

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Тарасенко В.П.

(підпис) (ініціали, прізвище)

“ _____ ” червня 2019 р.

Дипломний проект
на здобуття ступеня бакалавра
з напрямку підготовки 6.050102 «Комп'ютерна інженерія»

на тему:

Система захисту зображень для радіо мереж загального використання.

Виконав: студент IV курсу, групи КВ -53

(шифр групи)

Луц Ярослав Васильович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник к.т.н. доцент Орлова М.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант з нормоконтролю к.т.н. Квятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент

_____ (посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

(підпис)

**Національний технічний університет України
«Київський політехнічний інститут»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки **6.050102 «Комп'ютерна інженерія»**

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Тарасенко В.П.
(підпис) (ініціали,

прізвище)

«___» _____ 2019 р.

ЗАВДАННЯ

на дипломний проект студенту

Луц Ярославу Васильовичу

(прізвище, ім'я, по батькові)

1. Тема проекту **“Система захисту зображень в радіо мережах широкого застосування”**,

керівник проекту **доцент каф. СП і СКС, к.т.н., Орлова М.М.**,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «22» травня 2019 р. № 1330-С

2. Термін подання студентом проекту 10.06.2019.

3. Вихідні дані до проекту

Назва. Система захисту зображень для радіо мереж загального використання

4. Зміст пояснювальної записки

1. Аналіз існуючих алгоритмів стиснення зображень

2. Розробка швидкодіючого захищеного кодеку зображень для радіо мереж

3. Тестування і порівняння різних кодеків зображень

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Прогресивна передача даних. Схема структурна.

Узагальнена модель кодування зображень. Схема структурна.

Кодування зображень розробленим кодеком. Схема структурна.

Декодування зображень розробленим кодеком. Схема структурна.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Н.контроль	Клятченко Я.М., к.т.н., доцент		

7. Дата видачі завдання

16.10.2018

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Видача завдання на дипломне проектування	16.10.2018	
2.	Розробка технічного завдання	30.10.2018	
3.	Аналіз існуючих рішень	13.11.2018	
4.	Вибір середовища розробки	04.12.2018	
5.	Розробка програмного продукту	18.03.2019	
6.	Відлагодження програмного продукту	22.04.2019	
7.	Підготовка пояснювальної записки	13.05.2019	
8.	Оформлення матеріалів проекту	20.05.2019	
9.	Попередній огляд матеріалів диплому на кафедрі	27.05.2019	

Студент

(підпис)

Луц Я.В.

(ініціали, прізвище)

Керівник проекту

(підпис)

Орлова М.М.

(ініціали, прізвище)

АНОТАЦІЯ

Дипломний проект включає пояснювальну записку (55 стор., 15 рис.)

Мета роботи – експериментальне тестування, порівняння та вибір найбільш швидкодіючої реалізації кодексу зображень для визначення оптимальних за швидкодією алгоритмів стиснення зображень.

Для реалізації означеної мети проведено порівняльний аналіз існуючих методів та алгоритмів стиснення зображень за критерієм оптимального співвідношення швидкодії та обчислювальної складності.

В результаті роботи програмно реалізовано високошвидкісний кодек зображень, в якому застосовано оптимальні за швидкодією алгоритми стиснення, який стискає краще за jpg за рахунок використання алгоритмів передбачення значень пікселів, і має меншу обчислювальну складність. Додатковими перевагами розробленого кодеку зображень є прогресивна передача даних та додатковий вбудований захист для радіо мереж від завад.

Ключові слова: *високошвидкісний кодек зображень, оптимальні за швидкодією алгоритми стиснення, дискретне косинусне перетворення, цілочисельне косинусне перетворення, швидкі алгоритми перетворень, алгоритми передбачення, прогресивна передача даних .*

ABSTRACT

The diploma project includes an explanatory note (55 p., 15 fig.)

The purpose of the work is experimental testing, comparison and selection of the most high-speed implementation of the image codec to determine the performance-optimal image compression algorithms.

To achieve this goal, a comparative analysis of existing methods and image compression algorithms was carried out according to the criterion of the optimal ratio of performance and computational complexity.

As a result of the work, a high-speed image codec is implemented in software, in which the performance-optimal compression algorithms are used, which compress better than jpg due to the use of pixel value prediction algorithms and have less computational complexity. Additional benefits of the developed image codec are progressive data transmission and additional built-in interference protection for radio networks.

Keywords: *high-speed image codec, speed-optimal compression algorithms, discrete cosine transform, integer cosine transform, fast transform algorithms, prediction algorithms, progressive data transfer.*

№ п/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			Документація загальна		
			Новорозроблена		
1		ІАЛЦ.467200.002 ТЗ	Система захисту зображень для радіомереж загального використання. Технічне завдання	4	
2		ІАЛЦ.467200.003 ВП	Система захисту зображень для радіомереж загального використання. Відомість технічного проекту	1	
3	A4	ІАЛЦ.467200.004 ПЗ	Система захисту зображень для радіомереж загального використання. Пояснювальна записка	55	
4	A1	ІАЛЦ.467200.005 Д1	Прогресивна передача даних. Схема структурна.	1	
5	A1	ІАЛЦ.045450.006 Д1	Узагальнена схема кодування зображень. Схема структурна.	1	
6	A1	ІАЛЦ.045450.007 Д1	Кодування за схемою розробленого кодеку зображень. Схема структурна.	1	
7	A1	ІАЛЦ.045450.008 Д1	Декодування за схемою кодеку зображень. Схема структурна.	1	
		CD-ROM	Матеріали бакалаврського проекту	1	
			ІАЛЦ.045450.001 ОА		
Зм.	Арк	№ докум	Підпис	Дата	
Розроб.		Луц Я.В.			Система захисту зображень для радіомереж загального використання Опис альбому
Перевір.		Орлова М.М.			
Н. контр.		Клятченко Я.М.			
Затв.		Тарасенко В.П.			
					Лім. Арк. Аркушів 1 1
					КПІ ім. Ігоря Сікорського, ФПМ, КВ-53

Зміст

1.	НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ _____	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ _____	2
3.	ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ _____	2
4.	ДЖЕРЕЛА РОБОТИ _____	2
5.	ТЕХНІЧНІ ВИМОГИ _____	3
5.1.	Вимоги до системи, що розробляється _____	3
5.2.	Вимоги до апаратного забезпечення _____	3
5.3.	Вимоги до програмного забезпечення _____	3
6.	ЕТАПИ РОЗРОБКИ _____	4

					ІАЛЦ.467200.002 ТЗ		
Зм.	Арк.	№ докум.	Підп.	Дата	Система захисту зображень для радіо мереж загального використання Технічне завдання		
Розроб.		Луц Я.В.					
Перевір.		Орлова М.М.					
Н.контр.		Клятченко Я.М.					
Затв.		Гарасенко В.П.					
					<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
					1	4	
					КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування роботи – «Система захисту зображень для радіомереж загального використання».

Область застосування: оптимальне за швидкодією кодування, декодування і захист зображень для широкого кола застосувань, в першу чергу – в радіомережах інтернету речей.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є програмна реалізація швидкодіючого кодеку зображень, на основі оптимальних за швидкодією алгоритмів стиснення, з додатковими вбудованими швидкодіючими алгоритмами захисту даних.

4. ДЖЕРЕЛА РОБОТИ

Джерелами роботи є науково-технічна література по стисненню зображень, електронні статті та файли у мережі Інтернет, що досліджують та описують алгоритми стиснення і захисту.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до системи, що розробляється:

- розробка версій високоефективного по швидкодії кодеку зображень з різними алгоритмами для їх порівняння та оцінки;
- пошук оптимального графічного формату, що задовольняє найбільш важливим критеріям використання кодеку зображень (швидкодія, ефективність стиснення, прогресивна передача даних, анімація, альфа-канал прозорості та ін.)
- пошук додаткових переваг для кодеку зображень, в тому числі швидкодіючих алгоритмів захисту зображень в радіомережах.

5.2. Вимоги до апаратного забезпечення:

- Комп'ютер на базі процесора сімейства AMD, A4-4020 та вище або сімейства Intel, G3900 та вище;
- оперативна пам'ять 256 Мбайт або більше.

5.3. Вимоги до програмного забезпечення:

1. Операційна система одного з сімейств:

- Windows XP або вище;
- Linux 3.0 або вище;
- Mac OS X 10.7.3 (Lion) або вище.

