

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи стиснення відеоданих для підвищення
ефективності комп'ютерних систем

(тема)

Виконав:

студент II курсу, групи СПМ-23-1
Ігнат'єв О.О.
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: ст. викл. Ні Я.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.

кафедри _____

(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Ігнат'єву Олександрю Олексійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи стиснення відеоданих для підвищення ефективності комп'ютерних систем _____

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 січня 2025 р. _____

3. Вхідні дані до роботи _____

Windows 10 _____

Visual Studio Code _____

Node.js _____

Персональний комп'ютер _____

Доступ до мережі інтернет _____

4. Перелік питань, що потрібно опрацювати у роботі _____

Огляд алгоритмів стиснення: MPEG-4, H.264, HEVC _____

Оцінка ефективності існуючих систем _____

Проведення експериментів для підвищення ефективності стиснення відеоданих _____

Оцінка результатів експериментів _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____
17 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд алгоритмів стиснення: MPEG-4, H.264, HEVC	26.11.24-30.11.24	
2	Вибір та обґрунтування методики дослідження	02.12.24-05.12.24	
3	Вибір інструментальних засобів	06.12.24-10.12.24	
4	Оцінка ефективності існуючих систем	11.12.24-21.12.24	
5	Проведення експериментів для підвищення ефективності стиснення відеоданих	23.12.24-03.01.25	
6	Оформлення матеріалів кваліфікаційної роботи	04.01.25-07.01.25	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	08.01.25-11.01.25	
8	Подання кваліфікаційної роботи на рецензування	13.01.25-17.01.25	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Ні Я.С.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 16 рис., 8 табл., 1 дод., 16 джерел.

ВІДЕОСТИСНЕННЯ, КОДУВАННЯ, СЕГМЕНТАЦІЯ,
КВАНТУВАННЯ, MPEG-4, ІНФОРМАТИВНІ БЛОКИ, ДКП.

Метою кваліфікаційної роботи є вдосконалення процесів стиснення відео в стандарті MPEG-4 через інтеграцію адаптивних методів, таких як адаптивне DCT, адаптивне квантування, адаптивний аналіз руху.

Проведено ґрунтовний аналіз сучасних стандартів відеостиснення, зокрема MPEG-4. Запропоновано нові адаптивні підходи, зокрема сегментацію макроблоків за середньою довжиною змін для визначення інформативності блоків, що дало змогу розділити відеокадри на інформативні та неінформативні області. Також вдосконалено процеси ДКП (дискретного косинусного перетворення) і квантування. Адаптивний підхід до квантування забезпечує високу точність збереження деталей у критичних областях кадру та значне стиснення у менш важливих зонах. Інтеграція арифметичного кодування замість традиційного кодування Хаффмана дозволила досягти ще більшого рівня стиснення завдяки точнішому врахуванню ймовірностей символів.

Запропоновані вдосконалення демонструють суттєві переваги у порівнянні з класичним MPEG-4. Проведені експерименти показали підвищення коефіцієнта стиснення до 50%, покращення показників PSNR і SSIM, а також зниження обчислювальних витрат, що дозволяє ефективно передавати великі обсяги відеоданих навіть у мережах із обмеженою пропускнуою здатністю.

ABSTRACT

Master's thesis: 78 pages, 16 figures, 8 tables, 1 appendices, 16 sources.

VIDEO COMPRESSION, CODING, SEGMENTATION,
QUANTISATION, MPEG-4, INFORMATION BLOCKS, DCT.

The purpose of the qualification work is to improve the video compression processes in the MPEG-4 standard through the integration of adaptive methods, such as adaptive DCT, adaptive quantisation, adaptive motion analysis.

In the course of the qualification work, a thorough analysis of modern video compression standards, in particular MPEG-4. New adaptive approaches were proposed, including segmentation of macroblocks by the average length of changes to determine the information content of the blocks, which made it possible to divide video frames into informative and non-informative areas. The DCT (Discrete Cosine Transform) and quantisation processes were also improved. An adaptive quantisation approach ensures high detail preservation in critical areas of the frame and significant compression in less important areas. The integration of arithmetic coding instead of traditional Huffman coding has allowed to achieve even higher compression levels due to more accurate consideration of symbol probabilities.

The proposed improvements demonstrate significant advantages over the classical MPEG-4. Experiments have shown an increase in compression ratio of up to 50%, improved PSNR and SSIM, as well as reduced computational costs, which allows efficient transmission of large amounts of video data even in networks with limited bandwidth.

ЗМІСТ

ЗМІСТ	8
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Актуальність дослідження та практична затребуваність методів стиснення відеоданих	11
1.2 Аналіз існуючих стандартів та вимог до відеокомпресії в телекомунікаційних мережах.....	14
1.3 Критерії оцінювання ефективності функціонування комп'ютерних систем	19
1.4 Обґрунтування необхідності адаптації існуючих методів стиснення відеоданих до специфічних вимог телекомунікаційних мереж	21
1.5 Мета та задачі дослідження	23
2 АНАЛІЗ ТЕХНОЛОГІЧНОГО ТА МЕТОДОЛОГІЧНОГО ПІДГРУНТЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	25
2.1 Класифікація методів стиснення відео	25
2.2 Технічний аналіз найпоширеніших алгоритмів стиснення: MPEG- 4, H.264, HEVC	28
2.3 Порівняльний аналіз ефективності різних методів стосовно специфікацій відео	32
2.4 Огляд технологій програмування для вирішення поставленої задачі.....	35
3 МОДЕРНІЗАЦІЯ ІСНУЮЧОГО СТАНДАРТУ ВІДЕОКОМПРЕСІЇ.....	37
3.1 Модернізація підвищення ефективності функціонування систем шляхом інтегрування адаптивних методів	37

3.2 Інтеграція сегментації кадру з пошуком важливої частки відео	39
3.3 Опис модифікації етапу компенсації руху	41
3.4 Адаптація DST та квантування для покращення методу	43
3.5 Інтеграція арифметичного кодування до методу.....	46
РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ ПРОВЕДЕНИХ ЕКСПЕРИМЕНТІВ	51
4.1. Програмна реалізація сегментації за середньою довжиною	51
4.2 Реалізація додаткових методів для порівняння	54
4.3 Моделювання та порівняння методів.....	57
4.4 Оцінка ефективності запропонованого методу через аналіз відеоданих.....	62
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67
ДОДАТОК А.....	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Інформативний блок (inf) – Блок кадру, що містить важливі деталі і потребує високої точності обробки

Квантування – Процес округлення частотних компонентів для зменшення обсягу даних під час стиснення

Неінформативний блок (non-inf) – Блок кадру, що містить однорідні ділянки або фон і може бути оброблений із меншою точністю

Сегментація – Процес поділу кадру на макроблоки для визначення інформативних та неінформативних ділянок

CR – Compression Ratio (коефіцієнт стиснення, що показує відношення між обсягом вихідних і стиснених даних)

DCT – Дискретне косинусне перетворення (математичне перетворення, яке переводить дані з просторової у частотну область)

MPEG-4 – Moving Picture Experts Group 4 (стандарт стиснення відео та аудіо, що забезпечує високу ефективність зменшення обсягу даних)

PSNR – Peak Signal-to-Noise Ratio (метрика для оцінки якості стисненого відео)

RGB – Red, Green, Blue (кольоровий простір, що використовується для представлення зображень)

SSIM – Structural Similarity Index Measure (метрика для оцінки структурної схожості між оригінальним та стисненим відео)

YCbCr – Формат кольорового простору (яскравісна складова Y та кольорові складові Cb і Cr)

ВСТУП

В умовах сучасного стрімкого розвитку цифрових технологій та глобальної інтернет-економіки відео стало одним із головних засобів комунікації, розваг і передачі інформації. Щодня у світі створюються і передаються мільярди годин відеоконтенту, що призводить до постійного збільшення обсягів відеоданих. За останніми даними, відео вже складає понад 80% від усього інтернет-трафіку, і ця цифра продовжує зростати завдяки популярності стрімінгових сервісів, соціальних мереж, відеоконференцій та систем відеоспостереження.

З кожним роком вимоги до якості відео лише зростають: формати 4K і 8K стають все більш поширеними, а нові технології, як-от 360-градусне відео, віртуальна та доповнена реальність, задають нові стандарти у створенні та передачі відеоконтенту. Всі ці тенденції створюють серйозні виклики для телекомунікаційних мереж, адже традиційні підходи до обробки та передачі даних часто не справляються із підвищеними вимогами щодо пропускної здатності, швидкості та зберігання відео.

Одним із найбільш ефективних рішень для подолання цих викликів є використання сучасних методів стиснення відеоданих. Завдяки алгоритмам стиснення можна суттєво зменшити обсяг відео без помітної втрати якості зображення. Це дозволяє оптимізувати роботу мереж і пристроїв, знизити витрати на передачу та зберігання даних, а також забезпечити користувачам стабільний доступ до відеоконтенту. Без ефективних технологій стиснення обробка відеоданих вимагала б набагато більших ресурсів, а це могло б призвести до збоїв у роботі систем і погіршення користувацького досвіду.

Особливо важливу роль стиснення відео відіграє у телекомунікаційних мережах, де умови передачі даних можуть змінюватися в реальному часі. Наприклад, у мобільних мережах 4G і 5G або Wi-Fi з'єднаннях пропускна здатність може бути обмеженою або нестабільною. Сучасні алгоритми

стиснення допомагають забезпечити стабільну якість відеопередачі, навіть якщо умови мережі не ідеальні. Крім того, передача відео в режимі реального часу, як у випадку з відеоконференціями або відеоспостереженням, висуває додаткові вимоги до ефективності стиснення і мінімізації затримок.

Потреба у якісному стисненні відео також є критично важливою для низки галузей. Наприклад, у сфері телемедицини якість відео може мати вирішальне значення для проведення дистанційної діагностики або навіть хірургічних втручань. У безпекових системах відеоспостереження важливо отримувати чітке відео в режимі реального часу, аби мати можливість оперативно реагувати на події. В усіх цих випадках стиснення відео не тільки допомагає зменшити обсяг даних, але й дозволяє забезпечити високу якість зображення при мінімальних ресурсах.

Отже, дослідження та розвиток методів стиснення відеоданих набувають особливої важливості в сучасних умовах, коли обсяги відеоконтенту продовжують зростати, а вимоги до якості передачі даних стають все більш жорсткими. Розробка нових алгоритмів стиснення та їх адаптація до специфічних вимог телекомунікаційних мереж є ключовим завданням для забезпечення ефективності роботи комп'ютерних систем і надання якісного відеоконтенту користувачам.

Метою цієї роботи є детальний аналіз існуючих методів стиснення відеоданих, їх порівняння, а також розробка рекомендацій щодо їхнього вдосконалення та адаптації до умов сучасних телекомунікаційних мереж. Основна увага зосереджена на пошуку ефективних рішень, які допоможуть мінімізувати навантаження на інфраструктуру мережі, забезпечуючи при цьому високу якість відео і низькі затримки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність дослідження та практична затребуваність методів стиснення відеоданих

Розвиток сучасних технологій та зростання обсягів відеоконтенту сприяють активному інтересу до методів стиснення та обробки відеоданих. Мотивуючими факторами, що визначають актуальність дослідження та вдосконалення технологій стиснення відео, є такі:

- зростання обсягів відеоданих та тенденція до збільшення якості відео: постійне збільшення попиту на відеоконтент у форматах 4K, 8K, 360-градусне відео, VR та AR підвищує вимоги до технологій обробки, зберігання та передачі відео;

- обмеженість ресурсів для зберігання та передачі: інтернет-трафік, який формується відеоконтентом, потребує значних обчислювальних ресурсів і пропускної здатності мереж. ефективне стиснення дозволяє зменшити навантаження на мережі та оптимізувати використання ресурсів для зберігання;

- затребуваність у телемедицині, безпекових системах та промисловості: у багатьох сферах якісне відео є критично важливим для забезпечення оперативності та точності дій, зокрема у дистанційній діагностиці, відеоспостереженні та автоматизованих системах;

- зростання популярності мобільних пристроїв: в умовах обмеженої пропускної здатності мобільних мереж 4G та 5G стиснення відео дозволяє забезпечити якісний користувацький досвід при мінімальних витратах ресурсів;

- проблеми енергоефективності: питання енергоспоживання стає критичним при роботі з великими обсягами відеоданих. Сучасні методи стиснення допомагають зменшити споживання енергії як на стороні серверів,

так і на стороні користувачьких пристроїв.

З урахуванням цих факторів, сфери застосування відеоконтенту постійно розширюються. Далі представлена класифікаційна діаграма основних сфер застосування відеоданих (рисунок 1.1).



Рисунок 1.1 - Класифікаційна діаграма сфер застосування відеоданих

Діаграма наочно демонструє широкий спектр сфер застосування відеоданих, кожна з яких має свої особливі вимоги до якості, швидкості передачі та обробки. У кожній із цих сфер ефективне стиснення відео стає вирішальним фактором для забезпечення високої продуктивності, зниження навантаження на інфраструктуру та покращення загального користувачького досвіду. Наприклад, у стрімінгових сервісах основний акцент ставиться на оптимізацію передачі відео високої чіткості для кінцевого користувача, тоді

як у телемедицині критичним є мінімізація затримок і збереження максимальної точності зображення. У зв'язку з цим стає очевидною потреба у подальших дослідженнях і вдосконаленні технологій стиснення, які можуть вирішувати сучасні виклики у всіх цих сферах[1].

Ефективне стиснення відеоданих є одним із ключових напрямків розвитку сучасних інформаційних технологій. Воно дозволяє знижувати навантаження на телекомунікаційні мережі, зменшувати обсяг необхідної пам'яті для зберігання даних та покращувати користувацький досвід, забезпечуючи стабільну якість відео. У зв'язку з постійним зростанням обсягів відеоконтенту та вимог до його якості, розвиток і вдосконалення методів стиснення є критично важливим завданням [2,3].

Щоб забезпечити ефективне стиснення відеоданих і оптимально використовувати наявні ресурси, застосовуються різноманітні методи, які дозволяють балансувати між якістю відео та обсягом даних. Загальна схема стиснення (рисунок 1.2) відображає основні етапи, що забезпечують перетворення відеоконтенту в більш компактний формат без критичних втрат якості. Кожен етап цієї схеми спрямований на мінімізацію зайвих даних і зменшення навантаження на мережу або пристрій, що є особливо актуальним для сучасних додатків, які працюють з обмеженими ресурсами.

