

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ *Інфокомунікацій* _____
(повна назва)
Кафедра _____ *Інформаційно-мережної інженерії* _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ *другий (магістерський)* _____

_____ *Дослідження методів кодування без втрат у складі технологій* _____
_____ *стиснення відеоінформації* _____
(тема)

Виконав:
студент 2 курсу, групи _____ *ІММ-20-2* _____
_____ *Ресенко Я.Є.* _____
(прізвище, ініціали)

Спеціальність _____ *172. Телекомунікації та* _____
_____ *радіотехніка* _____
(код і повна назва спеціальності)

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

Освітня програма _____ *Інформаційно-мережна* _____
_____ *інженерія* _____
(повна назва освітньої програми)

Керівник _____ *ст.викл. Твердохліб В.В.* _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ *Безрук В.М.* _____
(підпис) (прізвище, ініціали)
2022 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник _____ /*В.В.Твердохліб*

Студент _____ /*Я. Ресненко*

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)
« 14 » березня 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Ресненко Ярославу Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів кодування без втрат у складі технологій стиснення відеоінформації

затверджена наказом університету від 14 березня 2022 р. № 379 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 27 травня 2022 р.

3. Вихідні дані до роботи Дослідити особливості розвитку технологічних концепцій телемедицини, голографічних комунікацій та автономних роботизованих систем. Визначити роль відео UHD для їх функціонування. Довести обмежені можливості існуючих методів кодування відеоданих щодо ефективної обробки відеоданих для забезпечення функціонування означених технологічних концепцій. Виконати аналіз одного з існуючих методів кодування без втрат, та методу, що використовує компактний опис сегментів у біноміально-поліадичному просторі. Обґрунтувати його ефективність порівняно з існуючими методами

4. Перелік питань, що потрібно опрацювати в роботі Вступ

1. Роль відеоінформації у впровадженні новітніх технологічних концепцій

2. Комплексні алгоритми кодування без втрат

3. Методи кодування без втрат, що входять до складу сучасних технологій обробки мультимедійних даних

4. Дослідження підходу до кодування без втрат на базі кодів з плаваючими вагами
Висновки

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, новітні технологічні
концепції, що використовують відеоінформацію, аналіз існуючих алгоритмів ущільнення
відеоданих, метод стиснення на базі кодів з плаваючими вагами)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ		
2	Роль відеоінформації у впровадженні новітніх технологічних концепцій		
3	Комплексні алгоритми кодування без втрат		
4	Методи кодування без втрат, що входять до складу сучасних технологій обробки мультимедійних даних		
5	Дослідження підходу до кодування без втрат на базі кодів з плаваючими вагами		
6	Висновки		
7	Оформлення пояснювальної записки		

Дата видачі завдання 14 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис) ст. викл. Твердохліб В.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 81 с., 26 рис., 4 табл., 17 джерел, 1 додаток

ВІДЕОКОДУВАННЯ, ТЕЛЕМЕДИЦИНА, СТРУКТУРНА
НАДМІРНІСТЬ, КОНТЕКСТНЕ МОДЕЛЮВАННЯ, Н.265, ПОЛІАДИЧНІ
ЧИСЛА

Об'єкт дослідження – методи кодування відеоінформації без втрат для збільшення коефіцієнту стиснення.

Мета роботи – дослідження можливостей удосконалення існуючих методів кодування відеоінформації шляхом застосування альтернативних методів кодування без втрат у складі технологій стиснення.

Виконано аналіз ряду існуючих новітніх технологічних концепцій, що знаходяться на стадії дослідження та часткового впровадження. Визначено роль відеоінформації для забезпечення функціонування кожної з них. Показано, що повноцінна робота зазначених перспективних проектів можлива за умови використання ними відео фіксації з надвисокою роздільною здатністю. Визначено недоліки існуючих технологій відео кодування. Окреслено ефективні підходи щодо їх удосконалення.

THE ABSTRACT

Explanatory note: 81p., 26 fig., 4 tabl., 17 sources, 1 app.

VIDEO CODING, TELEMEDICINE, STRUCTURAL EXCESS,
CONTEXT MODELING, H.265, POLYADIC NUMBERS

The object of research - methods of encoding video information without loss to increase the compression ratio.

The purpose of the work is to study the possibilities of improving the existing methods of video information encoding by using alternative lossless encoding methods in the compression technologies.

The analysis of a number of the existing newest technological concepts which are at a stage of research and partial introduction is executed. The role of video information to ensure the functioning of each of them is determined. It is shown that the full-fledged work of these promising projects is possible if they use video recording with ultra-high resolution. The shortcomings of existing video coding technologies have been identified. Effective approaches to their improvement are outlined.

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ.....	10
ВСТУП.....	11
1 РОЛЬ ВІДЕОІНФОРМАЦІЇ У ВПРОВАДЖЕННІ НОВІТНІХ ТЕХНОЛОГІЧНИХ КОНЦЕПЦІЙ	14
1.1 Перспективні технологічні концепції та проблематика їх розвитку	14
1.1.1 Телемедицина.....	14
1.1.2 Складові компоненти системи телемедицини	15
1.2.1 Системи голографічних комунікацій.....	18
1.2.2 Складові компоненти системи голографічних комунікацій.....	19
1.3.1 Автономні роботизовані системи	21
1.3.2 Структура автономної роботизованої системи	22
1.4 Попередні висновки за розглянутим матеріалом	25
1.5 Базові принципи побудови існуючих методів стиснення відео ...	26
2 КОМПЛЕКСНІ АЛГОРИТМИ КОДУВАННЯ БЕЗ ВТРАТ.....	29
2.1 Загальні напрямки побудови комплексних алгоритмів кодування	29
2.1.1 Статистичні та контекстні алгоритми стиснення.....	29
2.2 Загальні підходи щодо побудови комплексних алгоритмів кодування ...	31
2.2.1 Алгоритми контекстного кодування загальної групи	32
2.2.2 Спеціалізовані алгоритми	34
2.3 Чинники, що можуть впливати на потенційну ефективності загальних комплексних алгоритмів кодування	39
2.4 Висновки за розглянутим матеріалом	42
3 МЕТОДИ КОДУВАННЯ БЕЗ ВТРАТ, ЩО ВХОДЯТЬ ДО СКЛАДУ СУЧАСНИХ ТЕХНОЛОГІЙ ОБРОБКИ МУЛЬТИМЕДІЙНИХ ДАНИХ.....	43
3.1 Загальні відомості про технологічні етапи кодування у складі технологій компресії мультимедіа	43
3.2 Аналіз функціонування процесу стиснення без втрат на базі методу Хафмана.....	44

3.2.1 Загальна схема технологічного процесу кодування без втрат під час обробки зображень.....	44
3.3 Аналіз функціонування методу Хафмана.....	45
3.3.1 Процес кодування даних на базі методу Хафмана	46
3.3.2 Формування бінарного дерева Хафмана	47
3.3.3 Приклад кодування за Хафманом з графічною інтерпретацією процесу кодоутворення	48
3.4 Аналіз ефективності процесу та результатів кодоутворення на базі методу Хафмана	54
3.4.1 Інтерпретація даних, отриманих у ході дослідження ефективності методу Хафмана	57
3.5 Висновки за розділом	58
4 ДОСЛІДЖЕННЯ ПІДХОДУ ДО КОДУВАННЯ БЕЗ ВТРАТ НА БАЗІ КОДІВ З ПЛАВАЮЧИМИ ВАГАМИ.....	59
4.1 Вимоги до методу кодування без втрат, спрямованого на обробку графічних даних	59
4.2 Загальна концепція побудови методу кодування без втрат сегментів зображень у біноміально-поліадичному просторі	59
4.2.1 Формування компактного представлення сегменту зображення .	60
4.2.2 Алгоритм реконструювання сегментів зображення, кодованих у біноміально-поліадичному просторі	65
4.3 Порівняльний аналіз методів кодування за Хафманом та кодування на базі біноміально-поліадичної системи зчислення з позицій ефективності застосування у складі технології стиснення відео	68
4.3.1 Розрахунок швидкості кодоутворення для кодування за методом Хафмана	68
4.3.2 Розрахунок швидкості кодоутворення для кодів з плаваючими вагами	74
ВИСНОВКИ.....	78
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	80
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ	82

ПЕРЕЛІК СКОРОЧЕНЬ

IMT2020 –технологія високошвидкісного бездротового зв'язку 5 покоління;

WiMAX - технологія високошвидкісного бездротового зв'язку;

HSPA+, HSPA - технології бездротового зв'язку;

MPEG – сімейство стандартів відео кодування;

FullHD – (full-high definition) – повноформатне відео високої роздільної здатності;

HTTP/HTTPS – транспортні протоколи;

RLE – (run-length encoding) – технологія кодування ланцюжків двійкових символів;

JPEG – технологія кодування зображень;

UHD – (ultra-high definition) – загальна назва форматів відео та статичних зображень, починаючи з формату 4K і вище;

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;

JPEG2000 – технологія кодування зображень;

VBR – (variable bit-rate) – режим кодування, при якому вихідний сигнал формується зі змінною бітовою швидкістю;

CBR – (constant bit-rate) – режим кодування, при якому вихідний сигнал формується зі сталою бітовою швидкістю;

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

ВСТУП

У сьогоднішніх умовах сучасний рівень розвитку комунікаційних та інформаційних технологій та систем дає змогу суттєво розширити діапазон послуг, що надається на базі мереж. Зараз, зокрема, повністю реалізовано та сприймаються вже звичними:

- електронний банкінг;
- гео-інформаційні сервіси;
- сервіси замовлення та доставки товарів та послуг;
- різноманітні навчальні проекти;
- системи віртуальної реальності тощо.

Поряд з зазначеними проектами отримують утілення концептуально нові, наприклад:

- системи голографічної комунікації;
- телемедицина;
- автономні роботизовані системи тощо.

Реалізація хоча б жодного з перелічених напрямків раніше була фактично неможливою, як наслідок дії ряду чинників, зокрема:

- недостатня продуктивність каналів інфокомунікаційних мереж;
- надмірна інтенсивність джерела даних, що є, у сутності неспіврозмірною з існуючою пропускнуою здатністю мережі;
- занадто низька обчислювальна потужність кінцевих вузлів для того, щоб мати змогу прийому та обробки даних у реальному часі з по-трібними кондиціями якості, яких вимагає та чи інша система.

У свою чергу, на сьогодні проблематику недостатньої продуктивності каналів інфокомунікаційних мереж частково усунено. Це викликано впровадженням високошвидкісних технологій доступу - (HSPA, HSPA+, WiMAX, частково та у тестовому режимі – IMT2020 [1].

Також достатньо глибоко розв'язано проблематику недостатньої обчислювальної потужності клієнтських терміналів. Зокрема, цього вдалося досягти застосуванням систем балансування навантаження (big/little), впровадженням більш швидкісної архітектури тощо.

У той же час, проблема високої середньої інтенсивності джерел даних у мережі, яка постійно зростає, чим, фактично, повністю нівелює ефект від зростання пропускнув здатностей, залишається актуальною.

Найбільш суттєві фактори впливу тут – розширення аудиторії користувачів різноманітних сервісів на базі мережі, збільшення кількості сервісів, та недосконалість методі ущільнення даних.

Останній фактор може вважатися, з одного боку, таким, що є одним з найбільш важливих а з іншого боку – таким, який може бути якщо не усунуто повністю, то значно скорочено його вплив.

Так, у веб-середовищі сьогодні своєрідним стандартом є алгоритми ущільнення gzip та/або метод Brotli, які дозволяють скоротити обсяги надходячих у мережу текстових даних на базі протоколів HTTP/HTTPS, та підтримуються більшістю відомих браузерів та додатків, що забезпечують файлообмін.

У свою чергу, для ущільнення аудіо даних (потоків мовлення, VoIP тощо) широко застосовуються стандартизовані сімейства кодеків, наприклад, G711 для VoIP, або MP3 для AoD чи поточкових сервісів.

При цьому, для компактного передавання графічної інформації, залежно від її призначення, зараз найбільш широко використовується близько десятку технологій ущільнення.

Проте, жоден з перелічених типів даних не є критичним з позиції надмірного рівня інтенсивності, що ними породжується [1, 2].

Найбільш суттєвий рівень навантаження на мережу складає відеоінформація, що, по-перше, знаходиться в основі кожної з оголошених попередньо технологічних концепцій (системи голографічної комунікації, телемедицина та автономні роботизовані системи), а по-друге, на сьогодні займає, щонайменше, 4/5 усього трафіку у масштабі Всесвітньої мережі. Звідси є зрозумілим те, що суттєве зменшення середньої інтенсивності джерел даних, та, як наслідок цього, скорочення навантаження на мережу можливо шляхом ущільнення саме мережевої відеоінформації [2].

Однак, слід пам'ятати, що падіння інтенсивності відеоінформації шляхом її ущільнення спричинює у загальному випадку погіршення якості відновлених даних, як візуальної, так і вимірної за однією з відповідних метрик, що застосовуються сьогодні.

Отже, складаються умови, коли, з одного боку, ущільнення відеоінформації є не лише необхідним, але і потребує постійного збільшення його ступеню. З іншого ж боку, ряд систем та сервісів, зокрема, перелічені вище, не можуть функціонувати в умовах, коли у наслідок значного ступеню

ущільнення відеоінформації рівень помилки реконструйованих даних буде вищим деякого порогового значення.

Таким чином, розробка, удосконалення та застосування алгоритмів кодування без втрат є актуальним завданням.

1 РОЛЬ ВІДЕОІНФОРМАЦІЇ У ВПРОВАДЖЕННІ НОВІТНІХ ТЕХНОЛОГІЧНИХ КОНЦЕПЦІЙ

1.1 Перспективні технологічні концепції та проблематика їх розвитку

1.1.1 Телемедицина

Сьогодні телемедицина є принципово новою галуззю охорони здоров'я, яка передбачає використання електронні комунікації та інформаційні технології для надання медичної допомоги на відстані [3].

Перший випадок застосування телемедицина було зафіксовано у 1935 році, коли було налагоджено постійне використання телеелектрокардіографії Вітольдом Липинським і Мар'яном Франке.

На сьогодні можливості телемедицини суттєво розширилися. Це дає змогу:

- проведення медичних консультації у віддаленому режимі;
- трансляції хірургічних операцій у ході навчання майбутніх фахівців, у т.ч і у режимі «теленаставництва», тобто, коли досвідчений хірург координує дії свого менш досвідченого колеги;
- більш широкого застосування ендоскопічних технологій, що дає змогу проведення оперативного втручання з мінімальним травмуванням пацієнтів;
- ведення негласного контролю стану здоров'я особового складу з безконтактним вимірюванням температури тіла, магнітоенцефалограми, магнітокардіограми тощо;
- застосовувати мобільні телемедичні комплекси, що є актуальним для роботи на місцях техногенних та априродних катастроф, аварій;
- використання систем дистанційного біомоніторингу, що дасть змогу контролю стану здоров'я пацієнтів, що страждають хронічними захворюваннями та працівників об'єктів критичної інфраструктури;
- розвивати т.з. домашню телемедицину, тобто, дистанційно надавати медичні послуги пацієнтам, що знаходяться на домашньому лікуванні, або на значному віддаленні від найближчого медичного пункту, при цьому, їх стан дозволяє проходити курс лікування самостійно, але під наглядом лікаря.

Зараз утілено велику кількість проектів з галузі телемедицини. Наприклад - Satelife та Planet Heres орієнтуються, головним чином, на розповсюдження медичних знань та навчання кадрів. При цьому, проект НЕСТОР є прикладом платформи для управління мобільними телемедичними комплексами, а HOMER-D – для координації лікування у домашньому режимі. У свою чергу, проект ІТНАСА, та подібні йому, поєднує у собі усі зазначені вище напрямки діяльності, тобто, є значно більш масштабним.

На території Європи станом на 2016 рік фіксувалося надання порядку 30% медичних послуг на засадах телемедицини, що стосувалося консультації з лікарями та контролю ними перебігу стану лікування пацієнтів.

У той же час, пандемія COVID-19 спричинила додатковий імпульс щодо розвитку телемедицини у всьому світі.

На сьогодні за різними оцінками відсоток медичних послуг, наданих на її засадах, наближається до 45-47%.

1.1.2 Складові компоненти системи телемедицини

Розглянемо приклад уніфікованої системи телемедицини, тобто, такої, що здатна надавати послуги як консультативного характеру, так і будь-яких інших типів з зазначеного. При цьому, до її складу входять (рис.1.1) [3]:

1. Центр збору, контролю та обробки інформації.
2. Термінальне обладнання фахівців.
3. Системи збору біоінформації, зокрема:
 - температури тіла (ТТ);
 - артеріального тиску (АТ);
 - магнітоенцефалограми (МЕГ);
 - магнітокардіограми (МКГ);
 - маси тіла (МТ);
 - глюкометричного індексу (ГІ).

Майже усі перелічені системи складаються з відповідних датчиків, клієнтського додатку та серверної частини, яка веде збір статистики та, за потреби, інформує фахівців про відхилення від норми тих чи інших показників.

З точки зору вимогливості до параметрів мережі дані системи є некритичними, так як, по-перше, обсяг даних, який збирається даними системами, є досить незначним, та у сумі майже ніколи не перевищує одиниць а ще рідше - десятків кілобайт. По-друге, допустимим є деяка затримка надходження означених даних до центру збору, контролю та обробки інформації.

4. Системи візуального моніторингу.

Дана складова компонента є обов'язковою у випадку проведення ендоскопічних досліджень, ендоскопічних хірургічних втручань як у штатному режимі, так і таких, перебіг яких контролюється у реальному часі консилиумом фахівців відповідного профілю, що знаходяться віддалено, а також коли фіксується сітківка ока пацієнта [4].

До складу даної системи обов'язково входять камери зовнішньої фіксації, камери високої роздільної здатності та спеціалізовані ендоскопічні камери.

При цьому, камери зовнішньої фіксації використовуються для моніторингу усього перебігу процесу ззовні. Особливих вимог відносно них не висувається. Це можуть бути звичайні IP-камери з роздільною здатністю FullHD або HD-Ready.

У свою чергу, ендоскопічні камери (рис.1.2), та камери для спостереження сітківки мають забезпечувати винятково якісне зображення на дисплеях будь-яких типів. Тому стандартом зараз стає роздільна здатність не нижче, ніж 4К.

При цьому, додатковими вимогами є:

- можливість забезпечувати частоту слідування кадрів на рівні 60 та 120 к/с;

- можливість підтримки відеоряду у 32-розрядному форматі опису.

Такі вимоги зумовлені:

- необхідністю детального огляду ділянок оперативного втручання та контролю їх стану;

- для підтримки якісної візуалізації на будь-якому моніторі;

- забезпечення якісного зображення у режимі digital zoom потребує високої деталізації та високого рівня градації колірних відтінків (оптичний zoom неможливий, так як ендоскопічна камера є обмежена у габаритах).

Такі камери створюють найвище навантаження, так як інтенсивність, наприклад, відео потоку, що породжує одна камера NT604K може знаходитися у діапазоні від 44,7 (при 60 к/с), до майже 90 Мбіт/с (при 120 к/с) за умови, що забезпечується 32-розрядний колірний опис.

