених підходів, методів і алгоритмів має значне практичне значення для розробки швидкодіючих традиційних та гібридних кодеків з використанням нейромереж на стороні декодера, в тому числі для застосування в області інтернету речей, мобільного зв'язку, носимих пристроїв, супутникових та космічних технологій.

Теоретичне значення отриманих результатів полягає в подальшому розвитку теорії ортогональних перетворень, зокрема цілочисельних.

Результати досліджень були представлені в наступних темах інституту кібернетики імені В.М. Глушкова НАН України: ВФ 140.21, ВКМ 140.25.22, ВП. 140.19, ВП. 140.21.

Було зроблено 9 публікацій, одна стаття проіндексована у базах даних Scopus, дві статті у фахових журналах і дві статті опубліковано в працях міжнародних конференцій з індексацією в базах даних Scopus.

Ключові слова: адаптивні режими, адаптивне кодування, математичне моделювання, швидкі алгоритми, швидкодійний кодек, архітектура кодека, цілочислові перетворення, спрощені косинусні перетворення, синус-косинусні перетворення, кластерний аналіз, прогресивна передача даних.

ANNOTATION

Luts YA.V. Methods and algorithms for image processing and coding based on fast integer sine-cosine and discrete multiwavelet transforms. – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 113 "Applied Mathematics". – V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Kyiv. – 2025.

The **object of research** – approaches, methods and algorithms for encoding and processing, analysis of lossy images.

The **purpose of the work** is to study and develop new ideas, methods, algorithms for processing, analyzing and coding of lossy images.

The purpose of the work involves studying the prospects for the development of image analysis and coding, in particular, taking into account the development of neural networks and artificial intelligence. Another direction of work is the development of new methods and algorithms, including new orthogonal transformations, among which the greatest attention is paid to integer single-norm hybrid transformations based on basic transformations of various types - sine-cosine, modified cosine and other hybrid variants of transformations. Special attention is paid to simplified hybrid and multi-hybrid orthogonal transformations, which are effective for high-speed coding, taking into account the use of neural networks for decoding, restoration, and data generation.

Main results obtained:

A number of new approaches, methods and algorithms are proposed to increase the energy efficiency of coding of both traditional and hybrid codecs, using neural networks in addition to traditional methods, including a new simplified mode of separable coding using a single combination of cosine and sine transforms and a number of new integer transforms of order 4, 8 and 16. The proposed transforms can be an alternative to the sine transform of type VII for energy-efficient coding. The concept of multihybrid orthogonal transforms is developed, which complements

hybrid transforms. Two methods for constructing multihybrid transforms are proposed - based on two hybrid transforms and based on a combination of a hybrid and a known transform. Three multihybrid integer transforms are developed. A matrix method for constructing hybrid integer simplified sine–cosine transforms of type VII of order 16 based on two transforms of order 8: sine type VII and cosine type II has been developed. Two integer single-norm simplified sine–cosine transforms of low computational complexity have been developed, with it being 38.6% lower than that of the well-known algorithm for discrete sine transform of type VII of order 16 with integer approximation, which is used in the modern video coding standard VVC (Versatile Video Coding). A fast algorithm for calculating the inverse integer cosine–sine transform of type IV of order 16 with low multiplicative complexity has been developed. A method for constructing hybrid sine–cosine transform of type II of order 4 has been developed, and three new integer transforms with fast calculation algorithms have been proposed. A method for constructing a multihybrid simplified sine-cosine transform of type II of order 8 has been developed, two new integer transforms have been proposed and for them fast algorithms of low computational complexity without the multiplication operation have been developed. A method for constructing integer modified cosine and sine transforms of type II of order 8 is proposed, on the basis of which integer single-norm modified sine and cosine transforms of type II of order 8 with fast calculation algorithms are developed. The architecture of a hybrid codec using neural networks on the decoder side is considered, which prioritizes the informativeness and minimality of data encoded by traditional methods, since it uses the most generalized input data and applies generative neural networks to restore data and improve visual quality. Within the framework of this approach, a number of new methods and coding algorithms are proposed, in particular, an algorithm for extracting multiscale contours based on the criteria of scale, brightness and connectivity. A new algorithm for obtaining contours based on the sum of vertical and horizontal differences is proposed. It is proposed to transmit contours, simplified palettes, reference points and texture fragments as specialized information using traditional methods for further processing and

approximate restoration on the decoder side by neural network methods. A new measure for error estimation in the color space is proposed based on calculating the error in the proportions of vertical and horizontal differences. A computer technology for image encoding based on 2D fast multiwavelet transforms in the basis of fractal step functions with a multiwavelet package of size 8×8 for three levels of decomposition is developed.

Practical significance of the research results.

The application of the developed approaches, methods and algorithms has practical significance for the development of high-speed traditional and hybrid codecs using neural networks on the decoder side, including for use in the field of the Internet of Things, mobile communications, wearable devices, satellite and space technologies.

The theoretical significance of the results obtained lies in the further development of the theory of orthogonal transformations, in particular integer ones.

The research results were presented in the following topics of the V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine: VF 140.21, VKM 140.25.22, VP. 140.19, VP. 140.21.

9 publications were made, of which one article indexed in Scopus databases, two articles in professional journals and two articles published in proceedings of international conferences indexed in Scopus databases.

Keywords: adaptive modes, adaptive coding, mathematical modeling, fast algorithms, high-speed codec, codec architecture, integer transforms, simple cosine transforms, sine-cosine transforms, cluster analysis, progressive data transmission.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у періодичних наукових виданнях, проіндексованих у базах даних Web of Science Core Collection та/або Scopus

1. Луц Я.В. Швидке цілочислове спрощене синус–косинусне перетворення типу VII порядку 16 для відеокодування. Кібернетика і системний аналіз. 2025. Т. 61, № 2. С. 176-190. https://doi.org/10.34229/KCA2522-9664.25.2.15

Статті у періодичних наукових виданнях, включених на дату опублікування до переліку наукових фахових видань України за спеціальністю 113

1. Luts Ya.V., Luts V.K. About the Development of a High-Speed Simplified Image Codec. Cybernetics and Computer Technologies. 2021. 1. P. 61–66. (in Ukrainian) https://doi.org/10.34229/2707-451X.21.1.6
2. Луц В.К., Луц Я.В. Про використання модифікованого дельта-алгоритму для обробки та кодування зображень. Фізико-математичне моделювання та інформаційні технології, (36), 148-152. (2023). https://doi.org/10.15407/fmmit2023.36.148

Публікації на конференціях, що проіндексовані у базах даних Scopus

1. Hnativ L.O., Luts Ya.V. Discrete Cosine-Sine Transform and Fast 16-point Integer Cosine-Sine Transform Low Complexity for Video Coding. Proc. SPIE 13137, Applications of Digital Image Processing XLVII, 1313716 (30 September 2024) Optical Engineering + Applications, 2024, San Diego, California, United States, https://doi.org/10.1117/12.3035173

2. Hnativ L., Luts V. and Luts Ya. Fractal step multiwavelets and multiwavelet packets - a new multiwavelet technology for image processing and coding // Selected Papers of the III International Scientific Symposium “Intelligent Solutions” (IntSol-

2023). Symposium Proceedings, Kyiv - Uzhhorod, Ukraine, September 27-28, 2023. – 2023. – P. 282–294. https://ceur-ws.org/Vol-3538/

Публікації на конференціях

1. Луц В.К., Луц Я.В. «Адаптивні режими кодеків зображень», X Міжнародна науково-практична конференція Глушковські читання", м. Київ, 2 грудня 2021. С. 113-117 URI: http://glushkov.kpi.ua/GC21
2. Луц Я.В. Новий метод отримання контурів зображень. Міжн. наук.-пр. конф. «Інформаційні технології та комп'ютерне моделювання», 15 – 16 грудня 2022 р., Ів.-Франківськ. С.57-60. https://itcm.comp-sc.if.ua/2022/zbirnuk-2022.pdf
3. Луц Я.В. Новий алгоритм вибору розмірності перетворення при кодуванні зображень // Матеріали Міжнародної науково-практичної конференції "Інформаційні технології та комп'ютерне моделювання" (6-8 липня 2023 року). Івано-Франківськ. – 2023. – С. 65–66 https://itcm.comp-sc.if.ua/2023/zbirnuk-2023.pdf
4. Луц Я.В. Спосіб прогресивної передачі для кодеків зображень. IX наукова конференція «Прикладна математика та комп`ютинг», 14 листопада 2020. https://pmk.kpi.ua

ЗМІСТ

Вступ ……. 14
1 Аналіз і порівняння підходів кодування зображень ……..…..……23
1.1 Огляд відомих підходів кодування зображень …….……23
1.2 Застосування нейромереж для стиснення і відновлення зображень/кадрів …….…..42
1.2.1 Гібридні підходи: нейронні мережі і традиційні кодеки ……46
1.2.2 Нейромережеві методи стиснення даних ……..……...49
1.2.3 Застосування інтелектуальних нейромереж для розробки нових підходів кодування зображень/кадрів……..…….60
1.3 Висновки до розділу ……61
2 Концепція гібридного кодека: нові методи та алгоритми….. ….……...63
2.1 Задача мінімізації вхідних даних для нейромереж……...…….64
2.2 Спрощений алгоритм формування контурів……....…….65
2.3 Спрощена мода роздільних перетворень…….…..…69
2.4 Реперні точки і фрагменти текстур …….…..…..70
2.5 Похибка пропорцій вертикальних і горизонтальних різниць… …..…71
2.6 Висновки до розділу…….…72
3 Гібридні ортогональні перетворення …..…….……...…74
3.1 Основні поняття гібридних перетворень.. ……..…..…..75
3.2 Цілочислове косинус-синусне перетворення типу IV порядку 16 …..77
3.2.1 Метод побудови гібридного косинус-синусного перетворення типу IV порядку N …….…..…….78
3.2.2 Метод побудови цілочислового гібридного косинус-синусного перетворення типу IV порядку 16…….……80
3.2.3 Алгоритм швидкого обчислення косинус-синусного перетворення типу IV порядку 16…….…….…88
3.2.4 Експериментальні результати тестування……..……..…..90
3.3 Цілочислове спрощене однонормове синус-косинусне перетворення порядку 16……..…….96

3.3.1 Метод побудови цілочислового спрощеного однонормового синус-косинусного перетворення порядку 16 …….……..96

3.3.2 Алгоритм швидкого обчислення цілочислового спрощеного синус-косинусного перетворення порядку 16 ……..……...101

3.3.3 Експериментальні результати тестування ……...107

3.4 Цілочислове синусно-косинусне перетворення типу II порядку 4 ...113

3.5 Висновки до розділу ……...…….…….....121

4 Мультигібридні ортогональні перетворення ..…….…….123

4.1 Спрощене цілочислове однонормове синусно-косинусне перетворення типу II порядку 8……..124

4.2 Цілочислове однонормове модифіковане синусне перетворення типу II порядку 8 ……..……130

4.3 Цілочислове однонормове модифіковане косинусне перетворення типу II порядку 8 ……...……..……132

4.4 Висновки до розділу ……...……134

5 Дискретне мультивейвлет-перетворення на основі мультивейвлет-пакета розміру 8×8 ……..……....135

5.1Дискретне мультивейвлет-перетворення……...……136

5.2 Швидке мультивейвлет-перетворення……..……136

5.3 Метод побудови дискретного мультивейвлет-перетворення на основі мультивейвлет-пакета розміру 8×8 …….……..139

5.4 Алгоритми швидкого обчислення 8-точкових дискретних мультивейвлет-перетворень……...……141

5.5 Двовимірне дискретне мультивейвлет-перетворення……..……144

5.6 Мультивейвлет-метод кодування зображень на основі трьохрівневого двовимірного (2D) ШМВП з мультивейвлет-пакетом розміру 8×8 ……..….148

5.7 Висновки до розділу……...153

Висновки 154

Перелік джерел посилання.....156

Перелік скорочень, умовних познак, одиниць і термінів

ДКП – дискретне косинусне перетворення

ДП – досконалі перестановки

ДІП – двійково-інверсні перестановки

ПАХ – перетворення Адамара – Хаара

ПВК – перетворення високої кореляції

ПНК – перетворення низької кореляції

ПКЛ – перетворення Карунела-Лоєва

ПКГ – перестановки по коду Грея

СХП – слент-Хаара перетворення

СП – слент-перетворення

СКВ – середньоквадратичне відхилення

ЦКП – цілочислове косинусне перетворення

AVIF – AV1 Image Format, формат зображень на основі кодека відео AV1, має режими з втратами і без втрат

FLIF – Free Lossless Image Format, кодек зображень без втрат, використовує алгоритми передбачення значень пікселів і ентропійне кодування

H.265/HEVC – стандарт, формат стиснення відео

HDR – High Dynamic Range, збільшення діапазону яскравості

HEIF, HEIC, BPG – кодеки зображень на основі відеокодека H.265/HEVC

IoT – Internet of Things, інтернет речей

JPEG 2000 – кодек зображень на основі вейвлетів

JPEG XR – формат кодування зображень, розроблений корпорацією Microsoft

JPEG – найбільш популярний растровий формат кодування зображень з втратами, розроблено в 1995 році

JPEG-LS – версія формату JPEG без втрат

Raw – (англ. raw – сирий) –розповсюджений формат даних для нестиснених зображень, отриманих цифровими камерами та сканерами, що не має чіткої специфікації.

RDO – rate-distortion optimization, метод пошуку компромісу між бітрейтом та спотвореннями відео

RGB – адитивна кольорова модель, що кодує кольори за допомогою трьох складових каналів кольорів (Red, Green, Blue)

SAD – сума абсолютних різниць значень пікселів двох зображень чи кадрів

SATD – сума абсолютних різниць коефіцієнтів перетворень

SSIM – structure similarity, метрика порівняння зображень

VP9 – відкритий стандарт кодування відео від корпорації Google

VVC/H.266 – Versatile video coding, стандарт кодування відео

WebP – стандарт компресії зображень з втратами та без втрат на основі відеокодека VP8

YCbCr – кольоровий простір, в якому Y – компонент яскравості, Cb і Cr – компоненти кольоровості, які представляють різницю між синім кольором і яскравістю (Cb) та різницю між червоним кольором і яскравістю (Cr)

QP – параметр квантування

PSNR – пікове співвідношення сигнал/шум

PNG – Portable Network Graphics, формат зображень з втратами та без втрат, оптимальний для інфографіки

PSNR-HVS-M – пікове співвідношення сигнал/шум з урахуванням функції контрастної чутливості (CSF) та міжкоефіцієнтного маскування контрасту DCT

HVS - Human Vision System

ВСТУП

Актуальність. Сучасний світ стрімко рухається в напрямку цифровізації, а обсяг візуального контенту, що генерується та передається щодня, зростає експоненційно. Високоякісне відео, зображення, потокове мовлення, відеоконференції, системи відеоспостереження та штучного інтелекту – усе це потребує ефективного зберігання та передачі візуальних даних. Розвиток технологій 4K, 8K, 360°-відео, доповненої (AR, **A**ugmented **R**eality) та віртуальної (VR, **V**irtual **R**eality) реальності створює безпрецедентне навантаження на мережеву інфраструктуру та обчислювальні ресурси, що робить проблему стиснення даних однією з ключових у сучасних інформаційних технологіях. Для стиснення зображень та відео застосовуються традиційні кодеки, при цьому тривають дослідження нейромережевих підходів кодування.

Традиційні методи кодування зображень і відео забезпечили суттєві покращення ефективності стиснення. Однак навіть найсучасніші кодеки не можуть повністю задовольнити зростаючі вимоги до компресії, особливо в умовах обмеженої пропускної здатності або низького енергоспоживання (IoT, мобільні пристрої, космічні апарати). У зв'язку з цим зростає інтерес до альтернативних методів, зокрема до нейромережевого кодування, яке показало потенціал покращення традиційних підходів. Також досліджують можливості оптимізації та зниження обчислювальної складності традиційних кодеків.

Попри деякий прогрес, дослідження нейромережевих кодеків стикається з низкою викликів:

- Висока обчислювальна складність. Нейромережі потребують значно більше ресурсів для кодування та декодування порівняно з класичними кодеками, що суттєво обмежує їхнє використання, особливо в реальному часі та на пристроях із низькою потужністю.
- Проблеми стандартизації. Традиційні кодеки мають чітко визначену структуру та параметри, тоді як нейромережеві підходи часто

залежать від конкретної архітектури нейромережі та навчального набору.

- Стабільність та узагальнюваність. Нейромережеві моделі можуть демонструвати чудові результати на тренувальних даних, але зазнавати труднощів при роботі з новими типами контенту.

Таким чином, виключно нейромережеві підходи внаслідок їх високої обчислювальної складності на даний час залишаються перспективними для подальших досліджень, але більшу актуальність представляють гібридні підходи, на базі комбінації традиційних кодеків і нейромереж.

Серед гібридних підходів є два основних напрямки – застосування нейромереж одночасно для кодування і для декодування, і застосування нейромережевих методів лише для декодування. Останній варіант є значно менш обчислювально складним, і має найвищу актуальність на даний час. В цілому дослідження, спрямовані на розробку, оптимізацію та впровадження нових гібридних кодеків, мають ключове значення для подальшого розвитку цифрових технологій. Це важливо не тільки для традиційних сфер (стрімінг, зберігання відео), але й для високоспецифічних застосувань, таких як IoT, медицина, космічні місії та інтелектуальні відеоаналітичні системи.

Швидкий розвиток інтернету речей та пріорітет кодування вхідних даних над декодуванням для окремих галузей є важливими факторами підвищення актуальності швидкодійних підходів, методів і алгоритмів кодування і стиснення зображень/кадрів з втратами [1-12]. Для цих задач можуть адаптивно використовуватись як традиційні кодеки зображень та відео, так і гібридні кодеки з додатковим застосуванням нейромережевих методів.

Розглянуто концепцію нового гібридного кодека, що включає в себе наступні важливі особливості:

- мінімальну обчислювальну складність на стороні кодера з застосуванням традиційних, але швидкодіючих алгоритмів;
- пошук і кодування суттєвої спеціалізованої інформації для відновлення нейромережевими методами на боці декодера;

- значний коефіцієнт стиснення даних, розрахований на подальшу обробку й відновлення нейромережею на стороні декодера;
- гібридне застосування двох типів нейромереж, для наближеного відновлення даних і генерації даних.

Іншим напрямком є застосуваня аналогічних підходів кодування специфічної інформації (контури, палітри, реперні точки) в рамках традиційних кодеків, що потребує дослідження і розробки додаткових алгоритмів відновлення, реконструкції вхідних даних.

В роботі представлено результати розробок та досліджень нових ортогональних перетворень, які можливо застосувати не тільки для традиційних кодеків, але й для гібридних, на основі традиційних методів кодування з додатковим застосуванням нейромереж.

Об'єкт дослідження – підходи, методи і алгоритми кодування зображень та відеокадрів з втратами, аналізу та оброблення зображень і кадрів.

Мета роботи – дослідження та розробка підходів, методів, алгоритмів кодування і оброблення зображень та відеокадрів.

Мета роботи обумовлює дослідження перспектив розвитку методів і алгоритмів аналізу та кодування зображень, зокрема з урахуванням розвитку нейромереж та штучного інтелекту. Важливим напрямком роботи є розробка нових методів та алгоритмів, в тому числі нових ортогональних перетворень, серед яких найбільша увага приділяється цілочисловим однонормовим гібридним і мультигібридним перетворенням на основі базових перетворень різних типів, зокрема розроблено синусно-косинусні, модифіковані косинусні та інші комбінації відомих та отриманих перетворень. Окрема увага приділяється спрощеним гібридним та мультигібридним ортогональним перетворенням, які ефективні для швидкодійного кодування як традиційними кодеками, так і гібридними.

Методи дослідження. При розробці й дослідженні нових підходів, методів і алгоритмів кодування та стиснення зображень використовувалися підходи й методи теорії апроксимації і наближення, лінійної алгебри,

чисельного аналізу, комбінаторики, статистичного й математичного моделювання, теорії інформації та кластерного аналізу.

Наукова новизна отриманих результатів – більшість запропонованих підходів, методів та алгоритмів розглянуто вперше.

Розглянуто архітектуру гібридного кодека з використанням нейромереж на стороні декодера, що має пріорітетом інформативність та мінімальність закодованих традиційними методами спеціалізованих даних, що використовує найбільш узагальнені вхідні дані та застосовує генеративні нейромережі для відновлення даних та підвищення візуальної якості. В рамках цього підходу запропоновано ряд нових методів та алгоритмів кодування для зменшення обчислювальної складності на стороні кодера. Запропоновано новий спрощений режим роздільного кодування з використанням найбільш поширеної та ефективної комбінації – косинусного і синусного перетворень. Запропоновано нову міру для оцінки похибок в просторі кольорів на основі обчислення похибки пропорцій вертикальних та горизонтальних різниць між сусідніми пікселями. Розроблена концепція побудови мультигібридних ортогональних перетворень, яка дозволяє будувати нові перетворення на базі відомих двох класів. Запропоновано два методи побудови мультигібридних перетворень, на основі яких розроблено три нових цілочислових перетворення різних класів. Розроблено матричний метод побудови гібридних цілочислових спрощених синус–косинусних перетворень типу VII порядку 16 на основі двох перетворень порядку 8: синусного типу VII і косинусного типу II. Розроблено двоє гібридних цілочислових однонормових спрощених синус-косинусних перетворень низької обчислювальної складності, яка менше на 38,6% порівняно з відомим алгоритмом дискретного синусного перетворення типу VII порядку 16 з цілочисловою апроксимацією, що застосований у сучасному стандарті відеокодування VVC. Розроблено швидкий алгоритм обчислення оберненого цілочислового косинус-синусного перетворення типу IV порядку 16 низької мультиплікативної складності, і отримано оцінки його обчислювальної складності, яка у 8,25 разів менша і потребує на 115% більше операцій

додавання порівняно зі швидким алгоритмом дискретного косинус-синусного перетворення. Розроблено метод побудови гібридного синусно-косинусного перетворення типу II порядку 4, і запропоновано три нових цілочислових перетворення зі швидкими алгоритмами обчислення. Розроблено метод побудови спрощеного синусно-косинусного перетворення типу II порядку 8, запропоновано два нових цілочислових перетворення і розроблено для них швидкі алгоритми низької обчислювальної складності без виконання операції множення. Запропоновано метод побудови цілочислових модифікованих косинусних і синусних перетворень типу II порядку 8, на основі якого розроблено цілочислові однонормові модифіковані синусні та косинусні перетворення типу II порядку 8 зі швидкими алгоритмами обчислення. Розроблені спрощені синусно-косинусні та модифіковані косинусні перетворення можуть бути альтернативою відомому синусному перетворенню типу VII для адаптивного застосування в збалансованих і швидкодійних режимах кодування, в тому числі для застосування в запропонованому гібридному кодеку.

Розроблено комп'ютерну технологію кодування зображень на основі 2D швидких мультивейвлет-перетворень в базисі фрактальних ступінчастих функцій з мультивейвлет-пакетом розміру 8×8 для трьох рівнів розкладу. Мультивейвлет метод кодування на основі швидких мультивейвлет перетворень має низьку мультиплікативну складність, яка в 105 разів менша порівняно з відомим класичним вейвлет-методом для фільтрів з K=8 ненульовими коефіцієнтами.

Практичне значення результатів дослідження. Застосування розроблених підходів, методів і алгоритмів має велике практичне значення для розробки кодеків низької обчислювальної складності, як традиційних, так і гібридних з використанням нейромереж на стороні декодера. Це важливо для кодування в області інтернету речей, мобільного зв'язку, носимих пристроїв, супутникових та космічних технологій, а також збалансованого кодека для збереження та передачі зображень та відео в мережі інтернет.

Теоретичне значення отриманих результатів полягає в подальшому розвитку теорії ортогональних перетворень, зокрема отримано нові спрощені ортогональні цілочисельні однонормові перетворення низької обчислювальної складності, а також модифіковані косинусні та синусні перетворення II типу. Розширення сімейства відомих і мультигібридних перетворень можливо також шляхом використання комбінаторного перебору різноманітних пропорцій параметрів перетворень. В рамках цього підходу представляють цікавість ортогональні перетворення, що реалізуються розрідженими матрицями і тому мають меншу обчислювальну складність.

Особистий внесок здобувача. Всі положення й результати дисертаційної роботи, що виносяться на захист, отримано автором самостійно. У роботах, які опубліковані в співавторстві, здобувачеві належать наступні результати: у [1] запропоновано новий алгоритм прогресивної передачі даних для спрощеного кодека зображень; у [2] запропоновано новий алгоритм отримання контурів зображення, у [3] розроблено швидкий алгоритм обчислення оберненого цілочислового косинус-синусного перетворення порядку 16 типу IV; у [4] розроблена комп'ютерна технологія вейвлет-методу кодування зображень на основі 2D швидких мультивейвлет-перетворень в базисі фрактальних ступінчастих функцій з мультивейвлет-пакетом розміру 8×8 для трьох рівнів розкладу.

Апробація результатів дисертації. Результати дисертації доповідались на:

- міжнародному науковому симпозіуму «Питання оптимізації обчислень (ПОО-XLVIII)», присвяченому 100-річчю від дня народження академіка В.М. Глушкова, м. Львів, 19–22 вересня 2023 р.;
- міжнародній науково-практичній конференції «Інформаційні технології та комп'ютерне моделювання», ІТКМ-2022, 15 – 16 грудня 2022 р., м. Івано-Франківськ;

- X міжнародній науково-практичній конференції “Глушковські читання”, м. Київ, 2 грудня 2021 р.;
- науковому семінарі "Методи обчислювальної математики" Інституту кібернетики імені В.М. Глушкова НАН України, 1 квітня 2025 р., керівник семінару: академік НАН України В.К. Задірака.

Публікації. Основні наукові результати дисертаційної роботи викладено в 9 публікаціях, з яких: дві статті опубліковано в фахових журналах України, одна стаття опублікована англійською мовою в науковому журналі видавництва Springer та проіндексована в наукометричній базі Scopus, шість публікацій в збірниках матеріалів міжнародних та всеукраїнських наукових конференцій та симпозіумів, з них дві статті - проіндексовано в наукометричній базі Scopus.

Зв'язок роботи із науковими програмами. Дисертація пов’язана із тематикою, що виконується у відділі оптимізації чисельних методів Інституту кібернетики ім. В.М. Глушкова НАН України, відповідно частина результатів дисертації знайшли відображення в наукових темах: ВФ 140.18 «Розробити теоретичні основи побудови базисних систем нового класу мультивейвлетів для оброблення сигналів і зображень» № держреєстрації 0119U002275; ВКМ 140.24.21 «Розробити методи і алгоритми розподіленого моделювання та визначення станів віддалених об'єктів тривалого моніторингу на основі суперкомп'ютерних технологій», № держреєстрації 0121U110720; ВП.140.21 «Розробити ефективні методи та алгоритми оброблення, кодування і аналізу сигналів для дистанційного моніторингу функціональних станів осіб», № держреєстрації 0122U000830.

Структура дисертації.

В **першому розділі** зроблено огляд відомих підходів, методів і алгоритмів кодування зображень, включаючи застосування нейромереж, проведено аналіз і зроблено висновки про їх недоліки, актуальність і перспективи.

В **другому розділі** розглянуто новий гібридний кодек, що включає традиційне кодування на стороні кодера, але з використанням швидкодіючих

методів та алгоритмів, значне стиснення специфічної інформації для відновлення та додаткової генерації втрачених даних засобами нейромереж. Основною особливістю даного підходу є пошук і кодування спеціальної інформації, яка використовується нейромережами для приблизного відновлення початкових даних на стороні декодера.

В **третьому розділі** розглядаються гібридні ортогональні перетворення, наводяться отримані нові цілочислові однонормові перетворення різної розмірності: спрощене синусно-косинусне типу VII порядку 16, синусно-косинусне перетворення типу II порядку 4 та косинусно-синусне перетворення типу IV порядку 16. Всі перетворення мають швидкі алгоритми низької обчислювальної складності. Спрощене синусно-косинусне типу VII порядку 16 пропонується в якості альтернативи синусному перетворенню типу VII порядку 16 для адаптивних швидкодійних режимів кодування.

В **четвертому розділі** описано нові мультигібридні цілочислові ортогональні перетворення для адаптивного застосування: спрощене синусно-косинусне перетворення типу II порядку 8 на основі двох гібридних синусно-косинусних перетворень типу II порядку 4, модифіковане синусне і косинусне перетворення типу II порядку 8 на базі відомих синусного і косинусного перетворень з заміною складового перетворення порядку 2. Розглянуто умови і доцільність їх застосування, зокрема для запропонованого гібридного швидкодіючого кодека, приведено результати експериментального тестування і обчислювальна складність розроблених швидких алгоритмів. Модифіковане косинусне перетворення типу II порядку 8 запропоновано як якісна альтернатива синусному перетворенню типу VII порядку 8, а спрощене синусно-косинусне перетворення типу II порядку 8 – в якості швидкодіючої альтернативи синусному перетворенню типу VII порядку 8. Ці розроблені перетворення можуть застосовуватись адаптивно.

В **п’ятому розділі** дисертації описано мультивейвлет-метод кодування зображень на основі трьохрівневого двовимірного (2D) швидкого дискретного мультивейвлет-перетворення з мультивейвлет-пакетом розміру 8×8. Даний

метод можна використовувати як для кодування, так і для аналізу зображень, наприклад, для відомого методу мультифрактального аналізу.

Структура і об’єм роботи. Дисертація складається із вступу, 5 розділів, загальних висновків, списку літератури зі 118 найменувань. Загальний обсяг дисертаційних досліджень викладено на 164 сторінках друкованого тексту, де обсяг основного тексту – 141 сторінка, зокрема 15 рисунків, 26 таблиць.

РОЗДІЛ 1. ОГЛЯД ВІДОМИХ ПІДХОДІВ, МЕТОДІВ І АЛГОРИТМІВ КОДУВАННЯ ЗОБРАЖЕНЬ

1.1 Огляд відомих підходів кодування зображень

Цифрове зображення – масив даних, отриманий шляхом дискретизації (аналого-цифрового перетворення) оригіналу. Після кодування за допомогою певного алгоритму і запису на носій, цей масив даних стає файлом. Цифрові зображення за способом дискретизації оригіналу поділяються на векторні, растрові та змішаного типу.

Векторні зображення – зображення, що були створені за допомогою сукупності графічних примітивів (точок, ліній, кривих, полігонів).

Растрові зображення – зображення, що були створені за допомогою растрів (сіток) пікселів.

Зображення змішаного типу – зображення, що створені як за допомогою матриць пікселів, так і за домогою графічних примітивів, векторів та кривих Безьє. Найбільш перспективний формат даних.

Кожен тип зображень має свої переваги та недоліки. Найпоширенішим типом зображення, і водночас типом зображень, які розглядаються в даній роботі, є растровий тип.

Загальна схема представлення растрового зображення у пам’яті – у вигляді матриць даних пікселів, як представлено на рис. 1. В залежності від використовуваної колірної моделі та обраних розробником або користувачем параметрів, кожному пікселу відповідає декілька значень. Скажімо, у моделі RGB кожен піксел визначається трьома значеннями – характеристиками червоного (Red), зеленого (Green) та синього (Blue), які комбінуються і утворюють той колір, що найбільше відповідає кольору оригіналу. До цих трьох значень може бути додано четверте – значення так званого каналу прозорості (альфа-каналу), яке визначає рівень прозорості піксела. У такому випадку модель RGB (Red, Green, Blue) перетворюється у RGBA (Red, Green, Blue,

Alpha). Відповідно, одному зображенню може відповідати декілька матриць значень пікселів.

Серед широко використовуваних моделей кольорів відзначаються YCbCr (Luma, Blue-difference, Red-difference), який представляє зображення через яскравість (Y) і дві кольорові компоненти (Cb, Cr) і використовується у кодеках зображень і відеокодеках (JPEG, MPEG, H.264, H.265), дозволяючи ефективне стиснення; RGB і RGBA, які використовуються в моніторах, камерах та графічних редакторах; CMYK (Cyan, Magenta, Yellow, Black) та CMYKA, що є субтрактивною моделлю і використовується у друкуванні; HSV (Hue, Saturation, Value) і HSVA, що відокремлює відтінок (H), насиченість (S) та яскравість (V), забезпечуючи зручність корекції кольорів, і є популярною в графічних редакторах та комп'ютерному зорі, та інші.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

Рисунок 1 Представлення растрового зображення у пам’яті

При роботі із зображенням у цифровому форматі воно майже ніколи не передається у вигляді оригіналу, оскільки це неефективно з точки зору використання пам’яті. Замість оригіналу використовуються дані про зображення у стисненому за допомогою процесу кодування вигляді, які за допомогою програмного забезпечення декодуються і подаються на екран у відновленому вигляді. При цьому якість відновленого зображення залежить від типу і параметрів кодування.

Кодування зображень – процес перетворення зображення у компактну форму для зберігання або передачі із використанням певного алгоритму стиснення [5].

Декодування зображень – зворотний процес, що відновлює зображення з його закодованої форми, зазвичай із можливою втратою якості (у випадку із втратним стисненням) [6].

Кодек зображень – це програмний або апаратний алгоритм, що здійснює кодування (стиснення) та декодування (відновлення) зображень, часто з метою ефективного зберігання чи передачі. Кодеки можуть бути з втратами (JPEG, WebP) або без втрат (PNG, FLIF) [6-7].

Як показано на рис. 2, кожен кодек на вході приймає оригінальне зображення, здійснює його компресію і зберігає у стисненому вигляді. Ці дані потім за допомогою цього ж (або сумісного) кодека через процес декомпресії даних перетворюються у відновлене зображення, подібність якого до оригіналу визначається кодеком: так, кодек без втрат теоретично відновлює оригінал із стовідсотковою точністю, а от кодек зображення із втратами відновить зображення із певною похибкою, що залежить від алгоритму та параметрів кодування.

Внаслідок різноманіття задач та умов використання кодеків створюється безліч форматів, що забезпечують особливі комбінації параметрів: якості відновленого зображення, розміру файлу із стисненими даними, швидкості роботи та сумісності.

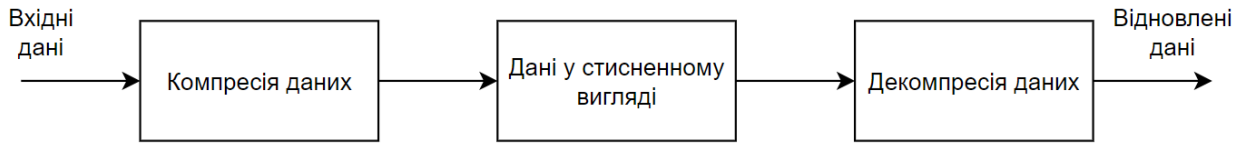

Рисунок 2 Найпростіша схема кодека

Різні алгоритми кодування зображень створені для різних сценаріїв – від швидкого стиснення до високоточної передачі даних. Їх класифікують за двома

узагальненими типами – алгоритми стиснення без втрат та алгоритми стиснення з втратами.

Алгоритми стиснення з втратами. Дискретні ортогональні перетворення. Алгоритми стиснення з втратами використовуються для значного зменшення розміру файлів за рахунок видалення інформації, менш важливої для людського зору. Вони базуються на перетвореннях сигналу, квантуванні та ентропійному кодуванні і використовуються у переважній більшості сучасних кодеків.

Серед основних видів алгоритмів стиснення із втратами виділяють наступні: алгоритми блочних перетворень; алгоритми вейвлет-перетворень; алгоритми фрактальних перетворень; алгоритми з використанням нейронних мереж глибокого навчання; алгоритми проріджения та квантування даних.

Найпоширенішими є алгоритми блочних перетворень [8-12], що відзначаються високою ефективністю стиснення і низькою обчислювальною складністю. Проте присутній недолік: за великих ступенях стиснення з'являються так звані «блочні артефакти», що помітно знижують якість відновленого зображення. Основна ідея полягає в тому, що просторове представлення пікселів перетворюється у частотне, де низькочастотні коефіцієнти містять основну інформацію, а високочастотні – дрібні деталі, які можна відкинути для стиснення. Загалом ці алгоритми можна описати наступним чином [13-14]:

- Розбиття зображення на блоки (використовуються блоки розмірності 4×4, 8×8, 16×16 та 32×32)
- Перетворення кожного блоку за допомогою дискретних або цілочисельних ортогональних перетворень, що переводить значення пікселів в частотну область
- Квантування коефіцієнтів перетворення – ділення коефіцієнтів на параметр квантування і заокруглення до цілого (менші коефіцієнти округлюються до нуля)

- Ентропійне кодування (різновид алгоритмів стиснення без втрат) отриманих квантованих коефіцієнтів

Вейвлет-перетворення, на відміну від блочних, працює з усім зображенням одночасно, розкладаючи його на різні рівні частот за допомогою вейвлетів. Після розкладу відбувається квантування коефіцієнтів або їх ієрархічне представлення, наприклад, використовуючи алгоритм SPIHT, і наступне ентропійне кодування коефіцієнтів, за необхідності. Вони відзначаються кращою якістю відновленого зображення при великих ступенях стиснення, але мають значні недоліки: по-перше, вони вимагають помітно більше обчислень на піксел, ніж блочні алгоритми [15-16]; по-друге, вони не можуть співпрацювати з алгоритмами передбачення даних, оскільки у випадку значної хаотичності вхідних даних їх ієрархічна структура представлення виявляється недостатньо ефективною.

Алгоритми фрактального стиснення шукають самоподібні області в зображенні та кодують їх трансформаційними коефіцієнтами. Таке стиснення добре масштабується без втрат якості і має високий рівень стиснення, але має критичні недоліки, такі як дуже повільне кодування і обмеженість практичного застосування (їх якість сильно коливається в залежності від виду вхідного зображення) [17-18]. Високоперспективною є ідея побудови мультивейвлетів фрактального типу, як у [4, 19-24], які мають низьку обчислювальну складність і порівняно високу якість відновленого зображення і можуть бути використані для ефективного аналізу зображень.

Замість традиційних перетворень можна використовувати нейромережі для прогнозування і ефективного кодування зображення. У цьому випадку, вхідне зображення пропускається через згорткову нейромережу (так званий «автоенкодер»), генеруються стиснені представлення («латентні вектори») і виконується ентропійне кодування латентних векторів. Такий підхід дозволяє досягти значних результатів у питаннях якості та стиснення у порівнянні із традиційними методами, особливо у випадку навчання нейромережі для

специфічних типів зображень, проте ставить високі вимоги до обчислювальних ресурсів [25-26].

Алгоритми проріджения та квантування даних займають окреме місце серед алгоритмів стиснення із втратами, оскільки вони можуть працювати напряму із значеннями пікселів (а можуть – із їх представленнями у частотній області). Тобто, вони можуть використовуватись як без перетворення сигналу, так і в комбінації з іншими методами. Проріджения – пропускання частини даних для зменшення їх обсягу, а квантування – представлення числових значень із меншою точністю з метою зменшення розміру файлу зображення, зазвичай для цього початкові дані ділять на обране число, параметр квантування, і заокруглюють до цілого.

Розглянемо детальніше застосування алгоритмів блочних перетворень. Оскільки алгоритми квантування коефіцієнтів грають важливу роль у досягненні великих степенів стиснення, суттєвою частиною кодеків зображень, які використовують блочні перетворення, є вибір ортонормованої системи базисних функцій перетворення, адже вони різняться за ефективністю апроксимації даних.