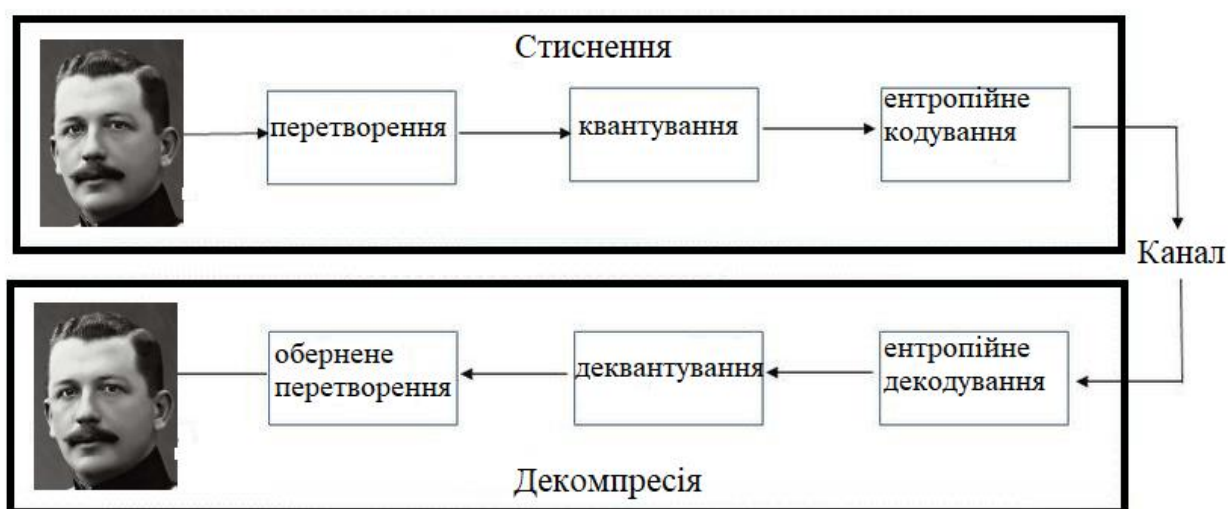


Рисунок 1.2 - Загальна схема стиснення

Стиснення відео має велике значення в сучасному світі, де обсяг відеоконтенту продовжує стрімко зростати. Підвищення роздільної здатності, вимоги до якості передачі в реальному часі та обмеження інфраструктури створюють безпрецедентні виклики для ефективної обробки відеоданих. Методи стиснення відіграють вирішальну роль у забезпеченні високоякісного користувацького досвіду та оптимізації ресурсів мережі.

Розвиток технологій стиснення дозволяє подолати труднощі, пов'язані з великими обсягами даних і високими вимогами до якості. У стрімінгових сервісах такі методи допомагають підтримувати якість зображення та стабільну передачу для кінцевого користувача, у той час як у галузях, як-от телемедицина і відеоспостереження, вони забезпечують високу точність і мінімальну затримку.

1.2 Аналіз існуючих стандартів та вимог до відеокомпресії в телекомунікаційних мережах

Сучасні відеокомпресійні стандарти розроблені для забезпечення ефективної передачі відеоданих через різні мережі, включаючи телекомунікаційні. Основні стандарти представлені на Рисунку 1.3.

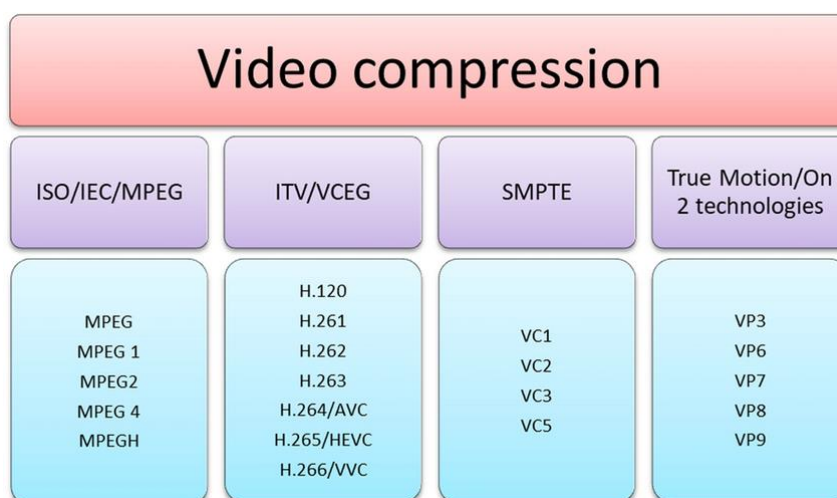


Рисунок 1.3 - Класифікаційна діаграма стандартів відеокомпресії

У таблиці 1.1 надамо основні характеристики представлених стандартів стиснення.

Таблиця 1.1 - Основні характеристики відеокомпресійних стандартів

Стандарт	MPE G-4	H.264 (AVC)	H.265 (HEVC)	VP8	VP9	AV1
Рік випуску	1998	2003	2013	2008	2013	2018
Компресія	Сере дня	Висока	Дуже висока	Висока	Дуже висока	Дуже висока
Максимальна роздільна спром.	До 4К	До 4К	До 8К	До 4К	До 8К	До 8К
Ефективність використання смуги пропускання	Низь ка	Висока	Дуже висока	Висока	Вища за VP8	Дуже висока
Затримка (latency)	Сере дня	Низька	Вища ніж H.265	Низька	Низька	Середня
Підтримка апаратного декодування	Шир ока	Широка	Широка(з алежить від пристроїв)	Середня	Середн я	Обмежена (Поступово розширюєт ься)
Застосув ання	Стрі мінг, DVD, відео на мобіл ьних	Стрімінг, відеокон ференції, телебаче ння	Стрімінг, UHD- телебачен ня	Веб-відео (YouTube, WebRTC)		Стрімінг, майбутнє відео у веб

Основні стандарти відеокомпресії, такі як MPEG-4, H.264, H.265, VP8, VP9 та AV1, відіграють ключову роль у забезпеченні якості та ефективності передачі відеоданих у сучасних телекомунікаційних мережах. Кожен із цих стандартів має свої характерні риси, які визначають його оптимальне застосування, переваги та обмеження[4,5].

MPEG-4 залишається популярним стандартом, забезпечуючи баланс між якістю зображення та обсягом файлів. Хоча його ефективність стиснення порівняно низька, а затримка обробки є середньою, MPEG-4 добре підходить для стрімінгового відео на мобільних пристроях, DVD та у веб-додатках. H.264 (AVC), як наступний етап розвитку, значно підвищує ефективність стиснення та якість передачі відео. Цей стандарт дозволяє передавати відео у високій роздільній здатності з меншими витратами пропускну здатності, що зробило його провідним у потокових сервісах і відеоконференціях завдяки зниженню затримок і покращенню адаптивності до мережових умов.

Подальший розвиток стандартів стиснення привів до створення H.265 (HEVC), що був розроблений для забезпечення максимальної компресії відео 4K та 8K роздільної здатності. H.265 удвічі підвищує ефективність стиснення у порівнянні з H.264, зменшуючи обсяги даних на 50% при збереженні якості відео. Проте висока складність алгоритмів H.265 потребує великих обчислювальних ресурсів, що обмежує його широке впровадження у мережах з низькою пропускну здатністю.

VP8 та VP9 були розроблені Google як безкоштовні альтернативи H.264 та H.265 відповідно. VP8, хоча і менш ефективний за H.264 у збереженні високої якості відео при низькому бітрейті, залишається популярним у веб-середовищі. VP9, у свою чергу, є конкурентом H.265 за рівнем компресії та якості, підтримуючи відео 8K. Завдяки ефективній інтеграції з платформою YouTube, VP9 став стандартом для веб-відео, що забезпечує високу якість при обмеженій пропускну здатності.

Останній стандарт, AV1, відзначається надзвичайно високою компресією без втрат якості. AV1 дозволяє значно знизити навантаження на

мережі, особливо в умовах адаптивної передачі відео. Завдяки підтримці високої динаміки кольорів (HDR) та ефективності стиснення, цей стандарт є перспективним вибором для стрімінгових платформ майбутнього, незважаючи на затримки, що все ще залишаються високими.

Таким чином, розвиток відеокompresійних стандартів відображає постійний пошук балансу між ефективністю стиснення, якістю зображення та вимогами до обчислювальних ресурсів. Кожен із стандартів пропонує свої переваги для різних сценаріїв застосування, що дозволяє обирати оптимальне рішення для конкретних умов роботи [4].

Основні вимоги до відеокompresії в телекомунікаційних мережах:

- ефективність використання смуги пропускання (bitrate efficiency): одним з ключових аспектів для компresії є здатність передавати високоякісне відео при мінімальному використанні пропускнуої здатності. Це критично важливо для мобільних мереж, Wi-Fi, а також для мереж зі змінними умовами (наприклад, при потоковій передачі);

- стійкість до втрат пакетів та джиттеру (packet loss and jitter): в умовах реальних телекомунікаційних мереж можливі втрати пакетів, збої та варіації у часі доставки (джиттер). Алгоритми компresії повинні мати механізми корекції помилок та адаптацію до таких умов, щоб уникати значних артефактів відео;

- низька затримка (latency): для відеоконференцій та інтерактивних додатків важлива низька затримка. Це досягається завдяки швидкому кодуванню та декодуванню, що дозволяє передавати відео в режимі реального часу. H.264 та VP8 добре підходять для таких випадків, але HEVC або AV1 можуть мати більші затримки через складніші алгоритми кодування;

- адаптивність до зміни пропускнуої здатності дозволяє змінювати бітрейт у реальному часі залежно від поточного стану мережі, що забезпечує безперервне відтворення навіть при коливаннях пропускнуої здатності.

Порівнюючи існуючі відеокompresні стандарти (Таблиця 1.2) можемо зробити деякі висновки:

Таблиця 1.2 - Порівняння стандартів за ефективністю та якістю

Параметр	Розмір файлу	Якість відео	Складність кодування	Стійкість до втрат	Підтримка HDR	Адаптивність до мереж
MPEG-4	Великий	Середня	Низька	Низька	Ні	Слабка
H.264 (AVC)	Середній	Висока	Середня	Висока	Частково	Висока
H.265 (HEVC)	Малий	Дуже висока	Висока	Висока	Так	Дуже висока
VP8	Середній	Висока	Середня	Висока	Ні	Висока
VP9	Малий	Дуже висока	Висока	Висока	Так	Дуже висока
AV1	Дуже малий	Дуже висока	Дуже висока	Висока	Так	Дуже висока

Як бачимо, кожен стандарт компресії має свої особливості та переваги для різних умов. Вибір стандарту залежить від конкретних вимог до якості відео, смуги пропускання, затримки та ресурсів мережі [4,5].

Вимоги до стандартів у різних типах мереж:

- мобільні мережі (3G, 4G, 5G). В таких мережах використовуються стандарти з високою ефективністю стиснення і адаптивністю до пропускну здатності (наприклад, H.264, H.265). Стандарти повинні забезпечувати високу стійкість до втрат пакетів і джиттеру, що важливо для передачі відео у реальному часі;

- Wi-Fi та локальні мережі. Для цих мереж більш придатними є стандарти з меншими вимогами до обчислювальних ресурсів (H.264, VP8), що дозволяє підтримувати потокову передачу відео в межах офісу або будинку;

- мережі широкого охоплення (кабельне/супутникове телебачення). В таких мережах пріоритет надається високій якості відео з підтримкою роздільної здатності до 4K і 8K, тому найчастіше використовується H.265 або VP9.

1.3 Критерії оцінювання ефективності функціонування комп'ютерних систем

Ефективність роботи комп'ютерних систем у телекомунікаційних мережах суттєво залежить від таких характеристик, як затримка (latency), швидкість передачі даних (throughput) та втрати пакетів (packet loss). Ці показники є вирішальними для підтримки належного рівня обслуговування і безперебійної передачі відео, що безпосередньо впливає на користувацький досвід [6].

Затримка (Latency) — це час, необхідний для передачі пакета даних від відправника до отримувача, і вона є ключовим показником для застосувань, які вимагають реального часу. Для відеоконференцій, телемедицини або систем, що працюють із віддаленим управлінням, низька затримка має вирішальне значення. Затримку зазвичай розділяють на односторонню (тобто час, за який пакет доходить до одержувача) і кругову затримку (час на передачу і повернення пакета). Джерелами затримки можуть бути як час передачі пакета по каналу, так і маршрутизаційна затримка, яка залежить від фізичної відстані та швидкості поширення сигналу. На додачу, затримка також виникає під час обробки пакетів і в чергах на комутаторах та маршрутизаторах, що може бути викликано високим навантаженням на мережу. В умовах, коли затримка перевищує 150–200 мс, її починають помічати користувачі, що погіршує їхнє враження від сервісу.

Швидкість передачі даних (Throughput) визначає обсяг інформації, яку можна передати мережею за певний час, і є важливим параметром для потокового відео і передачі великих файлів. Швидкість передачі залежить як від номінальної пропускної здатності мережі, так і від фактичного навантаження, яке може змінюватися в залежності від кількості активних користувачів і типу використовуваних протоколів, таких як TCP або UDP. Швидкість передачі часто обмежується реальними умовами роботи мережі та факторами, такими як черги на маршрутизаторах, затримки та втрати пакетів.

Оптимальна швидкість передачі забезпечує безперервне відтворення відео та запобігає буферизації, тоді як недостатня швидкість змушує знижувати якість відео або може викликати зупинки у відтворенні.

Втрата пакетів (Packet Loss) відбувається, коли один або кілька пакетів не досягають кінцевого пункту призначення, і є наслідком мережевого перевантаження, помилок у передачі або збоїв у маршрутизації. Такі втрати можуть бути випадковими або послідовними, що викликає серйозні перебої у відтворенні відео. Навіть мінімальні втрати можуть бути помітними при передачі високоякісного відео, оскільки вони викликають артефакти, зупинки у відтворенні або зниження чіткості зображення. Для подолання втрат пакетів використовують алгоритми корекції помилок, проте значні втрати все одно негативно впливають на якість передачі, що є особливо критичним для відео в режимі реального часу [7].

Підводячи аналіз всього вище сказаного про існуючі ключові параметри, пропонується порівняти вплив кожного з них на передачу відео (таблиця 1.3):

Таблиця 1.3 - Порівняльна таблиця впливу основних факторів на передачу відео

Параметр	Опис	Критичність для відео в реальному часі	Критичність для потокового відео
Затримка (Latency)	Час між передачею та отриманням пакета	Дуже висока	Середня
Швидкість (Throughput)	Обсяг даних, які передаються за секунду	Висока	Висока
Втрати пакетів (Packet Loss)	Кількість загублених пакетів	Дуже висока	Висока

Ефективність комп'ютерних систем для передачі відеоданих визначається балансом між затримкою, швидкістю передачі і втратами пакетів. Ці критерії є взаємозалежними, і оптимізація одного з них часто може негативно вплинути на інші, тому важливо знаходити компроміс залежно від конкретних вимог системи (наприклад, відеоконференції або стрімінг) [6].

1.4 Обґрунтування необхідності адаптації існуючих методів стиснення відеоданих до специфічних вимог телекомунікаційних мереж

Стиснення відеоданих сьогодні є одним із критичних процесів для забезпечення високої якості відео в умовах телекомунікаційних мереж, які часто стикаються з обмеженнями щодо пропускної здатності і вимогами до мінімізації затримок. Незважаючи на те, що стандарти стиснення, такі як H.264 або HEVC, надають високоякісні результати, вони були розроблені здебільшого для статичних або стабільних мереж. У динамічних умовах телекомунікацій, які характеризуються високим рівнем змінності пропускної здатності, втратами пакетів і великими навантаженнями, ці методи можуть працювати неефективно, що підкреслює необхідність їхньої адаптації.

