

Не містить відомостей, заборонених
до відкритого публікування

Керівник _____ /*А.В.Омельченко*

Студент _____ / *С.А. Наталюк*

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
Кафедра Інформаційно-мережної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 172. Телекомунікації та радіотехніка
(код і повна назва)
Тип програми Освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)
« 15 березня » 20 21 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Наталюку Сергію Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Метод ущільнення даних мультимедіа з використанням
конструкції змінної ваги

затверджена наказом університету від 12 березня 2021 р. № 350 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 травня 2021 р.

3. Вихідні дані до роботи Дослідити особливості відеотрафіку сервісів віртуальної
реальності та інтерактивного відео. Зробити висновки про вплив трафіку даних типів
на забезпечення якості мережесервісів. Довести обмежені можливості існуючих
методів ущільнення відеоданих щодо ефективної обробки відеоданих. Виконати аналіз
алгоритму кодування без втрат на базі кодових конструкцій змінної ваги.
Обґрунтувати його ефективність порівняно з існуючими методами

4. Перелік питань, що потрібно опрацювати в роботі Вступ

1. Сервіси мультимедіа потокової та інтерактивної трансляції реального часу

2. Підходи щодо зменшення мережевого навантаження, що створюють мультимедійні
відеосервіси

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) слайди презентації в форматі Power Point (назва та мета роботи, особливості формування трафіку відео, аналіз існуючих алгоритмів ущільнення відеоданих, метод кодування на базі колових конструкцій змінної ваги)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ		
2	Сервіси мільтимедіа потокової та інтерактивної трансляції реального часу__		
3	Підходи щодо зменшення мережевого навантаження, що створюють мультимедійні відеосервіси		
4	Стиснення без втрат фрагментів кадру у спектральному просторі з використанням ефективних кодових конструкцій		
5	Дослідження степеню складності алгоритму обробки фрагментів відеокадрів на базі блочних конструкцій зі змінною вагою		
6	Висновки		
7	Оформлення пояснювальної записки		

Дата видачі завдання 15 березня 2021 р.

Студент _____
 (підпис)

Керівник роботи _____
 (підпис)

С.А. Наталюк
доц. Омельченко А.В.
 (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 92 с., 23 рис., 2 табл., 24 джерела, 1 додаток

ВІДЕОКОДУВАННЯ, ВІРТУАЛЬНА РЕАЛЬНІСТЬ, ІНТЕРАКТИВНЕ ВІДЕО, ДИСКРЕТНО-КОСИНУСНЕ ПЕРЕТВОРЕННЯ, Н.265, ВАГОВИЙ КОЕФІЦІЄНТ

Об'єкт дослідження – методи кодування відеоданих за рахунок обробки структурних складників окремих кадрів потоку.

Мета роботи – дослідження можливостей удосконалення існуючих методів стиснення відео за рахунок використання кодових конструкцій змінної ваги.

Здійснено розгляд особливостей розвитку інформаційно-комунікаційних мереж та сервісів на їх базі. Виявлено проблематику у питаннях забезпечення якості надання відеосервісів в умовах зростання ступеню їх критичності та збільшення інформаційних інтенсивностей потоку. Показано, що продуктивність стандартизованих технологій стиснення відео обмежена. Розглянуто підхід до стиснення відеоданих у схемі MPEG на базі використання кодових конструкцій змінної ваги у якості альтернативи існуючим статистичним методам кодування.

THE ABSTRACT

Explanatory note: 92p., 23 fig., 2 tabl., 24 sources, 1 app.

VIDEO CODING, VIRTUAL REALITY, INTERACTIVE VIDEO,
DISCRETE-COSINUS TRANSFORMATION, H.265, WEIGHT COEFFICIENT

The object of research - methods of encoding video data by processing the structural components of individual frames of the stream.

The purpose of the work is to study the possibilities of improving the existing methods of video compression through the use of code constructions of variable weight.

The peculiarities of development of information and communication networks and services on their basis are considered. Problems in ensuring the quality of video services in terms of increasing the degree of their criticality and increasing the information intensity of the flow. It is shown that the performance of standardized video compression technologies is limited. The approach to video data compression in the MPEG scheme based on the use of code constructs of variable weight as an alternative to existing statistical coding methods is considered.

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП.....	10
1 СЕРВІСИ МІЛЬТИМЕДІА ПОТОКОВОЇ ТА ІНТЕРАКТИВНОЇ ТРАНСЛЯЦІЇ РЕАЛЬНОГО ЧАСУ.....	12
1.1 Загальні особливості існуючого стану розвитку мережевих сервісів.....	12
1.2 Сервіси віртуальної реальності.....	15
1.3 Інтерактивне відео.....	28
2 ПІДХОДИ ЩОДО ЗМЕНШЕННЯ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ, ЩО СТВОРЮЮТЬ МУЛЬТИМЕДІЙНІ ВІДЕОСЕРВІСИ.....	31
2.1 Загальні принципи підходів до скорочення навантаження за рахунок зменшення у мережі кількості пакетів мультимедійних сервісів.....	31
2.2 Механізми зменшення інформаційної інтенсивності відеоданих.....	32
2.2.1 Принцип функціонування платформи MPEG.....	33
2.3 Технологія H.265/HEVC.....	37
2.3.1 Особливості стандарту H.265 HEVC.....	38
2.3.2 Висновки за результатами аналізу технології H.265.....	45
2.4 Алгоритм SPIHT.....	47
2.4.1 Загальні принципи функціонування алгоритму.....	47
2.4.2 Етапність робити алгоритму SPIHT.....	51
2.4.3 Етапність робити алгоритму SPIHT.....	52
3 СТИСНЕННЯ БЕЗ ВТРАТ ФРАГМЕНТІВ КАДРУ У СПЕКТРАЛЬНОМУ ПРОСТОРІ З ВИКОРИСТАННЯМ ЕФЕКТИВНИХ КОДОВИХ КОНСТРУКЦІЙ.....	54
3.1 Вимоги до методу кодування без втрат у складі технології відеокодування.....	54
3.2 Поширені методи кодування без втрат у складі систем стиснення відео.....	57
3.3 Особливості та переваги застосування ефективних кодових конструкцій змінної ваги при формуванні кодограм фрагментів кадру у спектральному просторі.....	58
3.4 Стиснення без втрат фрагментів кадру у спектральному просторі з використанням кодових конструкцій змінної ваги.....	61

3.5 Реконструкція даних, кодованих у базисі кодового опису зі змінною вагою на боці прийому	65
4 ДОСЛІДЖЕННЯ СТЕПЕНЮ СКЛАДНОСТІ АЛГОРИТМУ ОБРОБКИ ФРАГМЕНІВ ВІДЕОКАДРІВ НА БАЗІ БЛОЧНИХ КОНСТРУКЦІЙ ЗІ ЗМІННОЮ ВАГОЮ.....	69
4.1 Підхід до дослідження складності реалізації алгоритму.....	69
4.2 Розрахунок обчислювальної складності методу кодуваннями зі змінною вагою	70
4.3 Визначення підходу щодо оптимізації алгоритму кодування на базі кодових структур змінної ваги	76
4.3.1 Розрахунок необхідної кількості інструкцій для процесу побудови базисів кодових чисел.....	76
4.3.2 Розрахунок необхідної кількості інструкцій у ході визначення вагових коефіцієнтів та при розрахунку кодового числа.....	77
4.4. Висновки щодо підсумкової складності алгоритму.....	77
ВИСНОВКИ.....	79
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	82
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	84

ПЕРЕЛІК СКОРОЧЕНЬ

- VR – (virtual reality) – віртуальна реальність;
- AWS – (Amazon Web Services) – комерційний публічний хмарний сервіс;
- WiFi – (Wireless Fidelity) – технологія бездротової передачі даних;
- LTE-A – (long term evolution advanced) – технологія високошвидкісного бездротового зв'язку покоління 4,5G;
- MPEG – сімейство стандартів відео кодування;
- FullHD – (full-high definition) – повноформатне відео високої роздільної здатності;
- QoS – технологія контролю якості послуг, які надає інформаційно-комунікаційна мережа;
- M-JPEG та M-JPEG2000 – стандарти відео кодування, що базуються на скороченні просторової надмірності відео;
- GOP – (group of pictures) – упорядкована група кадрів різних типів, структурна одиниця відеопотоку;
- RLE – (run-length encoding) – технологія кодування ланцюжків двійкових символів;
- JPEG – технологія кодування зображень;
- RGB – (red, green, blue) – трьохколірна модель;
- YCrCb – (y, chromatic red, chromatic blue) – яскравісно-хроматична колірна модель;
- H.265/HEVC – (high-efficiency video coding) — технологія високоефективного кодування відео H.265;
- H.265/ AVC – (advanced video coding) — технологія високоефективного кодування відео H.264;
- СAВАС – (context-adaptive binary adaptive coding) — контекстно-адаптивне двійкове арифметичне кодування;
- СAVLC – (context-adaptive variable length coding) — контекстно-адаптивне кодування кодами змінної довжини;
- SPIHT – (set partitioning in hierarchical trees) — алгоритм просторово впорядкованих ієрархічних дерев;
- JPEG2000 – технологія кодування зображень;
- MSE – (mean square error) – середньоквадратичне відхилення (СКВ);
- ДВП – дискретно-вейвлетне перетворення;

ДКП – дискретно-косинусне перетворення;

VoD – (video on demand) – група сервісів, що надають послуги доставки потокового відео на запит користувача;

VoIP – (voice over IP) – загальна назва сервісів, що надають послуги голосового зв'язку поверх мережі IP.

ВСТУП

Сфера застосування інформаційно-комунікаційних систем та технологій сьогодні є значно ширшою, ніж хоча б 5-10 років тому, та охоплює такі аспекти повсякденної діяльності людини, як:

- транспорт та зв'язок;
- навчання та освіта;
- охорона здоров'я та спорт;
- безпека;
- наукові дослідження;
- фінанси та банківська діяльність та інше.

Наслідком цього є одночасне функціонування великої кількості мережевих сервісів та додатків, що використовують всесвітню мережу у якості спільного розподіленого середовища доступу.

При цьому, окрім сервісів загального вжитку, таких, як інформаційні платформи, системи замовлення послуг онлайн або платіжних систем існує великий обсяг спеціалізованих систем [1, 2]. Це, у першу чергу, системи, що виконують завдання обслуговування об'єктів критичної інфраструктури, таких, як:

- технологічні комплекси, спрямовані на забезпечення життєдіяльності міст та великих територій, де проживає значна кількість громадян (ТЕЦ, гідроелектростанції та ядерні реактори, системи контролю викидів шкідливих відходів, очисні споруди, системи розподілу електроенергії та газопостачання тощо);

- технологічні комплекси, що не відносяться до рангу першочергово критичних, але є стратегічно важливими у межах країни чи її окремих територій (виробництва складного технологічного циклу).

Окрім зазначеного, стратегічно важливими також є, зокрема, системи геопозиціонування та різноманітні GIS-сервіси, збій у роботі яких може привести до аварійного стану систем та додатків, що для свого функціонування використовують дані, отримані від них.

У підсумку, розширення діапазону доступних до використання мережевих сервісів та збільшення кількості активних пристроїв, що виконують роль кінцевих терміналів у межах кожного з таких сервісів, створює умови, у яких може спостерігатися падіння якості надання сервісів. Очевидно, що

найбільшою мірою це є неприйнятним до сервісів критичного сегменту, вихід з ладу яких, або, навіть, аварійне функціонування може спричинити виникнення техногенних катастроф, соціальних зрушень та значних фінансових збитків.

Таким чином, паралельно з розвитком прикладних технологій для розвитку окремих мережевих сервісів мають також удосконалюватися мережева інфраструктура та технології мережевої взаємодії – усе те, що, у свою чергу, якраз і формує поняття «розподілене мережеве середовище».

Разом з тим, як можна спостерігати останні роки, потужності мережевої інфраструктури та можливості технологій передавання поступаються існуючим об'ємам даних, що транслюються мережею [1]. При цьому, значний відсоток – понад 82%, належить відеоданим, які, по-перше, вимогливими до показників втрат та затримки. А по-друге, потребують гарантованої смуги пропускання.

У той же час, ситуація загострюється тим, що у складі відеотрафіку збільшується відсоток сервісів, та, відповідно, зростає інформаційна інтенсивність найбільш вимогливих сервісів з класу відео – це, зокрема, відео сервіси інтерактивного типу, та сервіси віртуальної реальності (VR). Дані сервіси характеризуються дещо вищими вимогами до якості порівняно з, наприклад, відео потокового типу.

У таких умовах, одним з найбільш раціональних рішень щодо забезпечення якості будь-яких мережевих сервісів, є ущільнення трафіку відео. При цьому, відносно відеоінформаційних сервісів також має забезпечуватися певний встановлений рівень якості.

Відповідно, дослідження, розробка та удосконалення алгоритмів, механізмів та методів кодування відеоданих на сьогодні є одним з пріоритетних та актуальних завдань.

1 СЕРВІСИ МІЛЬТИМЕДІА ПОТОКОВОЇ ТА ІНТЕРАКТИВНОЇ ТРАНСЛЯЦІЇ РЕАЛЬНОГО ЧАСУ

1.1 Загальні особливості існуючого стану розвитку мережевих сервісів

Розширення можливостей технологій передавання даних та кінцевого обладнання, а також зростання пропускнуєї спроможності мережі створили умови для розвитку та удосконалення багатьох мережевих сервісів та систем.

При цьому, за увесь час виникнення та поступового розвитку Всесвітньої мережі спостерігається стійка залежність між кількістю G інформації, яка може бути оброблена доступними на поточний момент часу засобами, та рівнем L попиту соціуму на послуги, що передбачають обробку, зберігання та надсилання інформації. Така залежність, у загальному випадку, апроксимовано може вважатися лінійною, хоча і має більш складний характер (рис.1.1) [2].

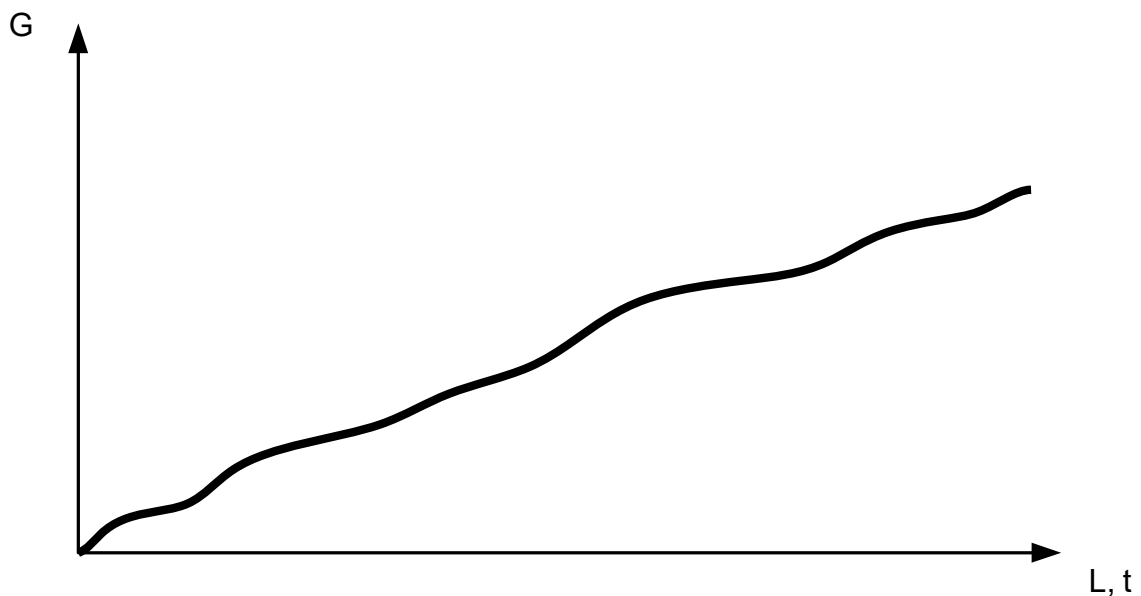


Рисунок 1.1 – Загальний характер взаємозв'язку попиту на ті чи інші мережеві сервіси та обсягу даних, що надсилаються мережею, з кількістю інформації, яка може бути оброблена доступними на поточний момент часу засобами

Водночас, рівень L попиту на послуги весь час спостереження є вищим, між фактичні можливості наявної мережевої інфраструктури, як показано наступним співвідношенням:

$$G' = \varphi(L) > G, \quad (1.1)$$

де G' - кількість даних у мережі, що має надсилатися у одиницю часу для того, щоб задовольнити користувацький попит у тих чи інших мережевих сервісах.

У свою чергу, величина L попиту інтерпретується як функціонал від кількості активних мережевих користувачів n_u та максимальної інтенсивності $g(t)_{\max}$ даних, що генерує одне джерело даних у одиницю часу, та наближено може обчислюватися, як добуток цих величин, тобто:

$$L = f(n_u; g(t)_{\max}) = n_u \times g(t)_{\max} \quad (1.2)$$

Разом з тим, відношення кількості G інформації, що може бути опрацьована доступними засобами, до середньої максимальної інтенсивності $\bar{g}(t)_{\max}^{(\text{sum})}$ даних, являє собою коефіцієнт K_{use} використання мережевих ресурсів, тобто:

$$K_{\text{use}} = \frac{G}{\bar{g}(t)_{\max}^{(\text{sum})}}, \quad (1.3)$$

де $\bar{g}(t)_{\max}^{(\text{sum})}$ обчислюється як $\bar{g}(t)_{\max}^{(\text{sum})} = \frac{\sum_{u=1}^U \bar{g}(t)_{\max}^{(u)}}{U}$;

U - кількість джерел даних.

При цьому, спостерігаються такі значення співвідношення (1.3), як:

1. $K_{\text{use}} > 1$, що відповідає випадкам, коли у мережу необхідно надіслати такі кількість даних, що перевищує наявну потужність мережевої інфраструктури для її своєчасної обробки. У даній ситуації спостерігається зниження якості мережевих сервісів та не гарантується їх підтримка на встановленому рівні.

2. $K_{\text{use}} < 1$. Це відповідає випадкам, коли продуктивність мережевих засобів (технологій передавання, протоколів та обладнання) перевищує

затребуваність сервісів на базі мережі. Ситуація є неоднозначною, оскільки при цьому:

- якість мережевих сервісів підтримується на заданому рівні;
- витрати на підтримку функціонування мережі можуть перевищувати прибуток від її використання, тобто, створюються умови для того, щоб провайдери сервісів зазнавали збитків.

3. $K_{use} \rightarrow 1$. Такий випадок є компромісним. При цьому забезпечується відносно високий рівень якості мережевих сервісів, одночасно зі збереженням цінової вартості комунікаційних послуг та значному рівні завантаженості мережі.

Сьогодні розвиток мереж найбільшою мірою нагадує умови 1, тобто, коли $K_{use} > 1$. Наслідком цього є падіння якості ряду сервісів [1]. У першу чергу таких, що характеризуються:

- значними рівнями $g(t)$ інформаційних інтенсивностей;
- критичність до змін умов функціонування.

Серед усієї множини мережевих сервісів до таких відносяться, зокрема, сервіси мультимедіа, що орієнтовані на відеотрафік (рис.1.2).