2. Додаткові інсталяційні пакети:

- C++

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Видача завдання на дипломне проектування	16.10.2018
2.	Розробка технічного завдання	30.10.2018
3.	Аналіз існуючих рішень	13.11.2018
4.	Вибір середовища розробки	4.12.2018
5.	Розробка програмного продукту	18.03.2019
6.	Відлагодження програмного продукту	22.04.2019
7.	Підготовка пояснювальної записки	13.05.2019
8.	Оформлення матеріалів проекту	20.05.2019
9.	Попередній огляд матеріалів диплому на кафедрі	27.05.2019

№ п/п	Формат	Позначення	Найменування	Кількість листів	Примітка						
			Документація загальна								
			Новорозроблена								
1	A4	ІАЛЦ.467200.004 ПЗ	Система захисту зображень для радіомереж загального використання.	55							
			Пояснювальна записка проекту								
2	A1	ІАЛЦ.467200.005 Д1	Прогресивна передача даних. Схеми структурна.	1							
3	A1	ІАЛЦ.467200.006 Д1	Узагальнена схема кодування зображень. Схеми структурна.	1							
4	A1	ІАЛЦ.467200.007 Д1	Кодування за схемою розробленого кодексу зображень. Схеми структурна.	1							
5	A1	ІАЛЦ.467200.008 Д1	Декодування за схемою розробленого кодексу зображень. Схеми структурна.	1							
		CD-ROM	Матеріали бакалаврського проекту	1							
			ІАЛЦ.045450.003 ВП								
Зм.	Арк	№ докум	Підпис	Дата							
Розроб.		Луц Я.В.			<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;"> <p>Система захисту зображень для радіомереж загального використання</p> <p>Відомість проекту</p> </div> <div style="text-align: center;"> <table border="1"> <thead> <tr> <th>Літ.</th> <th>Арк.</th> <th>Аркушів</th> </tr> </thead> <tbody> <tr> <td></td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p>КПІ ім. Ігоря Сікорського, ФПМ КВ-53</p> </div> </div>	Літ.	Арк.	Аркушів		1	1
Літ.	Арк.	Аркушів									
	1	1									
Перевір.		Орлова М.М.									
Н. контр.		Клятченко Я.М.									
Затв.		Тарасенко В.П.									

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП.....	5
1. АНАЛІЗ ІСНУЮЧИХ АЛГОРИТМІВ СТИСНЕННЯ ЗОБРАЖЕНЬ.....	9
1.1 Аналіз графічних форматів.....	9
1.2. Аналіз існуючих методів стиснення зображень.....	14
1.3. Обґрунтування теми дипломного проекту.....	19
1.4. Обґрунтування вибору середовища розробки.....	20
Висновки до розділу.....	21
2. РОЗРОБКА ШВИДКОДІЮЧОГО ЗАХИЩЕНОГО КОДЕКУ ЗОБРАЖЕНЬ ДЛЯ РАДІОМЕРЕЖ.....	23
2.1. Прогресивна передача даних.....	23
2.2. Передбачення.....	25
2.3. Перетворення та коефіцієнт стиснення з втратами.....	28
2.4. Захист зображень в радіомережах.....	35
Висновки до розділу.....	36
3. ТЕСТУВАННЯ І ПОРІВНЯННЯ РІЗНИХ КОДЕКІВ ЗОБРАЖЕНЬ.....	38
3.1. Параметри і задачі тестування.....	38
3.2. Розмірність та адаптивний вибір перетворення.....	42
3.3. Коефіцієнт стиснення з втратами.....	47
3.4. Вплив на загальне стиснення різних алгоритмів.....	49
Висновки до розділу.....	50
ВИСНОВОК.....	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	54
ДОДАТКИ	

					ІАЛЦ.467200.004 ПЗ				
Зм.	Лист	№ докум.	Підп.	Дата					
Розроб.	Луц Я.В.				<i>Система захисту зображень для радіо мереж загального використання</i>	Літ.	Лист	Листів	
Перев.	Орлова М.М.						1	55	
Н.контр.	Клятченко Я.М.				Пояснювальна записка				
Затв.	Тарасенко В.П.				КПІ ім. Ігоря Сікорського, ФПМ, КВ-53				

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.467200.005 Д1. Прогресивна передача даних. Схема структурна.
- ІАЛЦ.467200.006 Д1. Узагальнена модель кодування зображень. Схема структурна.
- ІАЛЦ.467200.007 Д1. Кодування зображень розробленим кодеком. Схема структурна.
- ІАЛЦ.467200.008 Д1. Декодування зображень розробленим кодеком. Схема структурна.

Додаток 2. Лістинг програмного коду

Додаток 3. Презентація

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ДКП – дискретне косинусне перетворення

ПКЛ – перетворення Карунена-Лоева

СКВ – середньоквадратичне відхилення

ЦКП – цілочисельне косинусне перетворення

AVIF – AV1 Still Image File Format, кодек зображень на основі відеокодека

AV1 від Mozilla, Гугл та інших фірм з Alliance for Open Media.

FLIF – кодек зображень без втрат, на основі передбачення і ентропійного кодування

H.265 – сучасний стандарт стиснення відео, наступник H.264

HDR – High Dynamic Range, збільшення діапазону яскравості

HEIF, HEIC – кодек зображень на основі специфікації відеокодека H.265

IoT – Internet of Things, інтернет речей

JPEG 2000 – кодек зображень із застосуванням вейвлетів

JPEG XR – кодек зображень розроблений фірмою Microsoft у 2006 р.

JPEG – розроблений у 1992 р. растровий формат збереження зображень, що використовує стиснення з втратами на основі блочного кодування ДКП

PSNR – пікове співвідношення сигнал-шум

QP – параметр квантування коефіцієнтів перетворення

Raw – (англ. raw сирий) – формат даних, що містить повні необроблені (або мінімально оброблені) дані і не має чіткої специфікації.

RDO – rate-distortion optimization, оптимізація співвідношення стиснення/шум

RGB – простір кольорів, формується градаціями складових каналів (Red, Green, Blue)

SAD – сума абсолютних різниць значень пікселів

SATD – сума абсолютних різниць значень коефіцієнтів перетворень

VP9 – відкритий стандарт стиснення відео від корпорації Google

VVC – Versatile video coding, відео кодек на заміну H.265

WebP – кодек зображень на основі відеокодека VP9

YCbCr – простір кольорів, де колір представляється через яскравість (Y) і дві різниці кольорів, Cb – різниця по Blue і Cr – різниця по Red.

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		4

ВСТУП

Актуальність дослідження та побудови швидкодіючого захищеного кодека зображень обумовлена розвитком індустрії інтернету речей, збільшенням кількості відеоданих і зображень, що генеруються та передаються, появою нових специфікацій графічних форматів (збільшення роздільної здатності до 4К, 8К і навіть 16К, HDR (збільшення діапазону яскравості), збільшення розрядності з 8 до 12-16 біт на піксель).

Інтернет речей (IoT) – це радіо мережа пристроїв, які взаємодіють між собою та керуючими пристроями, передаючи масиви різноманітної інформації, в тому числі – зображення та відеодані. Незважаючи на те, що передача інформації відбувається в основному бездротовим шляхом, де традиційно обмежена швидкість обміну даними, найбільш важливою проблемою є збільшення часу автономності пристроїв, оскільки більшість з них обладнані автономним (а отже – обмеженим) живленням.

Відповідно, для подовження строку використання пристроїв інтернету речей, актуальною задачею є зменшення обчислювальної складності програмного забезпечення та в цілому зниження вимог до потужності процесорів для обробки даних.

З іншого боку, спостерігається стійка тенденція до значного підвищення якості графічних даних – збільшення роздільної здатності, кількості біт на піксель, що вимагає значного зростання необхідних обчислювальних ресурсів. З цієї точки зору, також є доцільним розглянути шляхи зменшення обчислювальної складності при незначному збільшенні розміру графічних даних (апріорні оцінки збільшення розміру файлів – 25-30%, апріорні оцінки зменшення обчислювальної складності в 2-8 разів порівняно з кодеками зображень WebP та HEIF, та в 1,2 рази зменшення розміру файлу і в 1,5 рази збільшення швидкодії відносно кодеку JPEG).

Оцінка можливого збільшення розміру графічних файлів є одним із завдань даної роботи, оскільки доцільність використання спрощених, високошвидкісних алгоритмів стиснення зображень та відео (в подальших дослідженнях) залежить від співвідношення зменшення обчислювальних витрат та збільшення розміру графічних файлів.

В цілому, є три основні фактори, які мають значення при оцінці доцільності розробки оптимальних по швидкодії кодеків зображення та відео: обчислювальні ресурси при кодуванні та декодуванні (відновленні) даних, пропускна спроможність каналу передачі інформації та необхідна пам'ять для збереження даних. При низькій швидкості каналів зв'язку та малих об'ємах наявних накопичувачів даних перевага надається максимальному стисненню даних, а при високих значеннях цих параметрів – оптимальним по швидкодії алгоритмам для зменшення обчислювальної складності та більш швидкої обробки даних (наприклад, в реальному часі).

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

Але досягнення в області збільшення розміру накопичувачів даних та каналів зв'язку при меншій швидкості зростання потужності процесорів ставить задачу зменшення обчислювальної складності алгоритмів стиснення, незважаючи на деяке очікуване збільшення розміру графічних файлів.

Іншим важливим фактором є зміна співвідношення операцій кодування й декодування. Раніше кодування графічних даних було менш поширеним явищем, тому висока складність кодування (на порядок) відносно декодування не грала особливої ролі. Але внаслідок широкого розповсюдження пристроїв для первинного кодування графічних даних набирає важливості задача зменшення обчислювальної складності

кодування, особливо для області інтернету речей, де саме кодування зображень та відео є основним процесом. Для широкого розповсюдження використання відео зйомки (наприклад, повний відео архів життя людини) потрібно значне зниження вартості потрібних для цього ресурсів (обчислювальних та пам'яті).