Перш за все, телекомунікаційні мережі мають непередбачувану пропускну здатність, яка змінюється в залежності від часу доби, кількості користувачів і типу мережі. Така мінливість створює умови, де традиційні алгоритми стиснення з фіксованим бітрейтом не можуть гарантувати стабільної якості відео. У таких випадках адаптивне кодування, яке дозволяє динамічно змінювати якість потоку відповідно до пропускної здатності мережі, може значно покращити стабільність передачі і мінімізувати ймовірність втрат кадрів.

Також слід врахувати особливості мобільних мереж, де пропускну здатність обмежена, а навантаження на мережу сильно коливається. У таких мережах важливо забезпечити ефективне стиснення, яке дозволяє передавати

дані з мінімальними втратами, навіть під час пікових навантажень. Наприклад, використання алгоритмів, заснованих на штучному інтелекті або машинному навчанні, може допомогти зберігати якість відео на високому рівні за обмеженої пропускної здатності. Інтелектуальні алгоритми здатні навчатися на мережевих параметрах, прогнозувати можливі затримки і динамічно налаштовувати бітрейт, знижуючи його без суттєвих втрат якості.

Окрім цього, у бездротових мережах, де часті втрати пакетів та джиттер є звичним явищем через перешкоди або змінні умови сигналу, необхідно впроваджувати алгоритми, які можуть відновлювати втрачені кадри або зменшувати вплив артефактів на якість відео. У таких випадках технології, як Forward Error Correction (FEC) або адаптивне відновлення кадрів, дозволяють коригувати помилки на рівні стиснення відеоданих, що допомагає підтримувати стабільну якість навіть при значних втратах пакетів.

Складність завдань підвищується ще й через потребу підтримувати високу якість відео на різних пристроях, від мобільних телефонів до екранів із підтримкою 4K і 8K. Це вимагає розробки масштабованого кодування відео (Scalable Video Coding, SVC), яке дозволяє передавати відео у декількох якостях, щоб кожен пристрій міг автоматично обирати оптимальний рівень на основі своїх можливостей і стану мережі. Динамічне масштабування роздільної здатності також дозволяє оптимізувати якість відео в реальному часі, адаптуючи її до змінних умов мережі.

З іншого боку, розвиток мереж 5G, які відзначаються високою пропускною здатністю і низькою затримкою, відкриває нові можливості для застосування складних і більш ефективних методів стиснення, таких як H.266 (Versatile Video Coding, VVC). Цей стандарт стиснення розроблений спеціально для мереж із високою швидкістю передачі даних і забезпечує набагато більшу ефективність у порівнянні з попередниками, що дозволяє ще більше оптимізувати використання телекомунікаційних ресурсів.

Таким чином, адаптація існуючих методів стиснення відеоданих до специфічних вимог телекомунікаційних мереж не просто необхідна — вона є

вирішальним фактором для забезпечення якісної і стабільної роботи сучасних відеосистем[3].

1.5 Мета та задачі дослідження

Метою дослідження є створення ефективного адаптивного методу стиснення відеоданих, який дозволить покращити продуктивність комп'ютерних систем у телекомунікаційних мережах. Для досягнення цієї мети необхідно вирішити низку завдань, пов'язаних із аналізом існуючих методів стиснення, вивченням впливу мережевих параметрів на якість передачі відео, розробкою та тестуванням нових адаптивних алгоритмів.

Основні задачі дослідження:

- аналіз існуючих методів стиснення відеоданих та їхньої ефективності шляхом огляду сучасних стандартів відеокомпресії (H.264, H.265/HEVC, AV1 тощо) для визначення їх сильних та слабких сторін у контексті роботи в умовах телекомунікаційних мереж, а також оцінки їх здатності адаптуватися до змінних параметрів мережі;
- дослідження впливу мережевих параметрів на якість передачі відео через аналіз таких факторів, як затримка, втрати пакетів, швидкість передачі даних та джиттер, з метою оцінки, як ці параметри впливають на ефективність відеокомпресії та користувацький досвід (QoE) при роботі комп'ютерних систем;
- розробка адаптивного методу стиснення відеоданих із використанням сучасних технологій, таких як машинне навчання або інтелектуальні алгоритми, що дозволить динамічно змінювати параметри стиснення залежно від умов мережі для досягнення балансу між якістю відео та ефективністю його передачі;
- експериментальна перевірка розробленого методу шляхом моделювання та тестування його в умовах, що відображають змінні параметри телекомунікаційних мереж, включаючи пропускну здатність,

затримку та втрати пакетів, для порівняння ефективності розробленого методу з існуючими стандартами;

- розробка практичних рекомендацій щодо застосування адаптивних методів стиснення відеоданих у телекомунікаційних мережах різних типів, зокрема, для визначення оптимальних параметрів стиснення для мобільних, бездротових і оптичних мереж у таких сценаріях, як відеоконференції, стрімінгові платформи та трансляції високоякісного контенту.

2 АНАЛІЗ ТЕХНОЛОГІЧНОГО ТА МЕТОДОЛОГІЧНОГО ПІДГРУНТЯ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Класифікація методів стиснення відео

Стиснення відеоданих є ключовим аспектом сучасних технологій, що дозволяє ефективно зберігати та передавати відео в умовах обмежених ресурсів. Використання ефективних алгоритмів стиснення дає змогу зменшувати обсяг відеофайлів без значної втрати якості, знижуючи навантаження на мережу і обсяг пам'яті. Сучасні методи стиснення відео зазвичай поділяються на дві основні категорії: стиснення без втрат і стиснення з втратами (рисунок 2.1). Кожен із цих підходів має свої переваги та обмеження, тому застосовується у різних сценаріях, виходячи з вимог до якості та ресурсів.

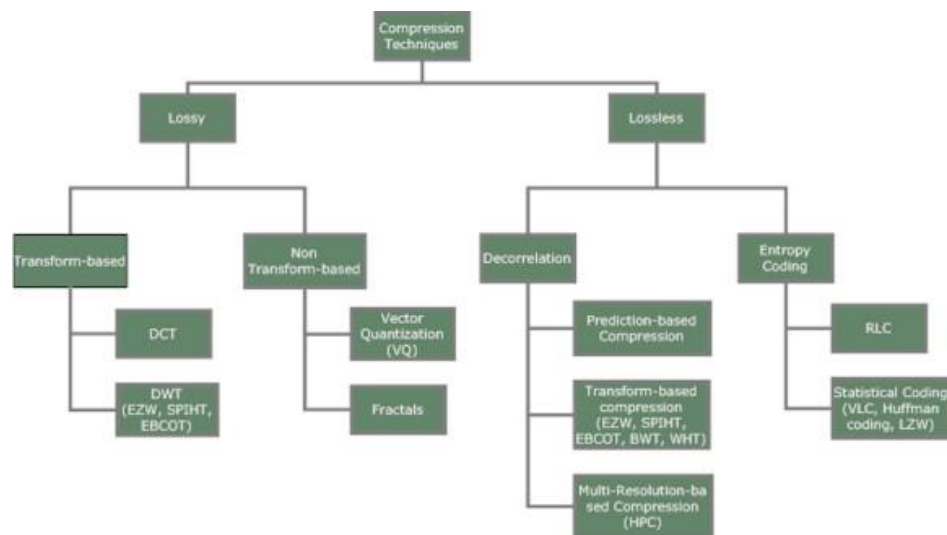


Рисунок 2.1 - Типи стиснення даних

Стиснення без втрат (Lossless Compression).

Методи стиснення без втрат дозволяють відновити відео в початковій якості, зберігаючи всі вихідні дані без жодних змін. Це означає, що

декодоване відео є повністю ідентичним оригіналу. Оскільки методи стиснення без втрат зберігають усі деталі, вони забезпечують точне відтворення, що є критичним для таких сфер, як медицина, наукові дослідження та архівування, де важливо зберегти кожен піксель.

Основні алгоритми стиснення без втрат включають[8]:

- run-Length Encoding (RLE), що стискає дані, замінюючи послідовності однакових пікселів на коротший опис: наприклад, замість довгого ряду білих пікселів зберігається лише «кількість білих пікселів». Цей метод є ефективним для зображень з малою кількістю кольорів;

- huffman Coding, який присвоює коротші коди більш поширеним символам і довші менш поширеним, створюючи компактне представлення даних;

- arithmetic Coding, що на відміну від Huffman Coding кодує дані у вигляді дробів, забезпечуючи ефективніше представлення довгих послідовностей;

- lempel-Ziv-Welch (LZW), що створює словник для певних послідовностей символів і замінює повторювані послідовності на коди, зменшуючи розмір даних;

- predictive Coding, де значення наступного пікселя передбачають на основі попередніх значень, записуючи тільки відмінності.

Переваги стиснення без втрат — це точне відновлення оригіналу і збереження всіх деталей. Водночас, такі методи мають менший коефіцієнт стиснення, що обмежує їхнє застосування для великих відеопотоків.

Стиснення з втратами (Lossy Compression).

Методи стиснення з втратами зменшують обсяг даних, видаляючи інформацію, яка вважається несуттєвою для людського зору. Це дозволяє досягти набагато більшого коефіцієнта стиснення порівняно з методами без втрат, зберігаючи при цьому візуальну якість на високому рівні. Стиснення з втратами широко використовується для передачі потокового відео, оскільки знижує вимоги до пропускної здатності, що дозволяє забезпечити якісний відеопотік навіть у завантажених мережах.

До основних методів стиснення з втратами належать[9]:

- transform Coding (Перетворювальне кодування), зокрема дискретне косинусне перетворення (DCT), яке є основою для стандартів JPEG, MPEG, H.264 і HEVC. Цей метод переводить зображення в частотну область, де високочастотні деталі, менш помітні для людського ока, можуть бути видалені або сильно стиснуті;

- wavelet-Based Coding використовує вейвлет-перетворення для зменшення обсягу даних, дозволяючи зберегти більшу кількість деталей і забезпечуючи кращий баланс між якістю та ступенем стиснення. Метод часто використовується в JPEG 2000;

- motion Compensation (Компенсація руху), де зберігаються тільки відмінності між послідовними кадрами, що значно зменшує обсяг даних при кодуванні відеопотоків. Цей метод є невід'ємною частиною кодеків, таких як MPEG і H.264;

- quantization (Квантування), який зменшує розрядність пікселів, усуваючи подібні значення. Агресивне квантування може призводити до артефактів, проте дозволяє значно зменшити розмір файлу.

Методи стиснення з втратами забезпечують високий рівень компресії, що робить їх оптимальними для відеопотоків у стрімінгових сервісах і відеоконференціях. Однак їхнім недоліком є незворотна втрата якості, яка може стати помітною при подальшій обробці або повторному стисненні.

Порівняння методів стиснення.

Враховуючи різноманітність підходів, кожен із методів стиснення має свої специфічні застосування і переваги. Залежно від вимог до точності передачі даних, швидкості стиснення та якості зображення, можна обрати оптимальний метод для конкретної задачі. Основні параметри, які впливають на вибір методу, включають рівень стиснення, можливість повного відновлення даних і відповідність обмеженням мережі.

Порівняння ключових характеристик методів стиснення представлено у таблиці 2.1 "Порівняльна таблиця методів стиснення".

Таблиця 2.1 - Порівняльна таблиця методів стиснення

Характеристика	Стиснення без втрат	Стиснення з втратами
Відновлення даних	Повне	Часткове
Ступінь стиснення	Низький (2:1 - 3:1)	Високий (до 100:1 і більше)
Якість відео	Незмінна	Може погіршитися
Використовувані методи	RLE, Huffman, LZW	DCT, компенсація руху, квантування
Застосування	Архівування, монтаж	Потокове відео, телебачення
Переваги	Точність	Ефективне стиснення
Недоліки	Великий обсяг даних	Можлива втрата якості

Таким чином, вибір між стисненням без втрат та з втратами залежить від конкретного випадку використання: стиснення без втрат підходить для критичних даних, тоді як стиснення з втратами дозволяє ефективно працювати з великими відеопотоками в умовах обмежених ресурсів.

Пропонується вдосконалити стандарт MPEG шляхом введення в його стандартний алгоритм нових методів для подальшого зменшення обчислювальних потужностей.

2.2 Технічний аналіз найпоширеніших алгоритмів стиснення: MPEG-4, H.264, HEVC

Сучасні алгоритми стиснення відеоданих, такі як MPEG-4, H.264 і HEVC, стали основними технологіями для ефективної передачі та зберігання відео. Кожен із цих стандартів використовує унікальні підходи до обробки даних, що дозволяє оптимізувати розмір файлів без помітної втрати якості.

Їхні відмінності, однак, є важливими для розуміння того, як ці алгоритми адаптуються до сучасних потреб користувачів, від мобільних

додатків до 4К- та 8К-контенту [10-12].

MPEG-4 (Moving Picture Experts Group-4) був розроблений як універсальний стандарт, що охоплює не тільки відео, але й аудіо, текст і графіку. Основна мета MPEG-4 — це ефективне стиснення мультимедійних даних із можливістю мінімальних втрат якості, що дозволяє зменшити обсяг файлів. MPEG-4 реалізує систему кадрів типів I, P і B, що кодуються з різною мірою залежності. I-кадри містять повну інформацію про зображення і є незалежними, тоді як P-кадри і B-кадри використовують дані попередніх і наступних кадрів, що значно зменшує обсяг інформації для кожного кадру. Компенсація руху — один із важливих інструментів MPEG-4, де відмінності між послідовними кадрами використовуються для зменшення даних.

Розглянемо загальний вигляд стандарту на рисунку 2.2.

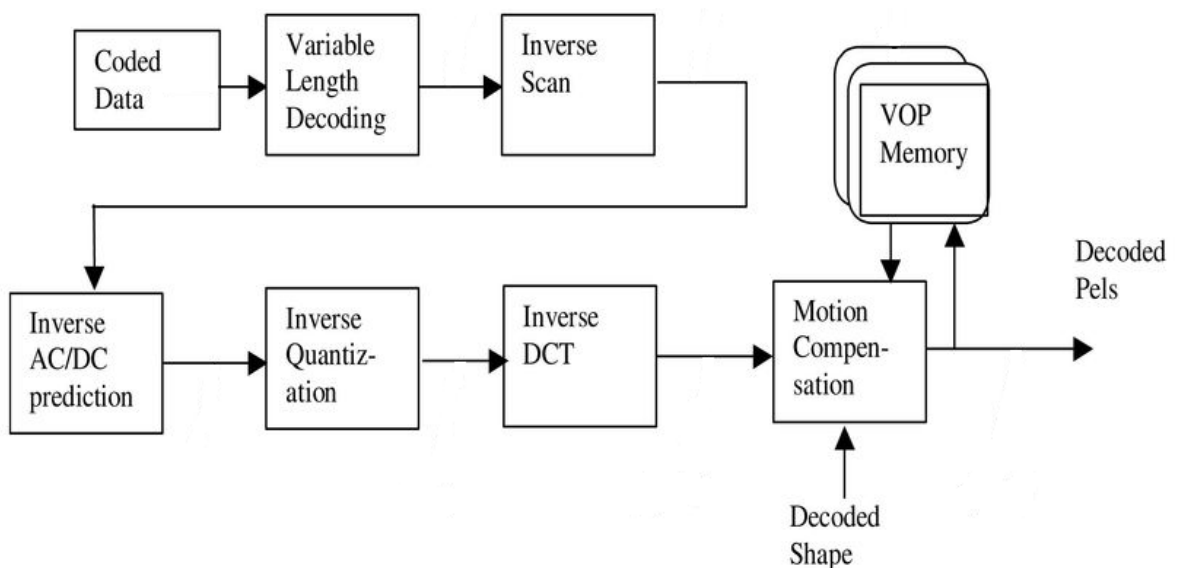


Рисунок 2.2 – Алгоритм стандарту MPEG-4

Як можемо побачити квантування у MPEG-4 базується на дискретному косинусному перетворенні (DCT), яке переводить зображення в частотну область, що дозволяє видаляти деталі, менш помітні для людського ока. Окрім того, MPEG-4 має об'єктно-орієнтоване кодування, яке забезпечує незалежне кодування різних об'єктів сцени (наприклад, рухомого об'єкта і фону).