Блочні перетворення можна класифікувати за такими двома характеристиками, як тип і порядок (розмірність). Існуючі типи перетворень включають в себе (але не обмежені цим переліком косинусні, синусні, гібридні: синус-косинусні, косинус-синусні, спрощені: прості косинусні, ступінчасті косинусні, модифіковані: модифіковані синусні, модифіковані косинусні. Окремо стоять такі прості блочні перетворення, як Уолша, Хаара, слент-перетворення та гібридні на їх основі. Водночас як косинусні, так і синусні перетворення мають різні типи, пронумеровані від I до VIII, що мають назву «розширені дискретні синусні / косинусні перетворення» і свої власні області використання [27-43]. Порядок перетворень найчастіше обмежується чотирма варіаціями: порядку 4, 8, 16 та 32, тільки в H.266 застосували перетворення порядку 64.

В середньому найефективніше стискають інформацію косинусні перетворення [6, 44], проте у разі потреби у зниженні обчислювальної складності інші типи блочних перетворень можуть показати себе краще з точки зору фінального співвідношення якості відновленого зображення, розміру стисненого файлу та швидкодії алгоритмів. Серед таких блочних перетворень можна зазначити перетворення Уолша, Хаара, Уолша-Хаара, слент-перетворення, прості косинусні та інші типи спрощених перетворень.

Окрім дискретних перетворень можуть з метою підвищення швидкодії використовуватись і цілочисельні перетворення, що оперують цілочисельними апроксимаціями замість чисел з плаваючою комою. Однак назва «цілочисельні перетворення» не набула поширення і інколи, коли мова йде про цілочисельні косинусні перетворення (ЦКП), помилково використовується термін дискретні косинусні перетворення (ДКП).

Задля підвищення швидкодії алгоритмів блочних перетворень були розроблені однонормові цілочисельні перетворення, які дозволяють зменшити обчислювальну складність процесу нормалізації коефіцієнтів перетворення. Також з метою заміни при нормуванні операції ділення на зсув почали використовувати числа степені двійки [8-11].

Алгоритми стиснення без втрат. Передбачення та ентропійне кодування. Алгоритми стиснення без втрат, як зрозуміло з їх назви, дозволяють відновити вихідні дані без втрат якості. Є як випадки, коли вони використовуються без алгоритмів стиснення з втратами задля якнайкращого збереження вхідних даних (у медицині, науці тощо), так і випадки, коли вони комбінуються із алгоритмами стиснення з втратами задля вищого результуючого степеня стиснення.

Ці алгоритми поділяються на два основні типи: алгоритми передбачення та ентропійні. Алгоритми передбачення використовують залежності між сусідніми пікселами для зменшення надлишковості: замість зберігання абсолютного значення піксела кодується його різниця від передбаченого значення, що дає менші значення, які краще стискаються. Ентропійні ж

алгоритми використовують статистичні властивості даних для їх ефективного кодування. Один з основних ентропійних алгоритмів, арифметичного кодування, змінює довжину коду представлення чисел залежно від імовірності (частоти) значень. Сучасні кодеки у своїй більшості поєднують використання алгоритмів передбачення та ентропійних.

Серед алгоритмів ентропійного стиснення вирізняють наступні типи: кодування Хаффмана, арифметичне кодування, кодування Шеннона-Фано та кодування словникового типу. Кодування Хаффмана використовується у JPEG разом з дискретними косинусними перетвореннями, PNG (як один із алгоритмів ентропійного кодування) та ZIP / GZIP. Арифметичне кодування вважається кращим за Хаффмана, оскільки має оптимальнішу довжину коду, і використовується у сучасних кодеках. Кодування Шеннона-Фано схоже на Хаффмана, але є ще менш ефективним і тому зараз майже не застосовується. Кодування словникового типу використовується у PNG, GIF, ZIP / GZIP / BZIP2.

Важливим з історичної точки зору моментом було застосування кодеком H.264 в 2003 році алгоритмів передбачення значень пікселів блоку на основі значень сусідніх блоків зображення [44]. При цьому використовувалося в залежності від обраної розмірності перетворення від 4 до 9 різних алгоритмів передбачення, з яких адаптивно обирався один найкращий. Потрібно зазначити, що з кожним наступним поколінням кодеків кількість алгоритмів передбачення все зростає (від 9 у H.264 до 35 у H.265 і 67 у H.266), що дозволяє збільшити коефіцієнт стиснення, але значно збільшує кількість обчислень.

У майбутньому можна очікувати появу нових алгоритмів передбачення, які будуть базуватися не на блоках, а на різноманітних геометричних формах, природно наявних у зображеннях. Один із можливих підходів до створення таких алгоритмів – застосування нейромереж.

Деякі алгоритми стиснення із масштабуванням дозволяють природним чином використовувати свої особливості для прогнозування. Скажімо, кодек LCEVC [45] дозволяє використовувати значення пікселів зображень меншого

масштабу для передбачення, і це дає виграш у обчислювальній складності, оскільки не потрібно адаптивно обирати один алгоритм передбачення із набору.

Напрямки оптимізації алгоритмів кодування можна описати наступним чином: по-перше, зменшення часу обчислень та зменшення використання пам’яті (підвищення швидкодії і зменшення апаратного навантаження); по-друге, підвищення ефективності стиснення; по-третє, покращення якості відновленого зображення.

Кожна область практичного застосування висуває свої вимоги до оптимізації та зосереджує пріоритет на тому, що є важливим для неї, породжуючи велику кількість спеціалізованих кодеків. Так, веб-браузери вимагають швидкого завантаження зображень, тож для кодування використовуються кодеки з більш високою швидкодією, такі як JPEG, PNG, WebP. Водночас, задачі зберігання та архівування даних, наприклад, iCloud, архіви наукових даних, медичні знімки та друк мають високий стандарт ефективності стиснення, і для них більше підходять такі кодеки, як HEIC (H.265-based), JPEG 2000, BPG, FLIF тощо. А у випадку, коли є потреба у якнайкращому відновленні оригінального зображення, скажімо, у фотографії, медицині, кіно та для наукових даних використовуються кодеки із високою якістю відновлення, наприклад, TIFF, EXR, FLIF та високоякісні режими кодеків JPEG 2000 і HEIF / HEIC.

У кодеках з використанням алгоритмів стиснення із втратами (яких переважна більшість) найбільш визначальною характеристикою ефективності стиснення є параметр квантування коефіцієнтів перетворення. Варто зазначити, що в результаті квантування відкидається частина інформації – найбільше втрачається даних із високою частотою, менше спотворюються дані із низькою частотою і найменший вплив квантування має на дані із середньою частотою. Чим більшим є параметр квантування, тим більшими є втрати даних, тож вибір цього значення є надзвичайно важливою задачею. Так, було створено набор алгоритмів Rate-distortion optimization (RDO) [46-49], який адаптивно обирає параметр квантування у відповідності із поставленою задачею – при наявності

обмежень на коефіцієнт стиснення або бітрейт (швидкість передачі даних) доводиться надавати пріоритет ефективності стиснення і відповідно більш високим значенням параметра квантування, а за відсутності таких обмежень пріоритетним критерієм є рівень похибок і параметр квантування мінімізується. Проблема адаптивного підбору параметру квантування є складною, із різними підходами та критеріями [50-53].

З плином часу вимоги до якості зображень змінюються, як і підходи кодеків до співвідношення між степенем стиснення і якістю. Максимальні роздільна здатність і кількість біт на канал кольору постійно зростають, медіа індустрія прагне надати все більшої якості і з'являються нові апаратні можливості, що дозволяють збільшувати точність кольорів до 10-12 біт на канал кольору (тобто замість $\left(2^8\right)^3 = 16\,777\,216$ кольорів один піксел може мати до $\left(2^{12}\right)^3 = 68719476736$ кольорів) і роздільну здатність до 8-16К. Якщо більш ранні кодеки, наприклад, JPEG, використовували матрицю квантування для відкидання даних із високою частотою, як це показано на рис. 3, що призводить до гарантованої втрати якості, то стандарт H.265 почав адаптивно застосовувати один параметр квантування до всіх коефіцієнтів перетворення [26] із метою збереження максимальної кількості даних.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

Рисунок 3 Матриця квантування коефіцієнтів перетворення JPEG

Принцип матриць квантування полягає у визначенні степеня стиснення кожного окремого частотного коефіцієнта із блока. При цьому

використовується знання архітектури блока частотних коефіцієнтів після перетворення: у випадку дискретного косинусного перетворення (яке використовується в JPEG) у лівому верхньому куті блоку зосереджуються низькочастотні коефіцієнти, які містять основну інформацію про зображення, а у правому нижньому – високі частоти, які передають інформацію про дрібні деталі, текстуру і шум у зображенні. Відповідно, до низькочастотних коефіцієнтів застосовується низький параметр квантування з метою збереження інформації, а високочастотні – діляться на більший параметр і втрачають значно більше інформації.

Концептуально перші кодеки були досить простими і мали фіксовані алгоритми і параметри стиснення. Однак із прогресом досліджень методів і алгоритмів, а також технічним прогресом, що зменшує обмеження за часом обробки, використовуваною пам’яттю та інш., набула популярності ідея використання не просто декількох послідовних алгоритмів стиснення, а великої кількості опціональних алгоритмів, із яких адаптивно обирається найкращий для певного зображення або частини зображення з урахуванням виставлених критеріїв і вимог. Серед особливо вагомих причин застосування адаптивного стиснення, окрім простого покращення якості або ефективності компресії за рахунок більшого об’єму роботи при кодуванні, потрібно зазначити наступні: по-перше, нерівномірний розподіл інформації у зображенні. Детальні об’єкти вимагають меншого стиснення, щоб уникнути втрат інформації, в той час як однорідні області (небо, фон тощо) можна стискати сильніше без помітних втрат якості. По-друге, у випадку пересилання зображень та відео у реальному часі – обмеження на бітрейт або пам’ять. Адаптивне стиснення може враховувати динамічні зміни у швидкості інтернету і забезпечувати стабільну якість або швидкість. По-третє, різні типи зображень мають різні ефективні підходи [9, 54].

Окрім очевидних переваг використання адаптивного стиснення, таких як вища ефективність стиснення, покращена якість відновлення, використання більш спеціалізованих алгоритмів та стабільність бітрейту для потокового

відео, а також ефективніше використання обчислювальних ресурсів, такий підхід має і свої недоліки. Чим складнішим є алгоритм підбору, тим більше зусиль витрачається на попередній аналіз зображення, що помітно уповільнює роботу кодека, а також зменшує ефективність його енерговитрат з точки зору автоматизованих систем (які мають обмежені ресурси і внаслідок більшої кількості обчислень будуть вимагати частішого обслуговування). Інша пов'язана з цим проблема – для роботи у реальному часі складніші кодеки вимагають потужніших процесорів. Також при збереженні додаткових мета-даних про адаптивні параметри збільшується вимога до RAM. Потрібно пам'ятати, що застосування адаптивного вибору потребує більше кодування, тестування і налаштування, і кодеки з адаптивним вибором є складнішими для впровадження у апаратне прискорення [55].

Алгоритми RDO є прикладом адаптивного підбору алгоритмів стиснення, іншим прикладом є так звані роздільні адаптивні алгоритми.

Суть роздільних адаптивних перетворень полягає у емпіричному переборі усіх можливих комбінацій синусних і косинусних перетворень одного порядку, що значно збільшує обчислювальну складність, але дозволяє обрати оптимальний за степенем стиснення і якістю алгоритм перетворення [56]. Замість просто матричних операцій (2D) використовуються різні комбінації 1D косинусного і синусного перетворень для рядків і стовпців. Незважаючи на додаткову обчислювальну складність, у зв'язку з пріоритетом підвищення ефективності стиснення такі перетворення були розглянуті як перспективний напрямок для досліджень і використані у кодеку H.266 (VVC) [57]. У порівнянні із попереднім поколінням, H.265, було розроблено роздільні перетворення вищих порядків (у H.265 вони обмежені порядком 4), що є більш ефективним з точки зору кількості службової інформації.

Відомі формати кодування зображень можна умовно поділити на три категорії: растрові, векторні, спеціалізовані.

Растрові формати є найбільш поширеними і дана робота також використовує саме їх. Векторні формати найчастіше використовуються при

роботі з текстом і графікою, наприклад, SVG (Scalable Vector Graphics) – XML формат, EPS (Encapsulated PostScript) – векторний формат для друку та графіки, PDF (Portable Document Format) – змішаний формат, який підтримує як растрові, так і векторні зображеня. Серед спеціалізованих форматів можна назвати DICOM – для медичних знімків або DDS (DirectDraw Surface) – для текстур у комп'ютерних іграх.

Растрові ж формати можна поділити на такі три групи: формати без стиснення; формати із стисненням без втрат; формати із стисненням з втратами. Стиснення не використовується у BMP (Bitmap) – найбільш базовому форматі, та у TIFF (Tagged Image File Format) – форматі для фотографії та друку, який підтримує багатошарові зображення. Відомими форматами із стисненням без втрат є PNG (в окремих режимах), JPEG LS та FLIF. Форматами стиснення з втратами, що є найближчими до тематики дисертації, є JPEG, JPEG 2000, WebP, AVIF, HEIC, BPG та інші. Зауважимо, що формат BMP було б доцільно стискати за допомогою найпростіших алгоритмів стиснення без втрат, тобто з мінімальними обчислювальними витратами, оскільки в цьому випадку досягається значний коефіцієнт стиснення.

JPEG (Joint Photographic Experts Group) [58] – це серія стандартів стиснення зображень, що охоплюють як втратні, так і безвтратні методи. Кодеки JPEG широко застосовуються у фотографії, відео, веб-графіці та професійній обробці зображень. Вперше JPEG було представлено у 1992 році і він не підтримував альфа-канал (прозорість) та анімацію, хоча на той час вже були відомі такі можливості.

Старий JPEG (JPEG 1) заснований на роботі з блоками 8×8, що дозволяє ефективно кодувати частотні компоненти. Він використовує блочне кодування, і підтримує кольорові простори YCbCr 4:2:0/4:2:2/4:4:4. Цей формат має високу швидкість кодування-декодування, добре підходить для фотографій та веб-графіки і підтримується усіма пристроями та програмами. Водночас, він спричиняє появу блочних артефактів при високому стисненні, не підтримує глибини кольорів більше 8 біт і не підтримує прозорості.

Йому на зміну послідовно прийшли такі формати тієї ж серії, як JPEG LS (1999), JPEG 2000, JPEG XR (2009) та JPEG XL (2021).

JPEG LS є форматом без або майже без втрат (“near lossless”) і заснований на LOCO-I (Low Complexity Lossless Compression for Images). Він використовує предиктивні і ентропійні алгоритми для стиснення без втрат. Незважаючи на стиснення краще, ніж у PNG, та низьку обчислювальну складність, популярності формат не набув, в тому числі через те, що не надає підтримки стиснення із втратами з високою ефективністю [59].

JPEG 2000 використовує вейвлетне перетворення замість ДКП, підтримує як втратне, так і безвтратне стиснення, підтримує HDR, 16-бітну глибину кольору та канал прозорості, а також має технологію прогресивного декодування, яка дозволяє передавати зображення із різним рівнем деталізації (спочатку розмите, а потім більш чітке). Він надає вищу якість відновленого зображення при тому ж стисненні в порівнянні із своїм попередником JPEG 1, не має блочних артефактів (завдяки використанню вейвлетного перетворення) і добре підходить для медицини, архівування і друку. Однак цей формат погано підтримується у веб-браузерах та пристроях і має велику обчислювальну складність.

JPEG XR розроблений Microsoft і базується на HD Photo і використовує гнучку структуру блочних перетворень (тобто може обирати між перетвореннями порядку 4, 8 та 16). Він підтримує HDR і 16-бітну глибину кольору, а також канал прозорості, проте не набув популярності, майже не підтримується сучасними браузерами та форматами і витіснений форматами HEIF / AVIF.

JPEG XL є найновішим у серії універсальним форматом. Поєднуючи в собі переваги JPEG і PNG, він використовує адаптивне квантування, перцепційне кодування, підтримує режим без втрат та має зворотну сумісність із JPEG. Форматом підтримується 32-бітна глибина кольору, HDR і прозорість, і при однаковому розмірі файлу він надає кращу якість, ніж WebP. Головна проблема формату – його новизна, яка обумовлює його малопоширеність і

погану підтримку у браузерах. Схему роботи JPEG XL можна побачити на рис. 4.

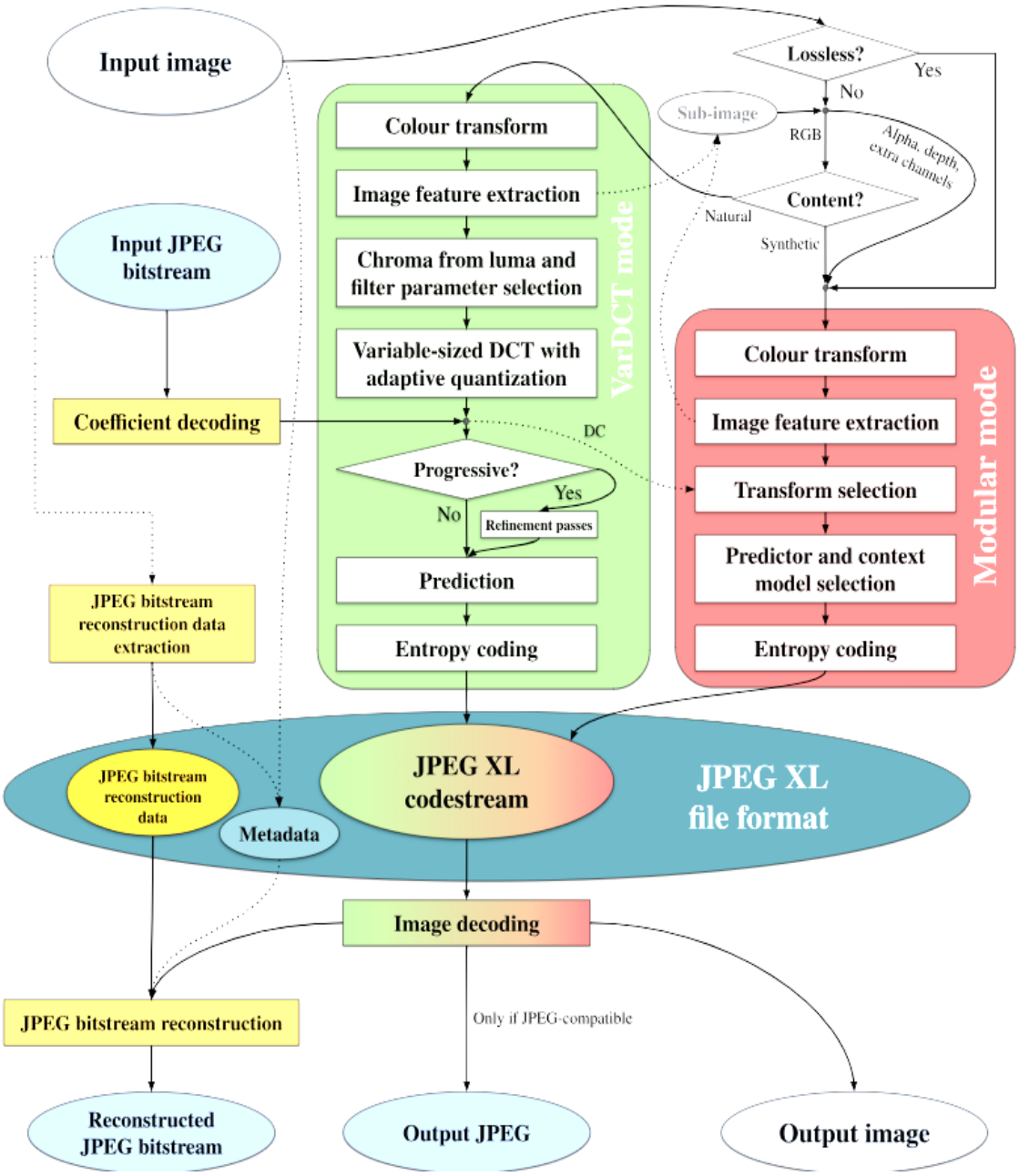

Рисунок 4 Схема роботи JPEG XL

WebP – це формат стиснення зображень, розроблений компанією Google у 2010 році. Його основна мета – зменшити розмір зображень у вебі без значної втрати якості, щоб прискорити завантаження сторінок та знизити споживання трафіку.

Він підтримує режими як стиснення з втратами, так і стиснення без втрат, має канал прозорості і підтримує анімацію, і використовує ефективні алгоритми кодування, засновані на VP8 (відеокодек WebM). У порівнянні із JPEG (JPEG 1) та PNG має на 25-35% менший розмір файлу за тієї ж якості. Кодек використовує предиктивне кодування та ДКП із адаптивним кодуванням у режимі стиснення з втратами, що покращує стиснення у порівнянні із попередниками, але збільшує обчислювальну складність.

У порівнянні з новішими кодеками, такими як AVIF та JPEG XL, WebP програє у ефективності стиснення, але його декодування працює швидше.

GIF (Graphics Interchange Format) є одним із найстаріших форматів стиснення зображень, розроблений у 1987 році. Але сучасна версія GIF з’явилася трохи пізніше, у 1989, коли до формату додали підтримку анімації, яка обумовила його довготривалу популярність у веб-дизайні.

GIF має підтримку альфа-каналу (прозорості), але тільки з 1 бітом, тобто формат не підтримує напівпрозорості і піксел або непрозорий, або повністю прозорий.

Формат використовує лише 8 біт для передачі обмеженої палітри кольорів (не 8 біт на кожен канал кольору, а 8 біт загалом). Таке обмеження, хоча й є невідповідним для фотографій або градієнтів, дозволяє ефективно стискати фон, зберігаючи основні деталі зображення. Ця концепція має перспективу у складі набору адаптивних алгоритмів стиснення для окремих типів зображень, хоча й не має широкого застосування у сучасних кодеках. Кольори, які відсутні у базовій палітрі формату, можна отримати за рахунок алгоритмів розсіювання, генерації шуму та накладання періодичної структури (dithering, noise та pattern) [5-6].

Наразі, хоча GIF все ще поширений у зв’язку з його універсальною підтримкою, він поступово витісняється ефективнішими форматами, такими як WebP, AVIF та APNG.

PNG (Portable Network Graphics) – це растровий формат зображень без втрат, розроблений у 1996 році як заміна GIF. Він підтримує прозорість, високу якість, а також ефективне стиснення без втрат.

Потрібно зазначити, що PNG має декілька варіантів реалізації, які різняться між собою глибиною кольорів та функціоналом прозорості. Так, PNG-8 є дуже подібним до GIF, за винятком підтримки анімації, і дуже добре працює з простими зображеннями, такими як іконки, логотипи або прості ілюстрації. PNG-24 взагалі не підтримує прозорості, але має 8-бітову глибину кольору для кожного каналу, і використовується у випадку фотографій та зображень із градієнтами. Він має більший розмір файлу за PNG-8 і його стиснення менш ефективне за сучасні формати, але у порівнянні з ними він має перевагу у швидкодії та максимальній точності передачі кольорів. PNG-32 – це копія PNG-24 із доданим функціоналом прозорості, має відносно великі розміри стиснених файлів, але за потреби у напівпрозорості все ще залишається одним із кращих форматів для логотипів та UI.

За замовчуванням використовується формат PNG-24, який використовується навіть для тих зображень, які було б значно ефективніше стискати в режимі PNG-8, тобто PNG-8 практично не використовується, незважаючи на його переваги для окремих ситуацій та умов.

HEIF (High Efficiency Image Format) – це сучасний формат стиснення зображень, розроблений організацією MPEG (Moving Picture Experts Group) та стандартизований як ISO/IEC 23008-12 у 2015 році.

HEIC (High Efficiency Image Container) – це реалізація HEIF, заснована на кодеку HEVC (H.265). Він використовується Apple та іншими виробниками для зберігання зображень з ефективнішим стисненням, ніж JPEG, PNG і GIF.

Цей формат підтримує стиснення як з втратами, так і без втрат, HDR із глибиною кольорів у 10-12 біт, анімацію та прозорість, а також надає можливість зберігати декілька варіантів одного зображення (Live Photos) [60-61].

HEIC використовує кодек H.265, що застосовує адаптивне роздільне кодування для обробки частотних компонентів (дискретні косинусні та синусні перетворення), квантування і адаптивне кодування, а також блоки різного розміру (до порядку 64) для кращого стиснення [8].

HEIF / HEIC використовується в першу чергу у мобільних пристроях завдяки високій ефективності стиснення, але має обмежену підтримку у Windows та веб-браузерах. В перспективі він може бути витіснений форматами AVIF та JPEG XL.

AVIF (AV1 Image File Format) – це сучасний високоефективний формат стиснення зображень, заснований на відеокодеку AV1. Він був розроблений Alliance for Open Media (AOMedia) та офіційно представлений у 2019 році як альтернатива JPEG, PNG, WebP та HEIC.

Формат надає функцію стиснення з втратами та без втрат, має ефективність стиснення кращу за JPEG, WebP та навіть HEIC, підтримує HDR та 10-12 біт на колір, прозорість та анімацію. При цьому він є безкоштовним і не має патентних обмежень.

Він користується адаптивним вибором між ДКП, ДСП, здійснює квантування з урахуванням особливостей людського сприйняття, має передбачення між блоками і застосовує контекстно-адаптивне ентропійне кодування.

AVIF використовується у багатьох веб-браузерах та сервісах [62]. Він поступається старим кодекам, таким як WebP або JPEG, тільки у складності обчислень (кодування потребує більше ресурсів). Але розроблений через кілька років після цього формату JPEG XL має краще безвтратне стиснення і вважається більш ефективним для архівування.

Серед інших форматів, вартих згадки, можна зазначити FLIF [63], DjVu [5-6, 64], TIFF [65], OpenEXR (EXR) [66].

FLIF (Free Lossless Image Format) – це формат стиснення без втрат, розроблений для кращої компресії, ніж PNG, JPEG 2000 та WebP. Він може працювати з будь-якою палітрою кольорів і має відносно високу ефективність

стиснення, але дуже повільне кодування. Що цікаво, формат використовує прогресивне завантаження, тобто зображення відображається поступово.

Перспективні ідеї використано у форматі DjVu – спеціалізованому для стиснення зображень, оптимізованому під скановані документи, книги, рукописи та технічні креслення. Він був розроблений у 1996 році в AT&T Labs і є ефективною альтернативою TIFF та PDF для зберігання документів із зображеннями. Формат відзначається використанням шарової структури (Mixed Raster Content, MRC), розділяючи текст, фонове зображення та ілюстрації для ефективнішого стиснення, а також функцією прогресивного декодування. Завдяки використанню режимів стиснення із втратами і без, а також використанню шарової структури формат має дуже значну перевагу у ефективності кодування порівняно із TIFF та PDF. Водночас наявні і недоліки: за високих степенів стиснення помітні втрати якості, і на відміну від PDF у редакторах формату немає підтримки редагування користувачем. Формат має дуже обмежену підтримку.

Формат TIFF (Tagged Image File Format) [65] є універсальним форматом для збереження зображень високої якості. Він є стандартом у архівуванні та поліграфії, використовується у медицині і наукових дослідженнях. Однак внаслідок стиснення без втрат або мінімального стиснення розміри файлу завжди дуже великі, і формат має обмежену підтримку у веб-браузерах.

EXR – формат, розроблений для використання у кіноіндустрії. Він використовується у CGI, 3D-графіці, кіновиробництві та візуальних ефектах і підтримує глибину кольору до 32 біт. Це особливо спеціалізований формат, який має дуже великі розміри файлу і не підтримується у браузерах. Він є ідеальним для 3D графіки та візуальних ефектів.

1.2 Застосування нейромереж для стиснення і відновлення зображень/кадрів з втратами

Швидке зростання обсягу візуальних даних за останні роки створило гостру потребу в більш ефективних методах стиснення зображень і відео. Відеоконтент становить понад 80% всього інтернет-трафіку [67], що накладає величезне навантаження на пропускну здатність і сховища. Традиційні відеокодеки, такі як H.264/AVC, H.265/HEVC і найновіший H.266/VVC, а також кодеки для зображень (JPEG, JPEG 2000, WebP), були оптимізовані протягом десятиліть для ефективного стиснення візуальних даних. Ці кодеки покладаються на спеціально розроблені алгоритми обробки сигналів (блочні перетворення, компенсація руху, ентропійне кодування тощо) і досягли значного покращення коефіцієнтів стиснення. Проте покращення в межах класичної схеми кодування (предиктивне кодування + перетворення + квантування + ентропійне кодування) стають дедалі менш значними. Дедалі важче знаходити нові покращення в просторово-часовому передбаченні чи перетвореннях, які суттєво перевищують ефективність поточних стандартів [68].

У той же час, глибоке навчання, особливо згорткові нейронні мережі (CNN), здійснили революцію в багатьох сферах обробки зображень і комп’ютерного зору. Нейромережі можуть вчитися ефективно представляти дані, що робить їх природно придатними для задач стиснення. Останніми роками дослідники почали розробляти методи нейронного стиснення зображень і відео, які намагаються вивчати весь процес кодування та декодування на основі даних [68]. Перші результати показали, що методи нейромережевого стиснення можуть досягати або навіть перевершувати ефективність традиційних кодеків за критерієм швидкість-спотворення (rate-distortion performance), хоча зі значно вищою обчислювальною складністю [67]. Це відкрило новий перспективний напрямок досліджень у стисненні, з

потенціалом використання нейромереж для оптимізації традиційних кодеків або навіть повної заміни традиційних методів у майбутньому.

Розглянемо сучасний стан технологій нейромережевого стиснення та проаналізуємо їх з різних точок зору: ефективність стиснення (співвідношення спотворень і бітрейту), енергетична та обчислювальна ефективність, порівняння з традиційними кодеками, гібридні підходи, що поєднують нейромережі та традиційні методи, теоретичні основи навчального кодування та практичні застосування в таких сферах, як потокове передавання, IoT, медична візуалізація та космічна галузь. Також розглянемо наявні експериментальні результати і гіпотези та нові ідеї для майбутніх досліджень – зокрема, проєктування кодеків, що використовують нейромережі на стороні декодування для відновлення зображень та відео з мінімальних даних.

Традиційне стиснення зображень і відео має багату історію, засновану на теорії інформації та обробці сигналів. Методи, починаючи з JPEG (1992 рік) і аж до H.265/HEVC (2013 рік), мають спільну структуру: ортогональні перетворення (наприклад, DCT) для декореляції пікселів, квантування для відкидання частини інформації та ентропійне кодування для усунення статистичної надмірності. Відеокодеки додатково застосовують міжкадрове передбачення (оцінку та компенсацію руху) і циклічні фільтри для зменшення артефактів стиснення. Ці методи були значно вдосконалені: наприклад, у HEVC використовується до 33 режимів внутрішнього передбачення та складні структури блочного розподілу на основі квадродерев, а також фільтри всередині циклу кодування (deblocking та SAO) для зменшення артефактів. У результаті десятиліть досліджень сучасні кодеки працюють близько до теоретичних меж стиснення у багатьох сценаріях, і подальші досягнення за допомогою традиційних методів даються важко [68].

Паралельно розвиток глибокого навчання відкрив нові можливості представлення та стиснення даних. Нейромережеве стиснення зображень з’явилося приблизно у 2016 році, коли автоенкодери вперше були навчені повністю для стиснення зображень. Toderici та ін. (2017) використовували

рекурентні нейромережі для поступового стиснення фрагментів зображення, досягаючи адаптивного бітрейту за декілька ітерацій [69]. Ballé та ін. (2017) запропонували згортковий автоенкодер із диференційовною апроксимацією квантування (за допомогою додавання рівномірного шуму), що дозволило навчати модель від початку до кінця з оптимізацією критерію "спотворення-бітрейт". Вони показали, що цей підхід може перевершити JPEG2000 за показниками PSNR і MS-SSIM. У 2018 році Ballé та співавтори додали допоміжну гіперприорну (hyperprior) мережу для моделювання просторових залежностей у латентному коді, завдяки чому ефективність стиснення зрівнялася з внутрішньокадровим режимом HEVC (H.265) [70-72]. Невдовзі після цього Minnen та ін. (2018) додали авторегресивну контекстну модель на боці декодера, що дозволило їм повністю перевершити внутрішньокадрове кодування HEVC [73]. Це стало значним досягненням — вперше нейромережевий компресор чітко перевершив провідний традиційний кодек для нерухомих зображень. Водночас автори зазначили, що для практичного використання потрібні рішення щодо апаратної підтримки й енергетичної ефективності, оскільки авторегресивне декодування погано піддається паралелізації та є повільним.

Для стиснення відео застосування цих ідей складніше через часову надмірність. Перші нейромережеві фреймворки відеокомпресії (наприклад, Wu та ін., 2018; Lu та ін., 2019) наслідували традиційний підхід (передбачення руху та залишків), але замінювали кожен компонент нейромережею, яку тренували спільно. Так, фреймворк DVC (Deep Video Compression) від Lu та ін. (2019) став першою повністю нейромережевим відеокодеком, інтегруючи оптичний потік для оцінки руху та окремі CNN для кодування руху і залишкових зображень, треновані разом для оптимізації співвідношення спотворення-бітрейт [74-76]. DVC показав результати кращі за H.264/AVC у PSNR і приблизно на рівні з H.265/HEVC у MS-SSIM. В останні кілька років з’явилося багато інших моделей нейромережевого стиснення відео, які фокусувалися на кращому передбаченні руху, інтерполяції потоку чи рекурентному кодуванні

послідовних кадрів. До 2021 року нейромережеві відеокодери могли перевершувати HEVC у певних умовах, але з практично неприйнятною складністю [67]. У 2023–2024 роках передові нейромережеві відеокодеки (наприклад, DCVC) вже повідомляли про результати, що перевищують ефективність навіть останнього стандарту H.266/VVC, хоч і без реальних часових обмежень [77].

Таким чином, література демонструє стрімку еволюцію нейромережевого підходу: нейромережеве стиснення перейшло від демонстрації принципової можливості до досягнення найкращих показників у цій галузі. Для зображень багато нейромережевих моделей зараз перевершують класичні кодеки, такі як JPEG2000, BPG (на основі H.265 intra) і навіть внутрішньокадровий режим VVC [78]. Щодо відео, нейромережеві методи пройшли шлях від переваги над H.264 до наближення або навіть перевищення H.265 та більш сучасних кодеків.

Рис. 5 ілюструє співвідношення "швидкодія-спотворення" для різних кодеків на стандартному наборі тестових зображень Kodak. Нейромережеві кодеки (сині й помаранчеві криві) зараз займають перші позиції, перевершуючи традиційні кодеки (зелені, фіолетові тощо) за PSNR у широкому діапазоні бітрейтів [79].

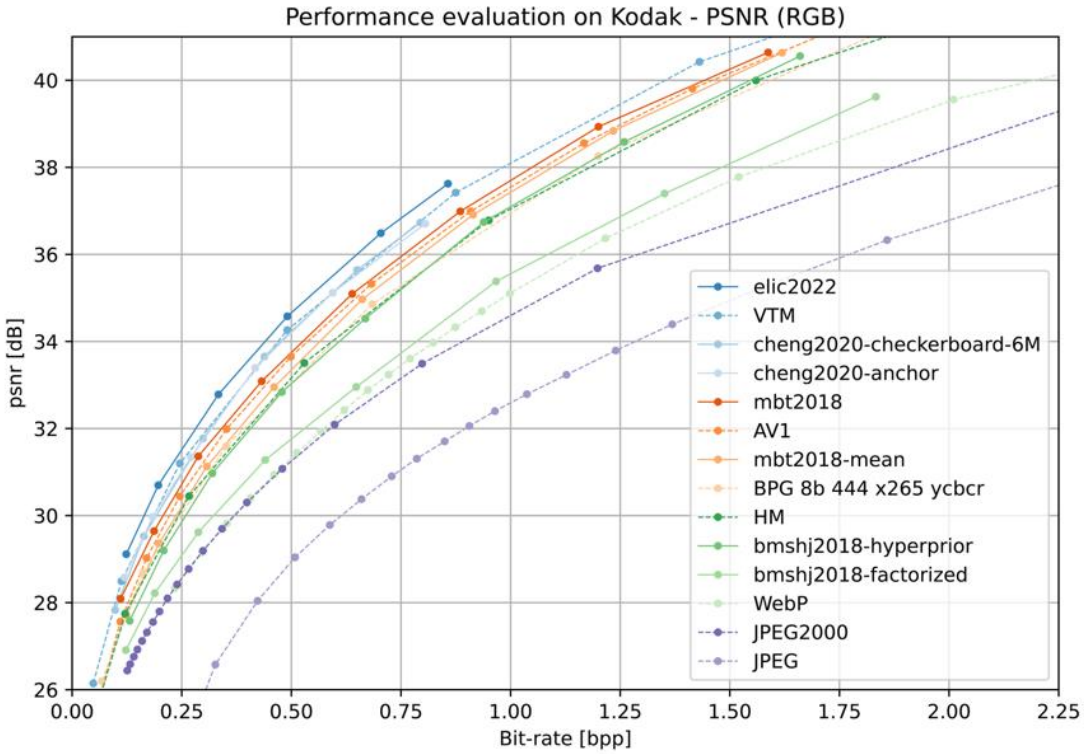

Рисунок 5 Rate-distortion performance

По вертикальній осі – PSNR (дБ), а по горизонтальній – бітрейт (біт на піксел). Вивчені підходи (наприклад, моделі «elic2022» синього кольору та моделі «Cheng2020» світло-блакитного кольору) досягають рівних або вищих показників PSNR на нижчих бітрейтах, ніж сучасні традиційні кодеки, такі як HEVC/H.265 (зелений, «HM» і світло-зелений, «BPG») і AV1 (штриховий оранжевий). Застарілі кодеки JPEG2000, JPEG помітно гірші.

Водночас, ці досягнення супроводжуються значним зростанням обчислювальної складності.

1.2.1 Гібридні підходи: нейронні мережі і традиційні кодеки

Значна частина досліджень присвячена гібридним підходам, які інтегрують нейромережі в структуру традиційних кодеків, замість повної їх заміни. Логіка такого підходу полягає в поєднанні найкращого з обох підходів: використання результатів оптимізації стандартних кодеків (що гарантує

надійність і низьку складність для певних задач) та застосування нейромереж у тих частинах кодека, де вони можуть принести максимальну користь (наприклад, зменшення артефактів або покращення передбачення).

Одним із перших успіхів стало використання CNN для пост-фільтрів або фільтрів у циклі кодування (in-loop filters) з метою зменшення артефактів стиснення. Усунення артефактів стиснення подібне до задачі зменшення шумів, яка ідеально підходить для CNN. Zhang та ін. (2018) запропонували Residual Highway CNN (RHCNN) як in-loop фільтр для HEVC [80]. Їхня 13-шарова CNN навчилася перетворювати декодований кадр (із артефактами) на якісніший, що дозволило отримати близько 4–6% економії бітрейту при однаковій якості для I-, P-, B-кадрів – суттєвий приріст на додачу до вже ефективного кодека. Однак це покращення супроводжувалося збільшенням часу кодування приблизно у 23 рази, що демонструє значний приріст обчислювальної складності. Подальші роботи розробляли багатокадрові або адаптивні за контентом фільтри (наприклад, застосування окремої нейромережі для кожного параметра квантування чи типу контенту), що дозволило додатково збільшити перевагу. Хоча це ефективно, воно потребує більше пам’яті та ресурсів. Наразі тривають дослідження, спрямовані на спрощення таких фільтрів [81-84].