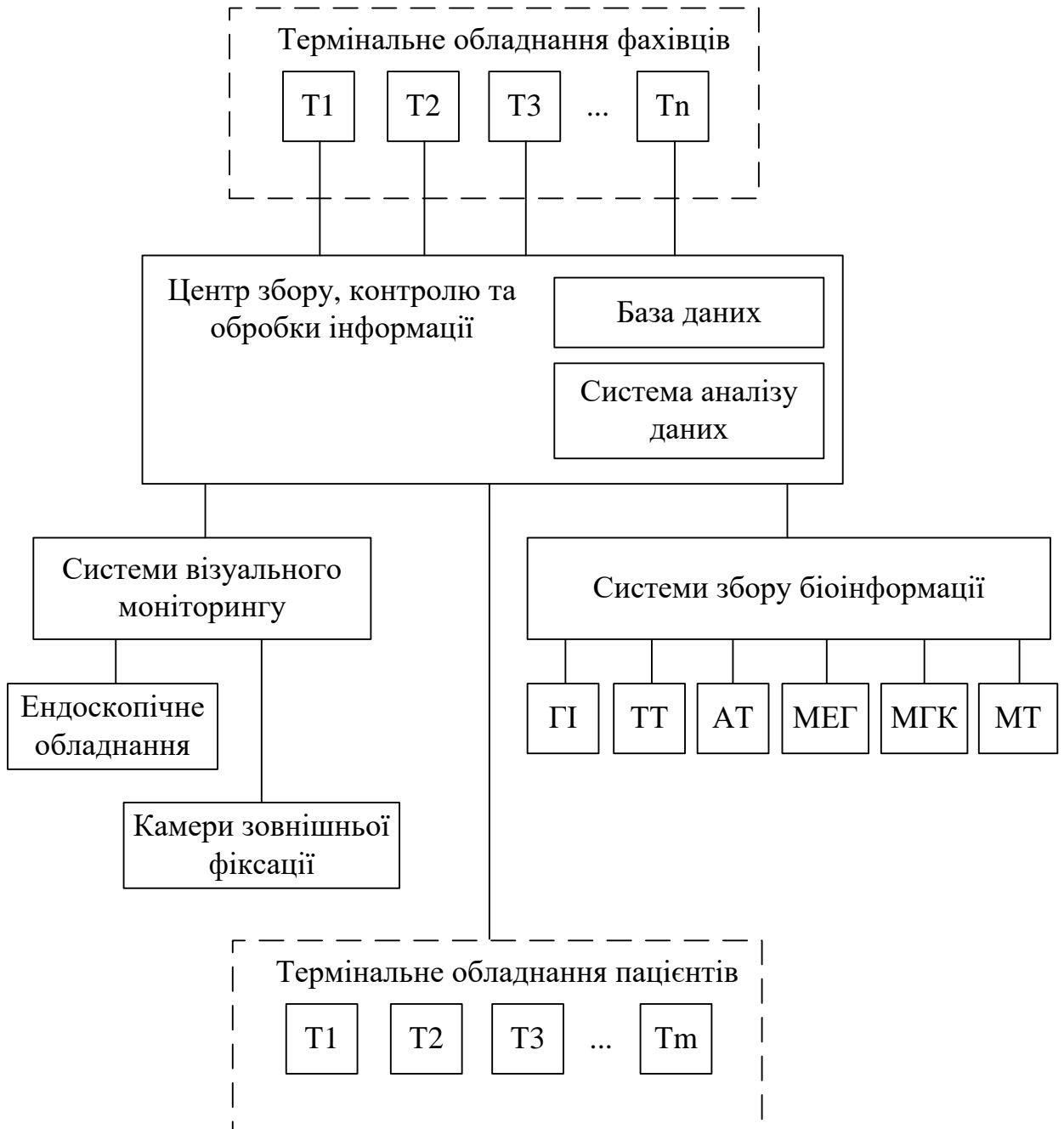


Рисунок 1.1 - Складові компоненти системи телемедицини

Наведені номінали інтенсивностей відповідають випадкам ущільнення відеоряду на рівні джерела з використанням кодеку H.264/AVC. Тобто, для

більшості мереж доступу забезпечення передачі відеоряду від ендоскопічної камери є фактично невирішеною задачею. Для її вирішення може бути застосовано або мережу доступу більш потужного стандарту (наприклад, WiMAX, або Gigabit Ethernet), або більш досконалий алгоритм ущільнення.

У першому випадку, коли мова йде, наприклад, про мобільні телемедичні комплекси, наявність швидкісної мережі доступу не може гарантуватися. У другому ж випадку, говорячи про алгоритми ущільнення, необхідно пам'ятати про те, що падіння інтенсивності відео інформаційного потоку веде до зниження якості отриманого відеоряду.



Рисунок 1.2 – Ендоскопічна камера NT604K

1.2.1 Системи голографічних комунікацій

На сьогодні системи голографічних комунікацій являють собою еволюцію систем відеозв'язку. При цьому, від традиційного відеозв'язку їх відрізняє наступне [5]:

- демонстрація співрозмовника та, взагалі, віддаленої сцени потребує від 3 і більше проєкційних пристроїв замість екрану;

- фіксація співрозмовника у ході віддаленої взаємодії здійснюється декількома відеокамерами (щонайменше трьома) з різних ракурсів, які у ході комунікації можуть змінюватися;

- для побудови відеоголограми кожне з джерел відеоінформації, задіяне у системі, повинне мати просторову роздільну здатність одного з UHD форматів, починаючи з 4K.

Також у випадку використання джерел відеоінформації у динамічному режимі, має забезпечуватися їх взаємна синхронізація, та синхронізація з модулем генерації комбінованого голографічного зображення на боці прийому.

1.2.2 Складові компоненти системи голографічних комунікацій

Загальна структурна схема системи голографічних комунікацій демонструється рис. 1.3.

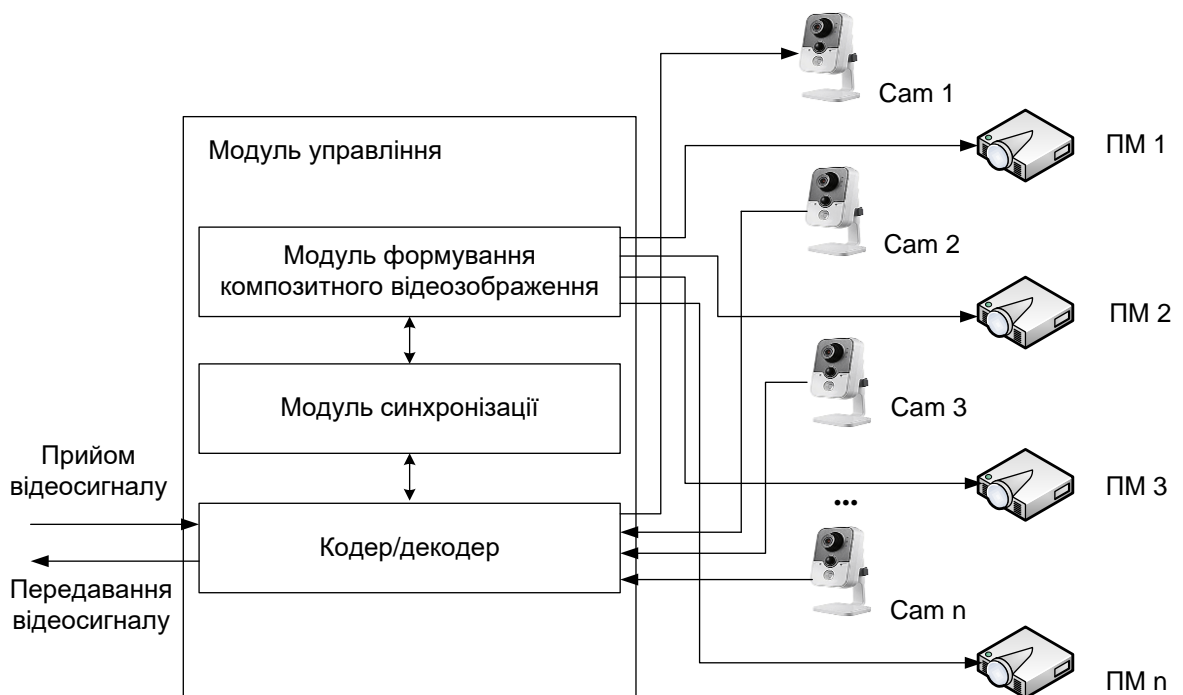


Рисунок 1.3 – Структурна схема системи голографічних комунікацій

У загальному випадку типова структура системи голографічного зв'язку включає у себе множину проєкційних пристроїв (ПП 1- ПП n), множину пристроїв відео фіксації (Cam 1 – Cam n) та модуль управління.

У свою чергу, модуль управління містить у собі пристрій кодування та декодування, що забезпечує компактне представлення інформації, яка надсилається до мережі, а також відновлення ущільнених даних, прийнятих від передавача.

При цьому, функції модулю синхронізації полягають у:

- узгодженні роботи камер/проєкційних пристроїв;
- активації та деактивації додаткових пристроїв фіксації у ході сеансу зв'язку;
- контролі параметрів кодування/декодування та ін.

Врешті речт, модуль формування композитного відео зображення виконує функції:

- прийому, аналізу та інтерпретації даних метаопису окремих відео потоків;
- управління розподілом відео потоків між проєкційними пристроями на базі прийнятих даних мета опису та відповідно до надходячих команд синхронізації.

Зараз існує ряд тестових систем голографічного зв'язку, які для повноцінного функціонування яких потребується стабільний канал з пропускнуою здатністю порядку 1-1,2 Гбіт/с. Це пояснюється тим, що:

- у системі з кожного боку задіяно ряд UHD пристроїв фіксації, кожне з яких породжує відеопотік інтенсивністю не нижче, ніж 75 Мбіт/с (розрахунки справедливі для випадку ущільнення відеоряду на базі H.264/AVC та H.265/HEVC);
- голографічний зв'язок функціонує як повноцінна дуплексна система, для якої передбачається резервування пропускнуої здатності у напрямку «Передавач → Приймач» та у зворотньому напрямку;
- для забезпечення функціонування системи використовується ряд каналів синхронізації, трафік яких має нульовий пріоритет, що створює додаткове навантаження на систему.

Як було виявлено у ході тестування голографічних каналів, зниження якості відеоінформації хоча б від одного з джерел на боці передавача, здатне суттєво зменшити як візуальну якість згенерованого композитного відео

зображення на прийомному боці, так і якість, оцінену за об'єктивними показниками.

У той же час, імплементація систем голографічних комунікацій для широкого загалу у розділеному середовищі навіть у тестовому варіанті на сьогодні є неможливою. Так само вона є неможливою і на базі мереж доступу 5 покоління, зокрема, на базі ІМТ2020, так як технологічні кондиції даної технології (1Гбіт/с upload, 10Гбіт/с download) не гарантують функціонування системи навіть у симплексному режимі.

Отже, як і у випадку телемедицини, існує потреба у скороченні інтенсивності відеоінформаційних потоків, що породжуються джерелами системи. Разом з тим, виникаючі у наслідок зменшення інтенсивності відеоданих (шляхом ущільнення) помилки навіть за умови, що візуальна якість окремо взятих потоків помітно не змінюється, ведуть до помітного або навіть суттєвого руйнування композитного проекційного відеозображення.

1.3.1 Автономні роботизовані системи

Сьогодні роботизовані системи знаходяться на стадії впровадження, при цьому, у рамках окремих галузей дані системи набули застосування. Це, у першу чергу, стосується [6]:

- виробництв у технологічних процесах конвеєрного типу;
- технологічних циклів у хімічної, харчової, легкої та важкої промисловостей, де участь людини є небажаною з причини небезпеки для здоров'я;
- систем контролю доступу;
- різноманітних smart-концепціях (smart city, smart house тощо);
- системах спеціального призначення та ін.

При цьому, функціонування роботизованих систем в автономному режимі зараз має ряд обмежень, а саме:

- на випадок виникнення позаштатної ситуації система потребує обов'язкового втручання оператора;
- можливості таких систем щодо самодіагностування частіше за все дозволяють лише відстежувати працездатність вузлів та агрегатів;
- можливості щодо навчання роботизованих систем обмежуються штатними умовами.

Для того, щоб усунути дані обмеження, роботизовані системи удосконалюються за рахунок використання технологій комп'ютерного зору.

Це, у свою чергу, дозволяє:

- виконувати розпізнавання об'єктів, з якими прямо чи опосередковано взаємодіє система (обличчя, сітківка ока, навколишній простір та оточуючі об'єкти, деталі та матеріали);
- здійснювати ефективне навчання системи.

Як наслідок – система стає адаптивною до змін умов функціонування, що, у свою чергу, дає змогу:

- більш точно позиціонувати рухомі елементи системи;
- мінімізувати ймовірність виникнення аварійних ситуацій у наслідок змін умов або при виникненні передумов до них;
- досягти теоретично повної автономності системи.

Далі розглянемо склад автономної роботизованої системи та оцінимо проблематику, що виникає у наслідок інтеграції систем комп'ютерного зору до її складу.

1.3.2 Структура автономної роботизованої системи

Типовий структурний склад експериментальних автономних роботизованих систем, що використовують технології комп'ютерного зору, показано на рис. 1.4.

Як видно з рис. 1.4, система умовно містить 3 базових складових – підсистему управління, підсистему управління двигунами та підсистему сенсорів і датчиків.

У свою чергу, підсистема двигунів складається з:

- ДПС – двигунів постійного струму;
- КЕД – крокових електричних двигунів;
- ПД – п'єзо двигунів;
- ПМ – «повітряних м'язів» (пристроїв для забезпечення тяги на базі пневмосистем з імітацією м'язових волокон);
- ЕАП – електроактивних полімерів – тягових пристроїв на базі особливого класу полімерів, що змінюють форму у відповідь на електричну стимуляцію;

- ДЕН – двигунів на базі еластичних нанотрубок – ще один клас пристроїв для створення тяги, що базується на принципі роботи м’язів.

При цьому, до складу підсистеми датчиків та сенсорів входять:

- ВД – відеодатчики;
- ВД UHD – відеодатчики надвисокої роздільної здатності;
- ДД – датчики дотику;
- ДО – датчики освітлення;
- ДВ – датчики відстані;
- ДГ – датчики гіроскопічні;
- ЕД – датчики-ехолоти.

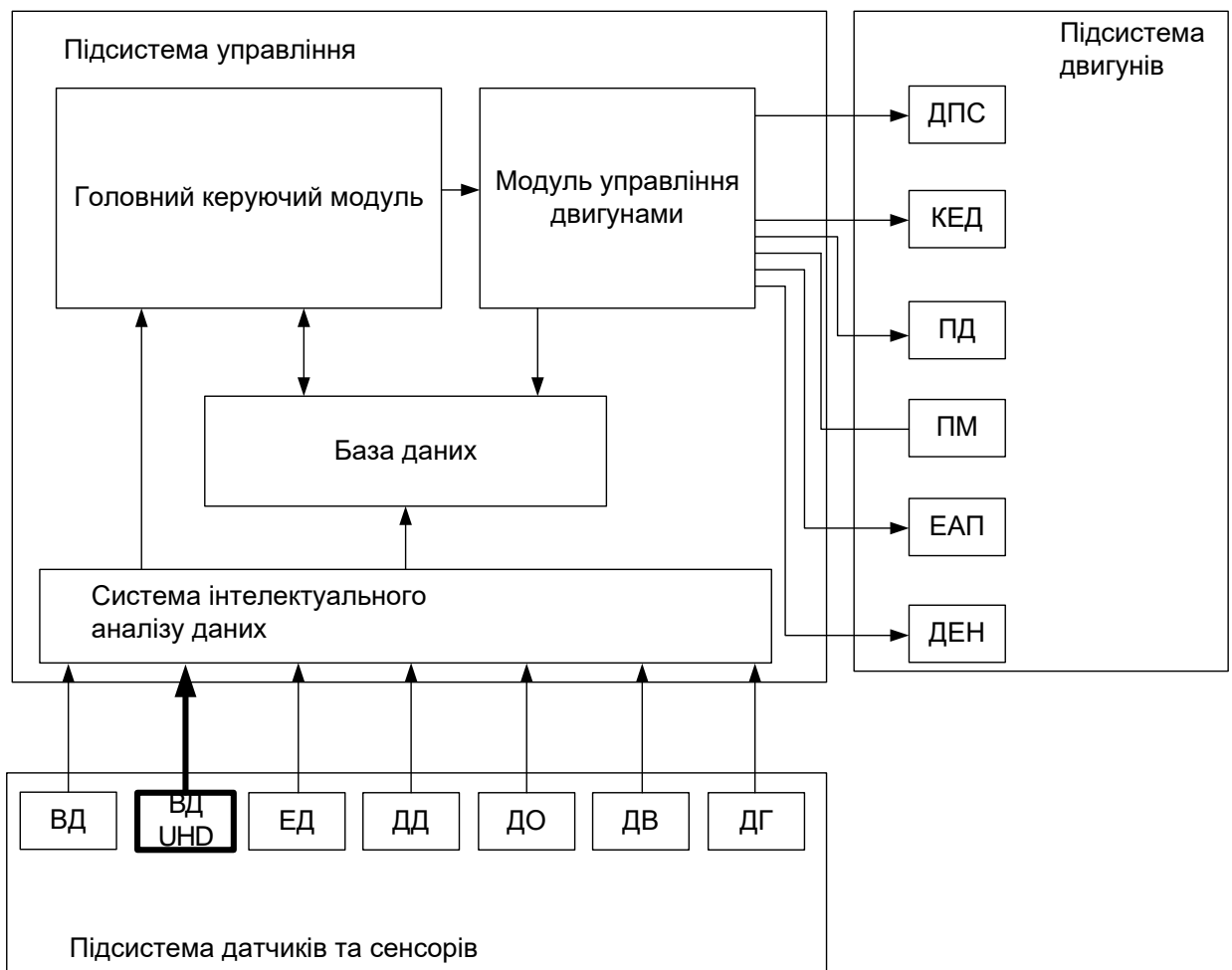


Рисунок 1.4 – Загальна схема автономної роботизованої системи

Підсистема управління при цьому формується такими компонентами, як:

- головний керуючий модуль;

- модуль управління двигунами;
- база даних;
- система інтелектуального аналізу даних.

Слід зазначити, що інтеграція технологій комп'ютерного зору, яка базується на розпізнаванні об'єктів, залежно від функцій та спрямованості системи потребує використання пристроїв відеофіксації, які ведуть зйомку у режимі UHD на рівні 4-8К.

До того ж, для ряду роботизованих технологічних систем необхідно вести фото та відеофіксацію не лише у видимому діапазоні, але також додатково, наприклад, у діапазоні UF. Відповідно, у таких умовах накладаються додаткові вимоги відносно пропускну здатності каналів зв'язку, якими надсилаються дані відео фіксації до систем інтелектуального аналізу даних.

Такі системи майже завжди реалізовано у вигляді управляючого серверного ПЗ і бази даних (рис.1.4). Дана архітектура зумовлена, по-перше, тим, що керуючий модуль повинен мати змогу рівно ймовірного звернення до будь-якого кінцевого пристрою системи, а також до бази даних, яка у процесі функціонування системи (у режимі навчання чи штатному) наповнюється. Отже, локальне розміщення управляючих модулів для кожного з вузлів окремо є не виправданим ні з технологічної, ні економічної точки зору.

Слід зазначити, що у випадку взаємно компактного розміщення кінцевих пристроїв роботизованої системи та модулів керування, тобто, коли можлива їх реалізація у межах ЛМЗ підприємства, проблема забезпечення трансляції відеоінформації у форматі 4-8К (приблизно 75 – 320 Мбіт/с для одного потоку) може бути вирішена за рахунок виділення кожному з кінцевих пристроїв високошвидкісної фізичної лінії зв'язку (зокрема, на базі Gigabit Ethernet).

У той же час, для розподілених територіально систем означена можливість відсутня.

Так, як свідчить аналіз перспективної моделі концепції Smart City, у її рамках існує ряд систем, що є значно рознесені географічно.

Для таких систем взаємодія між кінцевими пристроями та керуючими модулями здійснюється загальним розподіленим середовищем, у якості якого виступають мережі доступу загального призначення.

Прикладами таких роботизованих систем є інтелектуальна транспортна система, системи контролю доступу, розподілені лінії виробництв закритого циклу тощо.