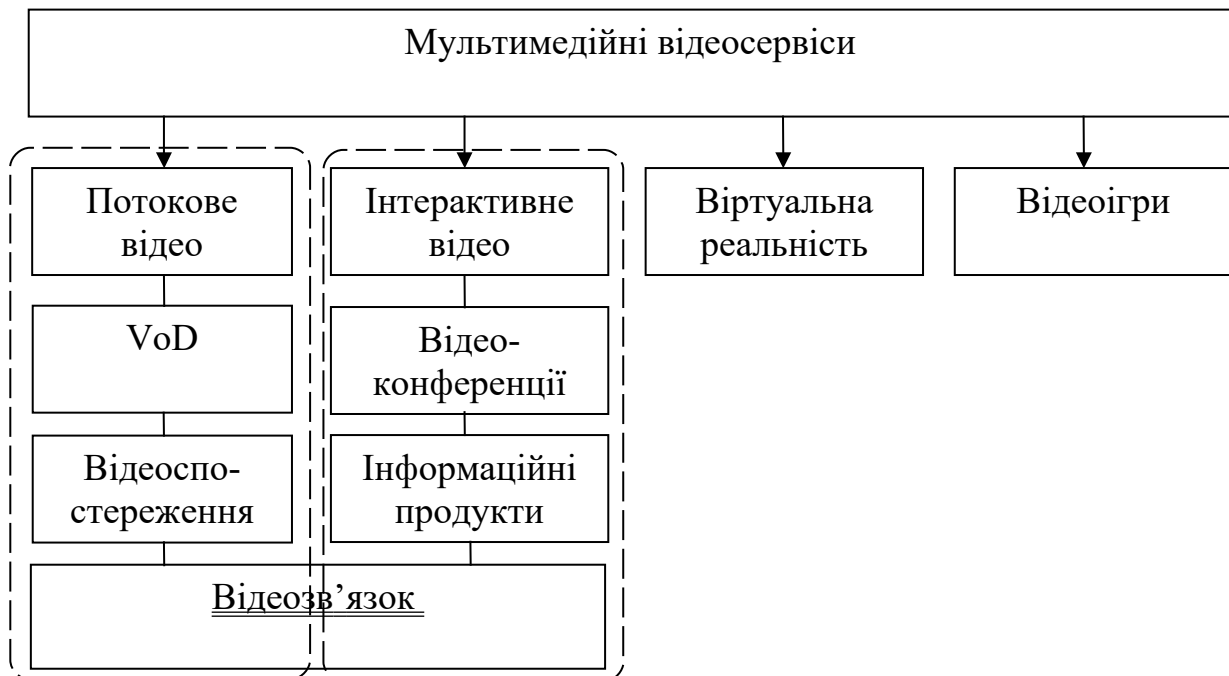


Рисунок 1.2 – Умовна класифікація існуючих мультимедійних відеосервісів

У наведеній рис.1.2 класифікації окремо, як найбільш критичні, можна виокремити сервіси віртуальної реальності (VR) та сервіси інтерактивного відео.

Далі розглянемо особливості означених типів сервісів.

1.2 Сервіси віртуальної реальності

Зараз існує 2 ключові підходи до розвитку сервісів віртуальної реальності [3, 4].

Перший з них передбачає наявність локально розміщеного серверу VR у межах локальної мережі. При цьому, сервер додатку віртуальної реальності може бути створено фактично на базі будь-якої робочої станції за умови її достатньої обчислювальної потужності.

Клієнтами VR-сервісів для такого формату їх реалізації можуть бути користувачі локальної мережі.

Такий підхід щодо архітектури сервісів віртуальної реальності розглядається як початковий рівень розвитку даного технологічного напрямку.

Йому властивий певний ряд недоліків, зокрема:

- залучення користувачів з зовнішньої мережі потребує значних фінансових та часових витрат, так як передбачає необхідність зміни апаратної платформи як власне серверу, так і мережевого обладнання мережі на ділянці «пограничний маршрутизатор-сервер VR»;
- зміна початкової структури мережі з виокремленням внутрішнього сегменту та сегменту, у якому локалізовано сервер VR з можливістю зовнішнього доступу.

У таких умовах значно раціональнішими є рішення, у рамках яких сервер додатків VR організовується на базі виділеної апаратної платформи (фізичний сервер), або хмарного рішення, яке виконує виключно функції створення та підтримки сервісів віртуальної реальності.

У той же час, як свідчить аналіз ряду хмарних сервісів, в умовах, коли необхідно забезпечити функціонування VR-додатків з деяким встановленим рівнем якості, більшість з них на сьогодні демонструють результати на рівні «нижче середнього».

Це, зокрема, стосується AWS та подібних, як наслідок відносно низької швидкості обміну даними з зовнішніми користувачами. Відповідно, високовимогливі VR-додатки сьогодні рекомендується утілювати на виділених фізичних серверах.

Разом з тим, незалежно від формату реалізації, на сьогодні існує 3 напрямки розвитку додатків VR, зокрема [3]:

- відео 360°;
- VR –симуляція (імітація);
- VR –заходи (VR-events).

Перший з напрямків реалізації VR-технологій на сьогодні є найповніше утіленим.

На цей випадок забезпечується можливість перегляду об'ємного відео у форматі 360° (рис.1.3), що досягається за рахунок використання спеціалізованих камер 360°, або шляхом 3D-моделювання віртуального оточення.



Рисунок 1.3 – Лінеаризований фрагмент відеоряду сферичної зйомки 360°

Такий формат VR є найпростішим для реалізації з наступних причин:

- глядач не бере участі у генеруванні відео контенту, окрім як шляхом зміщення точки перегляду;
- об'єкти відеоряду, з якими можлива віртуальна взаємодія, відсутні. При цьому, відеоряд може бути згенеровано на базі одного пристрою з можливістю панорамної зйомки.

Наприклад, це може бути портативний варіант, як Xiaomi Miija 360 Panoramic Camera (рис.1.4) або інший подібний пристрій [5].

Такі пристрої для створення об'ємного відеоряду генерують відеокадри з роздільною здатністю до 3,5 К, що відповідає просторовій роздільній здатності 2304×1152 пікселів при 60 к/с, або 3456×1728 (30 к/с).

На базі даного пристрою може бути реалізовано найпростіший варіант VR відео 360°.



Рисунок 1.4 – Панорамна камера Xiaomi Miija 360 Panoramic Camera

Формування відеопотоку виконується на основі кодеку H.264/AVC, відповідно, у зазначених режимах створення відеоряду це здатно створити навантаження $g(t)$ на мережу на рівні понад 16 Мбіт/с за умови роздільної здатності 3456×1728 при 30 к/с, або близько 14 Мбіт/с для режиму 2304×1152 пікселів при 60 к/с.

На відміну від розглянутого випадку, для створення відеоряду VR відео 360° на базі професійного обладнання, одночасно може використовуватися цілий ряд джерел відеоінформації, поєднаних у рамках одного пристрою.

Це, у свою чергу, відповідним чином впливає на підсумкове навантаження $g(t)$.

Так, для Xiaomi YI HALO (рис. 1.5), відеофіксація ведеться на 17 камер одночасно [6].

Для побудови відеоряду у цьому випадку також використовується кодек H.264/AVC, тобто, інформаційна інтенсивність $g(t)$ відео потоку для розглянутих режимів зйомки може сягати 225 Мбіт/с ($17 \times 2560 \times 1920$, 29,97 к/с), 181 Мбіт/с (8192×8192 , 29,97 к/с), та 149 Мбіт/с (5760×5760 , 50 к/с).

У підсумку відеофіксація може вестися у декількох режимах, зокрема:

- $17 \times 2560 \times 1920$ пікселів при 29,97 к/с;
- 8192×8192 пікселя при 29,97 к/с;
- 5760×5760 пікселів при 50 к/с та ін.

Таким чином, навіть найпростіший варіант додатків VR за певних умов може створювати суттєве навантаження $g(t)$ на мережу.



Рисунок 1.5 – Панорамна камера Xiaomi Yi HALO

У той же час, для групи сервісів VR-симуляції забезпечення деталізації відеосцен за рахунок генерування відеоряду високої роздільної здатності, а, у деяких випадках – також підтримкою високого показнику частоти слідування ψ кадрів (до 60 і більше к/с) є недостатнім.

У свою чергу, VR-симуляція сьогодні поступово набуває все більше застосунків у багатьох сферах, зокрема [4]:

1. Сфера розваг:
 - VR-квести та геймінг;
 - віртуальні подорожі.
2. Бізнес-сфера:
 - VR-переговори;
 - демонстрація дослідних зразків продукції;
 - огляд проектів об'єктів будівництва;
 - шоуруми об'єктів нерухомості тощо.
3. Сфера навчання:
 - навчання послідовності дій у виробничих процесах (розбір та ремонт двигуна автомобіля або літака);
 - тренування особового складу спецпідрозділів;

- різноманітні тренінги та майстер-класи;
- підготовка медичних працівників хірургічного профілю тощо;
- тренування для дій на випадок надзвичайних станів (землетрус, пожежа, повінь та ін.).

Розглянемо для прикладу ряд за стосунків VR-симуляції.

У випадку, коли мова йде про VR-симуляцію, де передбачається, головним чином, перегляд або самого середовища (об'єкти нерухомості, віртуальна подорож), або одного, чи цілого ряду ключових об'єктів у віртуальному середовищі, забезпечення взаємодії «користувач-середовище» сьогодні забезпечується одночасно за рядом інтерфейсів зворотнього зв'язку та систем сенсорики.

Так, станом на 2019 рік забезпечується взаємодія на базі таких каналів, як:

- аудіоряд;
- візуальний ряд;
- міостимуляція.

Тут під аудіорядом мається на увазі звуковий відгук системи та/або її окремих об'єктів та ті чи інші дії користувача, що здійснюються відносно них або за їх участю.

Також аудіоконтент можуть генерувати об'єкти системи, якщо сценарій середовища допускає також існування даного процесу незалежно від дій користувача.

У свою чергу, міостимуляція слугує для імітації тактильної взаємодії користувача з об'єктами системи (рис.1.6).

Для цього найчастіше використовуються:

- інформаційні рукавички (data gloves);
- костюм віртуальної реальності.

Даний канал взаємодії є надзвичайно важливим у будь-яких VR-сервісах, що передбачають з боку користувача не лише пасивне спостереження за середовищем та його складовими частинами, а також безпосередню дію (рис.1.7).

Це є справедливим як для ігрових додатків, так і для додатків, що спрямовані на забезпечення процесів навчання, а також медичних систем, які зараз орієнтовані на відновлювальну терапію та ряд супутніх програм.



Рисунок 1.6 – Приклад імітації тактильної взаємодії з об'єктами віртуальної реальності



Рисунок 1.7 – Реалістичний VR-тренажер для навчання пожежників

Врешті решт, візуальний ряд у системах VR-імітації створюється на базі комбінації відео потоків від деякої кількості сферичних камер високої роздільної здатності, або моделюється у 3D-просторі.

Незалежно від способу реалізації, візуальний ряд буде створювати найбільше навантаження $g(t)$ на мережу, яке може сягати рівня, що наближено відповідає потоку роздільної здатності 6-8К. Залежно від частоти слідування кадрів, це може складати сотень Мбіт/с.

Окрім необхідності забезпечення високої роздільної здатності, що потрібно для підтримки ефекту присутності для VR-імітації, прийнятна частота Ψ слідування кадрів має бути також значно вищою, ніж 30, що є стандартизованим для сучасних стандартів кодування відео.

Це пояснюється тим, що вестибулярний апарат людини пристосований до середовища, де зміна візуального оточення відбувається не за дискретними законами.

Відповідно, чим вищим буде даний показник, тим реалістичнішим буде віртуальне середовище.

У той же час, для зміни середовища або окремих його об'єктів відповідно до дій користувача, тобто, для взаємодії з VR-сценою, застосовуються [3, 4]:

1. 3D-контролери (замість 2D-контролерів у реальному середовищі – маніпуляторів типу «миша», сенсорних панелей і т.д.).
2. Системи детектування руху (motion tracking), що фіксують будь-які рухи тіла та повторюють їх у віртуальному середовищі.

Таке детектування може здійснюватися з використанням системи датчиків та/або додаткових відеокамер, що направлені на людину.

3. Системи айтрекінгу (eye tracking). Дані системи відстежують рух зіниць очей та дозволяють визначити, у яку точку простору людина дивиться у поточний момент.

Сьогодні системи VR-імітації, що використовують айтрекінг, використовуються головним чином у спеціалізованих медичних проектах та для наукових досліджень.

Таким чином, для відео 360° у випадку організації даної послуги на базі віддаленого VR-серверу головною вимогою є достатня пропускну здатність S мережі, що забезпечує доставку даних відповідно до вимог, які висуваються до інтерактивного відео, а саме [3, 7]:

- рівень Δp втрат пакетів даних у ході трансляції – менше 1 %;
- затримка доставки τ пакетів – менше 150 мсек.

Тоді для випадку, якщо для такого відео роздільна здатність та частота Ψ слідування кадрів будуть у підсумку створювати навантаження $g(t)$, що не перевищує 5-10 Мбіт/с.

У таких умовах означені вимоги може бути легко забезпечено для стандартизованого спільного розподіленого середовища на базі Fast Ethernet або WiFi чи LTE-A.

У той же час, мережеві додатки VR-імітації, у першу чергу ті, що створюють «ефект занурення» (ефект присутності) мають відповідати суттєво жорсткішим вимогам.

По-перше, хоча з точки зору вимог якості такі сервіси умовно прирівнюються до інтерактивного відео, водночас, згадані вище показники втрат Δp пакетів та затримок τ мають забезпечуватися за умов, коли:

- навантаження $g(t)$ на мережу, що створюється сервісом, може нерідко перевищувати відповідний показник для сервісів відео 360⁰ при тому, що пропускна здатність мережі лишається незмінною;

- окрім візуального ряду необхідно транслювати дані інших типів – аудіо ряд, сигнали тактильної взаємодії (міостимуляції), дані від трекерів руху, айтрекерів та ін.;

- дані усіх інтерфейсів зворотнього зв'язку мають транслюватися синхронно; бідь-які взаємні відхилення у часі, що є помітними для користувача, нівелюють ефект присутності.

У свою чергу, більшість VR-додатків ще одного типу, а саме – VR-заходи (VR-events), є усередненим рішенням по відношенню до VR-імітації та відео 360⁰ за ступінню складності реалізації та вимогами до якості.

Найчастіше на базі додатків даного типу у віртуальному середовищі проводяться:

- конференції;
- корпоративні заходи;
- презентації бренду/продукції;
- концерти;
- нетворкінг.

У загальному випадку VR-event потребує:

- моделювання оточення на базі відео 360⁰, або шляхом створення його 3D-моделі;

- наявності засобів зворотнього зв'язку аналогічно випадку VR-симуляції, але з суттєво нижчими технологічними вимогами до них; при цьому, ряд засобів може бути не обов'язковими, зокрема, ті, що забезпечують канал тактильної взаємодії;

- реалізації окремого специфічного модулю – віртуального аватару користувача (рис.1.8);

- присутності спікера заходу, що, у сутності, може бути прирівняним до звичайного користувача сервісу з тією відмінністю, що 3D-рендерінг спікера апріорі виконується з високою якістю відображення.

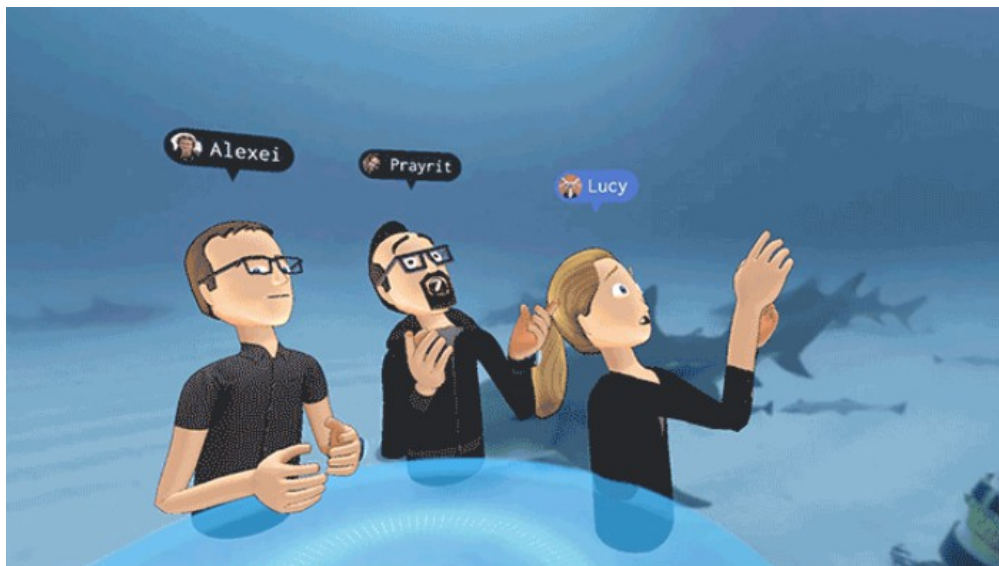
Отже, підсумкове навантаження $g(t)^{(\text{sum})}$ на мережу у цьому випадку сервісів віртуальної реальності також може бути суттєвим за рахунок діє таких чинників, як:

- високої динаміки змін сцен відеоряду, породжуваної анімацією оточення та аватарами учасників, що може спричинюватися їх великою кількістю у віртуальному заході;

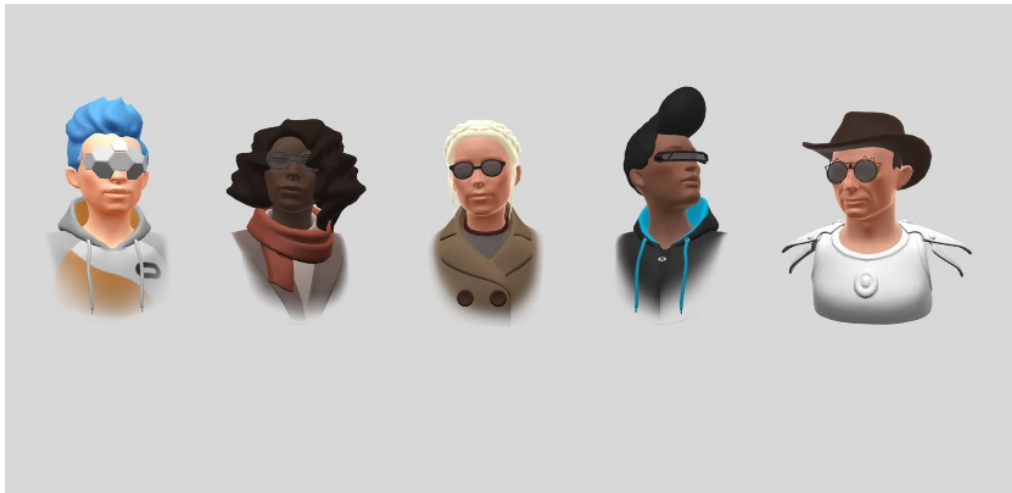
- значного рівня частоти слідування кадрів, якщо необхідно забезпечити максимальне наближення до природної динаміки рухів.

Таким чином, реалізація сервісів віртуальної реальності на базі інформаційно-комунікаційної мережі (спеціалізований сервер VR, хмарне рішення на базі універсальної платформи) передбачає одночасне передавання даних таких типів, як:

- аудіо;
- відеоряд;
- дані пристроїв трекінгу;
- команди тактильної взаємодії.



a)



б)

Рисунок 1.8 – Приклади віртуальних аватарів учасників VR-заходу у динаміці (а) та на стадії налаштування (б)

Підсумковий потік даних у напрямку «користувач-середовище», у цьому випадку, містить у своїй структурі одночасно усі зазначені типи інформації, звернення до яких здійснюється на базі зчитування та аналізу його метаопису (рис. 1.9).



Рисунок 1.9 – Приклад структури інформаційного потоку сервісу VR-симуляції

У підсумку, функціонування системи у цілому, та окремо – забезпечення якості сервісу напряму залежить від синхронності та своєчасності доставки пакетів даних кожного зі згаданих вище типів.

Зокрема, для систем VR-імітації та деяких VR-заходів (якщо за умовами проведення висунуто відповідні вимоги) ефект присутності буде зруйновано за відсутності чіткої синхронізації хоча б за одним з каналів взаємодії.

Далі виконаємо аналіз умов, за яких виникає порушення своєчасності доставки пакетів даних у рамках сервісів VR. Для цього розглянемо характеристики композитного потоку даних сервісу віртуальної реальності.

Так, відсоткове співвідношення даних різних типів у потоці ілюструє рис.1.10 [3].