H.264 (MPEG-4 Part 10, Advanced Video Coding), або AVC, став основним стандартом для відео високої чіткості завдяки підвищеній ефективності стиснення та підтримці широкого спектра пристроїв і платформ. Цей стандарт використовує більш складну структуру кадрів, де В-кадри ефективніше використовуються для передбачення і компенсації руху, дозволяючи кодувати навіть найдрібніші зміни з більшою точністю. Блоки розміром 4x4 пікселів у H.264 дозволяють значно підвищити точність аналізу руху. Це робить H.264 популярним вибором для відео з високою деталізацією, оскільки дозволяє зберігати дрібні деталі руху.

Ще одна перевага H.264 — це адаптивне квантування, що забезпечує стиснення різних частин кадру з урахуванням їх важливості для якості зображення. Наприклад, обличчя та інші важливі частини зображення можуть кодуватися з меншою втратою якості, тоді як фон або незначні ділянки — з більшим стисненням. Важливою особливістю H.264 є підтримка внутрішньокадрового передбачення, що знижує обсяг даних навіть у статичних кадрах.

HEVC (High Efficiency Video Coding) [10-12] або H.265, є наступним поколінням стандартів після H.264 і відзначається підвищеною ефективністю стиснення, особливо для відео з надвисокою роздільною здатністю. HEVC був розроблений з акцентом на підтримку 4K і 8K відео, що забезпечує майже вдвічі більшу ефективність стиснення в порівнянні з H.264, зберігаючи високу якість зображення при низькому бітрейті. Однією з основних новацій HEVC є використання великих блоків кодування (CTU), які дозволяють обробляти ділянки з однаковими характеристиками більш ефективно, що підходить для сцен з великим фоном.

HEVC використовує передбачення руху з високою точністю (до 1/4 або 1/8 пікселя), що дозволяє краще стискати динамічні сцени з рухомими об'єктами. Внутрішньокадрове передбачення в HEVC є більш гнучким і дозволяє вибирати різні напрямки для прогнозування пікселів. Цей стандарт також реалізує ентропійне кодування Context-Adaptive Binary Arithmetic

Coding (SAVAS), яке забезпечує вищу ефективність у порівнянні з попередниками, що робить його ідеальним для сучасних потокових сервісів і трансляцій високої чіткості.

Таким чином, кожен із розглянутих стандартів — MPEG-4, H.264 і HEVC — має свої сильні сторони і межі застосування. MPEG-4 підходить для базових завдань і роботи з обмеженими ресурсами, H.264 став еталоном для HD-контенту завдяки ефективному стисненню і високій якості, тоді як HEVC відповідає на вимоги нової ери надвисокої чіткості 4K і 8K. Основні відмінності та характеристики цих стандартів представлені у порівняльній таблиці 2.2.

Таблиця 2.2 - Порівняння MPEG-4, H.264, HEVC

Характеристика	MPEG-4	H.264	HEVC
Розмір блоків кодування	8x8	4x4 до 16x16	16x16 до 64x64
Компенсація руху	Піксельна	Підпіксельна	До 1/8 пікселя
Алгоритм перетворення	DCT	Integer DCT	Перемінна довжина DCT
Підтримка HD та UHD	Ні	Так	Так
Коефіцієнт стиснення	Низький	Високий	Дуже високий
Застосування	Веб-відео, архіви	Потокове відео, HDTV	4K/8K відео, потокові сервіси

Кожен із цих стандартів має свою область застосування і переваги. MPEG-4 залишається популярним для простих завдань, H.264 — для стрімінгових сервісів та телебачення, а HEVC забезпечує майбутнє з високоякісними відео у форматах 4K і 8K.

2.3 Порівняльний аналіз ефективності різних методів стосовно специфікацій відео

Відеоконтент буває різноманітним за своїми характеристиками, від коротких кліпів для соціальних мереж до фільмів у форматі 4K і 8K, що вимагає врахування специфікацій відео при виборі оптимального методу стиснення.

Кожен із сучасних стандартів — MPEG-4, H.264 і HEVC — має свої переваги та обмеження залежно від конкретних специфікацій, включаючи роздільну здатність, тип контенту (наприклад, спортивні події, кіно, анімація) і доступні мережеві ресурси[11].

MPEG-4 залишається стандартом вибору для контенту з низькою та середньою роздільною здатністю, зокрема для мобільного відео і веб-відео, де економія ресурсів має вирішальне значення. Він підходить для передачі відео в умовах обмеженої пропускної здатності, забезпечуючи базовий рівень якості при мінімальних ресурсах. Проте, для відео з високою роздільною здатністю (HD і вище) ефективність MPEG-4 значно знижується, оскільки він не здатний оптимально компресувати дані при високих вимогах до якості.

H.264, розроблений для покращеної компресії відео з високою роздільною здатністю, став провідним стандартом для HD- та Full HD-контенту. Його алгоритми дозволяють передавати контент із високою деталізацією навіть при середній пропускній здатності мережі. H.264 є ідеальним вибором для контенту, що включає багато руху, наприклад, для спортивних трансляцій чи динамічних сцен у фільмах, де важливо зберігати чіткість руху. Проте для відео 4K і вище H.264 вимагає значних ресурсів і не завжди є найефективнішим рішенням.

HEVC (H.265), націлений на роботу з відео високої і надвисокої роздільної здатності (4K, 8K), є вибором для сучасного стрімінгу та телебачення ультрависокої чіткості. Цей стандарт забезпечує подвійне стиснення порівняно з H.264, зберігаючи високу якість навіть при зниженому

бітрейті, що робить його підходящим для потокового відео 4K і 8K у мережах з обмеженою пропускною здатністю. Проте складність HEVC створює додаткові вимоги до обчислювальної потужності, що обмежує його використання на менш потужних пристроях, таких як мобільні телефони та планшети.

Для наочного порівняння ефективності MPEG-4, H.264 та HEVC стосовно специфікацій відео представлено таблицю 2.3, розглянемо різні роздільні здатності та різні сцени зйомки.

Таблиця 2.3 - Порівняння ефективності MPEG-4, H.264 та HEVC стосовно специфікацій відео

Специфікація відео	MPEG-4	H.264	HEVC (H.265)
1	2	3	4
Низька роздільна здатність (480p)	Ефективний для низького бітрейту та обмежених ресурсів; підходить для мобільних і веб-додатків	Підходить, але використовує більше ресурсів, ніж MPEG-4	Надлишковий, особливо для простих, статичних сцен
Середня роздільна здатність (720p, HD)	Прийнятний рівень стиснення, але з помітною втратою якості для динамічних сцен	Оптимальний баланс якості і стиснення, особливо для спортивних подій і кіно	Підходить для контенту з деталями, проте потребує більше ресурсів

Продовження таблиці 2.3

1	2	3	4
Висока роздільна здатність (1080p, Full HD)	Втрачає ефективність через підвищені вимоги до пропускнуої здатності	Оптимальний для стрімінгу та трансляцій високої чіткості	Висока якість, особливо при низькому бітрейті, але обчислювально складний
Ультрависока роздільна здатність (4K, 8K)	Вкрай неефективний, вимагає значного бітрейту	Може використовуватись для 4K, але значно підвищує навантаження	Ідеальний для 4K+ відео; мінімізує бітрейт, зберігаючи якість
Динамічні сцени (спорт, кіно)	Не зберігає чіткість руху при низькому бітрейті	Забезпечує високу якість при середньому бітрейті	Максимально ефективний для високої деталізації руху
Статичні сцени (презентації, навчальні відео)	Висока ефективність при низькому бітрейті	Забезпечує якість, але потребує більше ресурсів	Надлишковий для статичного контенту, але зберігає максимальну якість
Пропускна здатність мережі	Низькі вимоги, оптимальний для нестабільних мереж	Середні вимоги, підходить для HD-стрімінгу в стабільних мережах	Високі вимоги, оптимальний для мереж з високою пропускнуою здатністю

З таблиці видно, що ідеального універсального методу компресії для

всіх типів контенту не існує. MPEG-4 добре підходить для відео з низькою роздільною здатністю і статичних сцен, де потрібне економне використання ресурсів. H.264 демонструє оптимальні результати для HD-відео і є популярним стандартом для стрімінгу контенту, що потребує високої якості передачі руху. HEVC же стає найбільш ефективним для відео з високою роздільною здатністю, як-от 4K і 8K, де потрібно мінімізувати бітрейт, зберігаючи при цьому максимальну якість.

Таким чином, ефективність кожного методу стиснення значною мірою залежить від типу відеоконтенту та технічних умов, у яких він буде використовуватися. Це підкреслює важливість вибору алгоритму стиснення, що враховує специфікації відео для досягнення найкращого результату при мінімальних витратах ресурсів [12,13].

2.4 Огляд технологій програмування для вирішення поставленої задачі

Для ефективного стиснення відеоданих та забезпечення високої якості передачі відео існує широкий спектр технологій програмування, які дозволяють налаштовувати алгоритми під специфічні потреби і вимоги. У вирішенні задачі стиснення відео сучасні технології спрямовані на адаптивне управління якістю, мінімізацію затримок і оптимізацію бітрейту з урахуванням технічних можливостей пристроїв і обмежень мережі.

Одним із основних інструментів є мови програмування, такі як C++ і Python, що активно використовуються для реалізації алгоритмів стиснення завдяки своїй гнучкості та продуктивності. C++ дає змогу працювати з пам'яттю на низькому рівні, що особливо важливо для ресурсомістких завдань, як-от обробка великих обсягів відеоданих. Python, завдяки своїм бібліотекам, спрощує розробку і тестування алгоритмів, що дозволяє швидко впроваджувати прототипи рішень і вдосконалювати їх під час тестування. Використання таких бібліотек, як OpenCV для обробки зображень і відео, дає розробникам можливість застосовувати різноманітні методи обробки кадрів і

керувати стисненням на базовому рівні.

Для роботи з відеопотоками на рівні мережі широко використовуються FFmpeg і GStreamer. FFmpeg — потужна платформа для кодування, декодування і трансформації відео, яка підтримує всі основні кодеки та стандарти, як-от MPEG-4, H.264, HEVC. З її допомогою можна налаштовувати бітрейт і змінювати параметри відеопотоку в реальному часі. GStreamer, у свою чергу, пропонує зручний фреймворк для роботи з мультимедіа в режимі реального часу, що робить його незамінним інструментом для потокових додатків і відеоконференцій..

Значний прогрес у відеокомпресії досягнуто завдяки штучному інтелекту та машинному навчанню. Бібліотеки, такі як TensorFlow і PyTorch, надають широкий інструментарій для розробки нейронних мереж, які можуть адаптувати параметри стиснення відповідно до контексту відео. Інтелектуальні алгоритми дозволяють передбачити зміни в об'єктах сцени і заздалегідь налаштувати стиснення для зниження бітрейту без помітної втрати якості. Це стає особливо корисним для відео з динамічними сценами, де зміни у русі потребують більш точного контролю.

Для підтримки роботи додатків на різних платформах важливими є фреймворки для мобільної розробки, такі як Android SDK та iOS SDK. Ці інструменти дозволяють адаптувати відеоконтент до можливостей мобільних пристроїв, зокрема працювати з обмеженими ресурсами, забезпечувати плавність відтворення і економію енергії. Зокрема, Android підтримує апаратне прискорення для кодеків H.264 і HEVC, що допомагає зменшити затримки і підвищити продуктивність[14-16].

Отже, технології програмування для стиснення відео охоплюють цілий набір потужних інструментів, від базових мов і бібліотек до спеціалізованих фреймворків і хмарних сервісів, що забезпечують адаптивність, продуктивність і якість у сучасних умовах роботи з відеоданими.

3 МОДЕРНІЗАЦІЯ ІСНУЮЧОГО СТАНДАРТУ ВІДЕОКОМПРЕСІЇ

3.1 Модернізація підвищення ефективності функціонування систем шляхом інтегрування адаптивних методів

Процес стиснення відео залишається одним із найважливіших викликів у сучасних інформаційних технологіях, особливо в умовах безперервного зростання обсягів відеоданих та вимог до їх якості. Стандарт MPEG-4, як один із найбільш поширених алгоритмів стиснення, хоч і демонструє високу ефективність, має низку обмежень, які стають критичними в умовах змінних параметрів мережі, високої роздільної здатності та реального часу передачі відео. Основні слабкі місця включають рівномірну обробку макроблоків без урахування їхньої інформативності, а також недостатню оптимізацію процесів кодування і квантування.

Спираючись на ці обмеження, у роботі пропонується вдосконалений метод на основі MPEG-4, що інтегрує адаптивні підходи до сегментації кадрів, компенсації руху та кодування. Запропонована структурно-функціональна схема вдосконаленого алгоритму (рисунок 3.1) містить нові елементи, спрямовані на покращення розподілу обчислювальних ресурсів та зменшення втрат якості. Відмінність розробки полягає у використанні адаптивних алгоритмів, таких як сегментація за середньою довжиною змін, адаптивне квантування та інтеграція арифметичного кодування.

Ці вдосконалення спрямовані на створення оптимального балансу між якістю відео та обсягом даних, що передаються, забезпечуючи ефективність роботи навіть у складних умовах телекомунікаційних мереж.