1.4 Попередні висновки за розглянутим матеріалом

Визначено, що кожна з перспективних технологічних концепцій – телемедицина, автономні роботизовані системи та системи голографічного зв'язку тією чи іншою мають використовувати відеодані надвисокої роздільної здатності, які при цьому необхідно транслювати мережею.

Як зазначалося раніше, існуючі номінали пропускної здатності поширених сьогодні мереж доступу не дозволяють забезпечити трансляцію відео 4-8К у реальному часі. Це ж є справедливим для технологій 5 покоління, таких, як IMT2020 та подібних.

Таким чином, відеоряд, яких надсилається у мережу, має стискатися більш суттєво, ніж дозволяють зараз існуючі технології кодування – MPEG-4, H.264/AVC, H.265/HEVC.

Разом з тим, збільшення коефіцієнту стиснення на базі означених технологій незмінно веде до втрати об'єктивної та візуальної якості відновленого відеоряду і як наслідок – функціонування систем, що його використовують, не може бути гарантовано.

Отже, актуальним завданням є доробка існуючого технологічного базису кодування відеоінформації таким чином, щоб мінімізувати помилку, яка вноситься при підвищенні коефіцієнту K_{comp} стиснення.

Це еквівалентно наступному виразу:

$$\begin{cases} K_{\text{comp}} \uparrow \\ \Delta F = \text{const} \end{cases} \quad (1.1)$$

Разом з тим, з огляду на те, що кожен окремо взятий відеокадр UHD має роздільну здатність, що у 4 (4K), 16 (8K) та ще більше разів перевищує роздільну здатність FullHD, яка є зараз своєрідним стандартом, це, у свою чергу, суттєво збільшує навантаження на систему у ході кодування, та фактично унеможливує обробку UHD на більшості неспеціалізованих пристроях у реальному часі.

Тобто, доробка існуючих методів стиснення має виконуватися таким чином, щоб забезпечувалося зменшення складності алгоритму.

Для визначення ймовірних шляхів доробки існуючих методів кодування спочатку розглянемо принципи їх побудови.

1.5 Базові принципи побудови існуючих методів стиснення відео

В основі роботи будь-якого з поширених методів кодування відео знаходиться скорочення того чи іншого (або кількох одразу) типу надмірності.

Так як близько 95% усіх кодеків, які існують сьогодні, відносяться до різних поколінь та модифікацій MPEG, за основу візьмемо саме це сімейство технологій [7, 8].

У ході обробки за MPEG усувається часова надмірність (міжкадрове кодування) та надмірність просторова (внутрішньо кадрова обробка).

У свою чергу, найбільшим чином на загальний коефіцієнт стиснення впливає саме внутрішньо кадрова обробка, тому далі розглянемо методи цієї групи.

Загальна схема внутрішньо кадрової обробки, прийнятої сьогодні у MPEG, базується на алгоритмові JPEG з додаванням допоміжних механізмів, таких, зокрема, як додатковий поділ ділянок кадру та лінійне прогнозування. Така схема передбачає скорочення ряду типів надмірності, як показано на рис. 1.5. Слід додати, що внутрішньо кадрову обробку теоретично можливо організувати на базі JPEG2000.

Так, психо-візуальна надмірність, скорочується на етапах зміни формату колірною представлення (етап зміни колірною простору є підготовчим) та квантування. Зокрема, на етапі зміни формату колірною представлення усувається від 25 до 75% інформації про колірність. Дана процедура є незворотною, тобто, внесені зміни не можуть бути реконструйовані. Формат колірною представлення змінюється для колірною-різницевого каналів – хроматичної червоної (Cr) та хроматичної синьої (Cb). При цьому, яскравісний канал залишається незмінним.

У свою чергу, на етапі квантування, на якому здійснюється ділення компонент на коефіцієнти матриці квантування, також скорочується психо-візуальна надмірність.

Даний технологічний етап. Як і попередньо розглянутий, вносить незворотні зміни у вихідні дані. На цьому технологічному етапі чисельні значення компонент може бути скорочено до десятків разів.

З усього ланцюга перетворень JPEG саме тут досягається найбільше значення коефіцієнту K_{comp} стиснення

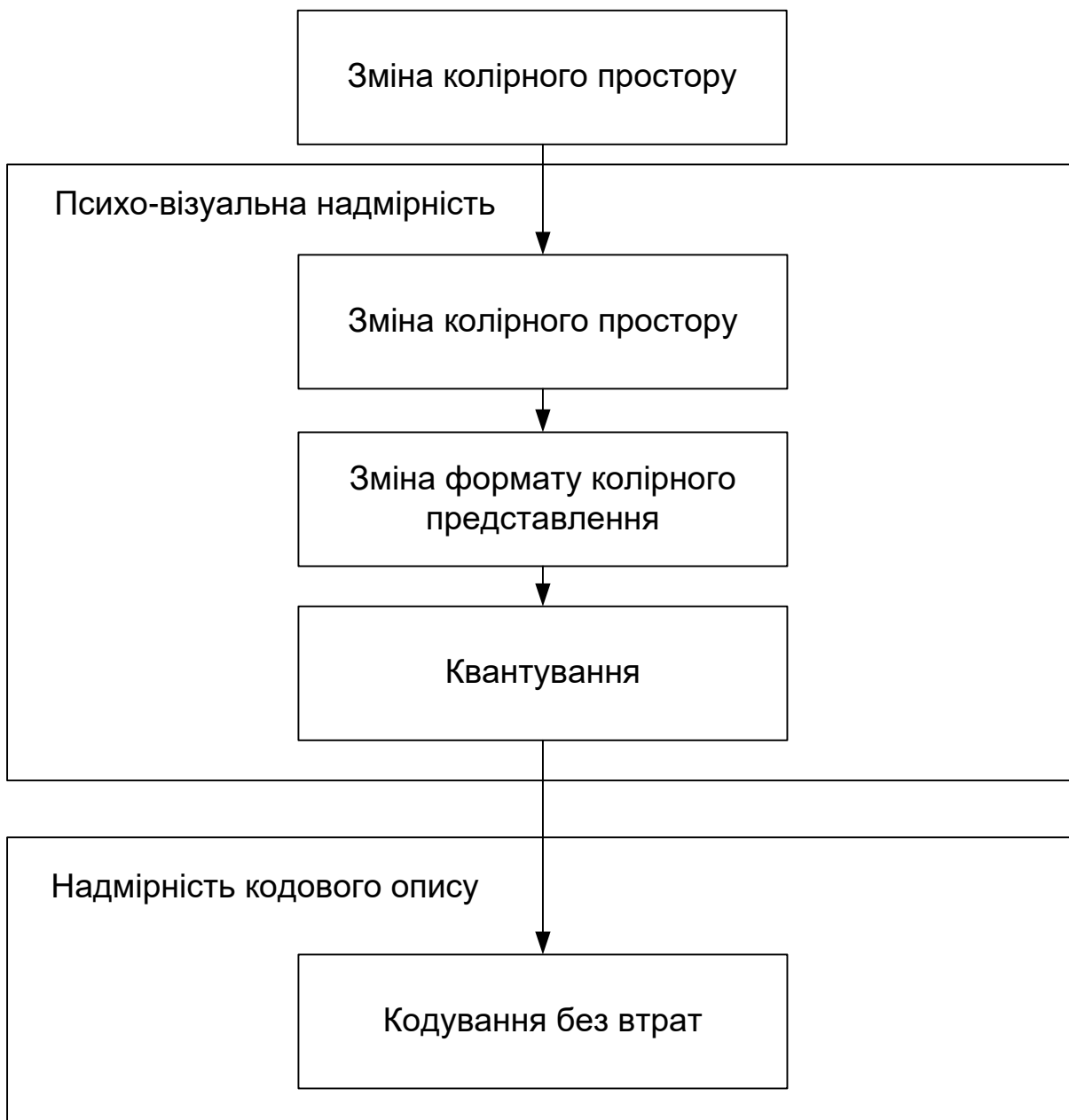


Рисунок 1.5 – Функціональні технологічні етапи, що застосовуються у схемі JPEG для скорочення різних типів надмірностей

Отже, і етап зміни формату колірною представлення, і етап квантування вносять похибку у реконструйовані дані. Відповідно, застосування на цих етапах опцій, що ведуть до росту коефіцієнту стиснення, спричинює виникнення ще більшої похибки.

Натомість, етап кодування без втрат, як фінальний крок обробки відеоінформації, з одного боку, не вносить додаткових викривлень.

На сьогодні базис JPEG дозволяє застосовувати для цього або кодування за методом Хафмана, або арифметичне кодування.

На етапі кодування без втрат передбачається виконання як самого кодування безпосередньо, так і передуючих дій, у ході яких вихідні дані форматуються до вигляду, який сприяє більш ефективному подальшому кодуванню. Проте недоліком даних методів кодування є відносно невеликі коефіцієнти K_{comp} стиснення, що не можуть задовольнити сучасні вимоги.

Разом з тим, можемо висловити припущення, що саме модифікація алгоритму внутрішньо кадрового кодування шляхом зміни чи доробки технологічного етапу кодування без втрат здатна забезпечити виконання умов (1.1).

Далі розглянемо особливості алгоритмів стиснення без втрат, що набули сьогодні широкого застосування.

2. КОМПЛЕКСНІ АЛГОРИТМИ КОДУВАННЯ БЕЗ ВТРАТ

2.1 Загальні напрямки побудови комплексних алгоритмів кодування

2.1.1 Статистичні та контекстні алгоритми стиснення

У сучасних умовах ефективні алгоритми стиснення даних будь-яких типів без втрат базуються на виявленні та подальшому використанні тих чи інших закономірностей, що можуть існувати у рамках тих чи інших структурних одиниць потоку інформації.

Так, в умовах постійно зростаючої інтенсивності інформаційних потоків одним з актуальних є підхід, що базується на двох етапній обробці даних на рівні джерела. Загальну схему даного підходу демонструє рис. 2.1 [7-9].

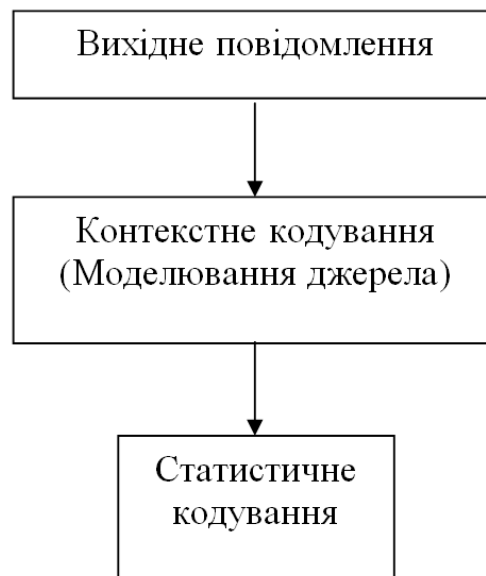


Рисунок 2.1 – Загальна схема двохетапного підходу до обробки даних на рівні джерела

Дана схема обробки ґрунтується на тому, що:

- різні ймовірності появи символів формують статистичні закономірності кодованих даних. Такі закономірності є властивими практично усім повідомленнями, інакше кажучи, такі закономірності можна назвати універсальними. Також слід зазначити, що між символами як на рівні

повідомлення, так і на рівні його частин (структурних одиниць) мають місце міжсимвольні зв'язки (двійкові послідовності, літери у словах). Це являє собою ще один тип універсальних закономірностей, у рамках якого ймовірність появи того чи іншого символу чуттєвою мірою визначається його контекстом, або, іншими словами - передісторією. У свою чергу, двох етапна схема обробки приймає до уваги обидва типи закономірностей – статистичні а також контекстні;

- у рамках поширених сьогодні технологіях кодування, що використовують двохетапну схему, виявлення зазначених закономірностей та подальша обробка даних на їх базі потребує виконання таких технологічних етапів, як: побудова контекстної моделі (інакше кажучи, кодування послідовностей символів, що є сусідніми з безпосередньо тим символом, ланцюжків сусідніх символів, для якого будується модель), після чого – безпосередньо кодування даних, отриманих за результатами виконання передуючого технологічного етапу;

- сутність побудови контекстної моделі зводиться до того, що відносно ряду тих чи інших характеристик початкових повідомлень виконується деяке їхнє узагальнення. Так, зокрема, якщо брати до прикладу текстову дані, то окремі слова чи словосполучення, що мають повтори, може бути подано у суттєво компактнішому вигляді, якщо зсилатися на них у передуючих сегментах тексту, уникаючи безпосереднього повторення. У даному випадку до складу цих посилань зазвичай входять такі дані, як номер першої літери даного слова у сегменті тексту або у повному тексті, довжина слова у літерах. Відповідно, при цьому відносно велика множини слів може бути замінена наведенням деяких числових характеристик, які формують посилання. Зрозуміло, що у такому випадку модель опису джерела даних суттєвою мірою буде спрощено та узагальнено. При цьому даний опис джерела з урахуванням узагальнення отримує назву ***інформаційної моделі джерела***. У зв'язку з цим технологічний етап побудови контекстної моделі може зватися ***моделюванням джерела***;

- зміни, що вносяться на етапі побудови інформаційної моделі джерела і носять узагальнюючий характер, сприяють збільшенню ефективності статистичних методів кодування даних. Так, у ряді випадків пряме використання даних методів має ряд обмежень застосовуваності та результативності. Причиною цьому є велика кількість різноманітних

вихідних алфавітів та статистичних моделей. Разом з тим, застосування контекстних моделей дозволяє застосовувати єдиний кодер статистичного типу для обробки джерел даних досить широкого діапазону класів. Як наслідок - двохетапний підхід до обробки даних на рівні джерела зараз набув досить широкого застосування.

2.2 Загальні підходи щодо побудови комплексних алгоритмів кодування

Серед застосовуваних сьогодні підходів до кодування даних існує велика різноманітність відносно використовуваних як алгоритмів побудови контекстних моделей, так і відносно побудови вже самих комплексних алгоритмів.

Разом з тим, уся існуюча сьогодні множина доступних до застосування алгоритмів формування контекстних моделей розподіляється на 2 загальні групи.

При цьому, першу групу складають загальні алгоритми а другу – т.з. спеціалізовані алгоритми [7, 9].

Відмінності між зазначеними групами алгоритмів полягають у тому, що алгоритми загальної групи є уніфікованими. Інакше кажучи, дані алгоритми може бути використано у ході обробки даних незалежно від їхнього типу та особливостей, забезпечуючи при цьому відносно високий рівень результативності.

Показниками результативності тут можуть бути [7-9]:

- коефіцієнт K_{comp} стиснення;
- середній час \bar{t} обробки структурної одиниці даних;
- раціональність побудови кодеру для забезпечення потрібного коефіцієнту K_{comp} стиснення за певних умов.

У свою чергу, спеціалізовані алгоритми спрямовані на кодування інформації деяких визначених типів (текст, зображення, аудіо дані тощо). Слід зазначити, що спеціалізовані алгоритми контекстного кодування ґрунтуються на урахуванні характерних особливостей та відмінностей тих типів даних, для обробки яких вони застосовуються.

Таким чином створюються умови для того, щоб могло бути досягнуто найбільшої ефективності їх використання.

2.2.1 Алгоритми контекстного кодування загальної групи

Алгоритми контекстного кодування загальної групи, а також їхнє місце у загальній ієрархії комплексних алгоритмів кодування ілюструється рисунком 2.2 [8, 10, 11].

Так, одними з найбільш простих у реалізації та з цієї причини широко застосовуваних зараз є методи кодування, в основі яких знаходиться алгоритм компресії серій однакових елементів. Даний алгоритм має назву RLE, або run-length encoding (кодування довжин серій). У рамках цього методу передбачається, що ланцюжкам однотипних елементів ділянки коду ставиться у відповідність пара елементів вигляду:

[символ; кількість повторень]

Перевагами даного алгоритму, як вже було попередньо зазначено, є простота його побудови. Утім, як самостійний метод кодування RLE має ряд обмежень, зокрема:

- відносно невисокий коефіцієнт K_{comp} стиснення;
- суттєва залежність коефіцієнту K_{comp} стиснення від особливостей змісту кодованих даних.

Наслідком цього є те, що RLE найбільш ефективним є для тих джерел повідомлень, та для даних тих типів, де ймовірність появи серій однакових елементів є відносно високою. Першочергово це справедливо для факсів, деяких графічних форматів (наприклад, PNG, PDF).

Разом з тим, алгоритм RLE набув широкого застосування як допоміжний технологічний етап каскаду перетворень у ході кодування зображень форматів JPEG та JPEG2000.

У свою чергу, до уніфікованих алгоритмів, тобто, таких, які може бути використано відносно даних будь-яких типів, широко використовуються зараз так звані словникові алгоритми. В основі роботи таких алгоритмів знаходиться співставлення послідовностей символів, що попередньо були виявлені у передуючій частині повідомлення, посиленнями на такі послідовності. Зараз використовується велика кількість підходів до пошуку та виявлення ланцюжків символів, що повторюються. При цьому, за певних

умов статистичні методи стиснення тут може бити застосовано у якості додаткового етапу кодування.

Окрім зазначених підходів, для побудови інформаційної моделі джерела зараз використовуються алгоритми прогнозування. У цьому випадку Тут контекстна модель являє собою множину послідовностей символів, з яких будується повідомлення, відносно кожної з яких зазначаються також відомості про частоту їх повторень. При цьому, один окремих символ також може прирівнюватися до елементарної послідовності. Алгоритми прогнозування забезпечують продуктивне статистичне кодування за рахунок побудови прогнозу вірогідностей появи тих чи інших елементів на кожній з позицій у рамках усього повідомлення або його якоїсь визначеної частини. У цьому випадку передбачається виконання постійного корегування утвореної таблиці кодових символів керуючись мінливим прогнозом. Алгоритми, що належать даному типові, мають загальну назву PPM, або Prediction by Partial Matching, тобто, прогнозування за частковим збігом.

Також до групи уніфікованих відноситься клас алгоритмів, які у своїй роботі застосовують операції сортування та/або сортування блочного типу, у рамках якого виконується обробка блоків даних.

В основі таких методів використовується ортогональне перетворення Барроуза та Уїллера (Burrows-wheeler transform, або BWT), у наслідок застосування якого виконується збір усієї множини однакових символів, які було виявлено у межах блоку на різних координатах, з наступним формуванням з них однорідних послідовностей. Після цього, відповідно, зазвичай застосовується або кодування RLE, або кодування одним зі статистичних методів. При цьому, ефективність використання статистичних методів може бути помітно збільшено шляхом використання перетворень «рух до початку», або MTF (move-to-front,) на передуючому безпосередньо самому кодуванню етапі обробки.

За результатами MTF-перетворення символам, частота появи яких у повідомленні є найбільшою, ставляться у відповідність найменші номери, при цьому інші символи ланцюжку, виключаючи перший, змінюються нульовими елементами.