Таким чином, в даній роботі вперше зроблено пошук оптимального співвідношення швидкодії та ефективності кодеку зображень (який може стати в подальшому основою для розробки відеокодека). В даній роботі не ставиться задача розробити оптимальний кодек зображень – оскільки це надскладна задача, для вирішення якої потрібен колектив висококваліфікованих спеціалістів, але отримати в першому наближенні необхідні результати, оцінити доцільність цієї задачі (знайти приблизні оцінки збільшення розміру файлів зображення та зменшення кількості обчислень) та порівняти існуючі альтернативні алгоритми стиснення з реалізацією і тестуванням обраних алгоритмів виглядає посиленою задачею.

Крім критерія швидкодії алгоритмів на формат майбутнього оптимального кодеку зображень мають вплив інші фактори, серед яких – додатковий вбудований захист, можливість прогресивної передачі даних (тобто формування, представлення декількох зменшених копій оригінального зображення), наявність чи відсутність каналу прозорості (альфа-канал), можливість використання анімації (наприклад, формат png мав би значно більшу популярність, якби він одразу підтримував анімацію).

Іншим фактором, що може мати відношення до особливостей реалізації кодеку зображень, є зручність для алгоритмів розпізнавання образів, зображень. Кодек, який включає в себе частину необхідної для розпізнавання статистики зображення як складові алгоритмів стиснення мав би перевагу перед кодеком, який надає лише безпосередньо зображення для аналізу алгоритмам розпізнавання.

Також представляє певну цікавість порівняння алгоритмів ентропійного кодування з метою знаходження оптимального за швидкістю кодека зображень. Але ці питання не розглядались в даній роботі по декільком причинам, в тому числі тому, що діапазон впливу різних алгоритмів ентропійного кодування на розмір отриманого графічного файлу незначний на фоні значного збільшення обчислювальної складності більш потужних алгоритмів. Відповідно, в даній роботі зроблено акцент на перших двох етапах стиснення даних – передбаченні значень пікселів, що дає досить значний стиск оригінального зображення, причому без втрат, і етапі використання перетворень, на якому відбувається стиск з втратами за рахунок заокруглення та відкидання частини отриманих коефіцієнтів перетворення.

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		8

1. АНАЛІЗ ІСНУЮЧИХ АЛГОРИТМІВ СТИСНЕННЯ ЗОБРАЖЕНЬ

1. 1. Аналіз графічних форматів

Основними сучасними графічними форматами, що найчастіше використовуються, є наступні [1, 2]:

- JPEG;
- WebP;
- PNG;
- GIF.

Також останнім часом набувають поширення нові графічні формати, які мають більш високий коефіцієнт стиснення (але в результаті значного збільшення обчислювальної складності), серед яких є можна виділити кодеки зображень на основі нових відеокодеків: формати зображень HEIF/HEIC та BPG [3, 4, 5] (обидва на основі відеокодека H.265/HEVC), AV1 Still Image File Format (AVIF) – на основі відеокодека AV1 [6]. WebP також базується на алгоритмах відеокодеків VP8 та VP9 [7].

Ще одним новим форматом є FLIF [8] – кодек зображень без втрат, лише на основі передбачення і ентропійного кодування, без використання перетворень. Як і більшість сучасних кодеків, він відрізняється значною обчислювальною складністю і має вузьку сферу застосування.

Існують і комплексні рішення, де застосовуються декілька графічних форматів одночасно, наприклад, DjVu [1,2] – технологія стиснення зображення з втратами, де використовуються декілька технологій, в тому числі: алгоритм відділення тексту від фону; вейвлетний алгоритм стиснення фону IW44; алгоритм стиснення чорно-білого зображення JB2; універсальний алгоритм стиснення ZP; алгоритм часткового відкриття по запиту; алгоритм виділення масок, слоїв зображення (передній план, фон та

чорно-біла маска). До кожного зі слів використовуються окремі алгоритми стискання.

Комплексний підхід, застосований у форматі DjVu, заслуговує уваги та може служити прикладом застосування інтегральної оцінки для створення нових графічних форматів, які включають в себе декілька алгоритмів для різного типу зображень і даних. З іншого боку, історія формату MNG (Multiple-image Network Graphics), що включав в себе два відомі формати одночасно – JPEG та PNG, причому з анімацією, демонструє інший приклад – значне збільшення необхідних для використання формату службових бібліотек є недоліком, що був основною причиною для відмови від його широкого застосування.

Отже, новий графічний формат повинен включати багато можливостей (анімацію, прозорість, прогресивну передачу, захист та ін.), мати комплексний підхід до типів зображень і одночасно бути достатньо простим, легким для виконання. Аналіз формату MNG з метою його спрощення, покращення та оптимізації може бути складовою досліджень по створенню графічних форматів нових поколінь.

Основними алгоритмами, що застосовуються для стиснення зображень з втратами є:

- вейвлети;
- блочні перетворення;
- фрактали;
- прорідження (відкидання) даних.

Основними алгоритмами, що застосовуються для стиснення зображень без втрат є:

- ентропійне кодування;
- передбачення (значень пікселів та/або коефіцієнтів перетворення).

Графічні формати для збереження зображень можна розділити на дві основні категорії (також можливий комплексний, змішаний підхід) [1, 2]:

- растрові (BMP, TIFF, PSD, JPEG, PNG, GIF);
- векторні (WMF, EPS, CDR, AI та інші).

Растровий файл є таблицею (bitmap), в кожній клітинці якої є піксель. Найпростішим растровим файлом з відомих форматів є BMP – він не стискає взагалі дані (хоча, якби до нього добавили хоча б один простий алгоритм для передбачення значень пікселів для стиску без втрат – він був би більш популярний внаслідок значного зменшення розміру файлу).

Векторний графічний файл – це набір геометричних примітивів. Більшість векторних форматів є гібридними, тобто можуть також містити у файлі растрові дані або посилання на них (технологія OPI).

Ще один давно відомий і вже морально застарілий формат GIF [1, 2] (Graphics Interchange Format) – використовує лише 8-бітовий колір і досить ефективно стискає суцільні кольорові області, при цьому зберігаючи необхідні деталі зображення. Кількість кольорів в зображенні може коливатись від 2 до 256 (але на вибір із наявної 24-бітової палітри). GIF також має так званий альфа-канал прозорості.

GIF підтримує покадрову зміну зображень, що зробило формат популярним для створення простої анімації. Для отримання додаткових кольорів, які відсутні в базовій палітрі, можливе застосування таких алгоритмів як розсіювання, накладення періодичної структури і генерація шуму (анг. dithering, pattern та noise) [1, 2]. GIF формат зберігає зображення в моделі кольорів RGB, що задані як індексовані кольори (Indexed color).

JPEG або JPG (Joint Photographic Experts Group) [9] – графічний формат з втратами для зберігання растрових даних в колірних моделях Grayscale (відтінки сірого), YCbCr та RGB. Не підтримує альфа-канал (прозорість) та анімацію, але все одно є найбільш поширеним форматом збереження зображень. Є варіант реалізації з прогресивною передачею даних, але незручний для користувача (значна розмитість на першому кроці відновлення зображення і висока обчислювальна складність).

PNG (portable network graphics) – був створений для заміни формату GIF, він має альфа-канал, непогано стискає чорно-білі зображення (текстові, графіки), але не мав спочатку підтримки анімації, що значно зашкодило його популярності. Формально PNG використовує тільки стиснення без втрат, але по факту використовує прорідження, тобто частина інформації втрачається в деяких режимах роботи PNG, і для фото він не оптимальний.

JPEG-2000 – кодек зображень, що базується на вейвлетах, тобто розкладі в базисі вейвлет-функцій. Переваги вейвлетів – значний діапазон базових функцій, які масштабуються і в сумі дозволяють досить якісно представити сигнал, зображення та інші дані (наприклад, мультифрактальний аналіз використовується для детального дослідження різноманітних даних з різних областей). Недоліки вейвлетів – значна обчислювальна складність та недостатньо ефективна передача дрібних деталей (оскільки вейвлети зазвичай мають два-три рівні ієрархії коефіцієнтів, тобто коефіцієнти першого рівня кодуються, стискаються ще один або два рази – і тому вейвлети значно краще підходять для великих коефіцієнтів стиснення ніж для малих). Вейвлети розглядались як альтернатива блочним перетворенням, але були забраковані внаслідок меншої якості при малих і середніх коефіцієнтах стиснення [10]. Фрактали подібні до вейвлетів, і мають схожі недоліки, але вони менш адаптовані до різноманіття даних, тобто використання фракталів може бути цікавим лише для вузьких класів зображень.

Найбільш перспективним з існуючих графічних форматів вважається WebP, але він не працює в колірному просторі RGB, і перед кодуванням зображення переводить його в YUV (з глибиною 8 біт) і форматом 4:2:0 [7] (тобто застосовує прорідження даних по двом шкалам, що є неефективним алгоритмом з точки зору збереження високої якості зображень). Більш того – для деяких зображень може використовуватися алгоритм upscaling, коли кодується не саме зображення, а його масштабована (зменшена) версія

(декодер проводить зворотнє перетворення, тобто збільшення зображення).