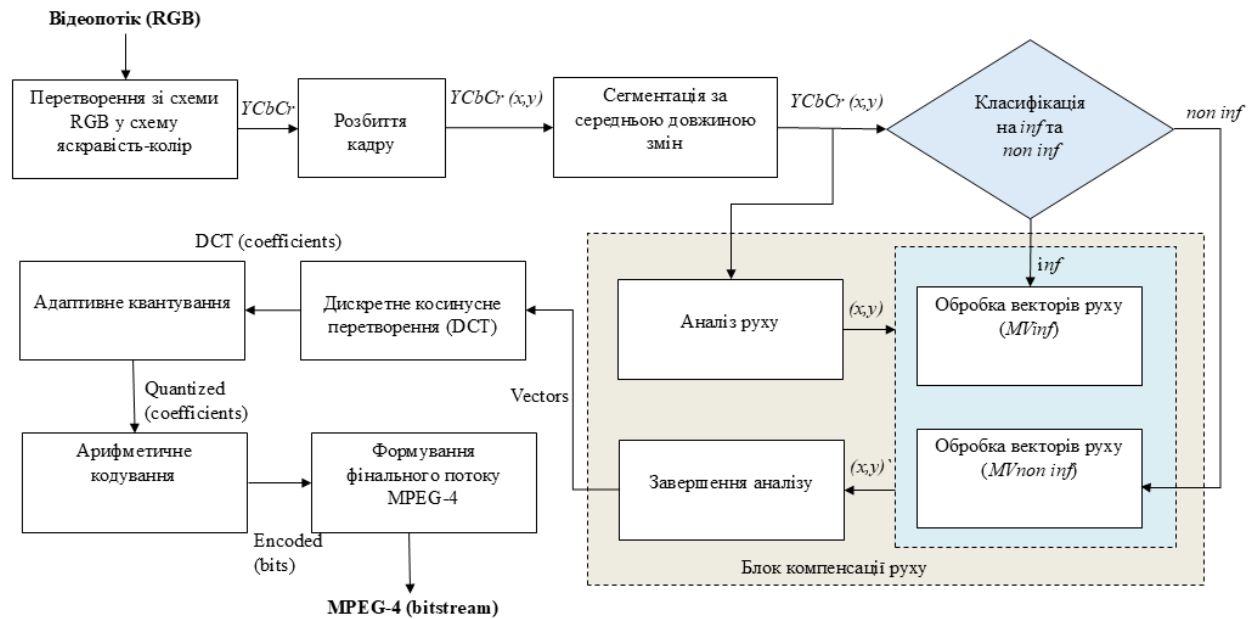


Рисунок 3.1 – Структурно-функціональна схема вдосконаленого MPEG-4

Запропонована структурно-функціональна схема демонструє тісну взаємодію всіх компонентів алгоритму, кожен з яких виконує специфічну функцію, спрямовану на покращення процесу стиснення відео. Сегментація кадру на інформативні та неінформативні блоки забезпечує оптимальний розподіл обчислювальних ресурсів. Блок компенсації руху зосереджується на аналізі динамічних змін у відеопотоці, дозволяючи мінімізувати обсяг переданих даних без втрати важливих деталей. Етапи дискретного косинусного перетворення (DCT) і квантування адаптивно обробляють інформативні блоки, зберігаючи їхню якість, тоді як менш значущі ділянки кадру стискаються з більшим ступенем компресії. Завершальний етап – ентропійне кодування, яке у новій версії алгоритму реалізоване через арифметичне кодування, – забезпечує максимально компактне представлення даних для їх передачі.

Взаємодія цих блоків у рамках вдосконаленого алгоритму дозволяє досягти оптимального балансу між якістю стиснення та ефективністю використання мережевих і обчислювальних ресурсів.

3.2 Інтеграція сегментації кадру з пошуком важливої частки відео

Новизна запропонованого методу починається з етапу розбиття відеопотоку на кадри та блоки, що створює базову структуру для подальшої обробки. Поділ кадрів на макроблоки дозволяє локалізувати обчислення, зменшуючи витрати ресурсів на ділянках, які мають менший вплив на якість відео. У традиційному підході MPEG-4 всі макроблоки обробляються однаково, що призводить до нераціонального використання обчислювальних потужностей.

Запропонований адаптивний алгоритм сегментації за середньою довжиною змін дозволяє визначати інформативність блоків (рисунок 3.2), зосереджуючи ресурси на найбільш важливих зонах кадру. Це забезпечує збереження високої деталізації в ключових областях і одночасно скорочує обсяг обробки для менш важливих ділянок, що є критично важливим для досягнення балансу між якістю відео та ефективністю стиснення.

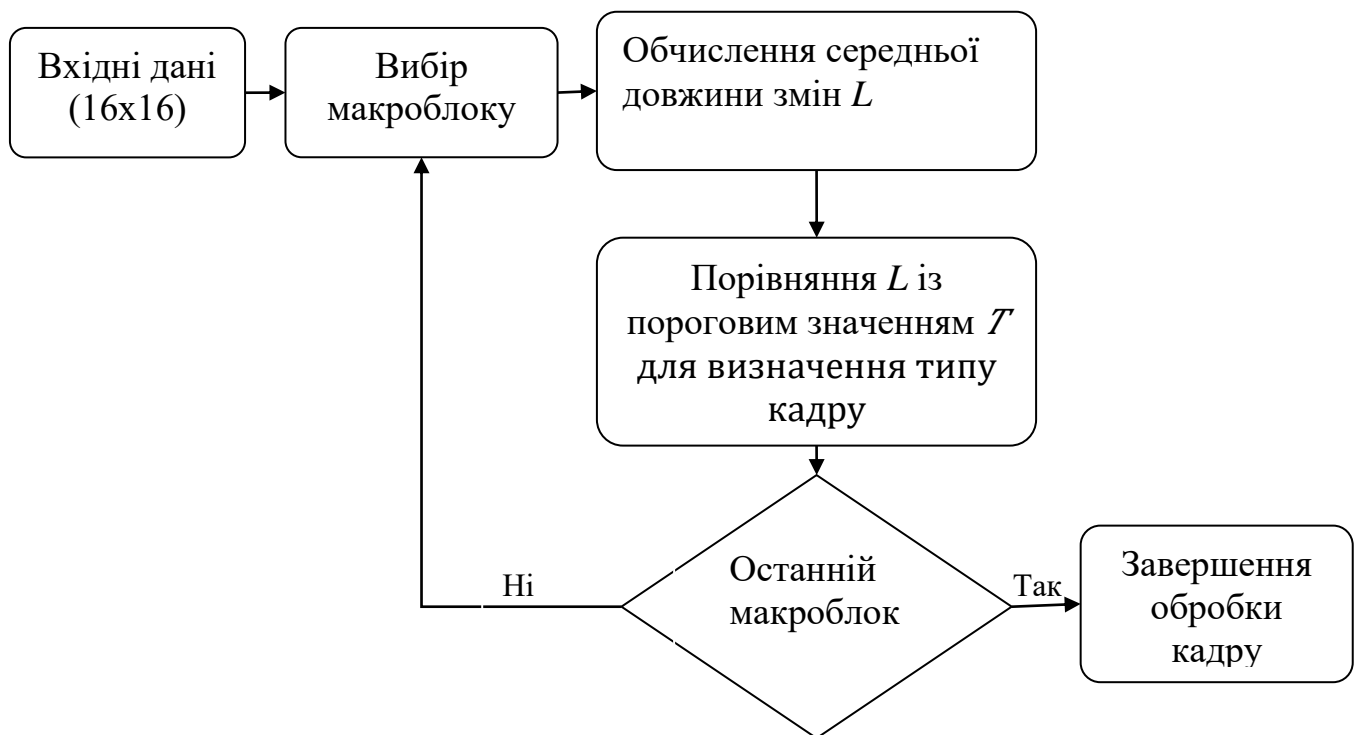


Рисунок 3.2 – Блок-схема сегментації за середньою довжиною

У межах кожного макроблоку аналізуємо зміни яскравості або кольору

між сусідніми пікселями. Якщо блок містить складну текстуру, контури або дрібні об'єкти, зміни будуть більшими, і середня довжина змін матиме високе значення. Для більш однорідних зон, таких як фон або рівні поверхні, середня довжина змін буде невеликою. Це дозволяє відрізнити важливі для сприйняття області від тих, які можуть бути оброблені з меншою точністю.

Середня довжина змін у блоці розміром $N \times N$ обчислюється за формулою:

$$L = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^{N-1} |P_{i,j} - P_{i,j+1}| + \sum_{i=1}^{N-1} \sum_{j=1}^N |P_{i,j} - P_{i+1,j}|,$$

де $P_{i,j}$ — значення яскравості пікселя у позиції (i, j)

Після розрахунку L блоки класифікуються за пороговим значенням T :

- якщо $L > T$, блок вважається інформативним;
- якщо $L \leq T$, блок класифікується як неінформативний.

Тож, можемо зробити деякі висновки з цього розділу. Інтеграція класифікації за середньою довжиною змін відкриває нові можливості для оптимізації обробки. Для інформативних блоків, які містять об'єкти з чіткими контурами чи дрібними деталями (обличчя, текст, динамічні об'єкти), можна зберегти максимальну точність. Неінформативні блоки, такі як однотонний фон чи статичні ділянки, обробляються з мінімальними витратами, що знижує обсяг даних.

Класичний підхід MPEG-4 обробляє всі блоки однаково, що часто призводить до неефективного використання ресурсів. Наприклад, у відео з великими статичними ділянками, як-от небо чи фон, значна частина ресурсів витрачається на обробку зон, які не впливають на сприйняття глядачем. Інтеграція сегментації за середньою довжиною змін дозволяє зосереджувати обробку на ключових об'єктах, таких як обличчя, транспорт чи інші динамічні елементи.

Це особливо важливо для застосувань, де якість відео і швидкість

передачі є критичними, як-от потокові сервіси або системи відеоспостереження. Наприклад, у камерах спостереження фон часто не має значення, тоді як об'єкти, що рухаються, повинні бути чіткими. Новий підхід зменшує обсяг даних, що передаються, одночасно підвищуючи якість важливих зон.

Сегментація за середньою довжиною змін додає адаптивності до процесу стиснення MPEG-4, дозволяючи обробляти відео з більшою ефективністю. Цей підхід забезпечує як оптимізацію ресурсів, так і збереження високої якості для критично важливих ділянок кадру. Це вдосконалення робить стандарт більш гнучким і придатним для сучасних умов, де важливі одночасно висока якість і економія ресурсів.

3.3 Опис модифікації етапу компенсації руху

На наступному етапі, після класифікації блоків на інформативні та неінформативні, ми пропонуємо вдосконалити процес компенсації руху, зробивши його адаптивним. У класичному підході MPEG-4 компенсація руху зосереджується на всіх макроблоках однаково, що може бути неефективним, особливо у сценах із великими статичними областями. Ми пропонуємо інтегрувати адаптивний аналіз руху, який враховує інформацію про інформативність блоків. Це дозволить зосередити обчислювальні ресурси на ключових областях кадру, залишаючи менш значущі зони з мінімальною обробкою.

Компенсація руху у MPEG-4 полягає у визначенні змін між кадрами. Система створює вектори руху для кожного макроблоку, які описують, як цей блок перемістився у просторі. Але замість однакового підходу до всіх блоків ми пропонуємо враховувати інформативність.

Для інформативних блоків, таких як об'єкти, що рухаються (наприклад, автомобіль, який змінює положення в кадрі), компенсація руху повинна бути максимально точною. Це означає, що система має виконувати більш

ретельний пошук відповідностей між блоками у поточному і попередньому кадрах. Для цього можна використовувати метрику середньоквадратичної різниці (MSE), яка обчислюється за формулою:

$$MSE = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (B_{i,j} - R_{i,j})^2,$$

де $B_{i,j}$ — піксель у поточному макроблоці; $R_{i,j}$ — піксель у порівнюваній області попереднього кадру; N — кількість пікселів у блоці.

Для неінформативних блоків, таких як статичний фон, пошук векторів руху може бути спрощений або взагалі пропущений, якщо блок залишається незмінним. Наприклад, для великих однотонних областей (небо чи стіна) система може зберігати їх без змін, зменшуючи витрати на обчислення.

Процес можна описати наступним чином:

Для кожного макроблоку визначається його інформативність на основі сегментації за середньою довжиною.

Інформативні блоки обробляються з високою точністю:

- виконується повний пошук найкращого співпадіння за метрикою MSE;
- вектор руху визначається для кожного блоку.

Неінформативні блоки отримують спрощену обробку:

- якщо зміни між кадрами незначні, вектор руху може бути встановлений як нульовий;
- якщо зміни присутні, застосовується менш точний пошук, щоб зекономити ресурси.

Тож маємо оптимізацію обчислень. Для статичних блоків, які не впливають на якість зображення, зменшуються витрати на обчислення, що прискорює загальний процес кодування.

Підвищення якості. Інформативні блоки отримують максимальну точність обробки, що дозволяє зберігати деталі у рухомих об'єктах.

Зменшення обсягу даних. Завдяки точному визначенню векторів руху

скорочується обсяг переданих змін між кадрами.

У класичному підході компенсація руху може витрачати ресурси на обробку великих статичних зон, таких як фон, який не змінюється між кадрами. Це не лише знижує ефективність, але й обмежує точність обробки рухомих об'єктів. Інтеграція адаптивного аналізу руху дозволяє вирішити цю проблему, концентруючи ресурси там, де вони потрібні найбільше.

Наприклад, у відео з камер спостереження основний інтерес зосереджений на рухомих об'єктах — автомобілях чи людях. Використання адаптивного підходу дозволяє обробляти ці об'єкти з високою точністю, тоді як фон буде стискатися з мінімальними витратами ресурсів. Це також знижує енергоспоживання, що особливо важливо для мобільних пристроїв чи систем із низьким рівнем енергії.

3.4 Адаптація DCT та квантування для покращення методу

Наступним етапом, на якому ми пропонуємо інтеграцію адаптивних методів, є процес перетворення відео в частотну область через дискретне косинусне перетворення (DCT) і наступне квантування. Ці два етапи є тісно пов'язаними, адже після того, як DCT перетворює зображення на частотні компоненти, саме квантування визначає, які з цих компонент будуть збережені з високою точністю, а які можна втратити, щоб зменшити обсяг даних. Однак, як і в попередніх етапах, ми можемо вдосконалити цей процес, зробивши його адаптивним і орієнтованим на інформативність блоків, яку ми визначили на попередньому етапі.

Процес дискретного косинусного перетворення (DCT) є надзвичайно важливим для стиснення відео, оскільки він дозволяє перетворити зображення з просторової області у частотну. У частотній області зображення можна представити як суму різних компонентів: високочастотних (дрібні деталі та текстури) і низькочастотних (основні форми та кольори). Ключовим моментом є те, що людське око більш чутливе

до низьких частот, а до високих частот — значно менш чутливе. Тому більшість інформації у відео зосереджена в низькочастотних компонентах, а високочастотні компоненти можуть бути значною мірою відкинуті без помітної втрати якості.

Традиційно в MPEG-4 кожен макроблок перетворюється за допомогою DCT, і всі компоненти цього блоку обробляються однаково. Проте, якщо ми адаптуємо цей процес і зосередимося **на** інформативних блоках, ми зможемо зберігати більш точні значення для низькочастотних компонентів у важливих областях і відкидати високочастотні компоненти в незначущих зонах.

Наприклад, якщо на кадрі є важливі деталі, як-от обличчя або текст, ці області мають велику кількість низькочастотних компонентів, які відповідають за форму і контури. У такому випадку ці блоки повинні бути оброблені з максимальною точністю на етапі DCT, а менш інформативні блоки фону, наприклад, де домінують рівні кольори і текстури, можуть бути спрощені. У цьому випадку використовуються адаптивні таблиці DCT, де для інформативних блоків використовуються детальніші таблиці з більш високими точностями, а для неінформативних блоків — простіші таблиці, які дозволяють знизити точність.