Нейромережі також використовувалися для покращення внутрішньокадрового передбачення. У HEVC внутрішнє передбачення намагається оцінити блок пікселів із сусідніх блоків (наприклад, шляхом кутової екстраполяції). CNN можуть вивчити більш складні просторові закономірності. Cui та ін. (2018) запропонували Intra Prediction CNN (IPCNN) [85], яка уточнює прогнози HEVC шляхом обробки попередньо передбаченого блоку (використовуючи звичайні режими HEVC) і видає покращений прогноз. Це було першим застосуванням CNN у внутрішньому передбаченні стандартного кодека. Аналогічні покращення на основі CNN були запропоновані для компенсації руху (наприклад, нейронна інтерполяція руху між опорними кадрами) та для ентропійного кодування (оцінка ймовірностей коефіцієнтів точніше, ніж у традиційних контекстних моделях). Таким чином,

майже кожний модуль гібридної структури кодека – від передбачення до квантування та циклічної фільтрації – був предметом досліджень щодо застосування нейронних підходів. Огляд цих підходів від Ма та ін. (2019) зазначає, що хоча кожен окремий метод дає певні додаткові переваги, сумарна складність та труднощі інтеграції означають необхідність уважної оцінки доцільності використання. Примітно, що ранні повністю нейромережеві відеокодеки ще поступалися HEVC, тому гібридні підходи виглядали як оптимальний шлях до короткострокового поліпшення [86].

Ще один клас гібридних підходів – адаптація роздільної здатності: стиснення відео чи зображень зі зниженою роздільною здатністю стандартним кодеком, після чого нейронна мережа на боці декодера відновлює зображення у високій якості (нейронна суперроздільна здатність). Це безпосередньо зменшує кількість переданих даних (наприклад, передається 480p замість 1080p), а потужна модель на боці декодера відновлює деталі. Декілька робіт підтвердили цю ідею. Наприклад, підхід із контентно-адаптивною вибіркою Khani та ін. (SRVC, 2021) показав, що при низьких бітрейтах (наприклад, 200 кбіт/с), HEVC на 480p у поєднанні з нейронною надроздільною здатністю дає якість повного 1080p відео, використовуючи лише 16% бітрейту [67]. Порівняно з повністю нейромережевим кодеком DVC, цей гібридний підхід був значно ефективнішим – SRVC вимагав лише 3% даних від DVC для однакової якості. Їх система складалася з двох потоків: «контентного потоку» (стисненого стандартним кодеком відео низької роздільної здатності) та «потоку моделей», який періодично оновлює нейромережу на стороні декодера, що дозволяє адаптувати модель до поточного контенту для покращення якості без значних витрат. Це працювало навіть у реальному часі (90 кадрів/сек на GPU), що робить такі гібридні рішення практичними. Інші дослідження також показали значне скорочення бітрейту за аналогічної якості за допомогою нейромережевої надроздільної здатності.

Таким чином, огляд гібридних підходів свідчить, що нейромережі можуть суттєво покращити традиційні кодеки, забезпечуючи помітний виграш

(наприклад, 5–10% економії бітрейту за рахунок покращення циклічних фільтрів або передбачення), зберігаючи структуру стандартних кодеків. Ці підходи сумісні з існуючими стандартами та апаратним забезпеченням, що є важливою практичною перевагою. Організації зі стандартизації (наприклад, MPEG) вже активно досліджують впровадження нейромережевих інструментів у кодеки наступного покоління [87]. Це свідчить про впевненість у тому, що гібридні підходи поєднання традиційних рішень і нейронних мереж відіграватимуть важливу роль у наступних стандартах стиснення.

1.2.2 Нейромережеві методи стиснення даних

На відміну від гібридних методів, що розглядають нейромережі як додатковий компонент, повністю нейромережеве стиснення має на меті повну заміну традиційного кодека нейронною моделлю, зазвичай автоенкодером. Далі ми детально розглянемо методологію таких кодеків і теоретичні принципи, на яких вони ґрунтуються.

На загальному рівні модель стиснення зображення складається з параметричного нелінійного перетворення g_a (аналізуюча мережа), що кодує зображення $\boldsymbol{x}$ у латентний код y = g_a(x), квантизатора Q, що створює дискретний код hat{y}=Q(y), та декодера g_s, що відновлює зображення hat{x}=g_s(hat{y}). Крім того, зазвичай застосовується модель ентропії (prior), що задає ймовірнісний розподіл p(hat{y}) для ентропійного кодування. Таку схему можна розглядати як нейромережевий варіант класичного трансформного кодування [70], де g_a і g_s є нейромережевими нелінійними перетвореннями замість фіксованих DCT чи вейвлетів. Вся система навчається наскрізно для мінімізації критерію швидкість–спотворення (rate–distortion), який зазвичай має вигляд:

$$L=D+\lambda R,$$

де D – спотворення (наприклад, середньоквадратична похибка або перцепційна метрика між $\boldsymbol{x}$ і hat{x}), а R оцінює бітрейт (зазвичай через ентропію коду

hat{y} за моделлю ймовірностей). Параметр λ регулює співвідношення між якістю і розміром даних, визначаючи різні точки на кривій швидкість–спотворення. Під час тренування операція квантування Q (яка недиференційовна) зазвичай апроксимується шляхом додавання рівномірного шуму або за допомогою диференційовних наближень, що дозволяє здійснювати зворотне поширення градієнту через автоенкодер і оптимізувати перетворення g_a та g_s для оптимального стиснення згідно з R–D-критерієм.

Теоретична основа такого підходу лежить у теорії швидкість–спотворення (Rate-Distortion Theory). Класичні результати Шеннона показують, що для мінімізації ентропії (швидкості передачі) за заданого спотворення необхідно кодувати дані так, щоб максимально використовувати інформацію про розподіл ймовірностей даних. Нейронні мережі ідеально підходять для моделювання складних розподілів, тому сучасні нейромережеві кодеки застосовують нейромережеві моделі ентропії для точнішої оцінки ймовірностей латентних кодів. Наприклад, Ballé та ін. (2018) запропонували додаткову гіперприорну мережу, яка моделює просторові варіації в латентному просторі у (наприклад, змінність дисперсії латентних каналів) [72]. Цей додатковий прихований код z=g_h(y) передається разом з основним кодом ***y*** і використовується декодером для оцінки умовних ймовірностей p(hat{y}|hat{z}) [88]. Такий підхід суттєво покращує стиснення завдяки адаптації ентропійного кодування до конкретного вмісту зображення. Minnen та ін. (2018) додали також авторегресивний контекст, тобто елементи латентного коду декодуються послідовно (по одному або по каналу), що дозволяє враховувати вже декодовані сусідні елементи для точнішої оцінки ентропії [73]. Використовуючи CNN з масками (аналогічно PixelCNN), вони змогли досягти дуже точної оцінки ймовірностей і, таким чином, значно знизити бітрейт за заданого спотворення. Такі методи фактично є нейромережевими аналогами контекстного адаптивного бінарного арифметичного кодування (CABAC) у HEVC, але значно потужніші через більшу гнучкість моделювання розподілів. Теоретично,

чим точніше модель ентропії p(hat{y}) відповідає істинному розподілу даних, тим ближче довжина коду наближається до нижньої межі ентропії.

Обчислювальна складність цих моделей є ключовим теоретичним і практичним питанням. Навчання потребує оптимізації мільйонів параметрів за допомогою стохастичного градієнтного спуску, що вимагає потужних GPU та великих наборів даних. Під час використання кодування вимагає одного проходу аналізуючої мережі й ентропійного кодування у, а декодування – ентропійного декодування та проходу синтезуючої мережі. Ці операції мають складність O(n) за кількістю пікселів, але з великим множником через кількість глибоких шарів мережі [88]. Авторегресивне декодування є особливо повільним, бо елементи декодуються послідовно без можливості паралельних обчислень, що суттєво обмежує швидкість і енергетичну ефективність. Зараз активно досліджуються методи скорочення складності, такі як більш паралельні контекстні моделі або спрощені prior-моделі. Також використовуються методи зменшення розмірів моделей (обрізання, квантування ваг) та спеціалізоване апаратне прискорення (TPU, NPU).

З точки зору математичної формалізації, нейромережеві кодеки оптимізують складну нелінійну R–D-цільову функцію, для якої немає аналітичного рішення, тому використовують емпіричну оптимізацію. Водночас нейромережі фактично вчаться реалізовувати нелінійне перетворення Карунена-Лоєва, узагальнюючи ідею PCA на складніші нелінійні розподіли даних [70]. Використання множника Лагранжа λ для налаштування компромісу між швидкістю та спотворенням є прямим застосуванням класичної теорії швидкість–спотворення, дозволяючи знаходити оптимальний компроміс між якістю і бітрейтом.

Таким чином, сучасні повністю нейромережеві кодеки досягають досить високих результатів завдяки поєднанню потужних нейромереж із класичними теоретичними принципами кодування, хоча їх практичне використання досі ускладнене через обчислювальні обмеження [88]. З іншого боку, потенціал

нейромережевого стиснення значно вищий, і можна очікувати революційні результати в майбутньому.

Порівняння ефективності: нейромережеві кодеки проти традиційних. Узагальнимо експериментальні результати, які порівнюють нейромережеві кодеки з традиційними кодеками за стандартними метриками, а також розглянемо енергетичну ефективність. Порівняння структуровані за типом контенту (зображення та відео) і за типом метрик (точність відтворення, сприйняттєва якість, обчислювальні витрати).

Приблизно до 2020 року нейромережеві кодеки для зображень почали стабільно перевершувати JPEG, JPEG2000 і часто навіть внутрішньокадрове кодування HEVC/H.265 і VVC/H.266. Наприклад, огляд зазначає, що група CNN-моделей [51]–[55] з комфортним відривом перевершує JPEG і JPEG2000 за метриками PSNR та SSIM [78]. У порівнянні з BPG (внутрішньокадровий H.265) – який часто вважають найсильнішим традиційним внутрішньокадровим кодеком – перші нейромережеві моделі показували змішані результати, але найновіші моделі вже перевершують його. Наприклад, у огляді було зазначено, що модель, яка поєднує CNN і покращення якості (Lee та ін., 2020), перевершила JPEG2000, BPG і навіть внутрішньокадрове кодування VVC (H.266 intra) на стандартних тестових зображеннях [89]. Інший приклад (2021) показав, що CNN з перцептивною і змагальною функціями втрат перевершує JPEG, JPEG2000, BPG за суб'єктивною якістю. Також модель «elic2022» та інші у 2021 році показали, що нейромережеві моделі досягають кращих результатів, ніж AV1 і VTM (еталонна реалізація H.266), по всіх бітрейтах [79]. За економією бітрейту нерідко бачимо скорочення на 20–30% для тієї ж якості порівняно з HEVC intra. Наприклад, одне дослідження повідомляє про зниження бітрейту на 24% порівняно з HEIC (на базі HEVC) при однаковій якості, хоча й зі зростанням часу декодування на 85%. GAN-компресори можуть давати навіть більшу економію бітрейту, але з певними сприйняттєвими спотвореннями, які не помітні в PSNR. Наприклад, метод HiFiC (Agustsson et al., 2020) створює фотореалістичні реконструкції на дуже

низьких бітрейтах – візуально значно кращі за JPEG при тому ж бітрейті, хоча PSNR може бути нижчим.

Щодо відео, до 2019 року нейромережеві кодеки досягли рівня H.264 та наблизилися до H.265 [74]. Подальші вдосконалення (натхненні традиційними кодеками – використання B-кадрів, покращені моделі руху) дозволили нейронним методам стати конкурентоспроможними з H.265 вже до 2021 року. Наприклад, було представлено навчальний кодек, що перевершив H.265 за певних умов. Інший приклад – SRVC, який при однаковій якості скоротив бітрейт на 80% порівняно з H.265 на тестових послідовностях низького бітрейту (200 кбіт/с 1080p відео). Це свідчить, що для низьких роздільностей або низького бітрейту нейромережі здатні компенсувати агресивну компресію значно ефективніше, ніж традиційні фільтри. Ще новіші дослідження (Yang et al., 2023 та інші) представили трансформерні моделі, які вже перевершують за якістю H.266 (VVC), припускаючи, що нейромережеві моделі вже здатні перевершити навіть наступне покоління традиційних кодеків (наприклад, H.267) [90].

Попри вражаючі досягнення у стисненні, практичне використання нейронних кодеків наразі обмежується через великі обчислювальні та енергетичні витрати. Традиційні кодеки оптимізовані під спеціалізоване обладнання, тоді як нейромережеві, особливо наскрізні, поки потребують значно більше ресурсів. Різниця може сягати від 10 до 1000 разів за швидкістю декодування чи енергоспоживанням [79]. Наприклад, модель 2018 року потребувала кількох секунд на декодування зображення, тоді як JPEG це робив за мілісекунди. Це робить нейронні кодеки поки неприйнятними для мобільних чи IoT пристроїв без апаратного прискорення.

Є значні покращення в цій галузі: модель Xiao et al. (2020) мала на 85% більший час декодування порівняно з HEVC, але зберігала 24% економії бітрейту, що може бути прийнятним у деяких сценаріях [88]. З іншого боку, для відео проблема ще гостріша: нейронний декодер 1080p відео може

обчислювати лише кілька кадрів на секунду на CPU, тоді як H.264 декодується в реальному часі навіть на малопотужних чипах.

Недавні роботи демонструють ефективніші підходи: зменшення моделей, квантування ваг, менші архітектури, спеціалізовані NPU. Наприклад, метод GRCNN (Chen et al., 2023) для медичних 3D-зображень використовував лише «одну тисячну» параметрів типових моделей, при цьому показуючи переваги якості над традиційними методами й забезпечуючи реальний час декодування на FPGA [91]. Інша робота (Qui et al., 2023) показала трансформерну модель, яка декодує кадри 1080p менш ніж за 100 мс на GPU (~10 кадрів/с), що значно наближає її до реального часу [77].

За якістю (PSNR, MS-SSIM, VMAF) нейронні кодеки особливо виграють за сприйняттєвими метриками. Вони часто показують значні переваги у MS-SSIM над HEVC. GAN-кодеки також суб'єктивно виглядають краще, навіть коли PSNR нижчий [78]. Майбутні системи можуть оптимізуватися під завдання (наприклад, для відеоспостереження важливіше добре розпізнавати об’єкти, ніж ідеальна якість кадру).

Однією з проблем є генералізація: нейронні кодеки можуть створювати дивні артефакти на незнайомих даних, оскільки часто тренуються на обмежених наборах даних. Для впровадження важливо забезпечити стабільність (навчання на різноманітних даних або посилення регуляризації). Також огляд Jamil et al. (2022) виявляє відкриті проблеми, такі як спотворення смугастої області (striped region distrortion, SRD) і артефакти згладжування, які спостерігалися в певних отриманих результатах стиснення і потребують подальшого дослідження [78].

Отже, аналіз показує, що нейромережеві кодеки вже перевершують традиційні за ефективністю, але для їхнього широкого впровадження потрібно вирішити проблему високої складності обчислень.

Застосування та особливості використання нейромереж. Сфери застосування нейромережевого стиснення охоплюють як традиційні галузі (стрімінг, мовлення, зберігання зображень), так і спеціальні сценарії (пристрої з

обмеженими ресурсами, медицина, космос). Розглянемо традиційний стрімінг та зберігання даних.

Основна мета тут – замінити або покращити існуючі кодеки для веб-відео, відео на вимогу, обміну зображеннями. Головною метрикою зазвичай є економія бітрейту при тій самій якості, оскільки головні витрати – це пропускна здатність і місце для зберігання. Нейромережеві методи можуть потенційно зменшити розмір файлів на ~20–30% для того ж рівня якості. Це надзвичайно цінно в масштабах (наприклад, зменшення глобального трафіку YouTube на 20%). Але серйозним викликом залишається впровадження та поширення: декодери мають бути замінені всюди (браузери, телевізори, смартфони), тому новий кодек, особливо з підтримкою нейромереж, потребує серйозної стандартизації й прийняття. Найімовірніший шлях у короткостроковій перспективі – через оновлення програмного забезпечення та додаткові покращення. Наприклад, відеосервіс може використовувати AV1 як базовий кодек, але потім застосовувати нейронний фільтр постобробки на стороні клієнта для підвищення якості. Якщо пристрій клієнта (наприклад, Smart TV з AI-чипом) це підтримує, він отримує покращену якість; якщо ні – бачить базову якість. Таке поступове впровадження вже відбувається у вигляді режимів AI-super-resolution у деяких телевізорах.

У сфері відеоконференцій важливими є низька затримка і вимоги до роботи в реальному часі. Нейронні кодеки повинні досягати ультра-низької затримки (наприклад, методи нейронного розподіленого кодування або дуже швидкі нейромережі). Поки такі технології дозрівають, домінуватимуть традиційні кодеки, можливо, з додатковою нейронною обробкою (наприклад, покращене знешумлення перед кодуванням веб-камери).

В Інтернеті речей багато вбудованих пристроїв (камери безпеки, дрони, датчики), які часто мають обмежену потужність і обчислювальні ресурси, але створюють величезні обсяги даних (наприклад, години відеоспостереження). Нейронні методи можуть суттєво скоротити обсяг даних, які передаються за обмеженими каналами зв'язку (наприклад, камери з батарейним живленням).

Пріоритетом IoT є енергоефективність. Вибір стоїть, чи витратити більше обчислень, щоб зменшити передачу даних, чи навпаки. Оптимальний баланс залежить від співвідношення витрат енергії на обчислення та передачу. Наприклад, дрон може використовувати нейронну компресію, щоб скоротити розмір відео вдвічі, що збільшить дальність передачі або зменшить затримку.

Деякі продукти вже застосовують нейронні методи стискання – наприклад, камера Pixel від Google використовує нейронне знешумлення і компресію при захопленні зображень (ефективніше зберігаючи RAW-фото).

У медицині зображення (МРТ, КТ, мікроскопія) мають дуже високу роздільну здатність і повинні зберігатися довгостроково та передаватися між медичними установами. Зазвичай потрібне безвтратне або майже безвтратне стиснення, оскільки втрата діагностично важливих деталей є неприпустимою. Традиційні безвтратні кодеки (PNG, JPEG-LS) дають лише помірну компресію (2:1 до 3:1 на медичних зображеннях). Нейронні методи можуть досягти більшої компресії завдяки точнішому моделюванню статистики зображень. Наприклад, Chen та Chen (2023) представили нейронний безвтратний компресор для медичних 3D-зображень, який перевершив традиційні методи (наприклад, JP3D на базі JPEG2000), при цьому швидкість декодування дозволяє практичне використання на FPGA [91]. Це передбачає можливість встановлення таких прискорювачів у лікарнях для реальної компресії безпосередньо під час створення зображень. Ще одна галузь – телемедицина: зменшення обсягу медичних даних прискорить дистанційну діагностику.

Космічні апарати мають дуже обмежену пропускну здатність каналів передачі, що ускладнює передачу сирих зображень високої якості. Традиційно використовуються кодеки на основі JPEG2000 або власні вейвлет-кодеки (наприклад, ICER на марсоходах). Нейронна компресія може суттєво покращити ситуацію, проте обмеженням є обчислювальні можливості космічних апаратів, які значно відстають від наземних процесорів. Деякі нові супутники вже мають нейронні прискорювачі для простих задач (наприклад, розпізнавання хмар). Реалістичним варіантом є наземно-космічна схема:

супутник передає частково оброблене зображення, а наземна станція потужною нейромережею відновлює повну якість. Це схоже на гібридні методи, адаптовані до космосу.

Є також вузькоспеціалізовані галузі, наприклад, високошвидкісне відео (1000 кадрів/с), де нейромережі можуть краще використовувати надлишковість кадрів. VR-відео у форматі 360 градусів також може виграти від нейронних методів, які краще обробляють особливості проєкції. Компресія хмар точок і світлових полів (light fields) – нові галузі, де MPEG також тестує нейромережеві підходи.

Дослідники активно шукають методи зменшення обчислювальної складності: обрізання моделей (pruning), квантування параметрів, використання спеціалізованих апаратних прискорювачів. Однак ці методи зазвичай зменшують перевагу нейромережевих методів перед традиційними. Так, зменшення складності мережі може знизити перевагу над HEVC з 30% до лише 5–10% за якістю.

Для реального впровадження нейронних кодеків потрібно знайти баланс: помірне зниження ефективності стиснення в обмін на суттєве скорочення обчислювальних витрат. Короткострокова перспектива – це гібридні рішення, а повністю нейромережеві підходи стануть можливими лише після появи ефективних нейромережевих прискорювачів.

Деякі дослідники запропонували концепцію так званого «генеративного стиснення» (generative compression), коли декодер є генеративною моделлю (як GAN), що додає реалістичні деталі, наприклад, Rippel et al. (2017) розвивали цей напрямок під назвою «generative compression» [92]. Цей підхід може скорочувати обсяг переданих даних приблизно вдвічі для контенту, де абсолютна точність не є принципово важливою (наприклад, художні фільми чи розважальні відео). Звичайно, існує ризик, що модель згенерує неправильну інформацію в контенті, де це критично (документальні, новинні матеріали), але для розважального відео це не проблема, доки результат візуально правдоподібний.

Ще одна цікава можливість — це спільне навчання компресійної моделі та нейронного денойзера (пост-фільтра), який виправлятиме певні дефекти декодування (наприклад, блокові артефакти) [88]. Традиційні кодеки витрачають значну частину бітрейту для мінімізації артефактів, але якщо відомо, що нейронний декодер здатний ефективно усунути ці артефакти, то можна перерозподілити біти на інші аспекти якості. Lee et al. (2020) реалізували подібну ідею у своїй моделі JointIQ-Net, одночасно тренуючи кодек та нейромережу для поліпшення якості, отримуючи кращий результат, ніж кодек і CNN-постфільтр, навчені окремо [89].

З погляду стандартизації це створює певні складнощі, адже нейромережеві кодеки сильно залежать від тренувальних наборів. Важливим напрямом є стандартизація способів представлення та передачі нейронних моделей. Наприклад, стандарт MPEG Neural Network Coding (NNC) вже дозволяє стискати нейронну модель до кількох відсотків від початкового розміру, що дає змогу передавати декодерні нейромережі разом із відеопотоком або оновлювати їх із часом, як це робив метод SRVC у меншому масштабі.

Величезна перевага гібридних методів – поступове впровадження: немає потреби у створенні абсолютно нової екосистеми. Наприклад, можна продовжувати використовувати потоки бітів H.265, лише модифікувавши декодер для включення нейронного фільтра постобробки – потік бітів залишається валідним H.265. Така зворотна сумісність є важливою.

Ще одним цікавим гібридним підходом є попередня обробка (pre-processing): використання штучного інтелекту для попередньої обробки зображень перед кодуванням, щоб зробити їх більш стислими (наприклад, штучний інтелект може інтелектуально згладити текстури, які займають багато бітів, але таким чином, щоб декодер потім міг повернути ці деталі назад). Це схоже на часткове виконання декодування заздалегідь. Деякі дослідження використовували CNN, щоб видалити із зображення деталі, які важко стискаються, а потім відновити їх після декодування за допомогою іншої CNN. Це форма кодування за принципом «аналіз/синтез», коли контент зводиться до

суті, а потім відновлюється. Ці непрямі гібридні підходи поки що є експериментальними, але демонструють творчі способи синергії з традиційними кодеками.

Аналіз медичних і космічних застосувань підкреслює, що різні сценарії використання мають різні пріоритети у стисканні даних. У медицині ключове значення має безвтратне стиснення – тут здатність нейромережевих методів тонко моделювати розподіли (особливо для 16-бітних медичних зображень із високою точністю) є великою перевагою. Метод GRCNN демонструє, що акцент на структурних характеристиках об'ємних даних (наприклад, зрізів МРТ, де сусідні зрізи дуже схожі) дозволяє перевершити універсальні компресори. Також було показано, що модель можна зробити достатньо малою для реалізації на FPGA, що є обнадійливим для використання в лікарнях (плати FPGA можна встановлювати безпосередньо на апарати МРТ для компресії в реальному часі). Головний висновок полягає в тому, що спеціально навчені компресори для конкретних типів даних (наприклад, гіперспектральні супутникові зображення або медичні об'ємні дані) можуть значно перевершувати універсальні кодеки, оскільки вони використовують унікальні статистичні властивості (наприклад, кореляцію між сусідніми зрізами у 3D-зображеннях). Це вказує на майбутнє, в якому алгоритми компресії можуть навчатися окремо для кожної галузі, а не бути універсальними. Це суперечить історії кодування зображень і відео (де один стандарт обслуговує усіх), але це може бути життєздатним рішенням, оскільки навчання моделі для конкретної галузі (наприклад, астрономічних зображень) цілком реальне, і декодер можна постачати разом із даними у разі потреби.

У космічній галузі пріоритетом є максимізація обсягу корисних даних на канал зв'язку та гарантія того, що критична інформація не буде втрачена. Однією зі стратегій може бути прогресивне кодування: спочатку надсилається базовий шар (з втратами, але дуже компактний), а потім послідовні уточнення. Насправді, нейромережеві кодеки часто за своєю природою підтримують прогресивне уточнення (наприклад, рекурентні нейромережі Тодерічі можуть

зупинятися після кількох ітерацій, створюючи версію з нижчою якістю). Це дуже корисно для космічних комунікацій, де можна спочатку швидко отримати зображення низької якості, а за наявності додаткового часу зв'язку – отримати покращення. Ще один перспективний напрямок – нейромережеве приховування помилок (AI-based error concealment): космічні канали часто мають втрати пакетів, і нейронний декодер може краще відновлювати втрачені пакети порівняно з традиційними методами приховування помилок.

Деякі дослідження вже розглядали моделі типу «вчитель-учень»: велика «мережа-вчитель» допомагає тренувати меншу «мережу-учня», яка використовується для реального декодування.

Мультимодальне та семантичне кодування – коли компресія може інтегруватися з розумінням контенту. Якщо кодек може визначити, що зображено (семантика), він може стискати більш інтелектуально (наприклад, знаючи, що певна область є обличчям, і застосовувати спеціальну модель до неї). Традиційні кодеки цього не роблять (вони байдужі до змісту, за винятком загальних моделей). Нейронні кодеки потенційно можуть це робити, особливо з фреймворками для «кодування для машин» (coding for machines), які спеціально розділяють інформацію, необхідну для візуалізації людьми та для машинного аналізу [79].

1.2.3 Застосування інтелектуальних нейромереж для розробки нових підходів, методів і алгоритмів стиснення і відновлення зображень/кадрів

Подальший розвиток методів та алгоритмів кодування зображень та відео може бути за рахунок застосування універсальних нейромереж загального призначення, тобто мультимодальних генеративних нейромереж на базі архітектури трансформерів.

Можна виділити три основні напрямки використання універсальних нейромереж в галузі для досягнення нових результатів:

- Аналіз й узгодження існуючої інформації з публікацій, оскільки в часи інфовибуху значна частина інформації не враховується;
- Організація взаємодії, обговорення вчених для кооперації зусиль, модерація та оцінка діяльності, вкладу, фіксація інтелектуальної власності, прискорення процесів прийняття галузевих рішень;
- Пошук нових підходів, методів та алгоритмів, найбільш перспективний та складний варіант застосування універсальних нейромереж.

Використання інтелектуальних нейромереж для оптимізації використання вже існуючих алгоритмів і кодеків є окремим напрямком досліджень гібридних підходів. Нейромережі можуть робити складний інтелектуальний пошук оптимальних локальних алгоритмів і параметрів кодування.

Іншим напрямком застосування інтелектуальних нейромереж є використання специфічних інтелектуальних фільтрів за вибором користувача, наприклад, карікатурні, мультиплікаційні, модифікуючі фільтри. Зауважимо, що в цілому підхід специфічних фільтрів даних має принципову похибку – адже метою є специфічне покращення візуальної якості, а не підвищення точності відновлення. При цьому може спостерігатись як зменшення кількості збереженої інформації, так і збільшення, у порівнянні з початковою інформацією без застосування інтелектуальних фільтрів, хоча в більшості випадків можна зберегти обидва бажані ефекти – зменшення інформації та специфічну фільтрацію.

1.3 Висновки до розділу

В першому підрозділі розділу надано загальні відомості з кодування і декодування зображень. Розглядається конструкція кодека, різновиди алгоритмів стиснення та відповідні оцінки ефективности роботи кодека, а саме: обчислювальна складність, ефективність стиснення і якість відновленого зображення відносно оригіналу. Серед алгоритмів стиснення виділяються дві

основні категорії: алгоритми стиснення без втрат і алгоритми стиснення із втратами. Алгоритми стиснення без втрат поділяються на предиктивні (алгоритми передбачення) і ентропійні (які зокрема використовуються у програмах-архіваторах). Серед алгоритмів стиснення з втратами виділяють алгоритми блочних перетворень, вейвлет-перетворень, фрактальних перетворень, проріджения і квантування даних і алгоритми з використанням нейронних мереж. Розглядаються переваги і недоліки кожного типу алгоритмів, а також їх комбінацій. Робиться стислий огляд відомих кодеків зображень із описом їх особливостей з точки зору оцінки перспективності наявних у них ідей у майбутніх дослідженнях.

У другому підрозділі увага зосереджується на використанні нейромереж у сфері кодування-декодування зображень і відео. Нейромережі спроможні на значне перевищення лімітів ефективності стиснення сучасних кодеків, але мають низку недоліків. Так, їх обчислювальна складність може два чи навіть три порядки перевищувати обчислювальну складність традиційних кодеків, що ставить під сумнів доцільність їх використання без подальшої розробки і дослідження. При цьому із зниженням обчислювальної складності також знижується і перевага у ефективності стиснення, яка відрізняє нейромережеві кодеки від традиційних. Також за використання нейромереж може виникнути проблема сприйняття нестадартних зображень у випадку, коли нейромережа мала обмежений набір даних для навчання: на незнайомих типах зображень можуть виникати дивні артефакти. Тож хоча в якості напрямку дослідження повністю нейромережеві кодеки є перспективними, проте для сучасного використання актуальними є гібридні кодеки. Перший варіант – оптимізація традиційного кодування із застосуванням нейромереж. Другий – використання даних традиційного кодера і застосування нейромережі для кінцевої обробки відновленого зображення. Можна поділити таку обробку на два типи: фільтри зображень, які просто покращують якість наявного відновленого зображення, і генерацію зображення, яка генерує відновлене зображення на основі даних кодера.

РОЗДІЛ 2. КОНЦЕПЦІЯ ГІБРИДНОГО КОДЕКА: НОВІ МЕТОДИ ТА АЛГОРИТМИ

Дослідники Rippel et al. (2017) запропонували концепцію так званого «генеративного стиснення» (generative compression), коли декодер є генеративною моделлю (як GAN), що додає реалістичні деталі [92]. Цей підхід може скорочувати обсяг переданих даних приблизно вдвічі для контенту, де абсолютна точність не є принципово важливою.

Пропонується новий напрямок гібридного використання нейромереж і традиційного кодування: передача спеціалізованої інформації для генерації зображень та відеокадрів. На стороні кодека використовуються традиційні алгоритми кодування, якими і кодується спеціалізована інформація, що необхідна нейромережі для відновлення, а на стороні декодера застосовується нейромережа для наближеної генерації зображень та відеокадрів.

Дві принципові відмінності запропонованого кодека:

1) Пошук і передача спеціалізованої інформації для нейромереж традиційними методами кодування;

2) Генерація + відновлення даних, замість просто відновлення даних.

Також пропонується виділяти три рівня достовірності декодування:

1) Художня генерація (для художніх фільмів), низька достовірність, мінімальний об’єм закодованих даних.

2) Побутове кодування – середня достовірність з пріорітетом на передачі ключової інф. Приклад – відео користувачів, відеозв’язок, де риси обличчя є важливою ключовою інформацією, а фон – генерується декодером.

3) Точне кодування – висока достовірність. Для спеціалізованих застосувань (медичні, космічні, інтернет-речей та інш).

Задача пошуку ключової спецалізованої інформації для відновлення нейромережами визначає даний напрямок досліджень. Ця задача є складною, та

може мати різні варіанти розв’язку. Розглянемо деякі підходи та алгоритми, з пріорітетом на мінімізації кількості закодованої інформації.

2.1 Задача мінімізації вхідних даних для нейромереж

Задача пошуку ключової специфічної інформації для відновлення нейромережами на стороні декодера є базовою для запропонованої архітектури гібридного кодека. Відповідно, в залежності від отриманих результатів розв’язку цієї задачі залежить конкретна реалізація запропонованої концепції гібридного кодека. Іншою важливою визначальною умовою є вибір рівня достовірності декодування.

У випадку використання спеціалізованих нейромереж для окремих галузей (медицини, космосу, IoT і т.д.) можуть виникати додаткові фактори, що вплинуть на вибір специфічної початкової інформації, яка кодується традиційними методами та алгоритмами.

У випадку низької та середньої достовірності може бути доцільною передача контурів та спрощеної палітри для задовільного відновлення на стороні декодера. Відповідно, пропонується новий метод формування контурів на базі відомого дельта-методу, а саме, обчислення суми вертикальних та горизонтальних різниць і представлення отриманого масиву даних у формі вторинного зображення.

Додатково до контурів та спрощеної палітри пропонується передавати реперні точки зображення, які прив’язані до певної зони зображення, або контуру, і передають інформацію про колір цієї точки, що дозволяє нейромережі більш точно використовувати дані про спрощену палітру (які є узагальненими, без прив’язки до конкретних точок зображення). Аналогічно, передача фрагментів зображення зі складними текстурами з прив’язкою до контурів зображення дозволить більш точно генерувати схожі текстури для всієї області зображення.

2.2 Спрощений алгоритм формування контурів

Використання дельта-алгоритму, який полягає в обчисленні різниць між сусідніми значеннями, з метою зменшення надлишковості даних, є добре і давно відомим методом. Але застосувати дельта-алгоритм до зображення з метою обчислення контурів пропонується вперше. Незважаючи на досить просту ідею, при реалізації виникають додаткові питання та декілька різних варіантів і модифікацій. Дельта-алгоритм був запропонований в якості інструменту отримання контурів в роботі [2]. Він дає кращі результати для зображень з високою та середньою кореляцією між пікселами, ніж для зображень з низькою кореляцією. З іншого боку, аналогічні особливості властиві і іншим методам виявлення контурів (наприклад, фільтрам Собеля, Превітта, Канні).

Окремо вертикальні та горизонтальні різниці недостатньо чітко передають зміни у протилежному напрямку, що можна замінити сумарним використанням вертикальних та горизонтальних різниць (див. рис. 6 і 7).

Є відомий метод обчислення модуля градієнта,

$$D = \sqrt{D_h^2 + D_v^2},$$

$$D_h = I(x+1, y) - I(x, y)$$

$$D_v = I(x, y+1) - I(x, y)$$

на підставі якого будуються градієнтні карти, які використовуються для аналізу контурів. Але запропонований підхід обчислення контурів має меншу обчислювальну складність:

$$D = \left|D_h\right| + \left|D_v\right|,$$

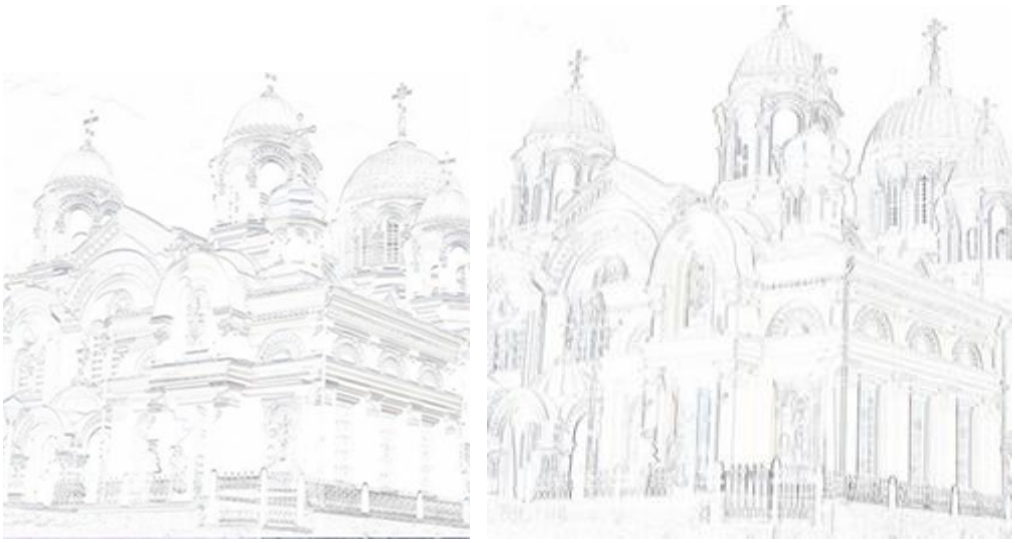

Рисунок 6 Вертикальні та горизонтальні різниці

Крім того, передача градієнтних карт в рамках кодування і декодування даних не використовується, тільки для аналізу даних. В цілому, передачі контурів недостатньо для задовільного відновлення початкових даних навіть з урахуванням генерації додаткових деталей. Передача контурів відбувається в рамках передачі комплексу даних – контурів, палітри кольорів, реперних точок, фрагментів текстур.

Рисунок 7 Сумарні різниці

Зауважимо, що використання суми різниць збільшує отримані значення, що є як перевагою – дає більш чіткі контури, так і недоліком, збільшуючи

яскравість (оскільки більш значні перепади мають менший коефіцієнт стиснення). В якості альтернативи розглядаються дві наступні модифікації: напівсума вертикальних і горизонтальних різниць, та шахматний розподіл.

В якості додаткових інструментів попередньої обробки зображення можна використати два алгоритми: зменшення яскравості пікселів та обнулення значень пікселів менше порогових, в тому числі – одночасне їх застосування. Зменшення яскравості може бути і адаптивним, для значень вище порогового.

Перспективними напрямками досліджень є побудова ієрархії контурів за критеріями масштабу, зв’язності та яскравості. Розподіл (множина) контурів за цими критеріями може дати більше інформації, ніж прості монохромні контури одного масштабу. Диференціація контурів за обраними критеріями дозволяє більш ефективно знаходити патерни контурів зображення, і, відповідно, розрізняти їх. Для виділення контурів певного масштабу достатньо сформувати зображення відповідного масштабу. При цьому залежність буде оберненою: контури найбільшого масштабу будуть на зображеннях найменшого масштабу, а контури мінімального розміру – на зображеннях максимального масштабу.

Таким чином, для формування послідовності контурів різного масштабу необхідно сформувати декілька зображень контурів різного масштабу. Кількість елементів послідовності обирається адаптивно, найчастіше від трьох до п’яти-шести рівней деталізації.

Аналогічно, формується послідовність контурів за критерієм яскравості, розбиваючи діапазон яскравості на два і більше піддіапазонів, до десяти, в залежності від типу зображення. Контури, що належать тому чи іншому діапазону яскравості, тобто мають значення, що вкладаються в певний діапазон яскравості, можна позначати як різними кольорами, так і використовувати градацію сірого кольору.

В результаті об’єднання і спільного використання критеріїв виділення контурів за яскравістю та масштабом виникають наступні граничні комбінації контурів:

1) Яскраві та великі;

2) Яскраві та малі;

3) Непомітні та великі;

4) Непомітні та малі.

В кожній підгрупі (кластері) в свою чергу виділяються чотири підгрупи (кластери) чотирьох типів наступного рівня деталізації, що утворює ієрархію контурів за критеріями яскравості та масштабу.

Відповідно, кожен патерн контурів представляється як комбінація контурів меншого масштабу. Аналогічно за критерієм яскравості. Що в результаті формує представлення контурів як комбінацію контурів різного масштабу та яскравості.