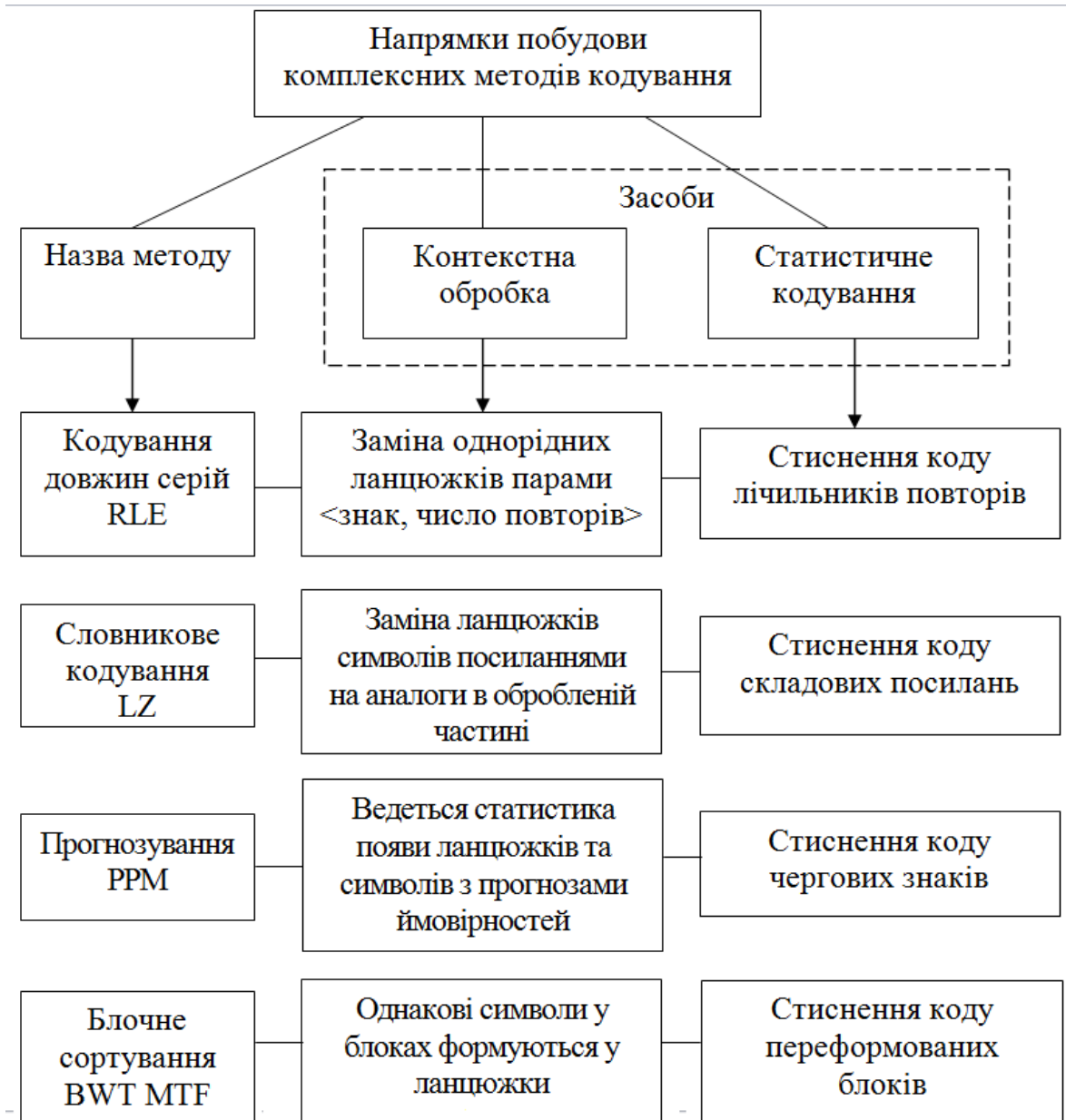


Рисунок 2.2 – Найбільш поширені алгоритми контекстного кодування загальної групи

2.2.2 Спеціалізовані алгоритми

Типовими алгоритмом даної групи є такі, що спрямовані на кодування зображень, що розглядаються як послідовності двовимірних масивів бінарних елементів. Найбільш наочно роботу даних алгоритмів може бути досліджено на прикладі кодування сегментів зображень за допомогою кодеку JPEG2000 [11, 12].

Так, у базисі JPEG2000 стиснення без втрат здійснюється на основі модифікованого арифметичного кодера (MQ-кодера). Тут відповідно до схеми кодування JPEG2000 передбачається, що відносно фрагментів зображення попередньо виконується такий перелік технологічних кроків, як:

1. Зміна формату опису сегментів вихідного зображення.

Цей технологічний етап передбачає що сегменти зображення, який формується двовимірний масивом $M \times N$ компонент у десятковому просторі опису, підлягають операції декомпозиції поелементно. У наслідок цього на базі одного вихідного масиву компонент $M \times N$ формується сукупність з β матриць такого ж розміру, які утворені двійковими елементами (рис.2.3). Зрозуміло, що β у даному випадку буде рівнятися розрядності компонент початкового масиву $m \times n$ компонент.

0	0	0	1
0	1	0	0
1	0	0	0
0	1	0	0

1	0	0	0
1	1	1	0
0	0	1	1
0	1	0	1

1	0	0	1
1	1	0	1
0	1	0	0
0	0	0	0

Рисунок 2.3 – Приклад послідовності бінарних матриць, отриманих у результаті виконання декомпозиції вихідного фрагменту

При цьому, процедура покомпонентної декомпозиції десяткових компонент C_{mn} , які формують вихідну матрицю $M \times N$, здійснюється відповідно до наступної формули:

$$C_{mn} = c_{mn}^{(\beta)} 2^{\beta-1} + c_{mn}^{(\beta-1)} 2^{\beta-2} + \dots \\ \dots + c_{mn}^{(\beta-\xi)} 2^{\beta-\xi-1} + \dots + c_{mn}^{(2)} 2 + c_{mn}^{(1)}, \quad (2.1)$$

де $c_{mn}^{(\beta)}$ - двійковий елемент (біт) y -го розряду, що локалізується на позиції m, n у рамках y -ї матриці.

2. Побудова контекстної моделі бінарних матриць.

У ході даного технологічного етапу виконується обробка кожної з β матриць двійкових елементів. При цьому така обробка передбачає виконання 3 проходів відносно матриць з $(\beta-1)$ -ї, по нульову.

У ході першого проходу, що має назву Significance propagation pass (розповсюдження значимості), підлягають обробці нульові двійкові елементи, які при цьому мають хоча б один сусідній значущий (не нульовий) біт.

У випадку обробки протягом даного проходу одиничного біта, компонента, якій він належить, маркується як значуща, а далі виконується кодування знаку компоненти.

У свою чергу, другий прохід - Magnitude refinement pass (уточнення амплітуди) передбачає обробку біт значущих компонент. При цьому визначення номер контексту здійснюється на базі сусідніх біт.

Протягом виконання третього проходу - Clean-up pass (прохід-очищення) здійснюється обробка не значимих компонент, для яких у ході першого проходу було отримано контекст рівний 0. Схема зчитування біт на третьому проході наводиться рис. 2.4. Тут застосовуються як контексти, що використовуються на першому проході так і два специфічних типи контекстів. У випадку, коли виконується обробка 4 біт у стовпцю, при цьому контекст кожного з них є нульовим, застосовується контекст "run-length" (RLE).

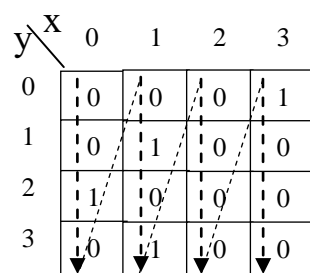


Рисунок 2.4 – Схема обходу біт двійкової матриці у ході виконання проходу clean-up pass

У свою чергу, обробка β -ї матриці, тобто, тієї, яка містить у собі біти старшого розряду, виконується за один прохід - Clean-up pass. Це зумовлено

тим, що дані відносно контексту початково невідомі, що не дозволяє у таких умовах визначити значущі компоненти.

Кодування кожного біту з усіх β двійкових матриць виконується виключно під час лише одного з проходів.

При цьому, у ході такої обробки для компоненти сегменту зображення, що кодується, розраховується показник *significance state*, або показник значущості (*SS*). Даний показник є бінарним, тобто, може дорівнювати 1 або 0. У випадку, коли $SS=1$, компонента маркується як значуща, відповідно при $SS=0$ компонента є незначима.

На першому кроці кодування уся множина компонент, що підлягають обробці, відносяться до множини незначущих. У свою чергу, після того, як у двійковому представленні компоненти виявлено одиничний біт, компонента надалі вважається значимою.

При цьому, контекст *CX*, що ставиться у відповідність кожній з компонент, розраховується на базі суми величин значущості сусідніх, тобто, 8 найближчих компонент, як зазначено на рисунку 2.5. Отже, беруться до уваги компоненти, що знаходяться поряд з поточною горизонтально, вертикально та діагонально ($H_0-H_1, V_0-V_3, D_0-D_3$).

D_0	V_0	D_1
H_0	X	H_1
D_2	V_1	D_3

Рисунок 2.5 - Розрахунок значення контексту для компоненти *X* на базі величин значущості сусідніх компонент

Слід зазначити, що величина *CX* визначається виключно поточною процедурою кодування та типом проходу, у ході якого дана процедура виконується, як зазначається табл. 2.1.

Так, як вже зазначалося раніше, початково компоненти вважаються незначущими [12].

При цьому, елементам першого стовпця (рис. 2.4) призначається контекст типу *RLC*. Отже, у відповідності з таблицею 2.1, $CX=17$, у свою

чергу, величина значущості $SS=1$, так як у межах даного стовпця виявлено одиничний біт. З огляду на це, далі обробці підлягають біти, які зазначають положення одиничного символу у стовпці.

Таблиця 2.1 – Перелік контекстів, що застосовуються у ході моделювання

Операція	Контекст, CX
Zero coding (ZC) (кодування нуля)	1
	2
	3
	4
	5
	6
	7
	8
Sign Coding (SC) (кодування знаку)	9
	10
	11
	12
	13
Magnitude Refinement Coding (MRC)	14
	15
	16
Run length Coding UNIFORM	17
	18

Далі застосовується контекст Uniform (для якого $CX=18$), при цьому показник значущості одиничного символу задається $SS=1$, а, відповідно, для наступного символу призначається $SS=0$, у свою чергу, значення контексту при цьому не змінюється.

Після цього, здійснюється операція кодування знака, або SC (sign coding), так як компонента, що знаходиться на позиції (2,0), переводиться до категорії значущих.

На той випадок, що при виконанні Clean-up pass обходу жодна з компонент не маркується як така, що є значущою, тоді одночасно з контекстом RLC на вхід MQ-кодеру надсилається нульовий символ. У подальшому, коли під час даного проходу виявляється хоча б один ненульовий біт, тобто, хоча б одна компонента переводиться до множини значущих, до MQ-кодеру надсилаються 3 біти, а саме:

- одиничний біт;
- 2 додаткові біти, які зазначають положення першої одиниці у компоненті.

У свою чергу, кодування зазначених додаткових біт здійснюється з використанням контекст Uniform. Далі, коли позицію першого бінарного елемента у компоненті визначено, інші елементи що залишилися у молодших розрядах цієї компоненти, кодуються таким же чином, як і під час першого проходу, задіюючи при цьому ті ж самі контексти.

У підсумку кодування β -ї матриці двійкових елементів (MSB-матриці) формується множина пар величин [CX, SS], які далі надсилаються до арифметичного кодера. При цьому, арифметичний кодер (MQ-кодер) забезпечує процес контекстно-залежного адаптивного арифметичного кодування.

Зазначений підхід пояснюється тим фактом, що під час процедури кодування значний обсяг біт після технологічного етапу квантування матиме нульові значення. Відповідно, у межах двійкових матриць, де локалізовано велику кількість нульових біт, об'єм даних буде незначним. У свою чергу, застосування контекстних моделей у базисі JPEG2000 дозволяє суттєво скоротити надлишковість кодового опису фрагментів зображення на технологічному етапі перед безпосередньо етапом стиснення з використанням MQ-кодеру.

2.3 Чинники, що можуть впливати на потенційну ефективності загальних комплексних алгоритмів кодування

Раніше зазначалося, що ефективність (результативність) тих чи інших алгоритмів кодування може бути оцінено за показниками коефіцієнта K_{comp} стиснення, середнього часу \bar{t} обробки структурної одиниці даних (швидкодії алгоритму) а також раціональності побудови кодеру для того, щоб можна

було досягти необхідного значення K_{comp} за час \bar{t} . При цьому, якщо трактування величини K_{comp} та \bar{t} у будь-якому випадку є однозначним, показник раціональності має бути уточнено. Наприклад, опосередковано свідчити про раціональність реалізації кодексу може об'єм пам'яті, який має виділятися у ході процедури стиснення. При цьому слід пам'ятати, що вимоги відносно швидкодії алгоритму та необхідного обсягу пам'яті здатні суттєвим чином відрізнитися для випадків кодування та декодування повідомлення або його частини [8, 10].

З урахуванням цього, результати порівняльного аналізу комплексних алгоритмів кодування залежно від тих чи інших показників будуть такими, як показано рис. 2.6

Також очевидним є те, що певні вимоги щодо показників, які характеризують ефективність комплексних методів кодування, за деяких умов можуть бути суперечливими.

Наприклад, нерідко максимальне значення K_{comp} може бути забезпечено шляхом збільшення складності та трудомісткості процесу кодування за умови виділення більшого об'єму пам'яті.

При цьому, допустимий баланс між показниками ефективності алгоритмів кодування може визначатися конкретної сферою їх використання. Так, для систем, де необхідно виконувати кодування у реальному масштабі часу, першочергово важливою є швидкість обробки.

У випадку ж, коли необхідно виконати архівацію даних для подальшого їх зберігання, ключовим показником є коефіцієнт стиснення.

На рис. 2.6 знаками «+» та «-» зазначено першочергову важливість того чи іншого показника для формування ефективності того чи іншого комплексного алгоритму кодування.

Таким чином, можемо бачити, що ефективність перелічених уніфікованих комплексних алгоритмів кодування може визначатися різною сукупністю показників.

Зокрема, алгоритмам словарного типу, що забезпечують найбільшу швидкість кодування за умови відносно високого коефіцієнту стиснення, властива найвища вимогливість до обсягу пам'яті, виділеної для їх функціонування.

Разом з тим, алгоритми, в основі яких знаходяться механізми прогнозування, на відміну від словарних алгоритмів характеризуються нижчою швидкістю. Такі алгоритми є невимогливими до обсягу доступної пам'яті, але при цьому здатні забезпечувати найвищі коефіцієнти стиснення для даних текстового типу, хоча тут не гарантується стиснення інформації інших типів на рівні з текстами.



Рисунок 2.6 – Порівняльний аналіз комплексних алгоритмів кодування залежно від тих чи інших показників

У свою чергу, позиція алгоритмів сортування, у сутності, є проміжною за зазначеними параметрами.

Також зазначимо, що продуктивність функціонування алгоритмів кодування може визначатися не лише базовими механізмами, але також особливостями їхньої програмної реалізації. Відповідно, у таких умовах один

і той же самий алгоритм може забезпечувати помітно різні показники ефективності.

2.4 Висновки за розглянутим матеріалом

Як свідчить аналіз розглянутих комплексних алгоритмів кодування, збільшенню ефективності функціонування кожного з них може сприяти або доопрацювання механізмів побудови інформаційної моделі джерела, або безпосередньо механізмів обробки отриманих контекстних моделей.

У першому випадку забезпечується скорочення надмірності представлення даних, які генерує джерело, відповідно, за рахунок цього забезпечується потенційна можливість досягти вищих коефіцієнтів стиснення.

У свою чергу, удосконалення механізмів кодування також дає змогу щодо збільшення величини K_{comp} , але вже шляхом більш продуктивного усунення статистичної, психовізуальної та семантичної надмірностей даних.

Разом з тим, такий підхід є, у сутності, екстенсивним, що пояснюється наступним:

- комплексні алгоритми обробки, які мають у своєму складі 2 ключових технологічних етапи, є потенційно ресурсоемкими вже на рівні архітектури. Відповідно, у результаті удосконалення механізмів обробки та включення додаткових технологічних етапів загальний алгоритм ускладнюється;

- ускладнення алгоритму обробки вимагає додаткових апаратних ресурсів. Це є критичним на випадки, коли кодування даних повинно виконуватися у реальному часі. Відповідно, у таких умовах може суттєво звужуватися сфера застосування даних алгоритмів, так як повноцінна їх імплементація буде можливою виключно на високопотужних пристроях.

Таким чином, для випадків, коли час обробки даних є критичним, необхідно забезпечити удосконалення існуючих алгоритмів кодування таким чином, щоб одночасно з підвищенням коефіцієнту стиснення підтримувалася можливість компресії даних з внесенням мінімально можливої затримки. Ймовірним напрямком тут може вважатися скорочення надмірностей нових типів, які в існуючому базисі технологій не розглядаються.

3. МЕТОДИ КОДУВАННЯ БЕЗ ВТРАТ, ЩО ВХОДЯТЬ ДО СКЛАДУ СУЧАСНИХ ТЕХНОЛОГІЙ ОБРОБКИ МУЛЬТИМЕДІЙНИХ ДАНИХ

3.1 Загальні відомості про технологічні етапи кодування у складі технологій компресії мультимедіа

На сьогодні, беручи до уваги поширеність у мережі графічної і відеоінформації, актуальним є побудова методів кодування, орієнтованих на обробку саме цих типів даних.

При цьому, схема кодування графічних об'єктів та відео, як масиву окремих кадрів, є майже повністю аналогічною. Тут передбачаються технологічні етапи, на яких вносяться незворотні зміни у кодовані фрагменти Ω_i зображень (кодування з втратами), та етапи, на яких забезпечується зменшення кількості біт з можливістю їх подальшого повного відновлення (кодування без втрат). Так як зараз MPEG-орієнтовані технології є домінуючими серед усіх існуючих методів кодування відео, подальший розгляд процесу обробки графічних та відеоданих буде виконуватися на прикладі саме цього технологічного базису.

Зокрема, незворотні зміни при кодуванні на таких технологічних етапах, як [10,13, 14]:

- зміщення колірному простору опису фрагментів Ω_i зображень (кадрів) F ;
- ортогональне перетворення з переведенням фрагментів Ω_i з просторового формату представлення у спектральну площину;
- квантизація компонентої форми представлення фрагментів Ω_i з подальшим округленням компонент $C(i)_{x,y}$, величина яких є нижчою, ніж деякий попередньо заданий рівень $\hat{C}(i)_{x,y}$.

У свою чергу, до етапів кодування без втрат відносяться:

- RLE-кодування;
- ентропійне кодування.

У базисі технології JPEG, за якою виконується обробка зображень та майже повністю, з невеликими модифікаціями – обробка фрагментів відеокадрів, ентропійне стиснення реалізовано на базі метода Хафмана та на базі арифметичного кодування.

Далі виконаємо стислий аналіз механізмів кодування без втрат на прикладі методу Хафмана.

3.2 Аналіз функціонування процесу стиснення без втрат на базі методу Хафмана

3.2.1 Загальна схема технологічного процесу кодування без втрат під час обробки зображень

Безпосередньо процес кодування за Хафманом, як фінальний технологічний етап обробки у базисі JPEG як для статичних зображень, так і для відеокадрів, попередньо потребує виконання підготовчих кроків [14].

Так, на першому з них відносно двовимірного масиву $\{C\}$ компонент $C(i)_{x,y}$, спектрального фрагменту Ω'_i , які попередньо підлягали квантуванню, виконується операція лінеаризації, у ході якої масив $\{C\}$ перетворюється на одномірний вектор. Це здійснюється у ході процесу зигзаг-скану, як показано рис. 3.1

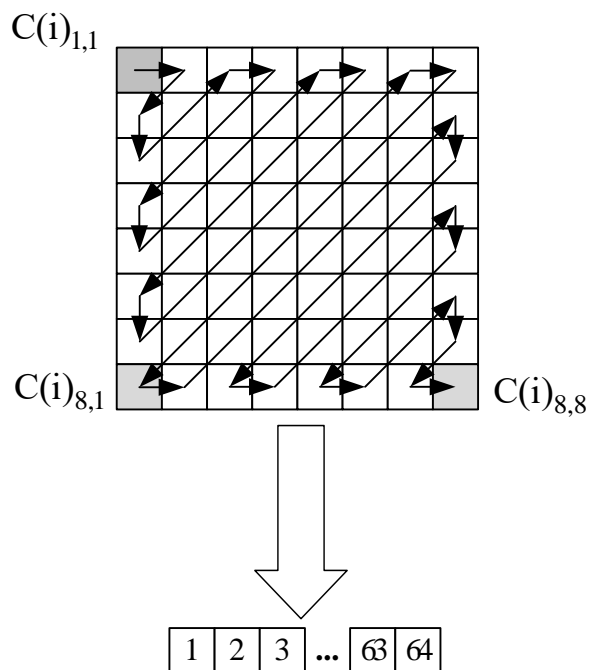


Рисунок 3.1 - Процес лінеаризації компонент з використанням процедури зигзаг-сканування

Далі отриманий таким чином 64-елементний вектор обробляється за алгоритмом RLE. Даний технологічний субетап, у сутності, реалізує процес контекстного моделювання, тобто, є підготовчим кроком відносно безпосереднього кодування. На цьому субетапі на базі лінеаризованого вектору $\vec{C}(i)_j$ десяткових значень компонент $C(i)_{x,y}$ утворюється двокомпонентний кортеж $C(i)_j$ виду [кількість повторень; символ], як зазначалося у розділі 3. При цьому, поле «кількість повторень» являє собою лічильником нульових елементів. У свою чергу, поле «число» зазначає величину, яку буде вписано до наступної чарунки.