Оскільки WebP був розрахований на використання в інтернеті, для браузерів, то акцент робився на максимально можливому стисненні даних, а не на збереженні високої якості зображень. Основна перевага WebP перед

JPEG – наявність додаткового стиснення за рахунок алгоритмів передбачення значень пікселів (але передбачення застосовується майже у всіх сучасних кодеках, оскільки це ефективне стиснення без втрат). JPEG, як і WebP, використовує ДКП (дискретне косинусне перетворення) замість ЦКП (цілочисельної апроксимації косинусного перетворення). Деяке уявлення про поширення різних форматів на інтернет-сайтах можна отримати з рисунка 1.1:

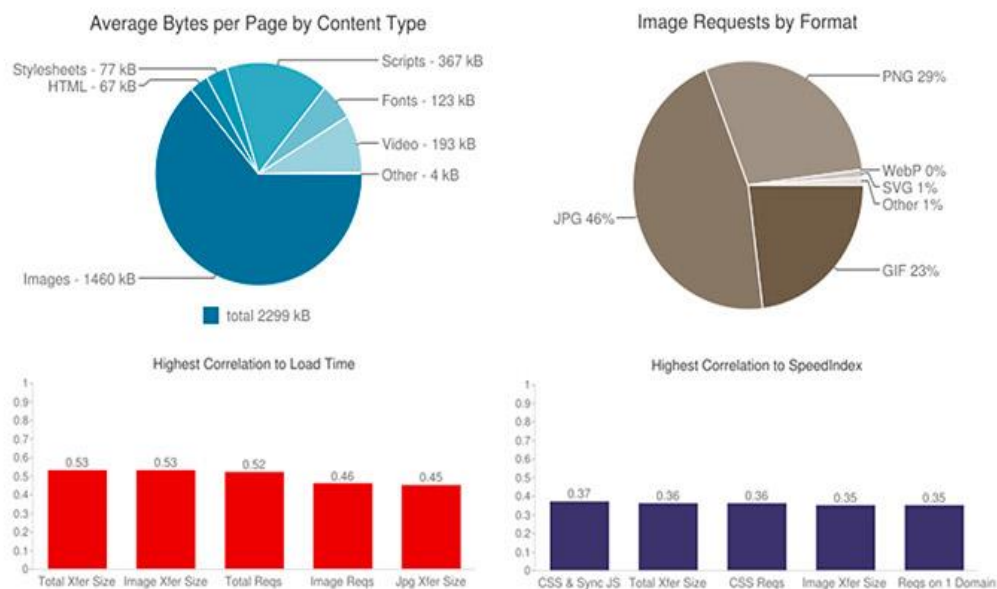


Рисунок 1.1 – Статистичні дані використання різних форматів в інтернеті

1. 2. Аналіз існуючих методів стиснення зображень

Перейдемо до більш детального дослідження, порівняння, аналізу існуючих методів і підходів стиснення зображень та відео [10-13].

Стиснення зображень відбувається в три етапи:

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

- ентропійне кодування (стиснення без втрат).

Відповідно, на першому і третьому етапах виникає обмеження можливості стиснення внаслідок умови «стиснення без втрат», а другий етап дозволяє й далі значно збільшувати коефіцієнт стиснення, але за рахунок втрат якості та даних.

Зазначимо, що при використанні перетворень при коефіцієнту квантування QR (quantization coefficient), що дорівнює одиниці (коефіцієнт QR – параметр, на який ділять отримані коефіцієнти перетворення для більшого стиснення), отримаємо стиснення без втрат, яке в залежності від типу даних, зображення буде коливатись в діапазоні від 0,5 до 0,9 (тобто стиснення за рахунок перетворення буде від 1,1 до 2 раз). А, наприклад, при $QR=8$ отримаємо високу якість відновленого зображення, але вже з незначними втратами початкової інформації.

Підкреслимо, що при переході від простору кольорів RGB до YCbCr також спостерігаються певні втрати даних, які вважаються прийнятними для людського ока, оскільки перепади яскравості око сприймає як більш важливу інформацію, ніж зміни кольорів.

Але з урахуванням тенденцій до підвищення якості зображень та додаткових критеріїв оцінки (наприклад, для розпізнавання) в майбутньому використання режимів кодеків із використанням RGB буде все частіше.

Інше зауваження щодо переводу RGB до YCbCr стосується можливості кращого стиснення – внаслідок переходу зменшується природна кореляція в блоках між значеннями пікселів, що зменшує можливості стиснення (оскільки високорельовані дані краще стискаються). Точніше – висока початкова кореляція по каналам RGB трансформується у високу кореляцію по каналу Y (яскравості) і більш низьку – по каналам різниць Cb та Cr.

З іншого боку, оскільки перетворення – це другий етап, то в результаті

передбачення пікселів на першому етапі кодування отримуються дані з низькою кореляцією, що буде спостерігатись як для RGB, так і для YCbCr.

Передбачення пікселів найчастіше формуються на підставі значень пікселів сусідніх блоків зображення, і для цього використовуються різні алгоритми, які послідовно застосовуються, після чого результати порівнюються та обирається найкращий для даного блоку пікселів алгоритм передбачення. При цьому складність оцінки співвідношення сигнал/шум досить висока, більша ніж складність кожного з алгоритмів. Але значна кількість алгоритмів передбачення також створює додаткові обчислення, наприклад, якщо у WebP використовувалось 10 алгоритмів передбачення [7], а у H.264 – 9 [14], то у кодека H.265 вже 35 алгоритмів [10] (як для відеокодека, так і для кодека зображень на його основі), а у майбутнього кодека VVC (Versatile video coding) взагалі планується 67 алгоритмів передбачення (prediction modes) [15].

Таким чином, постає питання – яке оптимальне число алгоритмів передбачення значень пікселів з точки зору енергоефективності кодеку? Для початку бажано перевірити найпростіший варіант – коли алгоритм передбачення універсальний, тобто один, і не потрібно додатково оцінювати й порівнювати результати. Саме ця гіпотеза й була перевірена в рамках даної роботи в першу чергу.

Для кращої ілюстрації впливу на коефіцієнт стиснення алгоритмів передбачення приведемо результат передбачення – зображення, яке сформоване з різниць (залишків) в результаті операції віднімання від абсолютних значень пікселів оригінального зображення («фото») передбачених алгоритмами значень (рис. 1.2). При цьому використовувався лише один алгоритм передбачення (на основі значення пікселів з сусіднього блоку), взятий з кодеку WebP. Для порівняння, приведемо інше зображення, яке відображає результат алгоритму передбачення на основі середнього значення блоку (рис. 1.3), який був протестований в даній роботі в якості

одного з можливих універсальних алгоритмів передбачення.



Рисунок 1.2 – Результат передбачення значень пікселів алгоритмом WebP



Рисунок 1.3 – Результат передбачення запропонованим алгоритмом

Для кращого сприйняття зображень була проведена інверсія кольорів (оскільки на білому фоні краще помітні зміни, перепади значень ніж на чорному фоні, як воно мало бути).

Зауважимо, що подібна висока ефективність алгоритмів передбачення буває лише на висококорельованих зображеннях, а для

низькокорельованих зображень результати будуть приблизно такими, як на рис. 1.4, де зображені різниці (залишки) для тестового зображення «пейзаж».



Рис. 1.4 – Результат передбачення запропонованим алгоритмом

Перейдемо до оцінки другого етапу кодування зображень – за допомогою перетворень. Тут потрібно розглянути наступні питання [10-13]:

- оптимальна розмірність перетворення (фіксована чи адаптивна);
- швидкі алгоритми реалізації перетворень;
- квантування коефіцієнтів перетворення (фіксоване чи адаптивне);
- 1D чи 2D перетворення;
- блоки з перекриттям чи без.

Почнемо з останнього. Наявність використання блоків з перекриттям в дуже малій кількості існуючих кодеків зображень і широке розповсюдження блоків без перекриття можна вважати підставою для вибору блоків без перекриття з метою зменшення обчислювальних затрат. Для чого взагалі використовується технологія, алгоритм з перекриттям сусідніх блоків? Для кращої якості збережених даних (подібний підхід,

прийом також реалізується в аудіокодеках), тобто для зменшення блочних артефактів, спотворення даних. Але якщо враховувати тенденцію на збільшення якості зображень, то перекриття блоків, тобто дублювання інформації, здається зайвим – висока якість буде забезпечена малими значеннями коефіцієнту стиснення, тобто квантування.

Питання щодо вибору 1D чи 2D алгоритму перетворення здається на перший погляд архаїчним, оскільки давно відомо, що 2D краще за 1D [14]. Але задля перевірки власної гіпотези порівняємо, протестуємо обидва варіанти, оскільки при 2D реалізації в 2 рази збільшується обчислювальна складність алгоритму, тому незначне зменшення коефіцієнту стиснення при 1D може бути обґрунтованим.

Наявність швидких алгоритмів реалізації перетворень є беззаперечною перевагою, оскільки за рахунок цього обчислювальна складність перетворення зменшується в декілька разів, від 3 до 11 (в залежності від розмірності перетворення) [16,17]. Відповідно, це питання автоматично вирішується на користь застосування відомих перетворень зі швидкими алгоритмами їх реалізацій [16,17]. Так само перевага надається ЦКП перед ДКП, оскільки цілочисельна ітерація косинусного перетворення при більшості умов працює швидше за дискретну версію.