Підкреслимо, що отримані дельта-алгоритмом контури є кольоровими, що можна оцінювати і як недолік, і як перевагу для більш детальних досліджень. Нескладно отримати монохромні контури для оцінки в першому наближенні, маючи кольорові.

Різниці між сусідніми значеннями також можна описувати як похідну в першому наближенні, тобто замість дельта-методу з обчисленням вертикальних та горизонтальних різниць можна говорити про частинні похідні та їх суми. Масив сум частинних похідних в свою чергу формує нове зображення, що відображає контури оригінального зображення. Таким чином, можна отримувати контури зображення, спрощено обчислюючи частинні похідні по $\boldsymbol{x}$ та по $\boldsymbol{y}$ і сумуючи їх, формуючи нове зображення.

Для того, щоб більш ефективно розрізняти контури і текстури з великими перепадами значень пікселів, можна використовувати критерій зв’язності.

Зв’язність контуру визначається наявністю сусідніх пікселів, які приймають близькі значення, тобто формують контур, що значно відрізняється від значень пікселів фону. Додатковою умовою існування контуру є обмеження кількості таких точок. Але тут виникає задача розрізняти розмитий або широкий контур і безконтурні текстури, які мають значні перепади. Для першого випадку зменшують масштаб та застосовують інші додаткові умови та

алгоритми, а в другому випадку – не виконується умова обмеження кількості подібних точок, особливо при зменшенному масштабі.

2.3 Спрощена мода роздільних перетворень

Обчислювальна складність при використанні роздільних адаптивних перетворень зростає внаслідок необхідності перебору та порівняння всіх можливих комбінацій синусного та косинусного перетворень. Відповідно, для зменшення обчислювальних витрат пропонується два додаткових спрощених режими: режим А передбачає використання двох мод з чотирьох, а саме – 2Д косинусне і роздільне косинус/синус, а найшвидший режим В використовує лише одну моду – роздільне перетворення косинус/синус. Зауважимо, що кожне перетворення створює специфічні, типові блочні артефакти. Відповідно, комбінація двох різних перетворень буде більш оптимальною, оскільки замість посилення типових артефактів спостерігається пересічення, комбінація двох типів артефактів. Комбінація синус-косинус програє комбінації косинус-синус за частотою використання, оскільки коефіцієнти перетворення після першого етапу мають нижчу кореляцію, ніж вхідні дані. Комбінація синус-синус обирається для найнижчої кореляції даних, яка на практиці виникає рідко, і крім того, має гірші показники стиснення за інші моди.

Таким чином, для спрощеного швидкодійного кодека пропонується використовувати одну моду косинус/синус замість чотирьох, а для оптимального кодека – дві моди, косинус/косинус та косинус/синус.

Спрощення схеми квантування перетворень. Як відомо, для збільшення коефіцієнту стиснення вхідних даних використовують різні параметри квантування. При цьому коефіцієнт квантування QPst=8 є числом степені двійки, що спрощує обчислювальну складність квантування перетворень у випадку, коли норма перетворення є також степенем двійки. Точність та ефективність перетворення для QPst=8 є трохи меншою, оскільки означає просте відкидання, а не заокруглення до цілого, як для QPst значень, які

не є степенем двійки. Але для спрощених перетворень для швидкодіючих режимів кодування можна надати перевагу зменшенню обчислювальної складності, яка буде залежати від порядку перетворення, чим більший порядок перетворення – тим менше зменшення обчислювальної складності, яке може складати від 5% для перетворень порядку 4 до 2% для перетворень порядку 8.

2.4 Реперні точки і фрагменти текстур

Реперні, або опорні, точки – це ключові точки зображення, що прив’язані до певних зон зображення. Відповідно, інформація про реперні точки є важливою для наближеного відновлення початкових даних з додатковою генерацією деталей запропонованим гібридним кодеком зображень. Грубо кажучи, передача кольорів реперних точок дозволить нейромережі точніше «розфарбовувати» контури зображення.

Альтернативний підхід – передача палітри кольорів зображення без прив’язки до конкретних зон зображення. В цьому випадку кількість інформації буде меншою, особливо за умови передачі спрощеної палітри кольорів, але точність відновлення може погіршитись.

Складні текстури вимагають передачі більшої кількості реперних точок, і мова може йти про передачу невеликого фрагменту текстур, тобто множини зв’язаних точок (пікселів) початкового зображення. На базі фрагменту текстур генеративна складова нейромережі може наближено відновити, тобто згенерувати, з деякою похибкою загальну текстуру відповідної області зображення, чи декількох схожих областей.

Реперні точки можуть передаватись як окремо, так і в складі контурів зображення із мінімальним масштабом, при якому ці реперні точки мають достатню кількість інформації, тобто відповідну порогову похибку.

В цілому задача стиснення контурів, реперних точок та фрагментів текстур може мати різні розв’язки, в рамках різних підходів (зокрема, є традиційний кодек, де контури кодуються за допомогою алгоритма Фібоначчі).

2.5 Похибка пропорцій вертикальних та горизонтальних різниць

Луц Василь у дисертаційній роботі запропонував новий алгоритм обчислення відстані в просторі кольорів з врахуванням змін яскравості та відтінків, оцінюючи зміни співвідношення значень пікселів окремих каналів кольору RGB за наступною формулою [93]:

$$V = \sqrt{\left(\frac{r_1}{g_1} - \frac{r_2}{g_2}\right)^2 + \left(\frac{r_1}{b_1} - \frac{r_2}{b_2}\right)^2 + \left(\frac{g_1}{b_1} - \frac{g_2}{b_2}\right)^2}$$

де r, g та b – значення пікселів за кольорами простору RGB.

Обґрунтування цього підходу в тому, що колір визначається трьома складовими, і це принципово відрізняється від інших відомих просторів, де використовується евклідова відстань. Найбільш поширений алгоритм обчислення відстані в просторі кольорів є застосування евклідової міри, але евклідова метрика не може розрізняти два різні варіанти відхилень від початкового кольору:

1) варіант збільшення (+x, +x, +x), варіант зменшення (-x, -x, -x) (тобто одночасне збільшення або зменшення по всім каналам кольорів RGB);
2) зміни відтінків кольорів: (+, -, +) або (-, +, +) або (+, +, -), коли два значення змінюються в одному напрямку, а третє значення – в іншому напрямку, незалежно від позиції.

Крім обчислення похибки відновлення значень пікселів відносно оригінальних значень, можна додатково обчислювати ще й похибку зміни структури, тобто похибку вертикальних та горизонтальних різниць відновленого зображення. Це нагадує відомий підхід обчислення похибки часткових похідних, який рідко застосовується.

Його можна модифікувати, якщо обчислювати пропорцію, тобто співвідношення перепадів, тобто вертикальних та горизонтальних різниць відносно кожного піксела, як для оригінального зображення, так і для відновленого, і відповідну похибку, зміну пропорції, наприклад, за наступним методом:

$$D_h = I(x+1, y) - I(x, y),$$
$$D_v = I(x, y+1) - I(x, y),$$
$$P = \left| \frac{D_h^a}{D_v^a} - \frac{D_h^b}{D_v^b} \right|$$

де D_h^a - горизонтальна різниця, або спрощена форма часткової похідної по ***x*** початкового зображення ***a***, D_h^b - горизонтальна різниця, або спрощена часткова похідна по ***x*** відновленого зображення ***b***, аналогічно, D_v^a - вертикальна різниця, або спрощена часткова похідна по ***y*** початкового зображення ***a***, D_v^b - вертикальна різниця, або спрощена часткова похідна по ***y*** відновленого зображення ***b***.

Відповідно, похибка пропорції перепадів по вертикалі та горизонталі обчислюється для кожного каналу RGB, і можна зробити загальну оцінку похибки по всім трьом каналам, використовуючи відому формулу евклідової відстані.

2.6 Висновки до розділу.

Запропоновано новий підхід гібридного використання нейромереж і традиційного кодування: пошук, кодування і передача спеціалізованої інформації для генерації зображень та відеокадрів. На стороні кодека використовуються традиційні алгоритми кодування цієї спеціалізованої

інформації, на стороні декодера – нейромережі для генерації зображень та відео. Дві принципові відмінності запропонованого гібридного кодека:

1) Пошук і передача спеціалізованої інформації для нейромереж традиційними методами кодування;

2) Генерація + відновлення даних, замість просто відновлення даних.

Три рівні достовірності декодування:

1) Художня генерація (для художніх фільмів), низька достовірність, мінімальний об’єм закодованих даних.

2) Побутове кодування – середня достовірність з пріорітетом на передачі ключової інформації. Приклад – відео користувачів, відеозв’язок, де риси обличчя є важливою ключовою інформацією, а фон – генерується декодером.

3) Точне кодування – висока достовірність. Для спеціалізованих застосувань (медичні, космічні, Інтернет речей та інш).

Задача пошуку ключової спецалізованої інформації для відновлення нейромережами визначає даний напрямок досліджень. Ця задача є складною, та може мати різні варіанти розв’язку.

У випадку низької та середньої достовірності може бути доцільною передача контурів та спрощеної палітри для задовільного відновлення на стороні декодера. Відповідно, пропонується новий спрощений метод формування контурів на базі обчислення суми вертикальних і горизонтальних різниць, перепадів і представлення отриманного масиву даних у формі вторинного зображення.

Для зменшення обчислювальних витрат пропонується спрощене роздільне перетворення косинус/синус.

Запропоновано обчислювати пропорцію, співвідношення вертикальних та горизонтальних перепадів, різниць для оригінального зображення і для відновленого, а також відповідну похибку, зміну пропорції, з метою оцінки структурних змін відновленого зображення відносно початкового.

РОЗДІЛ 3. ГІБРИДНІ ОРТОГОНАЛЬНІ ПЕРЕТВОРЕННЯ

Ортогональні перетворення є добре відомими і дослідженими. Ортогональні перетворення, як клас лінійних перетворень, відіграють ключову роль у багатьох галузях математики, фізики, інженерії та комп'ютерних наук. Їхня фундаментальна властивість – збереження евклідової структури простору, тобто довжин векторів та кутів між ними, – робить їх незамінними в задачах, де важлива геометрична інваріантність. Формально, ортогональне перетворення – це лінійне відображення векторного простору, яке зберігає скалярний добуток. У матричному вигляді це означає, що матриця ортогонального перетворення (ортогональна матриця) задовольняє умову:

$$A^T A = A A^T = I,$$

де A^T – транспонована матриця A, а I – одинична матриця.

З цього випливає, що стовпці (і рядки) ортогональної матриці утворюють ортонормований базис. Ортогональні перетворення є підкласом унітарних перетворень, які узагальнюють їх на комплексні векторні простори.

Окремим класом ортогональних перетворень є цілочислові ортогональні перетворення, які мають меншу обчислювальну складність порівняно з дискретними ортогональними перетвореннями. Цілочислові перетворення є цілочисловою апроксимацією дискретних ортогональних перетворень, тому їх не завжди розрізняють та відокремлюють від дискретних перетворень.

Новим кроком в теорії цілочислових ортогональних перетворень стало дослідження таких цілочислових перетворень, що мають незначну похибку ортогональності. Це дозволило розширити кількість можливих цілочислових апроксимацій дискретних перетворень. Обґрунтування застосування цілочислових перетворень з похибкою ортогональності є простим: похибка квантування коефіцієнтів перетворення є достатньо великою для того, аби розглядати новий клас перетворень, що мають похибку ортогональності.

Частковим випадком цілочислових ортогональних перетворень з похибкою є цілочислові перетворення з похибкою апроксимації, а не тільки ортогональності. В свою чергу, це значно розширює кількість можливих цілочислових ортогональних перетворень, хоча більшість перетворень з похибкою апроксимації програє за якістю відновлення даних дискретним перетворенням та цілочисловим перетворенням з малою похибкою апроксимації.

Але теорія адаптивних перетворень, що базується на адаптивному підборі перетворення під конкретні вхідні дані, дозволяє адаптивно застосовувати цілочислові ортогональні перетворення з похибкою апроксимації в якості оптимальних для конкретних умов та даних. Обчислювальна складність такого перебору перетворень зросла, що ставить завдання дослідити оптимальні комбінації цілочислових перетворень з похибкою апроксимації.

3.1 Основні поняття гібридних перетворень

Отже, існує досить багато різних типів ортогональних перетворень, серед яких можна виділити перетворення Фур'є, Адамара, Уолша, Хаара, дискретне косинусне перетворення (найбільш відоме є ДКП II типу, але в цілому їх вісім типів), дискретні синусні перетворення різних типів, слент-перетворення та інш.

Спроби розробляти й досліджувати ортогональні гібридні перетворення на базі двох відомих ортогональних перетворень привели до появи нових перетворень. При цьому загальної теорії гібридних перетворень не розглядалось, в кожному конкретному випадку дослідники комбінували існуючі відомі перетворення за власними ідеями і методами. Окремим випадком отримання нових ортогональних перетворень є модифікація відомих перетворень, а не гібридизація перетворень різних типів.

Визначення. Гібридним ортогональним перетворенням називається таке ортогональне перетворення, яке має половину функцій від ортогонального

перетворення одного типу, а іншу половину функцій – від ортогонального перетворення іншого типу.

Вперше гібридне дискретне ортогональне перетворення було запропоновано Кекре і Соланкі [27], яке отримало назву дискретне синусно-косинусне перетворення порядку N. Це перетворення побудовано на основі дискретних синусних і косинусних перетворень типу II порядку $N/2$.

Другим прикладом гібридних перетворень є цілочислове модифіковане синусно-косинусне перетворення типу VII порядку 8, яке вперше запропоновано Гнатівим Л. [29] на основі модифікованого синусного перетворення типу VII порядку 4 і косинусного перетворення типу II порядку 4, із швидким алгоритмом низької обчислювальної складності.

В роботі [37] Гнатівим Л. запропоновано матричний метод побудови дискретного косинус-синусного типу VII порядку N, і на його основі побудовано два цілочислові косинус-синусні перетворення типу VII порядку 8 із швидкими алгоритмами низької складності.

В цілому існує два підходи формування гібридних перетворень – комбінація перетворень меншого порядку і комбінація перетворень рівного порядку. Розглянемо їх детальніше, а також наведемо конкретні приклади застосування й оцінки ефективності відносно відомих типів ортогональних перетворень.

Базовий алгоритм гібридних ортогональних перетворень на основі двох ортогональних перетворень $T^1_{N/2}$ і $T^2_{N/2}$ меншого порядку можна в загальному вигляді записати формулою:

$$T_N = diag\left[T^1_{N/2}, T^2_{N/2}\right] H_N$$

$$H_N = \begin{bmatrix} I_{N/2} & \overline{I}_{N/2} \\ \overline{I}_{N/2} & -I_{N/2} \end{bmatrix}$$

$$\overline{I}_N = antidiag\left[I_N\right]$$

Зауважимо, що на базі двох ортогональних перетворень можна створити два гібридні перетворення, в залежності від того, яке з них буде йти першим.

Гібридні ортогональні перетворення, що розроблені на базі відомих ортогональних перетворень меншого порядку, можна називати спрощеними перетвореннями, оскільки вони використовують лише функції меншого порядку, відповідно, мають нижчу обчислювальну складність і нижчу ефективність кодування.

В якості гібридного спрощеного ортогонального перетворення на базі двох відомих ортогональних перетворень меншого порядку розглянемо нові цілочислові однонормові перетворення: косинус-синусне перетворення порядку 16, яке побудовано на основі косинусного перетворення типу II порядку 8 та синусного перетворення типу IV порядку 8, та синус-косинусне перетворення порядку 16, яке побудовано на базі двох перетворень порядку 8: синусного перетворення типу VII і косинусного перетворення типу II.

Аналогічно, маючи два ортогональні перетворення порядку N, можна розробити два гібридні ортогональні перетворення порядку N, в залежності від того, функції якого базового перетворення будуть взяті за перший рядок нового гібридного перетворення. Відповідно, нове гібридне перетворення буде мати половину функцій одного базового перетворення і половину функцій другого базового перетворення, тобто парні функції будуть від одного базового ортогонального перетворення, а непарні функції – від іншого.

Зауважимо, що обчислювальна складність гібридного ортогонального перетворення на основі базових перетворень такого ж самого порядку буде вищою, ніж на основі базових перетворень меншого порядку. Відповідно, більш популярним є метод на основі перетворень меншого порядку.

3.2 Цілочислове косинус-синусне перетворення типу IV порядку 16

На основі відомих косинусного типу II і синусного типу IV перетворень можна побудувати гібридне перетворення більшого порядку. Таке

перетворення буде мати нижчу обчислювальну складність, оскільки базується на перетвореннях меншої розмірності. Пропонується нове гібридне перетворення порядку 16 зі швидкими алгоритмами обчислення.

3.2.1 Метод побудови цілочислового однонормового косинус-синусного перетворення типу IV порядку *N*

В [3] побудовано матрицю дискретного косинус-синусного перетворення порядку *N*, в якій рядки з парними номерами представляють базисні синусні функції дискретного синусного перетворення типу IV (ДСП-IV), а рядки з непарними номерами представляють базисні косинусні функції дискретного косинусного перетворення типу II (ДКП-II). Матриці N-точкового DCT-II [94] і N-точкового ДСП-IV [95] визначають як

$$\left[C_N^{II}\right]_{k,n} = \sqrt{\frac{2}{N}}\ {}_k \cos\left[\frac{k\left(2n+1\right)}{2N}\right], k,n = 0,...,N-1, \quad (1)$$

$$\left[S_N^{IV}\right]_{k,n} = \sqrt{\frac{2}{N}} \sin\left[\frac{\left(2k+1\right)\left(2n+1\right)}{2N}\right], k,n = 0,...,N-1, \quad (2)$$

де константа нормування ${}_k$ визначається як

$${}_k = \begin{cases} 1/\sqrt{2}, k = 0 or\, k = N, \\ 1 \text{otherwise.} \end{cases}$$

Матрицю CS_N N-точкового ДКСП визначатимемо як

$$\left[CS_N\right]_{k,n}=\begin{cases}\sqrt{\frac{2}{N}}\ {}_k\cos\left[\frac{k\left(2n+1\right)}{2N}\right], n=0,...,N-1, k=0,2,...,N-2, k-\textit{парне},\\ \sqrt{\frac{2}{N}}\sin\left[\frac{\left(2k+1\right)\left(2n+1\right)}{2N}\right], n=0,...,\frac{N}{2}-1, k=1,3,...,N-1, k-\textit{непарне},\\ (-1)\sqrt{\frac{2}{N}}\sin\left[\frac{\left(2k+1\right)\left(2n+1\right)}{2n}\right], n=\frac{N}{2}-1,...,0, k=1,3,...,N-1.\end{cases} \tag{3}$$

Вираз (3) зводиться до

$$\left[SC_N\right]_{k,n}=\begin{cases}\sqrt{\frac{2}{N}}\sin\left[\frac{\left(k+1\right)\left(2n+1\right)}{2N}\right], n=0,...,N-1, k=0,2,...,N-2, k-\textit{парне},\\ \sqrt{\frac{2}{N}}\cos\left[\frac{\left(k+1\right)\left(2n+1\right)}{2N}\right]\end{cases}$$

Розглянемо матрицю $DCST_N^*$ розміру $N\times N$ ДКСП з переставленими рядками на основі зворотних досконаьних переставлень [96]

$$DCST_N^*=P_N DCST_N, \tag{4}$$

де P_N - матриця $N\times N$ зворотних досконалих перетворень (ДП), $P_N\left(0,N-1\right)=\left(0,2,4,...,1,3,...,N-1\right).$

Матрицю $DCST_N^*$ розміру $N\times N$ ДКСП з переставленими рядками можна побудувати, використовуючи рекурентний метод:

$$DCST_N^*=2^{-1/2}diag\left[C_{N/2}^{II}, S_{N/2}^{IV}\bar{I}_{N/2}\right]H_N^*, \tag{5}$$

де $C_{N/2}^{II}$ - матриця $\frac{N}{2}\times\frac{N}{2}$ DCT-II, $S_{N/2}^{IV}$ - матриця $\frac{N}{2}\times\frac{N}{2}$ ДСП-IV, H_N^* - фактор-матриця $N\times N$ із ненульовими елементами ±1,

$$H_N^* = \begin{bmatrix} I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & -I_{N/2} \end{bmatrix}, \bar{I}_{N/2} = antidiag\left[I_{N/2}\right], \bar{I}_{N/2} = \begin{bmatrix} & & 1 \\ & 1 & \\ \ldots & & \\ 1 & & \end{bmatrix},$$

$I_{N/2}, \bar{I}_{N/2}$ - одинична та антидіагональна одинична матриці $\frac{N}{2} \times \frac{N}{2}$.

3.2.2 Метод побудови цілочислового гібридного косинус-синусного перетворення типу IV порядку 16

На основі методу (5) побудуємо цілочислове косинус-синусне перетворення (ЦКСП) порядку 16. Для цього розглянемо матрицю $ICST_{16}^*$ розміру 16×16 ЦКСП з переставленими рядками на основі двійково-інверсних перестановок (ДІП), перестановок по коду грея (ПКГ) і зворотних ПКГ [96]:

$$ICST_{16}^* = \bar{P}_{16}\tilde{P}_{16}ICST_{16}, \tag{6}$$

де $\tilde{P}_{16}$ - блочно-діагональна матриця 16×16 з матрицями 8×8 P_8 ДІП, $\tilde{P}_{16} = diag\left[P_8, P_8\right], P_8\left(\overline{0,7}\right) = \left(0,4,2,6,1,5,3,7\right), \bar{P}_{16}$ - блочно-діагональна матриця 16×16, яка містить матриці 4×4 G_4 ПКГ і P_4 ДІП та матрицю 8×8 $\bar{G}_8$ зворотних ПКГ, $G_4 = diag\left[I_2, \bar{I}_2\right]$,

$P_4 = diag\left[1, \bar{I}_2, 1\right]$, $\bar{G}_8\left(\overline{0,7}\right) = \left(0,1,3,2,7,6,4,5\right)$, $\bar{G}_8 = diag\left[G_4, \bar{G}_4\right]$,

$\bar{G}_4 = antidiag\left[I_2, \bar{I}_2\right], I_2, \bar{I}_2$ - одинична та антидіагональна одинична матриці 2×2, $\bar{I}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \bar{P}_{16} = diag\left[G_4, P_4, \bar{G}_8\right].$

Матрицю $ICST_{16}^{*}$ розміру 16×16 ЦКП з переставленими рядками запишемо через матрицю ядра ЦКСП:

$$ICST_{16}^{*} = B_{16}CS_{16}^{*}, \tag{7}$$

де CS_{16}^{*} - матриця 16×16 ядра ЦКСП з переставленими рядками, B_{16} - діагональна матриця 16×16 коефіцієнтів нормування.

Матрицю CS_{16}^{*} можна побудувати на основі рекурентного методу (5):

$$CS_{16}^{*} = diag\left[C_8^{II*}, S_8^{IV*}\overline{I}_8\right]H_{16}^{*}, \tag{8}$$

де C_8^{II*} - матриця 8×8 ядра ЦКП-II з переставленими рядками на основі ДІП і ПКГ, S_8^{IV*} - матриця 8×8 ядра ЦСП-IV з переставленими рядками на основі ДІП і зворотних ПКГ, H_{16}^{*} - фактор-матриця 16×16 з ненульовими елементами ±1,

$$H_{16}^{*} = \begin{bmatrix} I_8 & \overline{I}_8 \\ \overline{I}_8 & -I_8 \end{bmatrix}, \overline{I}_8 = antidiag\left[I_8\right], \overline{I}_8 = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \cdots & & \\ 1 & & & \end{bmatrix},$$

$I_8, \overline{I}_8$ - одинична і антидіагональна одинична матриці 8×8.

Матриці S_8^{IV*} і C_8^{IV*} ядер ЦСП-IV і ЦКП-IV порядку 8 і із переставленими рядками містять цілі елементи $\pm a', b', \pm c', d', \pm e', f', \pm g', h', \pm a_i, \pm b_i, \pm c_i, i = \overline{1,8}$, а матриця C_8 ядра ІСТ-II порядку 8 з переставленими рядками містить цілі елементи $\pm a, \pm b, \pm c, \pm d, \pm e, \pm f, \pm g, \pm h, \pm i, \pm j, \pm k$ та мають вигляд:

$$S_8^{IV} = \begin{bmatrix} a' & b' & c' & d' & e' & f' & g' & h' \\ a_1 & a_2 & a_3 & a_4 & a_5 & -a_6 & -a_7 & -a_8 \\ b_1 & b_2 & b_3 & -b_4 & -b_5 & -b_6 & b_7 & b_8 \\ c_1 & c_2 & -c_3 & -c_4 & -c_5 & c_6 & c_7 & -c_8 \\ c_8 & c_7 & -c_6 & -c_5 & c_4 & -c_3 & -c_2 & c_1 \\ b_8 & -b_7 & -b_6 & b_5 & -b_4 & -b_3 & b_2 & -b_1 \\ a_8 & -a_7 & a_6 & a_5 & -a_4 & a_3 & -a_2 & a_1 \\ h' & -g' & f' & -e' & d' & -c' & b' & -a' \end{bmatrix} \quad (9)$$

$a' < b' < c' < d' < e' < f' < g' < h'$.

$$C_8^{II} = \begin{bmatrix} k & k & k & k & k & k & k & k \\ a & b & c & d & -d & -c & -b & -a \\ i & j & -j & -i & -i & -j & j & i \\ e & -f & -g & -h & h & g & f & -c \\ k & -k & -k & k & k & -k & -k & k \\ h & -g & f & e & -e & -f & g & -h \\ j & -i & i & -j & -j & i & -i & j \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \quad (10)$$

$a > b > c > d$, $i > j$, $g > e > h > f$.

Мають місце співвідношення [97]:

$$\begin{cases} a + d = e \\ a - d = h \end{cases}, \begin{cases} b + c = g \\ b - c = f \end{cases}, \begin{cases} a^2 + d^2 = b^2 + c^2 = \\ e^2 + h^2 = f^2 + g^2 = 2 \end{cases}$$

Матрицю S_8^{IV*} ядра IST-IV з переставленими рядками можна побудувати на основі матриці ядра ІСТ порядку 4 у вигляді добутку чотирьох матриць:

$$S_8^{IV*} = \tilde{I}_8 H_8^0 diag\left[T_4^*, T_4^*\right] R_8, \quad (11)$$

де R_8 - матриця 8×8 повороту-розтягу, яка на основній діагоналі містить цілі елементи $\pm S_i\ \left(i=\overline{1,4}\right),$ а на протилежній діагоналі містить цілі елементи $\pm r_i$, H_8^0 - фактор матриця 8×8 із ненульовими елементами ± 1, T_4^* - матриця 4×4 ІСТ із переставленими рядками на основі BRP і GCP, $T_4^*=G_4P_4T_4$, $\tilde{I}_8$ - блочно-діагональна матриця 8×8 з одиничними матрицями 4×4 $\pm I_4$, $\tilde{I}_8=diag\left[I_4,-I_4\right].$

При цьому

$$R_8=\begin{bmatrix} S_1 & & & & & & & r_1 \\ & S_2 & & & & & r_2 & \\ & & S_3 & & & r_3 & & \\ & & & S_4 & r_4 & & & \\ & & & -r_4 & S_4 & & & \\ & & r_3 & & & -S_3 & & \\ & -r_2 & & & & & S_2 & \\ r_1 & & & & & & & -S_1 \end{bmatrix},$$

$$H_8^0=\begin{bmatrix} 1 & & & & 0 & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & 0 & 1 \\ & & & 1 & & & 1 & 0 \\ 0 & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & 0 & 1 & & & -1 & \\ & & 1 & 0 & & & & -1 \end{bmatrix}$$

де $S_i<S_{i+1}, r_i>r_{i+1}, i=\overline{1,3},$

$$T_4^* = \begin{bmatrix} P & P & P & P \\ P & -P & -P & P \\ S & -r & r & -S \\ r & S & -S & -r \end{bmatrix}, \quad r > S, P = 2^n. \tag{12}$$

Для однонормового IST-IV фактор-матриця H_8^0 містить ненульові елементи 1 і $\pm P_i / 2^{m_i}, i = 1,2,3,$ P_i - цілі числа.

Матриця CS_{16} ядра ICST порядку 16 на основі (7)-(10) має вигляд

$$CS_{16} = \begin{bmatrix} k & k & k & k & k & k & k & k & k & \dots & k \\ a' & b' & c' & d' & e' & f' & g' & h' & -h' & \dots & -a' \\ a & b & c & d & -d & -c & -b & -a & -a & \dots & a \\ a_1 & a_2 & a_3 & a_4 & a_5 & -a_6 & -a_7 & -a_8 & a_8 & \dots & -a_1 \\ i & j & -j & -i & -i & -j & j & i & i & \dots & i \\ b_1 & b_2 & b_3 & -b_4 & -b_5 & -b_6 & b_7 & b_8 & -b_8 & \dots & -b_1 \\ e & -f & -g & -h & h & g & f & -e & -e & \dots & e \\ c_1 & c_2 & -c_3 & -c_4 & -c_5 & c_6 & c_7 & -c_8 & c_8 & \dots & -c_1 \\ k & -k & -k & k & k & -k & -k & k & k & \dots & k \\ c_8 & c_7 & -c_6 & -c_5 & c_4 & -c_3 & -c_2 & c_1 & -c_1 & \dots & -c_8 \\ h & -g & f & e & -e & -f & g & -h & -h & \dots & h \\ b_8 & -b_7 & -b_6 & b_5 & -b_4 & -b_3 & b_2 & -b_1 & b_1 & \dots & -b_8 \\ j & -i & i & -j & -j & i & -i & j & j & \dots & j \\ a_8 & -a_7 & a_6 & a_5 & -a_4 & a_3 & -a_2 & a_1 & -a_1 & \dots & -a_8 \\ d & -c & b & -a & a & -b & c & -d & -d & \dots & d \\ h' & -g' & f' & -e' & d' & -c' & b' & -a' & a' & \dots & -h' \end{bmatrix} \tag{13}$$

Елементи s_i, r_i матриці R_8 приймають наступні значення: $s_1 = 5, s_2 = 14, s_3 = 21,$ $s_4 = 29, r_1 = 45, r_2 = 43, r_3 = 40, r_4 = 35.$

Елементи матриці T_4^* приймають значення: $r = 5,$ $s = 2,$ $p = 2.$ Елементи матриці C_8^{II} однонормового масштабованого ICT-II [32] порядку 8 приймають

наступні значення: $a=180, b=152, c=96, d=36,$ $e=152,\ f=36, g=176,$ $h=100, i=172,\ j=56, k=128.$

Елементи матриці S_8^{IV} однонормового масштабованого IST-IV порядку 8 приймають наступні значення:

$a'=20, b'=56, c'=84, d'=116, e'=140,\ f'=160, g'=172, h'=180.$

Значення елементів $a_i, b_i, c_i \left(i=\overline{1,8}\right)$ наведено у таблиці 1.

Таблиця 1

Елементи	Значення елементів матриці CS_{16} при i							
	1	2	3	4	5	6	7	8
a_i	48	139	180	160	87	18	116	175
b_i	85	181	117	55	173	138	11	160
c_i	113	161	53	181	16	172	82	141

Матриця S_8^{IV} запропонованого масштабованого IST-IV порядку 8 має вигляд:

$$S_8^{IV}=\begin{pmatrix} 20 & 56 & 84 & 116 & 140 & 160 & 172 & 180 \\ 48 & 139 & 180 & 160 & 87 & -18 & -116 & -175 \\ 85 & 181 & 117 & -55 & -173 & -138 & 11 & 160 \\ 113 & 161 & -53 & -181 & -16 & 172 & 82 & -141 \\ 141 & 82 & -172 & -16 & 181 & -53 & -161 & 113 \\ 160 & -11 & -138 & 173 & -55 & -117 & 181 & -85 \\ 175 & -116 & 18 & 87 & -160 & 180 & -139 & 48 \\ 180 & -172 & 160 & -140 & 116 & -84 & 56 & -20 \end{pmatrix}.$$

Матриця C_8^{II} запропонованого масштабованого ICT-II порядку 8 має вигляд:

$$C_8^{II} = \begin{pmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 180 & 152 & 96 & 36 & -36 & -96 & -152 & -180 \\ 172 & 56 & -56 & -172 & -172 & -56 & 56 & 172 \\ 152 & -36 & -176 & -100 & 100 & 176 & 36 & -152 \\ 128 & -128 & -128 & 128 & 128 & -128 & -128 & 128 \\ 100 & -176 & 36 & 152 & -152 & -36 & 176 & -100 \\ 56 & -172 & 172 & -56 & -56 & 172 & -172 & 56 \\ 36 & -96 & 152 & -180 & 180 & -152 & 96 & -36 \end{pmatrix}.$$

Матриця CS_{16} запропонованого масштабованого ICST порядку 16 має вигляд (14):

$$CS_{16} = \begin{bmatrix} 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 & \dots & 128 \\ 20 & 56 & 84 & 116 & 140 & 160 & 172 & 180 & -180 & \dots & -20 \\ 180 & 152 & 96 & 36 & -36 & -96 & -152 & -180 & -180 & \dots & 180 \\ 48 & 139 & 180 & 160 & 87 & -18 & -116 & -175 & 175 & \dots & -48 \\ 172 & 56 & -56 & -172 & -172 & -56 & 56 & 172 & 172 & \dots & 172 \\ 85 & 181 & 117 & -55 & -173 & -138 & 11 & 160 & -160 & \dots & -85 \\ 152 & -36 & -176 & -100 & 100 & 176 & 36 & -152 & -152 & \dots & 152 \\ 113 & 161 & -53 & -181 & -16 & 172 & 82 & -141 & 141 & \dots & -113 \\ 128 & -128 & -128 & 128 & 128 & -128 & -128 & 128 & 128 & \dots & 128 \\ 141 & 82 & -172 & -16 & 181 & -53 & -161 & 113 & -113 & \dots & -141 \\ 100 & -176 & 36 & 152 & -152 & -36 & 176 & -100 & -100 & \dots & 100 \\ 160 & -11 & -138 & 173 & -55 & -117 & 181 & -85 & 85 & \dots & -160 \\ 56 & -172 & 172 & -56 & -56 & 172 & -172 & 56 & 56 & \dots & 56 \\ 175 & -116 & 18 & 87 & -160 & 180 & -139 & 48 & -48 & \dots & -175 \\ 36 & -96 & 152 & -180 & 180 & -152 & 96 & -36 & -36 & \dots & 36 \\ 180 & -172 & 160 & -140 & 116 & -84 & 56 & -20 & 20 & \dots & -180 \end{bmatrix}$$

(14)

Базисні вектори матриці CS_{16} мають квадрат норми, яка наближається до степеня числа два: $\|C_i\|^2 = 262144 \pm \Delta i$, де $\|C_i\|^2$ - квадрат норми i-ї базисної функції матриці перетворення (яка прямує до одиниці для перетворення з

ортонормованим базисом), Δi (у відсотках) – відхилення квадрату норми i-ї базисної функції, яке набуває значень $\Delta i = 0{,}12 - 0{,}4\%, i = \overline{0{,}15}$, неортогональність для косинусних функцій із парними номерами становить $0{,}1 - 2\%$, а для синусних функцій із непарними номерами складає $0{,}01 - 0{,}26\%$.

Алгоритм швидкого обчислення 8-точкового прямого ІСТ-ІІ. Матрицю ядра ІСТ-ІІ порядку 8 можна факторизовано представити як добуток чотирьох матриць:

$$C_8^{II*} = C_{8,4} C_{8,3} C_{8,2} C_{8,1}, \tag{15}$$

де $C_{8,i}$ - i-і, $i = \overline{1,4}$, фактор-матриці 8×8 швидкого алгоритму, запропонованого в [32] обчислення 8-точкового прямого ІСТ-ІІ:

$$\begin{gathered}
C_{8,1} = H_8^*, C_{8,2} = diag\left[H_4^*, R_4 \right], C_{8,3} = diag\left[T_2, Q_2, H_4 \right], C_{8,4} = diag\left[I_4, H_4^0 \right], \\
T_2 = \begin{bmatrix} k & k \\ k & -k \end{bmatrix}, Q_2 = \begin{bmatrix} -i & j \\ j & i \end{bmatrix}, H_4^* = \begin{bmatrix} I_2 & \bar{I}_2 \\ \bar{I}_2 & -I_2 \end{bmatrix}, \bar{I}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
H_4^0 = diag\left[1, H_2, 1 \right], H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, H_4 = diag\left[H_2, \bar{H}_2 \right], \bar{H}_2 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix},
\end{gathered} \tag{16}$$

H_4 - блочно-діагональна матриця 4×4 з матрицею 2×2 H_2 Адамара і матрицею 2×2 $\bar{H}_2$, R_4 - матриця 4×4 повороту-розтягу, яка на основній діагоналі містить цілі елементи $\pm c$ і d, а на іншій діагоналі – цілі елементи $\pm a$ і b. При цьому

$$H_4 = \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & -1 & 1 \\ & & 1 & 1 \end{bmatrix}, \quad R_4 = \begin{bmatrix} d & & & a \\ & c & b & \\ & b & -c & \\ -a & & & d \end{bmatrix}.$$

Для однонормового ІСТ-ІІ матриця H_4^0 містить ненульові елементи 1 і $\pm P / 2^m$, P, m - цілі числа.

Матрицю T_4^* ядра ІСТ порядку 4 запишемо як добуток двох фактор-матриць:

$$T_4^* = T_{4,2}T_{4,1}, \tag{17}$$

де $T_{4,k}$ - k-і, $k=1,2,$ фактор-матриці 4×4 швидкого алгоритму обчислення 4-точкового прямого ІСТ:

$$T_{4,1} = H_4^*, T_{4,2} = diag\left[T_2', R_2\right], T_2' = \begin{bmatrix} P & P \\ P & -P \end{bmatrix}, R_2 = \begin{bmatrix} -r & s \\ s & r \end{bmatrix}. \tag{18}$$

T_2' - матриця 2×2 з цілими елементами $\pm P$, R_2 - матриця 2×2 повороту-розтягу з цілими елементами $\pm r, s.$

3.2.3 Алгоритм швидкого обчислення косинус-синусного перетворення типу IV порядку 16

Алгоритм швидкого обчислення 16-точкового прямого ICST. Матрицю CS_{16}^* ядра ICST порядку 16 відповідно до (8) та з урахуванням (11), (17) і швидкого алгоритму обчислення 8-точкового прямого ІСТ-ІІ згідно (15), (16), можна факторизовано представити як добуток шістьох матриць:

$$CS_{16}^* = T_{16,6}, T_{16,5}, T_{16,4}, T_{16,3}, T_{16,2}, T_{16,1}, \tag{19}$$

де $T_{16,k}$ - k-і, $k=\overline{1,6},$ фактор-матриці 16×16 запропонованого швидкого алгоритму обчислення 16-точкового прямого ICST:

$$T_{16,1} = H_{16}^{*}, T_{16,2} = diag\left[H_8^{*}, \bar{R}_8\right], T_{16,3} = diag\left[H_4^{*}, R_4, H_4^{*}, H_4^{*}\right],$$
$$T_{16,4} = diag\left[T_2, Q_2, H_4, \bar{T}_2, R_2, \bar{T}_2, R_2\right], T_{16,5} = diag\left[I_4, H_4^{0}, H_8^{0}\right], \quad (20)$$
$$T_{16,6} = diag\left[I_8, I_4, -I_4\right], \bar{R}_8 = R_8\bar{I}_8.$$

При цьому

$$\bar{R}_8 = \begin{bmatrix} r_1 & & & & & & & s_1 \\ & r_2 & & & & & s_2 & \\ & & r_3 & & & s_3 & & \\ & & & r_4 & s_4 & & & \\ & & & s_4 & -r_4 & & & \\ & & -s_3 & & & r_3 & & \\ & s_2 & & & & & -r_2 & \\ -s_1 & & & & & & & r_1 \end{bmatrix}. \quad (21)$$

Алгоритм швидкого обчислення 16-точкового зворотного ICST. Матрицю CS_{16}^{-1} ядра зворотного ICST порядку 16 можна отримати транспонуванням:

$$CS_{16}^{-1} = CS_{16}^{*T} / k, \quad (22)$$

де $k = 2^m$, m - ціле число.