Наприклад, для наступного фрагменту вектору $\vec{C}(i)_j$, а саме:

57 3 0 0 0 0 0 0 -12 0 0 0 0 0 1 0 0 5

у наслідок виконання RLE буде сформовано кортеж $C(i)_j$ вигляду:

(0,57) (0,3) (6,-12) (5,1) (2,5).

Далі утворена таким чином послідовність надходить до ентропійного кодеру.

Отже, етап кодування без втрат умовно містить у собі 3 субетапи – лінеаризацію компонент, RLE та безпосередньо кодування.

Таким чином, розглянутий процес кодування є потенційно складним, що створює умови, у яких обробка відеокадрів або статичних зображень у реальному часі не гарантується.

Далі виконаємо аналіз самого методу Хафмана.

3.3 Аналіз функціонування методу Хафмана

Виходячи з особливостей змісту фрагментів зображень, які підлягають кодуванню, метод Хафмана дозволяє скоротити кількість початкових біт, які надходять від 1/5 до майже 9/10 відносно вихідного обсягу. При цьому, статистично прийнятий усереднений коефіцієнт K_{comp} стиснення для методу Хафмана – 2, тобто, вважається, що скорочується 1/2 від початкового об'єму біт [14].

Основу алгоритму Хафманом складає алгоритм опису символів повідомлення залежно від частоти їх появи. Тобто, ті символи, що зустрічаються у межах повідомлення найбільш часто, відповідно мають найвищу ймовірність появи, описуються бінарними конструкціями мінімальної довжини.

У свою чергу, символам, поява яких є менш вірогідною, призначаються конструкції дещо більшої довжини.

Таким чином, змінюється початковий алфавіт, тобто, вихідна множина символів, за допомогою якої описується повідомлення. Дана операція здійснюється у такий спосіб, щоб забезпечити щонайменше значення середньої довжини коду на один символ початкового повідомлення.

При цьому перший етап кодування передбачає розрахунок ймовірності γ появи кожного символу, на базі яких виконується формування повідомлення.

Далі, ґрунтуючись на визначених величинах ймовірностей γ , виконується створення кодового дерева Хафмана, за допомогою якого на подальших кроках кодуються символи повідомлення.

Отже, умовно алгоритму складається з 2 субетапів, а саме:

- розрахунок та формування таблиці ймовірностей γ появи символів у повідомленні;
- створення кодового дерева за результатами розрахованих ймовірностей.

3.3.1 Процес кодування даних на базі методу Хафмана

Припустимо, що у межах довільного повідомлення γ відомі усі ймовірності появи символів. Відповідно, у цьому випадку процес кодування буде полягати виключно у формуванні кодового дерева.

Далі, для більш наочного розгляду процесу кодування за Хафманом, кодове дерево буде створюватися у графічний спосіб.

При цьому, процес кодування використовує специфічну термінологію, яку необхідно розглянути детально.

Так, наприклад, якщо *граф* у рамках методу Хафмана розглядається як множина сукупностей дуг та вузлів, які є спрямованими від одного вузла до

іншого, то *дерево*, як частковий випадок графа, має свою специфіку. Зокрема, дерево має відповідати таким вимогам, як [14]:

- входження в один вузол лише єдиної дуги;
- у межах дерева може міститися лише єдиний вузол, у який не входить жодної дуги (корінь дерева);
- у результаті переміщень дугами дерева від його кореня може бути досягнуто будь-якого вузла.

У свою чергу, *лист дерева* являє собою вузол, з якого дуги не виходять взагалі. При цьому, з двох вузлів дерева, що поєднуються дугою, вузол, що її породжує, має назву *батьківського*, тоді як інший вузол зветься *нащадком*.

Також, за умови, що один батьківський вузол є спільним для двох вузлів, тоді такі вузли зветься *братами*.

У підсумку, саме *бінарне дерево* у даному випадку це таке дерево, всі вузли якого, за винятком листів, породжують по дві дуги.

Врешті решт, *дерево кодування Хафмана* таке бінарне дерево, що формується вузлами, кожен з яких має вагу. Важливою умовою тут є те, що значення ваги батьківського вузла рівна сумі ваг його нащадків.

3.3.2 Формування бінарного дерева Хафмана

У загальному випадку у ході побудови бінарного кодового дерева передбачається виконання ряду кроків, а саме:

1. Побудова множини вільних вузлів бінарного дерева на базі символів початкового алфавіту. Тут вага кожного вузла дорівнює величині γ ймовірності появи символу у фрагменті повідомлення чи самому повідомленні (дивлячись на те, на рівні яких саме структурних одиниць повідомлення виконується кодування. Так, у нашому випадку, для JPEG це - двокомпонентний кортеж $S(i)_j$, отриманий у наслідок виконання RLE).

2. Вибір пари вільних вузлів, що мають мінімальні величини ваг. На той випадок, що у межах дерева існує більше, ніж два вільних вузла, яким відповідають мінімальні ваги, може бути задіяно будь-яку пару з них.

3. Обирається батьківський вузол для вузлів, які було обрано на передуючому кроці.

4. Батьківський вузол вноситься до переліку вільних вузлів, тоді як двоє його нащадків з даного переліку виключаються.

5. Дугам, які породжуються батьківським вузлом, призначаються бінарні елементи. Одній дузі – одиничний а іншій, відповідно, нульовий.

6. Виконується повторення кроків з 2 по 5. Таке повторення здійснюється доти, поки перелік вільних вузлів буде містити лише один вузол. Даний вузол, при цьому, являтиме собою корінь бінарного дерева. Відповідно, величина його ваги є сумою ймовірності появи усіх символів, що формують кодоване повідомлення, тобто, є рівною одиниці.

Далі, рухаючись утвореним бінарним деревом зверху від кореня бінарного дерева, при цьому послідовно фіксуючи двійкові символи, які ставляться у відповідність дугам, отримуються кодові значення підсумкового алфавіту.

3.3.3 Приклад кодування за Хафманом з графічною інтерпретацією процесу кодоутворення

Важливо зауважити, що у рамках методу Хафмана символами повідомлення можуть розглядатися послідовності двійкових елементів.

Нехай маємо вихідне повідомлення у вигляді двокомпонентного кортежу $S(i)_j$. Для випадку даного повідомлення алфавіт містить у собі 8 символів (послідовностей бінарних елементів), розподіл ймовірностей появи яких ілюструє табл. 3.1. Тут символи s_1-s_8 відповідають 8 двійковим послідовностям.

Таблиця 3.1 – Ймовірності появи символів у межах алфавіту початкового повідомлення

Символ	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
Імовірність	0.23	0.19	0.16	0.16	0.10	0.10	0.05	0.01

Формування бінарного дерева потребує створення переліку окремих листів, як показано рисунком 3.2, що здійснюється покроково.

0,23	0,19	0,16	0,16	0,10	0,10	0,05	0,01
s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8

Рисунок 3.2 - Масив вільних вузлів-листів

Далі з множини листів бінарного дерева необхідно вибрати два листи, яким відповідають мінімальні величини ваг. У нашому випадку, це листи s_7 та s_8 . Після цього обрані листи s_7 та s_8 приєднуються до батьківського вузла. Отже, у результаті виконання даної операції, вага батьківського вузла буде дорівнювати $0,05+0,01=0,06$.

На наступному кроці вузли s_7 та s_8 буде видалено з переліку вільних. При цьому, вузол s_7 є відповідним нульовій дузі батьківського вузла, а вузол s_8 – одиничній дузі.

У результаті завершення першого кроку роботи алгоритму, бінарне дерево буде таким, як ілюструє рисунок 3.3.

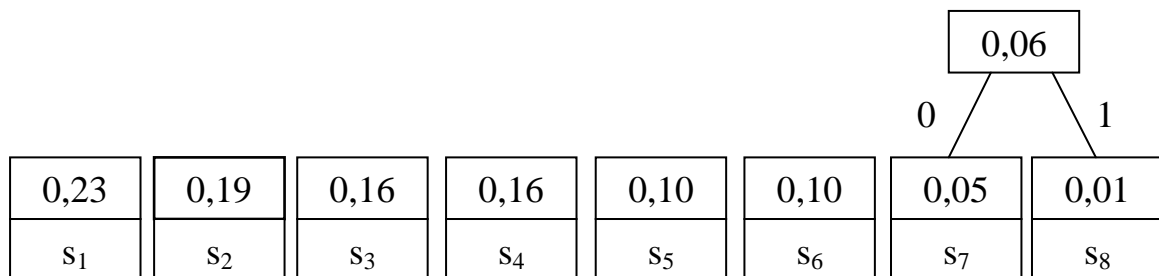


Рисунок 3.3 – Бінарне дерево Хафмана, утворене після виконання першого кроку обробки

У ході другого кроку обробки при визначенні листів, яким відповідають найменші значення ваг, обираються лист s_6 а також отриманий на попередньому кроці вузол (s_7+s_8) . Для даних обраних листів створюється батьківський вузол, якому, у свою чергу, відповідатиме значення ваги, рівне 0,16.

Після цього нульовій дузі батьківського вузла ставиться у відповідність вузол s_6 відповідає, а одиничній дузі - вузол (s_7+s_8) .

При цьому, бінарне дерево Хафмана на поточному кроці кодування буде таким, як показано рис. 3.4.

У свою чергу, протягом наступного кроку обираються символи s_3 , s_4 , s_5 а також вільний вузол $(s_6+s_7+s_8)$, як такі, яким відповідають величини мінімальних ймовірностей на поточному кроці.

Отже, у ході поточного кроку може бути створено батьківський вузол для листів s_5 та $(s_6+s_7+s_8)$, підсумкова вага якого рівнятиметься 0,26.

Для цього випадку бінарне дерево Хафмана буде таким, як ілюструє рисунок 3.5. Тут існує ряд варіантів поєднання вузлів, яким відповідають мінімальні величини ваг.

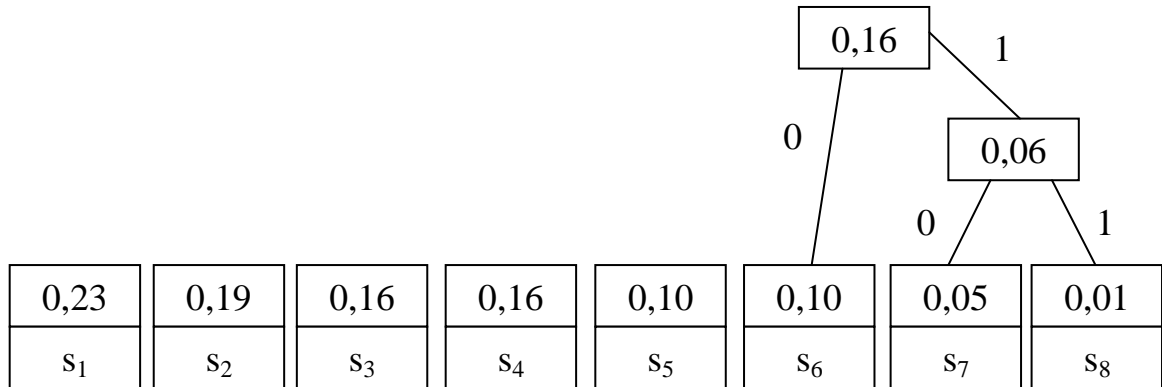


Рисунок 3.4 – Бінарне дерево Хафмана, отримане у результаті виконання другого кроку формування коду

Разом з тим, зазначимо, що увесь перелік таких варіантів буде цілком коректним, не дивлячись на те, що при цьому може бути утворено кодограми, що будуть відрізнятися між собою. Це пояснюється тим, що кожна з таких кодограм буде однаково ефективною в умовах наявного розподілу ймовірностей появи символів.

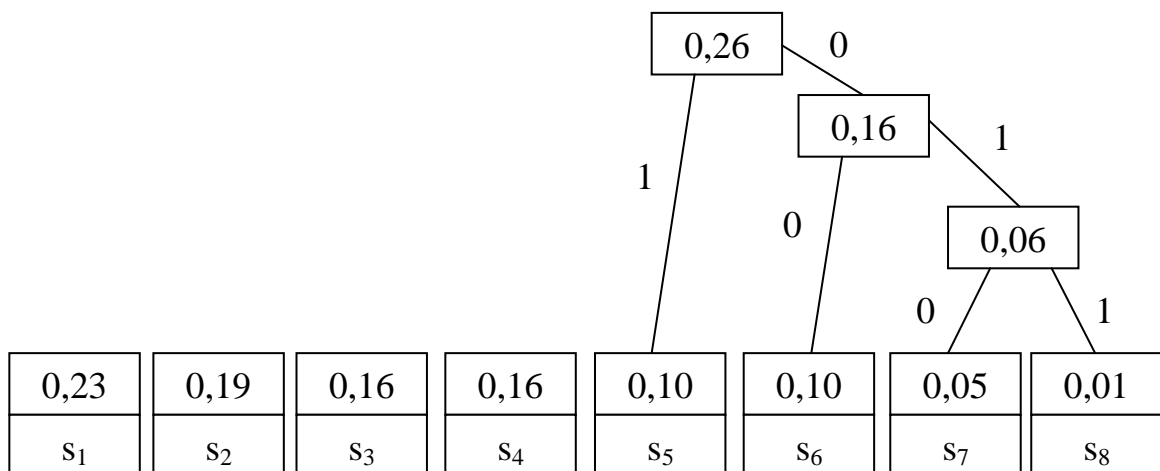


Рисунок 3.5 - Бінарне дерево Хафмана, сформоване у ході поточного кроку кодування

Дані, на послідуєчих кроках кодування, зокрема, на четвертому, двома листами, яким відповідають мінімальні ваги ϵs_3 та s_4 .

Бінарне дерево, що відповідає цьому випадку, ілюструється рис. 3.6.

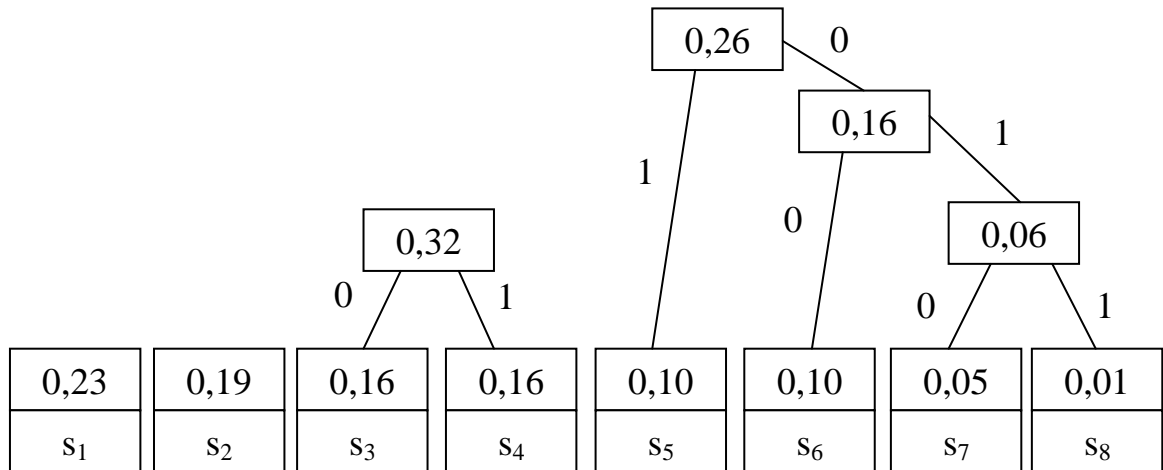


Рисунок 3.6 – Бінарне дерево Хафмана, сформоване у результаті виконання четвертого кроку кодування

Відповідно, для наступного кроку кодоутворення, мінімальні ваги відповідають вузлам s_1 та s_2 .

Тут вага s_1 рівна 0,23 а s_2 – 0,19. У свою чергу, це відповідним чином позначиться на структурі бінарного дерева, яке для цього кроку зображується рис.3.7:

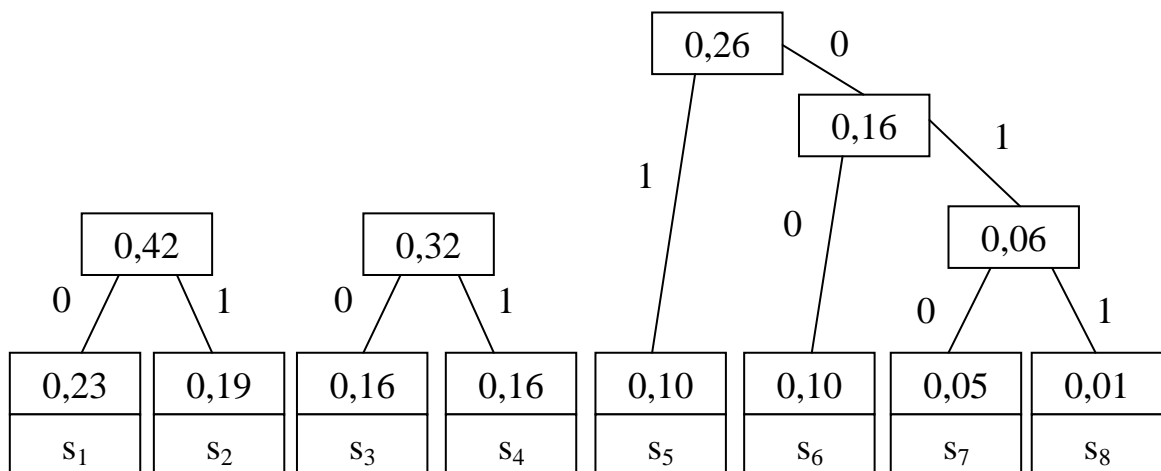


Рисунок 3.7 – Структура кодового дерева, отримана у підсумку кодування вузлів s_1 і s_2

У результаті виконання означених вище дій вільними лишаються лише вузли, яким відповідають значення ваг 0,42, 0,32 та 0,26 відповідно.

Дані ваги попередньо було утворено, як результати складення вузлів s_1 та s_2 , s_3 та s_4 і s_5 , s_6 , s_7 та s_8 .

Далі парою вузлів з мінімальними величинами ваг будуть ті яким відповідають значення 0,32 і 0,26.

Відповідно, це приведе до змін у структурі бінарного дерева, точніше, сприятиме його доповненню.