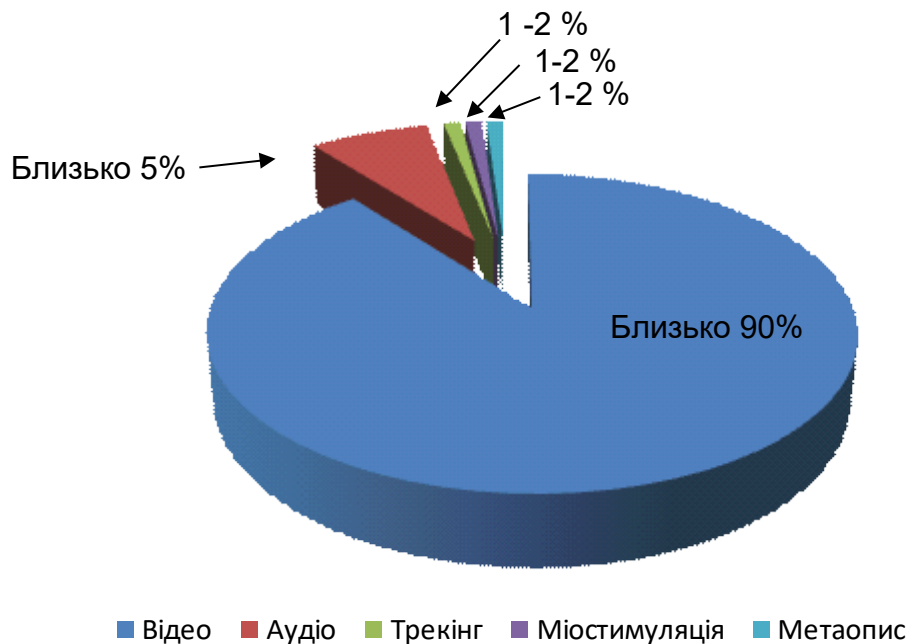


Рисунок 1.10 – Відсоткове співвідношення даних різних типів у композитному потоці сервісу VR

При цьому, як вже було зазначено, дані мають надходити в обох напрямках синхронно за архітектурою сервісу. Тобто, якщо хоча б якась частина даних надійде з запізненнями, або частковою втратою вище деякого гранично допустимого значення, це вплине на функціонування процесу взаємодії.

У той же час, стосовно передавання мережею, до даних типів інформації висовуються різні вимоги у рамках системи QoS, зокрема [7]:

- аудіодані у ряді випадків оцінюються за вимогами VoIP, тобто, потребують забезпечення величини затримки на рівні $\tau < 150$ мсек та показника втрат $\Delta p < 0,25\%$;

- для інтерактивних відеоданих має забезпечуватися менше 1 % втрачених пакетів при величині затримки доставки пакетів менше ніж 150 мсек;

- для даних трекінгу та команд міостимуляції необхідно мінімізувати величину втрат та затримок доставки, в ідеальному варіанті реалізації максимально наблизивши їх до 0.

Разом з тим, рішення цього завдання має цілу низку проблемних питань, що пов'язано з протиріччями між вимогами до якості різних типів трафіку за QoS та рівнем навантаження (інформаційною інтенсивністю), який створюється на мережу кожним з них. При цьому, таке навантаження повністю відповідає діаграмі на рис. (1.10).

Таблиця 1.1 – Сутність протиріч між вимогами до якості різних типів трафіку за QoS та рівнем навантаження, який створюється на мережу кожним з них

Тип трафіку	Вимоги до затримки та рівня втрат	Вимоги до пропускної здатності мережі	Сутність протиріччя
Дані трекінгу	$\rightarrow \min$	$\approx 10-100$ Кбіт/с	Пріоритет на рівні з іншими типами трафіку у VR
Команди міостимуляції	$\rightarrow \min$	$\approx 10-100$ Кбіт/с	Пріоритет на рівні з іншими типами трафіку у VR
Відеоряд	$\tau < 150, \Delta\rho < 1\%$	$\geq g(t)_i^{(\max)}$	Вищий рівень допустимих помилок відносно аудіо ряду, з необхідністю гарантування полоси пропускання
Аудіо	$\tau < 150, \Delta\rho < 0,25\%$	$\geq g(t)_i^{(\max)}$	Необхідність мінімізації втрат без зміни пріоритету та вимог до полоси пропускання
Дані метаопису потоку	$\rightarrow \min$	$\approx 10-100$ Кбіт/с	Пріоритет на рівні з іншими типами трафіку у VR

У табл.1.1 величина $g(t)_i^{(\max)}$ відповідає найвищому рівневі інформаційної інтенсивності потоку, розрахованому за його наявних технічних показників (розмір кадру, частота слідування кадру тощо).

Аналіз табл.1.1 та загальних засад архітектури сервісів VR свідчить про те, що:

- попри не найвищі вимоги щодо рівня втрат та затримок, відеоряд вимагає найбільшої гарантованої полоси пропускання мережі;

- за умов необхідності синхронного надходження різних сегментів потоку, кожен з типів трафіку має обслуговуватися як дані найвищого пріоритету, тобто, виникає ситуація невизначеності, що створює умови для можливості втрат певної частини пакетів у випадку переповнення буферів мережевих пристроїв.

Таким чином, для створення умов, у яких може бути забезпечено надання послуг VR-симуляції та VR-events, доцільним є скорочення рівня навантаження, що створюється на мережу.

1.3 Інтерактивне відео

Сфера використання сервісів інтерактивного відео сьогодні розширюється, і, окрім вже звичних сервісів відео конференцій додатково включає у себе [8, 9]:

- сервіси та додатки презентаційного характеру;
- інформаційні продукти (у т.ч. навчального спрямування);
- ігрові сервіси та ін.

На відміну від сервісів потокового мовлення, трафік відео у випадку інтерактивних сервісів передбачає передавання пакетів відео щонайменше у двох напрямках - на випадок 2 учасників інтерактивної взаємодії.

Відповідно, за умови, що рівень навантаження, яке створює на мережу i -е джерело інтерактивної відеовзаємодії, дорівнює $g(t)_i$, а $(i+1)$ -е джерело - $g(t)_{i+1}$, сумарне навантаження $g(t)^{(\text{sum})}$ у довільний момент часу t у ході сеансу інтерактивного відео може бути визначено як:

$$g(t)^{(\text{sum})} = g(t)_i + g(t)_{i+1}; \quad (1.4)$$

$$g(t)_{i+1} = \overline{g(t)_{i+1}^{(\text{max})}; g(t)_{i+1}^{(\text{min})}};$$

$$g(t)_i = \overline{g(t)_i^{(\text{max})}; g(t)_i^{(\text{min})}},$$

де $g(t)_i^{(\text{max})}$ та $g(t)_{i+1}^{(\text{max})}$ - максимальні значення навантаження (величини інформаційної інтенсивності), що створюють на мережу i -е та $(i+1)$ -е джерела;

$g(t)_i^{(\text{min})}$ та $g(t)_{i+1}^{(\text{min})}$ - мінімальні значення навантаження (величини інформаційної інтенсивності), що створюють на мережу i -е та $(i+1)$ -е джерела.

У виразі (1.4) замість фіксованих значень навантажень вказано ймовірні діапазони значень. Така закономірність пояснюється особливостями формування трафіку відео у MPEG-парадигмі, за якою за замовчуванням кодування здійснюється зі змінною інформаційною інтенсивністю, що зумовлено наступним [10]:

- трафік формується покадрово, при цьому, залежно від характеру змісту, інтенсивність $g(F)_i$ i -го кадру може значно різнитися з величиною $g(F)_{i+1}$ для $(i+1)$ -го кадру;
- потік відео формується на основі В, Р та І-кадрів, що архітектурно можуть вносити відмінні рівні навантаження (рис. 1.11).

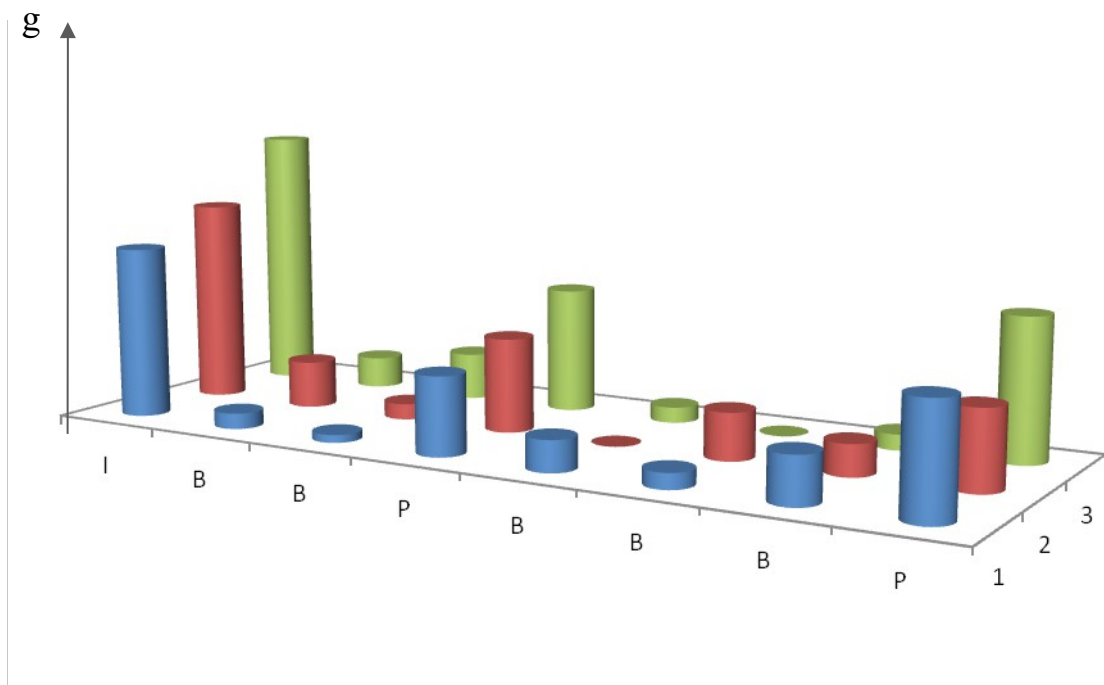


Рисунок 1.11 – Внески у величину інформаційної інтенсивності відеопотоку, що створюють кадри типів І, Р та В трьох груп

Отже, тоді за умови, що у ході одного сеансу інтерактивної взаємодії, наприклад, відеоконференції, одночасно задіяно n учасників, тобто, n джерел, загальне навантаження, яке при цьому створюється на мережу, аналогічно до виразу (1.4) може обчислюватися на базі виразу:

$$g(t)^{(\text{sum})} = \sum_{i=1}^n g(t)_i \quad (1.5)$$

Відповідно, на випадок, коли усі учасники генерують окремі відео потоки роздільної здатності 1920x1080 (FullHD) при 30 к/с, при $n=2$ $g(t)^{(\text{sum})} = \overline{3;10}$ Мбіт/с, при $n=5$ - $g(t)^{(\text{sum})} = \overline{7,5; 25}$ Мбіт/с і т.д.

Така закономірність справедлива також для сервісів віртуальної реальності, що відносяться до типів VR-events та VR-симуляції (для випадків, коли у сеансі взаємодії з віртуальним середовищем та один з одним одночасно бере участь кілька користувачів).

Разом з тим, для забезпечення функціонування сервісів інтерактивного відео у загальному випадку достатньо забезпечити передавання пакетів з підтримкою рівнів втрат та затримок, що не перевищують рекомендовані QoS значення, оскільки тут не використовується інших каналів взаємодії окрім аудіо та відео.

У той же час, беручи до уваги:

- зростання попиту на відеосервіси, зокрема, сегменту інтерактивного відео;
 - простоту реалізації каналу інтерактивного відео;
 - зростання навантаження, що породжує одне джерело відеоданих, з темпами, що перевищують ріст пропускної спроможності мережі [1, 2];
- можна зазначити, що складаються умови, у яких надання сервісів з бажаним рівнем якості унеможлиблюється.

Відповідно, означена проблематика як для VR, так і сервісів інтерактивного відео є спільною, таким чином, шляхи її подолання також є аналогічними.

2 ПІДХОДИ ЩОДО ЗМЕНШЕННЯ МЕРЕЖЕВОГО НАВАНТАЖЕННЯ, ЩО СТВОРЮЮТЬ МУЛЬТИМЕДІЙНІ ВІДЕОСЕРВІСИ

2.1 Загальні принципи підходів до скорочення навантаження за рахунок зменшення у мережі кількості пакетів мультимедійних сервісів

За показником надлишковості даних, такі типи інформації, як відео та звук, суттєво перевищують інші типи.

Відповідно, якщо для відео та звукових файлів втрата певної частини Δg початкових даних у багатьох випадках може взагалі не вплинути на їхню семантичну цілісність (тобто, змістовне наповнення), для файлів інших типів це буде критичним та приведе до повного їх руйнування [10]. Зокрема, це справедливо для:

- виконуваних файлів (EXE, BAT, COM, MSI тощо);
- ряду текстових файлів з внутрішнім форматуванням (DOC, CPP, JS, LIB, DLL та ін.);
- файлів, попередньо стиснених на базі одного з алгоритмів пакування (ARJ, 7zip, RAR).

Така закономірність дозволяє, з одного боку, тієї чи іншою мірою (а у ряді випадків – суттєво) зменшити інформаційну інтенсивність $g(t)$ даних означених типів, тим самим зменшуючи мережеве навантаження та за рахунок цього сприяючи зростанню якості мережевих сервісів.

З іншого боку, якщо говорити про відео, результатом втрати деякої частини Δg вихідної інформації є неможливість реконструювати початкові відеокадри з потрібним рівнем Y якості. Тобто, при цьому якість сервісу також буде на недостатньому рівні.

Відтак, необхідно визначити межу, нижче якої незворотне скорочення об'єму початкової інформації відеоданих недопустиме.

Такий підхід дозволяє:

- ефективно скоротити рівень інформаційної інтенсивності $g(t)$ відео;
- забезпечити можливість внесення рівня похибки M , за якої рівень якості Y буде не нижчим, ніж вимагають умови надання сервісу.

Зазначене є еквівалентним виразу:

$$\begin{cases} g'(t) \rightarrow g(t)^{(\min)}; \\ Y \geq Y_{\text{set}}, \end{cases} \quad (2.1)$$

де Y_{set} - рівень якості, встановлений за умовами надання сервісу;

$g'(t)$ - інформаційна інтенсивність відеоданих, отримана шляхом незворотнього зменшення обсягу вихідних даних.

Тут вираз (2.1) може розглядатися у якості загальних умов, за яких мережеве навантаження зменшується з одночасним забезпеченням якості мережевих сервісів.

Водночас виникає дилема, що викликана протиріччям відносно значення отриманої величини $g'(t)$ у наслідок задіяння механізмів незворотного скорочення об'єму початкової інформації відеоданих з одного боку [10, 11-13]. З іншого боку – відповідності отриманої величини $g'(t)$ фактичному необхідному значенню, при якому нерівність (2.1) буде мати сенс. Це відповідає випадку, коли значення $g'(t)$ не перевищує рівня поточної пропускної спроможності мережі $R(t)$.

На цей випадок умова (2.1) буде уточнена та подана у вигляді:

$$\begin{cases} g'(t) \leq R(t); \\ Y \geq Y_{\text{set}}. \end{cases} \quad (2.2)$$

Разом з тим, якщо шляхом скорочення деякої частини вихідних даних досягти виконання умов (2.2) не вдається, мають бути задіяні додаткові механізми обробки відеоданих.

При цьому, оскільки попередньо деяке викривлення у вихідні дані було внесено, додаткові механізми повинні сприяти подальшому скороченню $g'(t)$ без втрат. Далі розглянемо існуючі методи скорочення інформаційної інтенсивності.

2.2 Механізми зменшення інформаційної інтенсивності відеоданих

З усієї існуючої множини технологій та методів, спрямованих на ущільнення, тобто, скорочення інформаційної інтенсивності відеоданих, статус стандартів та широку застосовуваність мають:

- технології, що відносяться до сімейства MPEG;

- технології M-JPEG та M-JPEG2000;
- фрактальні алгоритми.

Якщо говорити про відеодані інтерактивного типу а також про сервіси VR на базі інфокомунікаційної мережі, зазначимо, що для обробки трафіку відео таких сервісів сьогодні використовуються технології кодування MPEG [13-18]. Це пояснюється тим, що:

- фрактальні алгоритми не гарантують стабільного коефіцієнту K стиснення відеоданих, а за певних умов обробка одного кадру може займати значний обсяг часу;
- M-JPEG, M-JPEG2000 та їм подібні забезпечують високу якість відновлених даних, проте для сцен з високим рівнем динаміки дозволяють отримати коефіцієнт стиснення K , у 5-10 разів нижчий, ніж для MPEG, з тієї причини, що алгоритми не усувають часової надмірності відео.

Для того, щоб зробити висновок відносно потенційних можливостей платформи MPEG щодо забезпечення якості надання послуг VR та інтерактивного відео в умовах збільшення їх кількості, а також постійного зростання їхньої інформаційної інтенсивності, виконаємо аналіз її функціонування.

2.2.1 Принцип функціонування платформи MPEG

Базис даної технології кодування ґрунтується на скороченні просторової та часової надмірностей [13].

Часова надмірність опису відеопотоку зумовлена тим, що для ряду сусідніх кадрів, що надходять один за одним, деякі ділянки внутрішнього змісту можуть повністю або частково співпадати. Відповідно, на такий випадок має сенс визначити подібні ділянки для того, щоб у послідовності подібних кадрів перший з них надсилати повністю, а для інших передавати виключно фрагменти, що відображають зміни відеосцени, за рахунок чого додатково скоротити інформаційну інтенсивність $g(t)$ відеопотоку.

У свою чергу, надмірність просторового опису сприяє зменшенню кількості біт для представлення кожного окремого кадру у потоці.

Обробка відео потоку у такій моделі кодування розпочинається з етапу **формування кадрів** у межах груп (GOP). При цьому, група містить у собі один опорний (intra), або I-кадр, та фіксовану кількість B та P-кадрів, що слідують один за одним у певній послідовності (рис.1.11). При цьому, оскільки I-кадр містить повну відеосцену, а P та B-кадри – ділянки змін, відповідно їхня інформаційна інтенсивність може бути суттєво нижчою, за рахунок чого

загальна величини $g(t)$ для відео потоку скорочується. Кількість кадрів P та B типів, розмірність групи та особливості її формування регламентуються певними стандартами у межах MPEG.

Далі, незалежно від типу, кадри обробляються на базі стандарту JPEG, який являє собою каскад технологічних етапів, у ході яких відносно вихідних даних виконується послідовність дій, спрямованих на скорочення їхньої інформаційної інтенсивності [13, 16, 17].

У загальному випадку, такі технологічні етапи може бути віднесено до:

- тих, що вносять незворотні зміни (кодування з втратами);
- сервісних (підготовчих);
- кодування без втрат.

При цьому, до етапів, що вносять незворотні зміни відносяться:

- вибір формату колірної субдискретизації;
- переведення фрагментів кадру у спектральний простір опису;
- квантування з подальшим округленням значень компонент.

У свою чергу, етап кодування RLE та фінальний етап обробки - етап статистичного кодування, забезпечують стиснення даних без втрат.

До підготовчих, або сервісних, належать такі етапи, як:

- конвертація колірного опису відеокадрів;
- розбиття вихідного кадру на фрагменти 8×8 пікселів (сегментація);
- лінеаризація компонент;

переведення фрагментів кадру у спектральний простір опису; даний етап одночасно вважається як сервісним, так і таким, що вносить незворотні зміни.

Стосовно етапу переведення фрагментів кадру у спектральний опис зазначимо, що у ході даної операції здійснюється трансформування змісту сегменту на базі дискретно-косинусного (у ряді випадків – дискретно-синусного) перетворення. Оскільки зазначені функції не є цілочисельними, відповідно, значення пікселів у просторовому представленні, які буде отримано у наслідок зворотнього перетворення, не будуть повністю відповідати початковим.

Перелік технологічних етапів кодування за JPEG, та операції, що виконуються на кожному з них, наведено рис.2.1 [14, 17].

Розглянемо їх детальніше.

Так, *етап конвертації колірного опису* відеокадрів, як один з підготовчих, подає множину пікселів, початково поданих у форматі RGB, у яскравісно-хроматичному описі. Тут замість каналів червоного, зеленого та

синього відтінків буде використано одну компоненту яскравості, та дві колірні-різницеви, або хроматичні – хроматична червона та хроматична синя, таким чином маємо – YCrCb.

Така зміна колірного опису пікселів виконується з міркування, що зір людини значно точніше розрізняє зміни яскравості, ніж колірних відтінків. Тому далі, у ході колірної субдискретизації, канал яскравості буде залишено незмінним, а канали CrCb може бути проріджено, за рахунок чого на даному етапі може бути відкинута до 3/4 початкової інформації про колірність кадру, тим самим зменшивши інформаційну інтенсивність $g(t)$ кадру.

У свою чергу, **етап сегментації кадру** дозволяє виділяти менший обсяг пам'яті кодеру для обробки, ніж для випадку подальшого кодування кадру, як єдиного об'єкту.