Матрицю CS_{16}^{-1} на основі (22) і (19)-(21) з урахуванням симетричності фактор-матриць

$\left(H_N^{*T} = H_N^{*}, N = 4,8,16, H_4^{T} = H_4, H_4^{0T} = H_4^{0}, T_2^{T} = T_2, T_2'^{T} = T_2', Q_2^{T} = Q_2, R_2^{T} = R_2\right)$

можна факторизовано представити добутком шістьох матриць:

$$CS_{16}^{-1} = T_{16,1}T_{16i,2}^{T}T_{16i,3}^{T}T_{16i,4}T_{16,5}^{T}T_{16,6}, \quad (23)$$

де $T_{16i,k}^{T}$ - k-i, $k=\overline{1,6}$, транспоновані фактор-матриці 16×16 запропонованого швидкого алгоритму обчислення 16-точкового зворотного ICST:

$$
\begin{aligned}
&T_{16,1}=H_{16}^{*},\ T_{16i,2}^{T}=diag\left[H_{8}^{*},\bar{R}_{8i}^{T}\right],\ T_{16i,3}^{T}=diag\left[H_{4}^{*},R_{4i}^{T},H_{4}^{*},H_{4}^{*}\right],\\
&T_{16i,4}=diag\left[T_{2i},Q_{2i},H_{4},T_{2i}',R_{2i},T_{2i}',R_{2i}\right],\ T_{16,5}^{T}=diag\left[I_{4},H_{4}^{0},H_{8}^{0T}\right],\\
&T_{16,6}=diag\left[I_{8},I_{4},-I_{4}\right].
\end{aligned}
\qquad (24)
$$

При цьому

$$
\bar{R}_{8i}^{T}=\begin{bmatrix}
s_1 & & & & & & & r_1\\
& s_2 & & & & & r_2 & \\
& & s_3 & & & r_3 & & \\
& & & s_4 & r_4 & & & \\
& & & -r_4 & s_4 & & & \\
& & r_3 & & & -s_3 & & \\
& -r_2 & & & & & s_2 & \\
r_1 & & & & & & & -s_1
\end{bmatrix}/k'. \qquad (25)
$$

3.2.4 Експериментальні результати тестування

У таблиці 2 представлено схему обчислення спеціальних парних множень, які використовуються в операції «метелик» для реалізації запропонованого швидкого алгоритму обчислення 1D 16-точкового зворотного ICST.

Для реалізації запропонованого швидкого алгоритму обчислення 1D 8-точкового зворотного IST-IV потрібно 2 операції множення, 62 операцій додавання і 38 операцій зсуву. Для обчислення відомого швидкого алгоритму [32] обчислення 1D 8-точкового зворотного ICT-II потрібно дві операції множення, 40 операцій додавання і 16 операцій зсуву.

Таблиця 2

Множники $\bar{R}_{8i}^T$, R_{4i}^T, $Q_{2i}^T\left(R_{2i}^T\right)$ і елементів матриць H_4^{0T}, H_8^{0T}		Алгоритм виконання операцій $y = s_i * x;$ $z = r_i * x;$	Кількість операцій для реалізації «метелика»			Кількість використаних операцій
			Додавання	Зсув	Множення	
$S_1 = \frac{5}{32}$	$r_1 = \frac{45}{32}$	$x_1 = x \square\ 2;\ x_2 = x + x_1;$ $y = x_2 \square\ 3;\ z = x_1 + y$	2	2	-	2
$S_2 = \frac{14}{32}$	$r_2 = \frac{43}{32}$	$x_1 = x - (x \square\ 3);\ y = x_1 \square\ 1;$ $x_2 = x + x_1;\ z = x_1 + (x_2 \square\ 2)$	3	3	-	2
$S_3 = \frac{21}{32}$	$r_3 = \frac{40}{32}$	$x_1 = x \square\ 2;\ z = x + x_1;$ $x_2 = z \square\ 3;\ y = x_2 + (x \square\ 1)$	2	3	-	2
$S_4 = \frac{29}{32}$	$r_4 = \frac{35}{32}$	$x_1 = x - (x \square\ 2);\ y = x - (x_1 \square\ 3);$ $z = x + (x_1 \square\ 3)$	3	2	-	2
$s = 2/4$	$r = \frac{5}{4}$	$y = x \square\ 1;$ $z = x + (x \square\ 2)$	1	2	-	4
$P_1 = \frac{181}{256}$	0	$x_1 = 181 * x;$ $y = x_1 \square\ 16;\ z = 0$	-	1	1	2
$P_2 = \frac{191}{256}$	0	$x_1 = x - (x \square\ 2);\ y = x_1 - (x \square\ 16);$ $z = 0$	2	2	-	4
$d = \frac{9}{32}$	$a = \frac{45}{32}$	$x_1 = x + (x \square\ 3);$ $y = x_1 \square\ 2;\ z = x_1 + y$	2	2	-	2
$c = \frac{24}{32}$	$b = \frac{38}{32}$	$y = x - (x \square\ 2);$ $z = x + (y \square\ 2)$	2	2	-	2
$i = \frac{43}{32}$	$j = \frac{14}{32}$	$x_1 = x - (x \square\ 3);$ $z = x_1 \square\ 1;\ x_2 = x + x_1;$ $y = x_1 + (x_2 \square\ 2)$	3	3	-	2
$P / 2^m$	0	$x_1 = P * x;\ y = x_1 \square\ m;$ $z = 0$		1	1	2
Усього		-	42	54	4	

Для реалізації запропонованого швидкого алгоритму обчислення 1D 16-точкового зворотного ICST потрібно 4 операції множення, 118 операцій додавання і 54 операції зсуву. Алгоритм має низьку мультиплікативну складність, яка у 8,25 разів менша і потребує на 115% більше операцій додавання порівняно з швидким алгоритмом для дискретного косинус-синусного перетворення, розробленим на основі відомих алгоритмів для DCT-II і DST-IV [98] порядку 8.

Обчислювальну складність запропонованого швидкого алгоритму обчислення 2D 16-точкових роздільних адаптивних ICT-II [32] + ICST, відомого швидкого перетворення DCT-II [98] + DCST наведено у табл. 3.

Таблиця 3

Операції (операцій/ піксел)	Оцінка обчислювальної складності 2D роздільних адаптивних перетворень для блоків 16×16		Порівняльний аналіз
	Запропонованих зворотних	Відомого	
	ICT-II[32] + ICST	DCT-II [98] + DCST	
Множення	0,75	4,0625	У 5,42 разів менше
Додавання + зсув	20,5	10,0625	На 103,73% більше додавань

У табл. 4 наведено експериментальні результати кодування за характеристикою стандартної кількісної оцінки спотворень PSNR (дБ.) для стиснених тестових зображень класу B з роздільною здатністю 1920x1072 пікселів (місто), і двох зображень класу C з роздільною здатністю 1536x1024 пікселів для нормального і високого (22-37) діапазону QP (параметра квантування) [44] на основі 2D запропонованих DCST і ICST і на основі відомого DCT-II [98] та запропонованого DCST роздільних DCT-II/DCST та на основі відомого ICT-II [32] та запропонованого ICST роздільних ICT-II/ICST. У табл. 5 представлені експериментальні результати за коефіцієнтом стиснення *K*:1 на основі 2D запропонованих DCST і ICST та роздільних DCT-II/DCST і ICT-II/ICST. У табл. 4 і 5 наведено середні значення результатів кодування за

характеристиками PSNR і коефіцієнтом стиснення К для трьох тестових зображень класів В і С, де НК – низькокорельовані зображення. В табл. 6 наведено результати ефективності кодування за характеристикою PSNR на основі середніх значень для трьох тестових зображень. Ці результати представляють різниці на основі запропонованих ICST і DCST і на основі запропонованих роздільних перетворень ICT-II/ICST і роздільних DCT-II/DCST. У табл. 7 наведено результати ефективності кодування за коефіцієнтом стиснення К (у %) на основі запропонованих ICST і DCST і на основі запропонованих роздільних перетворень ICT-II/ICST і роздільних DCT-II/DCST.

Таблиця 4

Зображення з блоками 16×16	Перетворення	Результати ефективності кодування за характеристикою PSNR (дБ) для QP (діапазони)				
		QP =4	QP =22	QP =27	QP =32	QP =37
Клас В Місто (НК) 1920x1072	DCST	51,395	42,447	38,276	34,105	29,656
	ICST	45,278	41,005	37,699	33,852	29,568
	DCT/DCST	51,397	42,543	38,438	34,316	29,913
	ICT/ICST	45,290	41,071	37,841	34,051	29,822
Клас С Алея (НК) 1536x1024	DCST	51,161	41,057	36,434	32,024	27,452
	ICST	43,146	39,232	35,720	31,744	27,354
	DCT/DCST	51,162	41,097	36,487	32,089	27,576
	ICT/ICST	43,150	39,259	35,741	31,806	27,477
Клас С Пейзаж (НК) 1536x1024	DCST	51,124	40,920	36,389	32,066	27,612
	ICST	44,285	39,522	35,850	31,856	27,533
	DCT/ DCST	51,127	40,957	36,461	32,192	27,830
	ICT/ICST	44,291	39,546	35,914	31,974	27,750
Середні значення	DCST	51,227	41,475	37,033	32,732	28,240
	ICST	44,236	39,920	36,423	32,484	28,152
	DCT/DCST	51,229	41,532	37,129	32,866	28,440
	ICT/ICST	44,244	39,959	36,499	32,610	28,350

Під час тестування використовувалися роздільні адаптивні перетворення DCT/DCST і ICT/ICST без intra-прогнозування, де DCT і ICT застосовувалися для перетворення стовпців, а DCST і ICST — для перетворення рядків з метою спрощеного порівняння за двома характеристиками ефективності кодування (за якістю і ступенем стиснення) запропонованого цілочислового косинус-синусного перетворення з дискретним косинус-синусним перетворенням.

Таблиця 5

Зображення з блоками 16×16	Перетворення	Результати кодування за коєфіцієнтом стиснення *K* :1 для QP (діапазони)				
		4	22	27	32	37
Клас B Місто (НК) 1920x1072	DCST	1,437	2,225	2,722	3,583	5,916
	ICST	1,415	2,218	2,715	3,577	5,902
	DCT/DCST	1,456	2,342	2,888	3,821	6,382
	ICT/ICST	1,434	2,335	2,882	3,819	6,371
Клас C Алея (НК) 1536x1024	DCST	1,095	1,378	1,621	2,131	3,337
	ICST	1,087	1,378	1,617	2,126	3,328
	DCT/DCST	1,102	1,409	1,672	2,225	3,536
	ICT/ICST	1,094	1,409	1,670	2,221	3,530
Клас C Пейзаж (НК) 1536x1024	DCST	1,099	1,405	1,726	2,327	3,915
	ICST	1,094	1,403	1,723	2,323	3,904
	DCT/DCST	1,107	1,453	1,811	2,475	4,202
	ICT/ICST	1,102	1,452	1,808	2,472	4,195
Середні значення	DCST	1,210	1,669	2,023	2,680	4,389
	ICST	1,199	1,666	2,018	2,675	4,378
	DCT/DCST	1,222	1,735	2,124	2,840	4,707
	ICT/ICST	1,210	1,732	2,120	2,837	4,699

Таблиця 6

Відносно перетворень	Результати ефективності кодування за характеристикою PSNR (дБ) для QP (діапазони)				
	QP =4	QP =22	QP =27	QP =32	QP =37
DCST	-6,991	-1,555	-0,61	-0,248	-0,088
DCT/ DCST	-6,985	-1,573	-0,63	-0,256	-0,09

Таблиця 7

Відносно перетворення	Результати ефективності кодування за коефіцієнтом стиснення *K*:1 (у відсотках) для QP (діапазони)				
	QP =4	QP =22	QP =27	QP =32	QP =37
DCST	-0,909	-0,180	-0,247	-0,187	-0,251
DCT/ DCST	-0,982	-0,173	-0,188	-0,106	-0,170

Запропоноване цілочислове косинус-синусне перетворення порядку 16 порівняно із дискретним косинус-синусним перетворенням порядку 16 за характеристикою PSNR при нормальному (22,27) діапазоні QP для трьох тестових зображень класів B і C зменшує середнє значення PSNR на 1,555 і 0,61 дБ., для нормального (32) і високого (37) діапазону QP зменшує середнє значення PSNR на 0,248 дБ. і 0,088 дБ. При цьому зменшення середнього значення PSNR на 0,248 дБ при QP =32 буде візуально не помітним, згідно з прийнятим Комітетом MPEG суб'єктивним порогом непомітності PSNR < 0,5 дБ. Тобто розроблене перетворення є ефективним для нормального (32) і високого (37) діапазону QP, і може адаптивно застосовуватись з метою зменшення обчислювальної складності. При цьому середнє значення коефіцієнта стиснення К зменшується на 0,187 і 0,251%, а для нормального діапазону QP (22,27) середнє значення коефіцієнта стиснення К зменшується на 0,18% і 0,247%.

Аналогічні дуже близькі результати ефективності тестування за характеристикою PSNR (дБ.) і коефіцієнтом стиснення К (в %) отримані на основі роздільних перетворень ICT-II/ICST і DCT-II/DCST. За результатами

експериментального аналізу, запропоноване перетворення ICST порядку 16 є ефективним як в режимі 2D, так і в режимі роздільних перетворень ICT-II/ICST, так як забезпечує високі результати ефективності кодування за точністю і коефіцієнтом стиснення. Є резерв підвищення ефективності кодування за рахунок більш точної апроксимації DCT-II.

3.3 Цілочислове спрощене однонормове синус-косинусне перетворення порядку 16

На основі двох відомих перетворень – косинусного II типу і синусного VII типу можна побудувати гібридне ортогональне перетворення більшого порядку. В залежності від того, яке перетворення буде задавати першу базисну функцію, може бути два варіанти гібридних перетворень. Але синусно-косинусне гібридне перетворення забезпечує високі результати ефективності кодування для низько- і середньокорельованих зображень, тому воно може бути заміною для синусного перетворення VII типу в схемі роздільних адаптивних перетворень (останнє має високу обчислювальну складність), і більш ефективне для низькокорельованих даних.

3.3.1 Метод побудови цілочислового спрощеного синус–косинусного перетворення типу VII порядку 16

Побудуємо матрицю спрощеного цілочислового однонормового синус–косинусного перетворення типу VII порядку 16 (ЦСКП–VII), в якій рядки з парними номерами представляють базисні синусні функції ЦСП–VII порядку 8, симетрично відображені, а рядки з непарними номерами представляють базисні функції ДКП–II порядку 8, антисиметрично продовжені відносно середини інтервалу.

Матриці N–точкових ДСП–VII і ДКП–II визначаються як [94,98]

$$\left[S_N^{VII}\right]_{k,n} = \frac{2}{\sqrt{2N+1}} \sin\left[\frac{(2k+1)(n+1)}{2N+1}\right], k, n = 0,..., N-1,$$

$$\left[C_N^{II}\right]_{k,n} = \sqrt{\frac{2}{N}}\ {}_k \cos\left[\frac{k(2n+1)}{2N}\right], k, n = 0,..., N-1,$$

де константа нормування ${}_k$ визначається як

$${}_k = \begin{cases} 1\sqrt{2}, k = 0 \text{ або } k = N, \\ 1 \text{ в інших випадках.} \end{cases}$$

Для цілочислових апроксимацій різної точності дискретного косинусного перетворення порядку 8 розроблені швидкі алгоритми обчислення [32,99], в той час як для цілочислової апроксимації синусного перетворення типу VII порядку 8 швидких алгоритмів не розроблено.

Розглянемо матрицю $ISCT_{16}^{*} - VII$ цілочислового спрощеного синус–косинусного перетворення типу VII порядку 16 з переставленими рядками на основі зворотних досконалих переставлень (ЗДП) [96]:

$$ISCT_{16}^{*} - VII = P_2 P_1 P_{16} ISCT_{16} - VII, \quad (26)$$

де P_{16} – матриця 16×16 зворотніх досконалих перестановок, $P_{16}(0,15) = (0,2,4,6,8,10,12,14,1,3,5,7,9,11,13,15)$. P_1 – блочно–діагональна матриця 16×16 з одиничною матрицею 8×8 I_8 і матрицею 8×8 P_8 ЗДП, $P_1 = diag[I_8, P_8]$, P_2 – блочно–діагональна матриця 16×16 з одиничними матрицями I_8, I_4 і матрицею 4×4 P_4 ЗДП $P_4 = diag[1, \bar{I}_2, 1]$, $\bar{I}_2 = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}$, P_4 – діагональна матриця 4×4 з елементами 1 і матрицею 2×2 $\bar{I}_2$.

Матриця $ISCT_{16}^{*}-VII$ спрощеного ЦСКП–VII з переставленими рядками може бути представлена матрицею ядра спрощеного ЦСКП–VII:

$$ISCT_{16}^{*}-VII=B_{16}SC_{16}^{VII*}, \tag{27}$$

де SC_{16}^{VII*} – матриця 16×16 ядра простого ЦСКП–VII з переставленими рядками, B_{16} – діагональна матриця 16×16 коефіцієнтів нормування.

Матрицю SC_{16}^{VII*} можна побудувати з використанням рекурентного методу:

$$SC_{16}^{VII*}=diag\left[S_8^{VII},\overline{C}_8^{*}\right]H_{16}^{*}, \tag{28}$$

де H_{16}^{*} – фактор–матриця 16×16 із ненульовими елементами ±1,

$$H_{16}^{*}=\begin{bmatrix} I_8 & \overline{I}_8 \\ \overline{I}_8 & -I_8 \end{bmatrix},\ \overline{I}_8=antidiag\left[I_8\right],\ \overline{I}_8=\begin{bmatrix} & \cdots & 1 \\ & 1 & \\ 1 & & \end{bmatrix}, \tag{29}$$

$I_8,\overline{I}_8$ – одинична і антидіагональна одинична матриці 8×8, S_8^{VII} – матриця 8×8 ядра ЦСП–VII, яка містить цілі елементи: $\pm A,\pm B,\pm C,\pm D,\pm E,\pm F,\pm G,\pm H$, $\overline{C}_8^{*}$ – матриця 8×8, яка може бути представлена матрицею ядра ЦКП з переставленими рядками на основі ЗДП:

$$\overline{C}_8^{*}=C_8^{*}\overline{I}_8\ ,\ C_8^{*}=P_8'P_8C_8, \tag{30}$$

C_8^{*} – матриця 8×8ядра ЦКП з переставленими рядками, P_8' – блочно-діагональна матриця 8×8 з матрицею 4×4 P_4 ЗДП і одиничною матрицею

4×4 I_4, $P_8' = diag\left[P_4, I_4\right]$, C_8 – матриця 8×8 ядра ЦКП, яка містить цілі елементи: $\pm a, \pm b, \pm c, \pm d, \pm e, \pm f, \pm g, \pm h, \pm i, \pm j, \pm k$.

Матриця S_8^{VII} на основі [100] і матриця C_8 [32] мають вигляд

$$S_8^{VII} = \begin{bmatrix} A & B & C & D & E & F & G & H \\ C & F & H & E & B & -A & D & -G \\ E & G & B & -C & -H & -D & A & F \\ G & C & -D & -F & A & H & B & -E \\ H & -A & -G & B & F & -C & -E & D \\ F & -E & -A & G & -D & -B & H & -C \\ D & -H & E & -A & -C & G & -F & B \\ B & -D & F & -H & G & -E & C & -A \end{bmatrix}$$

$$C_8 = \begin{bmatrix} k & k & k & k & k & k & k & k \\ a & b & c & d & -d & -c & -b & -a \\ i & j & -j & -i & -i & -j & j & i \\ e & -f & -g & -h & h & g & f & -e \\ k & -k & -k & k & k & -k & -k & k \\ h & -g & f & e & -e & -f & g & -h \\ j & -i & i & -j & -j & i & -i & j \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix}. \qquad (31)$$

де $A < B < C < D < E < F < G < H; a > b > c > d, g > e > h > f, i > j$.

Матриця SC_{16}^{VII} ядра ЦСКП–VII порядку 16 на основі (28), (30) і враховуючи представлення матриць S_8^{VII} і C_8 згідно (31), має вигляд

$$SC_{16}^{VII}=\begin{bmatrix} A & B & C & D & E & F & G & H & H & ... & B & A \\ k & k & k & k & k & k & k & k & -k & ... & -k & -k \\ C & F & H & E & B & -A & -D & -G & -G & ... & F & C \\ a & b & c & d & -d & -c & -b & -a & a & ... & -b & -a \\ E & G & B & -C & -H & -D & A & F & F & ... & G & E \\ i & j & -j & -i & -i & -j & j & i & -i & ... & -j & -i \\ G & C & -D & -F & A & H & B & -E & -E & ... & C & G \\ e & -f & -g & -h & h & g & f & -e & e & ... & f & -e \\ H & -A & -G & B & F & -C & -E & D & D & ... & -A & H \\ k & -k & -k & k & k & -k & -k & k & -k & ... & k & -k \\ F & -E & -A & G & -D & -B & H & -C & -C & ... & -E & F \\ h & -g & f & e & -e & -f & g & -h & h & ... & g & -h \\ D & -H & E & -A & -C & G & -F & B & B & ... & -H & D \\ j & -i & i & -j & -j & i & -i & j & -j & ... & i & -j \\ B & -D & F & -H & G & -E & C & -A & -A & ... & -D & B \\ d & -c & b & -a & a & -b & c & -d & d & ... & c & -d \end{bmatrix}. \quad (32)$$

Елементи матриці S_8^{VII} запропонованого однонормового масштабованого ЦСП–VII набувають наступних значень $A=8, B=16, C=23, D=29,$ $E=34, F=39,\ G=42, H=45$.

Елементи матриці SC_{16}^{VII} згідно (32) запропонованого однонормового масштабованого простого ЦСКП–VII набувають таких значень: $a=45, b=38,\ c=24, d=9, e=38,\ f=9,\ g=44,\ h=25, i=43,\ j=14, k=32,$ $A=8, B=16, C=23, D=29,\ E=34, F=39,\ G=42, H=45$.

Матриця SC_{16}^{VII} запропонованого масштабованого ЦСКП–VII має вигляд:

$$SC_{16}^{VII} = \begin{bmatrix} 8 & 16 & 23 & 29 & 34 & 39 & 42 & 45 & 45 & ... & 16 & 8 \\ 32 & 32 & 32 & 32 & 32 & 32 & 32 & 32 & -32 & ... & -32 & -32 \\ 23 & 39 & 45 & 34 & 16 & -8 & -29 & -42 & -42 & ... & 39 & 23 \\ 45 & 38 & 24 & 9 & -9 & -24 & -38 & -45 & 45 & ... & -38 & -45 \\ 34 & 42 & 16 & -23 & -45 & -29 & 8 & 39 & 39 & ... & 42 & 35 \\ 43 & 14 & -14 & -43 & -43 & -14 & 14 & 43 & -43 & ... & -14 & -43 \\ 42 & 23 & -29 & -39 & 8 & 45 & 16 & -34 & -34 & ... & 23 & 42 \\ 38 & -9 & -44 & -25 & 25 & 24 & 9 & -38 & 38 & ... & 9 & -38 \\ 45 & -8 & -42 & 16 & 39 & -23 & -34 & 29 & 29 & ... & -8 & 45 \\ 32 & -32 & -32 & 32 & 32 & -32 & -32 & 32 & -32 & ... & 32 & -32 \\ 39 & -34 & -8 & 42 & -29 & -16 & 45 & -23 & -23 & ... & -35 & 39 \\ 25 & -44 & 9 & 38 & -38 & -9 & 44 & -25 & 25 & ... & 44 & -25 \\ 29 & -45 & 34 & -8 & -23 & 42 & -39 & 16 & 16 & ... & -45 & 29 \\ 14 & -43 & 43 & -14 & -14 & 43 & -43 & 14 & -14 & ... & 43 & -14 \\ 16 & -29 & 39 & -45 & 42 & -34 & 23 & -8 & -8 & ... & -29 & 16 \\ 9 & -24 & 38 & -45 & 45 & -38 & 29 & -9 & 9 & ... & 24 & -9 \end{bmatrix}$$

(33)

Базисні вектори матриці SC_{16}^{VII} мають норму, квадрат якої наближається до числа степені два: $\|S_i\|^2 = 16384 \pm \Delta_i$, де $\|S_i\|^2$ – квадрат норми i–ї базисної функції ЦСКП–VII (що наближається до одиниці для перетворення з ортонормованим базисом), Δ_i (у відсотках) – відхилення квадрату норми i–ї базисної функції, яке становить $\Delta_{2i} = 0,4\%, i = \overline{0,7}$, $\Delta_{2i-1} = 0,15 - 0,7\%, i = \overline{1,8}$, неортогональність для парних синусних функцій становить $0,024 - 0,46\%$, а для непарних косинусних функцій – $0,27 - 2\%$.

3.3.2 Алгоритм швидкого обчислення цілочислового спрощеного синус-косинусного перетворення порядку 16

Матрицю C_8^* ядра ЦКП порядку 8 можна факторизовано представити як добуток чотирьох матриць:

$$C_8^* = C_{8,4}C_{8,3}C_{8,2}C_{8,1}, \tag{34}$$

де $C_{8,i}$ – i–ті, $i = \overline{1,4}$, фактор–матриці 8×8 швидкого алгоритму 8–точкового прямого однонормового ЦКП, запропонованого у роботі [32]:

$$C_{8,1} = H_8^*, C_{8,2} = diag\left[H_4^*, R_4\right], C_{8,3} = diag\left[T_2, Q_2, H_4\right], C_{8,4} = diag\left[I_4, H_4^0\right],$$
$$H_4^* = \begin{bmatrix} I_2 & \bar{I}_2 \\ \bar{I}_2 & -I_2 \end{bmatrix}, R_4 = \begin{bmatrix} d & & & a \\ & c & b & \\ & b & -c & \\ -a & & & d \end{bmatrix}, H_4^0 = diag\left[1, H_2, 1\right], H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$
$$H_4 = diag\left[H_2, \bar{H}_2\right], H_8^* = \begin{bmatrix} I_4 & \bar{I}_4 \\ \bar{I}_4 & -I_4 \end{bmatrix}, \bar{H}_2 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} k & k \\ k & -k \end{bmatrix}, Q_2 = \begin{bmatrix} j & i \\ -i & j \end{bmatrix}.$$
(35)

Граф алгоритму 8–точкового швидкого прямого ЦКП наведено в [32].

Для однонормового ЦКП матриця H_4^0 містить ненульові елементи 1 і $\pm P / 2^m, P, m$ – цілі.

Матрицю SC_{16}^{VII*} відповідно до (28), (30) та з урахуванням алгоритму швидкого прямого ЦКП порядку 8 згідно (34), (35) можна факторизовано представити як добуток п’яти матриць:

$$SC_{16}^{VII*} = S_{16,5}S_{16,4}S_{16,3}S_{16,2}S_{16,1}, \tag{36}$$

де $S_{16,k}$ – k–і, $k = \overline{1,5}$, фактор–матриці 16×16 алгоритму запропонованого 16–точкового швидкого спрощеного ЦСКП–VII:

$$S_{16,1} = H_{16}^*, S_{16,2} = diag\left[S_8^{VII}, \bar{H}_8^*\right], S_{16,k+1} = diag\left[I_8, C_8, k\right], k = 2,3,4i5 \tag{37}$$

$$\bar{H}_8^* = H_8^* \bar{I}_8,\ \bar{H}_8^* = \begin{bmatrix} I_4 & \bar{I}_4 \\ -\bar{I}_4 & I_4 \end{bmatrix}.$$

Алгоритм швидкого обчислення 16–точкового зворотного спрощеного ЦСКП–VII. Матрицю $\left(SC_{16}^{VII}\right)^{-1}$ ядра зворотного спрощеного ЦСКП–VII порядку 16 можна отримати транспонуванням:

$$\left(SC_{16}^{VII}\right)^{-1} = SC_{16}^{VII*T} \tag{38}$$

Матрицю $\left(SC_{16}^{VII}\right)^{-1}$ на основі (38), (36), (37) з урахуванням симетричності фактор матриць $\left(H_{16}^{*T} = H_{16}^*, H_4^{*T} = H_4^*, H_4^{0T} = H_4^0, H_4^T = H_4, T_2^T = T_2\right)$ можна факторизовано представити як добуток п’яти транспонованих фактор–матриць:

$$\left(SC_{16}^{VII}\right)^{-1} = S_{16,1} S_{16,2}^T S_{16,3}^T S_{16,4}^T S_{16,5},\tag{39}$$

де $S_{16,k}^T$ – k–і, $k = \overline{1,5}$, транспоновані фактор–матриці 16×16 алгоритму запропонованого 16–точкового швидкого обчислення спрощеного ЦСКП–VII:

$$\begin{aligned} &S_{16,1} = H_{16}^*,\ S_{16,2}^T = diag\left[S_8^{VII\,T}, \bar{H}_8^{*T}\right],\ S_{16,k+1}^T = diag\left[I_8, C_{8,k}^T\right], k = 2,3,4 i 5, \\ &C_{8,2}^T = diag\left[H_4^*, R_{4i}^T\right],\ C_{8,3}^T = diag\left[T_2, Q_2^T, H_4\right],\ C_{8,4}^T = C_{8,4}, \\ &Q_2^T = \begin{bmatrix} \bar{j} & -i \\ i & \bar{j} \end{bmatrix},\ R_4^T = \begin{bmatrix} d & & & -a \\ & c & b & \\ & b & -c & \\ a & & & d \end{bmatrix},\ \bar{H}_8^{*T} = \begin{bmatrix} I_4 & -\bar{I}_4 \\ \bar{I}_4 & I_4 \end{bmatrix}, \end{aligned} \tag{40}$$

$S_8^{VII\,T}$ – транспонована матриця 8×8 ядра ЦСП–VII [101].

Тут $C_{8,k}^{T}$ – k–і, $k=\overline{1,4}$, транспоновані фактор–матриці 8×8 алгоритму 8–точкового швидкого обчислення зворотного ЦКП, запропонованого у роботі [32], де наведено і граф алгоритму.

Обчислювальна складність. Для реалізації відомого швидкого алгоритму обчислення 1D 8–точкового прямого ЦКП-ІІ, запропонованого в [32], потрібно 14 операцій множення, 26 операцій додавання і 6 операцій зсуву.

Для реалізації запропонованого цілочислового однонормового синусного перетворення типу VII порядку 8, на основі алгоритму матричного множення, потрібно 64 операції множення і 56 операцій додавання.

Обчислювальну складність запропонованого алгоритму швидкого обчислення 1D 16–точкового спрощеного ЦСКП–VII і відомого алгоритму обчислення з цілочисловою апроксимацією ДСП–VII [102], яке застосовано у стандарті VVC [54], наведено у табл. 8

Таблиця 8

Операції	Оцінка обчислювальної складності 1D 16–точкових перетворень		Порівняльний аналіз
	Запропонованого ЦСКП–VII	Відомих ДСП–VII із [102] VVC	
	Часткова факторизація (partial butterfly)	Матричне множення	
Множення	78	127	На 38,58% менше
Додавання + зсув	98+4	171	На 40,35% менше додавань

Цілочислові швидкі роздільні адаптивні перетворення ЦКП-ІІ/ЦСКП–VII для блоків 16 16. Для реалізації алгоритму швидкого обчислення 1D 16–точкового ЦКП-ІІ, запропонованого в [32], потрібно 40 операцій множення, 72 операції додавання і 16 операцій зсуву. Обчислювальну складність запропонованого швидкого алгоритму обчислення 2D 16–точкових роздільних ЦКП-ІІ [32]/ЦСКП–VII і відомих алгоритмів роздільних ЦКП-ІІ (H.265) [8,103]/ЦСП–VII [102] VVC наведено у табл. 9.

Таблиця 9

Операції (оп./пікс.)	Оцінка обчислювальної складності 2D роздільних перетворень для блоків 16×16		Порівняльний аналіз
	Запропонованого ЦКП-II/ЦСКП–VII	Відомих ЦКП-II [103] H.265/ДСП–VII [102] VVC	
Множення	7,375	13,3125	На 44,6% менше множень
Додавання + зсув	11,75	17,9375	На 34,49% менше додавань

Спрощене ЦСКП–VII з точнішою апроксимацією. Для апроксимації дискретного спрощеного синус-косинусного перетворення гібридного типу VII-II порядку 16 було застосоване відоме двохкомпонентне цілочислове косинусне перетворення типу II порядку 8 зі швидкими алгоритмами обчислення [32]. Нескладно розглянути інший варіант апроксимації з використанням однокомпонентного цілочислового косинусного перетворення порядку 8 зі швидкими алгоритмами обчислення, наприклад, із [99]. Для збільшення точності апроксимації синусного перетворення типу VII замість $\boldsymbol{k}$=21 краще взяти $\boldsymbol{k}$=63, тобто до цілочислового косинусного перетворення порядку 8 з $\boldsymbol{k}$=21 застосувати масштабуючий множник, який дорівнює трьом. Відповідно, отримаємо наступну матрицю цілочислового косинусного перетворення типу II порядку 8:

$$C_8^{II} = \begin{bmatrix} 63 & 63 & 63 & 63 & 63 & 63 & 63 & 63 \\ 90 & 72 & 48 & 18 & -18 & -48 & -72 & -90 \\ 75 & 48 & -48 & -75 & -75 & -48 & 48 & 75 \\ 72 & -18 & -90 & -48 & 48 & 90 & 18 & -72 \\ 63 & -63 & -63 & 63 & 63 & -63 & -63 & 63 \\ 48 & -90 & 18 & 72 & -72 & -18 & 90 & -48 \\ 48 & -75 & 75 & -48 & -48 & 75 & -75 & 48 \\ 18 & -48 & 72 & -90 & 90 & -72 & 48 & -18 \end{bmatrix} \quad (41)$$

Алгоритм швидкого обчислення масштабованого перетворення буде мати ту саму схему, що і для $\boldsymbol{k}$=21 з [99], але обчислювальна складність незначно зміниться, оскільки параметри ***r, s, k*** збільшаться в три рази, при цьому параметр ***q*** залишиться той самий. Відповідно, прямий алгоритм швидкого обчислення буде мати 14 множень, 28 додавань і 4 зсуви, разом 46 операцій, тобто обидва розглянуті цілочислові косинусні перетворення порядку 8 мають однакову обчислювальну складність.

Нескладно обчислити цілочислову апроксимацію синусного перетворення типу VII порядку 8 для $\boldsymbol{k}$=63, що буде мати вигляд:

$$S_8^{VII} = \begin{bmatrix} 16 & 31 & 46 & 58 & 69 & 78 & 83 & 86 \\ 46 & 78 & 86 & 69 & 31 & -16 & -58 & -83 \\ 69 & 83 & 31 & -46 & -86 & -58 & 16 & 78 \\ 83 & 46 & -58 & -78 & 16 & 86 & 31 & -69 \\ 86 & -16 & -83 & 31 & 78 & -46 & -69 & 58 \\ 78 & -69 & -16 & 83 & -58 & -31 & 86 & -46 \\ 58 & -86 & 69 & -16 & -46 & 83 & -78 & 31 \\ 31 & -58 & 78 & -86 & 83 & -69 & 46 & -16 \end{bmatrix} \quad (42)$$

Обчислювальна складність цього синусного перетворення з $\boldsymbol{k}$=63 також не зміниться у порівнянні з апроксимацією для $\boldsymbol{k}$=64.

Побудоване на основі двох перетворень порядку 8 (41) і (42) цілочислове спрощене синус-косинусне перетворення порядку 16 буде мати вигляд:

$$SC_{16}^{VII} = \begin{bmatrix} 16 & 31 & 46 & 58 & 69 & 78 & 83 & 86 & 86 & \cdots & 16 \\ 63 & 63 & 63 & 63 & 63 & 63 & 63 & 63 & -63 & \cdots & -63 \\ 46 & 78 & 86 & 69 & 31 & -16 & -58 & -83 & -83 & \cdots & 46 \\ 90 & 72 & 48 & 18 & -18 & -48 & -72 & -90 & 90 & \cdots & -90 \\ 69 & 83 & 31 & -46 & -86 & -58 & 16 & 78 & 78 & \cdots & 69 \\ 75 & 48 & -48 & -75 & -75 & -48 & 48 & 75 & -75 & \cdots & -75 \\ 83 & 46 & -58 & -78 & 16 & 86 & 31 & -69 & -69 & \cdots & 83 \\ 72 & -18 & -90 & -48 & 48 & 90 & 18 & -72 & 72 & \cdots & -72 \\ 86 & -16 & -83 & 31 & 78 & -46 & -69 & 58 & 58 & \cdots & 86 \\ 63 & -63 & -63 & 63 & 63 & -63 & -63 & 63 & -63 & \cdots & -63 \\ 78 & -69 & -16 & 83 & -58 & -31 & 86 & -46 & -46 & \cdots & 78 \\ 48 & -90 & 18 & 72 & -72 & -18 & 90 & -48 & 48 & \cdots & -48 \\ 58 & -86 & 69 & -16 & -46 & 83 & -78 & 31 & 31 & \cdots & 58 \\ 48 & -75 & 75 & -48 & -48 & 75 & -75 & 48 & -48 & \cdots & -48 \\ 31 & -58 & 78 & -86 & 83 & -69 & 46 & -16 & -16 & \cdots & 31 \end{bmatrix}$$

Обчислювальна складність для нього буде така сама, як і для першого запропонованого цілочислового спрощеного однонормового синус-косинусного перетворення з $\boldsymbol{k}$=64.

3.3.3 Експериментальні результати тестування

Наведемо результати тестування запропонованих перетворень на тестових даних. Для тестування будемо використовувати дві характеристики: PSNR та коефіцієнт стиснення *K*:1. Для обчислення коефіцієнту стиснення розроблено програмне забезпечення, яке використовує один алгоритм передбачення значень пікселів на основі сусіднього блоку (за середнім), а потім зберігає коефіцієнти перетворення у файл. Файл в свою чергу стискаємо програмою архіватором 7zip, і потім обчислюємо коефіцієнт стиснення як співвідношення між стиснутим файлом коефіцієнтів перетворення і файлом вхідного тестового зображення у форматі bmp. В якості тестових зображень

візьмемо синтетичне зображення «шум», як приклад зображення, що має дуже низький коефіцієнт кореляції між пікселами, та зображення «трава», як приклад зображення середньої кореляції (з урахуванням застосування до нього алгоритму передбачення). Зображення шум є штучно згенерованим, з метою отримати максимально низьку кореляцію даних, як було запропоновано в [99]. Результати тестування можна сприймати як оцінку зверху та оцінку знизу, оскільки після традиційного застосування декількох алгоритмів передбачення, кореляція буде меншою, ніж для зображення «трава» з одним алгоритмом передбачення, але більшою, ніж для зображення «шум», в цілому ближче до результатів тестування зображення «шум».