Питання щодо адаптивного вибору коефіцієнтів квантування для кожного блоку та адаптивного вибору розмірності перетворення для різних областей зображення є найбільш складними, оскільки зростання обчислювальних затрат, ресурсів при адаптивній реалізації в деякій мірі компенсується збільшенням коефіцієнту стиснення в результаті. Наскільки це доцільно – покаже тестування, при якому була перевірена гіпотеза щодо застосування оптимального (але фіксованого) для даного типу зображень перетворення та коефіцієнту квантування. Можливо, можна буде оптимізувати процес вибору оптимальних параметрів кодування за рахунок менш складних по кількості обчислень алгоритмів, оскільки існуючі

алгоритми rate-distortion optimization (оптимізації співвідношення стиснення/шум) є досить затратними, вимогливими до кількості ресурсів. Також цікавим питанням є оцінка типу зображення та областей зображення.

1.3. Обґрунтування теми дипломного проекту

Як було вказано у вступі, актуальність і доцільність створення оптимального по швидкодії, високошвидкісного кодеку зображень ґрунтується на двох тенденціях найближчого майбутнього: широкого розповсюдження 5G радіо мереж, інтернету речей, де енергоефективність (низька обчислювальна складність, а отже – висока швидкість обробки даних) буде важливим критерієм застосування, і очікуване підвищення якості зображень та відео в цілому: збільшенні динамічного діапазону, кількості біт на піксель з 8 до 10, 12 та 16, роздільної здатності зображень до 4К, 8К і навіть 16К, що також ставить питання оптимального співвідношення обчислювальна складність/коефіцієнт стиснення.

Існуюча тенденція збільшення складності кодеків зображень та відео обумовлювалась різними факторами, серед яких в першу чергу можна виділити пропускну спроможність каналів ефірного телебачення і мобільного інтернету. Оскільки ці фактори втрачають своє значення при наявності каналів інтернет-зв'язку з високою швидкістю, а технології збереження даних розвиваються краще, ніж збільшення обчислювальної потужності процесорів, поставлена задача виглядає доцільною та актуальною.

Зауважимо ще один аргумент, який не був розглянутий у вступі: збільшення розміру графічних файлів на 20-30% не є критичним на фоні широкого розповсюдження використання даних зі значною надлишковістю, в середньому у 3-5 разів більшою за оптимальне стиснення. Іншими словами, більшість зображень, які генеруються користувачами інтернет-мереж, мають надмірно великий показник якості (особливо це стосується

зображень у форматі JPEG), який можна значно (в рази) зменшити за допомогою програм-оптимізаторів графічних файлів без зменшення візуальної якості. Крім того, відносно формату JPEG не буде спостерігатись збільшення розміру графічного файлу, як мінімум – буде отриманий такий же коефіцієнт стиснення, або трохи більший (на 10-15%, за рахунок використання більш ефективного ентропійного кодування, арифметичного замість Хаффмана та внаслідок використання алгоритмів передбачення значень пікселів).

Тому ще одним аргументом на користь нового графічного формату є необхідність використання прогресивної передачі даних, яка дозволить значно скоротити об'єми переданих зображень в мережі інтернет за рахунок адаптації до конкретних вимог та умов (хоча запропонована реалізація ґрунтується на створенні декількох зменшених копій початкового зображення, тобто обумовлює збільшення об'єму даних на 30-40%).

Ще одним параметром оптимального кодека зображень для радіо мереж виглядає можливість додаткового захисту від завад за рахунок використання завадостійких алгоритмів до найбільш важливої інформації зображення, що дозволить при необхідності відновити частину інформації (до таких важливих даних відноситься службова інформація, статистичні дані та зменшені зображення).

1. 4. Обґрунтування вибору середовища розробки

Враховуючи поставлені завдання, в якості середовища розробки можна було обрати декілька можливих варіантів, в тому числі й С++, на якому й було зупинено вибір, як одному з найпоширеніших і відомих. Програми, написані на С++, працюють ефективніше за інші (лише оптимізований код на асемблері може працювати ще швидше). Крім того, дану мову відрізняють: кроссплатформеність (легкість перенесення програм на інші платформи); мінімальне число ключових слів; більш

строга перевірка типів; безпосередній доступ до пам'яті комп'ютера через використання посилань; передачу параметрів в функцію за значенням, а не за посиланням (передача за посиланням можлива і виконується за допомогою покажчиків).

Висновки до розділу

Кожен з існуючих графічних форматів має певні недоліки, що обґрунтовує необхідність розробки нового графічного формату (або навіть декількох, якщо врахувати вимоги наявності альфа-каналу і анімації).

Коротко про основні недоліки кожного відомого формату.

JPEG – не використовує передбачення значень пікселів (потужного інструменту додаткового стиску без втрат), хоча й використовує простий алгоритм для передбачення DC-коефіцієнтів перетворення, що частково компенсує відсутність передбачення пікселів, крім того – віднімання від значень пікселів числа 128 теж є одним з найпростіших методів передбачення (оскільки 128 – це середина діапазону 0-256); використовує ДКП замість сучасних ЦКП зі швидкими алгоритмами (основне зауваження по обчислювальній складності); погано реалізовано прогресивна передача даних (розмиття та великий обчислювальний ресурс); використовує таблицю для квантування коефіцієнтів, де занадто великі значення коефіцієнту квантування для низьких частот (що призводить до збільшення коефіцієнту стиснення, але робить більш помітними блочні артефакти).

WebP – також використовує ДКП замість ЦКП зі швидкими алгоритмами; занадто низькі параметри вимог до якості зображення (в тому числі – використовує прорідження у форматі 4:2:0); не має прогресивної передачі даних.

PNG, FLIF – використовують прорідження замість перетворення для режиму стиснення з втратами, що неефективно (перетворення краще і

обчислювальна складність його виконання незначна).

HEIF/HEIC, BPG і AVIF – занадто вимогливі до обчислювальних ресурсів; також не мають режиму прогресивної передачі даних.

Новий графічний формат має бути оптимальним по швидкодії, якісно і швидко реалізовувати прогресивну передачу даних, використовувати передбачення для збільшення коефіцієнту стиснення, ЦКП із швидкими алгоритмами для збільшення швидкодії, мати режими з високою якістю зображень, бути адаптованим до різних класів, типів зображення (в тому числі – до схем, таблиць, тексту).

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		22

2. РОЗРОБКА ШВИДКОДІЮЧОГО ЗАХИЩЕНОГО КОДЕКУ ЗОБРАЖЕНЬ ДЛЯ РАДІО МЕРЕЖ

2. 1. Прогресивна передача даних

Розглянемо більш детально основні питання, алгоритми та особливості очікуваного швидкодіючого і захищеного кодексу зображень для радіо мереж.

Як було зазначено у вступі та у першому розділі дипломного проекту, важливим параметром кодексу зображень виступає можливість прогресивної передачі даних. Для цього є декілька підстав, причин, в тому числі зручність для користувачів, яким для попереднього огляду не потрібно в більшості випадків занадто великих параметрів якості та розміру зображення, і навпаки – у яких іноді виникає необхідність більш якісного ніж звичайно зображення. Саме це й надає використання алгоритму прогресивної передачі даних, який реалізується у формі декількох послідовно збільшуваних зображень, причому перша та друга ітерації, етапи змінюють один одного автоматично, а третя, остання (та четверта – при її наявності, опціонально, в залежності від того, наскільки велика роздільна здатність оригіналу) за розміром версії зображення (тобто оригінальне зображення) надаються за запитом, при необхідності.

Відповідно, широке розповсюдження прогресивної передачі зображень дозволить, з одного боку, зменшити об'єми передачі зображень в радіомережах, з іншого боку – дозволить отримувати при необхідності за запитом зображення більшої якості.

Іншими факторами, перевагами прогресивної передачі зображень є технічні, а саме:

- зменшені версії зображень можуть бути використані для алгоритму

передбачення значень пікселів (точніше, в процесі передбачення запропонованим в данній роботі алгоритмом формуються декілька зменшених зображень як побічний ефект);

- зменшені версії зображення дублюють основну інформацію базового зображення, а отже, можуть бути використані для відновлення зображення при частковій втраті частини даних внаслідок завад у радіомережах, тобто виконують функції додаткового захисту даних;
- для алгоритмів розпізнавання образів на зображення використання зменшених зображень для перших ітерацій може значно зменшити обчислювальну складність та прискорити швидкодію.

Зараз прогресивна передача даних реалізована в одному з режимів формату JPG, але використовується рідко, оскільки має недоліки: розмиття зображення, яке виникає внаслідок відображення оригінального зображення з високою роздільною здатністю при наявності дуже малої кількості коефіцієнтів перетворення.

Для наочності приведемо приклад розмиття зображення на рис. 2.1



Рисунок 2.1 – Приклад розмиття перших двох ітерацій зображення при прогресивній передачі даних

Другим недоліком реалізації прогресивного JPG є збільшення в декілька раз обчислювальної складності, оскільки кожна ітерація повторно проходить всі етапи декодування.

Таким чином, прогресивна передача зображень є важливою особливістю запропонованого кодека зображень для застосування в радіо мережах, причому реалізація має низьку обчислювальну складність, а зменшені зображення передаються з високою якістю, і є складовою частиною роботи кодеку зображень (використовуються для передбачення значень пікселів).