Наступний етап – етап ортогонального перетворення, за рахунок зміни просторового представлення сегменту у спектральний, готує компоненти до етапу квантування. Головна закономірність такого перетворення полягає у тому, що у результаті нього здійснюється групування компонент за значенням амплітуд. Так, максимальні амплітуди у межах трансформованого сегменту відповідають компонентам низькочастотної (НЧ) області. При цьому, по мірі зростання частоти, амплітуди зменшуються. Тобто, основна частина енергії концентрується у межах ЧН-області.

Закономірність розподілу амплітуд компонент використовується на наступному етапі обробки – у ході **етапу квантування**. Тут здійснюється поелементне ділення компонент сегменту на коефіцієнти матриці квантування. При цьому, для найбільш значимих компонент (з НЧ-області) використовуються найменші величини дільників. По мірі збільшення частотності компонент, відповідні дільники збільшуються. Таким чином, ВЧ-область буде ущільнена найбільшим чином. У підсумку тут утвориться велика кількість компонент, величини яких наближаються до 0. Це дозволяє на наступному етапі – **етапі округлення** замінити їх значення 0.

Таким чином, у областях ВЧ на частково СЧ утворюється велика кількість ланцюжків, що складаються з нульових значень. Відповідно, це дозволяє після виконання **етапу лінеаризації**, у ході якого матриця 8×8 за рахунок операції зиг-заг сканування перетворюється у вектор з 64 елементів,

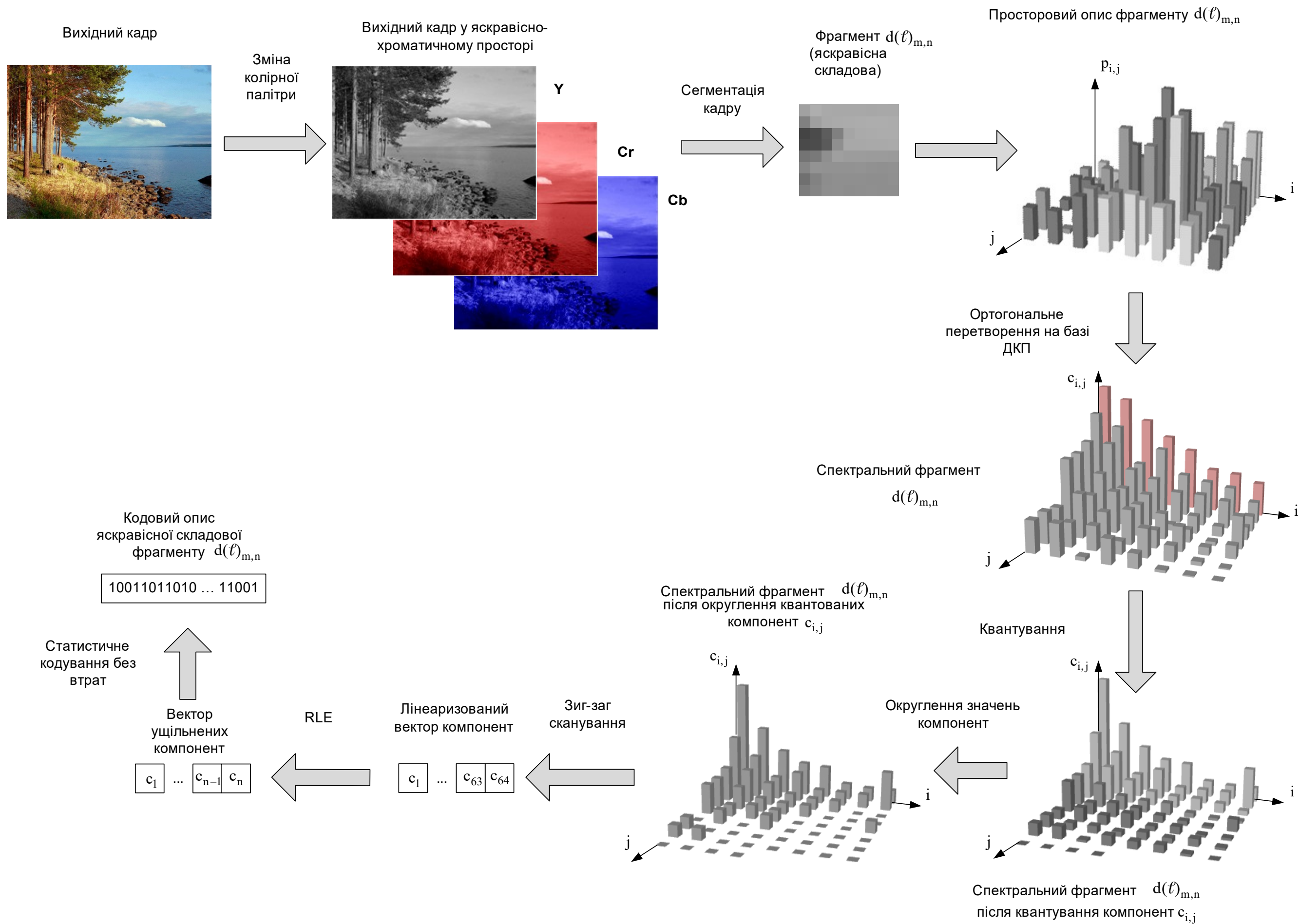


Рисунок 2.1 – Загальна схема виконання перетворень за алгоритмом JPEG фрагментів відеокадру у складі базису MPEG

ділянки, що містять нульові значення, подати у стислому вигляді за рахунок **кодування RLE**. Тут формуються послідовності парних елементів загальної структури (нуль, число), де «нуль» - лічильником нульових елементів, а «число» — величина, що записується у наступну чарунку.

Наприклад:

вихідний вектор 55 3 0 0 0 -7 0 0 0 0 5

вектор після RLE (0,55) (0,3) (3,-7) (4,5).

Нарешті, на етапі **кодування без** втрат здійснюється формування коду фрагменту $d(\ell)_{m,n}$.

За наведеною схемою функціонують усі технології сімейства MPEG, у т.ч. найбільш на сьогодні продуктивні, що мають ряд нововведень відносно зазначеного.

2.3 Технологія H.265/HEVC

На сьогодні найбільш ефективною з позиції використання для зберігання стиснених відеоданих, або потокової та інтерактивної передачі може вважатися рекомендація ITU-T H.265 або HEVC (англ. High Efficiency Video Coding - високоефективне кодування відеозображень). Дана рекомендація є подальшим розвитком існуючих стандартів з кодування відео (H.261, H.262, H.263 та H.264). Рекомендація була розроблена у зв'язку зі зростаючою потребою у забезпеченні більш високої якості стиснених відеоданих для різних додатків [19].

Це дозволило, з одного боку, збільшити коефіцієнт K стиснення з забезпеченням ефективнішого показника PSNR. З іншого боку, в умовах, коли незначне зменшення PSNR не є суттєвим, відповідно, може бути досягнуто збільшення величини K . У середньому вважається, що для H.265/HEVC $K \approx 80$.

Так само як і у випадку всіх попередніх стандартів відеокодування ITU-T і ISO/IEC, у HEVC стандартизовані тільки відеопотік і синтаксис [15, 19], та декодування кадрів.

При цьому у стандарті H.265 High Efficiency Video Coding (HEVC), аналогічно стандартам ITU-T H.261, MPEG2, MPEG4 Part10 / H.264 AVC застосовується гібридний підхід кодування відео на рівні блоків.

Гібридний підхід полягає в усуненні часової статистичної надмірності шляхом застосування міжкадрового (inter) прогнозування, а також усунення

просторової надмірності шляхом внутрішньокадрового (intra) прогнозування. Також застосовується кодування з перетворенням прогнозованих сигналів для подальшого усунення статистичної надмірності.

У загальному вигляді, структура і алгоритм роботи кодера H.265, аналогічні прийнятим в стандарті H.264, та відповідають схемі MPEG.

Так, має місце застосування механізмів класифікації кадрів на I, P та B типи з розбиттям потоку кадрів на групи, що спрямовано на скорочення міжкадрової, або часової, надмірності.

У свою чергу, обробка окремих кадрів виконується за модифікованим сценарієм JPEG-кодування, тобто, містить такі технологічні етапи, як [17]:

- перетворення колірному простору та вибір формату колірної субдискретизації;
- сегментація кадрів на блоки pхп;
- ортогональне перетворення блоків кадру, для переведення у спектральний формат опису;
- квантування компонент спектрального опису та подальше їх округлення;
- кодування без втрат лінеаризованої множини компонент.

2.3.1 Особливості стандарту H.265 HEVC

У рамках стандарту утілено ряд поліпшень технологічного процесу кодування, що в кінцевому підсумку дає значне збільшення якості кодованих даних при тих же величинах коефіцієнта K стиснення. Зокрема, порівняння однакових відеопослідовностей, кодованих за різними стандартами у схемі MPEG показує, що для однакових значень пікового відношення сигнал/ шум забезпечується [19]:

- зниження інтенсивності порядку на 35% порівняно зі стандартом MPEG4 Part10 / H.264 / AVC (високий профіль);
- зниження інтенсивності на 70% порівняно з технологією MPEG2 (основний профіль).

Однією з відмінностей H.265 від попередньої рекомендації є те, що на вхід кодера HEVC надходить відео з прогресивною розгорткою. У HEVC не представлено явних функцій кодування черезрядкової розгортки через обмеження на її підтримку сучасними пристроями відображення.

Крім того, на відміну від H.264, у H.265 замість макроблоків використовуються секції з деревовидної структурою (CTU - coding tree unit),

розмір яких вибирається кодером у залежності від вимог відповідних додатків (наприклад, затримка кодування / декодування) і вимог до пам'яті, а так само архітектури побудови кодера, та може мати більший розмір, ніж традиційні макроблоки у стандарті H.264.

Використання секцій і блоків більшого розміру підвищує ефективність кодування зображень з високою роздільною здатністю, а також збільшує швидкість декодування, проте при цьому зростає обчислювальна складність.

Кожна секція з деревовидної структурою складається з трьох деревовидних блоків окомпонент відліків (СТВ - coding tree block): яскравості і двох відповідних колірно-різницевих, а також елементів синтаксису.

При розмірах $N \times N$ блоку яскравості компонент, відповідні колірно-різницеві блоки можуть бути обрані рівними $\frac{N}{2} \times \frac{N}{2}$ при $N = 16, 32$ або 64 . При цьому, зі збільшенням розмірів блоків спостерігається збільшення ступеня стиснення.

Деревовидні блоки компонент (яскравості і хроматичні) потім можуть бути розділені на кілька блоків кодування (СВ - coding block).

Секція з деревовидної структурою при цьому містить синтаксис квадродерева, який підтримує розбиття на блоки з перемінним розміром у залежності від характеристик тієї області, яка охоплена деревовидним блоком компонент.

Відповідний процес показаний на рис. 2.2. При цьому, розмір блоків кодування може змінюватися від максимально великого блоку, рівного деревовидному блоку компонент, до мінімального - розміром 8×8 . Цей процес описується елементом синтаксису і передається декодеру.

У свою чергу, яскравісний блок і колірно-різницеві блоки, спільно з відповідним синтаксисом, формують секцію кодування (CU - coding unit).

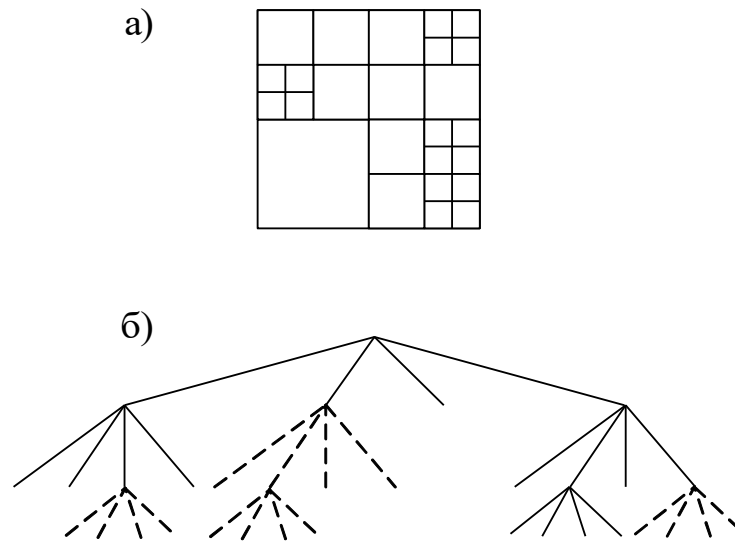


Рисунок 2.2 – Поділ деревовидного блоку на блоки кодування і блоки перетворення; а) розбиття на деревовидній блок, б) відповідне квадродерево

На рис. 2.2 суцільними лініями показані межі деревовидних блоків, для яких застосовується внутрішньо- або міжкадрове кодування. У свою чергу, пунктирні лінії відповідають межам блоків перетворень (залишкові дані після перетворення).

Далі для кожної секції кодування СУ вибирається режим прогнозування, який може бути як внутрішньокадровим, так і міжкадровим.

При цьому, у процесі вибору внутрішньокадрового прогнозування, до яскравішого блоку кодування СВ може бути застосовано один з 35 просторових внутрішньокадрових режимів прогнозування (Рис. 2.3).

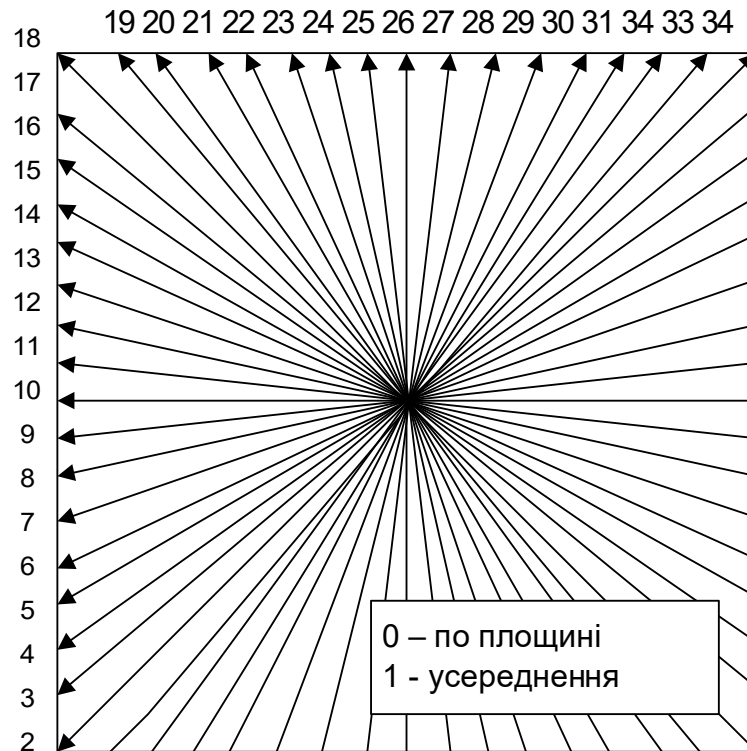


Рисунок 2.3 – Режими і напрямки внутрішньокадрового прогнозування у стандарті H.265

У свою чергу, до обох колірно-різницевих СВ застосовується однаковий режим внутрішньокадрового прогнозування. При цьому може використовуватися режим, аналогічний встановленому для складової яскравості, або застосовуватися окремий режим. Тут може використовуватися горизонтальний, вертикальний, площинний, діагональний вниз-вліво або усереднений (DC) режим прогнозування. При цьому, режим внутрішньокадрового прогнозування застосовується окремо до кожного блоку вихідного кадру.

При використанні міжкадрового прогнозування, виконується розподіл яскравісних і колірно-різницевих блоків на блоки прогнозування (PB - prediction block). При цьому дані блоки можуть бути як рівними вихідним блокам, так і бути розбиті на 2 або 4 блоки передбачення. Далі яскравісного і колірно-різницевого блоків прогнозування, спільно з відповідним синтаксисом, формується секція прогнозування (PU - prediction unit).

У свою чергу, кожна секція PU містить один або два вектора руху для одностороннього або двостороннього кодування з прогнозуванням відповідно. Усі блоки PB, відповідні блокам кодування, можуть мати такий

же розмір, або, як при внутрішньокадровому прогнозі, можуть бути розбиті на два або чотири блоки, розміром $M \times M$, $M/2 \times M$, $M \times M/2$ або $M/2 \times M/2$ (рис. 2.4). Окрім того, може бути використано асиметричне розкладання на блоки міжкадрового прогнозування (AMP - asymmetric motion partitioning), тоді яскравісний блок розміром $M \times M$ може бути так само розбитий на два прогнозованих блоки яскравості. При цьому, один з яскравісних блоків прогнозування буде містити у собі $N \times (N/4)$ або $(N/4) \times N$ компонент, у той час як інший яскравісний блок складатиметься з решти $N \times (3 \times N/4)$ або $(3 \times N/4) \times N$ компонент.

Асиметричне розкладання на блоки міжкадрового прогнозування, так само, відповідно може бути застосовано до двох колірно-різницевих блоків.

Як і для H.2644/AVC [15], у рамках H.265/HEVC підтримується прогнозування вектора руху в "одну чверть" пікселя. Яскравісний сигнал прогнозування обчислюється за допомогою фільтрів 7 або 8 порядків (у залежності від зсуву пікселя). При цьому, для ц колірно-різницевих блоків застосовується інтерполяційний фільтр 4 порядку.

HEVC також підтримує множинні опорні кадри і концепцію I, P, B слайсів, як і у H.264/MPEG4 AVC.

У той же час, стандарті H.265 покращено кодування параметрів руху у порівнянні з попередніми стандартами. Тут застосовується так званий режим змішування (merge mode), в якому кодуються не параметри руху, а обчислюється список ймовірних параметрів (candidate list) руху для відповідної секції прогнозування. В основному, даний список включає у себе параметри руху сусідніх блоків у просторі, а також параметри руху, передбачені у часі, які обчислюються на основі даних про рух зміщених блоків в опорному кадрі. При цьому, вибір набору параметрів руху повідомляється передачею індексу у список можливих кандидатів. Тут використання блоків великого розміру для компенсації руху і режиму змішування дозволяють ефективно передавати дані про рух для великих послідовно розташованих елементів зображення. Якщо секція прогнозування кодується без використання режиму змішування, то передаються відповідні опорні індекси і різниця прогнозованого вектора руху.

У свою чергу, прогнозування виконується за допомогою алгоритму покращеного передбачення вектора руху (AMVP - advanced motion vector prediction). У цьому алгоритмі для кожного вектора руху складається перелік можливих параметрів, який може включати в себе вектори руху сусідніх

блоків з однаковим опорних індексом, а так же вектор руху, прогнозований у часі. Вектор руху кодується передачею індексу у список ймовірних параметрів для визначення обраного прогнозування і кодування вектора різниці.

Після виконання процесу прогнозу, далі виконується переведення даних у спектральний опис на базі блочного перетворення, з використанням дискретно-косинусного або дискретно-синусового перетворення. При цьому, для кодування залишкових даних між- або внутрішньокадрового прогнозування, кодований блок може бути подано як єдиний яскравісний блок, або ж може бути розділений на 4 блоки, рівні за розміром. У свою чергу, ці 4 блоки перетворення можуть бути також далі розділені на чотири менших блоку перетворення (рис.2.4). Аналогічний процес поділу застосовується і до колірно-різницевих блоків (за винятком додаткового розбиття блоків на менші). Такий процес зветься побудовою залишкового квадродерева (residual quadtree (RQT)).