На цьому кроці структура буде такою, як показано рис.3.8):

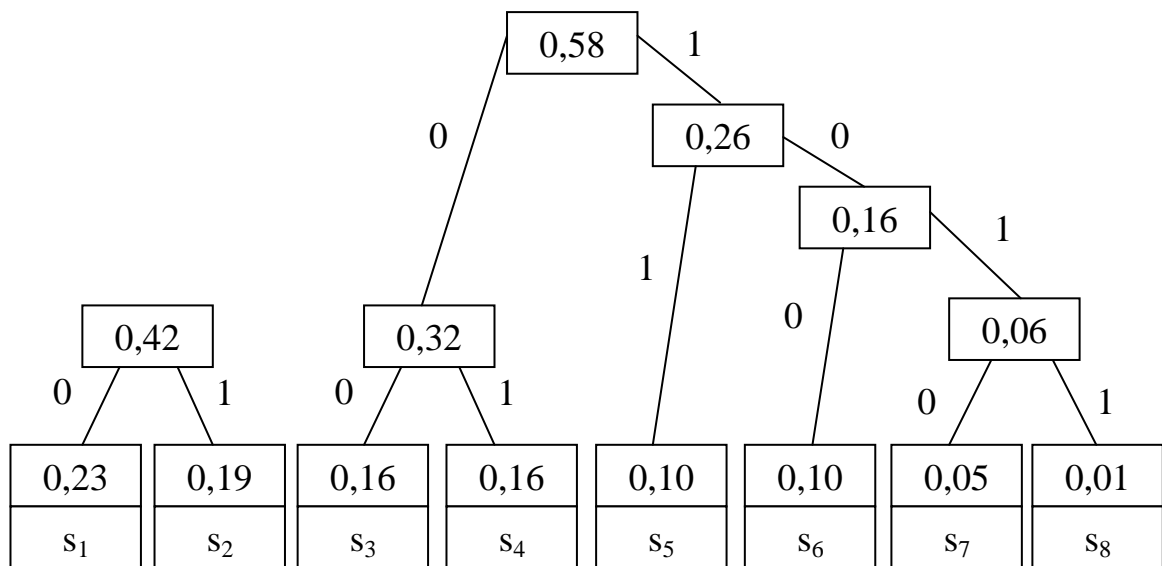


Рисунок 3.8 – Структура бінарного кодового дерева на поточному кроці кодування (оброблено вузли, яким відповідають величини ваг 0,32 та 0,26)

Врешті решт бачимо, що у структурі бінарного дерева залишилося лише дві вільні вершини, яка на поточному кроці поєднуються.

У результаті цього бінарне дерево Хафмана можемо вважати повністю сформованим.

Його структура у цьому випадку показана рис. 3.9.

На базі побудованого дерева, кожен із вихідних символів повідомлення буде представлено кодом, що відображає шлях від кореневого вузла до листа, відповідного потрібному символі.

У результаті, символи початкового алфавіту будуть кодуватися так, як показано табл. 3.2.

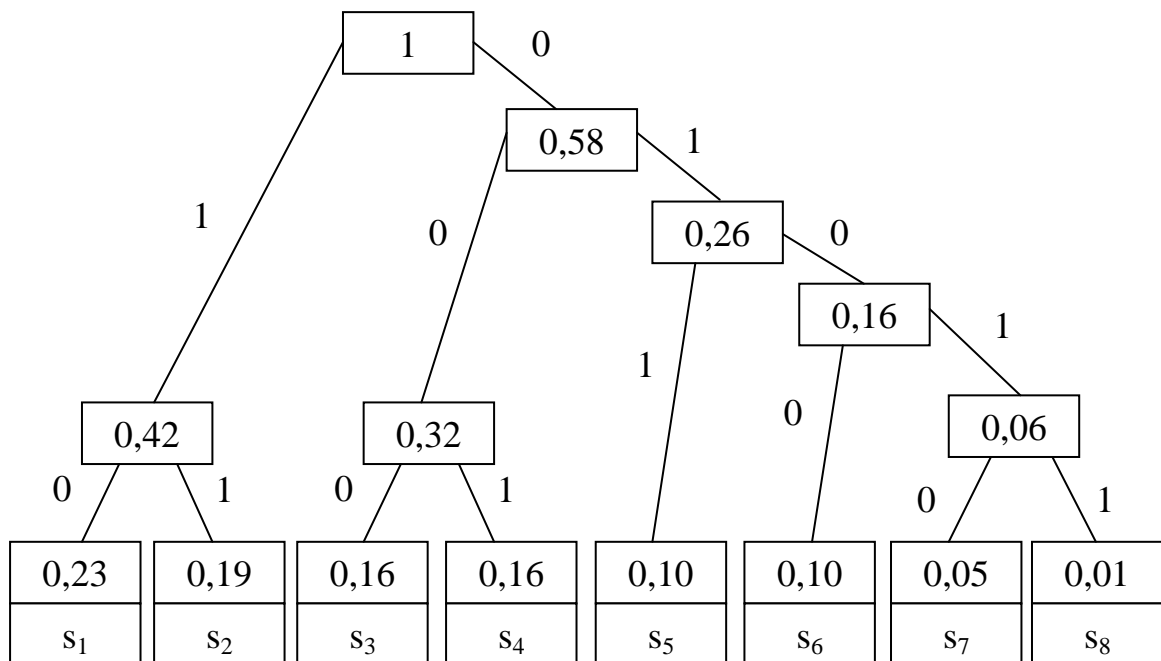


Рисунок 3.9 – Підсумкова структура бінарного дерева Хафмана

Таблиця 3.2 – Отриманий кодовий опис вихідних символів алфавіту повідомлення, сформований з використанням алгоритму Хафмана

Символ	s ₁	s ₂	s ₃	s ₄	s ₅	s ₆	s ₇	s ₈
Код	10	11	000	001	011	0100	01010	01011

Як вже зазначалося раніше, на базі ймовірностей появи символів, наведених табл. 3.1, можливо побудувати альтернативні структури бінарного кодового дерева.

Приклад одного з можливих варіантів побудови альтернативного дерева показує рис.3.10.

Відповідно, сформований алфавіт, який відповідає цьому варіанту бінарного дерева, міститься у табл. 3.3.

Розгляд таблиці 3.3 дозволяє побачити, що у даному випадку, коли побудовано альтернативне бінарне кодове дерево Хафмана, сформовані коди є префіксними, так само, як і у попередньому випадку, що зображено рис.3.9

та табл. 3.2. Але також символам повідомлення, ймовірність появи яких є найвищими, ставляться у відповідність коди найменшої довжини.

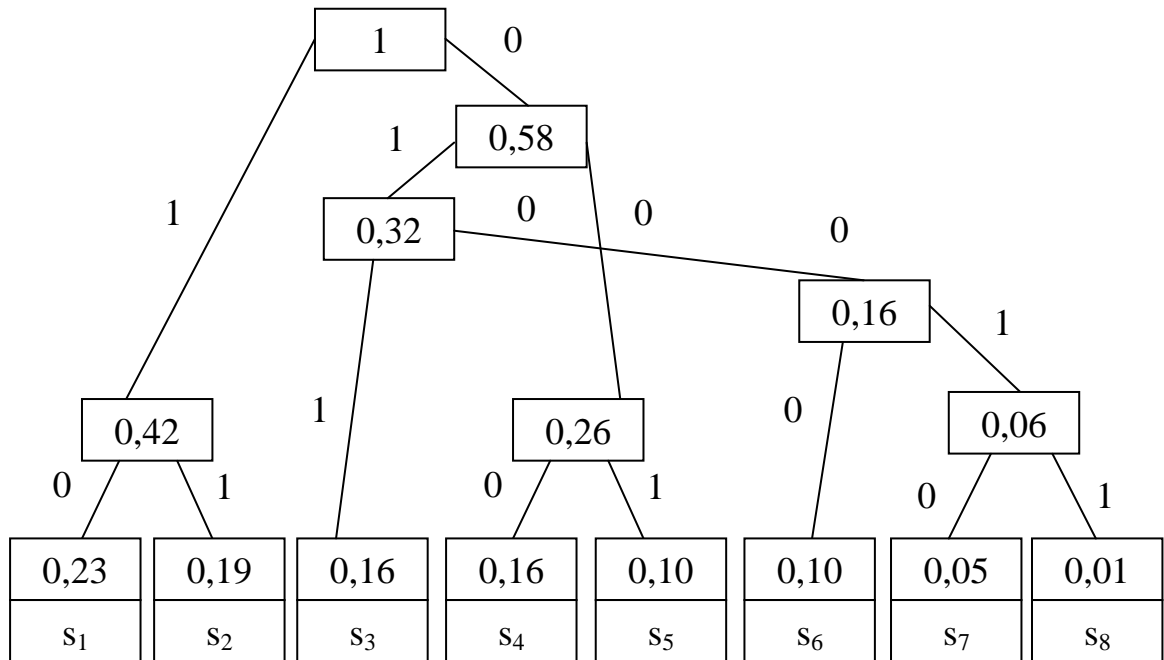


Рисунок 3.10 – Структура бінарного кодового дерева в альтернативному варіанті формування

Таблиця 3.3 – Сформований алфавіт повідомлення, який відповідає альтернативному варіанту бінарного кодового дерева

Символ	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈
Код	11	10	011	001	000	0101	01001	01000

3.4 Аналіз ефективності процесу та результатів кодоутворення на базі методу Хафмана

Для виконання аналізу пропонується використати наступний перелік показників [13, 15]:

- значення коефіцієнту K_{comp} стиснення;
- величина середньої довжини ℓ_{cp} коду. Цей параметр є одним з ключових для кодів змінної довжини. Середня довжина коду характеризує

середню кількість біт, яка потребується для представлення символу утвореного алфавіту повідомлення;

- значення δ дисперсії. Даний показник свідчить про те, який великим є відхилення довжин, що відповідають індивідуальним кодовим конструкціям (що описують кожен окремий символ у повідомленні) відносно середньої довжини кодограми. Вважається, що ефективна підсумкова кодограма повідомлення повинна мати дисперсію мінімально можливу за даних умов.

Так, загальний обсяг біт, що необхідно використати для опису вихідного повідомлення довільного змісту розраховується відповідно до виразу:

$$B = n \times \eta, \quad (3.1)$$

де n - кількість символів у повідомленні;

η - необхідний обсяг біт для представлення одного символу.

Вираз (1.2) є також справедливим для даних, які попередньо було кодовано.

Відповідно, якщо у нашому випадку опис символів повідомлення потребує 8 розрядів, тобто використовується біт/символ, тоді загальний обсяг біт для представлення повідомлення буде рівнятися $8 \times 25 = 192$ біт.

При цьому, як свідчать дані табл. 3.2, представлення кодограми, отриманої на базі методу Хафмана, потребує використання 27 бінарних символів. Отже, на цей випадок коефіцієнт K_{comp} стиснення буде

$$\text{дорівнювати } K_{\text{comp}} = \frac{192}{27} = 7,1.$$

У свою чергу, розрахунок значення l_{cp} середньої довжина утвореної кодової конструкції виконується на базі наступного виразу:

$$l_{\text{cp}} = \sum_i p_i \times l_i, \quad (3.2)$$

де p_i - ймовірність появи символу;

l_i - довжина одного символу повідомлення.

У нашому випадку на базі виразу (3.2) маємо:

$$\begin{aligned} l_{cp} = & 0,23 \times 2 + 0,19 \times 2 + 0,16 \times 3 + 0,16 \times 3 + \\ & + 0,1 \times 3 + 0,1 \times 4 + 0,05 \times 5 + 0,01 \times 5 = 4,65 \text{ біт/символ} \end{aligned}$$

Стосовно величини коефіцієнту дисперсії коду зазначимо, що на випадок виключно запису кодової послідовності символів на носій, даний параметр, взагалі не має значення.

Мінімізація дисперсії для кодових конструкцій, утворених на базі методу Хаффмана, є актуальною тоді, коли передбачається передавання кодової послідовності у мережу.

Така закономірність пояснюється тим, що у випадку кодів з великою дисперсією, кодоутворення буде мати місце у VBR-режимі, тобто, здійснюватиметься зі змінною бітовою швидкістю. При цьому, оскільки за замовчуванням трансляція даних відбувається каналами, що функцінують у CBR-режимі, тобто, мають постійну бітову швидкість, у такому випадку необхідно задіювати буфер. Відповідно, біти утвореного коду надсилаються кодером до буферу у міру їх генерації, звідки далі надсилаються у канал, що забезпечує постійну бітову швидкість для передачі.

При цьому очевидно, що передавання у буфер кодограм, яким відповідає рівень дисперсії, близький до нульового або рівний нулю, буде здійснюватися з постійною бітовою швидкістю. Отже, у цьому випадку достатнім для використання буде буфер невеликого об'єму. У свою чергу, з ростом рівня дисперсії коду буде також збільшуватися необхідний обсяг буферу.

Так як на етапі побудови бінарного дерева Хаффмана попередньо було утворено декілька його варіантів, визначити ефективний варіант кодограми з –поміж ряду існуючих можливо на базі розрахунку величин дисперсії для кожної з них.

Так, величина дисперсії може бути розрахована на базі наступного виразу:

$$\delta = \sum_i p_i \times (l_i - l_{cp})^2 \quad (3.3)$$

Тоді за виразом (3.3) для першого утвореного бінарного маємо наступне:

$$\delta = 0,23 \times (2 - 4,65)^2 + 0,19 \times (2 - 4,65)^2 + 0,16 \times (3 - 4,65)^2 + 0,16 \times (3 - 4,65)^2 + 0,1 \times (3 - 4,65)^2 + 0,1 \times (4 - 4,65)^2 + 0,05 \times (5 - 4,65)^2 + 0,01 \times (5 - 4,65)^2 = 4,143$$

У нашому випадку величина δ дисперсії для кожного з утворених бінарних дерев Хафмана буде однаковою. Звідси виходить, що кожний з варіантів кодоутворення може розцінюватися, як ефективний.

3.4.1 Інтерпретація даних, отриманих у ході дослідження ефективності методу Хафмана

1. Як видно з аналізу процедури кодування за методом Хафмана, отримано досить високе значення коефіцієнту K_{comp} стиснення. Разом з тим, очевидно, що зростання значення K_{comp} на пряму залежить від зміни величини ймовірностей появи символів. Стосовно ж розглянутого прикладу маємо відносно невелику довжину послідовності, що підлягає кодуванню, символам якої відповідають ймовірності досить великих номіналів. Тобто, у випадку, коли зменшуються ймовірності появи символів повідомлення, результативність досліджуваного методу буде суттєво зменшуватися також. У свою чергу, для двокомпонентних кортежів $C(i)_j$, утворених на базі компонент сегментів Ω'_i спектрального представлення реальних зображень, статистично доведено, по-перше, нестабільність розподілу ймовірностей появи символів а по-друге – значно нижчі величини ймовірностей появи символів. Це у підсумку у середньому дозволяє стверджувати, що метод Хафмана забезпечує приблизно двократне скорочення обсягу вихідного повідомлення на той випадок, коли кодуються графічні дані.

2. Кодування на базі алгоритму Хафмана є неефективним для кодування безпосередньо бінарного алфавіту, тобто, алфавіту, що складається з двох символів – 1 та 0. Це зумовлено тим, що при кодуванні за методом Хафмана символ вихідного повідомлення не може бути описано кодограмою, що має довжину меншу, ніж один біт.

3. Кодування за методом Хафмана у складі технологій обробки відео та зображень не є самостійним алгоритмом та додатково потребує виконання технологічних етапів лінеаризації та RLE (побудова контекстної моделі).

3.5 Висновки за розділом

У результаті виконаного аналізу одержаних результатів можемо зазначити наступне:

- символам повідомлення, яким відповідають максимальні величини ймовірностей їх знаходження у кодованій послідовності, ставляться у відповідність кодограми мінімальної довжини. У свою чергу, символи, ймовірність появи яких у повідомленні невисока, кодуються двійковими конструкціями більшого розміру. Таким чином забезпечується зменшення початкового обсягу біт для опису кодованого повідомлення;

- побудова кодограм символів повідомлення здійснюється так, що при цьому уникаються випадки збігів кодової конструкції з початком іншої кодової конструкції, яка є довшою. У таких умовах, забезпечується можливість декодування стисненого повідомлення без необхідності застосування додаткових службових даних;

- для реальних зображень не гарантується хоча б відносно високих стабільних величин ймовірностей появи символів у кодованій послідовності, відповідно, не гарантується суттєве збільшення величини коефіцієнта стиснення вище статистично відомого значення;

- на базі методу Хафмана обробка даних, поданих у вигляді бінарного алфавіту, є неефективною;

- технологічний етап кодування без втрат у базисі JPEG на основі алгоритму Хафмана є достатньо ресурсоємним, як наслідок того, що окрім безпосереднього кодування включає у себе субетапи лінеаризації та контекстного моделювання на базі RLE.

4. ДОСЛІДЖЕННЯ ПІДХОДУ ДО КОДУВАННЯ БЕЗ ВТРАТ НА БАЗІ КОДІВ З ПЛАВАЮЧИМИ ВАГАМИ

4.1 Вимоги до методу кодування без втрат, спрямованого на обробку графічних даних

Виходячи з виявлених недоліків кодування без втрат в існуючому базисі обробки кадрів за ідеологією JPEG, а також керуючись тим, що алгоритм кодування без втрат у складі теології ущільнення відео має забезпечувати збільшення коефіцієнту стиснення, при цьому не створюючи суттєвого обчислювального навантаження на систему, пропонується до розгляду підхід до ущільнення відеокадрів на базі компактного представлення їх компонентної форми опису у біноміально-поліадичному просторі.

Згідно до існуючих статистичних даних, усереднена величина коефіцієнту K_{comp} стиснення для даного методу становить 2,5-2,7, це щонайменше на 20% вище, ніж для статистичних методів, які зараз є традиційними для JPEG [16].

4.2 Загальна концепція побудови методу кодування без втрат сегментів зображень у біноміально-поліадичному просторі

У загальному випадку кодування окремих кадрів F відеопотоку у рамках ідеології JPEG передбачає обробку сегментів Ω'_i у спектральному представленні, які за замовчуванням мають розмір 8×8 .

Даний підхід є справедливим для безпосередньо кодування зображень на засадах JPEG та відеокадрів з використанням базису MPEG до рекомендації H.264 включно. При цьому, починаючи з рекомендації H.265, розмір сегменту Ω'_i у деяких випадках може змінюватися до 4×4 та 2×2 .

Виходячи з цього, далі більш раціональним є встановити розмір для сегменту рівним $K \times L$, де $K = L = \{2; 4; 8\}$ [17].

4.2.1 Формування компактного представлення сегменту зображення

Розглянемо варіант побудови кодового представлення сегментів Ω'_i зображення у біноміально-поліадичному просторі опису.

Для даного варіанту формування коду $E(\Omega'_i)$ сегментів Ω'_i на обробку надходить кожен сегмент окремо.

При цьому, у структура кодової конструкції $E(\Omega'_i)$ сегменту має у своєму складі n поліадичних Θ_k чисел з плаваючою вагою [16].

Кожне таке число Θ_j , у свою чергу, може бути побудовано на базі окремого рядку або стовпця сегменту Ω'_i зображення F .

Таким чином, у загальному випадку кодова конструкція $E(\Omega'_i)$ сегменту зображення являтиме собою поєднання окремих кодограм поліадичних чисел Θ_ℓ , тобто:

$$E(\Omega'_i) = \{\Theta_1; \Theta_2; \dots; \Theta_\ell; \dots; \Theta_L\} = \bigcup_{\ell=1}^L \Theta_\ell \quad (4.1)$$

У свою чергу, формування коду числа Θ_ℓ здійснюється на базі десяткових величин $C(i)_{x,y}$ компонент рядку або стовпця (залежно від обраного напрямку обходу сегменту Ω'_i). Таким чином, тут компонента розглядається як ключовий елемент поліадичних Θ_ℓ чисел з плаваючою вагою.

Умовимося, що кодові значення чисел Θ_ℓ будуть обчислюватися у напрямку стовпців.

Тоді далі, говорячи про розрахунок значення поліадичного Θ_ℓ числа, будемо розуміти обчислення величини Θ_ℓ для ℓ -го стовпцю сегменту.

Сам процес обчислення значення Θ_ℓ потребує відомостей про величини компонент $C(i)_{x,y}$ (елементи поліадичного числа) та значення плаваючих коефіцієнтів $W(i)_\ell$ коду.

Тут для зчитування величин $C(i)_{x,y}$ елементів поліадичного числа виконується операція постовпцьового скану сегментів.