Після того, як макроблоки перетворюються в частотну область, наступним кроком є квантування. Квантування відповідає за округлення частотних компонентів до найближчих дозволених значень, що дозволяє значно зменшити кількість бітів, необхідних для збереження зображення. Однак, як і у випадку з DCT, класичний підхід квантує всі блоки однаково, що може бути неефективно для зображень з високим ступенем інформативності.

Ми пропонуємо інтегрувати диференціальне квантування, яке передбачає різні стратегії квантування для інформативних і неінформативних блоків. Для інформативних блоків, де важливо зберегти деталі, можна застосовувати мінімальне квантування, тобто зберігати більше інформації і забезпечувати високу точність. Для неінформативних блоків, таких як фон

або монотонні області, де зміни між кадрами незначні, можна використовувати вищий крок квантування, що дозволить знизити точність і досягти більшого стиснення.

Формула для адаптивного квантування виглядатиме наступним чином:

$$Q_{adapt} = Q_0 \times f(D),$$

де Q_0 — стандартне значення кроку квантування для загального випадку; $f(D)$ — функція, що визначає ступінь стиснення на основі інформативності блоку, де D — дисперсія блоку. Для інформативних блоків $f(D) \approx 1$, а для неінформативних блоків $f(D) \gg 1$.

Цей підхід дозволяє зберегти якість у критичних для сприйняття областях кадру, одночасно зменшуючи обсяг даних у менш важливих ділянках.

Тож маємо такі покращення:

- інформативні блоки стискаються з високою точністю, зберігаючи важливі деталі, а неінформативні блоки — з більшим кроком, що дозволяє значно зменшити розмір файлу;
- адаптивний підхід дозволяє зменшити витрати на обробку для незначущих областей, зберігаючи потужність для більш важливих частин кадру;
- оскільки важливі деталі зберігаються з максимальною точністю, якість відео залишається на високому рівні, навіть при значному стисненні.

Інтеграція адаптивного DCT та квантування дозволяє ефективніше обробляти відео, зберігаючи високу якість у важливих областях, одночасно зменшуючи розмір файлу. Це покращує стандарт MPEG-4, роблячи його ще більш ефективним для відео з високою деталізацією та складними сценами. Наприклад, у відео з великою кількістю руху або з динамічними змінами ці методи дозволяють зберігати чіткість і точність в рухомих об'єктах, таких як

люди чи транспорт, при значному зменшенні обсягу даних для фону.

Це робить MPEG-4 ідеальним для широкого спектра застосувань, від відеоконференцій до потокових сервісів, де баланс між якістю та ефективністю є вирішальним.

3.5 Інтеграція арифметичного кодування до методу

Наступним кроком у вдосконаленні стандарту MPEG-4 є заміна традиційного кодування Хаффмана на арифметичне кодування (рисунок 3.3). Цей підхід дозволяє ефективніше стискати відео за рахунок більш компактного представлення даних, особливо у випадках з нерівномірними ймовірностями символів. У порівнянні з кодуванням Хаффмана, арифметичне кодування забезпечує вищу ефективність, наближаючись до теоретичної межі ентропійного стиснення.

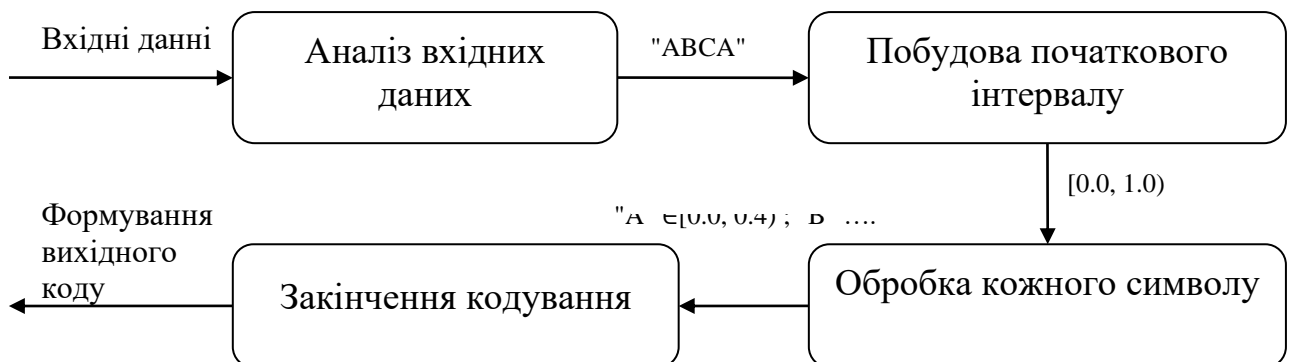


Рисунок 3.3 – Блок-схема арифметичного кодування

Спочатку пропонується розглянути метод Хаффмана, який використовується у методі MPEG-4 до нашої модернізації. Він використовує принцип побудови кодів на основі ймовірностей символів, при цьому кожен символ отримує певний код фіксованої довжини. Цей підхід є досить ефективним, але має обмеження в разі, коли ймовірності символів сильно варіюються. Метод Хаффмана дозволяє досягти оптимального стиснення тільки в специфічних випадках, коли символи мають наближено рівні

ймовірності появи.

У той час як арифметичне кодування пропонує представляти послідовність символів як дробове число між 0 і 1. Це дозволяє використовувати змінні інтервали для кожного символу в залежності від його ймовірності, що дає більшу гнучкість та більш ефективне стиснення, особливо в умовах нерівномірних ймовірностей. Арифметичне кодування дозволяє досягти кращих результатів, наближаючись до теоретичної межі ентропійного стиснення.

Для порівняння, будемо використовувати текст: "ХНУРЕ- перший серед кращих". Перш за все, давайте визначимо частоти появи символів у цьому тексті:

- Х: 1/27;
- Н: 2/27;
- У: 2/27;
- Р: 3/27;
- Є: 1/27;
- і так далі.

Після цього, ми можемо закодувати це повідомлення за допомогою арифметичного кодування.

Початковий інтервал $[0, 1)$ визначається для всіх символів.

Для кожного символу, в залежності від його ймовірності, визначається підінтервал. Наприклад:

- Х: $[0, 0.03704)$;
- Н: $[0.03704, 0.14814)$;
- У: $[0.14814, 0.25926)$;
- Р: $[0.25926, 0.44444)$;
- і так далі.

Далі ми по черзі будемо уточнювати інтервали для кожного символу в послідовності, поки не обробимо весь текст.

Якщо взяти перші кілька символів:

- "X" буде знаходитися в діапазоні [0, 0.03704);
- "H" в діапазоні [0.03704, 0.14814);
- "U" в діапазоні [0.14814, 0.25926);
- "P" в діапазоні [0.25926, 0.44444);
- і так далі.

Після обробки всіх символів, ми отримуємо кількість, що належить останньому інтервалу, яке і буде зашифрованим числом.

Результатом цього кодування буде число, що належить інтервалу, наприклад, 0.123456, яке можна використовувати для відновлення початкового тексту.

Метод Хаффмана для цього ж тексту призначить коди певної фіксованої довжини для кожного символу. Наприклад, у випадку, коли ймовірності символів нерівномірні, Хаффман буде використовувати кодовані слова з фіксованою довжиною, що в даному випадку може призвести до менш ефективного стиснення. Якщо у нас є символи з високою ймовірністю (наприклад, "E" або "P"), то метод Хаффмана буде призначати їм коротші коди, але при цьому загальна довжина коду для всіх символів не може бути меншою за певну мінімальну межу, що робить його менш ефективним для даного набору даних, в порівнянні з арифметичним кодуванням.

Арифметичне кодування дає більшу гнучкість і здатне адаптуватися до точних ймовірностей, що дозволяє значно зменшити розмір стисненого файлу.

Тож можемо зазначити такі переваги та недоліки арифметичного кодування:

- у кодуванні Хаффмана довжина коду для символів обмежена цілими числами, тоді як арифметичне кодування працює з дробовими ймовірностями, забезпечуючи більше стиснення;
- кодування Хаффмана вимагає наявності таблиці кодів, яка додається до метаданих. Арифметичне кодування обходиться без цього, оскільки ймовірності можуть бути відновлені з контексту;

- арифметичне кодування вимагає більше обчислювальних ресурсів через високу обчислювальну складність. Це може бути обмеженням для деяких систем із низькою продуктивністю;

- арифметичне кодування демонструє особливо високу ефективність для даних із великою різницею у ймовірностях символів. Наприклад, якщо один символ зустрічається у 90% випадків, воно забезпечить суттєво менший розмір файлу, ніж кодування Хаффмана.

В рамках цього розділу ми пропонували низку адаптивних методів для покращення стандарту MPEG-4, спрямованих на підвищення ефективності стиснення відео без значних втрат якості. Всі ці методи орієнтовані на оптимізацію процесів, які традиційно однаково застосовуються до всіх блоків кадрів, незалежно від їхньої значущості для сприйняття людиною. В основі наших пропозицій лежать адаптивні алгоритми, які дозволяють зосередити ресурси на обробці найбільш інформативних ділянок відео, знижуючи обсяг даних у менш важливих зонах. Ось як виглядає основна концепція нашого підходу:

- 1 адаптивне розбиття відео на кадри та макроблоки з використанням сегментації по середній довжині для визначення інформативності блоків. Цей підхід дозволяє класифікувати макроблоки на інформативні та неінформативні, що дозволяє застосовувати різні стратегії стиснення для кожної групи;

- 2 адаптивна компенсація руху. Завдяки цьому підходу ми зменшуємо обчислювальні ресурси для обробки статичних областей, зберігаючи точність аналізу руху в важливих ділянках кадру. Це дозволяє досягти високої точності для об'єктів, які рухаються, не витрачаючи ресурси на фонові зони;

- 3 адаптивне застосування DCT і квантування. Для інформативних блоків застосовується мінімальне квантування, що дозволяє зберегти більше деталей, а для неінформативних — використовується великий крок квантування для досягнення високого ступеня стиснення. Це дозволяє зберігати високу якість у критичних областях, таких як обличчя, текст чи

важливі об'єкти, і при цьому досягати великого стиснення для фону та менше важливих частин;

4 інтеграція арифметичного кодування. Це рішення дозволяє значно покращити ефективність стиснення завдяки більш точному використанню ймовірностей символів. Арифметичне кодування забезпечує компактніше представлення даних, мінімізує обсяг службової інформації і краще працює з нерівномірними наборами даних.

Всі ці методи разом створюють адаптивний підхід, який дозволяє знижувати обсяг даних при одночасному збереженні високої якості відео, що робить наш підхід ефективнішим порівняно з класичним MPEG-4.

АНАЛІЗ РЕЗУЛЬТАТІВ ПРОВЕДЕНИХ ЕКСПЕРИМЕНТІВ

4.1 Програмна реалізація сегментації за середньою довжиною

Для реалізації методу сегментації за середньою довжиною було обрано середовище Node.js, яке забезпечує зручну обробку зображень за допомогою бібліотеки 'canvas'. Вибір мови програмування JavaScript зумовлений її гнучкістю та потужними можливостями для роботи з зображеннями в контексті веб-технологій, а також доступністю необхідних інструментів для обробки та маніпулювання графічними даними.

Основною задачею було здійснення сегментації зображення за допомогою блоків розміру 16x16 пікселів, визначаючи середню довжину змін в кожному блоці, що дозволяє виділяти важливі або інформативні сегменти зображення. Ось як це реалізовано в нашій програмі.

Для початку зображення завантажується за допомогою функції loadImage, потім відображається на канвасі для подальшої обробки. Основним етапом є переведення кольорового зображення в градації сірого, що дозволяє спростити подальшу обробку. Для цього використовуємо функцію convertToGrayscale(лістинг 4.1), яка обчислює яскравість пікселів зображення.

Лістинг 4.1 – Конвертація у формат яскравість-колір (файл Method1.ts)

```
function convertToGrayscale(imageData: ImageData):
  Uint8ClampedArray {
  const { width, height, data } = imageData;
  const grayData = new Uint8ClampedArray(width * height);
  for (let i = 0; i < data.length; i += 4) {
    const r = data[i];
    const g = data[i + 1];
    const b = data[i + 2];
    const gray = Math.round(0.299 * r + 0.587 * g + 0.114 *
b);
    grayData[i / 4] = gray;
  }
  return grayData;
}
```

Наступним кроком є сегментація зображення (лістинг 4.2). Для цього використовуються блоки розміру 16x16 пікселів. Кожен блок обробляється для визначення середньої довжини змін, яка є мірою варіативності пікселів в середині блоку. Чим більша середня довжина, тим більш інформативним є блок.

Лістинг 4.2 – Конвертація у формат яскравість-колір (файл Method1.ts)

```
function segmentImage(
  grayData: Uint8ClampedArray,
  width: number,
  height: number
): Segment[] {
  const result: Segment[] = [];
  const avgLengths: number[] = [];

  for (let y = 0; y < height; y += BLOCK_SIZE) {
    for (let x = 0; x < width; x += BLOCK_SIZE) {
      const block = getBlock(grayData, width, x, y);
      const avgLength = calculateAverageLength(block);
      avgLengths.push(avgLength);
      result.push({ x, y, avgLength, isInformative: false
    });
  }
}

// Расчёт динамического порога
const threshold =
  avgLengths.reduce((sum, val) => sum + val, 0) /
  avgLengths.length;

// Классификация блоков по порогу
result.forEach((block) => {
  block.isInformative = block.avgLength > threshold;
});

return result;
}
```

В процесі сегментації також проводиться розрахунок динамічного порогу(T), що визначає, чи є блок інформативним, чи ні. Блоки з середньою довжиною, більшою за поріг, позначаються як інформативні. Це робиться для того, щоб у ході подальшого порівняння з іншими методами поріг встановлювався автоматично в залежності від самого зображення.

Розрахунок середньої довжини зміни виконує функція `calculateAverageLength` (лістинг 4.3).

Лістинг 4.3 – Відтворення результатів (файл `Method1.ts`)

```
function calculateAverageLength(block: number[]): number {
  let sum = 0;
  for (let i = 0; i < block.length - 1; i++) {
    sum += Math.abs(block[i] - block[i + 1]);
  }
  return sum / block.length;
}
```

Для візуалізації результатів сегментації використовується функція `drawResult`, яка накладає на зображення прямокутники, що виділяють інформативні та неінформативні блоки (лістинг 4.4). Інформативні блоки позначаються зеленим кольором, а неінформативні — червоним.