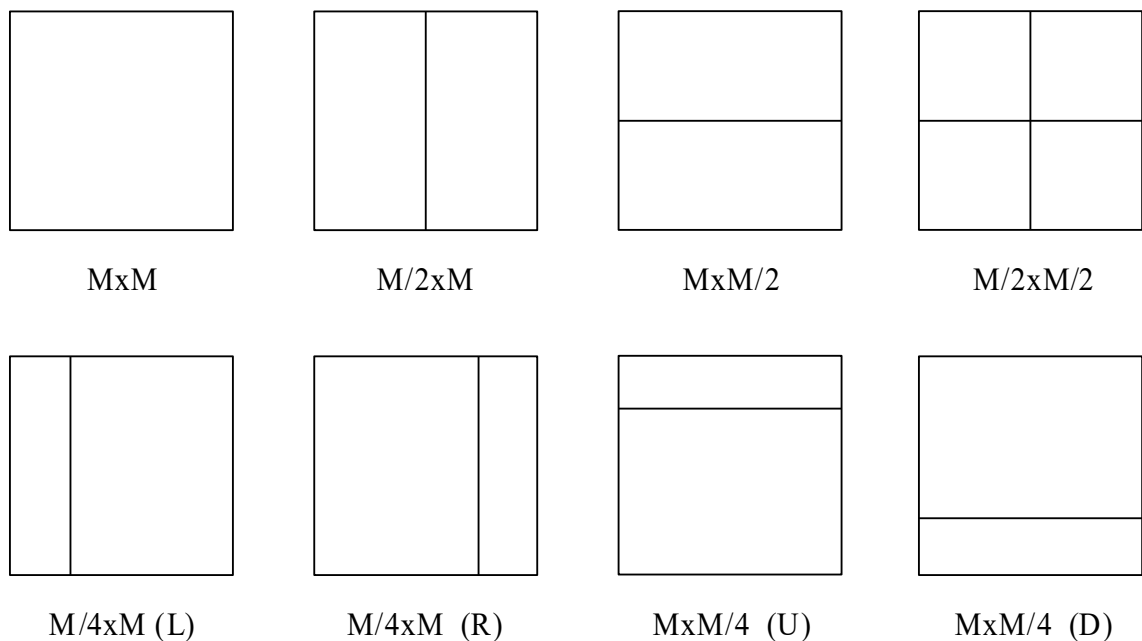


Рисунок 2.4 – Розкладання блоків кодування на блоки прогнозування при міжкадровому прогнозі. (При внутрішньо кадровому прогнозі можуть бути використані тільки режими $M \times M$ і $M/2 \times M$)

Далі яскравісний і колірно-різницеві блоки, що залишилися, разом з відповідним синтаксисом формують секцію перетворення (TU - transform unit). При цьому, для кожного блоку перетворення (яскравості і двох колірно-

різницевих) застосовується процес двовимірного перетворення. У свою чергу, максимальний і мінімальний розмір блоків перетворення вибирається кодером. Усі блоки перетворення у цьому випадку є квадратними блоками з розміром 4x4, 8x8, 16x16 або 32x32 пікселя.

Як і в H.264/MPEG4/AVC, HEVC використовує цілочисельне зворотне перетворення. У загальному випадку, таке перетворення може розглядатися як цілочисельна апроксимація дискретно-косинусного перетворення.

У той же час, HEVC для компонент яскравості внутрішньокадрового блоку розміром 4x4 застосовує альтернативне перетворення на базі цілочисельної апроксимації дискретно-синусного перетворення.

Стандарт H.265 HEVC, на відміну від стандарту H.264/MPEG4 part10/AVC, який підтримує дві схеми ентропійного кодування CAVLC (контекстно-адаптивне кодування кодами змінної довжини) і CABAC (контекстно-адаптивне двійкове арифметичне кодування) містить одну схему кодування CABAC. Дана схема схожа зі стандартом H.264, і містить ряд поліпшень, які полягають у застосуванні більш складної схеми вибору контексту, яка є більш ефективна для великих блоків перетворення. Це дозволяє поліпшити продуктивність і зменшити вимоги до пам'яті. У середньому виграш схеми CABAC відносно традиційних методів статистичного кодування, прийнятих у JPEG – Хафмана, або арифметичного, складає 10%.

У HEVC застосовується схожий з MPEG4/AVC деблокуючий фільтр, який застосовується в ході внутрішньокадрового прогнозування. У той же час, його конструкція дещо спрощена щодо прийняття рішень і процесу фільтрації, та зручніша для паралельної обробки.

Також у структурі HEVC присутня додаткова опціональна операція адаптивного зсуву відліків (SAO - sample adaptive offset), яка застосовується після деблокуючого фільтра для кращого відновлення амплітуд початкових пікселів з використанням значень зміщення з переданої кодером таблиці. SAO є нелінійним фільтром, який може застосовуватися у кожному деревовидному блоці кодування в одному з двох режимів:

- режим зсуву по межах;
- режим зсуву по смузі.

При цьому, режим зсуву по межах застосовується шляхом порівняння значення компоненти з двома з восьми сусідніх в чотирьох напрямках:

- по горизонталі;

- по вертикалі;
- під кутом 45 градусів;
- під кутом 135 градусів.

На основі цього порівняння, компонента класифікується як така, що належить одній з п'яти категорій, а саме:

- мінімум;
- максимум;
- межа, в якій відлік має найменше значення;
- межа, в якій відлік має найбільше значення;
- монотонна область.

На наступному кроці, якщо компонента потрапляє в одну з перших чотирьох категорій, до її амплітуди додається зсув, значення якого береться з переданої кодером таблиці зсувів.

У свою чергу, режим зсуву по смузі застосовується на основі вимірювання значення амплітуди окремої компоненти.

Компонента класифікується за амплітудою до однієї з 32 категорій, так званих смуг. Потім до відліків, що належать чотирьом з 32 послідовних смуг, додається значення зміщення з переданої кодером таблиці.

Використання чотирьох послідовних смуг визначається тим, що в однакових областях, які схильні до спотворень типу "смуги" амплітуди компонент, як правило, розташовані у вузькому діапазоні.

У Н.265 передбачено 3 профіля, а саме:

- основний (Main);
- основний 10 (Main 10);
- основний профіль нерухомих зображень (Main Still Picture). Таке зменшення кількості профілів в першу чергу пов'язано з необхідністю забезпечення можливості взаємодії між різними додатками і пристроями.

2.3.2 Висновки за результатами аналізу технології Н.265

Розглянувши усі нововведення, що було утілено у рамках технології Н.265 відносно попередньої рекомендації Н.264, а також базису MPEG взагалі, можна зробити такі висновки:

- рекомендація Н.265 забезпечує більш гнучке, ніж для випадку попередніх технологічних рішень, розподілення інформаційної інтенсивності між фрагментами відеокадру, що досягається зменшенням розміру

структурних одиниць, на рівні яких виконується обробка, до 4x4 компоненти, тоді як попередні рекомендації таких механізмів не мають;

- аналогічно зазначеному у попередньому пункті, можливість змін розміру структурних одиниць у ході кодування до 32x32, забезпечує збільшення оперативності обробки фрагментів кадру, чим прискорює процес кодування взагалі;

- у рамках рекомендації алгоритмом стиснення, що забезпечує кодування без втрат, є контекстно-адаптивне двійкове арифметичне кодування, ефективність якого залежить від особливостей змісту даних, що надходять на обробку;

- вираш на етапі кодування без втрат становить є у середньому на 10% більшим порівняно з статистичними методами, прийнятими для застосування раніше;

- алгоритм САВАС для ефективного функціонування потребує виконання етапу контекстного моделювання за прийнятими у стандарті алгоритмами; на сьогодні даний етап є досить ресурсоємним.

Таким чином, перевагами даного стандарту є:

- забезпечення досить високої якості Y реконструйованих відеоданих при величинах коефіцієнту K стиснення, що є близькими до максимальних;

- відносно раціональний розподіл інформаційної інтенсивності між фрагментами кадру.

Разом з тим, рекомендації H.265 властиві також такі недоліки:

- залежність між особливостями змісту кадру та рівнем K стиснення, при якому забезпечується потрібний рівень Y якості відновлених даних відео;

- складний алгоритм кодування баз втрат, який при цьому дозволяє збільшити загальний коефіцієнт стиснення незначно;

- вимогливість до обчислювальних ресурсів.

Отже, у наслідок існування виявлених недоліків, у наявній технологічній базі кінцевих пристроїв технологія на сьогодні є обмеженою. Разом з тим, оскільки на базі неї суттєвого збільшення коефіцієнту стиснення без ускладнення алгоритму не досягнуто, використання H.265 як базису для критичних відеосервісів, зокрема, VR та інтерактивного відео, можливе лише за умови обмежених значень роздільної здатності кадрів та невисокої (порядку 30 к/с) частоти їх слідування.

Для того, щоб забезпечити додаткові можливості щодо усунення надмірності просторового опису відеокадрів, розглядається варіант застосування алгоритмів кодування, що забезпечують прогресуюче формування зображень.

Розглянемо один з таких алгоритмів, а саме – SPIHT, для того, щоб зробити висновок про доцільність застосування такого та подібних підходів як одного з ключових механізмів для забезпечення додаткового скорочення інформаційної інтенсивності відео, що сьогодні є актуальним для інтерактивного відео та систем VR.

2.4 Алгоритм SPIHT

2.4.1 Загальні принципи функціонування алгоритму

Алгоритмом, потенційно здатним забезпечити скорочення рівня $g(t)$ інтенсивності відеоданих, є SPIHT (Просторова Впорядковані Ієрархічні Древа) [20]. На базі алгоритму SPIHT існує ряд методів ущільнення відеопотоку, орієнтованих на обробку окремих кадрів. Даний алгоритм відноситься до методів кодування зображень з втратами.

SPIHT відрізняється простотою реалізації, швидкодією і високою ефективністю кодування будь-яких типів зображень.

При цьому, SPIHT дозволяє здійснювати оптимальну прогресуючу передачу зображень. Тут ключовою особливістю алгоритму є те, що на будь-якому етапі декодування якість відображуваного в поточний момент кадру є найкращою для введеного обсягу інформації про даний образ.

Іншою відмінною рисою алгоритму SPIHT є використання вкладеного кодування. Цю властивість можна проілюструвати наступним чином: якщо кодер, що використовує вкладене кодування, генерує два файли - великий, об'ємом $g(t)_{\text{big}}$ біт і маленький, обсягом $g(t)_{\text{small}}$ біт, то менший файл збігається з першими бітами більшого файлу.

Припустимо, що три користувача очікують деякий відправлений ним кодоване зображення. При цьому, кожен з користувачів очікує доставку зображення з різною якістю Y .

Наприклад, перший з них очікує доставку зображення з якістю Y , що відповідає обсягу у 10 кБ для його опису.

У свою чергу, другому і третьому користувачам будуть доставлені зображення з якістю Y , що відповідає 20 кБ та 50 кБ. У цьому випадку

більшість методів обробки зображень з частковою втратою даних буде здійснювати кодування вихідного зображення три рази з різною якістю, для того, щоб згенерувати три різних файли, кожен з яких буде представлений різною кількістю біт. У той же час, алгоритм SPIHT згенерує для цих цілей всього один файл. При цьому для того, щоб створити зображення з різною якістю, користувачам буде надіслано три початкових фрагменти цього файлу, довжини яких відповідно рівні 10 кБ, 20 кБ і 50 кБ.

Можливість вилучення файлів різної якості з вихідного образу в алгоритмі SPIHT обумовлена тим, що на одному з його етапів застосовується вейвлетна трансформація вихідного зображення на базі перетворення Хаара.

Крім того, основна структура даних цього алгоритму, просторово орієнтоване дерево, використовує той факт, що різні піддіапазони відображають різні геометричні особливості образу.

Перетворення Хаара можна застосовувати до зображення кілька разів поспіль. При цьому утворюються різні області (піддіапазони), що складаються з середніх і дрібних деталей.

Таким чином, виконується декомпозиція вихідного зображення на піддіапазони так, що нижні піддіапазони відповідають високим частотам зображення, а верхні піддіапазони відповідають низьким частотам образу. При цьому, у їх межах концентрується основна частина енергії зображення (рис.2.5). У загальному випадку, це відповідає механізму дискретно-вейвлетного (ДВП) перетворення, що застосовується в технології JPEG2000.

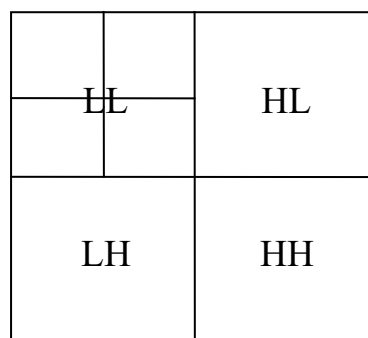


Рисунок 2.5 – Розбиття вихідного зображення на піддіапазони у результаті вейвлет-трансформації

При цьому очевидно, що величини компонент $c_{i,j}$ деталей зображення зменшуються при переміщенні від високого рівня до більш низького. Крім

того, є певна просторова подібність між піддіапазонами. Зокрема, деякі частини зображення займають одну і ту ж просторову позицію в усіх піддіапазонах.

На рис. 2.5 піддіапазон LL формується у результаті фільтрації низьких частот по рядках і стовпцях, HL - високих частот по рядках і низьких за стовпцями; LH - низьких частот по рядках і високим за стовпцями; HH - високих частотах по рядках і стовпцях.

Розглянемо загальний принцип роботи алгоритму SPIHT [21].

Позначимо пікселі вихідного зображення через $P_{i,j}$, де i, j - координати пікселя в оригінальному документі. Для оптимізації переведення зображення в компоненти трансформанти вейвлетного перетворення $C_{i,j}$ може бути використано будь-яка множина фільтрів T .

У свою чергу, множина компонент $C_{i,j}$ утворює трансформований кадр F . Таке перетворення у загальному вигляді може бути подано наступним чином:

$$F = T(p) \quad (2.3)$$

При прогресуючому методі передачі декодер починає з того, що привласнює нульове значення реконструйованому образу F . Потім він приймає (кодовані) перетворені компоненти, декодує їх і використовує для отримання поліпшеного образу F , який, у свою чергу, генерує покращене зображення, як показано наступним виразом:

$$p = T^{-1}(c) \quad (2.4)$$

Основна мета прогресуючого методу полягає в оперативній передачі найважливішої частини інформації про відеокадр. Ця інформація дає найбільше скорочення розбіжності вихідного і реконструйованого образів.

Для кількісного виміру цієї розбіжності метод SPIHT використовує середньоквадратичну помилку MSE.

Найбільші (по абсолютній величині) компоненти $C_{i,j}$ несуть у собі інформацію, яка найбільше скорочує розбіжність MSE, тому прогресуюче

кодування має відправляти ці коефіцієнти у першу чергу. У цьому полягає перший важливий принцип SPIHT.

Інший принцип заснований на закономірності, за якою ймовірність того, що найбільш значущі біти двійкового опису цілих чисел можуть бути одиницями, зростає, коли ці числа наближаються до максимальних значень. (Наприклад, в 16-бітному комп'ютері число +5 має опис 0 | 000.0101, а велике число +65382 запишеться у вигляді 0 | 11111101100110. Це наводить на думку, що старші біти містять найбільш значущу частину інформації, і їх також слід надсилати (або записувати в стислий масив даних) у першу чергу.

Головним недоліком цього алгоритму є великий обсяг інформації координат переданих значущих компонент. Для того, щоб оптимізувати цей процес, в алгоритмі SPIHT вся множина компонент розбивається на підмножини, наприклад, так як показано на рис. 2.6.

T11	T12	T2
T13	T14	
T3		T4

Рисунок 2.6 – Розбиття множини компонент вейвлет-трансформації на підмножини

Далі, отримані піддіапазони аналізуються. У ході цього зчитується кожне з множин компонент і виконується перевірка на наявність в їх складі значущих компонент, тобто компонент, що задовольняють умові:

$$|c_{ij}| \geq 2^{\Phi(g(t)_{small})} \quad (2.5)$$

Якщо у множині компонент піддіапазону знайдено хоча б одну таку компоненту, то дана множина також вважається значущою.

Далі така множина розбивається на підмножини, для яких виконується така ж перевірка компонент, і так поки, що не буде отримано множини, що

складається з однієї компоненти. У результаті декодера передається інформація, яка вказує, чи є конкретна множина значущою, чи ні. Завдяки ієрархічній структурі множин, загальний обсяг інформації про розташування значущих коефіцієнтів стає меншим, ніж проста передача їх координат.

Розглянемо в загальному вигляді черговість виконання алгоритму SPIHT.

2.4.2 Етапність роботи алгоритму SPIHT

На першому кроці роботи алгоритму кодований кадр підлягає вейвлетному трансформуванню з використанням того чи іншого набору вейвлетних фільтрів в одному з базисів перетворення (ДВП, Хаара, Уолша і т.д.). Після цього, отримані таким чином компоненти $c_{i,j}$ подаються у вигляді цілих чисел фіксованою розрядності [20, 21].

Припустимо, що коефіцієнти представлені у вигляді цілих чисел зі знаком, розрядність яких дорівнює 16. У цьому випадку старший біт є знаковим, а в інших двійкових 15 розрядах записаний модуль цього числа. Такий опис числа відрізняється від комплементарного уявлення чисел зі знаком, яке традиційно застосовується в комп'ютерах. Тоді значення компонент будуть змінюватись у діапазоні від $-(2^{15} - 1)$ до $(2^{15} - 1)$. Очевидно, що кількість розрядів $\Phi(g(t)_{\text{small}})$ для опису модуля чисел буде визначатися таким чином: $\Phi(g(t)_{\text{small}}) = \lceil \log_2(2^{15} - 1) \rceil = 14$.

На наступному кроці виконання алгоритму SPIHT виконується сортування отриманих компонент. Тут необхідно передати число I компонент, які задовольняють нерівності:

$$2^{\Phi(g(t)_{\text{small}})} <_{i,j} < 2^{\Phi(g(t)_{\text{small}})+1}. \quad (2.6)$$

Після цього виконується передача I пар координат і I знаків цих компонент.

Далі, після того, як знаки і координати компонент передані, здійснюється передача $(\Phi(g(t)_{\text{small}}) - 1)$ -х найстарших біт усіх компонент, що задовольняють нерівності (2.6). Ці компоненти були обрані на кроці сортування попередньої ітерації циклу (але не поточної ітерації).

Наступний етап роботи алгоритму включає у себе виконання ітерацій обчислення. При цьому, величина $\Phi(g(t)_{\text{small}})$ зменшується на 1.

Надалі, якщо необхідно зробити ще одну ітерацію, алгоритм переходить на крок сортування компонент. Як правило, остання ітерація відбувається при $\Phi(g(t)_{\text{small}}) = 0$, Але кодер може зупинитися раніше.

У цьому випадку найменш важлива частина інформації (найменш значущі біти усіх компонент вейвлетного трансформування) не надсилатимуться. У цьому полягає природне відкидання частини інформації в методі SPIHT. Це еквівалентно скалярному квантуванню, але результат виходить краще, оскільки коефіцієнти передаються в упорядкованій послідовності. В альтернативі кодер передає весь образ (тобто, усі біти усіх вейвлетних коефіцієнтів), а декодер може зупинити процес декодування у будь-який момент, коли відновлюване зображення досягло необхідної якості Y . При цьому, необхідний рівень Y якість або зумовлюється користувачем, або встановлюється декодером автоматично.

2.4.3 Особливості обробки кадрів відеопотоку на базі SPIHT.

Перш за все, як кодер, так і декодер, мають використовувати єдиний тест при перевірці множин на значимість. У той же час, алгоритм кодування використовує три списки, які називаються:

- список значущих пікселів (LSP, list of significant pixels);
- список незначущих пікселів (LIP, list of insignificant pixels);
- список незначущих множин (LIS, list of insignificant sets).

У дані списки заносяться координати (i, j) так, що у списках LIP та LSP вони являють індивідуальні коефіцієнти, а у списку LIS вони подаються як множини $C_{i,j}$.

У свою чергу, список LIP містить координати компонент, які були незначущими на попередній стадії сортування. На поточному етапі вони перевіряються, і якщо множина є значимою, переміщуються у список LSP [21].

Аналогічним чином множини з LIS тестуються у послідовному порядку, і, якщо у ході цього виявляється, що множина стала значущою, відповідно, вона видаляється з LIS і далі розкладається. Нові підмножини, що складаються більш ніж з одного елемента, поміщуються назад до списку LIS, де вони пізніше будуть піддані тестуванню, а одноелементні підмножини

перевіряються і додаються у список LSP або LIP у залежності від результатів тесту.

Стадія поправки передає n -ний найстарший біт записів зі списку LSP.

Проте, хоча у разі його інтеграції в існуючі системи обробки відеоданих алгоритм SPIHT володіє потенційною можливістю забезпечити скорочення інформаційної інтенсивності $g(t)$ відео, йому властиві такі недоліки:

- алгоритм орієнтований на роботу у технологічному базисі, що використовує у якості ортогонального перетворення механізми вейвлетного трансформації. Отже, це ускладнює його застосування у схемі MPEG, що використовує для внутрішньокадрового кодування технологію JPEG, базовану на дискретно-косинусном перетворенні;

- відсутність чітких механізмів контролю якості зображень;

- у ряді випадків алгоритм забезпечує недостатній показник коефіцієнта стиснення.