Спочатку оцінимо точність апроксимації двох застосованих цілочислових апроксимацій синусного перетворення типу VII порядку 8. Для цього отримаємо та порівняємо значення PSNR при коефіцієнті квантування QPst=1, тобто без стиснення. Для синусного перетворення типу VII порядку 8 для $\boldsymbol{k}$=64 значення PSNR для зображення шум буде дорівнювати 44,88 дБ., а для $\boldsymbol{k}$=63 тестування дає значення PSNR в 47,4 дБ., що свідчить про більш точну апроксимацію. Таким чином, друге запропоноване спрощене синус-косинусне перетворення типу VII порядку 16 з $\boldsymbol{k}$=63 також буде мати більшу точність апроксимації, ніж перше з $\boldsymbol{k}$=64, за рахунок більш точних апроксимацій складових перетворень порядку 8.

Результати тестування спрощеного синус-косинусного перетворення для $\boldsymbol{k}$=63, побудованого на базі синусного перетворення типу VII порядку 8 і косинусного перетворення типу II порядку 8 наведено в табл. 10, 11 і 12. У табл. 10 наведено експериментальні результати кодування за характеристикою PSNR (дБ) за схемою роздільних адаптивних перетворень для найбільш поширеної комбінації косинус+синус, тобто ЦКП-II/ЦСКП (розроблене синус-косинусне перетворення пропонується на заміну синусного перетворення) та ЦКП-II/ЦСП–VII, що використовуються в стандарті відеокодування VVC, а також для порівняння ефективності протестовано 2D ЦСП–VII і 2D ЦСКП, тобто відоме та запропоноване.

Таблиця 10

Зображення	Перетворення	Результати кодування за характеристикою PSNR (дБ) для QP			
		22	27	32	37
Шум	ЦКП-II/ ЦСКП	40,65	35,95	31,34	26,13
	ЦКП-II/ ЦСП–VII	40,67	35,97	31,35	26,15
	2D ЦСП–VII	40,305	35,837	31,301	26,111
	2D ЦСКП	40,143	35,773	31,274	26,109
трава	ЦКП-II/ ЦСКП	41,08	36,34	31,74	26,86
	ЦКП-II/ ЦСП–VII	41,02	36,33	31,74	26,87
	2D ЦСП–VII	40,670	36,089	31,568	26,773
	2D ЦСКП	40,772	36,286	31,527	26,736

У табл. 11 наведено експериментальні результати ефективності кодування за характеристикою PSNR (дБ) для комбінації ЦКП-II/ЦСКП відносно ЦКП-II/ЦСП–VII і для 2D ЦСКП відносно 2D ЦСП–VII.

Експериментальні результати кодування за коефіцієнтом стиснення *K*:1 за схемою роздільних адаптивних перетворень для ЦКП-II/ЦСКП та ЦКП-II/ЦСП–VII, а також для 2D ЦСП–VII і 2D ЦСКП, наведено в табл. 12. Коефіцієнт стиснення для схеми роздільних адаптивних перетворень є більшим, ніж для 2D ЦСП–VII, що ілюструє перевагу комбінації косинус/синус перед комбінацією синус/синус.

Таблиця 11

Зображення	Відносно	Результати ефективності кодування за характеристикою PSNR (дБ) для QP			
		22	27	32	37
Шум	ЦКП-II/ЦСП–VII	0,06	0,01	0,02	-0,014
	2D ЦСП–VII	-0,162	-0,064	-0,027	-0,002
трава	ЦКП-II/ЦСП–VII	0,116	0,146	-0,01	-0,014
	2D ЦСП–VII	0,102	0,197	-0,041	-0,037

Таблиця 12

Зображення	Перетворення	Результати кодування за коєфіцієнтом стиснення К:1 для QP			
		22	27	32	37
Шум	ЦКП-II/ ЦСКП	1,85	2,29	2,82	3,70
	ЦКП-II/ ЦСП–VII	1,88	2,32	2,85	3,73
	2D ЦСП–VII	1,84	2,27	2,79	3,65
	2D ЦСКП	1,79	2,22	2,72	3,56
трава	ЦКП-II/ ЦСКП	2,08	2,59	3,28	4,65
	ЦКП-II/ ЦСП–VII	2,09	2,62	3,33	4,72
	2D ЦСП–VII	2,04	2,54	3,23	4,58
	2D ЦСКП	1,99	2,50	3,14	4,45

Результати ефективності кодування за коефіцієнтом стиснення *K* (у відсотках, %) в схемі роздільних адаптивних перетворень для запропонованого ЦКП-II/ЦСКП відносно відомих ЦКП-II/ЦСП–VII наведені у табл. 13.

Таблиця 13

Зображення	Результати ефективності кодування за характеристикою К (в %) для QP			
	22	27	32	37
Шум	-1,60	-1,29	-1,05	-0,80
Трава	-0,48	-1,15	-1,5	-1,48

Запропоноване роздільне направлене ЦКП/ЦСКП–VII для блоку 16×16 у порівнянні із відомими роздільними перетвореннями ЦКП (H.265)/ЦСП–VII за характеристикою PSNR для зображення «трава» збільшує значення PSNR на 0,146 дБ для QP=27, а для інших значень QP і для зображення «шум» спостерігаються незначні коливання в обидві сторони. При цьому значення коефіцієнту стиснення *K* зменшується на 1,29-1,05% для найбільш поширених значень QP 27-32 для зображення «шум», і на 1,15-1,48% для зображення «трава».

Тестування на моделях. Крім тестування на реальних тестових даних, проводять тестування на моделях зображень, для яких обчислюють такі характеристики, як ефективність перетворення та coding gane G_{TC}.

Графік ефективності перетворення, що представлено на рис. 8, демонструє досить високі декореляційні властивості запропонованого перетворення, хоча треба враховувати, що це теоретично можливі результати для перетворень, без врахування впливу квантування.

Результати за характеристикою coding gane G_{TC} для параметрів корреляції від 0,2 до 0,6 наведено в табл. 14. Вони демонструють приблизно таку ж картину, як і тестування на реальних даних – тобто незначне відставання запропонованого перетворення. Після застосування алгоритмів передбачення значень пікселів корреляція вхідних даних падає до 0,2-0,4, в залежності від ефективності алгоритмів і кореляції вхідних даних, тому тестувати на більших значеннях кореляції немає підстав. Зауважимо, що при низьких значеннях корреляції ефективність різних перетворень є достатньо близькою. Також для порівняння наведено результати тестування перетворення Адамара порядку 16, яке є найшвидшим для перетворень з повною матрицею значень. Розріджені матриці перетворень можуть мати більшу швидкодію, ніж повні, і представляють певну цікавість для подальших досліджень перетворень для збільшення швидкодії кодування.

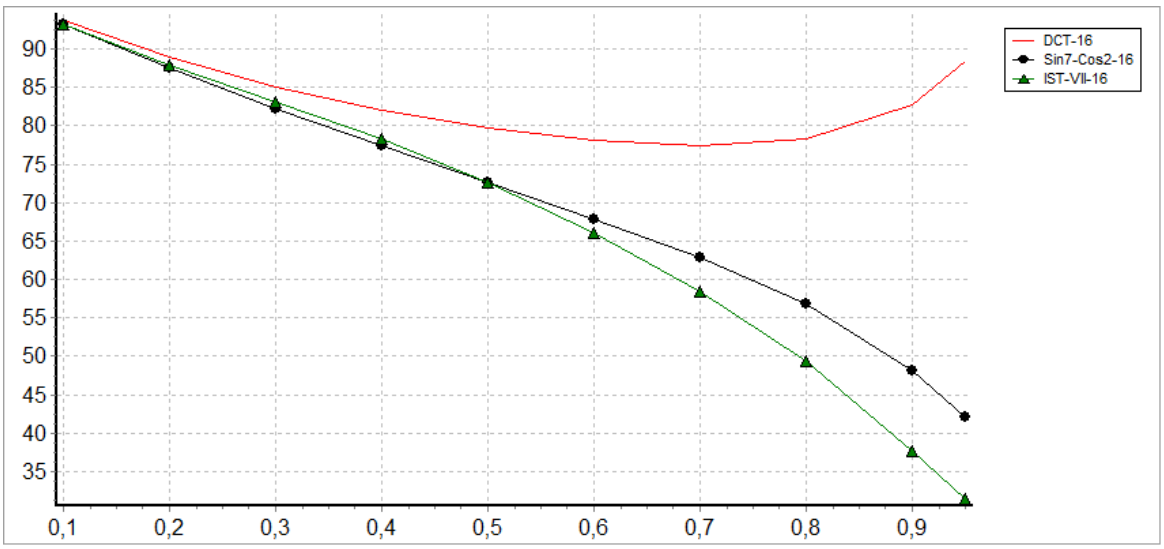

Рисунок 8 Графік характеристики ефективність кодування

Таблиця 14

Перетворення для блоків 16×16	Оцінка ефективності кодування за характеристикою 1D G_{TC} (дБ) для коефіцієнтів кореляції $, \ _h = 0,2 \div 0,6$, 1D ізотропний марковський процес				
	0,2	0,3	0,4	0,5	0,6
IST-VII	0,16	0,38	0,69	1,14	1,76
DCT-16	0,16	0,37	0,69	1,14	1,78
Adamara 16	0,12	0,28	0,52	0,87	1,37
ISCT VII-II 16	0,15	0,34	0,64	1,06	1,64

Також можна навести графіки середньоквадратичної похибки mse для моделей зображень, для обраного значення коефіцієнта корреляції. Наведемо графіки для значень параметра корреляції 0,3 на рис. 9.

Як бачимо, за цією характеристикою запропоноване перетворення досить близьке до синусного VII типу, аналогічно до характеристики PSNR для тестування на реальних даних. Тобто якість відновлених даних після застосування запропонованого перетворення є високою, не гірше за відоме синусне перетворення типу VII, а основний недолік пов’язано з меншим коефіцієнтом стиснення. Порівняння коефіцієнта стиснення запропонованого перетворення і відомого здійснювалось для всіх коефіцієнтів перетворення, що є оцінкою з певною похибкою, оскільки для реального кодування використовують лише частину коефіцієнтів перетворення, а не всі. Тобто для результатів реального кодування різниця між запропонованим і відомим перетвореннями може бути як трохи меншою, так і трохи більшою. Також не було враховано застосування алгоритму зчитування коефіцієнтів за певною схемою. Але ці похибки досить незначні, і в цілому отримані результати тестування й порівняння можна вважати достатніми для оцінки в першому наближенні.

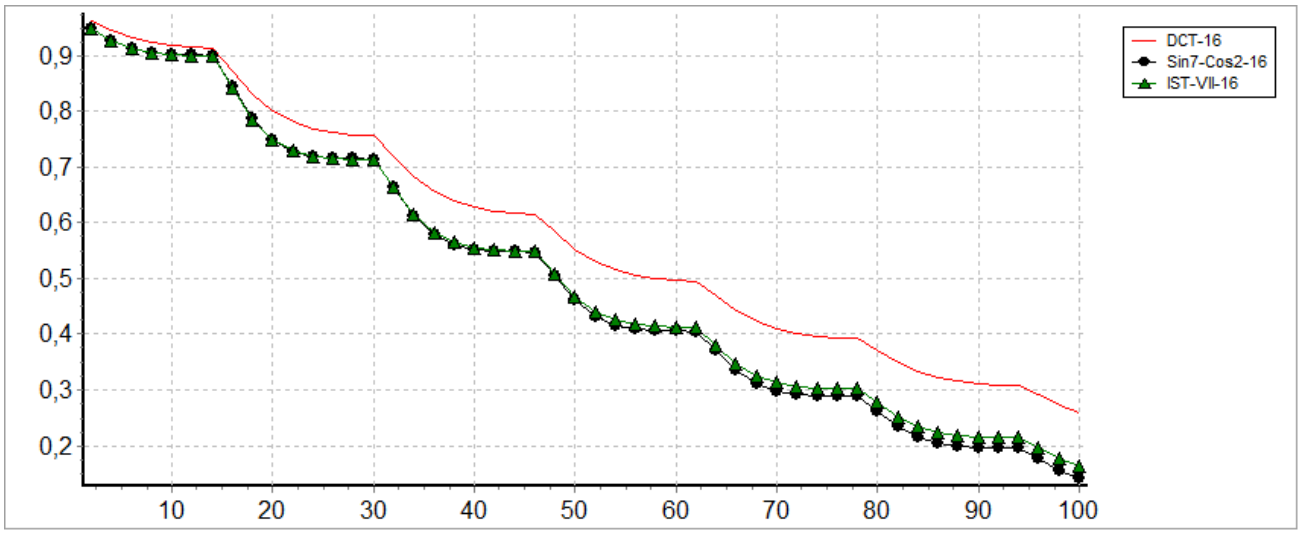

Рисунок 9. Графік mse в залежності від кількості коефіцієнтів перетворення для параметра корреляції 0,3

3.4 Цілочислове синусно-косинусне перетворення типу II порядку 4

Цілочислові перетворення різних типів зі швидкими алгоритмами обчислення широко застосовуються в різних галузях. Найпростішим з них є перетворення Адамара, що має швидкий алгоритм низької складності без операцій множення, і має одну норму за замовчуванням. Завдяки низькій обчислювальній складності перетворення Адамара має численні застосування, незважаючи на більш низьку ефективність стиснення даних (приблизно на 3-5% менше за косинусне перетворення для порядку 4, на 6-10% менше для порядку 8 і на 10-15% для порядку 16, в залежності від кореляції вхідних даних: при більш низькій кореляції ефективність вища). Також перетворення Хаара, Хаара-Адамара мають лінійну обчислювальну складність, але потребують додатковий обчислювальний ресурс на операції нормування.

Окреме місце займають однонормові цілочислові перетворення, які мають меншу обчислювальну складність, ніж їх аналоги з різною нормою. Наприклад, в кодеку H.264 використовується цілочислове косинусне перетворення порядку 4, яке потребує лише 12 операцій, якщо не враховувати

нормування, для якого використовується наступна матриця нормування, що збільшує обчислювальну складність на 12 операцій множення:

$$\begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

$a = 1/2, b = 1/\sqrt{10}.$

Для задач інтернету речей потрібні ефективні перетворення низької обчислювальної складності, на основі яких будуються спрощені кодеки [37].

Метод побудови синус-косинусного перетворення порядку 4. Метою модифікації відомого синусного перетворення типу II порядку 4 є підвищення ефективності перетворення, зокрема, для використання в якості альтернативи обчислювально більш складного синусного перетворення типу VII в роздільних адаптивних перетвореннях. Схема модифікації для цілочислового синусного перетворення порядку 4 включає використання двох значень параметрів перетворень замість трьох, а також заміну двох рядків синусного перетворення на аналогічні рядки косинусного перетворення порядку 4, що в результаті формує синус-косинусне перетворення порядку 4. Іншими словами, масштабована матриця Адамара порядку 2, яка є складовою синусного перетворення типу II порядку 4, заміняється на матрицю косинусного перетворення типу II порядку 2, яка є складовою косинусного перетворення порядку 4. Аналогічну заміну можна застосовувати в синусних та косинусних перетвореннях II типу будь-якого порядку, але з ростом порядку ефективність доданого косинусного перетворення порядку 2 замість перетворення Адамара порядку 2 буде зменшуватись пропорційно. Обчислювальна складність в результаті такої заміни незначно збільшується.

Шаблон косинусного перетворення типу II порядку 4 має вигляд:

$$C_4 = \begin{bmatrix} k & k & k & k \\ i & j & -j & -i \\ k & -k & -k & k \\ j & -i & i & -j \end{bmatrix}, \; i > j \, .$$

Аналогічно, синусне перетворення порядку 4 буде мати вигляд

$$S_4 = \begin{bmatrix} j & i & i & j \\ k & k & -k & -k \\ i & -j & -j & i \\ k & -k & k & -k \end{bmatrix}.$$

Якщо взяти рядки з параметрами ***i*** та ***j*** з косинусного перетворення, і замінити ними рядки з параметром ***k*** синусного перетворення, отримаємо синус-косинусне перетворення, що має лише два параметри ***i*** та ***j***, та приймає наступний вигляд:

$$SC_4 = \begin{bmatrix} j & i & i & j \\ i & j & -j & -i \\ i & -j & -j & i \\ j & -i & i & -j \end{bmatrix}, \; j < i$$

Нескладно розглянути схему побудови шляхом використання матриці порядку 2×2 косинусного перетворення замість матриці масштабованого перетворення Адамара, яка є складовою синусного перетворення типу II порядку 4.

Синусне перетворення порядку 4 складається з двох перетворень порядку 2:

$$S_2 = \begin{bmatrix} j & i \\ i & -j \end{bmatrix},\ T_2 = \begin{bmatrix} k & k \\ -k & k \end{bmatrix},$$

Оскільки

$$S_4 = diag\left[S_2, T_2 \right] H_4,$$

$$\text{де } H_4 = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{vmatrix}.$$

Як нескладно помітити, перетворення T_2 є масштабованим з параметром ***k*** відомим перетворенням Адамара порядку 2:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Як відомо, перетворення Адамара є гіршим за ефективністю ніж синусні та косинусні перетворення, тому доцільним виглядає замінити його на більш складне перетворення порядку 2, наприклад, взяте з косинусного перетворення порядку 4.

Косинусне перетворення порядку 4 теж складається з двох перетворень порядку 2

$$T_2 = \begin{bmatrix} k & k \\ k & -k \end{bmatrix},\ C_2 = \begin{bmatrix} j & i \\ -i & j \end{bmatrix}$$

$$C_4 = diag\left[T_2, C_2 \right] H_4$$

Відповідно, ми можемо взяти матрицю C_2 з косинусного перетворення, і отримати нове перетворення, що складається з синусного та косинусного перетворення порядку 2:

$$SC_4 = diag\left[S_2, C_2 \right] H_4$$

Таким чином, отримане синус-косинусне перетворення типу II порядку 4 теоретично повинно мати кращу ефективність за традиційне синусне перетворення типу II порядка 4.

Зауважимо, що косинусне перетворення порядку 4 також включає в якості складової частини масштабоване перетворення Адамара порядку 2, і можна було б аналогічно замінити його на синусне перетворення порядку 2 з параметрами ***i*** та ***j***, але в результаті було б отримане те ж саме синус-косинусне перетворення.

Цілочисельні однонормові синус-косинусні перетворення типу II порядку 4. Для розробки нового синус-косинусного перетворення з однією нормою треба підібрати три значення параметрів (k, i, j) згідно вимоги однієї норми

$$2k^2 = i^2 + j^2$$

Для порядку 4 ця задача є досить простою, але повністю під цю умову підходять дуже мало можливих наборів значень, наприклад, добре відоме перетворення (13,17,7), що є повністю ортогональним і не має похибки нормування. З урахуванням незначної похибки відхилення від однієї норми виникає більше варіантів перетворення, наприклад, (7,9,4), яке має похибку нормування в 1% для 2D варіанту та 0,5% - для схеми роздільних перетворень.

Для параметрів (7,9,4) матриця цілочислового синус-косинусного перетворення типу II порядку 4 буде мати вигляд:

$$SC_4^7 = \begin{bmatrix} 4 & 9 & 9 & 4 \\ 9 & 4 & -4 & -9 \\ 9 & -4 & -4 & 9 \\ 4 & -9 & 9 & -4 \end{bmatrix}$$

А для (13,17,7) відповідно

$$SC_4^{13} = \begin{bmatrix} 7 & 17 & 17 & 7 \\ 17 & 7 & -7 & -17 \\ 17 & -7 & -7 & 17 \\ 7 & -17 & 17 & -7 \end{bmatrix}$$

З точки зору обчислювальної складності, перетворення SC_4^7 буде мати перевагу, оскільки потребує 4 множення, 8 додавань і 4 зсуви, тобто 16 операцій, або 20 операцій додавання і зсувів, якщо замінити кожну операцію множення на число 9 на 2 операції зсуву та додавання. Крім того, операція нормування також буде мати малу обчислювальну складність в схемі роздільних адаптивних перетворень. Друге перетворення SC_4^{13} буде мати трохи кращу точність для малих параметрів квантування від QPst 1 до 8, але більшу обчислювальну складність: 8 множень та 8 додавань, або 24 операції додавання і зсуву без операцій множення, а також більшу обчислювальну складність при нормуванні.

В роботі [28] було запропоновано швидкий алгоритм обчислення синусного перетворення типу VII порядку 4, яке має 5 операцій множення та 11 операцій додавання. Таким чином, застосовувати запропоноване синус-косинусне перетворення замість синусного перетворення типу VII порядку 4 недоцільно, альтернативні перетворення нижчої обчислювальної складності

більш необхідні для порядку 8, де для синусного перетворення типу VII нема швидких алгоритмів.

Швидкий алгоритм обчислення синус-косинусного перетворення типу II порядку 4. Матриця ядра IST порядка 4 може бути представлена як добуток двох матриць:

$$S_4^* = S_{4,2} S_{4,1},$$

де $S_{4,1}$, $S_{4,2}$ – фактор-матриці 4×4 швидкого алгоритма 4-точечного прямого IST:

$$S_{4,1} = H_4^*,\ S_{4,2} = diag\left[S_2, C_2 \right],$$

$$H_4^* = \begin{bmatrix} I_2 & \bar{I}_2 \\ \bar{I}_2 & -I_2 \end{bmatrix},\ \bar{I}_2 = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix},\ \bar{I}_2 = antidiag\left[I_2 \right].$$

Тут I_2 і $\bar{I}_2$ - одинична діагональна і антидіагональна матриці 2×2. Матриці S_2, C_2 мають вигляд:

$$S_2 = \begin{bmatrix} j & i \\ i & -j \end{bmatrix},\ C_2 = \begin{bmatrix} j & i \\ -i & j \end{bmatrix}$$

Цілочислові синусні і косинусні перетворення порядку II для роздільних адаптивних перетворень. Для спрощеної схеми квантування, при якій використовуються значення параметру квантування, які дорівнюють ступеню двійки, можна розглянути такі пари роздільних адаптивних перетворень, які при сумісному квантуванні будуть мати мінімальну обчислювальну складність. Наприклад, якщо використати косинусне

перетворення порядку 4 з [99] з $\boldsymbol{k}$=18, та синусне перетворення з $\boldsymbol{k}$ =7, то операція квантування буде включати ділення на $7 \cdot 18 \cdot 4$, тобто ділення на 63 і 8, що заміняється множенням на 65 (яке можна замінити на дві операції – додавання і зсув) і відповідним зсувом, що об'єднується в один зсув з $2 \cdot 4$.

Експериментальні результати тестування. Наведемо результати тестування запропонованих перетворень на тестових зображеннях. Для тестування будемо використовувати дві характеристики: PSNR та коефіцієнт стиснення $\boldsymbol{K}$:1. Для обчислення коефіцієнту стиснення розроблено програмне забезпечення, що зберігає коефіцієнти перетворення у файл, який стискається програмою архіватором 7zip, і потім обчислюється коефіцієнт стиснення як співвідношення між стиснутим файлом коефіцієнтів перетворення і файлом вхідного зображення у форматі bmp. В якості тестового зображення взято синтетичне зображення «шум» з роботи [99], яке має дуже низький коефіцієнт кореляції між пікселами, і моделює максимально ефективний результат роботи алгоритмів передбачення значень пікселів.

В таблицях 15 і 16 наведено результати тестування роздільних перетворень порядку 4, синусного перетворення типу VII і розробленого синус-косинусного перетворення типу II. Зауважимо, що коефіцієнт стиснення для перетворень порядку 4 очікувано менший, ніж для аналогічних перетворень порядку 8, а от PSNR навпаки, вищий.

Таблиця 15

Перетворення ICT/IST 4×4	Результати ефективності кодування за характеристикою PSNR (дБ) для QP			
	22	**27**	**32**	37
ICT/ISCT	40,627	**35,951**	**31,352**	26,140
ICT/IST-VII	40,685	**35,968**	**31,350**	26,136
Різниця	-0,058	-0,017	0,002	-0,004

Таблиця 16

Перетворення ICT/IST 4×4	Результати кодування за коєфіцієнтом стиснення ***K***:1 для QP			
	22	**27**	**32**	37
ICT/ISCT	1,68	**2,07**	**2,55**	3,29
ICT/IST-VII	1,71	**2,11**	**2,60**	3,33
Різниця в %	1,75	**1,90**	**1,92**	1,20

Таким чином, результати тестування показують, що для порядку 4 запропоноване синус-косинусне перетворення відстає від синусного перетворення типу VII до 1,92% для низької кореляції вхідних даних.

3.4 Висновки до розділу

Гібридним ортогональним перетворенням є таке ортогональне перетворення, яке має половину функцій від ортогонального перетворення одного типу, а другу половину – від ортогонального перетворення іншого типу. В цілому можливі два підходи формування гібридних перетворень – комбінація перетворень меншого порядку (відомий класичний метод) і комбінація перетворень рівного порядку.

Розглянуто три приклади гібридних перетворень – косинус-синусне типу IV порядку 16, спрощене синусно-косинусне типу VII порядку 16 та синусно-косинусне перетворення типу II порядку 4. Запропоноване спрощене цілочислове однонормове синусно-косинусне перетворення типу VII порядку 16 може бути альтернативою синусного перетворення типу VII порядку 16 за критерієм обчислювальної складності. При цьому значення коефіцієнту стиснення *K* для запропонованого перетворення зменшується на 1,3-1,05% для найбільш поширених значень QP 27-32 для зображення «шум», і на 1,15-1,5% для зображення «трава» відносно відомого перетворення.

Синусно-косинусне перетворення типу II порядку 4 побудовано на базі двох відомих перетворень типу II порядку 4 – косинусного і синусного. Воно має кращі показники, ніж синусне перетворення, і може використовуватись як базове перетворення для отримання нових ортогональних перетворень, зокрема спрощенного синус-косинусного перетворення типу II порядку 8, яке розглядається в наступному розділі. Метод заміни масшабованого перетворення Адамара порядку 2 на синусне перетворення порядку 2, що був застосований для отримання синусно-косинусного перетворення типу II порядку 4, може бути застосований до синусних та косинусних перетворень будь-якого порядку. В наступному розділі розглядаються модифікації синусного і косинусного перетворень порядку 8 на основі цього методу.

РОЗДІЛ 4. МУЛЬТИГІБРИДНІ ОРТОГОНАЛЬНІ ПЕРЕТВОРЕННЯ

Подальшим розвитком і доповненням гібридних підходів є запропонована концепція мультигібридних ортогональних перетворень.

Пропонується два методи отримання нових ортогональних цілочислових однонормових мультигібридних перетворень.

Перший метод базується на використанні двох відомих гібридних перетворень, на базі яких будується нове мультигібридне перетворення. Враховуючи, що гібридні перетворення є спрощеними, оскільки будуються на основі відомих ортогональних перетворень меншого порядку, то мультигібридні перетворення, отримані цим методом, будуть ще простіші, тобто наближені до простих перетворень типу Адамара, Уолша, Хаара. Таким чином, мультигібридні перетворення, отримані на базі двох гібридних перетворень, є проміжними за складністю між простими перетвореннями типу Адамара і складними перетвореннями типу косинусних та синусних. В якості прикладу можна побудувати спрощене синус-косинусне перетворення типу II порядку 8 на базі двох однакових гібридних перетворень – синус-косинусних типу II порядку 4.

Другий метод отримання мультигібридних перетворень – це модифікація відомих перетворень шляхом заміни меншої частини перетворення на інше перетворення. В якості прикладу можна розглянути модифікацію синусного або косинусного перетворення типу II шляхом заміни складового перетворення порядку 2 на інше перетворення, скажімо, за алгоритмом, що використано для побудови синус-косинусного перетворення порядку 4. Тобто масшабоване перетворення Адамара порядку 2, яке є складовою частиною косинусних та синусних ортогональних перетворень типу II, заміняється на синусне перетворення порядку 2. Отримані в результаті модифіковані косинусні та синусні перетворення можуть бути альтернативою синусним перетворенням типу VII за критерієм обчислювальної складності.

Розширенням сімейства мультигібридних та відомих перетворень є пошук таких цілочисельних апроксимацій для дискретних перетворень, що мають більшу похибку апроксимації, ніж відомі цілочисельні перетворення. Отримані в результаті модифіковані цілочисельні перетворення можуть мати адаптивне застосування разом з відомими перетвореннями. Частковим випадком таких мультигібридних перетворень можуть бути розріджені перетворення, матриці яких мають нулі, що зменшує обчислювальну складність подібних ортогональних перетворень. Це перспективний напрямок досліджень, особливо для застосування в швидкодійних традиційних чи гібридних кодеках зображень та відео.

4.1 Спрощене цілочислове однонормове синус-косинусне перетворення типу II порядку 8

На основі двох синус-косинусних перетворень порядку 4 можна побудувати синус-косинусне перетворення порядку 8, яке буде мати меншу обчислювальну складність і відповідно меншу ефективність, ніж чисте синусне перетворення типу II порядку 8, тому його можна назвати спрощеним синус-косинусним перетворенням типу II порядку 8.

Даний метод є відомим, наприклад, з його використанням було побудовано перетворення порядку 8 на основі двох перетворень порядку 4, а саме – синусного перетворення типу 7 і косинусного перетворення типу II в роботі [29], і може бути представлений як

$$T_8 = diag\left[T_4^1, T_4^2\right] H_8 .$$

В нашому випадку $T_4^1 = T_4^2 = SC_4$, тобто

$$SC_8 = diag\left[SC_4, SC_4\right] H_8$$

де $H_8^* = \begin{bmatrix} I_4 & \bar{I}_4 \\ \bar{I}_4 & -I_4 \end{bmatrix}$, $\bar{I}_4 = antidiag[I_4]$, I_4, $\bar{I}_4$ – одиничні діагональна і антидіагональна матриці 4×4.

Оскільки перший рядок матриці є синусним, це визначає, що дане перетворення може застосовуватись в схемі роздільних адаптивних перетворень замість синусного перетворення, незважаючи на наявність косинусних функцій. Аналогічно, гібридні косинус-синусні перетворення можуть бути застосовані замість косинусного перетворення для роздільних перетворень.

Матриця спрощеного синус-косинусного перетворення для параметрів (k, i, j) буде мати наступний узагальнений вигляд:

$$SC_8 = \begin{bmatrix} j & i & i & j & j & i & i & j \\ j & i & i & j & -j & -i & -i & -j \\ i & j & -j & -i & i & j & -j & -i \\ i & j & -j & -i & -i & -j & j & i \\ i & -j & -j & i & i & -j & -j & i \\ i & -j & -j & i & -i & j & j & -i \\ j & -i & i & -j & j & -i & i & -j \\ j & -i & i & -j & -j & i & -i & j \end{bmatrix}$$

Параметр ***k*** визначає норму матриці, і визначається як

$$2k^2 = i^2 + j^2$$

Для нормування роздільних перетворень отримані коефіцієнти перетворення, як відомо, діляться на $k_1 k_2 N$, де ***N*** – порядок перетворення.

Матриця спрощеного синус-косинусного перетворення для параметрів (7,9,4) буде мати наступний вигляд:

$$SC_8 = \begin{bmatrix} 4 & 9 & 9 & 4 & 4 & 9 & 9 & 4 \\ 4 & 9 & 9 & 4 & -4 & -9 & -9 & -4 \\ 9 & 4 & -4 & -9 & -9 & -4 & 4 & 9 \\ 9 & 4 & -4 & -9 & 9 & 4 & -4 & -9 \\ 9 & -4 & -4 & 9 & 9 & -4 & -4 & 9 \\ 9 & -4 & -4 & 9 & -9 & 4 & 4 & -9 \\ 4 & -9 & 9 & -4 & -4 & 9 & -9 & 4 \\ 4 & -9 & 9 & -4 & 4 & -9 & 9 & -4 \end{bmatrix}$$

Використання двох однакових перетворень меншого порядку є частковим випадком отримання гібридних перетворень на основі перетворень меншого порядку. Наприклад, можна на основі косинусного перетворення типу II порядку 4 запропонувати спрощене косинусне перетворення порядку 8 зменшеної складності відносно традиційного косинусного перетворення порядку 8.

Швидкий алгоритм обчислення спрощеного цілочислового синус-косинусного перетворення типу II порядку 8. Матрицю CS_8 синус-косинусного перетворення порядку 8 можна факторизовано представити як добуток трьох матриць:

$$CS_8 = T_{8,3} T_{8,2} T_{8,1}$$

де $T_{8,k}$ – k-і, $k = \overline{1,3}$, фактор-матриці 8×8 запропонованого швидкого алгоритму обчислення спрощеного синус-косинусного перетворення

$$T_{8,1} = H_8,\ T_{8,2} = diag\left[H_4, H_4\right],\ T_{8,3} = diag\left[S_2, C_2, S_2, C_2\right]$$

Матриці S_2, C_2 мають вигляд:

$$S_2 = \begin{bmatrix} j & i \\ i & -j \end{bmatrix},\ C_2 = \begin{bmatrix} j & i \\ -i & j \end{bmatrix}$$

Обчислювальна схема спрощеного синус-косинусного перетворення порядку 8 буде мати в 2 рази більше операцій ніж синус-косинусне перетворення 4 порядку, і додатково 8 операцій для матриці H_8, тобто 48 операцій у випадку без використання операцій множення, або 32 операції додавання і зсуву та 8 операцій множення, для параметрів (7,9,4), що разом складає 40 операцій.

Експериментальні результати тестування. Наведемо результати тестування запропонованих перетворень на тестових зображеннях. Для тестування будемо використовувати дві характеристики: PSNR та коефіцієнт стиснення ***K***:1. Для обчислення коефіцієнту стиснення розроблено програмне забезпечення, що зберігає коефіцієнти перетворення у файл, який стискається програмою архіватором 7zip, і потім обчислюється коефіцієнт стиснення як співвідношення між стиснутим файлом коефіцієнтів перетворення і файлом вхідного зображення у форматі bmp. В якості тестового зображення взято синтетичне зображення «шум» з роботи [99], яке має дуже низький коефіцієнт кореляції між пікселами, і моделює максимально ефективний результат роботи алгоритмів передбачення значень пікселів.

В табл. 17 наведено результати тестування трьох мод, комбінацій двох цілочислових однонормових перетворень – косинусного II типу [103] та синусного 7 типу порядку 8, за параметром коефіцієнта стиснення ***K***:1. Четверта комбінація – косинус/косинус не тестується, оскільки вона за теорією має найкращий стиск, але й більш помітні артефакти. Як бачимо, найменший стиск у комбінації синус/синус, найбільший з трьох протестованих комбінацій – у косинус/синус. Що підтверджує запропоновану концепцію спрощених мод, яка отримана теоретично.

Таблиця 17

Перетворення ICT/IST 8×8	Результати кодування за коєфіцієнтом стиснення ***K***:1 для QP			
	22	**27**	**32**	37
ICT/IST-VII	1,81	**2,22**	**2,73**	3,54
IST-VII/IST-VII	1,77	**2,18**	**2,67**	3,45
IST-VII/ICT	1,79	**2,20**	**2,70**	3,50

В таблицях 18 і 19 наведено результати порівняння для однієї моди – косинус/синус для запропонованого спрощеного синус-косинусного і відомого синусного типу VII перетворень порядку 8. Як не дивно, але спрощене синус-косинусне перетворення навіть має трохи кращі показники за характеристикою PSNR, хоча в цілому приблизно на одному рівні. Найбільший коефіцієнт стиснення очікувано має комбінація відомих перетворень, а саме косинусного II типу та синусного типу VII. Жирним шрифтом виділено найбільш поширені QP 27 і 32.

Таблиця 18

Перетворення 8×8	Результати ефективності кодування за характеристикою PSNR (дБ) для QP			
	22	**27**	**32**	37
ICT/ISCT	40,76	**36,0**	**31,36**	26,15
ICT/IST-VII	40,62	**35,97**	**31,36**	26,15
Різниця	0,14	**0,03**	**0**	0

Таблиця 19

Перетворення 8×8	Результати кодування за коєфіцієнтом стиснення ***K***:1 для QP			
	22	**27**	**32**	37
ICT/ISCT	1,78	**2,19**	**2,69**	3,50
ICT/IST-VII	1,81	**2,22**	**2,73**	3,54
Різниця в %	-1,69	**-1,37**	**-1,49**	-1,14

Спрощене синус-косинусне перетворення відстає від синусного перетворення типу VII до 1,5%. З іншого боку, для QP=37 перевага синусного перетворення типу VII зменшується до 1,14%. Враховуючи, що обчислювальна складність синусного перетворення типу VII порядку 8 складає 120 операцій, тобто в 3 рази більше, ніж у розробленого спрощеного синус-косинусного перетворення типу II, є можливим адаптивно застосовувати спрощене перетворення для швидкодіючих мод [104] для низької кореляції і для високих коефіцієнтів стиснення.

Розглянемо тестування запропонованого синус-косинусного перетворення порядку 8 на моделях, і обчислимо характеристики кодінг гейн та ефективність перетворення ета на ізотропній моделі. Результати тестування за характеристикою ефективності перетворення наведено на рис. 10.

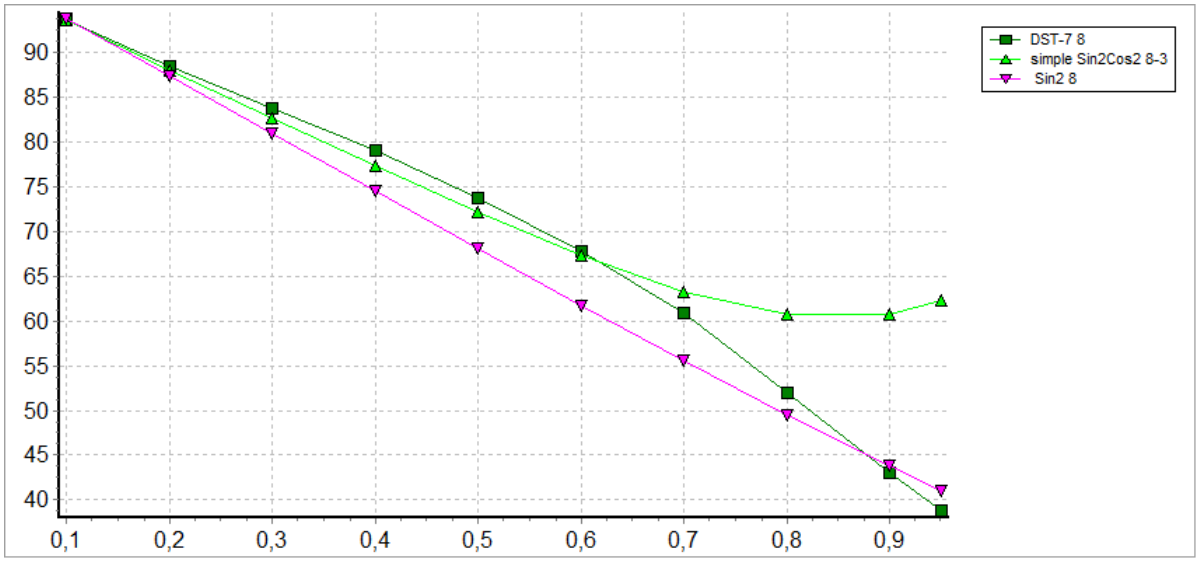

Рисунок 10 Результати тестування за характеристикою ефективність перетворення

Результати кодування за характеристикою 1D G_{2TC} наведено в таблиці 20.

Таблиця 20

Перетворення для блоків 8×8	Оцінка ефективності кодування за характеристикою 1D G_{2TC} (дБ) для коефіцієнтів кореляції $, \ _h = 0,2 \div 0,7$ 1D ізотропний марковський процес					
	0,2					
DCT-8	0,14	0,34	0,630	1,05	1,64	2,50
IST-VII-8	0,15	0,34	0,630	1,04	1,60	2,38
Simple Sin-Cos-II-8	0,14	0,32	0,58	0,94	1,43	2,10
Sin-II-8	0,14	0,31	0,56	0,89	1,32	1,89

Як бачимо, за результатами тестів на моделях, запропоноване спрощене синус-косинусне перетворення порядку 8 має кращі показники, ніж синусне перетворення типу II порядку 8, а за характеристикою ефективність перетворення досить близьке до синусного перетворення типу VII порядку 8.