Лістинг 4.4 – Відтворення результатів (файл `Method1.ts`)

```
function drawResult(ctx:CanvasRenderingContext2D,
segments:Segment[]) {
  segments.forEach(({ x, y, isInformative }) => {
    if (!isInformative){
      ctx.strokeStyle = 'red';
      ctx.lineWidth = 2;
      ctx.strokeRect(x, y, BLOCK_SIZE, BLOCK_SIZE);
    }
  });
  segments.forEach(({ x, y, isInformative }) => {
    if (isInformative){
      ctx.strokeStyle = 'lightgreen';
      ctx.lineWidth = 2;
      ctx.strokeRect(x, y, BLOCK_SIZE, BLOCK_SIZE);
    }
  });
}
```

Цей процес дозволяє чітко визначити наскільки точно метод знаходить блоки які частини зображення несуть найбільше корисної інформації, що може бути використано для подальшої обробки або аналізу.

Таким чином, програма надає ефективний спосіб сегментації зображень на основі середньої довжини змін, що дозволяє виділяти значущі частини

зображення для подальшої обробки.

4.2 Реалізація додаткових методів для порівняння

Для подальшого порівняння з сегментацією за середньою довжиною пропонується реалізувати ще декілька методів, а саме, метод Хрящева та метод порогової різниці.

Метод Хрящева полягає у визначенні інформативності блоків зображення на основі дисперсії значень пікселів у кожному блоці. Якщо дисперсія блоку перевищує певний динамічно обчислений поріг, блок вважається інформативним.

Пропонується навести алгоритм методу, а потім його програмну реалізацію:

- для полегшення обробки зображення перетворюємо його у відтінки сірого, використовуючи стандартну формулу:

$$Gray = 0.299 \times R + 0.587 \times G + 0.144 \times B ,$$

це дає змогу зменшити кількість кольорів, з якими потрібно працювати;

- розбиваємо зображення на блоки розміром 16x16 пікселів. Для кожного блоку обчислюємо дисперсію його значень;

- для кожного блоку обчислюється дисперсія пікселів, що є мірою варіативності значень пікселів у блоці. Дисперсія визначається за формулою:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 ,$$

де x_i — значення пікселя, \bar{x} — середнє значення для блоку;

- порогове значення обчислюється як середнє значення всіх дисперсій, отриманих по всіх блоках. Це дозволяє адаптувати поріг до конкретного

зображення;

- блоки з дисперсією, що перевищує поріг, вважаються інформативними.

Програмна реалізація виконувалась у тій же середі що і сегментація за середньою довжиною (лістинг4.5).

Лістинг 4.5 – Реалізація методу Хрящева (файл Method2.ts)

```
// Сегментація зображення (Метод Хрящева)
function segmentImage(
  grayData: Uint8ClampedArray,
  width: number,
  height: number
): { x: number; y: number; dispersion: number; isInformative:
boolean }[] {
  const result = [];
  const dispersions: number[] = [];
  for (let y = 0; y < height; y += BLOCK_SIZE) {
    for (let x = 0; x < width; x += BLOCK_SIZE) {
      const block = getBlock(grayData, width, x, y);
      const dispersion = calculateDispersion(block);
      dispersions.push(dispersion);
      result.push({ x, y, dispersion, isInformative: false
});
    }
  }
  // Розрахунок динамічного порогу
  const threshold =
    dispersions.reduce((sum, val) => sum + val, 0) /
dispersions.length;
  console.log(`Рассчитанный порог: ${threshold}`);
  // Класифікація блоків за порогом
  result.forEach((block) => {
    block.isInformative = block.dispersion > threshold;
  });
  return result;
}
// Розрахунок дисперсії (Метод Хрящева)
function calculateDispersion(block: number[]): number {
  const mean = block.reduce((sum, val) => sum + val, 0) /
block.length;
  const dispersion = block.reduce((sum, val) => sum +
Math.pow(val - mean, 2), 0) / block.length;
  return dispersion;
}
```

Тепер перейдемо до реалізації сегментації за пороговою різницею.

Метод порогової різниці використовується для виявлення інформативних блоків зображення на основі максимальної різниці значень пікселів у кожному блоці. Ідея цього методу полягає в тому, щоб визначити, чи є різниця між найяскравішим і найтемнішим пікселем у блоці значною. Якщо різниця велика, то блок вважається інформативним.

Будемо знову виконувати пояснення у два кроки: алгоритм та програмна реалізація.

Алгоритм:

- подібно до методу Хрящева, ми починаємо з перетворення зображення в відтінки сірого для спрощення подальшої обробки;
- розбиваємо зображення на блоки розміром 16x16 пікселів, так само як і в попередньому методі;
- для кожного блоку обчислюється різниця між максимальним і мінімальним значенням пікселів у блоці. Це дає нам міру контрасту між пікселями блоку;
- порогове значення обчислюється як середнє значення всіх різниць для всіх блоків, що дозволяє адаптувати поріг до конкретного зображення;
- блоки з різницею, що перевищує поріг, вважаються інформативними.

Тепер пропонується програмно реалізувати цей метод (лістинг 4.6).

Лістинг 4.6 – Реалізація методу порогової різниці (файл Method3.ts)

```
// Сегментація зображення (Метод порогової різниці)
function segmentImage(
  grayData: Uint8ClampedArray,
  width: number,
  height: number
): { x: number; y: number; difference: number; isInformative:
boolean }[] {
  const result = [];
  const differences: number[] = [];

  for (let y = 0; y < height; y += BLOCK_SIZE) {
    for (let x = 0; x < width; x += BLOCK_SIZE) {
      const block = getBlock(grayData, width, x, y);
      const difference = calculateDifference(block);
      differences.push(difference);
      result.push({ x, y, difference, isInformative: false
```

```

});
    }
}

// Расчёт динамического порога
const threshold =
    differences.reduce((sum, val) => sum + val, 0) /
differences.length;
console.log(`Расчитанный порог: ${threshold}`);

// Классификация блоков по порогу
result.forEach((block) => {
    block.isInformative = block.difference > threshold;
});

return result;
}

// Расчёт разницы (Метод пороговой разницы)
function calculateDifference(block: number[]): number {
    const min = Math.min(...block);
    const max = Math.max(...block);
    return max - min;
}

```

Після цього можна почати модулювання та порівняння методів.

4.3 Моделювання та порівняння методів

У світі складності й упорядкування кожен фрагмент інформації несе в собі унікальну історію. Подібно до того, як філософ досліджує природу буття, ми ставимо перед собою завдання дослідити глибини відеоданих. Цей розділ присвячений моделюванню методів сегментації, розроблених у попередніх частинах, на прикладі трьох відеокадрів, які виступають у ролі символічних репрезентацій реального світу:

- кадр простий – мінімалізм і гармонія. Зображення з обмеженою кількістю деталей, де переважають великі однорідні області;
- кадр середньої складності – золота середина, баланс хаосу та порядку. Помірна кількість деталей, які вимагають зусиль для інтерпретації;
- кадр високої складності – як лабіринт думок, насичений дрібними деталями, текстурами й інформаційною глибиною.

Основна ідея цього експерименту полягає у вивченні того, як різні методи сегментації – за середньою довжиною, за методом Хрящева та за методом порогової різниці – справляються із завданням ідентифікації інформативних макроблоків у кожному з цих трьох випадків.

Через порівняння та осмислення результатів ми спробуємо знайти відповідь на вічне питання: чи є універсальний підхід до аналізу й сегментації, чи кожен кадр вимагає свого унікального інструментарію?

Пропонується почати з огляду кадрів які будуть представлені до аналізу. Кадр перший – простий, представляє собою зображення яке має багато однорідних областей і невелику інформативну частину (рисунок 4.1).



Рисунок 4.1– Перший кадр для аналізу

Як не дивовижно, усі методи гарно впоралися з цим зображенням і змогли коректно відобразити інформативні блоки(рисунок 4.2). Але вже тут можна побачити різницю в усіх методах, у той час як запропонований метод, та метод Хрящева намагаються коректно знайти інформативні блоки в середині зображення, яких там 90%, метод порогової різниці просто обводить усю пам'ятку як інформативну, з філософської точки зору, це звісно правильно, але якщо ми кажемо про коректність пошуку інформативних блоків, то результати говорять про більш коректну роботу сегментації за середньою довжиною(близько 98% достовірності). Також слід приділити увагу часу на виконання сегментації, в цьому компоненті сегментація за

середньою довжиною солідно попереду.

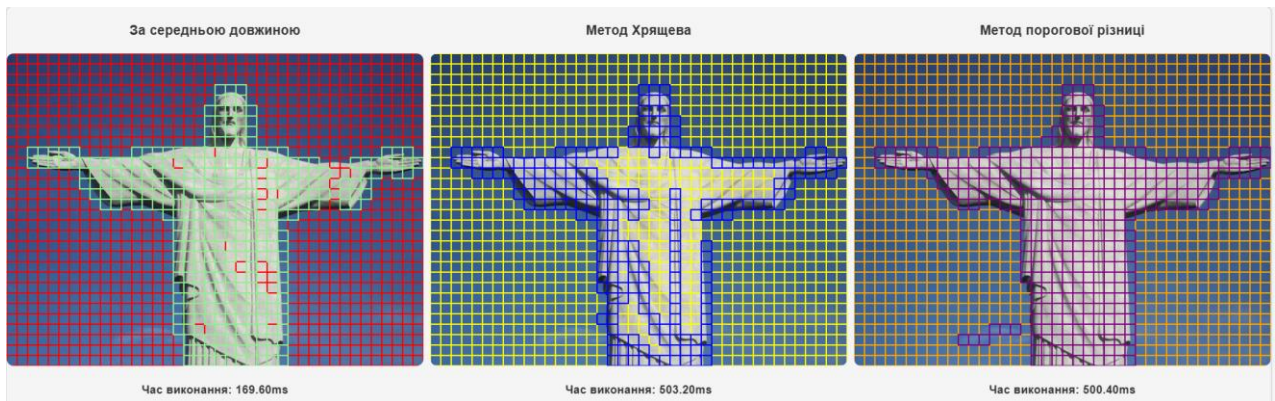


Рисунок 4.2 – Результати сегментації першого кадру для віднесення до \inf та $\text{non } \inf$

Другий кадр – несе в собі вже більшу кількість деталей, але все також присутні великі однорідні області(рисунок 4.3). Із цікавих випробовувань для методів сегментації, тут присутні захід сонця, кордони якого є інформативними, хмари, ліс, корабель і звісно велика кількість інформативних переходів на скелі.



Рисунок 4.3 – Другий кадр для аналізу

Результати вже не такі втішні як на першому кадрі(рисунок 4.4):

- метод порогової різниці «проковтнув» половину зображення

рахуючи її інформативною;

- метод Хрящева не зміг впоратися зі скелею;
- метод за середньою довжиною впорався зі всіма поставленими задачами і коректно поділив зображення.

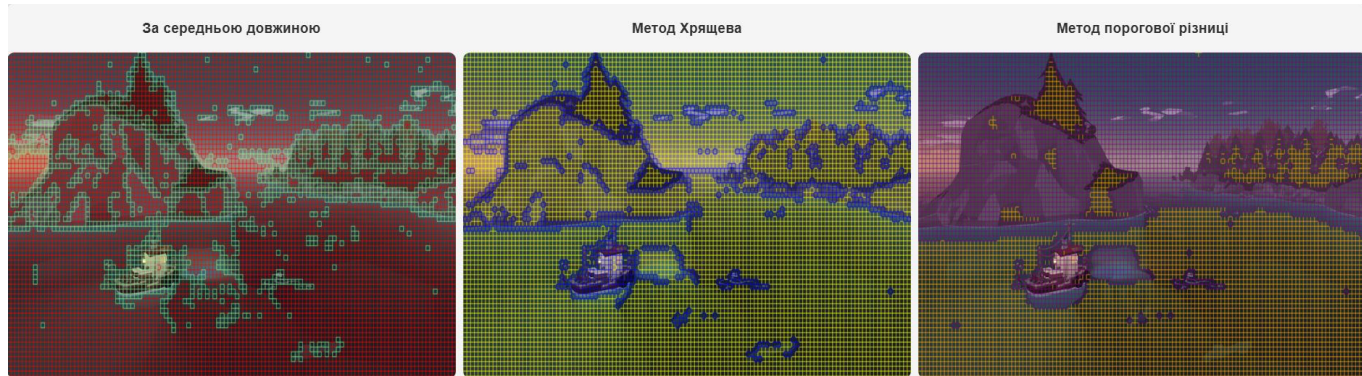


Рисунок 4.4 – Результати сегментації другого кадру

Третій кадр представляє собою фрагмент вечірньої зйомки з не дуже якісною картинкою і великою кількістю яскравих деталей (рисунок 4.5).



Рисунок 4.5 – Третій кадр для аналізу

На відміну від другого кадру тут всі три методи впоралися приблизно з однаковою достовірністю. Можна помітити що метод порогової різниці знову порахував інформативними більшу кількість блоків ніж інші методи, але в даному випадку він тим самим зміг знайти деякі інформативні блоки,

які інші методи знайти не змогли, не дивлячись на це, він так само зарахував деякі неінформативні блоки як інформативні, а тому його достовірність не є кращою. У цьому випадку сегментація за середньою довжиною все ще є кращою, але вже тільки завдяки часу, де наш метод, як і раніше, обходить конкурентів майже на 50%(рисунок 4.6).

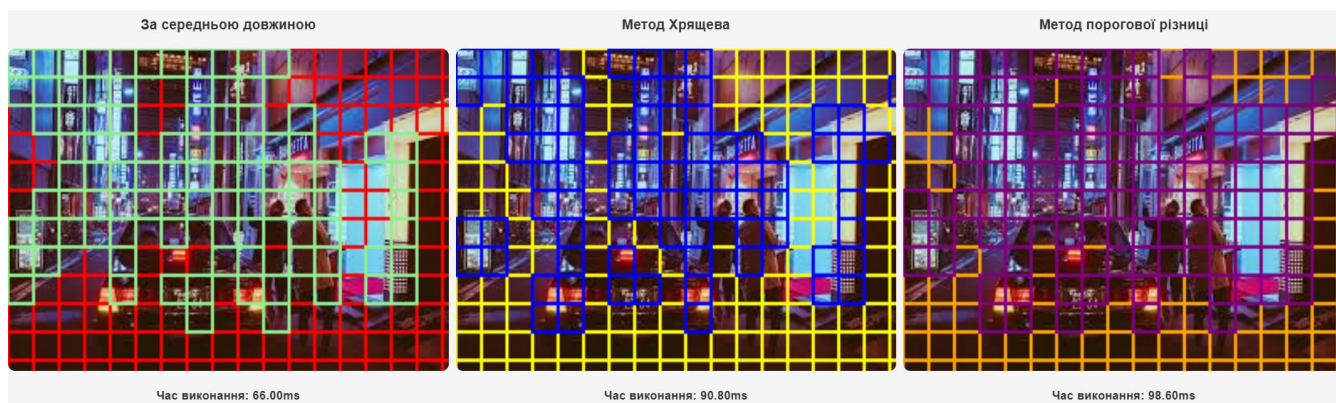


Рисунок 4.6 – Результат сегментації третього кадру

Підводячи підсумки представимо таблицю 4.1 зі всіма результатами.