З іншого боку, запропонований алгоритм реалізації прогресивної передачі даних при деяких умовах (коли надання за запитом основного зображення буде спостерігатись часто) може привести й до збільшення кількості переданих даних, оскільки два зменшених зображення займають 31% від базового, тобто як мінімум в 1,31 рази розмір файлу буде збільшено, крім того, при застосуванні значного (до 10 раз) зменшення обчислювальної складності кодека зображень буде збільшення розміру файлу ще на 15-20% (потрібні тестування для більш точної оцінки – вони будуть проведені у третьому розділі роботи). Таким чином, запропонований підхід збільшить розміри файлу зображення в 1,5 рази, що може бути компенсованим лише у випадку, коли більше половини користувачів будуть задовольнятися зменшеними в 2-4 рази версіями початкового зображення.

2. 2. Передбачення

Застосовуються два варіанти передбачення:

- 1) передбачення значень пікселів зображення (на підставі значень пікселів із сусідніх блоків);
- 2) передбачення значень коефіцієнтів перетворень (враховуючи значення коефіцієнтів сусідніх блоків).

Перший варіант є більш розповсюдженим, але другий теж використовується, в тому числі в сучасних кодексах зображень, наприклад, таких як JPEG XR та AVIF. Також передбачення значень коефіцієнтів перетворення використовується у форматі JPEG, але в найпростішій реалізації (передбачуються лише DC-коефіцієнти, які є найбільш великими).

Як було зазначено вище, у WebP використовується 10 алгоритмів передбачення [7], у HEIF, кодека зображень на основі H.265, вже 35 алгоритмів [10], а у очікуваного кодека VVC (Versatile video coding) взагалі планується 67 алгоритмів передбачення [15].

Розглянемо детальніше алгоритми передбачення на прикладі HEIF. Найпростішим є так зване DC-передбачення, яке розраховується як середнє арифметичне референтних відліків (тобто тих значень пікселів, на основі яких роблять передбачення). Воно розраховується за формулою:

$$V_{DC} = \left\lfloor \frac{1}{2N} \left(\sum_{x=0}^{N-1} t(x) + \sum_{y=0}^{N-1} l(y) + N \right) \right\rfloor$$

Другим режимом є Planar, який представляє собою лінійну інтерполяцію, як це показано на рис. 2.2 [14].

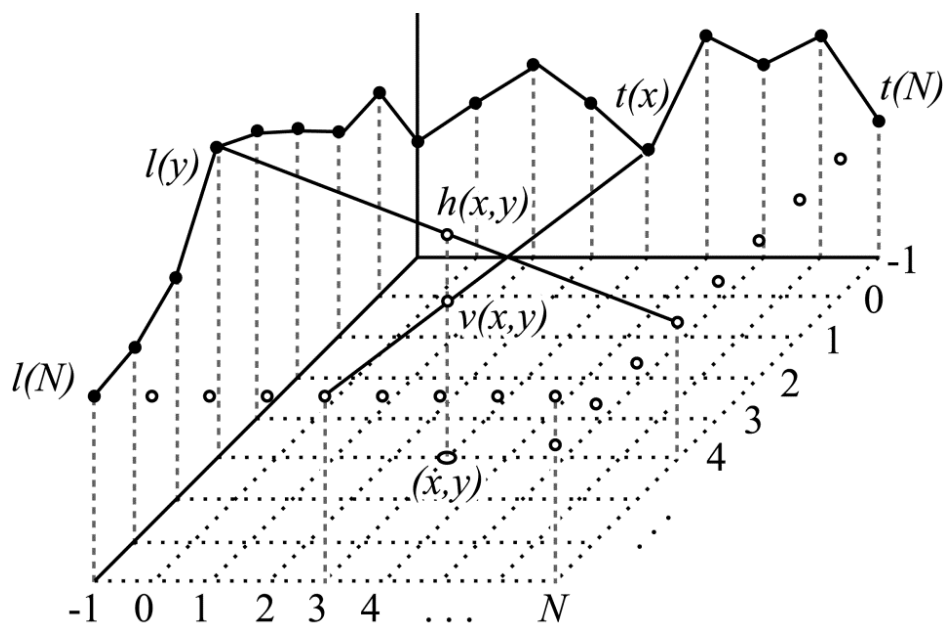


Рисунок 2.2 – Обчислення передбачення-інтерполяції Planar

Інші режими передбачення називаються кутовими, оскільки для передбачення використовують значення референтних пікселів з певного напрямку, як це показано на рис. 2.3 [11].

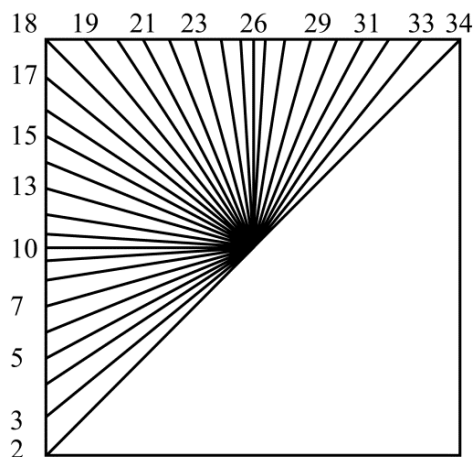


Рисунок 2.3 – Напрями для кутових режимів передбачення

Для майбутнього кодеку VVC планується подвоїти кількість кутових режимів передбачення, що на 3-5% процентів підвищить стиснення, але у два-три рази збільшить обчислювальну складність.

Наскільки обґрунтованим і доцільним є постійне збільшення кількості алгоритмів передбачення? Враховуючи, що існує теоретична межа для цього класу алгоритмів – такий підхід викликає сумніви.

З позиції класифікації блоків зображення та зображень за критерієм кореляції між пікселями можна виділити декілька типів – низької, середньої та високої кореляції. При цьому для блоків і зображень з високою кореляцією будь-які алгоритми передбачення будуть давати високі результати, і навпаки – для блоків і зображень з низькою кореляцією (тобто наявністю границь, різких переходів кольору та яскравості) будь-які алгоритми передбачення мають низьку ефективність, оскільки з’являються значення пікселів, які значно відрізняються від референтних значень попередніх пікселів.

Іншим фактором є складність оцінки та порівняння результатів кожного алгоритму передбачення, тобто алгоритми RDO (rate-distortion optimization), оптимізації співвідношення стиснення/шум.

Таким чином, враховуючи принципове обмеження на максимально можливе стиснення і велику обчислювальну складність відомих алгоритмів передбачення значень пікселів, доцільним виглядає зменшення кількості алгоритмів передбачення до декількох або навіть до одного чи двох.

В якості можливих кандидатів алгоритмів передбачення залишаються наступні: DC-передбачення, Planar (з відомих) та два запропонованих – на основі середнього значення прогнозованого блоку та на підставі статистичних даних. Враховуючи можливість використання середнього значення блоку для формування зменшеного зображення у режимі прогресивної передачі даних кодеком, найбільш перспективним виглядає саме цей варіант. Саме він був реалізований і продемонстрований на рис. 1.3 і 1.4 (для тестових зображень «фото» і «пейзаж» – висококорельованого та низькокорельованого відповідно).

Варіант передбачення значень коефіцієнтів перетворення є альтернативним, тобто доцільним є використання лише одного з алгоритмів передбачення – або пікселів, або коефіцієнтів.

2. 3. Перетворення та коефіцієнт стиснення з втратами

Етап перетворення викликає найбільшу увагу дослідників, оскільки тут спостерігається більше різноманітних варіантів та можливостей, ніж на першому чи третьому етапі стиснення зображень.

Розмірність перетворення.

У найпростішому варіанті, що реалізовано в JPG, використовується лише одне фіксоване перетворення – ДКП розмірності 8.

У найскладнішому варіанті, що застосовано в HEIF, адаптивно

використовуються 4 перетворення ЦКП, розміром від 4x4 до 32x32 [10], і для блоків розмірності 4 додатково до ЦКП застосовується ЦСП (цілочисельне синусне перетворення), причому в різних режимах, вибір якого також складає велику обчислювальну складність, оскільки для кожного рядку та стовпця застосовують всі можливі комбінації з двох перетворень ДКП і ДСП (дискретне косинусне перетворення), до яких потім застосовують складні алгоритми обчислення співвідношення сигнал/шум (а вигаєш від застосування ДСП складає лише 1% додаткового стиснення максимум).

Виникає питання оцінки доцільності адаптивного використання перетворень різної розмірності – наскільки великий додатковий коефіцієнт стиснення отримується за рахунок цього, і наскільки це збільшує загальну кількість обчислювальних операцій (складність).

Приведемо на рис. 2.4 ілюстрацію роботи алгоритму розбиття на блоки, яке використовується в H.265/ HEIF (як для передбачення, так і для вибору розмірності перетворення).