4.2 Цілочислове однонормове модифіковане синусне перетворення типу II порядку 8

Другий метод побудови мультигібридних ортогональних перетворень використовує часткову заміну окремих рядків традиційної матриці, наприклад, заміну двох рядків (що рівноцінно заміні складової матриці порядку 2).

Застосуємо це підхід до відомого синусного перетворення типу II порядку 8 в якості ілюстрації запропонованого методу.

Матриця шаблону цілочислового однонормового синусного перетворення типу II порядку 8 має вигляд:

$$\begin{bmatrix} a & b & c & d & d & c & b & a \\ i & j & j & i & -i & -j & -j & -i \\ b & d & a & -c & -c & a & d & b \\ k & k & -k & -k & k & k & -k & -k \\ c & a & -d & b & b & -d & a & c \\ j & -i & -i & j & -j & i & i & -j \\ d & -c & b & -a & -a & b & -c & d \\ k & -k & k & -k & k & -k & k & -k \end{bmatrix}$$

Після заміни параметру k на параметри i, j аналогічно тому, як це було зроблено при побудові синус-косинусного перетворення порядку 4 вище, отримаємо новий шаблон нового перетворення, яке можна називати модифікованим синусним перетворенням типу II порядку 8:

$$\begin{bmatrix} a & b & c & d & d & c & b & a \\ i & j & j & i & -i & -j & -j & -i \\ b & d & a & -c & -c & a & d & b \\ j & i & -i & -j & j & i & -i & -j \\ c & a & -d & b & b & -d & a & c \\ j & -i & -i & j & -j & i & i & -j \\ d & -c & b & -a & -a & b & -c & d \\ i & -j & j & -i & i & -j & j & -i \end{bmatrix}$$

Ця незначна модифікація в результаті заміни масшабованого перетворення Адамара з параметром масштабування k порядку 2 на синусне перетворення порядку 2 підвищить ефективність декореляції отриманого перетворення. Але оскільки синусне перетворення типу II практично не використовується для кодування, ми не будемо тестувати отримане нове перетворення. Крім того, воно може бути альтернативою синусного перетворення типу VII, але аналогічна модифікація для косинусного перетворення типу II буде мати кращі показники ефективності, відповідно, саме

воно буде альтернативою для синусного перетворення типу VII. Вище було розроблено спрощене синусно-косинусне перетворення, яке також пропонується як альтернатива синусного перетворення типу VII, але воно може використовуватись адаптивно з модифікованим косинусним перетворенням типу II, оскільки має меншу обчислювальну складність, і меншу ефективність.

Обчислювальна складність отриманого модифікованого синусного перетворення типу II порядку 8 незначно (на дві операції) збільшиться порівняно з оригінальним синусним перетворенням.

Подібна модифікація може бути реалізована для синусних перетворень типу II будь-якого порядку.

4.3 Цілочислове однонормове модифіковане косинусне перетворення типу II порядку 8

Застосуємо другий метод до модифікації відомого косинусного перетворення типу II порядку 8, який має матрицю-шаблон наступного вигляду:

$$\begin{bmatrix} k & k & k & k & k & k & k & k \\ a & b & c & d & -d & -c & -b & -a \\ i & j & -j & -i & -i & -j & j & i \\ b & -d & -a & -c & c & a & d & -b \\ k & -k & -k & k & k & -k & -k & k \\ c & -a & d & b & -b & -d & a & -c \\ j & -i & i & -j & -j & i & -i & j \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix}$$

В результаті аналогічної заміни масшабованого перетворення Адамара з параметром масштабування k порядку 2 на синусне перетворення порядку 2 отримаємо модифіковане косинусне перетворення типу II порядку 8:

$$\begin{bmatrix} j & i & i & j & j & i & i & j \\ a & b & c & d & -d & -c & -b & -a \\ i & j & -j & -i & -i & -j & j & i \\ b & -d & -a & -c & c & a & d & -b \\ i & -j & -j & i & i & -j & -j & i \\ c & -a & d & b & -b & -d & a & -c \\ j & -i & i & -j & -j & i & -i & j \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix}$$

В результаті оригінальне косинусне перетворення отримало першу функцію у формі синусної функції, що дає можливість застосувати нове модифіковане косинусне перетворення типу II порядку 8 в якості альтернативи синусному перетворенню типу VII, яке не має швидких алгоритмів обчислення.

Обчислювальна складність розробленого цілочислового однонормового модифікованого косинусного перетворення типу II порядку 8 збільшиться на 2 операції відносно традиційного косинусного перетворення типу II порядку 8.

Подібна модифікація може бути застосована і для косинусних перетворень типу II порядку *N*, тобто нескладно отримати альтернативи синусному перетворенню типу VII для будь-якого порядку.

Протестуємо модифіковане косинусне перетворення типу II порядку 8 на тестовому зображенні шум, в якості альтернативи синусного перетворення типу VII порядку 8. В табл. 21 показано результати тестування за характеристикою PSNR, а в табл. 22 за характеристикою коефіцієнта стиснення К:1.

Таблиця 21

Перетворення 8×8	Результати ефективності кодування за характеристикою PSNR (дБ) для QP			
	22	**27**	**32**	37
ICT/MICT	40,78	**36,02**	**31,37**	26,15
ICT/IST-VII	40,62	**35,97**	**31,36**	26,15
Різниця	0,16	**0,05**	**0,01**	0

Таблиця 22

Перетворення 8×8	Результати кодування за коєфіцієнтом стиснення ***K***:1 для QP			
	22	**27**	**32**	37
ICT/ISCT	1,79	**2,196**	**2,70**	3,50
ICT/IST-VII	1,81	**2,22**	**2,73**	3,54
Різниця в %	-1,1	**-1,08**	**-1,1**	-1,1

Результати тестування демонструють, що запропонована альтернатива синусному перетворенню типу VII є достатньо ефективною, маючи такі ж показники якості і на 1,1% менший коефіцієнт стиснення.

4.4 Висновки до розділу

Розглянуто два методи отримання нових ортогональних цілочислових однонормових мультигібридних перетворень. Перший метод на основі двох відомих гібридних перетворень будує нове мультигібридне перетворення. Другий метод отримання мультигібридних перетворень – модифікація відомих перетворень шляхом заміни меншої частини перетворення на інше перетворення. Розширенням сімейства мультигібридних та відомих перетворень є такі цілочисельні апроксимації для дискретних перетворень, що мають більшу похибку апроксимації, ніж відомі цілочисельні перетворення. Частковим випадком таких перетворень можуть бути розріджені перетворення, матриці яких мають нулі, що зменшує обчислювальну складність подібних ортогональних перетворень. Це перспективний напрямок досліджень, особливо для застосування в швидкодійних традиційних чи гібридних кодеках зображень та відео. На основі двох синус-косинусних перетворень порядку 4 побудовано спрощене однонормове цілочислове синус-косинусне перетворення типу II порядку 8 низької обчислювальної складності. Було модифіковано два відомі перетворення: синусне і косинусне типу II порядку 8, для використання в якості заміни синусного перетворення в схемі роздільних адаптивних перетворень.

РОЗДІЛ 5. ДИСКРЕТНЕ МУЛЬТИВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ НА ОСНОВІ МУЛЬТИВЕЙВЛЕТ-ПАКЕТА РОЗМІРУ 8 8

Мультивейвлети мають більшу гнучкість конструювання за рахунок використання декількох масштабних і вейвлет функцій. Але побудувати мультивейвлети складніше, ніж вейвлети, оскільки рівняння масштабування включають матричні коефіцієнти, які не комутують між собою. Тому пошук набору коефіцієнтів, що генерує рішення оберненого рівняння, є досить складним. Перший приклад ортогонального безперервного МВ був отриманий Geronimo, Hardin, Massopust [105]. Отримані функції масштабування та вейвлети були кусково-подібні, а побудова базувалася на методах з теорії інтегральних функціональних систем, що генерують фрактальні функції.

Переваги мультивейвлетів над вейвлетами. У дослідженні [106] МВ та мультивейвлет-пакети (МВП) були описані ортогональні та біортогональні МВ. Але мультивейвлет-пакети збільшують обчислювальну складність внаслідок необхідності вибору базису. Для стиснення зображень застосовуються мультивейвлети з алгоритмом SPIHT [107]. Сумалатха і Субраманьям порівняли ефективність різних мультивейвлетів у стисненні медичних зображень та показали, що мультивейвлет SA4 демонструє кращу ефективність порівняно з GHM та CL мультивейвлетами [108]. Зазначається, що мультивейвлети краще виявляють та представляють контури порівняно з вейвлетами. У дослідженні [109] було запропоновано використання мультивейвлета SA4 з алгоритмом SPEC для стиснення зображень, що призвело до покращення PSNR на 3 дБ порівняно з скалярними вейвлетами. У іншому дослідженні [110] мультивейвлети з алгоритмом SPIHT використовувались для стиснення відбитків пальців. Рема та інші [111, 112, 113] застосували мультивейвлети SA4 та алгоритм SPIHT з генетичним алгоритмом для оптимізації коефіцієнтів передфільтра для стиснення відбитків пальців. У іншому дослідженні [114] було запропоновано метод виявлення контурів за допомогою мультивейвлетів та алгоритму. У [21, 115] введено

новий клас мультивейвлетів на основі ступінчастих функцій (СФ) і на їх основі побудовано три ортонормовані базисні системи ступінчастих мультивейвлетів (СМВ), які описані в [23].

5.1 Дискретне мультивейвлет-перетворення

Для функції $f(t)$, яка представлена послідовністю чисел, дискретне мультивейвлет-перетворення (ДМВП) визначається парою перетворень [23]

$$W\ (j_0, i_0) = \frac{1}{\sqrt{N_1}} \sum_{t=0}^{N_1-1} f(t)\ \ {}_{(j_0,i_0)}(t), j \geq j_0, \tag{43}$$

$$W\ {}_{i}^{(k)}(j,i) = \frac{1}{\sqrt{N_1}} \sum_{t=0}^{N_1-1} f(t)\ \ {}_{j,i}^{(k)}(t), i = \overline{1, N_1 - 1}, \tag{44}$$

де N_1 - число мультивейвлет-функцій рангу κ, які представляють мультивейвлет пакет розміру $N_1 \times N_1$, $N_1 = 2^p$, $p \geq 2$, ${}_{j,i}^{(k)}(t)$- i-та функція ФСМВ рангу k, $k = 0,1,...p-2$ для j-го рівня розкладу, $j = 1,2,...m-1$; ${}_{j_0,i_0}(t)$- масштабна функція Хаара $j_0 = 0$ та $i_0 = 0$, ${}_{0,0}(t) = 1$. При цьому додавання проводиться для значень t, i та j для функцій $f(x), x = \overline{0, 2^{n-1}}$, $m = \left[\frac{n}{p}\right]$.

5.2. Швидке мультивейвлет-перетворення

Швидке мультивейвлет-перетворення (ШМВП) представляє собою ефективний метод реалізації обчислення ДМВП, який використовує взаємозв'язок між коефіцієнтами ДМВП сусідніх рівнів розкладу [23]. Коефіцієнти наближення $W_h(j+1)$ і коефіцієнти деталей $W\ {}_{i}^{(k)}(j+1,i)$ рівня $j+1$ можуть бути обчислені через коефіцієнти наближення $W_h(j,i_0)$ рівня j.

Теорема [23]

$$W\ \left(j+1,i_0\right)=\sum_n\ {}_{i_0}\left(n\right)W\ \left(j,n\right),\ n=0,1,\ldots N_1-1 \tag{45}$$

$$W_{\ {}_i^{(k)}}\left(j+1,i\right)=\sum_n\ {}_i^{(k)}\left(n\right)W\ \left(j,n\right),\ i=1,\ldots N_1-1,\ k=0,1,\ldots p-2 \tag{46}$$

Теорема доводиться по аналогії до доведення теореми Малла [116].

Вирази (45), (46) представляють алгоритм швидкого мультивейвлет-перетворення (ША МВП), який може бути обчислений із допомогою тільки операцій скалярного добутку без операцій згортки (еквівалентно фільтрації) і проріджуваної виборки з фактором 2, що потребує відомий алгоритм Малла швидкого вейвлет-перетворення (ШВП) [116, 117].

На рис. 11 (а) представлена блок-схема трьохрівневого швидкого мультивейвлет-перетворення з мультивейвлет-пакетами розміру 4×4 [4]. Простір V_J (функція $f\left(n\right)$) може бути виражена у вигляді

$$V_J=V_{J+3}\oplus W_{J+3,3}\oplus W_{J+3,2}\oplus W_{J+3,1}\oplus W_{J+2,3}\oplus W_{J+2,2}\oplus W_{J+2,1}\oplus W_{J+1,3}\oplus W_{J+1,2}\oplus W_{J+1,1},$$

що представляє двохрівневе дерево з 10 різними розкладами. В результаті ми отримаємо дерево підпросторів аналізу (рис. 11 (б)) і дерево коефіцієнтів (рис. 11 (в)) для трьохрівневого ШМВП блока аналізу на рис. 11(а).

При цьому відоме класичне трьохмасштабне швидке вейвлет-перетворення припускає наявність трьох можливих розкладів, дерево аналізів вейвлет-пакету приводить до 26 різних розкладів. В загальному випадку P-масштабні перетворення на основі класичних вейвлет-пакетів (і відповідне їм дерево аналізу, яке складається з P+1 рівня) дають можливість отримати різні розклади в кількості

$$D\left(P+1\right)=\left[D\left(P\right)\right]^2+1$$

де $D\left(1\right)=1.$

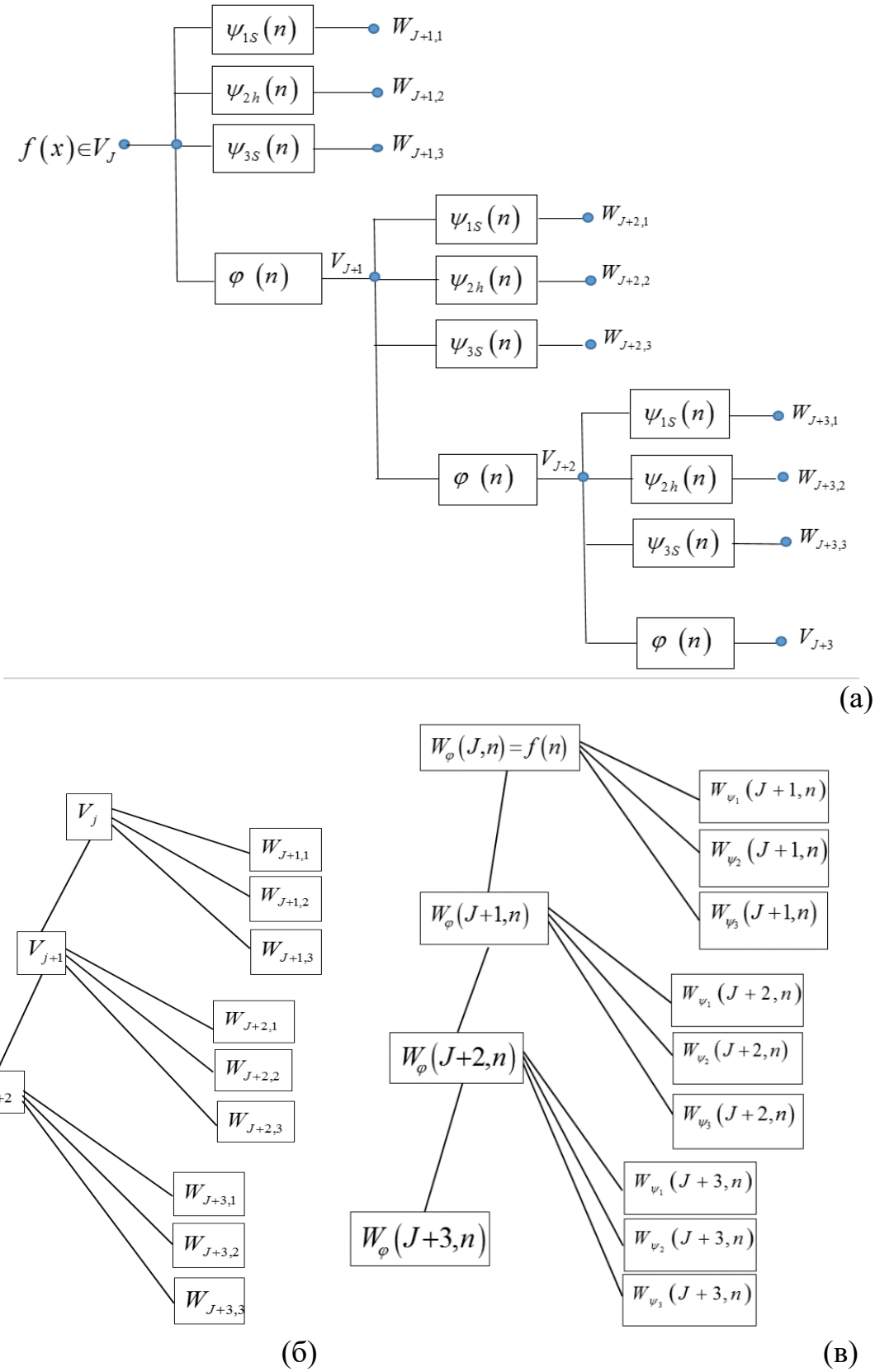

Рисунок 11 Блок-схема трьохрівневого ШМВП з одним мультивейвлет-пакетом розміру 4×4, (а) – блок-схема, (б) – дерево підпросторів аналізу і (в) – дерево коефіцієнтів

При такому великому числі допустимих розкладів перетворення на основі застосування пакетів дозволяють краще контролювати процес розділення спектра, що підлягає розкладу функції на частини.

5.3 Дискретне мультивейвлет-перетворення на основі мультивейвлет-пакета розміру 8 8

Розглянемо матрицю SWT_8^* ДМВП, яка представляє МВП розміру 8×8 з переставленими рядками на основі ДІП [21]

$$SWT_8^* = P_8 SWT_8, \tag{47}$$

де P_8 - матриця 8×8 ДІП, $P_8(0,7) = (0,4,2,6,1,5,3,7)$;

Матрицю SWT_8^* розміру 8×8 ДМВП з переставленими рядками можна записати через матрицю МВП

$$SWT_8^* = B_8 S_8^* \tag{48}$$

де S_8^* - матриця 8×8 МВП з переставленими рядками, B_8 - діагональна матриця 8×8 коефіцієнтів нормування.

Матрицю S_8^* можна побудувати на основі рекурентного методу [115]:

$$S_8^* = \tilde{B}_8 R_8 H_8 diag\left[S_4^*, S_4^*\right] \tag{49}$$

де H_8 - фактор матриця 8×8 із ненульовими елементами ± 1,

$$H_8 = diag\left[I_4, -1, I_3\right] + antidiag\left[I_4, I_2 \otimes I_2^0\right], \tag{50}$$

$$I_4^0 = diag\left[I_2^0, -I_2^0\right],\ I_2^0 = diag[1,0],\ I_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

I_4^0 - діагональна матриця 4×4, яка містить матриці $\pm I_2^0$ розміру 2×2, I_2, I_3, I_4 - одиничні матриці розміру 2×2, 3×3 і 4×4, $\otimes$ - знак кронекерівського множення двох матриць, $\tilde{B}_8$ - діагональна матриця з елементами 1 і $\sqrt{2}$, $\tilde{B}_8 = diag\left[B_2\right]$, $B_2 = diag\left[1, \sqrt{2}\right]$, R_8 - діагональна матриця 8×8, яка містить одиничну матрицю I_4 розміру 4×4, матрицю $R_3^{(1)}$ розміру 3×3 оператора «повороту-стиснення/розтягу» [4] першого рангу $\left(k=1\right)$, і одиницю,

$$R_8 = diag\left[I_4, R_3^{(1)}, 1\right] \tag{51}$$

Матриця $R_3^{(1)}$ містить ненульові елементи, одиницю і константи $r_1^{(1)}$, $r_2^{(1)}$, $s_1^{(1)}$ і $s_2^{(1)}$ першого рангу $\left(k=1\right)$, які задовольняють умові оператора $R_3^{(1)}$:

$$r_1^{(1)} + s_1^{(1)} = r_2^{(1)} - s_2^{(1)} = 1.\ R_3^{(1)} = \begin{bmatrix} r_1^{(1)} & & s_1^{(1)} \\ & 1 & \\ -s_2^{(1)} & & r_2^{(1)} \end{bmatrix} \tag{52}$$

Матриця S_8 МВП розміру 8×8 має вигляд [23]:

$$S_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & s_1 & s_2 & s_3 & -s_3 & -s_2 & -s_1 & -1 \\ 1 & s_4 & -s_4 & -1 & -1 & -s_4 & s_4 & 1 \\ 1 & q_1 & -q_2 & -q_3 & q_3 & q_2 & -q_1 & -1 \\ \sqrt{2}(1 & -1 & -1 & 1 & & & &) \\ \sqrt{2}(& & & & 1 & -1 & -1 & 1) \\ \sqrt{2}(1 & -q_4 & q_4 & -1 & & & &) \\ \sqrt{2}(& & & & 1 & -q_4 & q_4 & -1) \end{bmatrix} \quad (53)$$

Для прикладу, елементи s_i, q_i, $i=\overline{1,4}$, матриці S_8 для функцій ${}^{(1)}_{1}(t)$, ${}^{(1)}_{3}(t)$, ${}^{(0)}_{2}(t)$, ${}^{(0)}_{3(1,j)}(2t-j)$, $j=0,1$ з нелінійними ФСФ [14,15,19] набувають таких значень: $s_1=7/8$, $s_2=3/8$, $s_3=1/4$, $s_4=2/3$, $q_1=7/17$, $q_2=33/17$, $q_3=43/17$, $q_4=3/2$. При цьому константи матриці $R_3^{(1)}$ приймають такі значення: $r_{1,H}^{(1)}=5/8$, $r_{2,H}^{(1)}=30/17$, $s_{1,H}^{(1)}=3/8$ і $s_{2,H}^{(1)}=13/17$.

5.4 Алгоритми швидкого обчислення 8-точкових дискретних мультивейвлет-перетворень

Алгоритм швидкого обчислення 8-точкового прямого ДМВП. Матрицю SWT_8^* згідно (48)-(52) та з урахуванням швидкого алгоритму 4-точкового ДМВП можна представити як добуток п’яти фактор-матриць[21]:

$$SWT_8^* = B_8 S_{8,5} S_{8,4} S_{8,3} S_{8,2} S_{8,1} \quad (54)$$

де $S_{8,k}$ - k-і, $k=\overline{1,5}$, фактор-матриці 8×8 запропонованого в [21] алгоритму швидкого обчислення 8-точкового ДМВП, B_8 - діагональна матриця 8×8 коефіцієнтів нормування:

$$S_{8,1} = diag\left[S_{4,1}, S_{4,1}\right] = I_4 \otimes H_2,\ S_{8,2} = diag\left[H_4, H_4\right],$$

$$S_{8,3} = diag\left[S_{4,3}, S_{4,3}\right] = diag\left[I_2, R_2^{(0)}, I_2, R_2^{(0)}\right], \qquad (55)$$

$$S_{8,4} = H_8,\ S_{8,5} = R_8 = diag\left[I_4, R_3^{(1)}, 1\right],$$

$$B_8 = 2^{-3/2} diag\left[1, \sqrt{2}, b_2, b_4, b_1, \sqrt{2}, b_3, b_4\right].$$

$$b_1 = 8\sqrt{2/63},\ b_2 = 3\sqrt{2/13},\ b_3 = 17/\sqrt{819}, b_4 = 4/\sqrt{13}$$

Алгоритм швидкого обчислення 8-точкового оберненого ДМВП. Матрицю SWT_8^{-1} оберненого ДМВП порядку 8 можна отримати транспонуванням:

$$SWT_8^{-1} = SWT_8^{*T} \qquad (56)$$

Матрицю SWT_8^{-1} на основі (49)÷(51) із урахуванням симетричності матриці $\left(H_2^T = H_2\right)$ можна представити як добуток п’яти транспонованих фактор-матриць [21]:

$$SWT_8^{-1} = S_{8,1} S_{8,2}^T S_{8,3}^T S_{8,4}^T S_{8,5}^T B_8, \qquad (57)$$

де $S_{8,k}^T$ - k-і, $k = \overline{2,5}$, транспоновані фактор-матриці 8×8 запропонованого алгоритму в [21] швидкого обчислення 8-точкового оберненого ДМВП:

$$S_{8,2}^T = diag\left[H_4^T, H_4^T\right],\ S_{8,3}^T = diag\left[I_2, R_2^{(0)T}, I_2, R_2^{(0)T}\right], \qquad (58)$$

$$S_{8,4}^T = H_8^T,\ S_{8,5}^T = R_8^T = diag\left[I_4, R_3^{(1)T}, 1\right],$$

$H_8^T = diag\left[I_4, -1, I_3\right] + antidiag\left[I_2 \otimes I_2^{(0)}, I_4^{(0)}\right]$. При цьому

$$R_3^{(1)T} = \begin{bmatrix} r_1^{(1)} & & -s_2^{(1)} \\ & 1 & \\ s_1^{(1)} & & r_2^{(1)} \end{bmatrix},\ H_8^T = \begin{bmatrix} 1 & & & & 1 & & & \\ & 1 & & & & 0 & & \\ & & 1 & & & & 1 & \\ & & & 1 & & & & 0 \\ 1 & & & & -1 & & & \\ & 0 & & & & 1 & & \\ & & -1 & & & & 1 & \\ & & & 0 & & & & 1 \end{bmatrix}.$$

Обчислювальна складність алгоритму швидкого мультивейвлет-перетворення (ШМВП) для трьох рівнів розкладу з мультивейвлет-пакетом (МВП) розміру 8 8. Обчислювальна складність ША обчислення ДМВП для вхідної послідовності, яка представляє сигнал розмірності N=8 складає: $M_8 = 6$ множень, а $A_8 = 30$ додавань для функцій із лінійними змінами та $A'_8 = 34$ додавання – з нелінійними змінами. Для схеми трьох рівнів розкладу з використанням ША мультивейвлет-перетворення на основі мультивейвлет-пакету розміру 8×8 потрібно $M_{N,8} = M_8 \sum_{i=1}^{3} N / 8^i = \frac{M_8 73N}{512} = \frac{219N}{64}$ множень і $A_{N,8} = A_8 \sum_{i=1}^{3} N / 8^i = \frac{A_8 73N}{512} = \frac{1095N}{256}$ додавань для функцій із лінійними змінами та $A'_{N,8} = A'_8 \sum_{i=1}^{3} N / 8^i = \frac{A'_8 73N}{512} = \frac{1241N}{256}$ додавань для функцій із нелінійними змінами.

Відомий алгоритм Койфмана-Мейєра-Вікергаузера [116, 118] швидкого класичного вейвлет-перетворення для фільтрів з K ненульовими коефіцієнтами вейвлет-пакета усього дерева глибиною $\log_2 N$ потребує $K\ N\ \log_2 N$ множень і додавань, що при K=8 складає $8N\ \log_2 N$ операцій.

Запропонований в [23] ША обчислення мультивейвлет-перетворення порівняно з відомим алгоритмом Койфмана-Мейєра-Вікергаузера для фільтрів з

8-ми ненульовими коефіцієнтами потребує в $K_M = \frac{8N\log_2 N}{219N/256} = \frac{2048\log_2 N}{219}$ разів менше множень, що для N=2^{10} становить 93,5 рази і в $K_A = \frac{8N\log_2 N}{1095N/256} = \frac{2048\log_2 N}{1095}$ разів менше додавань, що становить 18,7 разів із лінійними змінами. Для функцій із нелінійними змінами $K_{A'} = \frac{8N\log_2 N}{1241N/256} = \frac{2048\log_2 N}{1241}$ разів менше додавань, що складає 16,5 разів менше.

5.5 Двовимірне дискретне мультивейвлет-перетворення

Визначимо ДМВП функції $f(x,y)$ розмірів $M \times N$ наступним чином [24]:

$$W\left(j_0,m,n\right) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f\left(x,y\right) \ {}_{j_0}\left(x,y\right), \tag{59}$$

$$W^{i'}_{i}\left(j,m,n\right) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f\left(x,y\right) \ {}^{i'}_{i,j,m,n}\left(x,y\right),\ i' = \{H,V,D\},\ i = \overline{1,N_1-1} \tag{60}$$

Як в одновимірному випадку, j_0 - початковий рівень розкладу, і коефіцієнти $W\left(j_0,m,n\right)$ визначають наближення функції $f(x,y)$ при рівні j_0. Коєфіцієнти $W^{i'}_{i}\left(j,m,n\right)$ визначають горизонтальні, вертикальні і діагональні деталі для рівнів $j \geq j_0$. Звичайно вважаємо j_0=1 і вибираємо числа N і M так, щоб вони були степенем двійки. $N = M = 2^{J'},\ J' \geq 2,\ m,n = 0,1,2,\ldots 2^{j'}-1$. При цьому $J = 0,1,2,\ldots r,\ r = [j'/p],\ N_1 = 2^p,\ p \geq 2.$

Вхідна функція $f(x,y)$ може бути відновлена за заданими коефіцієнтами W і $W^{i'}_{\psi_i}$ в (59) і (60) з допомогою оберненого ДМВП [24]:

$$f(x,y)=\frac{1}{\sqrt{MN}}\sum_{m}\sum_{n}W\ (j_0,m,n)\ \ {}_{j_0,m,n}(x,y)+$$

$$+\frac{1}{\sqrt{MN}}\sum_{i'=H,V,D}\sum_{i=1}^{n_1-1}\sum_{i=1}\sum_{j=1}^{r}\sum_{m}\sum_{n}W^{i'}{}_{i}(j,m,n)\ \ {}_{i,j,m,n}(x,y) \qquad (61)$$

Як і одновимірне ДМВП, двовимірне ДМВП може бути реалізовано із допомогою тільки операцій скалярного добутку без операцій згортки (еквівалентно фільтрації) і проріджуваної виборки з фактором 2, що потребує відомий класичний метод двовимірного швидкого вейвлет-перетворення [20]. Оскільки використовувані масштабна і мультивейвлет-функції є роздільними, то спочатку обчислюється одновимірне швидке МВП (ШМВП) по рядкам від функції $f(x,y)$, а потім обчислюється одновимірне ШМВП по стовпцям від отриманого результату. На рис. 12 представлена блок-схема однорівневого двовимірного ШМВП з одним мультивейвлет-пакетом розміру 4×4 [4]. Зауважимо, що як у випадку свого одновимірного аналога на рис. 11 для отримання коефіцієнтів наближення і деталей для рівня розкладу $j+1$ двовимірне ШМВП оперує з коефіцієнтами наближення рівня розкладу j. У двовимірному випадку, однак, ми маємо діло з трьом наборами коефіцієнтів деталей – горизонтальним, вертикальним і діагональним.

Однорівневий блок мультивейвлет-пакетів може застосовуватись повторно (для чого коефіцієнти наближення на виході цього блока мультивейвлетів потрібно подати на вхід такого ж наступного блоку мультивейвлетів), результатом чого буде p-рівневе перетворення $j=j+1,\ j+2,...,\ j+p$. Як і в одновимірному випадку, зображення $f(x,y)$ використовується в якості коефіцієнтів $W(j,m,n)$ на вході. Множенням стовпців зображення на послідовності (n) і ${}_{k}(n)$, $k=1,2,3$, ми отримаємо чотири частини зображення із зменшенням вчетверо роздільної здатності по горизонталі. Високочастотні або детальні частини характеризують високочастотні у вертикальному напрямі складові зображення. Низькочастотна

або наближення містить інформацію про низькі у вертикальному напрямі частоти. До чотирьох частин зображення далі застосовується аналогічна процедура по рядкам. Це дає на виході шістнадцять зображень [23] (16 частин вихідного зображення), які можуть бути представлені чотирма групами зображень, по одному, по три і по шість зображень в группі W, $\left(W_{1}^{H}, W_{2}^{H}, W_{3}^{H}\right)$, $\left(W_{1}^{V}, W_{2}^{V}, W_{3}^{V}\right)$, $\left(W_{1,2}^{VH}, W_{1,3}^{VH}, W_{2,1}^{VH}, W_{2,3}^{VH}, W_{3,1}^{VH}, W_{3,2}^{VH}\right)$ і $\left(W_{1}^{D}, W_{2}^{D}, W_{3}^{D}\right)$.

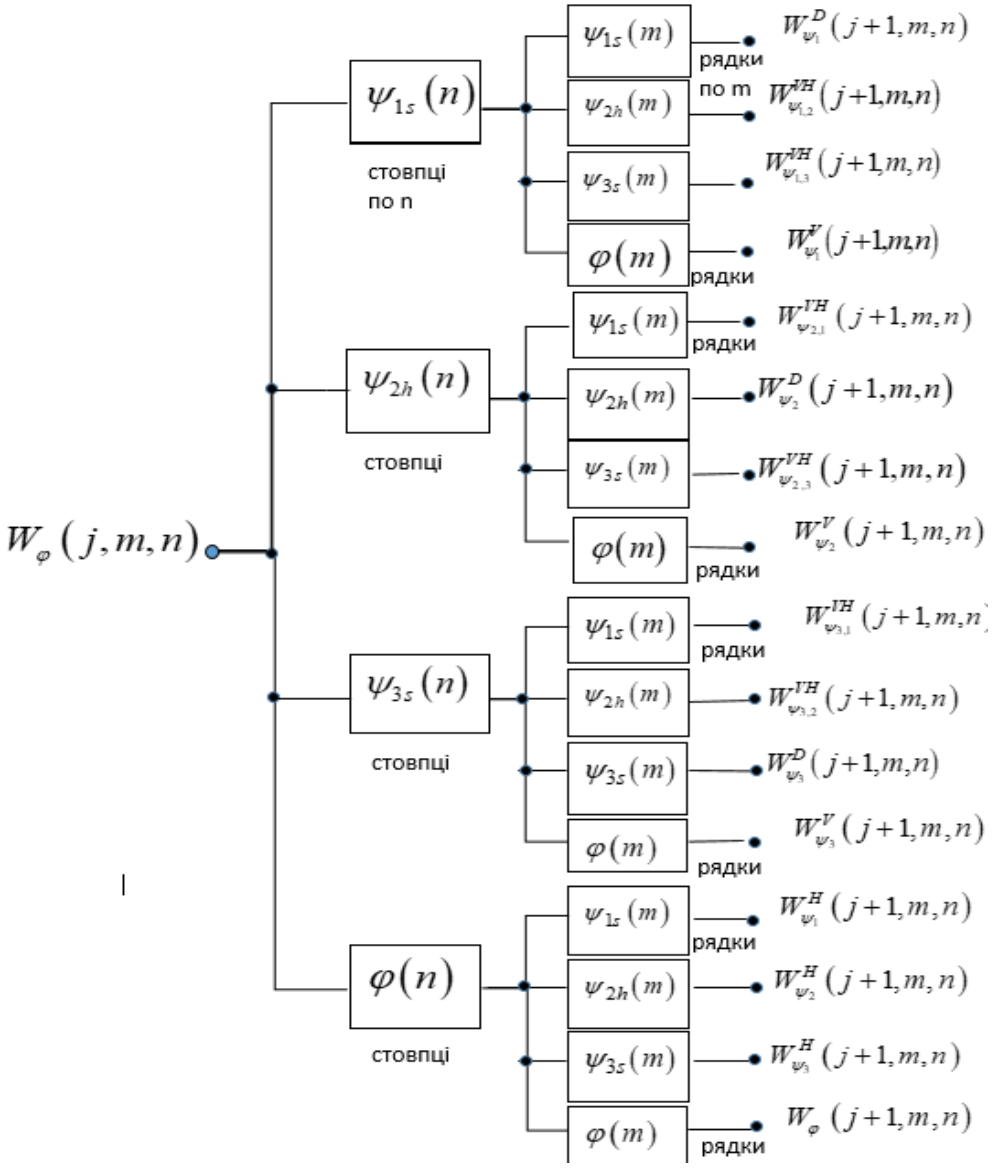

Рисунок 12 Однорівневе двовимірне ШМВП з мультивейвлет-пакетом 4×4

На рис. 13 показано результат скалярного добутку зображення $f(x,y)$ і двовимірної масштабуючої функції і мультивейвлет-функції для одного рівня розкладу [23].

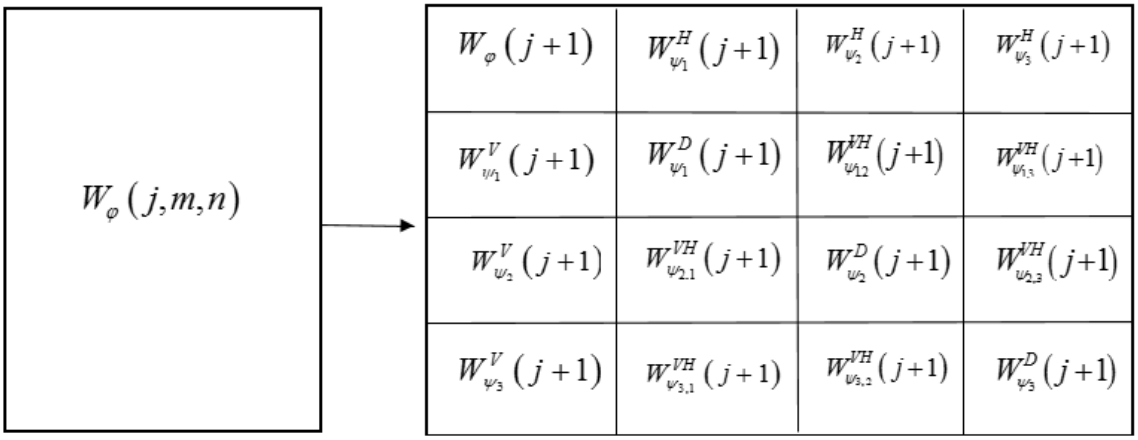

Рисунок 13 Двовимірне ШМВП для одного рівня розкладу

На рис. 14 представлено відповідне однорівневе шістнадцятеторічне дерево аналізу ШМВП (для одного рівня розкладу) [4].