У той же час, продуктивність алгоритму SPIHT, як і інших методів кодування з втратами, може бути підвищена за допомогою статистичного кодування вихідного потоку даних.

Одночасно з тим, характер розподілу знаків та індивідуальних бітів компонент вейвлетного трансформування, отриманих у ході кожної ітерації, близький до рівномірного.

Звідси випливає, що статистичне кодування потенційно не здатне забезпечити істотного скорочення обсягу даних для опису зображення. У цьому випадку забезпечується незначне збільшення коефіцієнта K стиснення, яке не компенсує часові витрати на його виконання кодером і декодером.

Таким чином, для забезпечення істотного збільшення коефіцієнту K стиснення відеопотоку без ускладнення алгоритму обробки, необхідно застосовувати альтернативні методи, що базуються на усуненні типів надмірності, які ігноруються існуючими підходами.

3 СТИСНЕННЯ БЕЗ ВТРАТ ФРАГМЕНТІВ КАДРУ У СПЕКТРАЛЬНОМУ ПРОСТОРИ З ВИКОРИСТАННЯМ ЕФЕКТИВНИХ КОДОВИХ КОНСТРУКЦІЙ

3.1 Вимоги до методу кодування без втрат у складі технології відеокодування

До будь-якого методу кодування без втрат, що бере участь у обробці відеоданих як складова частина технології кодування, може бути висунуто ряд загальних вимог. Зокрема, таких, як [16-18]:

- забезпечення щонайбільшого коефіцієнту стиснення H_{comp} ;
- продуктивність Z в умовах, коли доступними для використання є досить обмежені обчислювальні ресурси системи, на базі якої він функціонує;
- симетричність механізмів кодування та декодування.

У загальному випадку дані вимоги може бути описано наступною системою співвідношень:

$$\begin{cases} H_{\text{comp}} \rightarrow \max; \\ Z \geq Z_{\text{min}}; \\ \Xi_{\text{cod}} = \Xi_{\text{dec}}, \end{cases} \quad (3.1)$$

де Z - коефіцієнт продуктивності методу, фізично сутність якого полягає у можливості функціонування на рівні, не меншому, ніж деякий мінімальний, який, у свою чергу, забезпечує такі показники кодування, як:

$$\begin{cases} \xi \rightarrow \min; \\ t_{\text{cod}} \leq t_{\text{trh}}, \end{cases} \quad (3.2)$$

де ξ - ймовірність виникнення фатальної помилки у ході кодування;

t_{cod} та t_{trh} - величини часу обробки, та порогового значення часу обробки відповідно; вважається, що при $t_{\text{cod}} > t_{\text{trh}}$ функціонування процесу кодування може бути призупинено, що найбільш чутливим є для відеосервісів реального часу;

Ξ_{cod} та Ξ_{dec} - показники етапності та складу процесів кодування та декодування відповідно.

Стосовно вимоги щодо рівня продуктивності Z методу кодування необхідно зазначити, що залежно від алгоритму, на якому базується метод, ключовим показником, що забезпечує виконання вимоги (3.1) у тій частині, що стосується рівня продуктивності, а саме - $Z \geq Z_{\text{min}}$ може бути [17]:

- доступний обсяг V_{mem} ОЗП для зберігання даних проміжних обчислень;
- доступний обсяг V_{store} ПЗП (для зберігання кодеку, файлів, що необхідні для його функціонування а також програмних констант);
- потужності Ψ_{cpu} процесорної системи, або процесору.

На той випадок, коли метод кодування забезпечує високий коефіцієнт стиснення H_{comp} , але при цьому вимагає значних обчислювальних потужностей; алгоритм може вважатися нераціональним.

Відповідно, його застосовуваність буде обмежена високопродуктивними системами, тобто, на базі систем загального використання реалізація такого алгоритму є неможливою.

Разом з тим, так як процеси кодування та декодування у ряді випадків є синхронними, що функціонують у парі, ресурсоємність кожного з них має істотне значення для функціонування системи у цілому. Першочергово це важливо якраз саме для таких систем та сервісів, що базуються на використанні відеоданих, як:

- системи відеозв'язку;
- системи інтерактивної відеовзаємодії;
- потокове відео реального часу.

У загальному випадку існують такі співвідношення складностей реалізації, та, відповідно, ресурсоємності механізму кодування та відновлення, як:

1. Механізм стиснення є складнішим, ніж механізм декодування, та потребує суттєво більшого обсягу обчислювальних ресурсів ($V_{\text{mem}}, V_{\text{store}}, \Psi_{\text{cpu}}$), ніж алгоритм відновлення. Така ситуація є допустима для випадків зберігання відеоданих у сховищі, проте є потенційно неприйнятною для означених вище відео сервісів та систем.

2. Метод кодування є значно менш ресурсоємним, ніж механізм реконструювання. Це характерно для умов, коли операція компресії здійснюється на базі простої апаратної платформи, для якої рівень доступних ресурсів є критичним (космічний апарат, безпілотний літальний апарат, розподілена мережа датчиків тощо).

3. Рівень ресурсоємності для механізмів кодування та реконструкції відеоданих є однаковим, або приблизно рівним. Для систем, де здійснюється формування, трансляція та перегляд відео контенту у реальному часі такий випадок є найбільш прийнятним.

Поняття співвідношень ресурсоємності кодеру та декодеру є багато у чому спільним поняттям симетричності алгоритму.

У свою чергу, симетричність алгоритму передбачає наявність однакової кількості N_{op} елементарних операцій, що виконуються як у процесі стиснення, так і у процесі відновлення деякої кількості даних.

Це пояснюється тим, що означені процеси містять однакові технологічні етапи.

Відповідно, обчислювальне навантаження на кодер/декодер у складі передавача та приймача буде однаковим, що створює умови для:

- можливості онлайн-перегляду даних відео фіксації у реальному часі;
- можливості застосування кодеку у системах інтерактивного відеозв'язку, оскільки їх функціонування робиться можливим у наслідок синхронізації відеопотоків.

При цьому, є допустимим деяка незначна розбіжність ΔN_{op} у кількості $N_{op}^{(enc)}$ елементарних операцій для кодування та декодування $N_{op}^{(dec)}$, що ілюструється наступним співвідношенням:

$$\Delta N_{op} = \left| N_{op}^{(enc)} - N_{op}^{(dec)} \right| \rightarrow \min, \quad (3.3)$$

або:

$$\Delta N_{op} = \frac{N_{op}^{(enc)}}{N_{op}^{(dec)}} \rightarrow 1 \quad (3.4)$$

Тут вирази (3.3) та (3.4) може бути інтерпретовано як умови, що визначають можливість застосування асиметричного алгоритму кодування для застосування у складі системи відеокодування.

При цьому, відповідність умовам (3.4) є ознакою симетричного алгоритму, тоді як у випадку, коли справедливою є умова (3.3), можемо говорити про те, що алгоритм є квазісиметричним та також може бути задіяний у складі технології відео кодування, що передбачає можливість обробки відеоданих у реальному часі.

Разом з тим, умови (3.3) та (3.4) не є суттєвими для випадків, коли мова йде про сервіси та системи, що не передбачають побудову коду відеоданих у реальному часі [13, 14]. Так, на випадок сервісів VoD (video on demand – відео на запит), кількість $N_{op}^{(enc)}$ елементарних операцій у ході кодування, а, відповідно, час кодування $t^{(enc)}$ не є критичним показником. Зокрема, коли здійснюється звернення до попередньо кодованих відеоданих, важливим є час $t^{(dec)}$ декомпресії. Це справедливо як для відео, так і аудіоданих, зображень та архівів бібліотек. При цьому, допустимим є навіть застосування алгоритмів фрактального стиснення, для яких величина ΔN_{op} , обчислена за виразом (3.4), може сягати 2000 і більше.

Таким чином, метод кодування без втрат має бути:

- симетричним (або квазісиметричним);
- відповідним вимогам (3.2), тобто, його продуктивність має бути не нижчою за таку, що забезпечує його функціонування без виникнення помилок та затримок у роботі;
- забезпечувати щонайбільший з можливих рівень коефіцієнта стиснення.

3.2 Поширені методи кодування без втрат у складі систем стиснення відео

На сьогодні у базисі MPEG на фінальному кроці побудови кодового представлення відеоданих використовуються методи ентропійного та арифметичного кодування. Це, зокрема, кодування за Хафманом та арифметичне стиснення [14].

Кожен з означених підходів передбачає формування особливого алфавіту $A(\ell)_{m,n}$ для побудови кодового опису m, n -го фрагменту $d(\ell)_{m,n}$ ℓ -го кадру відеопотоку, що залежить від особливостей змісту кожного кодованого фрагменту, тобто, справедливим є твердження про те, що:

$$A(\ell)_{m,n} \neq A(\ell)_{m+1,n} \neq \dots \neq A(\ell)_{m+\varepsilon,n+\chi} \quad (3.5)$$

У залежності від особливостей змісту даних довільного типу та походження, що йдуть на обробку, алгоритм Хафмана забезпечує зменшення об'єму вихідних даних від 20 до 90% у порівнянні з початковим об'ємом.

Водночас, статистично доведено, що на випадок стиснення відеоданих як для методу Хафмана, так і для арифметичних кодів, коефіцієнт стиснення лише у окремих випадках перевищує значення $N_{\text{comp}} = 2$.

Даний показник досягається за рахунок зменшення статистичної надмірності кодового представлення фрагментів.

Разом з тим, існування надмірностей інших класів, у сутності, не здійснюється. У підсумку це веде до того, що подальше збільшення показника N_{comp} у базисі існуючих методів кодування є неможливим.

З урахуванням вищезазначеного, доцільно застосовувати методи, що фокусуються на скороченні надмірностей, що більш значним чином впливають на кількість біт, що використовуються для представлення фрагментів відеоданих.

У зв'язку з цим, для обробки відеоданих на етапі кодування без втрат базової схеми може бути застосовано методи на базі формування ефективних кодових конструкцій змінної ваги.

3.3 Особливості та переваги застосування ефективних кодових конструкцій змінної ваги при формуванні кодограм фрагментів кадру у спектральному просторі

Головною перевагою кодових конструкцій змінної ваги відносно стандартизованих методів кодування без втрат є забезпечення у середньому на 30% більшого коефіцієнту стиснення [22].

Окрім того, існує ряд конструктивних особливостей, що роблять його більш прийнятним для застосування у складі технологій відео кодування.

Так, базовий алгоритм стиснення відео для усунення кодової надлишковості опису даних, що має в основі ентропійне кодування (зазвичай це кодування за Хафманом), або арифметичне кодування, є останнім у ланцюжку перетворень вихідного (m,n) -го фрагменту $d^{(t)}_{m,n}$, що належить i -му кадру за алгоритмом JPEG (рис. 3.1).

У рамках цього, на обробку надходять множини фрагментів $d^{(t)}_{m,n}$ у спектральному форматі представлення, що являють собою матриці 8×8 , утворені компонентами $c_{i,j}$, що попередньо були квантовані та підлягали операції округлення (рис.2.1) [14, 17, 23].

Для того, щоб кодер міг здійснити обробку надходячих даних, забезпечуючи при цьому максимізацію коефіцієнта стиснення на поточному K_{enc} етапі, необхідно певним чином форматувати дані, тобто, змінити вихідний формат представлення $d^{(t)}_{m,n}$ (рис. 3.1).

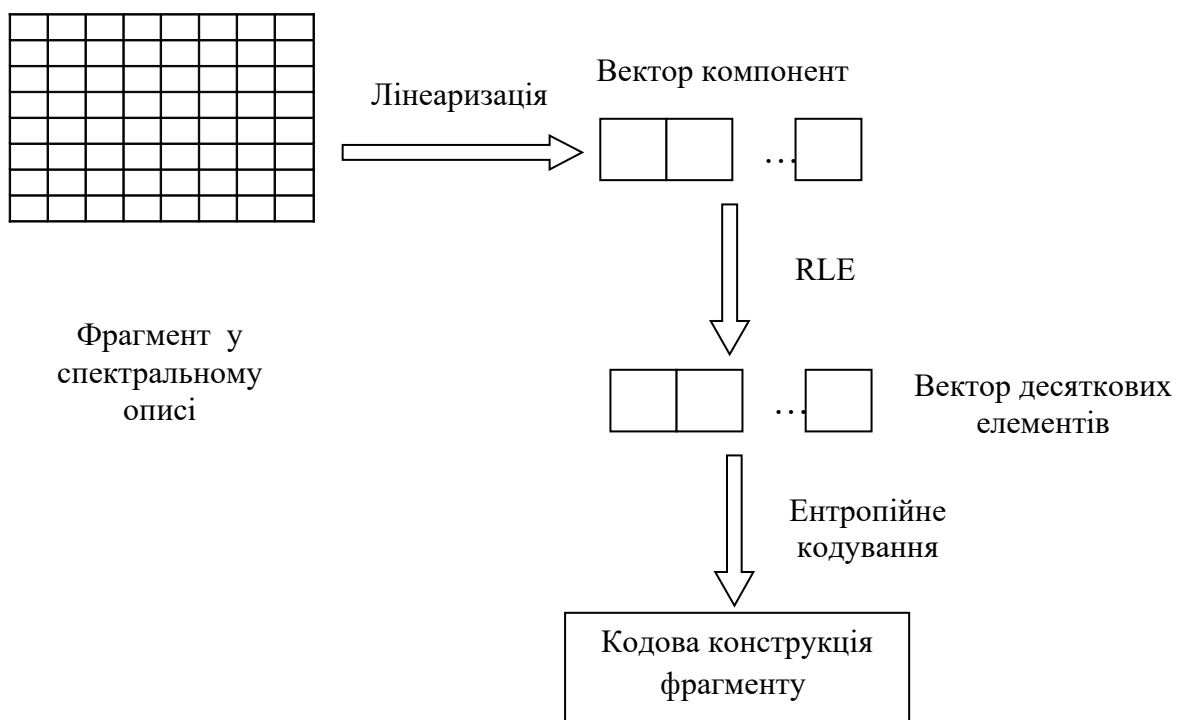


Рисунок 3.1 – Схема технологічних етапів кодування без втрат у схемі JPEG

Для цього фрагмент $d^{(t)}_{m,n}$ попередньо лінеаризується, тобто, підлягає переведенню з двовимірної форми до вигляду вектору у ході процедури зиг-заг сканування значень $c_{i,j}$ компонент, починаючи з DC-компоненти.

Далі утворений вектор підлягає операції кодування RLE (run-length encoding), що змінює кількість елементів у десятковому форматі, які надходять до кодеру. Здійснюється це у результаті зміни кількості нульових елементів областей ВЧ та СЧ компонент $c_{i,j}$ парюю значень «символ; кількість повторів».

У підсумку, замість 64 початкових компонент $c_{i,j}$ утворюється n елементів, при цьому виконується умова:

$$n \leq 64. \quad (3.6)$$

У свою чергу, на наступному кроці сформовані дані підлягають статистичному кодуванню.

На відміну від зазначеного сценарію, у випадку застосування блочного кодування змінної ваги, обробка виконується безпосередньо відносно фрагменту $d(\ell)_{m,n}$ у спектральному просторі, компоненти якого попередньо квантовані та після цього округлювалися.

Отже, при цьому відсутні такі технологічні етапи, як:

- лінеаризація на базі зиг-заг сканування компонент $c_{i,j}$;
- RLE-кодування.

Таким чином, застосування кодових конструкцій змінної ваги при кодуванні фрагментів $d(\ell)_{m,n}$ відеокадрів створює умови для зменшення кількості математичних розрахунків, а значить – сприяє скороченню обчислювальної складності загального методу [22].

Окрім зазначеного, ще одна відмінність між кодовими конструкціями змінної ваги та статистичним кодуванням полягає у тому, що у випадку статистичних кодів здійснюється зміна алфавіту опису кодованого повідомлення, для чого попередньо виконується розрахунок його ймовірнісно-статистичних характеристик.

У той же час, на випадок кодів зі змінною вагою зазначена процедура не потребується, що також позначається на швидкості побудови коду у бік її зростання.

3.4 Стиснення без втрат фрагментів кадру у спектральному просторі з використанням кодових конструкцій змінної ваги

Код змінної ваги може розглядатися як повноцінна альтернатива статистичним підходам, традиційним для JPEG, що відносно них має ряд переваг, зазначених раніше. Окрім цього, оскільки блочне кодування змінної ваги може виконуватися відносно безпосередньо масиву значень $d(i)_{m,n}$ фрагменту у початковому форматі, додаткових операцій для інтеграції даного методу кодування у схему JPEG не потребується [22].

У базисі кодового опису зі змінною вагою фрагмент $d(\ell)_{m,n}$ у спектральному просторі розглядається як сукупність з I або J блочних чисел P_j (залежно від напрямку обходу фрагменту – за стовпцями, чи за рядками), тобто, $d(\ell)_{m,n} = \{P_1; P_2; \dots; P_j; \dots; P_J\}$.

При цьому, побудова кожного кодового числа P_j виконується на базі кожного j -го стовпцю фрагменту $d(\ell)_{m,n}$ з використанням значень десяткових компонент $c_{i,j}$ як складових числа P_j .

Оскільки для обробки використовуються стандартизовані фрагменти $d(\ell)_{m,n}$ у спектральному описі, які прийняті у JPEG, відповідно, кількість стовпців буде рівною 8.

Відтак, наближено кодований фрагмент $E(d(\ell)_{m,n})$ може бути подано наступним виразом:

$$E(d(\ell)_{m,n}) = \overset{\vee}{\prod}_{j=1}^8 P_j \quad (3.7)$$

У свою чергу, для обчислення кодового числа P_j для j -го стовпцю задіюються вагові коефіцієнти $\Theta_w^{(j)}$ та складники $c_{i,j}$ цього числа. При цьому, виявлення складників кодових чисел здійснюється у ході порядкового сканування.

Слід зазначити, що розрахунок значень кодових чисел P_j може бути виконано також на базі рядків фрагменту $d(\ell)_{m,n}$.

На цей випадок існує характерна закономірність кодових конструкцій змінної ваги, яка демонструється наступним співвідношенням:

$$\begin{cases} P_i \neq P_j \\ (\Phi(d(\ell)_{m,n})|P_i) = (\Phi(d(\ell)_{m,n})|P_j) \end{cases} \quad (3.8)$$

де $\Phi(d(\ell)_{m,n})$ - кількість біт, що витрачається на опис кодованого на базі конструкцій змінної ваги фрагменту $d(\ell)_{m,n}$, початково поданого у спектральному просторі;

P_i та P_j - кодові числа, утворені на базі i -го рядка та j -го стовпця відповідно, при цьому вважається, що $i = j$.

Властивість (3.8) свідчить про те, що кодовий опис $E(d(\ell)_{m,n})$ фрагменту $d(\ell)_{m,n}$, утворений на базі кодових чисел, сформованих по рядках, потребує такої ж кількості біт $\Phi(d(\ell)_{m,n})$, як і на випадок побудови кодового опису по стовпцям. При цьому, величина $\Phi(d(\ell)_{m,n})$ визначається як:

$$\Phi(d(\ell)_{m,n}) = \log_2(d(\ell)_{m,n}) \quad (3.9)$$

У свою чергу, розрахунок кодового значення числа P_j виконується на базі наступної формули [22]:

$$P_j = \sum_{j=1}^8 [c_{i,j} \times \Theta_w^{(j)}], \quad (3.10)$$

де $\Theta_w^{(j)}$ - ваговий коефіцієнт для j -го стовпця спектрального фрагменту $d(\ell)_{m,n}$ рядку, що визначається на основі системи $\{B\}$ базисних елементів $b_{i,j}$, що у загальному вигляді задається виразом:

$$\Theta_w^{(j)} = \psi\{b_{i,j}\} = \varphi\{B\}, \quad j = \overline{1;8}. \quad (3.11)$$

У свою чергу, тут побудова система базисних елементів кодових чисел здійснюється на рівні кожного з рядків фрагменту $d(\ell)_{m,n}$ у спектральному просторі опису, при цьому базисний елемент визначається як найбільше значення компоненти $c_{i,j}$ у рядку, тобто:

$$b_i = \max(c_{i,j}) + 1, \quad (3.12)$$

тут збільшення величини максимальної компоненти $c_{i,j}$ у рядку на одиницю виключає ймовірність виникнення ситуації невизначеності у ході формування кодового опису на той випадок, коли рядок буде містити виключно нульові елементи.