0	1	2	8	25	26	44	45	59	60	63								
	3	4	7															
	5	6																
	9	10	11	14	15	27	28	31	32	37	38	41	46	47	50	51	61	62
		12	13	16	17	29	30	33	34	39	40							
		18	19	35	36	42	43	48	49	52	53							
20	21	22	54	55	58	64	65											
	23	24	56	57														

Рисунок 2.4 – Розбиття на блоки розмірності від 4 до 64 на етапах передбачення і вибору перетворення у кодеку H.265/ HEIF

В роботі [16] було висунуто гіпотезу, що для спрощеного (тобто швидкодіючого) кодека зображень (а також відео) достатньо лише одного

перетворення розмірності 16, оскільки, згідно статистики при дослідженні Н.265/HEIF [11], основними перетвореннями є ЦКП 8x8 і 16x16, а на перетворення розмірності 4 та 32 припадає значно менше випадків застосування.

Тобто, виникає два можливі варіанти: адаптивно використовувати перетворення розмірності 8 і 16 (що призводить до необхідності робити додаткові обчислення співвідношення сигнал/шум для кожного перетворення, і ці обчислення значно складніші, ніж самі перетворення, яких потрібно зробити в рази більше для порівняння), або використати одне з них, при цьому перевага надається перетворенню 16x16, оскільки воно краще стискає, ніж перетворення розмірності 8, крім того, з урахуванням тенденції збільшення роздільної здатності зображень/відео, саме перетворення розмірності 16 буде частіше застосовуватись в адаптивному режимі (внаслідок збільшення кількості блоків, областей з високою кореляцією між пікселями блоку при збільшенні роздільної здатності зображення).

В рамках даної роботи отримані та проаналізовані результати порівняння застосування перетворень різної розмірності до тестових зображень різних типів (з низькою кореляцією – «пейзаж» і високою кореляцією значень пікселів – «фото»).

Попередня гіпотеза щодо необхідності та доцільності адаптивного використання перетворень різної розмірності наступна: адаптивне кодування є більш ефективним, і може використовуватись за умови знаходження більш швидкодіючих за відомі алгоритмів для оцінки, вибору застосування того чи іншого перетворення, в якості якого можуть виступати різні статистичні дані, які автоматично або достатньо просто отримуються в процесі кодування, наприклад – діапазон значень блоку (тобто відстань від мінімальним та максимальним значеннями пікселів блоку).

Обґрунтування даної гіпотези базується на припущенні, що перетворення більшої розмірності дає можливість краще стискати блоки, області зображення з високою кореляцією між пікселями зображення – тобто, з незначними перепадами яскравості та кольорів, а перетворення меншої розмірності – дозволяє краще передавати значні коливання яскравості та кольорів, тобто області з низькою кореляцією.

Задача адаптивного вибору перетворень для кращого стиснення зображень була розглянута в рамках даної роботи, а результати дослідження, тестування більш простих версій алгоритму оцінки вибору перетворення буде проведено в наступному розділі.

Зауваження: з точки зору комплексного підходу, потрібно враховувати інші фактори, які можуть впливати на кінцевий результат. Наприклад, перетворення розмірності 16 або 32 може краще стискати за рахунок кращої акумуляції інформації на один коефіцієнт, ніж перетворення меншої розмірності (4 та 8), але після ентропійного стиснення отриманих коефіцієнтів перетворення співвідношення може трохи змінитись, внаслідок меншого діапазону, розбросу даних у перетворень меншої розмірності (що дозволить краще стиснути їх коефіцієнти). Хоча в цілому, скоріш за все, перевага залишиться – проте буде не настільки значною (оцінки і результати тестування будуть приведені в наступному розділі).

Перетворення з перекриттям.

Частина кодеків використовують не просто блочні перетворення, а блочні перетворення з перекриттям блоків, задля зменшення блочних артефактів – в результаті узгодження відновлених даних між собою (один і той же піксель входить в два блоки, і має два різні, хоча й близькі значення, на підставі яких формується одне).

Розглянемо, наскільки обґрунтовано та доцільно використовувати даний підхід. Для оцінки достатньо провести простий експеримент, тестування – порівняти якість відновленого зображення для двох алгоритмів, враховуючи, що блоки з перекриттям генерують більшу кількість коефіцієнтів, внаслідок більшої кількості блоків для того ж зображення. Отже, потрібно пропорційно збільшити кількість коефіцієнтів для перетворення без перекриття, щоб отримати й порівняти зображення. Зрозуміло, що з точки зору теорії, ми отримаємо приблизно ту ж саму якість й для блочних перетворень без перекриття за рахунок збільшення кількості коефіцієнтів, а з позиції обчислювальної складності – більшу складність не тільки за рахунок збільшення кількості перетворень, а й внаслідок необхідності узгодження значень пікселів. З іншого боку, кодек AVIF використовує саме перетворення з перекриттям [6].

Однорівневі та дворівневі перетворення

Ще одним цікавим питанням для дослідження є застосування ще одного рівня перетворень – вже до коефіцієнтів, отриманих на першому рівні перетворення. Наприклад, такі кодеки, як WebP від компанії Гугл та JPEG XR, розроблений фірмою Майкрософт у 2006 році, використовують дворівневе перетворення (по аналогії з вейвлет-технологією), стискаючи коефіцієнти косинусного перетворення шляхом перетворень Адамара-Уолша (WebP) та Адамара (JPEG XR). Але особливістю даних кодеків є використання перетворень першого рівня малої розмірності – 4x4. Відповідно, коефіцієнти сусідніх блоків будуть мати вищу кореляцію, ніж для перетворень більшої розмірності (8 та 16, не кажучи вже про 32). Саме тому обґрунтованим є застосування другого рівня перетворення, що дозволяє додатково зменшити кореляцію даних. Але при використанні перетворень більшої розмірності – дворівневе перетворення дасть значно

менший рівень додаткового стиснення, і є недоцільним. Відповідно, у нашому кодеку зображень буде використовуватись лише одновимірне перетворення розмірності 8 та 16.

Матриця квантування коефіцієнтів перетворення.

Не тільки JPG, але й більш нові кодеки зображень, наприклад, AVIF, використовують матрицю квантування замість одного значення параметру квантування коефіцієнтів QR. AVIF використовує вісім наборів параметрів квантування, які можна адаптивно вибирати для кожного блоку і макроблоку, що мають індивідуальні параметри для двох площин кольоровості і можуть використовувати просторове прогнозування [6].

Але JPG мав мету отримати максимальне стиснення, тому й було вибрано занадто великі коефіцієнти квантування для низьких частот (що й призвело до посилення блочних ефектів). Для наочного підтвердження наведемо матрицю квантування коефіцієнтів перетворення на рис. 2.5.

Таким чином, враховуючи незначну перевагу матриці квантування перед використанням одного значення QR по якості, але збільшення обчислювальної складності, надамо перевагу останньому підходу, який широко застосовується у різних кодеках.

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

Рисунок 2.5 – Матриця квантування коефіцієнтів перетворення JPG

Цілочисельна апроксимація ДКП, або ДКП.

Питання використання ДКП або ЦКП – одне з найпростіших, оскільки ЦКП має переваги перед ДКП при більшості умов. Отже, обираємо варіант ЦКП як більш прогресивний та швидкодійний.

Швидкі алгоритми виконання ЦКП або часткова факторизація.

Аналогічно, це питання також вирішується однозначно на користь наявності швидких алгоритмів, які дозволяють в декілька раз збільшити швидкодію (за рахунок зменшення кількості операцій, які необхідні для виконання перетворення). Для перетворень різної розмірності будуть різні значення збільшення швидкодії, в діапазоні від 3 до 11 раз [16, 17].

Тип перетворення.

Крім загальноновизнаного лідера серед блочних перетворень – косинусного перетворення (дискретного або цілочисельного) використовуються й інші альтернативи, серед яких синусне перетворення, яке застосовує в тому числі кодек AVIF [6], та спрощені варіанти – як спрощені косинусні, так і різні версії похилих перетворень – слент, ступінчасті, косинусно-синусні та інш. [16, 17]. Відповідно, в роботі перевірені деякі альтернативи ЦКП (косинусно-синусне та спрощене косинусне), хоча це не має великого впливу на кінцевий результат, адже незначне збільшення швидкодії не має особливого значення на загальному фоні.

Паралельні обчислення.

Важливим фактором при розгляді умов та вимог для потенційного кодеку зображень є можливість паралельних обчислень, використовуючи

сучасні багатоядерні процесори. В цілому, це не представляє складності використання, особливо на етапі перетворення.

2. 4. Захист зображень в радіомережах

Враховуючи комплексний підхід до формування вимог та побажань до можливого кодексу зображень, розглянемо таку опцію, як додатковий вбудований захист зображень в радіомережах.

В першу чергу захист потрібен від завад, які при певних умовах можуть коливатись в значному діапазоні. Відповідно, для оптимального кодексу зображень потрібні швидкодіючі алгоритми завадостійкого кодування, наприклад, такі, як запропоновано в [18]. Враховуючи факт наявності трьох версій зображення, стійкість зображень до передачі в радіо мережах з підвищеним рівнем завад збільшується, оскільки є можливість відновити частину втраченої інформації шляхом порівняння між собою різних версій зображення і корекції. Крім того, запропонований кодек зображень зберігає як частину процесу кодування інформацію про статистичні особливості зображення, які також можливо застосувати для корекції впливу завад.