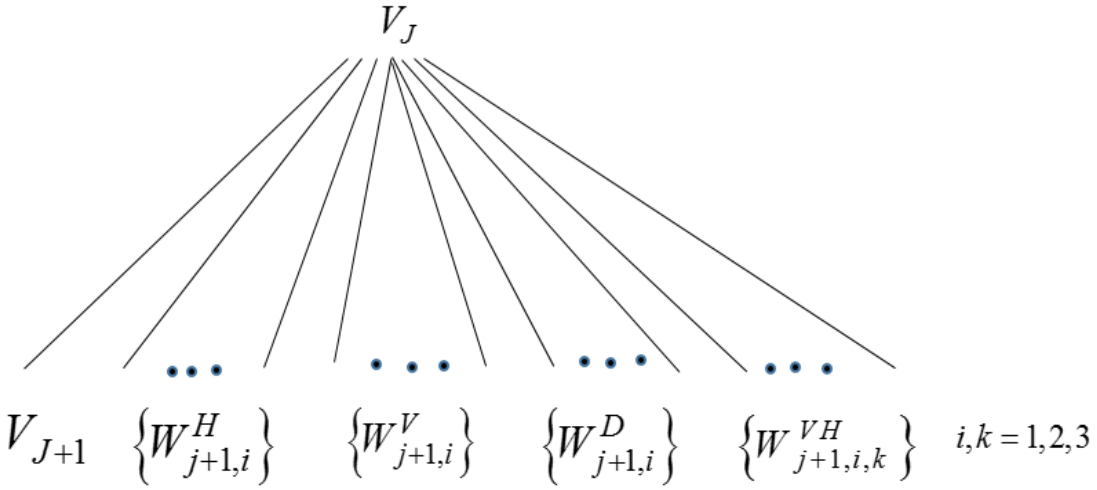

Рисунок 14 Дерево підпросторів аналізу для одного рівня розкладу

Відмітимо, що частотна площина виявляється розділеною на п'ять складових частин різної площі. Низькочастотна частина діапазону у центрі відповідає коефіцієнтам перетворення $W(j+1,m,n)$ і масштабному простору V_{j+1}. Це повністю узгоджується з одновимірним випадком. У двовимірному

випадку ми маємо чотири (замість трьох) підпростори мультивейвлетів. Вони позначаються $\left\{W_{j+1,i}^{H}\right\}$, $\left\{W_{j+1,i}^{V}\right\}$, $\left\{W_{j+1,i}^{D}\right\}$, $\left\{W_{j+1,i,j}^{VH}\right\}$ і відповідають коефіцієнтам $\left\{W_{i}^{H}(j+1,m,n)\right\}$, $\left\{W_{i}^{V}(j+1,m,n)\right\}$, $\left\{W_{i}^{D}(j+1,m,n)\right\}$ і $\left\{W_{i,j}^{VH}(j+1,m,n)\right\}$.

Дерево аналізу для двовимірних P-масштабних класичних відомих вейвлет-пакетів дає можливість побудувати різні розклади в кількості [5]

$$D(P+1)=\left[D(P)\right]^{4}+1,$$

де $D(1)=1$. Таким чином, повне число різних розкладів, які можна отримати із трьохмасштабного дерева, рівне 21522. При такому великому числі розкладів двовимірні перетворення на основі застосування пакетів дозволяють краще контролювати процесс розділення двовимірного спектра, що підлягає розкладу зображення на частини. Однако це приводить до суттєвого збільшення обчислювальної складності.

5.6 Мультивейвлет-метод кодування зображень на основі двовимірного (2D) ШМВП з мультивейвлет-пакетом розміру 8 8 для трьох рівнів розкладу

В роботі [21] запропонована нова мультивейвлет-технологія і метод кодування зображень на основі 2D ДМВП в базисі фрактальних ступінчастих з використанням швидких алгоритмів. Розроблено метод кодування зображень на основі двохрівневого розкладу з використання 2D-швидкого ДМВП з мультивейвлет-пакетами для першого рівня розміру 4×4 і для другого рівня – розміру 32×32. Як показали експериментальні результати, запропонований метод мультивейвлет-кодування в порівнянні з відомим блочним методом на основі цілочислового косинусного перетворення (ЦКП) для блоку 32×32, яке застосовується в стандарті з відеокодування H.265 за характеристикою

кількісної оцінки спотворень PSNR (дБ) [6] для семи тестових зображень класів A, B, C з роздільною здатністю 2560x1536, 2048x1280, 1280x768 знижує середнє значення на 0,43-1,06 дБ при середніх значеннях коефіцієнта стиснення від 4 до 58, а при високих – середнє значення зниження PSNR становить 0,62 дБ. При цьому візуально забезпечується краща якість, ніж H.265, оскільки відсутні блокові спотворення, які підсилюються при високих ступенях стиснення для H.265. Обчислювальна складність запропонованого в [21] методу у порівнянні з відомим [117, 118] класичним вейвлет-методом з K ненульовими коефіцієнтами вейвлет-пакета усього дерева глибиною $\log_2 N$ для зображення $N \times N$ 2D ШВП (алгоритм Койфмана-Мейєра-Вікергаузера) за операціями множення зменшується у 80 раз, а у порівнянні з методом блочного кодування на основі відомого алгоритму швидкого ЦКП [10] у 3 рази, а на основі ЦКП (H.265) – в 11 разів.

Розглянемо мультивейвлет-метод кодування зображень на основі трьохрівневого двовимірного ШМВП з мультивейвлет-пакетом розміру 8×8.

Обґрунтування рівня розкладу. Важливим фактором, який впливає на обчислювальну складність і рівень похибок відновлення мультвейвлет-кодування, є число рівнів розкладу перетворення. Оскільки p-рівневе швидке мультивейвлет-перетворення потребує p ітерацій перетворення, число операцій при обчисленні прямого та оберненого збільшується із збільшенням числа рівнів розкладу. Проте, квантування коефіцієнтів вищого рівня розкладу вилпває на всю велику область відновленного зображення. В багатьох застосуваннях, таких як пошук по базі даних зображень або передача зображень для поступового відновлення (прогресивна передача), число рівнів перетворення визначається роздільною здатністю збережених або переданих зображень, а також масштабом найменшої з використаних копій. Відмітимо, що основне стиснення відбувається при початкових розкладах. Як показано в [5], при збільшенні числа рівнів розкладу більше трьох, мало змінюється число коєфіцієнтів, що обнуляються.

Метод кодування зображень на основі 2D швидких мультивейвлет-перетворень з мультивейвлет-пакетом розміру 8×8 для трьох рівнів [23].

Для схеми трьох рівнів розкладу при кодуванні зображення розміру $N\times N$ на основі 8-точкового 2D ШМВП з мультивейвлет-пакетом 8×8 потрібно $M_{N,8}=M_8\sum_{i=1}^{3}2N^2/8^{2i-1}=\frac{M_8 4161N^2}{16384}$ множень, що при M_8 = 6 складає $\frac{12483N^2}{8192}$ множень, або на один піксел потрібно $M_{8/p}=1{,}52$ множення на піксел, і $A_{N,8}=A_8\sum_{i=1}^{3}2N^2/8^{2i-1}=\frac{A_8 4161N^2}{16384}$ додавань, що при A_8 = 30 становить $\frac{15*4161N^2}{8192}$ додавань, що на один піксел потрібно $A_{8/p}=7{,}62$ додавань на піксел для функцій з лінійними змінами. Для функцій з нелінійними змінами - A'_8 = 34 потрібно $A'_{N,8}==\frac{17*4161N^2}{8192}$ додавань, що складає 8,63 додавань на піксел.

Відомий алгоритм Койфмана-Мейєра-Вікергаузера [118] 2D ШВП для фільтрів з K ненульовими коефіцієнтами вейвлет-пакета усього дерева глибиною $\log_2 N$ для зображення $N\times N$ потребує $2K$ $N^2\log_2 N$ множень і додавань, що при K = 8 складає $16N^2\times\times\log_2 N$ операцій, або 16 $\log_2 N$ операцій на один піксел.

Метод кодування зображень на основі трьохрівневого 8-точкового 2D ШМВП порівняно з відомим алгоритмом Койфмана-Мейєра-Вікергаузера 2D ШВП для фільтрів з 8-ми ненульовими коефіцієнтами потребує в $K_M=\frac{16\log_2 N}{1{,}51}=10{,}52\log_2 N$ разів менше множень, що для N=2^{10} становить 105,2 разів і в $K_A=\frac{16\log_2 N}{7{,}62}=2{,}1\log_2 N$ рази менше додавань, що у 21 раз менше для функцій з лінійними змінами. Для функцій із нелінійними змінами

$K_{A'} = \frac{16\log_2 N}{8,63} = 1,85\log_2 N$ разів менше додавань, що становить 18,5 разів менше.

Обчислювальна 2D складність методу ШМВП представлено у табл. 23, для порівняння з блочним кодуванням було обране цілочислове косинусне перетворення порядку 16, оскільки воно має проміжну складність між перетвореннями порядку 8 і 32.

Оцінки обчислювальної складності мультивейвлет-методу кодування на основі трьохрівневого 2D ШМВП з лінійними змінами, відомого класичного вейвлет-методу на основі алгоритму Койфмана-Мейєра-Вікергаузера [118] 2D ШВП з вейвлет-пакетами для фільтрів з К=8 ненульовими коефіцієнтами вейвлет-пакета усього дерева глибиною $\log_2 N$ (N=1024), блочного методу кодування на основі відомого 16-точкового алгоритму оберненого 2D ЦКП для блоку 16×16, яке застосовується в стандартах з відеокодування H.265/HEVC [10] і VVC, наведено у табл. 23.

Таблиця 23

Операції на піксел	Оцінка обчислювальної складності 2D перетворень			Порівняльний аналіз методу ШМВП відносно	
	метод ШМВП для трьох рівнів розкладу з МВП 8×8	Метод ШВП з вейвлет-пакетами [118], К=8,	Блочний метод з відомим ЦКП для блоків 16×16 H.265/HEVC [10]	Метод ШВП [118]	ЦКП (H.265) [10]
Множення	1,52	160	10,75	У 105 разів менше	У 7,02 разів менше
Додавання + зсув	7,62	160	14,5	У 21 раз менше	У 1,9 рази менше

Експериментальні результати і аналіз.

Для тестування було взято чотири тестові зображення (рис. 15). Результати ефективності кодування ШМВП відносно блочного косинусного перетворення порядку 16 за характеристикою PSNR (дБ.) наведено в табл. 24.

Таблиця 24

Клас	Зображення	Результати ефективності кодування за характеристикою PSNR (дБ) для QP				
		22	27	32	37	42
B 2048x1024	Фото (ВК)	-0,287	-0,234	-0,336	-0,381	-0,25
	Пейзаж (НК)	-0,158	-0,049	-0,029	-0,15	-0,292
C 1536x1024	Алея (НК)	-0,31	-0,099	-0,049	-0,173	-0,346
	Храм (СК)	-0,108	0,063	-0,018	-0,252	-0,453

Результати тестування на основі методу ШМВП з МВП 8×8 за коефіцієнтом стиснення К:1 наведено в табл. 25, а для відомого блочного методу на основі ЦКП (H.265) з блоком 16×16 – в табл. 26.

Таблиця 25

Клас	Зображення	Результати кодування за коефіцієнтом стиснення (К:1) для QP				
		22	27	32	37	42
B 2048x1024	Фото (ВК)	5,95	12,79	31,95	75,19	184,50
	Пейзаж (НК)	1,48	1,81	2,45	4,10	8,59
C 1536x1024	Алея (НК)	1,42	1,68	2,17	3,42	6,68
	Храм (СК)	2,99	4,17	6,18	10,52	19,86

Таблиця 26

Клас	Зображення	Результати кодування за коефіцієнтом стиснення (К:1) для QP				
		22	27	32	37	42
B 2048x1024	Фото (ВК)	5,90	13,64	30,40	66,89	123,00
	Пейзаж (НК)	1,49	1,87	2,59	4,46	9,28
C 1536x1024	Алея (НК)	1,49	1,73	2,33	3,74	7,26
	Храм (СК)	2,97	4,37	6,80	11,93	21,97

Результати ефективності кодування на основі методу ШМВП з МВП 8×8 за коефіцієнтом стиснення К:1 (у відсотках) наведено в табл. 27

Таблиця 27

Клас	Зображення	Результати ефективності кодування за коефіцієнтом стиснення (у відсотках) для QP				
		22	27	32	37	42
B 2048x1024	Фото (ВК)	0,85	-6,23	5,10	12,41	50,00
	Пейзаж (НК)	-0,67	-3,21	-5,41	-8,07	-7,44
C 1536x1024	Алея (НК)	-4,70	-2,89	-6,87	-8,56	-7,99
	Храм (СК)	0,67	-4,58	-9,12	-11,82	-9,60

а)

б)

в)

г)

Рисунок 15 Тестові зображення

5.7 Висновки до розділу.

Розроблена комп’ютерна технологія вейвлет-методу кодування зображень на основі 2D швидких мультивейвлет-перетворень в базисі фрактальних ступінчастих функцій з мультивейвлет-пакетом розміру 8×8 для трьох рівнів розкладу. Мультивейвлет метод кодування на основі ШМВП має низьку мультиплікативну складність, яка в 105 разів менша порівняно з відомим класичним вейвлет-методом.

ВИСНОВКИ

Розглянуто архітектуру гібридного кодека з використанням нейромереж на стороні декодера, що має пріорітетом інформативність та мінімальність закодованих традиційними методами спеціалізованих даних, що використовує найбільш узагальнені вхідні дані та застосовує генеративні нейромережі для відновлення даних та підвищення візуальної якості. В рамках цього підходу запропоновано ряд нових методів та алгоритмів кодування для зменшення обчислювальної складності на стороні кодера. Запропоновано новий спрощений режим роздільного кодування з використанням найбільш поширеної та ефективної комбінації – косинусного і синусного перетворень. Запропоновано нову міру для оцінки похибок в просторі кольорів на основі обчислення похибки пропорцій вертикальних та горизонтальних різниць між сусідніми пікселами. Розроблена концепція побудови мультигібридних ортогональних перетворень, яка дозволяє будувати нові перетворення на базі відомих двох класів. Запропоновано два методи побудови мультигібридних перетворень – на основі комбінації відомого і гібридного перетворення і на основі комбінації двох гібридних перетворень. На основі цих методів розроблено три нових цілочислових перетворення різних класів. Розроблено матричний метод побудови гібридних цілочислових спрощених синус–косинусних перетворень типу VII порядку 16 на основі двох перетворень порядку 8: синусного типу VII і косинусного типу II. Розроблено двоє гібридних цілочислових однонормових спрощених синус-косинусних перетворень низької обчислювальної складності, яка менше на 38,6% порівняно з відомим алгоритмом дискретного синусного перетворення типу VII порядку 16 з цілочисловою апроксимацією, що застосований у сучасному стандарті відеокодування VVC. Розроблено швидкий алгоритм обчислення оберненого цілочислового косинус-синусного перетворення типу IV порядку 16 низької мультиплікативної складності, Розроблено метод побудови гібридного синусно-косинусного перетворення типу II порядку 4, і запропоновано три нових

цілочислових перетворення зі швидкими алгоритмами обчислення. Розроблено метод побудови спрощеного синусно-косинусного перетворення типу II порядку 8, запропоновано два нових цілочислових перетворення і розроблено для них швидкі алгоритми низької обчислювальної складності без виконання операції множення. Запропоновано метод побудови цілочислових модифікованих косинусних і синусних перетворень типу II порядку 8, на основі якого розроблено цілочислові однонормові модифіковані синусні та косинусні перетворення типу II порядку 8 зі швидкими алгоритмами обчислення. Розроблені спрощені синусно-косинусні та модифіковані косинусні перетворення можуть бути альтернативою відомому синусному перетворенню типу VII для адаптивного застосування в збалансованих і швидкодійних режимах кодування, в тому числі для застосування в запропонованому гібридному кодеку. Розглянуто розширення сімейства відомих і мультигібридних перетворень шляхом використання комбінаторного перебору різноманітних пропорцій параметрів перетворень. В рамках цього підходу представляють цікавість ортогональні перетворення, що реалізуються розрідженими матрицями і тому мають меншу обчислювальну складність.

Розроблено комп’ютерну технологію кодування зображень на основі 2D швидких мультивейвлет-перетворень в базисі фрактальних ступінчастих функцій з мультивейвлет-пакетом розміру 8×8 для трьох рівнів розкладу. Мультивейвлет метод кодування на основі швидких мультивейвлет перетворень має низьку мультиплікативну складність, яка в 105 разів менша порівняно з відомим класичним вейвлет-методом для фільтрів з K=8 ненульовими коефіцієнтами.

Застосування розроблених підходів, методів і алгоритмів має значення для розробки кодеків низької обчислювальної складності, як традиційних, так і гібридних з використанням нейромереж на стороні декодера. Це важливо для кодування в області інтернету речей, мобільного зв’язку, носимих пристроїв, супутникових та космічних технологій, а також збалансованого кодека для збереження та передачі зображень та відео в мережі інтернет.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Luts Ya.V., Luts V.K. "About the Development of a High-Speed Simplified Image Codec. Cybernetics and Computer Technologies." 2021. 1. P. 61–66. (in Ukrainian) https://doi.org/10.34229/2707-451X.21.1.6
2. Луц В.К., Луц Я.В. "Про використання модифікованого дельта-алгоритму для обробки та кодування зображень." Фізико-математичне моделювання та інформаційні технології, (36), 148-152. (2023) https://doi.org/10.15407/fmmit2023.36.148
3. Hnativ L.O., Luts Ya.V. "Discrete Cosine-Sine Transform and Fast 16-point Integer Cosine-Sine Transform Low Complexity for Video Coding." Proc. SPIE 13137, Applications of Digital Image Processing XLVII, 1313716 (30 September 2024) Optical Engineering + Applications, 2024, San Diego, California, United States, https://doi.org/10.1117/12.3035173
4. Hnativ L., Luts V. and Luts Ya. "Fractal step multiwavelets and multiwavelet packets - a new multiwavelet technology for image processing and coding." Selected Papers of the III International Scientific Symposium “Intelligent Solutions” (IntSol-2023). Symposium Proceedings, Kyiv - Uzhhorod, Ukraine, September 27-28, 2023. – 2023. – P. 282–294. https://ceur-ws.org/Vol-3538/
5. Gonzalez, Rafael C., and Woods, Richard E. "Digital Image Processing." 3rd Edition, Prentice Hall, 2008. 954 p
6. Salomon, David "Data Compression: The Complete Reference." 4th Edition, Springer, 2007. 1091 p
7. Sayood, K. "Introduction to Data Compression." 3rd Edition, Morgan Kaufmann, Burlington, MA, 2017. 703 p
8. 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
9. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand "Overview of the High Efficiency Video Coding (HEVC) Standard." IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.
10. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)." IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1669-1684, Dec. 2012.
11. F. Bossen, B. Bross, K. Sühring, and D. Flynn "HEVC Complexity and Implementation Analysis." IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012.
12. Versatile Video Coding (VVC) | JVET. URL: https://jvet.hhi.fraunhofer.de/ Дата звернення 18.09.2024

13. N. Ahmed, T. Natarajan and K. R. Rao "Discrete Cosine Transform." IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, Jan. 1974. https://doi.org/10.1109/T-C.1974.223784
14. Pennebaker, W.B. and Mitchell, J.L. "JPEG Still Image Data Compression Standard." Kluwer Academic Publishers, Norwell, 1992. 638 p
15. Daubechies, Ingrid "Ten Lectures on Wavelets." Philadelphia, Pa. : Society for Industrial and Applied Mathematics, 1992. 357 p. https://doi.org/10.1137/1.9781611970104
16. Taubman, D. and Marcellin, M. "JPEG 2000: Image Compression Fundamentals, Standards and Practice." Kluwer Academic Publishers, Norwell, 2001. 798 p
17. Barnsley, M.F. "Fractals Everywhere." Academic Press, San Diego, 1988. 560 p
18. Fisher, Y. "Fractal Image Compression Theory and Application." University of California, Institute for Nonlinear Science, Springer-Verlay, New York, Inc, 1995. 360 p
19. L.O. Hnativ."New integer fractal step multi wavelets and their fast transform for high efficiency image and video coding." Issues Optimization Computing (IOC-XLII): Proceedings Intern. Conf., Zakarpattya region 21 – 25 September 2015.
20. L.O. Hnativ, V.K. Luts "High efficiency image and video coding based on integer wavelet transforms at the basis of the fractal piecewise linear step functions." Issues Optimization Computing (IOC-XLII): Proceedings Intern. Conf., Zakarpattya region 21 – 25 September 2015
21. L.O. Hnativ "Fractal step multiwavelets - a new wavelet technology for signal processing and image encoding." Theses of International Scientific Conference "Modern Informatics: Problems, Achievements and Prospects for Development", dedicated to the 60th anniversary of the V.M. Glushkov Institute of Cybernetics of NAS of Ukraine. December 13-15, 2017. Kyiv, V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, 2017. pp 195-197
22. L.O. Hnativ, Luts V. K. "New image and video encoding technology based on integer fractal stepped multiwavelets." Theses of International Scientific Conference "Modern Informatics: Problems, Achievements and Prospects for Development", dedicated to the 60th anniversary of the V.M. Glushkov Institute of Cybernetics of NAS of Ukraine. December 13-15, 2017. Kyiv, V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, 2017. pp 197-199.
23. Hnativ, L.O. "Ортонормовані базиси фрактальних ступінчастих мультивейвлетів – нова мультивейвлет-технологія для оброблення сигналів і зображень." Фізико-математичне моделювання та інформаційні технології, 2021, (32), 91-95. https://doi.org/10.15407/fmmit2021.32.091

24. Hnativ, L. "Дискретні фрактальні ступінчасті мультивейвлети і мультивейвлет-пакети – нова мультивейвлет-технологія для оброблення та кодування зображень." Фізико-математичне моделювання та інформаційні технології, 2023, (36), 53-57. https://doi.org/10.15407/fmmit2023.36.053
25. G. Toderici et al. "Full resolution image compression with recurrent neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5306-5314
26. Ballé, Johannes & Minnen, David & Singh, Saurabh & Hwang, Sung & Johnston, Nick "Variational image compression with a scale hyperprior." arXiv preprint, 2018. https://doi.org/10.48550/arXiv.1802.01436
27. Kekre H.B., Solanki J.K. "Comparative performance of various trigonometric unitary transforms for transform image coding." Int. J. Electronics, 1978, vol. 44, no. 3, pp. 305-315. https://doi.org/10.1080/00207217808900979
28. Hnativ, L.O., Luts, V.K. "Algorithms for Fast Implementation of 4-Point Integer Sine Type VII Transforms without Multiplication and Separable Directional Adaptive Transforms for Intra Prediction in Image/Video Coding." Cybern Syst Anal 56, 2020, pp. 159–170. https://doi.org/10.1007/s10559-020-00231-y
29. Hnativ, L.O., Luts, V.K. "Integer Modified Sine-Cosine Transforms Type VII. A construction Method and Separable Directional Adaptive Transforms for Intra Prediction with 8 × 8 Chroma Blocks." Image/Video Coding. Cybern Syst Anal 57, 2021, pp. 155–164. https://doi.org/10.1007/s10559-021-00339-9
30. Луц В.К. "Просте цілочисельне косинусне перетворення порядку 16 для кодування сигналів і зображень." Штучний інтелект. 2014. № 4. С. 31–43
31. Hnativ, L.O. "Integer Cosine Transforms: Methods to Construct new Order 8, 16 Fast Transforms and Their Application." Cybern Syst Anal 50, 2014, pp. 913–929. https://doi.org/10.1007/s10559-014-9682-9
32. Hnativ, L.O. "Integer Cosine Transforms for High-Efficiency Image and Video Coding." Cybern Syst Anal 52, 2016, pp. 802–816. https://doi.org/10.1007/s10559-016-9881-7
33. Hnativ, L.O. "A Method to Construct a Large-Size Simple Integer Cosine Transform for High-Resolution Image and Video Coding." Cybern Syst Anal 52, 2016, pp. 956–966. https://doi.org/10.1007/s10559-016-9898-y
34. Hnativ, L.O. "High-Order Integer Cosine Transform: Construction Method and Fast Algorithms for High-Resolution Image and Video Coding." Cybern Syst Anal 53, 2017, pp. 615–626. https://doi.org/10.1007/s10559-017-9964-0
35. Hnativ, L.O. "Simple High-Order Integer Cosine Transform for High-Resolution Video Coding." Cybern Syst Anal 54, 2018, pp. 490–501. https://doi.org/10.1007/s10559-018-0050-z

36. Hnativ, L.O. "Modified Integer Sine Transforms. Construction Method and Separable Directional Adaptive Transforms for Intra Prediction in Image/Video Coding." Cybern Syst Anal 56, 2020, pp. 331–342. https://doi.org/10.1007/s10559-020-00250-9
37. Hnativ, L.O. "Discrete Cosine-Sine Type VII Transform and Fast Integer Transforms for Intra Prediction of Images and Video Coding." Cybern Syst Anal 57, 2021, pp. 827–835. https://doi.org/10.1007/s10559-021-00408-z
38. J. Dong and K.N. Ngan "A Universal Approach to Developing Fast Algorithm for Simplified Order-16 ICT." IEEE International Symposium on Circuits and Systems, New Orleans, U.S.A., 27-30 May, 2007, pp. 281-284. https://doi.org/10.1109/ISCAS.2007.378390
39. Lee B., Kim M., Park C. et al. "A 16x16 transform kernel with quantization for (ultra) high definition video coding." ITU-T VCEG, Doc. VCEG-AK13, Yokohoma, Japan, April 2009.
40. C. K. Fong and W. K. Cham "Simple order-16 Integer Transform for video coding." 2010 IEEE International Conference on Image Processing, Hong Kong, China, 2010, pp. 161-164, https://doi.org/10.1109/ICIP.2010.5651737
41. Gordon S., Marpe D., Wiegand T. "Simplified Use of 8x8 Transforms – Updated Proposal & Results", Doc JVT-KO28, 11th Meeting: Munich, Germany, 15-19 March, 2004, P.1-22.
42. Ma S., Fan X., Gao W. "Low complexity integer transform and High Definition Coding." SPIE 49th Annual Meeting, V. 58. Aug 02-06, 2004, Denver, CO, USA, Bellingham, WA, USA: SPIE Press, 2004. - P. 547-554. https://doi.org/10.1117/12.563808
43. N. Brahimi, T. Bouden, T. Brahimi, and L. Boubchir "Lossy image compression based on efficient multiplier-less 8-points DCT." Multimedia Systems, 2022, vol. 28, no. 1, pp. 171-182.
44. Richardson, Iain E. G. "Video Coding: H.264 and MPEG-4 – Standards of the Next Generation." Academic Press, 2003. 302 p.
45. https://www.lcevc.org/ MPEG-5 part 2 LCEVC (Low Complexity Enhancement Video Coding) Дата звернення 02.11.2024
46. Y. Zhang, R. Tian, J. Liu and N. Wang "Fast rate distortion optimized quantization for HEVC." 2015 Visual Communications and Image Processing (VCIP), Singapore, 2015, pp. 1-4, https://doi.org/10.1109/VCIP.2015.7457854
47. M. Babic, M. A. Farahani, and T. Wuest "Image Based Quality Inspection in Smart Manufacturing Systems: A Literature Review." Procedia CIRP, 2021, vol. 103, pp. 262- 267.
48. R. Matsumoto "Introducing the perception-distortion tradeoff into the rate distortion theory of general information sources." IEICE Communications Express, 2018, vol. 7, no. 11, pp. 427-431.

49. J. Stankowski, C. Korzeniewski, M. Domański and T. Grajek "Rate-distortion optimized quantization in HEVC: Performance limitations." 2015 Picture Coding Symposium (PCS), Cairns, QLD, Australia, 2015, pp. 85-89, https://doi.org/10.1109/PCS.2015.7170052
50. F. Li, S. Krivenko, and V. Lukin "A Two-step Procedure for Image Lossy Compression by ADCTC With a Desired Quality." IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2020, pp. 307-312, https://doi.org/10.1109/DESSERT50317.2020.9125000
51. S. Krivenko, O. Krylova, E. Bataeva, and V. Lukin, "Smart lossy compression of images based on distortion prediction." Telecommunications Radio Engineering, 2018, vol. 77, no. 17.
52. S. Krivenko, F. Li, V. Lukin, B. Vozel, and O. Krylova "Prediction of visual quality metrics in lossy image compression." IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO), 2020, pp. 478-483.
53. F. Li, S. Krivenko, and V. Lukin "A Two-step Approach to Providing a Desired Visual Quality in Image Lossy Compression." IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, pp. 502-506.
54. Bross, Benjamin & Wang, Ye-Kui & Ye, Yan & Liu, Shan & Chen, Jianle & Sullivan, Gary & Ohm, Jens-Rainer "Overview of the Versatile Video Coding (VVC) Standard and its Applications." IEEE Transactions on Circuits and Systems for Video Technology 31. 2021, pp. 3736-3764, https://doi.org/10.1109/TCSVT.2021.3101953
55. ITU-T Recommendation H.266/ISO/IEC 23090-3: 2020. Versatile Video Coding, 2020.
56. Ye Y. and Karczewicz M. "Improved intra coding." ITU-T SG16Q6. Doc. VCEG-AG11, Shenzhen, China, Oct. 2007.
57. X. Zhao et al., "Transform Coding in the VVC Standard." IEEE Transactions on Circuits and Systems for Video Technology, Oct. 2021, vol. 31, no. 10, pp. 3878-3890, https://doi.org/10.1109/TCSVT.2021.3087706
58. JPEG. URL: https://jpeg.org Дата звернення 16.11.2024
59. ISO/IEC 14495-1:1999. Information technology - Lossless and near-lossless compression of continuous-tone still images: Baseline. Switzerland, 1999.
60. ISO/IEC 23008-12:2017 – Information technology – Hight efficiency coding and media delivery in heterogeneous environments – Part 12: Image File Format. URL: https://www.iso.org/standard/66067.html
61. HEIF – High Efficiency Image File Format. URL: https://nokiatech.github.io/heif/ Дата звернення 19.11.2024
62. AV1 Image File Format (AVIF). URL: https://aomediacodec.github.io/av1-avif/ Дата звернення 21.11.2024.

63. FLIF – Free Lossless Image Format. URL: http://flif.info/ Дата звернення 21.11.2024
64. Мюрей Д.Д., Райпер У. "Енциклопедія форматів графічних файлів." BHV-Київ, 1997. — 672 с.
65. Aldus Corporation, "TIFF Revision 6.0." 411 First Avenue South Seattle, WA 98104-2871, 1992, 121 p. URL: https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf Дата звернення 22.11.2024
66. About OpenEXR. URL: https://openexr.com/en/latest/about.html Дата звернення 24.11.2024
67. M. Khani, V. Sivaraman, M. Alizadeh "Efficient Video Compression via Content-Adaptive Super-Resolution." Proc. IEEE ICCV 2021, pp. 4521-4530.
68. S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang and S. Wang "Image and Video Compression With Neural Networks: A Review." IEEE Transactions on Circuits and Systems for Video Technology, June 2020, vol. 30, no. 6, pp. 1683-1698, https://doi.org/10.1109/TCSVT.2019.2910119
69. G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell "Full Resolution Image Compression with Recurrent Neural Networks." CVPR, 2017, pp. 5435–5443.
70. J. Ballé, V. Laparra, and E. P. Simoncelli "End-to-end optimized image compression." arXiv:1611.01704, 2017.
71. J. Ballé, V. Laparra, and E. P. Simoncelli "End-to-end optimization of nonlinear transform codes for perceptual quality." Picture Coding Symposium (PCS), 2016, pp. 1–5.
72. J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston "Variational image compression with a scale hyperprior." International Conference on Learning Representations, 2018.
73. D. Minnen, J. Ballé, and G. D. Toderici "Joint autoregressive and hierarchical priors for learned image compression." Advances in Neural Information Processing Systems 31. Curran Associates, Inc., 2018, pp. 10 771–10 780.
74. G. Lu et al. "DVC: An End-to-End Deep Video Compression Framework." Proc. IEEE CVPR 2019.
75. C.-Y. Wu, N. Singhal, and P. Krahenbuhl "Video compression through image interpolation." ECCV, September 2018.
76. C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krahenbuhl "Compressed video action recognition." CVPR, pages 6026–6035, 2018.
77. Z. Qiu, Z. He, Z. Jin "Accelerating Learned Video Compression via Low-Resolution Representation Learning." arXiv:2407.16418, 2024.
78. S. Jamil, M. Jalil Piran, M. Rahman "Learning-Driven Lossy Image Compression: A Comprehensive Survey." arXiv:2201.09240, 2022.

79. Learned Image Compression: Introduction - Mateen Ulhaq. URL: https://yodaembedding.github.io/post/learned-image-compression/ Дата звернення 25.11.2024
80. Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai "Residual Highway Convolutional Neural Networks for in-loop Filtering in HEVC." IEEE Trans. on Image Processing, 2018.
81. C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma "Spatial-temporal residue network based in-loop filter for video coding." Visual Communications and Image Processing (VCIP), 2017, pp. 1–4.
82. C. Jia, S. Wang, X. Zhang, J. Liu, S. Pu, S. Wang, and S. Ma "Content-Aware Convolutional Neural Network for In-loop Filtering in High Efficiency Video Coding." Accepted by IEEE Trans. on Image Processing, 2019.
83. X. Zhang, S. Wang, K. Gu, W. Lin, S. Ma, and W. Gao "Just-noticeable difference-based perceptual optimization for JPEG compression." IEEE Signal Processing Letters, vol. 24, no. 1, pp. 96–100, 2017.
84. X. Song, J. Yao, L. Zhou, L. Wang, X. Wu, D. Xie, and S. Pu "A practical convolutional neural network as loop filter for intra frame." arXiv:1805.06121, 2018.
85. Y. Cui et al. "Convolutional Neural Networks for Intra Prediction in HEVC." Proc. PCS 2018.
86. S. Ma et al. "Image and Video Compression with Neural Networks: A Review." IEEE Trans. Circuits and Systems for Video Technology, 2019, 29(10): pp. 2872-2886.
87. MPEG 139 – MPEG. URL: https://www.mpeg.org/meetings/mpeg-139/ Дата звернення 01.12.2024
88. L. Xiao, H. Wang, N. Ling "Improved image coding autoencoder with deep learning." arXiv:2002.12521, 2020.
89. J. Lee, S. Cho, M. Kim "End-to-end Joint Learning of Image Compression and Quality Enhancement with Improved Entropy Minimization." arXiv:1912.12817, 2020.
90. W. Yang et al. "Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement." IEEE TPAMI, 2020.
91. Q. Chen, J. Chen "Streaming Lossless Volumetric Compression of Medical Images Using Gated R-CNN." arXiv:2311.16200, 2023.
92. M. Rippel, L. Bourdev "Real-Time Adaptive Image Compression." Proc. ICML 2017.
93. Луц В.К. (2024) "Методи і алгоритми для адаптивного кодування зображень" URL: https://incyb.kiev.ua/storage/editor/files/disertaciya-luts-vk.pdf
94. Britanak V., Rao K.R. and Yip P. "Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations." Oxford.

UK: Academic Press. - Elsevier. 2007. https://doi.org/10.1016/B978-0123736246/50001-4

95. Wang Z. "Fast algorithms for the discrete W transform and for the discrete Fourier transform." IEEE Trans. Acoust., Speech, Signal Process. 1984. Vol. 32, No. 8. P. 803-816 . https://doi.org/10.1109/TASSP.1984.1164450
96. B.M. Shevchuk, V.K. Zadiraka, L.O. Hnativ, and S.V. Fraer "Technology of Multifunctional Data Processing and Transmission in Monitoring Networks." [in Ukrainian], Naukova Dumka, Kyiv, 2010. 375 P. https://doi.org/10.1007/s10559-016-9810-9
97. Joshi R., Reznik Y., Karczewiz M. "Efficient large size transforms for high perfomance video coding." Proc. SPIE 7798 Appl. of Digital Image Process. XXXIII, 2010, pp. 7798.31 1–7
98. Yip, P., Rao, K.R. "Fast decimation-in-time algorithms for a family of discrete sine and cosine transforms." Circuits Systems and Signal Process 3, 1984, pp. 387–408. https://doi.org/10.1007/BF01599167
99. Luts V. K. "Fast integer cosine transform of order-8 for high-speed video coding," Proc. SPIE 13137, Applications of Digital Image Processing XLVII, 1313716 (California, 30 September 2024); https://doi.org/10.1117/12.3035188
100. Clarke R.J. "Relation between the Karhunen-Loeve and Sine transforms." Electron. Lett., 1984, vol. 20, no. 1. pp. 12-13. https://doi.org/10.1049/el:19840010
101. Chivukula R.K. and Reznik Y.A. "Fast computing of discrete cosine and sine transforms of types VI and VII." Proc. SPIE 8135, Appl. of Digital Image Processing, XXX IV, no 813505, Sept. 2011, pp. 1-10. https://doi.org/10.1117/12.903685
102. Z. Zhang, X. Zhao, X. Li, Z. Li, and S. Liu "Fast adaptive multiple transform for versatile video coding." Proc. Data Compress. Conf. (DCC), Mar. 2019, pp. 63–72. https://doi.org/10.1109/DCC.2019.00022
103. Fuldseth A., Bjøntegaard G., Sadafale M., at al. "CE10: Core transform design for HEVC." ITU-T, Doc. JCTVC-G495, Geneva, CH, Nov. 2011
104. V. K. Luts "New Approaches and Methods of Adaptive Image Encoding." Cybern Syst Anal., 2024, vol. 60, pp. 331–340. https://doi.org/10.1007/s10559-024-00673-8
105. Jeffrey S. Geronimo; Douglas P. Hardin; Peter R. Massopust. "Fractal Functions and Wavelet Expansions Based on Several Scaling Functions." Journal of Approximation Theory, 1994, Vol.78, №3, pp. 373-401.
106. Martin M.B., Bell A.E. "New image compression techniques using multiwavelets and multiwavelet packets." IEEE Trans Image Process. 2001, Vol.10, №4, pp. 500-510.
107. U.S. Ragupathy, D. Baskar and A. Tamilarasi "New Method of Image Compression using Multiwavelets and Set Partitioning Algorithm." Proceedings

of 10th IEEE International Conference on Industrial and Information Systems, 2008, pp. 1-6.
108. R.Sumalatha, M.V.Subramanyam "Medical Image Compression Using Multiwavelets for Telemedicine Applications." International Journal of Scientific & Engineering Research. 2011, Vol. 2, № 9, pp. 1-4.
109. M. Ashok and T.B. Reddy "Image Compression Techniques Using Modified high quality Multi wavelets." International Journal of Advanced Computer Science and Applications (IJACSA), 2011, Vol.2, №7, pp.153-158.
110. S. Radhakrishnan and J. Subramaniam "Fingerprint Compression using Multiwavelets." International Journal of Signal Processing, 2006, Vol. 2, № 2, pp. 78-87.
111. Rema N.R., Shanavaz K.T. and Mythili P. "Better fingerprint image compression at lower bit-rates: an approach using multiwavelets with optimised prefilter coefficients." ICTACT Journal on Image and Video Processing. 2017. Vol. 8 ,№ 1, pp. 1588-1595.
112. N.R. Rema and P. Mythili "Improved fingerprint compression technique with decimated multi-wavelet coefficients for low bit rates." ICTACT journal on image and video processing, 2018, Vol. 9, № 1, pp. 1801-1806.
113. N. R. Rema and P. Mythili "Extremely High Compression and Identification of Fingerprint Images Using SA4 Multiwavelet Transform." International Journal of Image and Graphics 2021,Vol. 21, No. 3, 2150037, https://doi.org/10.1142/S0219467821500376.
114. N. M. Sindhuja, T. Prathiba, S. N. Rani. "Edge Detection of Synthetic Image Using GHM Multiwavelet Transform." International Journal of Computer Applications in Engineering Sciences. 2012, Vol II, № IV, December, pp. 399-403.
115. Hnativ L.O. "The method of constructing a family of orthonormal basis systems of fractal multiwavelets based on step functions." Pr. International sh. "Issues of optimization of calculations (POO-HLII), Kyiv: Institute of Cybernetics National Academy of Sciences, 2015, P. 37-38.
116. S. Mallat. "A Wavelet Tour of Signal Processing." 1998, 577 p.
117. Mallat S "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." IEEE Trans. Pattern Anal. Machine Intell. 1989, Vol. 11,№7, pp. 674-693.
118. Coifman R R, Meyer Y., Wickerhauser V. "Wavelet analysis and signal processing, in: Wavelets and Their Applications." Boston, 1992. pp. 157-178.