Ваговий коефіцієнт $\Theta_w^{(j)}$, у свою чергу, обчислюється з використанням системи $\{B\}$ базисів. Тоді, з урахуванням виразів (3.11) та (3.12), для розрахунку вагового коефіцієнту $\Theta_w^{(j)}$ буде задіяно наступний вираз:

$$\Theta_w^{(j)} = \prod_{k=j+1}^8 b_{i,k}. \quad (3.13)$$

З виразу (3.13) бачимо, що ваговий коефіцієнт $\Theta_w^{(j)}$ елементу $c_{i,j}$, що знаходиться на m -й позиції у стовпці, буде розраховуватися як добуток базисних елементів блочних чисел, починаючи з $(j+1)$ -го.

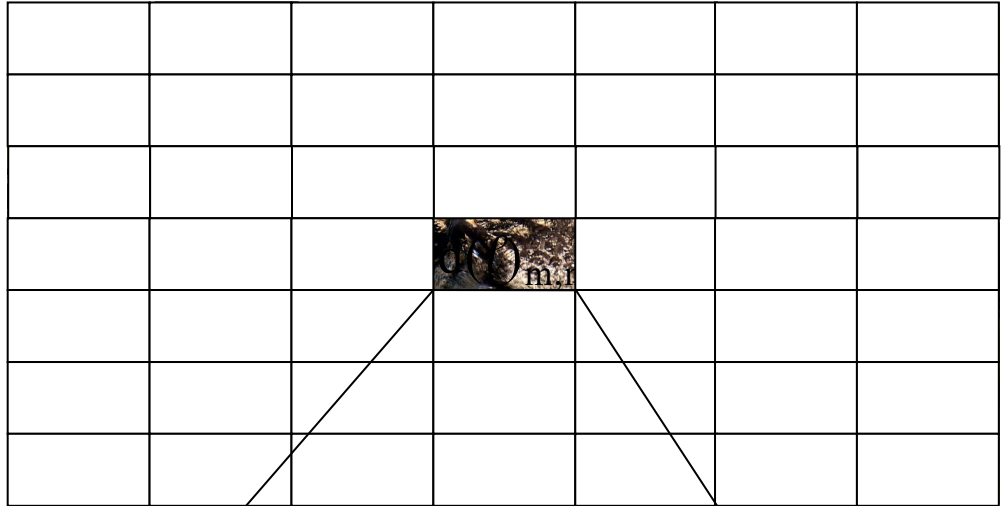
Далі на базі виразу (3.14) підсумкове кодове значення $d(\ell)_{m,n}^{(enc)}$ фрагменту $d(\ell)_{m,n}$, у спектральному просторі буде являти собою конкатенацію обчислених кодових значень чисел P_j , а саме:

$$d(\ell)_{m,n}^{(enc)} = P_1 \& P_2 \& \dots \& P_j \& \dots \& P_J. \quad (3.14)$$

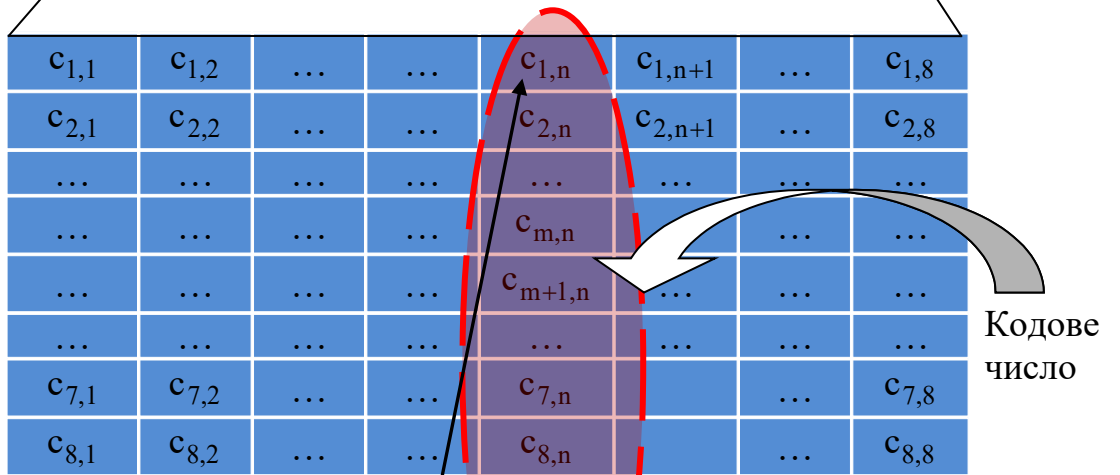
Оскільки при цьому відомо, що за замовчуванням $I=J=8$, вираз (3.14) може бути переписано у вигляді:

$$\hat{d}(t)_{m,n}^{(enc)} = \prod_{j=1}^8 P_j \tag{3.15}$$

Вихідний кадр у спектральному описі



Фрагмент



Елементи кодового числа

Кодове число

Рисунок 3.2 – Локалізація блочних чисел та їхніх складників у межах фрагментів кадру у спектральному форматі опису

Врешті решт, підсумковий об'єм біт $\Phi(d(\ell)_{m,n}^{(enc)})$, що буде витрачено на кодове представлення фрагменту $d(\ell)_{m,n}$, може бути розрахований на базі наступного співвідношення:

$$\Phi(d(\ell)_{m,n}^{(enc)}) = \log_2 \left[\sum_{i=1}^8 \sum_{j=1}^8 [c_{i,j} \times \Theta_w^{(j)}] \right] \quad (3.16)$$

У підсумку загальний процес побудови кодового опису $d(\ell)_{m,n}^{(enc)}$ фрагменту $d(\ell)_{m,n}$ відеокадру у спектральному описі на основі кодів змінної ваги міститиме у собі такі технологічні етапи, як:

1. Розрахунок базисів кодових чисел для кожного рядку фрагменту $d(\ell)_{m,n}$ керуючись виразом (3.12).
2. Обчислення значень вагових коефіцієнтів $\Theta_w^{(j)}$ кодових чисел P_j за формулою (3.13).
3. Знаходження значення кодового числа P_j за співвідношенням (3.10).
4. Формування кодового опису $d(\ell)_{m,n}^{(enc)}$ фрагменту $d(\ell)_{m,n}$ з використанням попередньо розрахованих значень кодових чисел відповідно до виразу (3.14).

3.5 Реконструкція даних, кодованих у базисі кодового опису зі змінною вагою на боці прийому

Послідовність біт, що описує кодове подання фрагменту $d(\ell)_{m,n}^{(enc)}$ відеокадру у спектральному просторі, являє собою послідовність з 8 десяткових блочних чисел, як показано виразами (3.14) та (3.15), як ілюструє рис. 3.3.

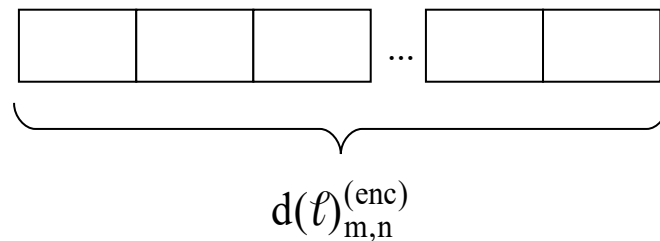


Рисунок 3.3 – Структура кодового представлення фрагменту $d(t)_{m,n}^{(enc)}$ відеокадру на базі сукупності кодових чисел

Для того, щоб мати змогу однозначно реконструювати отриманий таким чином масив даних до вигляду початкового спектрального фрагменту $d(t)_{m,n}$, звідки шляхом зворотнього ДКП – вихідний фрагмент $d(t)_{m,n}$, необхідно мати [22]:

- принцип відновлення фрагменту $d(t)_{m,n}$ з кодограми $d(t)_{m,n}^{(enc)}$;
- службові дані, на базі яких здійснюється відновлення за наявним принципом реконструкції.

У нашому випадку такими службовими даними є система $\{B\}$ базисів, попередньо розрахованих для формування кодового опису фрагменту $d(t)_{m,n}$.

Відповідно, у випадку стандартизованого фрагменту $d(t)_{m,n}$ розміром 8×8 , для його кодового опису потребується 8 базисів b_i .

При цьому, за умов, що для представлення кожного базису b_i попередньо виділяється фіксована кількість β біт, додаткових службових даних, як то – допоміжних маркерів та маркерних сегментів не потребується.

Отже, система $\{B\}$ базисів є єдиними службовими даними, необхідними для декодування стисненої інформації.

Таким чином, у структурі кодового представлення $d^*(t)_{m,n}^{(enc)}$ фрагменту $d(t)_{m,n}$, який надсилається у мережу та далі буде відновлюватися на прийомному боці, міститься 2 складові, а саме:

- безпосередньо кодограма $d(t)_{m,n}^{(enc)}$ фрагменту $d(t)_{m,n}$;
- службові дані у вигляді поля послідовності значень b_i обчислених базисів (рис. 3.4).

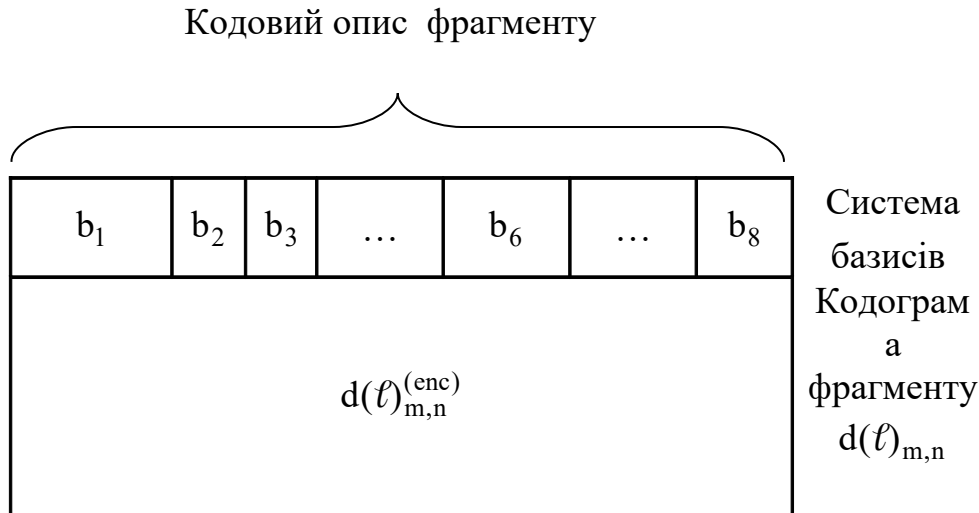


Рисунок 3.4 – Структура кодового опису фрагменту $d(t)_{m,n}$

Окрім того, що зазначений підхід до формування службових даних забезпечує однозначне декодування стисненої інформації та не веде до ускладнення структури кодограми, забезпечується також мінімізація розширення розміру $d^*(t)_{m,n}^{(enc)}$ внесенням до її складу системи базисів.

Для того, щоб принцип формування коду, проілюстрований рис.3.4, був гарантовано працездатним, необхідно дотримуватися умови, за якої кількість біт $\Phi(P_j)$, необхідних для представлення кодового числа P_j , має не перевищувати розмір W_{code} кодового слова (комп'ютерного слова), тобто:

$$\Phi(P_j) \leq W_{code}. \quad (3.17)$$

Разом з тим, за однією з властивостей кодових структур зі змінною вагою, розмір кодового слова гарантовано не переповниться, якщо встановити його на рівні не нижчому, ніж значення добутку величин базисів, тобто:

$$W_{\text{code}} \leq \prod_{k=j+1}^8 b_{i,k} \quad (3.18)$$

Таким чином, вимога (3.17) виконується.

Тоді процедура реконструкції даних на рівні окремих компонент $c_{i,j}$, беручи до уваги, що попередньо було встановлено розмір кодового слова W_{code} , може бути описана на базі наступного виразу:

$$c_{i,j} = \left[\frac{P_j}{\Theta_w^{(j)}} \right] - \left[\frac{P_j}{b_i \times \Theta_w^{(j)}} \right] \times b_i, \quad (3.19)$$

У формулі (3.19) квадратні дужки являють собою цілочисельну частину відношення величин.

Тобто, реконструкція вихідного $d(\ell)_{m,n}$ фрагменту кадру у спектральному представленні здійснюється покомпонентно. Далі отримані фрагменти $d(\ell)_{m,n}$ підлягають зворотньому ДКП .

4. ДОСЛІДЖЕННЯ СТЕПЕНЮ СКЛАДНОСТІ АЛГОРИТМУ ОБРОБКИ ФРАГМЕНІВ ВІДЕОКАДРІВ НА БАЗІ БЛОЧНИХ КОНСТРУКЦІЙ ЗІ ЗМІННОЮ ВАГОЮ

4.1 Підхід до дослідження складності реалізації алгоритму

Серед головних характеристик методу кодування, зазначених у параграфі 4, зазначено його продуктивність Z .

Незалежно від оточення, у межах якого функціонує метод, одним з ключових показників його продуктивності – як існуючої, так і потенційної, є складність реалізації його алгоритму.

У свою чергу, один з підходів щодо об'єктивного оцінювання ступеню продуктивності алгоритму полягає у дослідженні раціональності його реалізації.

При цьому, раціональність реалізації може бути інтерпретовано як кількість елементарних обчислювальних операцій N_{op} , що необхідно виконати для обробки мінімального обсягу даних.

Відповідно, керуючись показником N_{op} , як метрикою продуктивності алгоритму, необхідно виконати розрахунок фактичного необхідного обсягу елементарних операцій. Після цього необхідно виконати порівняння обчисленої величини з відповідними показниками продуктивності $N_{сри}$ існуючих обчислювальних систем.

На випадок, коли справедливим є наступне:

$$N_{op} \leq N_{сри}, \quad (4.1)$$

можемо зробити висновок про те, що:

- алгоритм є раціональним та придатним для застосування взагалі;
- обчислювальна складність алгоритму дозволяє його застосовувати на базі пристроїв загального застосування, що не потребують збільшених обчислювальних потужностей [24].

В інакшому випадку - алгоритм, що складає основу методу, потребує раціоналізації та не може застосовуватися у теперішньому вигляді.

У свою чергу, величини N_{cpu} для ряду існуючих обчислювальних систем наведено табл. 4.1 [24].

Таблиця 4.1 – Значення N_{cpu} показників продуктивності ряду існуючих обчислювальних систем

Процесор	Тактова частота, МГц	Інструкцій за такт	Пікова продуктивність, гігафлоп (млрд. опер./сек.)
AMD Athlon 64 X2 4200	2200	3	13,3
AMD Phenom II X6 1100T Black Edition (HDE00ZF)	4000	4	64
Intel Pentium 4	3000	2	6
Intel Core 2 Duo	3000	4	24
Intel Core 2 Quad Q8300	2500	4	40
Intel Core 2 Quad	3500	4	56

4.2 Розрахунок обчислювальної складності методу кодуваннями зі змінною вагою

Загальний процес формування кодової структури зі змінною вагою для фрагменту $d(\ell)_{m,n}$ умовно може бути поділено на ряд етапів (рис.4.1), а саме:

- побудова системи $\{B\}$ базисів кодових чисел P_j ;
- знаходження вагових коефіцієнтів $\Theta_w^{(j)}$;
- обчислення значень кодових чисел;
- побудова кодограм фрагментів $d(\ell)_{m,n}$.

Отже, у ході першого кроку обробки, на етапі формування системи $\{B\}$ базисів, виконується пошук найбільшого значення компоненти $c_{i,j}$.

Після того, як означені дії виконано, використовуючи дані, отримані у ході поточного кроку обробки та керуючись виразом (3.12), обчислюється значення величини b_i .

Таким чином, у процесі розрахунку одного базису b_i кодового числа використовуються такі операції, як:

- логічне “АБО”;
- складання за модулем 2 (“XOR”).

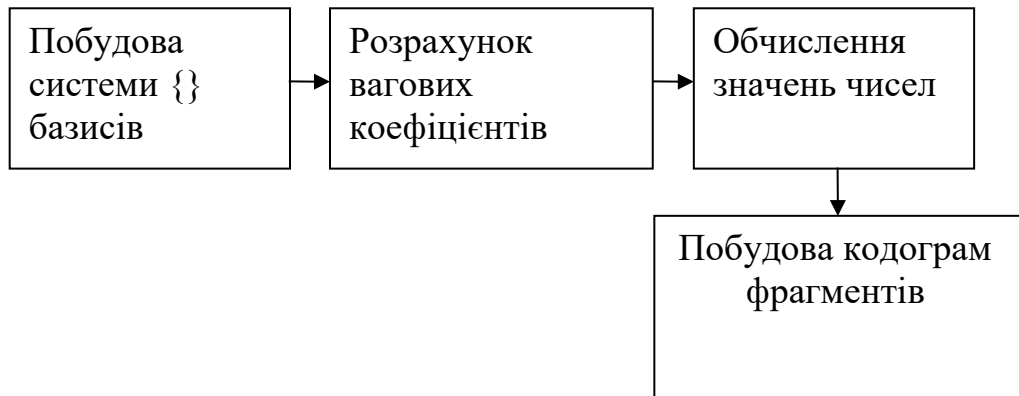


Рисунок 4.1 – Умовний поділ процесу формування кодової структури зі змінною вагою для фрагменту $d(\ell)_{m,n}$ на технологічні етапи

Так, на базі логічної операції “ АБО” передбачається знаходження максимальної величини компоненти $c_{i,j}$ у кожному i -му рядку фрагменту $d(\ell)_{m,n}$. Тобто, така операція виконується для усіх 8 рядків фрагменту $d(\ell)_{m,n}$, де вона застосовується відносно усіх 8 елементів рядку.

Таким чином, загальна кількість $N_{op}^{(max)}$ операцій визначення найбільшого елемента $c_{i,j}$ рядка буде дорівнювати $N_{op}^{(max)} = 8 \times 8 = 64$.

Далі, у ході операції збільшення значення знайденої максимальної $\max(c_{i,j})$ компоненти у рядку необхідно задіяти логічну операцію XOR. Отже, при 8 рядках у структурі $d(\ell)_{m,n}$, необхідно виконати операцію XOR 8 разів. Тобто, обсяг операцій $N_{op}^{(inc)}$ збільшення на 1 долівнює 8. Тоді сумарний обсяг операцій $N_{op}^{(b_i)}$ етапу побудови системи базисів розраховується за виразом:

$$N_{op}^{(b_i)} = N_{op}^{(max)} + N_{op}^{(inc)}, \quad (4.2)$$

На базі виразу (4.2) отримуємо $N_{op}^{(b_i)} = 64 + 8 = 72$.

Далі, у ході наступного етапу обробки, у відповідності з виразом (4.13), виконується розрахунок вагових коефіцієнтів $\Theta_w^{(j)}$. При цьому, кожна з величин обчислюються у вигляді добутку певної множини базисів b_i кодового числа. Отже, буде використовуватися логічна операція “mul” (множення).

Тут необхідно брати до уваги те, що обчислення вагового коефіцієнту $\Theta_w^{(j)}$ потребує застосування базисів b_i , починаючи з наступного рядку, тобто, з рядку, що має індекс $(i+1)$.

Тобто, під час розрахунку кожного наступного вагового коефіцієнту $\Theta_w^{(j)}$ обсяг базисів b_i , які задіюються для цього, буде зменшуватися на 1.

На базі зазначеного виходить, що загальний обсяг операцій “mul”, що буде використано у процесі розрахунку значення $\Theta_w^{(j)}$, для одного кодового числа обчислюється за виразом:

$$N_{op}^{(\Theta_w^{(j)})} = \sum_{\beta=i-1}^1 N_{\beta}^{(mul)}, \quad (4.3)$$

де $N_{\beta}^{(mul)}$ - обсяг операцій множення, що виконуються у ході обчислення одного вагового коефіцієнту $\Theta_w^{(j)}$.