При цьому, відмінності між формуванням поліадичних чисел за рядками та стовпцями полягають лише у тому, що отримані таким чином кодограми будуть нерівними між собою, тобто:

$$\Theta_k \neq \Theta_\ell, \quad (4.2)$$

хоча при цьому компресивні властивості біноміально-поліадичного опису будуть незмінними, як показано наступним виразом:

$$\Psi(E(\Omega'_i)) | \Theta_k = \Psi(E(\Omega'_i)) | \Theta_\ell. \quad (4.3)$$

Інакше кажучи, у випадку побудови кодових конструкцій $E(\Omega'_i)$ сегменту за рядками та стовпцями загальний обсяг біт $\Psi(E(\Omega'_i))$, який необхідний їх компактного представлення, буде однаковим [16].

У свою чергу, значення $\Psi(E(\Omega'_i))$ може бути розраховано наступним чином:

$$\Psi(E(\Omega'_i)) = \log_2 E(\Omega'_i) \quad (4.4)$$

Безпосередньо обчислення самого значення поліадичного числа Θ_ℓ здійснюється з використанням наступного виразу:

$$\Theta_\ell = \sum_{x=1}^K W(i)_\ell^{(x)} \cdot C(i)_{x,y}, \quad (4.5)$$

де $W(i)_\ell$ - складник, що являє собою плаваючий коефіцієнт для ℓ -го стовпця, що належить i -му сегменту Ω'_i .

У той же час, розрахунок значення $W(i)_\ell$ плаваючого коефіцієнта виконується з використанням множини $\{\Xi\}$ основ $\xi_{x,y}$ рядку.

При цьому, множина $\{\Xi\}$ основ може бути описана на базі наступного виразу:

$$\{\Xi\} = \phi\{\xi_{x,y}\} \quad y = \overline{1;L}. \quad (4.6)$$

Безпосередньо сама множина $\{\Xi\}$ основ поліадичних чисел формується окремо для кожного з рядків сегменту Ω'_i .

Тут елемент окремий елемент $\xi_{x,y}$ множини $\{\Xi\}$ - основа рядку – обчислюється за максимальною величиною елементу поліадичного числа, тобто, десяткової компоненти $C(i)_{x,y}$ у відповідності до наступного виразу:

$$\xi_x = \max(C(i)_{x,y}) + 1. \quad (4.7)$$

У виразі (4.7) фізичний сенс збільшення складника $\max(C(i)_{x,y})$ на одиницю пояснюється необхідністю усунення випадків невизначеності під час побудови кодових конструкцій поліадичних чисел Θ_ℓ .

Така невизначеність зумовлюється недетермінованістю значень $C(i)_{x,y}$ елементів, серед яких присутні також рівні 0.

При цьому, у випадку, коли деякий рядок x буде містити цілком нульові компоненти, $C(i)_{x,y}$, вираз (4.5) не матиме змісту.

У свою чергу, розрахунок плаваючих коефіцієнтів $W(i)_\ell$ на базі відомостей про множину $\{\Xi\}$ основ ξ_x рядків виконується згідно з наступним виразом:

$$W(i)_\ell = \prod_{\alpha=x+1}^K \xi_\alpha, \quad (4.8)$$

де K – кількість рядків (елементів $C(i)_{x,y}$ у стовпці).

Аналіз формули (4.7) свідчить про те, що обчислення плаваючих коефіцієнтів, як результатів добутку основ ξ_α рядків сегменту Ω'_i , виконується на базі елементів поліадичних чисел, не беручи до уваги елемент з індексом x (з $(x+1)$ -го включно).

Розрахувавши таким чином у спосіб, зазначений виразом (4.5), величини поліадичних чисел Θ_ℓ стовпців сегменту Ω'_i , підсумкову конструкцію для опису кодограми далі можемо розглядати у вигляді конкатенації окремих поліадичних чисел, з 1 по L -е, тобто:

$$E(\Omega'_i) = \Theta_1 \& \Theta_2 \& \dots \& \Theta_\ell \& \dots \& \Theta_L. \quad (4.9)$$

Графічно процес побудови поліадичного коду буде таким, як зображено схемою на рис. 4.1.

Таким чином, у процесі побудови компактного представлення сегментів Ω'_i зображення може бути виокремлено ряд технологічних етапів, зокрема:

1. Обчислення основ поліадичних чисел на рівні кожного окремого рядку у межах кодованого сегменту Ω'_i на базі формули (4.7).

2. Розрахунок величин плаваючих коефіцієнтів $W(i)_\ell$ поліадичних чисел Θ_ℓ , керуючись при цьому виразом (4.8).

3. Обчислення поліадичних числа Θ_ℓ на базі виразу (4.5).

4. Побудова кодограми $E(\Omega'_i)$ сегменту Ω'_i поєднанням раніше обчислених величин поліадичних чисел за формулою (4.9).

При цьому, загальна кількість біт $\Psi(E(\Omega'_i))$, необхідна для побудови компактного представлення $E(\Omega'_i)$ сегменту Ω'_i , обчислюватиметься відповідно до наступного виразу:

$$\Psi(E(\Omega'_i)) = \ell \log_2 [E(\Omega'_i)] \quad (4.10)$$

У підсумку структура кодограми, що являє собою компактний опис $E(\Omega'_i)$ сегменту Ω'_i , може передаватися у мережу у вигляді масиву, що містить у собі L поліадичних чисел у десятковій формі опису, як показано рис. 4.2.

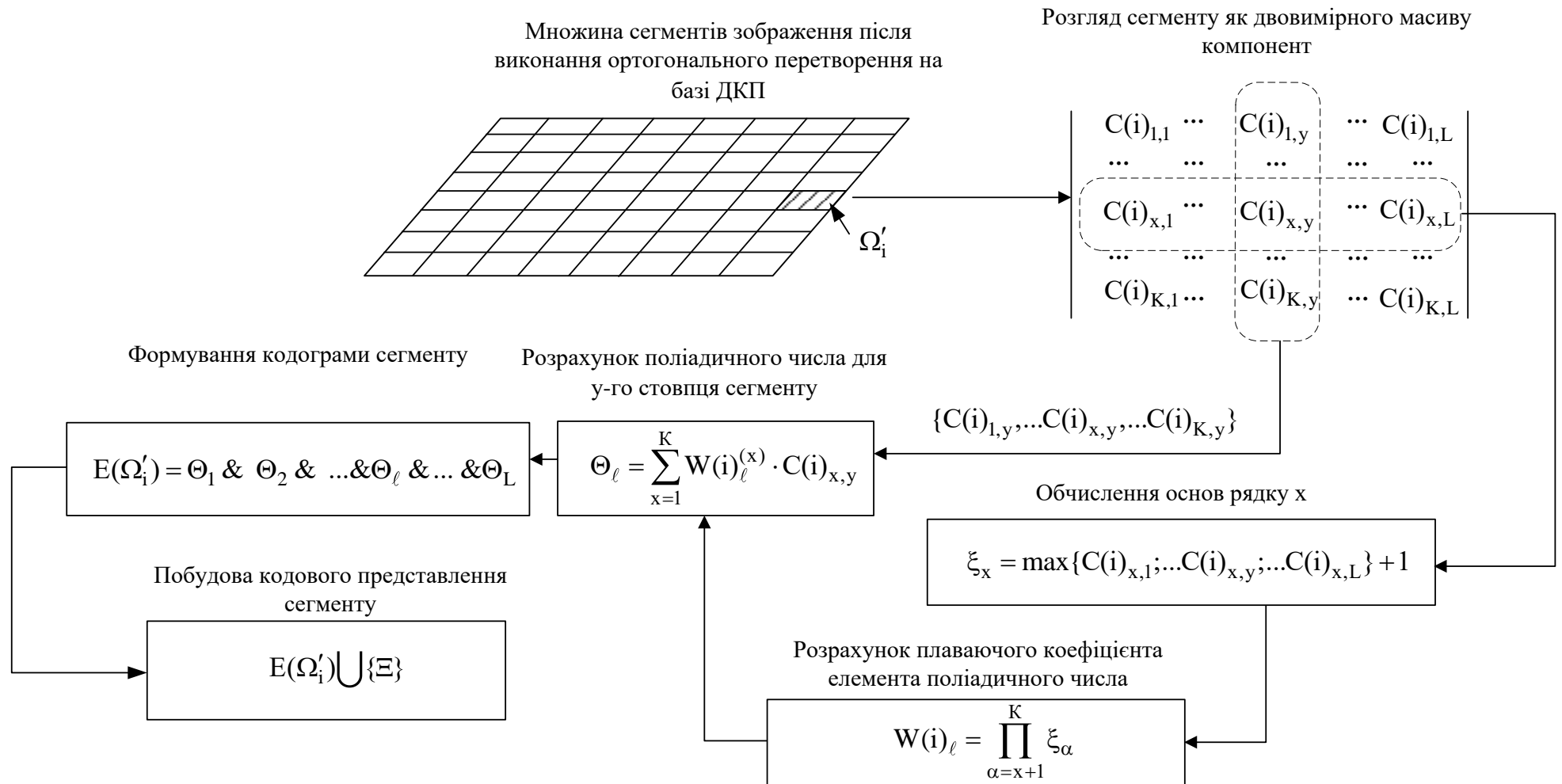


Рисунок 4.1 – Схема формування кодового опису сегменту у біноміально-поліадичному просторі

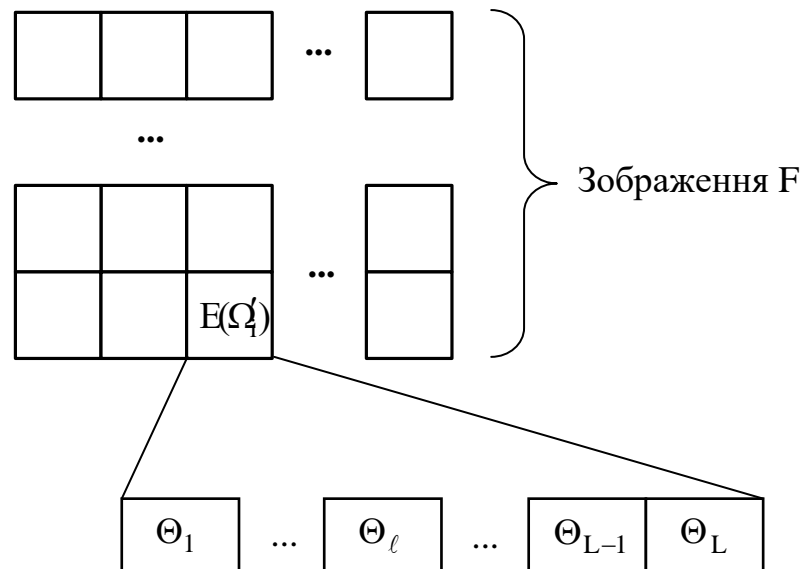


Рисунок 4.2 – Загальна структура кодограми сегменту зображення, утворена з використанням масиву розрахованих поліадичних чисел

4.2.2 Алгоритм реконструювання сегментів зображення, кодованих у біноміально-поліадичному просторі

Щоб створити умови, у яких можливо безпомилково відновити сегмент Ω'_i , попередньо компактно представлений у біноміально-поліадичному просторі, до початкового вигляду, має бути забезпечено виконання ряду вимог, а саме [16]:

1. Обсяг $\Psi(\Theta_\ell)$ біт, що необхідний для опису поліадичного числа Θ_ℓ , повинен бути не більшим, ніж розрядність V_{enc} кодового слова, що аналогічно наступному співвідношенню:

$$\Psi(\Theta_\ell) \leq V_{\text{enc}}. \quad (4.11)$$

2. Необхідно забезпечити принцип реконструкції згортки $E(\Omega'_i)$ одновимірного вектору поліадичних чисел Θ_ℓ до початкового вигляду - двовимірного масиву компонент $S(i)_{x,y}$ (сегмент Ω'_i).

3. Система службових даних, необхідна для відновлення отриманої згортки $E(\Omega'_i)$, повинна виключати ймовірність помилкового кодоутворення у ході відновлення двовимірного масиву компонент $C(i)_{x,y}$, які формують сегмент Ω'_i . При цьому, кількість біт на представлення службових даних має мінімізуватися для того, щоб уникнути помітного зменшення коефіцієнту K_{comp} стиснення.

Виходячи з зазначених вимог, пропонується розглядати множину $\{\Xi\}$ основ ξ_x рядків як головну та єдину систему службових даних.

У рамках такого підходу за умов, що початковий сегмент Ω'_i , як масив компонент $K \times L$ при $K=L$, його кодова конструкція $E(\Omega'_i)$ буде містити L поліадичних чисел та L основ ξ_x рядків.

Якщо при цьому брати до уваги те, що кількість біт V_{enc} на представлення кожної основи ξ_x буде такою ж, як і для будь-якої компоненти $C(i)_{x,y}$, тоді, порівняно з початковою кількістю біт на опис вихідного сегменту Ω'_i , службові дані будуть займати $\frac{1}{K} \left(\frac{1}{L}\right)$ від загального обсягу. Це є цілком прийнятним для сучасних технологій кодування мультимедіа.

З урахуванням зазначеного, сам процес відновлення інформації вихідного сегменту Ω'_i , як множини компонент $C(i)_{x,y}$, з кодової конструкції $E(\Omega'_i)$ буде здійснюватися поелементно у відповідності до наступного виразу:

$$C(i)_{x,y} = \left[\frac{\Theta_\ell}{W(i)_\ell} \right] - \left[\frac{\Theta_\ell}{\xi_x \times W(i)_\ell} \right] \times \xi_x, \quad (4.12)$$

Отже, з виразу (4.12) бачимо, що у процесі покомпонентного відновлення сегменту Ω'_i у спектральному представленні кожна з компонент визначається у вигляді різниці цілочисельних частин відношень поліадичного числа Θ_ℓ до відповідного плаваючого коефіцієнту $W(i)_\ell$ і поліадичного числа Θ_ℓ до добутку плаваючого коефіцієнту $W(i)_\ell$ і відповідної основи ξ_x рядку.

Таким чином, за умови, що вираз (4.11) задовольняється, повністю виконуються вимоги 2-3 до принципу відновлення утвореної кодограми $E(\Omega'_i)$ сегменту у біноміально-поліадичному просторі.

У той же час, характерною особливістю біноміально-поліадичних кодів є те, що кількість $\Psi(\Theta_\ell)$ біт, які резервуються для представлення поліадичного числа Θ_ℓ , ніколи не перевищить значення розрядність V_{enc} кодового слова (вираз 4.11) за умов, що попередньо величину V_{enc} буде задано на рівні, який є не меншим величини накопиченого добутку основ ξ_x рядків, що аналогічно виразу:

$$V_{\text{enc}} \leq \prod_{\alpha=x+1}^K \xi_\alpha \quad (4.13)$$

Таким чином, виконання умови 1 також забезпечується.

У свою чергу, структура кодового представлення сегменту Ω'_i у біноміально-поліадичному просторі буде такою, як зображено на рис. 4.3.

Як можна бачити тут, структура містить 2 базові складники, такі, як сама кодограма $E(\Omega'_i)$ сегменту, та система $\{\Xi\}$ розрахованих основ ξ_x , необхідна для реконструкції масиву компонент до початкового вигляду.

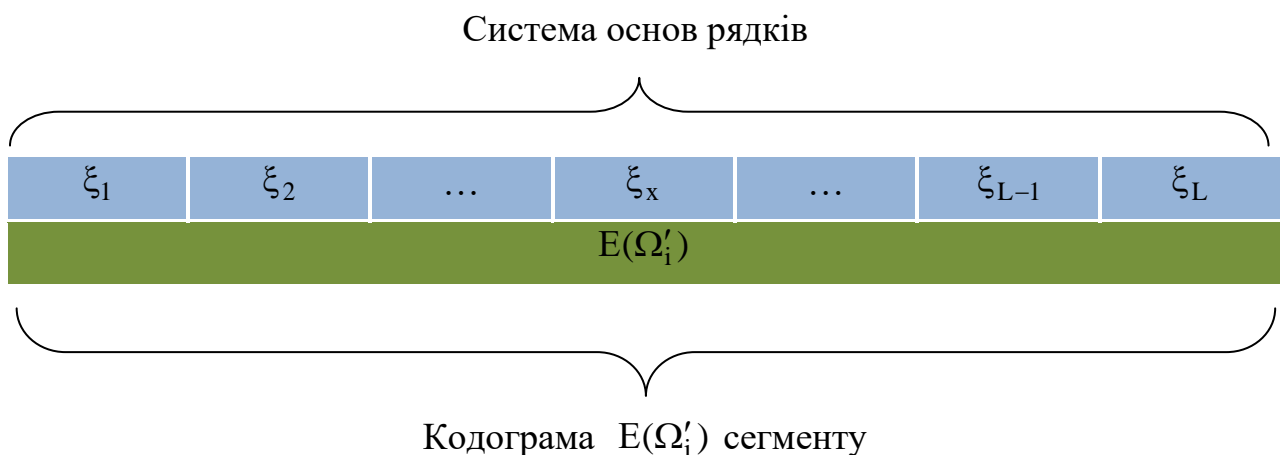


Рисунок 4.3 – Структура кодового представлення сегменту Ω'_i у біноміально-поліадичному просторі

4.3 Порівняльний аналіз методів кодування за Хафманом та кодування на базі біноміально-поліадичної системи зчислення з позицій ефективності застосування у складі технології стиснення відео

Для того, щоб мати змогу адекватного порівняння означених раніше методів кодування, першочергово умовимося з приводу критеріїв, за якими буде здійснюватися співставлення.

Так, головним критерієм може бути *швидкість* Λ *кодоутворення* на рівні одного сегменту. Даний критерій наближено може інтерпретуватися, як *кількість операцій*, необхідна для побудови компактного представлення сегменту Ω'_i .

Розрахуємо кількість операцій, які необхідно виконати для побудови коду $E(\Omega'_i)$ сегменту Ω'_i на випадок використання методу Хафмана та його компактного представлення у біноміально-поліадичному просторі.

4.3.1 Розрахунок швидкості кодоутворення для кодування за методом Хафмана

Загальна швидкість кодоутворення, виражена через кількість N_Σ арифметичних операцій у ході обробки одного сегменту Ω'_i зображення, для даного випадку може бути розрахована як [17]:

$$N_\Sigma = N_{\text{huff}} + N_{\text{rle}} + N_{\text{zz}}, \quad (4.14)$$

де N_{huff} - кількість арифметичних операцій у ході кодування за Хафманом;

N_{rle} - кількість арифметичних операцій у ході кодування RLE;

N_{zz} - кількість арифметичних операцій у ході лінеаризації (зигзаг-скану).

Кодування за Хафманом.

У загальному випадку метод потребує попередньої побудови контекстної моделі (лінеаризація та RLE). Обчислення можливого обсягу операцій для даних технологічних етапів буде виконано також.

Як зазначалося раніше, кодування за методом Хафмана спочатку потребує обчислення ймовірностей ρ появи символів вихідного алфавіту. При цьому, обчислення ймовірності ρ для одного символу виконується за виразом:

$$\rho = \frac{d_s}{D}, \quad (4.15)$$

де d_s - кількість повторів одного символу у вихідній послідовності;

D - загально кількість символів вихідної послідовності.

Очевидно, що величина D у даному випадку не є константою та суттєво залежить від особливостей змісту конкретного сегменту Ω'_i . Так, у найпростішому випадку (рис.4.4) після етапу квантизації та подальшого округлення сегмент Ω'_i може містити у собі одну DC-компоненту, та 63 нульові компоненти.

DC	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рисунок 4.4 – Найпростіший випадок сегменту Ω'_i після квантування та округлення величин компонент

Отже, у наслідок лінеаризації та виконання процедури RLE відносно даного сегменту утвориться послідовність вигляду:

$$(0, DC) (63, 0)$$

Таким чином, на обробку у ході визначення ймовірності ρ появи символів за виразом (4.15) у цьому випадку надходить 4 елементи.