Таблиця 4.1 – Результати проведених досліджень

Метод сегментації	Перший кадр		Другий кадр		Третій кадр	
	Достовірність, %	Час, ms	Достовірність, %	Час, ms	Достовірність, %	Час, ms
Середня довжина	97.5	170	96	400	91.5	62
Метод Хрящева	94	503	89	3000	91	121
Порогова різниця	96	500	67.8	2800	92	77

Результати дослідження підтвердили перевагу методу сегментації за

середньою довжиною в контексті точності та швидкості. У той час як метод порогової різниці інколи демонструє надмірну схильність до інформативності, а метод Хрящева стикається з труднощами у складних випадках, метод середньої довжини забезпечує баланс між достовірністю й ефективністю.

Таким чином, можна зробити висновок, що універсального підходу, який би ідеально працював у всіх сценаріях, не існує. Проте для більшості практичних задач оптимальним є саме метод сегментації за середньою довжиною, який враховує як точність, так і обчислювальну ефективність. У цьому сенсі, кожен кадр стає ще одним прикладом того, як технологія може гармонійно взаємодіяти зі складністю світу.

4.4 Оцінка ефективності запропонованого методу через аналіз відеоданих

Для оцінки ефективності представленого методу було обрано 5 відео, довжиною приблизно по 10 секунд, всі вони були поділені на кадри (рисунок 4.7), після чого запропонованим методом сегментації кадри були поділені на інформативні та не інформативні.

На ефективність запропонованого методу нам вкаже загальний обсяг обчислень які він потребує.

Для розрахунку загального обсягу обчислень пропонується використовувати формулу, яка включає в себе потужності, які потребує сегментація:

$$R_{proposed} = R_{segmentation} + R_{inf} + \alpha R_{non_inf},$$

$R_{segmentation}$ — обчислення, необхідні для сегментації (наприклад, аналіз середньої довжини змін для кожного блоку); R_{inf} — обчислення для

інформативних блоків; R_{non_inf} — обчислення для неінформативних блоків;
 α — коефіцієнт зниження обчислювальних витрат для неінформативних
 блоків ($\alpha = 0.3$).

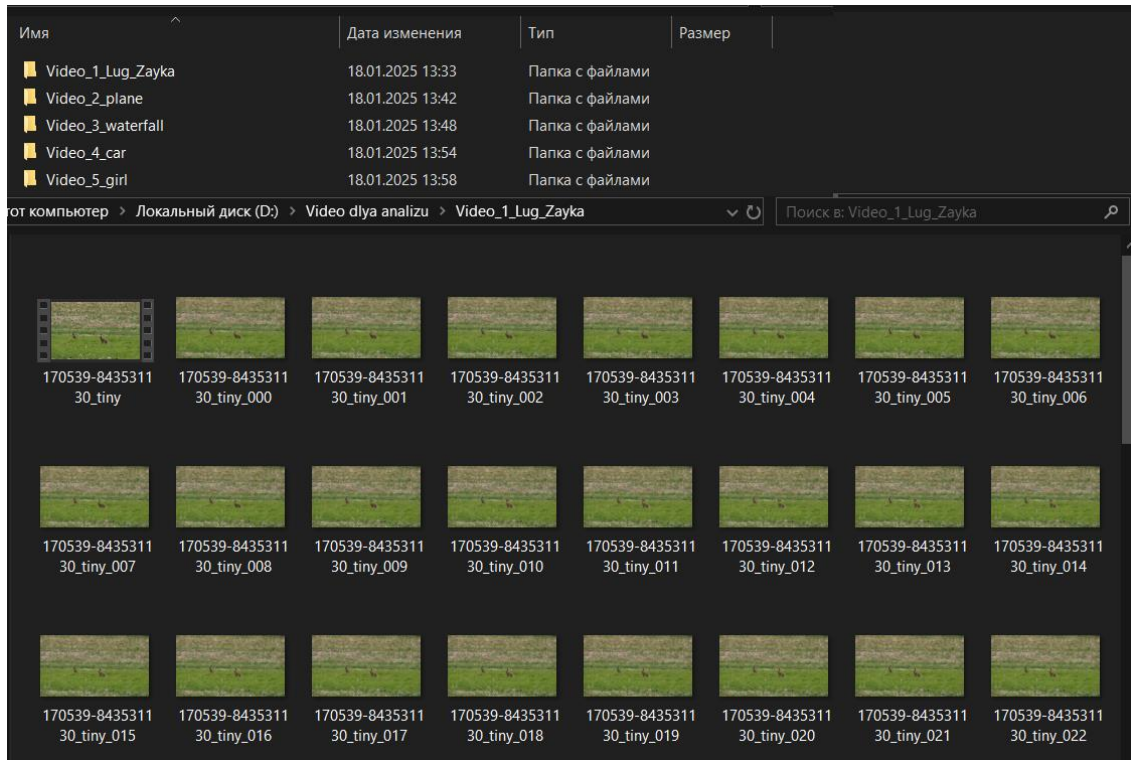


Рисунок 4.7 – Відео для аналізу

Тоді перевагу методу будемо обчислювати як:

$$E = \frac{R_{MPEG-4} - R_{proposed}}{R_{MPEG-4}} \times 100\%.$$

Після проведення експериментів можемо перейти до підрахунку результатів.

На основі експериментів було визначено, що витрати на сегментацію складають у середньому 15% від обсягу обчислень класичного MPEG-4 для кадру:

$$R_{segmentation} = 0.15 \times R_{MPEG-4}.$$

Пропонується на прикладі файлу “Video_5_girl”, провести підрахунок ефективності.

Було проведено аналіз на вибірці 115 кадрів. Де у середньому кількість інформативних блоків дорівнювала 30%(рисунок4.8).

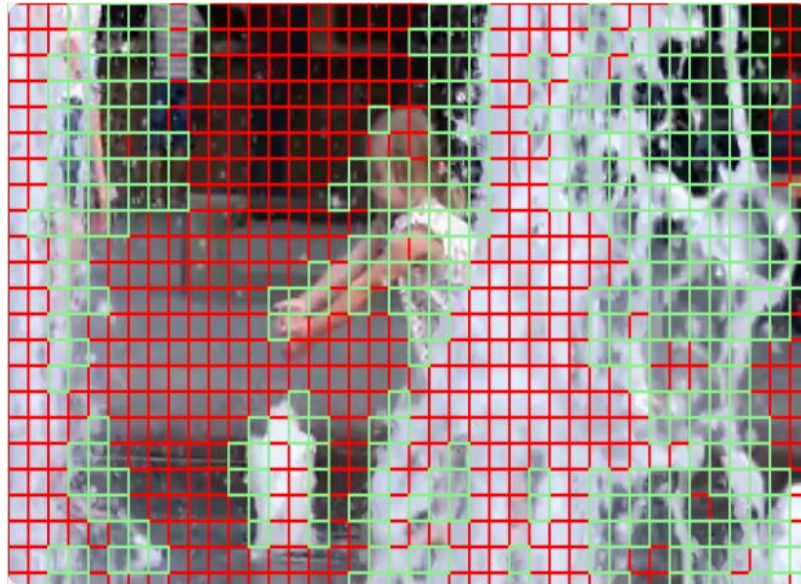


Рисунок 4.8 – Сегментація одного з кадрів відео

Тож, ми маємо середню кількість інформативних блоків, середні витрати на сегментацію, та коефіцієнт для неінформативних блоків. Можемо починати розрахунок ефективності:

- спочатку порахуємо обсяг обчислень для цього відео:

$$R_{propesed} = 0.15 \cdot R_{MPEG-4} + 0.3 \cdot R_{MPEG-4} + 0.7 \cdot 0.3 \cdot R_{MPEG-4} = 0.66 \cdot R_{MPEG-4},$$

тож запропонований метод для цього відео буде мати тільки 66% від обсягу MPEG-4;

- порахуємо економію(ефективність) обчислень:

$$E = \frac{R_{MPEG-4} - 0.66 \cdot R_{MPEG-4}}{R_{MPEG-4}} \times 100\% = 34\%,$$

тож ефективність для цього відео буде приблизно дорівнювати 34%.

Таким чином було проведено аналіз і інших відео, у таблиці 4.2 наведено результати усіх випробовувань,

Таблиця 4.2 – Результати дослідження на основі 5 запропонованих відео

Експериментна частина	Розмір відео, Мб	Розмір кадру	Кількість кадрів	Середня кількість інформативних блоків, %	Ефективність обчислень, %
Відео 1	4,71	1280x720	126	35	30.5
Відео 2	0,64	640x360	106	25	37.5
Відео 3	3,33	1280x720	86	20	41
Відео 4	3,78	1280x720	105	30	34
Відео 5	0,79	640x360	115	30	34
Середні значення	2,65	-	108	28	35.4

Результати експерименту показали, що запропонований метод сегментації демонструє свою ефективність у різних умовах. Сцени з меншою кількістю деталей, як літак у небі чи водоспад, виявилися ідеальними для методу, оскільки велика частка блоків в таких відео є неінформативною, що дозволяє суттєво скоротити обчислювальні витрати. Водночас, у динамічних сценах, таких як луг із зайцями або машина в пустелі, кількість інформативних блоків збільшується, що зменшує рівень загальної оптимізації, але все одно залишає помітний вигравш у ресурсах.

У середньому запропонований підхід забезпечує економію обчислень на рівні 35.4%, що свідчить про його здатність балансувати між якістю обробки та ефективністю використання ресурсів. Така оптимізація особливо важлива для складних систем, де обробка відеоданих вимагає високої швидкості й точності. Метод вдало адаптується до різних сцен, зберігаючи деталізацію в ключових областях кадру, при цьому уникаючи зайвої витрати ресурсів на менш важливі ділянки. Розроблений алгоритм є універсальним інструментом для сучасних завдань відео стиснення.

ВИСНОВКИ

Проведене дослідження довело, що вдосконалення стандарту MPEG-4 через інтеграцію адаптивних методів дозволяє значно підвищити ефективність стиснення відео. Сегментація кадрів за середньою довжиною змін стала ключовим етапом, що забезпечив оптимальний розподіл обчислювальних ресурсів, зосереджуючи обробку на найбільш важливих ділянках кадру. Це дозволило мінімізувати витрати на менш інформативні зони, що є особливо важливим у реальних умовах роботи інформаційно-телекомунікаційних мереж.

Дослідження, проведені на вибірці з п'яти відео різної динаміки, підтвердили ефективність підходу, продемонструвавши середній рівень оптимізації обчислень на рівні 35,4%. Такий результат свідчить про те, що запропонований метод здатен не лише адаптуватися до різних типів сцен, але й зберігати високу якість відео при значному зменшенні обсягу даних.

Важливим є той факт, що модернізація враховує обмеження сучасних мереж, таких як затримки, втрати пакетів та варіативність пропускну здатності. Це дозволяє зробити процес обробки відео більш адаптивним і стійким до змінних умов, що є ключовим фактором для сучасних систем, таких як потокові сервіси, відеоконференції та відеоспостереження.

Запропоновані вдосконалення демонструють потенціал для подальшого розвитку стандартів відеокодування. Адаптивні алгоритми та сегментація відкривають нові можливості для створення більш ефективних систем, здатних працювати в умовах обмежених ресурсів і високих вимог до якості відео. Ці результати показують, що майбутнє відеокомпресії лежить у поєднанні інноваційних підходів із врахуванням потреб сучасних мереж та користувачів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Oleksandr Ignatyev et al./ "A Method for Increasing the Reliability of Video Information in Information and Telecommunication Networks with Adaptive Coding and Quantisation" 2024 IEEE 5th International Conference on Advanced Trends in Information Theory (ATIT). IEEE Conference, November 21 – 23, 2024, Lviv, Ukraine, pp.x-xx.

2. Ігнат'єв О.О., Барковська О.Ю. Методи оцінювання систем стиснення відеоданих // Проблеми інформатизації : XII міжнародна науково-технічна конференція. - 21-22 листопада 2024. –с.75. doi: <https://doi.org/10.32620/PI.24.t2>.

3. The Concept Of Creating A Complex Cryptocompression Image Protection System In Infocommunications / V. Barannik та ін. 2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT), м. Kyiv, Ukraine, 15–17 груд. 2022 р. 2022. URL: <https://doi.org/10.1109/atit58178.2022.10024210>.

4. Model of Steganographic System Depending on Indirect Conditional Dependencies / N. Barannik та ін. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT), м. Kyiv, Ukraine, 15–17 груд. 2021 р. 2021. URL: <https://doi.org/10.1109/atit54053.2021.9678547>.

5. Sabha A., Selwal A. Towards machine vision-based video analysis in smart cities: a survey, framework, applications and open issues. *Multimedia Tools and Applications*. 2023. URL: <https://doi.org/10.1007/s11042-023-16434-2>.

6. Nassra I., Capella J. V. Data Compression Techniques in IoT-enabled Wireless Body Sensor Networks: A Systematic Literature Review and Research Trends for QoS Improvement. *Internet of Things*. 2023. P. 100806. URL: <https://doi.org/10.1016/j.iot.2023.100806>.

7. A Method of Encoding Video Information to Increase its Reliability in an Information and Telecommunications Network / V. Barannik et al. 2023 IEEE 5th

International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 21–25 November 2023. 2023. URL: <https://doi.org/10.1109/aict61584.2023.10452419>.

8. Action recognition in compressed domains: A survey / Y. Ming et al. *Neurocomputing*. 2024. Vol. 577. P. 127389. URL: <https://doi.org/10.1016/j.neucom.2024.127389>.

9. Bocharova I. Video-coding standards. *Compression for Multimedia*. Cambridge. P. 197–222. URL: <https://doi.org/10.1017/cbo9780511804069.010>.

10. Grubb D. S., Cotton I. W. Criteria for evaluation of data communications services. *Computer Networks (1976)*. 1977. Vol. 1, no. 6. P. 325–340. URL: [https://doi.org/10.1016/0376-5075\(77\)90009-5](https://doi.org/10.1016/0376-5075(77)90009-5).

11. Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos / J. Bienik et al. *Sensors*. 2023. Vol. 23, no. 5. P. 2744. URL: <https://doi.org/10.3390/s23052744>.

12. Bull D. R., Zhang F. Lossless compression methods. *Intelligent Image and Video Compression*. 2021. P. 225–270. URL: <https://doi.org/10.1016/b978-0-12-820353-8.00016-5>.

13. Bhaskaran V., Konstantinides K. Fundamentals of Lossy Image Compression. *Image and Video Compression Standards*. Boston, MA, 1995. P. 53–86. URL: https://doi.org/10.1007/978-1-4757-2358-8_3.

14. Domínguez H. O., Rao K. R. Beyond High Efficiency Video Coding (HEVC). *Versatile Video Coding*. New York, 2022. P. 3–18. URL: <https://doi.org/10.1201/9781003339991-2>,

15. Performance Comparison of VVC, AV1, HEVC, and AVC for High Resolutions / M. Uhrina et al. *Electronics*. 2024. Vol. 13, no. 5. P. 953. URL: <https://doi.org/10.3390/electronics13050953>.

16. 6 Points of Comparison for VP9, H.265, or H.264 - Red5. *Red5*. URL: <https://www.red5.net/blog/6-points-of-comparison-for-vp9-or-h265/>.