У підсумку, на базі виразу (4.3) за умови, що кількості рядків $i=8$, сумарна кількість операцій множення буде дорівнювати

$$N_{op}^{(\Theta_w^{(j)})} = 7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$$

У ході наступного етапу кодування, під час розрахунку значень кодових чисел P_j , яке виконується відповідно до виразу (3.10), застосовується операція обчислення суми відносно i -складників кодового числа. Такими складниками є величини компонент $c_{i,j}$ фрагменту $d(t)_{m,n}$. Отже, тут буде застосовано логічну операцію “ADD”. Оскільки при цьому необхідно отримати суму з i складників, відповідно, дану операцію буде використано $(I-1)$ разів.

Тоді беручи до уваги те, що логічну операцію “ADD” необхідно задіяти відносно j стовпців, як ілюструє рис.3.2, відповідно, співвідношення, на базі якого розраховується загальний обсяг операцій “ADD”, тобто, $N(\text{sum})$, буде наступним:

$$N_{\text{op}}^{(\text{sum})} = (I - 1) \Gamma J. \quad (4.4)$$

Тут при тому, що $I = J = 8$, сумарний обсяг операцій “ADD” дорівнюватиме 56.

У свою чергу, у ході множення значень компонент $c_{i,j}$, що розглядаються як елементи кодових чисел P_j , та вагових коефіцієнтів $\Theta_w^{(j)}$, також використовується певна кількість операцій, також використовується певна кількість операцій, також використовується певна кількість операцій “mul”. Якщо взяти до уваги, що у рядку міститься кількість компонент $c_{i,j}$ рівна I , результуючий обсяг $N_{\text{op}}^{(\text{mul})}$ операцій множення “mul” розраховуватиметься згідно виразу:

$$N_{\text{op}}^{(\Theta_w^{(j)})} = N(\Theta_w^{(j)}) \Gamma I \Gamma J, \quad (4.5)$$

де I та J – загальна кількість рядків та стовпців у складі фрагменту $d(\ell)_{m,n}$ відповідно, добуток значень яких, у свою чергу, еквівалентний кількості елементів $c_{i,j}$;

$N_{\text{op}}^{(\Theta_w^{(j)})}$ - попередньо розрахована кількість операцій для обчислення вагових коефіцієнтів $\Theta_w^{(j)}$.

Керуючись виразом (4.5), обчислюємо результуючу величину $N_{\text{op}}^{(\text{mul})}$:

$$N_{\text{op}}^{(\text{mul})} = 8 \cdot 8 \cdot 8 = 512.$$

Отже, тоді кількість $N_{\text{op}}^{(\text{bn})}$ елементарних операцій, що виконуються у ході формування кодового значення кодового числа P_j , розраховується за наступним виразом:

$$N_{\text{op}}^{(\text{bn})} = N(\text{mul}) + N(\text{sum}). \quad (4.6)$$

Таким чином, з виразу (4.6) отримуємо $N_{\text{op}}^{(\text{bn})} = 512 + 56 = 568$.

Далі розрахуємо підсумковий обсяг елементарних операцій, що застосовуються у ході кодування фрагменту $d(\ell)_{m,n}$ у спектральному описі.

Так, у підсумку маємо:

- у ході визначення системи $\{B\}$ базисів кодового числа 64 рази виконується логічна операція “АБО”, та 8 разів – логічна операція XOR (

$$N_{\text{op}}^{(\text{max})} = 64, N_{\text{op}}^{(\text{inc})} = 8);$$

- розрахунок вагових коефіцієнтів $\Theta_w^{(j)}$ потребує виконання 28 операцій множення $N_{\text{op}}^{(\Theta_w^{(j)})} = 28$;

- сумарна кількість операцій, необхідних для формування коду кодового числа дорівнює 568 ($N_{\text{op}}^{(\text{bn})} = 568$).

Таким чином, загальний обсяг N_{op} елементарних операцій, що виконуються у процесі побудови кодового представлення фрагменту $d(\ell)_{m,n}$ являє собою суму кількості операцій, виконуваних у ході кожного з технологічних етапів блочного кодування зі змінною вагою, що еквівалентно виразу:

$$N_{op} = N_{op}^{(max)} + N_{op}^{(inc)} + N_{op}^{(\Theta_w^{(j)})} + N_{op}^{(bn)}, \quad (4.7)$$

звідки N_{op} розраховується як $N_{op} = 64 + 8 + 28 + 568 = 668$.

Маючи сформований підхід до розрахунку обчислювальної складності алгоритму, що знаходиться в основі методу кодування зі змінними вагами, далі можемо розрахувати кількість необхідних операцій для компресії того чи іншого обсягу відеоданих, що надходять на обробку.

Уявімо, що обробці підлягає відеокадр, що має просторову роздільну здатність 1920×1080 пікселів, тобто, кадр є формату FullHD.

Виконаємо розрахунок загальної кількості операцій, що мають бути виконані у процесі формування ефективного коду зі змінною вагою для такого кадру.

Так, для кадру 1920×1080 пікселів при тому, що фрагмент має стандартизований розміру (8×8 пікселів) загальна кількість $N(d(\ell)_{m,n})$ фрагментів $d(\ell)_{m,n}$ буде розраховуватися як:

$$N(d(\ell)_{m,n}) = \frac{H}{8} \times \frac{W}{8}, \quad (4.8)$$

де H та W – висота та ширина кадру у пікселях відповідно.

$$\text{З виразу (4.8) отримуємо } N(d(\ell)_{m,n}) = \frac{1080}{8} \times \frac{1920}{8} = 240 \times 135 = 32400.$$

При цьому, кодування кожного з 32400 фрагментів потребує виконання 668 операцій, як попередньо було розраховано за виразом (4.7). Отже, для кадру

FullHD сумарна кількість $N_{op}^{(\Sigma)}$ операцій буде рівною

$$N_{op}^{(\Sigma)} = N_{op}^{(\Sigma)} \times N(d(\ell)_{m,n}) = 32400 \times 668 = 21643200.$$

Очевидно, що при такому обсязі необхідних для виконання операцій,

кодування у реальному часі створює досить високе обчислювальне навантаження на випадок, коли усі елементарні операції виконуються незалежно. Наближено можна сказати, що при цьому функціонування процесу кодування у реальному часі не гарантується.

У той же час, алгоритм, на якому базується метод кодування, може бути оптимізовано шляхом корегування порядку виконання операцій.

4.3 Визначення підходу щодо оптимізації алгоритму кодування на базі кодових структур змінної ваги

У загальному випадку будь-який алгоритм містить у собі ту чи іншу кількість елементарних дій одного типу, наприклад, “XOR”, “ADD” та ін. Також це стосується ланцюжків комбінацій операцій, що потребують багаторазового виклику у ході роботи алгоритму. У рамках побудови раціональної реалізації алгоритму, доцільним є підхід, що передбачає об’єднання однотипних дій, або ланцюжків команд у рамках т.з. інструкцій, або елементарних однотипних операцій [24].

Для того, щоб оцінити можливість поєднання деякої кількості операцій інструкціями, необхідно попередньо визначити:

- усі множини елементарних дій, що виконуються на єдиному етапі загального технологічного циклу;
- поєднання елементарних дій у ланцюжки, що зазнають багаторазового виклику у межах єдиних етапів загального технологічного процесу.

У нашому випадку обсяг необхідних операцій для побудови кодового опису попередньо було визначено. Відповідно, на базі отриманих відомостей може бути обчислено необхідну кількість інструкцій, що будуть поєднувати у собі елементарні операції на кожному з етапів кодування.

4.3.1 Розрахунок необхідної кількості інструкцій для процесу побудови базисів кодових чисел

У ході зазначеного технологічного етапу, процес визначення найбільшого елементу у рядку на базі виразу (3.12) забезпечується виконанням множини логічних операцій “АБО”. Таким чином, їх може бути поєднано однією інструкцією.

У свою чергу, для збільшення на 1 величини визначеного максимального елементу $c_{i,j}(\max)$ рядку виконується операції “XOR”. Відповідно, множина зазначених операцій також може поєднуватися однією інструкцією. Тобто, загальний обсяг інструкцій, необхідних на етапі формування системи базисів – 2.

4.3.2 Розрахунок необхідної кількості інструкцій у ході визначення вагових коефіцієнтів та при розрахунку кодового числа

На наступному кроці роботи алгоритму, а саме – при обчислення вагових коефіцієнтів $\Theta_w^{(j)}$ за виразом (4.13), виконуються виключно операції множення, що дає змогу поєднати їх також однією інструкцією. Тобто, для розрахунку 8 значень $\Theta_w^{(j)}$ (для одного фрагменту $d(\ell)_{m,n}$) задіюватиметься 1 інструкція.

Також, однією інструкцією поєднуються операції множення у ході обчислення значення кодового числа P_j . Окрім цього, на цьому ж технологічному етапу може бути поєднано усю множину операцій додавання (ADD) також однією інструкцією.

4.4 Висновки щодо підсумкової складності алгоритму

У підсумку, загальний обсяг інструкцій, необхідних для забезпечення функціонування процесу побудови кодових конструкцій змінної ваги, буде таким:

- етап знаходження системи $\{B\}$ базисів b_i - 2 інструкції;

- етап розрахунку вагових коефіцієнтів $\Theta_w^{(j)}$ - 1 інструкція;
- обчислення значення кодового числа P_j - 2 інструкції.

Таким чином, сумарна кількість інструкцій, що є необхідними для реалізації алгоритму – 5.

Разом уся множина дій, що виконуються у ході роботи алгоритму, не є одночасовими. Умовно їх можна віднести до 3 часових періодів, як свідчить діаграма на рис. 4.2.

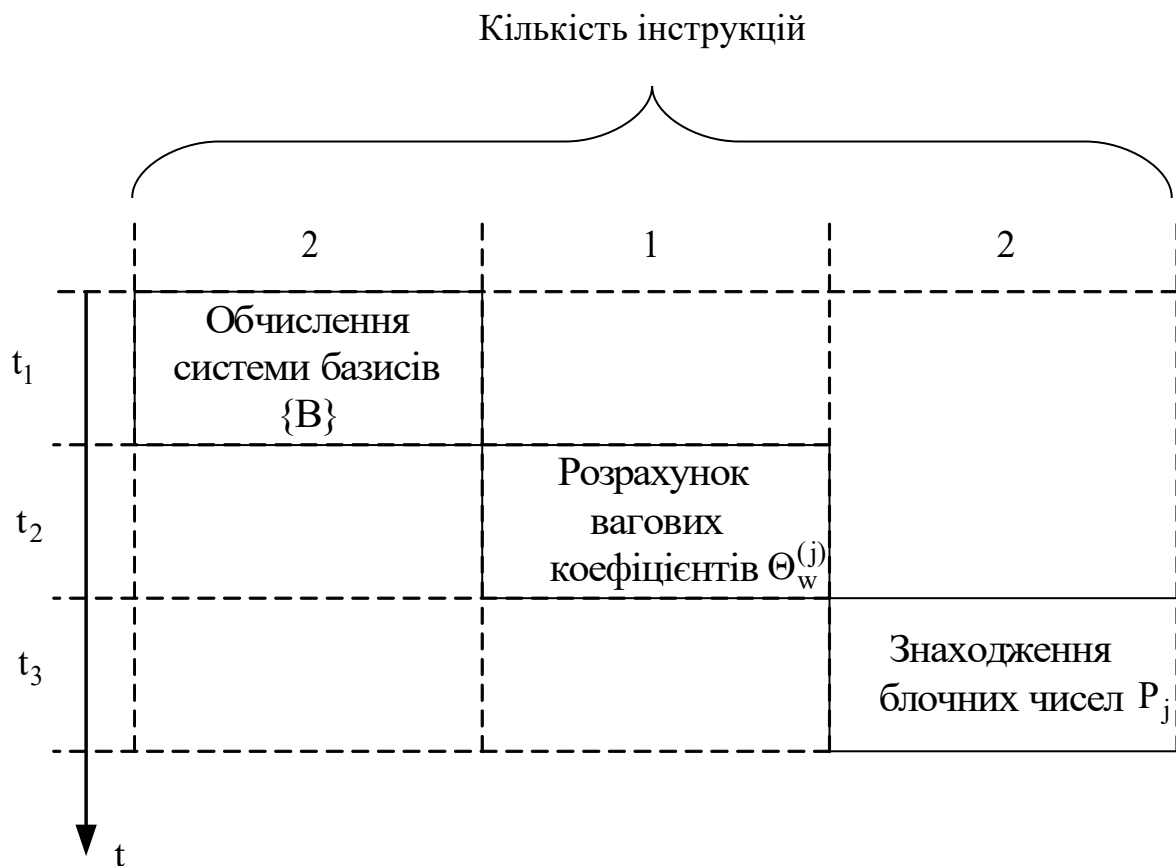


Рисунок 4.2 – Зміна необхідної кількості інструкцій у ході роботи алгоритму з часом

Отже, як свідчить аналіз рис. 5.2, найбільша кількість інструкцій, що виконуються одночасно, не більше 2. У той же час, співставлення отриманих даних з відомостями про обчислювальну потужність сучасних апаратних платформ свідчить про те, що розроблений алгоритм взагалі не створює

помітного навантаження на систему, тобто, є раціональним

Таким чином, згідно табл. 4.1, алгоритм є раціональним.

ВИСНОВКИ

Дана атестаційна робота містить у собі огляд існуючого стану розвитку інформаціо-комунікаційних систем з позицій співвідношення рівня попиту на мережеві сервіси та додатки та існуючих можливостей щодо їх забезпечення.

На базі здійсненого розгляду сучасних особливостей розвитку систем та технологічного базису з галузей телекомунікацій та інформаційних технологій показано, що:

- фактичні можливості мережевої інфраструктури не забезпечують гарантованого передавання усього наявного обсягу мережевої інформації з дотриманням встановленого рівня оперативності та цілісності даних;
- найбільше навантаження на мережу здійснює трафік мультимедіа, а саме – відеодані, відсоток яких у загальній масі трафіку мережі сягає понад 80-82%; у свою чергу, відеотрафіку відповідають найвищі рівні інформаційної інтенсивності, порівняно з іншими типами трафіку;
- найбільш критичними з точки зору чутливості до затримок та втрат пакетів є голос VoIP, сервіси VR та інтерактивного відео; також останнім часом зростає попит саме на ці типи відеосервісів.

При цьому існує тенденція до постійного зростання інформаційних інтенсивностей потоків, що належать відеосервісам як з причин зостання попиту на них та збільшення аудиторії користувачів, так і з причин росту роздільної здатності кадрів та частоти їх надходження.

Отже, для того, щоб створити умови, у яких потенційно можливо забезпечити підвищення рівня надання мережевих сервісів усіх типів, доцільним є скоротити обсяг пакетів (тобто, зменшити інформаційну інтенсивність), що належать відео сервісам та додаткам. Разом з тим, зменшення інформаційних інтенсивностей відео незмінно приводить до зниження якості реконструйованих даних.

Таким чином, у ході обробки відеотрафіку необхідно значну увагу приділяти розробці та удосконаленню методів кодування відеоданих як з втратами, так і без втрат.

Показано, що у складі базової схеми MPEG для обробки відеокадрів передбачено ряд технологічних етапів, на яких вносяться незворотні зміни у кодовані дані (тобто, виконується кодування з втратами):

- зміна формату колірної субдискретизації;

- квантування та округлення;
- ортогональне перетворення на базі ДКП.

У свою чергу, кодування без втрат забезпечується на таких етапах, як:

- RLE;
- статистичне кодування (за Хафманом або арифметичне).

При цьому, для фрагментів відеокадрів різного змісту існуючі методи кодування без втрат забезпечують також різні показники стиснення, але зазвичай не більше, ніж двократне зменшення відносно обсягу початкових даних. Разом з тим, найбільш продуктивні технології кодування з сімейства MPEG, зокрема, H.265/HEVC, дозволяють збільшити цей показник у середньому на 30%.

У той же час, методи кодування на базі формування кодових конструкцій змінної ваги не мають значної залежності між коефіцієнтом стиснення та особливостями змісту кодованих фрагментів.

Для них забезпечується більш оперативне, порівняно з статистичними методами, виконання алгоритму за рахунок:

- відсутності необхідності попереднього виконання етапів лінеаризації компонент та RLE;
- відсутності необхідності побудови алфавіту для кодування елементів певного сегменту даних;
- відносної простоти алгоритму.

При цьому, використання кодових конструкцій змінної ваги на етапі кодування без втрат, замість традиційних методів статистичного стиснення, дозволяє досягти збільшення коефіцієнту стиснення у середньому на 30% вищого.

Окрім цього, як свідчить виконана оцінка обчислювальної складності методу кодування, алгоритм є раціональним, оскільки загальна кількість елементарних операцій, що мають виконуватися одночасно, не перевищує показники продуктивності сучасних обчислювальних систем.

Тобто, метод кодування на базі використання кодових структур зі зміною вагою може бути імплементовано до складу технологій платформи MPEG, як альтернативу методам статистичного ущільнення з можливістю використання на базі неспеціалізованих технологічних платформ.

Також, внутрішня обчислювальна архітектура алгоритму містить велику кількість однотипних операцій, та їх однакових ланцюжків, які може бути поєднано у межах інструкцій, тим самим скорочуючи обчислювальне навантаження, що здійснюється на систему.

Таким чином, усі пункти завдання до атестаційної роботи виконано повністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. VNI Forecast Highlights Tool [Електронний ресурс] – Режим доступу: https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html
2. Video quality testing in mobile networks | Rohde &Schwarz [Електронний ресурс] – Режим доступу: https://www.rohde-schwarz.com/solutions/test-and-measurement/mobile-network-testing/expertise/video-quality-testing/video-quality-testing_232054.html?change_c=true
3. Гонта А. Практическое пособие по видеонаблюдению / А. Гонта. – М. : Полисет, 2017. – 188 с.
4. Кругль Г. Профессиональное видеонаблюдение. Практика и технологии аналогового и цифрового CCTV / Герман Кругль. – 2-е изд. пер. с англ. – М.: Секьюрити Фокус, 2011. – 640 с.
5. New Open Industry Standard Introduced for Connecting Next-Generation VR Headsets to PCs, Other Devices [Електронний ресурс] – Режим доступу: <https://www.globenewswire.com/news-release/2018/07/17/1538342/0/en/New-Open-Industry-Standard-Introduced-for-Connecting-Next-Generation-VR-Headsets-to-PCs-Other-Devices.html>
6. Качество обслуживания в операторских сетях [Електронний ресурс] – Режим доступу: https://www.opennet.ru/docs/RUS/qos_oper/
7. Milestone Content Portal [Електронний ресурс] – Режим доступу: <https://content.milestonesys.com/media/?mediaId=03F1B94C-5A2B-4031-870A431B57382C4D>
8. Haas, Erik. (2002). Aeronautical Channel Modeling. Vehicular Technology, IEEE Transactions on. 51. 254 - 264. 10.1109/25.994803.
9. For Self-Driving Cars, There’s Big Meaning Behind One Big Number: 4 Terabytes [Електронний ресурс] – Режим доступу: <https://newsroom.intel.com/editorials/self-driving-cars-big-meaning-behind-one-number-4-terabytes/>
10. Periscope: WATCH live [Електронний ресурс] – Режим доступу: <https://www.periscope.tv/>

11. Meerkat Live Streaming [Электронный ресурс] – Режим доступа: <https://www.meerkatapp.co/>
12. The Q-live game network [Электронный ресурс] – Режим доступа: <https://www.stre.am/>
13. Requirements and future of smart cities – Express Computer [Электронный ресурс] – Режим доступа: <https://www.expresscomputer.in/columns/icts-and-smart-cities/2434/>
14. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 3-е изд. – СПб. : Питер, 2006. – 958 с.
15. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1073 с.
16. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : ДИАЛОГ – МИФИ, 2003. – 384 с.
17. Чемпен Н., Чемпен Д. Цифровые технологии мультимедиа. – М.: Вильямс, 2006. – 624 с.
18. Ричардсон Ян. H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia / Ян Ричардсон. – Город. : Издательство, 2005. – 368 с.
19. Айфичер Эммануил С. Цифровая обработка сигналов: практический поход / Эммануил С. Айфичер, Барри У. Джервис. – 2-е изд. – М. : Вильямс, 2008. – 992 с.
20. Shi, Yun Q. Image and video compression for multimedia engineering: fundamentals, algorithms, and standards / Yun Q Shi, Huifang Sun.
21. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2004. – 368 с.
22. Баранник В.В. Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков – Х.: ХУПС, 2010. – 234 с.
23. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.
24. Мельник А. Архітектура комп'ютера. – Луцьк: ВНТУ, 2008.