Тут спершу визначається величина d_s за процедурою, що містить такі технологічні кроки, як:

1. Зчитування символу s_i .
2. Обчислення різниці Δ_s між символами s_i та s_{i+1} :

$$\Delta_s = |s_i| - |s_{i+1}| \quad (4.16)$$

3. Запис у чарунку пам'яті отриманого значення $\Delta_s(|s_i|; |s_{i+1}|)$.
4. Виконання дій, означених пунктами 1-3 для усіх i у діапазоні $i = \overline{((i+1); (D-1))}$.
5. Обчислення кількості повторів d_s символу s_i відповідно до наступного принципу:

$$\begin{cases} \mu := 1 \mid \Delta_s(|s_i|; |s_{i+1}|) = 0; \\ d_s = \sum \mu, \end{cases} \quad (4.17)$$

де μ - лічильник кількості нульових різниць Δ_s між значеннями символів.

Тобто, за виразом (4.17) обчислюється кількість символів, що мають між собою нульову різницю, отже, є однаковими.

Дана операція повторюється також для діапазону значень $i = \overline{((i+1); (D-1))}$.

6. Розрахунок ймовірності p появи символу за виразом (4.15).

Отже, у загальному випадку, якщо знехтувати операціями зчитування потоку символів, які у сутності не вносять обчислювального навантаження, та операціями запису до чарунок пам'яті (mov) тут виконуються такі математичні операції, як:

- віднімання (sub);
- додавання (add);
- ділення (div).

Тобто, сумарний обсяг N_{huff} операцій при цьому визначається наступним чином:

$$N_{\text{huff}} = N_{\text{sub}} + N_{\text{div}} + N_{\text{add}}. \quad (4.18)$$

При цьому, незалежно від початкових особливостей змісту сегменту Ω'_i , операцію sub виконується $(D-1)$ разів для кожного символу. Таким чином, загальна кількість N_{sub} операцій віднімання буде розраховуватися за виразом:

$$N_{\text{sub}} = D \times (D - 1) \quad (4.19)$$

У свою чергу, кількість операцій додавання буде різною залежно від особливостей змісту послідовності символів після RLE.

У той же час, обсяг N_{div} операцій ділення div може бути визначено, як:

$$N_{\text{div}} = \sum_{i=1}^I d_s^{(i)} \quad (4.20)$$

Інакше кажучи, з виразу (4.20) можна бачити, що тут фактично розраховується кількість різних символів, для кожного з яких буде розраховано ймовірність появи у повідомленні.

Отже, для випадку, зображеному рис. 4.4, N_{sub} за виразом (4.19) дорівнює 2.

Водночас, кількість операцій add $N_{\text{add}} = 62$.

Стосовно кількості операцій N_{div} ділення у ході визначення ймовірностей появи символів у найпростішому випадку маємо $N_{\text{div}} = 3$.

Таким чином, $N_{\text{huff}} = 2 + 62 + 3 = 67$.

Тепер розглянемо зворотній випадок, коли жоден символ s_i послідовності, утвореної у наслідок виконання лінеаризації та RLE, не має повторів. Даний випадок відповідає умовам, коли на обробку надходить сегмент Ω'_i високої структурної складності, який кодується з забезпеченням максимально можливого рівня Q якості. Про все це свідчить відсутність нульових ланцюжків нижче головної діагоналі (рис.4.5).

DC	207	139	169	131	64	24	13
205	66	122	177	89	78	30	18
198	57	160	118	102	42	22	17
111	147	128	75	37	31	15	8
95	86	103	31	20	16	6	1
71	53	40	44	27	5	4	2
25	38	21	28	12	10	3	7
23	26	29	19	14	9	11	0

Рисунок 4.5 – Найбільш складний випадок сегменту Ω'_i після квантування та округлення величин компонент

Зрозуміло, що у цьому випадку кількість елементів D після операції RLE рівнятиметься 64. Таким чином, за (4.18) необхідний обсяг N_{sub} буде рівним $N_{\text{sub}} = 64 \times 63 = 4032$.

Також на цей випадок, беручи до уваги те, що кількість повторів $d_s^{(i)}$ елементів буде рівнятися 0, це також може суттєво вплинути на розрахунок ймовірності ρ появи символів.

При формальному підході до даного процесу кількість операцій ділення буде рівною $N_{\text{div}} = 64$. Утім, оскільки даний випадок не є тривіальним, для нього може бути встановлено ознаку, за якою при $D=64$ розрахунку ймовірності не виконується взагалі. При цьому, $N_{\text{div}} = 0$, а для кожного символу присвоюється $\rho = 0,015625$, або $1/64$. Однак слід зазначити, що, виходячи з принципу роботи алгоритму Хафмана, даний випадок є безперспективним з позиції подальшої обробки, оскільки стиснення у даному випадку не може бути забезпечено.

У той же час, так як $D=64$, тоді обсяг операцій add $N_{\text{add}} = 0$.

Отже, для найгіршого випадку $N_{\text{huff}} = 4032 + 64 + 0 = 4096$ при формальному підході, або $N_{\text{huff}} = 4032$, якщо урахувати ознаку, що у послідовності відсутні подібні елементи.

Побудова контекстної моделі

Тут розглядаються етапи лінеаризації та RLE.

Як зазначалося раніше, операція зчитування даних з потоку не несе помітного навантаження та умовно може не враховуватися. У той же час,

зчитування даних виконується за певним законом, що описує зиг-заг сканування. Послідовність зчитування є наступною – (0,0), (0,1), (1,0), (2,0) і т.д. Дана послідовність формується у відповідній таблиці сканування. Тому тут необхідно забезпечити чітке позиціонування.

У свою чергу, за замовчуванням позиціонування виконується шляхом зчитування компонент у напрямку рядків. При цьому, загальний процес зчитування компоненти $C(i)_{x,y}$ потребує:

- попереднього встановлення лічильника позицій у положення (0,0);
- ідентифікації поточного (x,y) значення лічильника (операція «логічне АБО»);
- зчитування компоненти $C(i)_{x,y}$, якщо поточна позиція дорівнює одному зі значень таблиці сканування;
- поступового збільшення індексів зчитуваних позицій.

Таким чином, кількість N_{zz} операцій у ході зигзаг-сканування, якщо знехтувати елементарними операціями зчитування та позиціонування, буде рівнятися $N_{zz} = 64$ та той випадок, що розглядається сегмент 8×8 .

У свою чергу, на етапі RLE, що виконується після лінеаризації, розглядається послідовність компонент, що попередньо була зчитана та проіндексована таким чином, щоб замість двомірної ($C(i)_{x,y}$) перейти до лінійної ($C(i)_{\delta}, \delta = \overline{0; 63}$) індексації. Далі виконується зчитування значення поточної компоненти $C(i)_{\delta}$ для подальшого формування коду. При цьому, очевидно, що:

$$N_{RLE} = N_{or} + N_{let}, \quad (4.21)$$

де N_{or} - кількість операцій зчитування компонент з виявленням нульових значень на базі логічного АБО;

N_{let} - кількість операцій присвоювання.

Тут очевидно, що $N_{or} = \text{const} = 64$ для сегменту 8×8 . У свою чергу, значення N_{let} буде дорівнювати 2 у випадку, що сегмент Ω'_i формується виключно нульовими елементами, або 4, коли DC-компонента буде значимою (рис.4.4).

У найгіршому ж випадку, коли у межах спектрального опису сегменту Ω'_i не буде виявлено жодного нульового ланцюжка, $N_{\text{let}} = 128$, так як послідовність формується на базі усіх 64 початкових значень компонент.

Таким чином, маємо для найлегшого випадку $N_{\text{RLE}} = 64 + 2 = 66$. При цьому, для найгіршого випадку $N_{\text{RLE}} = 64 + 128 = 192$.

Тоді, у підсумку $N_{\Sigma} = 67 + 64 + 66 = 197$ для випадку нульового сегменту.

Відповідно, $N_{\Sigma} = 4032 + 64 + 192 = 4288$, коли у межах кодованого сегменту не виявлено ланцюжків елементів, тобто, у найгіршому випадку.

4.3.2 Розрахунок швидкості кодоутворення для кодування біноміально-поліадичним методом

Процес кодування спектрального опису сегменту Ω'_i відеокадру за рахунок компактного представлення у біноміально-поліадичному просторі містить у собі такі окремі технологічні етапи, як:

- формування системи $\{\Xi\}$ основ поліадичних чисел Θ_{ℓ} ;
- розрахунок плаваючих коефіцієнтів $W(i)_{\ell}$;
- знаходження значень поліадичних чисел Θ_{ℓ} у межах сегменту;
- формування кодової конструкції сегменту Ω'_i , як поєднання елементів множини знайдених поліадичних чисел Θ_{ℓ} .

Так, під час побудови системи основ поліадичних чисел Θ_{ℓ} , у рядках сегменту компонента $C(i)_{x,y}$, величина якої у межах рядку є максимальною.

Далі, за результатами виконання даного етапу, обчислюються основи ξ_x рядків (вираз 4.7).

Отже, операція обчислення основи ξ_x рядку включає у себе наступні арифметичні операції:

- логічне АБО;
- inc (збільшення на 1).

Зокрема, операція АБО використовується у ході виявлення найбільшого значення компоненти $C(i)_{x,y}$ рядку сегменту Ω'_i . Звідси виходить, що дана операція задіюється відносно кожного рядку, що

формують сегмент Ω'_i . При цьому, у межах кожного рядку ця операція виконується щодо кожного з елементів рядку. Тобто, якщо розглядається сегмент Ω'_i розміром 8×8 , сумарний обсяг N_{\max} виконуваних арифметичних дій під час даної процедури визначатиметься як $N_{\max} = 8 \times 8 = 64$.

Наступний крок обчислення основ ξ_x рядку, який передбачує інкрементацію попередньо виявленого у рядку максимального $\max(C(i)_{x,y})$ значення компоненти.

Таким чином, у випадку сегменту Ω'_i розміром 8×8 маємо 8-кратне виконання операції inc . Відтак, загальна кількість N_{base} математичних операцій, які необхідно виконати для формування системи основ обчислюється згідно з виразом:

$$N_{\text{base}} = N_{\max} + N_{\text{inc}}, \quad (4.22)$$

Отже, на етапі побудови системи основ рядків витрачається $N_{\text{base}} = 64 + 8 = 72$ математичні операції.

У свою чергу, після виконання розрахунків основ усіх рядків спектрального опису Ω'_i сегменту мають обчислюватися плаваючі коефіцієнти $W(i)_\ell$, як це зазначає вираз (4.8). Тут розраховується значення добутку деякої кількості основ ξ_x рядків. Відповідно, тут буде використано операцію mul .

Як видно з виразу (4.8), для розрахунку значення плаваючого коефіцієнту використовується деякий обсяг основ рядків. При цьому, якщо розраховується плаваючий коефіцієнт $W(i)_\ell$ i -го рядку, для цього застосовуються основи рядків, починаючи з наступного (з $(i + 1)$ -го рядку).

Таким чином, у ході процедури обчислення кожного наступного плаваючого коефіцієнту $W(i)_\ell$ кількість використовуваних при цьому основ рядків буде скорочуватися на 1.

Отже, тоді сумарна кількість N_{mul} необхідних математичних операцій множення (mul) у ході обчислення величин усіх плаваючих коефіцієнтів $W(i)_\ell$ для формування поліадичного числа може бути знайдена, як:

$$N_{mul} = \sum_{\beta=i-1}^1 N(\beta)_{mul}, \quad (4.23)$$

де $N_{\beta}^{(mul)}$ - необхідна кількість операцій mul, які задіюються у процесі розрахунку плаваючого коефіцієнту $W(i)_{\ell}$.

Відповідно, беручи до уваги вираз (4.8), для сегменту Ω'_i , що має розмір 8×8 , необхідний підсумковий обсяг операції mul при обчисленні плаваючих коефіцієнтів одного поліадичного числа рівнятиметься $N_{mul} = 7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$.

Далі, під час обчислення поліадичних чисел Θ_{ℓ} (вираз 4.5), знаходиться сума добутків елементів поліадичних чисел на попередньо розраховані плаваючі коефіцієнти. Тут елементами поліадичних чисел є компоненти $C(i)_{x,y}$. Тобто, використовуватиметься операція add. При цьому, так як додаванню підлягають K доданків, операцію додавання буде задіяно на 1 менше, ніж самих доданків ($K-1$).

Так як операція додавання виконуватиметься для усіх L стовпців, тоді обчислення сумарної кількості операцій додавання може бути здійснено за виразом:

$$N_{add} = (K - 1) \times L. \quad (4.24)$$

За умови, що за замовчуванням маємо $K=L=8$, тоді загальна кількість операцій додавання буде рівною 56.

Також далі, при перемноженні плаваючих коефіцієнтів $W(i)_{\ell}$ та елементів поліадичних чисел, для чого також застосовується арифметична операція mul. Оскільки у рядку міститься L , а у стовпці - K елементів позиційних чисел (компонент), загальна кількість $N_{mul}^{(final)}$ операцій mul тут можна обчислити за наступною формулою:

$$N_{\text{mul}}^{(\text{final})} = N_{\text{fc}} \times K \times L, \quad (4.25)$$

N_{fc} - обсяг плаваючих коефіцієнтів у сегменті.

Керуючись виразом (4.25), обчислюємо результуючу величину $N_{\text{mul}}^{(\text{final})} = 8 \times 8 \times 8 = 512$.

Таким чином, загальний обсяг арифметичних операцій, що виконуються у ході побудови кодового опису сегменту Ω'_i , може обчислюватися за виразом:

$$N = N_{\text{mul}}^{(\text{final})} + N_{\text{add}} + N_{\text{mul}} + N_{\text{base}}. \quad (4.26)$$

Отже, $N = 512 + 56 + 28 + 72 = 668$.

Таким чином, формування компактного кодового представлення сегменту Ω'_i у біноміально-поліадичному просторі потребує у 8 разів менше обсягу математичних обчислень, ніж у випадку кодоутворення на базі методу Хафмана для найгіршого випадку і майже у 4 рази – для випадку обробки за Хафманом сегменту середньої складності.

Як видно з процесу розрахунку, кількість операцій у ході побудови біноміально-поліадичного коду сегменту є сталою та не залежить від його особливостей. Це, у свою чергу, дозволяє більш гнучко керувати обчислювальним навантаженням на пристрій, на базі якого побудовано кодер, та резервувати певний відсоток потужності.

ВИСНОВКИ

У відповідності до умов технічного завдання, у ході виконання роботи було досліджено методи кодування без втрат, які входять до складу технологій відеокодування. Зокрема, було виконано:

1. Виявлення проблематики розвитку ряду новітніх технологічних концепцій, а саме:

- телемедицини;
- систем голографічних комунікацій;
- автономних роботизованих систем.

Було визначено, що повноцінне утілення та подальше ефективне функціонування кожної з зазначених концепцій потребує застосування у їх складі систем відеофіксації надвисокої роздільної здатності. У той же час, доступний зараз технологічний базис не дозволяє налагодити функціонування даних систем на базі інфокомунікаційної мережі. При цьому, впровадження мереж доступу 5 покоління також не гарантує сталої роботи означених систем. Можливим шляхом рішення даної проблеми є доробка та модернізація поширених методів кодування відео з тим, щоб досягти додаткового скорочення його інтенсивності з одночасною підтримкою якості.

2. Аналіз роботи сучасних методів кодування відеоінформаційних ресурсів для виявлення механізмів, модифікація яких потенційно здатна збільшити підсумковий коефіцієнт стиснення без внесення додаткової похибки. Так, визначено, що у схемі MPEG, яка є поширеною сьогодні, зараз існує лише один такий механізм – механізм статистичного кодування. На сьогодні частіше за все такі механізми реалізовано на базі методу Хафмана чи арифметичному кодуванні. Середня величина стиснення, яку вони забезпечують, є незначною. У свою чергу, інші технологічні механізми, що задіюються у ході кодування, можуть забезпечувати значно вищі, ніж статистичне кодування без втрат, коефіцієнти стиснення, хоча при цьому суттєво росте рівень внесеної помилки.

3. Досліджено існуючі підходи до кодування без втрат. Виявлено, що найбільш ефективними тут зараз є комплексні алгоритми, що мають у своєму складі власне сам механізм стиснення, та механізми побудови контекстної моделі, що, у свою чергу, підвищує ефективність ключового алгоритму стиснення без втрат.

4. Виконано оцінку роботи методу кодування за Хафманом. Розглянуто покроково процес формування коду для довільних даних. Показано, що технологічний етап стиснення на базі даного методу потребує окремого виконання етапів лінеаризації та RLE (побудови контекстної моделі). Це ж є справедливим і для арифметичного кодування.

5. Досліджено особливості стиснення без втрат на базі компактного подання сегментів відео зображення у біноміально-поліадичному просторі. Даний підхід забезпечує у середньому на 20% вищий коефіцієнт стиснення ніж для випадків статистичного кодування, що автоматично робить його перспективним для застосування у якості альтернативи існуючим методам.

6. Виконано оцінку швидкості кодоутворення для методу Хафмана та підходу на базі біноміально-поліадичного кодування. За результатами цього виявлено, що:

- метод Хафмана для побудови коду одного сегменту потребує виконання у середньому майже у 4 рази більше математичних дій, ніж для випадку формування біноміально-поліадичного опису;

- кількість дій у ході побудови коду для випадку методу Хафмана може суттєво варіюватися та залежить від особливостей змісту оброблюваних фрагментів;

- для процесу побудови біноміально-поліадичного коду кількість математичних операцій завжди є сталою та не залежить від будь-яких характеристик сегментів кадру.

7. Менша ресурсоемність методу біноміально-поліадичного кодування та прогнозованість необхідного обчислювального навантаження для його реалізації дозволяє, по-перше, утілювати кодек на його базі навіть для систем, що не відзначаються значною потужністю а по-друге – прогнозувати можливий обсяг необхідних ресурсів, що здатне підвищити ефективність функціонування системи у цілому.

Отже, усі завдання опрацьовано та виконано у повному об'ємі.

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. Telemedicine is here to stay [Електронний ресурс] – Режим доступу: https://www.buildingbetterhealthcare.com/news/article_page/Telemedicine_is_here_to_stay/178218/cn164706
4. Ultra-HDR хирургическая эндовидеокамера HDMI / USB 3.0 [Електронний ресурс] – Режим доступу: <https://endoscopy.dp.ua/catalog/equipment/NT604K-4K-Endoskopicheskaya-videocamera-s--Ultra-HDR-videochipom>
5. CNET | Holographic Communication Report [Електронний ресурс] – Режим доступу: https://www.cnet.com/tech/mobile/holographic_communication_report
6. Siegwart R., Nourbakhsh I., Scaramuzz D. Introduction to Autonomous Mobile Robots / R. Siegwart, I. Nourbakhsh, D Scaramuzz. The Mit Press, Zweite Auflage, 2017.
7. Shi, Yun Q. Image and video compression for multimedia engineering: fundamentals, algorithms, and standards / Yun Q Shi, Huifang Sun
8. 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
9. 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.
10. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1073 с.
11. Айфичер Эммануил С. Цифровая обработка сигналов: практический поход / Эммануил С. Айфичер, Барри У. Джервис. – 2-е изд. – М. : Вильямс, 2008. – 992 с.

12. Чемпен Н., Чемпен Д. Цифровые технологии мультимедиа. – М.: Вильямс, 2006. – 624 с.
13. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.
14. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – М.: Техносфера, 2004. – 368 с.
15. Ричардсон Ян. H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia / Ян Ричардсон. – Город. : Издательство, 2005. – 368 с.
16. Баранник В.В. Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков – Х.: ХУПС, 2010. – 234 с.
17. Красильников Н.Н. Цифровая обработка изображений. – М.: Вузовская книга, 2011. – 320 с.