Іншими варіантами підвищення завадостійкості в радіо мережах є алгоритми кодування з підвищеною стійкістю до завад, наприклад, такі, які розглянуто в [19, 20]. Хоча вони застосовуються централізовано до всього потоку даних, а не до окремих даних, і не мають прямого відношення до теми дослідження, їх швидкодія та ефективність є підставою для розгляду та рекомендацій до використання.

Також представляють цікавість алгоритми протидії несанкціонованій зміні даних, які по ряду причин стають більш відомими та поширеними. В рамках використання подібних алгоритмів опціонально в запропонованому кодексу зображень можна використовувати можливість порівняння декількох версій зображення, тобто додаткового кодування

найменшої з версій та статистичних характеристик зображення з метою зафіксувати неочікувані зміни в зображенні. В рамках концепції збільшення службової інформації можливо також фіксувати параметри користування зображенням (час доступу, операції та інш.)

Висновки до розділу

В даному розділі були дані відповіді на основні питання, які мають відношення до стиснення та підвищення швидкодії кодеку зображень. Частина пов'язаних тем не були розглянуті, серед них – ентропійне кодування, фільтрація, додаткові можливості (анімація, альфа-канал та інш.)

Також не були розглянуті такі питання, як адаптивні алгоритми стиснення різних типів зображень, параметри та критерії визначення статистичних даних зображення та деякі інші вузько спеціалізовані питання.

Що стосується алгоритмів фільтрації, то, з одного боку, вони є важливою частиною кодека, і дозволяють покращувати візуальну якість зображень. З іншого боку – фільтрація вносить певні викривлення, спотворення в оригінальне зображення, створюючи ефект «замилування», розмиваючи деталі та обриси об'єктів. Крім того, фільтрація збільшує загальну обчислювальну складність кодеку. Враховуючи ці аргументи, в запропонованому кодеку зображень використані два альтернативних варіанти, опціонально – без фільтрації та зі спрощеною фільтрацією з простим у обчисленні алгоритмом на кшталт сігма-фільтру.

Щодо ентропійного кодування, яке займає значну частину обчислювальної складності кодеку зображень, то деякі ідеї та гіпотези використання статистичної інформації про зображення для зменшення кількості операцій подальшого ентропійного кодування будуть перевірені в майбутніх дослідженнях. Одним з цікавих алгоритмів для збільшення стиснення, використаним в одній з останніх версій покращення формату

jpg (mozjpeg), є підхід до адаптивного збільшення або зменшення коефіцієнтів перетворення на одиницю для покращення стиснення (щоб додатково збільшити кількість таких значень коефіцієнтів, які зустрічаються найчастіше, за рахунок тих, які відрізняються лише на одиницю). Як не дивно, цей алгоритм не тільки дав на 10% кращий результат стиснення, але й забезпечив покращення стійкості зображення до спотворень.

Основні висновки до даного розділу, які обумовлюють очікувані особливості розробленого кодеку зображень, полягають в наступному:

- прогресивна передача даних (якісна та швидкодіюча);
- дублювання основної інформації зображення на трьох рівнях;
- лише один алгоритм передбачення значень пікселів;
- однорівневі цілочисельні косинусні перетворення з адаптивним вибором розмірності перетворення (8 або 16) зі швидкими алгоритмами виконання;
- прості критерії вибору перетворення і коефіцієнта стиснення (оцінка діапазону значень більшості блоків зображення);
- адаптивний вибір алгоритмів стиснення різних типів зображень;
- додаткові статистичні дані;
- можливість розпаралелювання алгоритмів.

Режими роботи, які включені в базовий профіль запропонованого кодеку зображень: 1) з прогресивною передачею даних (ППД); 2) без ППД; 3) режим стиснення зображень з малою кількістю кольорів (графіки, тексти); 4) режим стиснення зображень з великою кількістю кольорів (фото, пейзажи); 5) режим стиснення з фіксованими параметрами для високої якості (підходить для всіх зображень); 6) режим стиснення з адаптивним вибором розмірності перетворення і коефіцієнтом стиснення (для більшого стиснення).

3. ТЕСТУВАННЯ І ПОРІВНЯННЯ РІЗНИХ КОДЕКІВ ЗОБРАЖЕНЬ

3. 1. Параметри і задачі тестування

В даному розділі розглянуті та проаналізовані експериментальні результати тестування та порівняння різних кодеків зображень, як відомих, так і розроблених демо-версій кодека зображень з різними початковими параметрами.

Метою тестування й порівняння відомих кодеків зображень між собою є отримання оцінок по стисненню та швидкодії як для кодеків в цілому, так і для приблизної оцінки особливостей конкретної реалізації кожного з етапів стиснення зображень у різних кодеках.

Наприклад, у таких найбільш відомих кодеках як WebP [7], JPG [9] та HEIF[4] спостерігається суттєва різниця на всіх етапах: WebP використовує 10 алгоритмів передбачення значень пікселів, HEIF – 35, а JPG – жодного (але частково компенсує це відніманням числа 128 від абсолютних значень пікселів та найпростішим різницеvim алгоритмом передбачення значень DC-коефіцієнтів), JPG використовує в якості алгоритмів ентропійного кодування – кодування довжин серій (run-length encoding, RLE) та потім кодування за Хаффманом, а WebP та HEIF застосовують більш ефективні алгоритми ентропійного кодування (наприклад, арифметичне кодування, яке використане у WebP, на 7-10% краще за Хаффмана [7]).

Також значні відмінності спостерігаються на етапі перетворень: JPG використовує фіксоване перетворення розмірності 8x8 ДКП, HEIF – адаптивно застосовує ЦКП розмірності від 4 до 32 та ЦСП 4x4 (з частковою факторизацією), а WebP – задовольнилось найпростішим перетворення ДКП 4x4, до коефіцієнтів якого додатково застосовується перетворення Адамара-Уолша). Відповідно, маємо дати відповідь на

запитання: наскільки обґрунтована необхідність значного ускладнення архітектури кодека (в три рази більше алгоритмів передбачення та адаптивно обраних перетворень, а також – складні алгоритми фільтрації).

Для тестування іноді беруть зображення у форматі jrg, які переводять у bmp, потім стискають іншими кодеками, й отримують високі результати: на 40-50% чи навіть 80% краще (хоча на сайті WebP [7] вказуються результати лише на 25-35% краще за JPG). Але якщо взяти й повторно стиснути jrg, то це буде вже стиснення іншого зображення, відновленого, яке має трохи інші характеристики кореляції значень пікселів, й його можна додатково стиснути (тобто jrg при першому стисненні втратив частину даних, кореляція збільшилась, й тому повторне стиснення навіть тим самим кодеком jrg покаже додатково 10-20%). Крім того, певну роль грають особливості порівняння результатів стиснення різними кодеками та загальна складність цього питання (використання різних параметрів та характеристик зображення, наприклад, SSIM, structure similarity, тобто індекс структурної подібності, PSNR, пікове співвідношення сигнал-шум, СКВ, середньоквадратичне відхилення, SAD, сума абсолютних різниць значень пікселів, середнє значення похибки на піксель та інші).

З іншого боку, самостійно отримати точні та об'єктивні дані тестування різних кодеків та складових кодеку між собою, наприклад, перетворень різної розмірності (від 4 до 32, які, як відомо, застосовуються зараз у сучасних кодеках зображень та відео) не настільки просте завдання, оскільки існують додаткові умови та фактори, що впливають на кінцевий результат порівняння. Наприклад, в залежності від того, застосовується один параметр QR для квантування всіх коефіцієнтів перетворення чи таблиця квантування (як у JPG та AVIF) – будуть різні результати стиснення. Перетворення більшої розмірності при наявності таблиці квантування буде мати більш високий результат стиснення, ніж при фіксованому QR, оскільки буде мати більше нулів серед коефіцієнтів,

причому в кінці послідовності коефіцієнтів, що добре стискається ентропійним кодеком. Іншим параметром буде вибір форми запису, збереження коефіцієнтів – при використанні послідовного запису та считування за формою «зігзаг» буде краще стиснення, ніж просто у рядок. Коефіцієнт стиснення також впливає на порівняння перетворень різної розмірності – при високій якості збереження зображення різниця між перетвореннями менша, ніж при високих коефіцієнтах стиснення (при яких перетворення більшої розмірності мають перевагу). Також впливає тип зображення, значення кореляції пікселів початкового зображення – при високій кореляції перетворення більшої розмірності мають перевагу, а при низькій кореляції, коли є значні перепади значень в кодованому блоці – кращі перетворення низької розмірності (в тому числі внаслідок того факту, що при розбитті великого блоку на менші не всі блоки меншого розміру мають низьку кореляцію та перепади значень пікселів).

Параметри та умови тестування.

Визначимо основні параметри та умови тестування і порівняння кодеків зображень та їх складових.

По-перше, ми будемо досліджувати та порівнювати лише два відомих кодеки зображень – WebP і JPG, та декілька версій розробленого кодеку. Чому? Тому що кодеки AVIF і HEIF, по-перше, мають занадто великі обчислювальні вимоги та потреби, по-друге, до цього часу не мають доступної специфікації, реалізації для порівняння й тестування (для HEIF це пов'язано також з фактом патентних спорів трьох різних груп, які володіють патентами на нього). Хоча в інтернеті є декілька сайтів, які надають послугу онлайн переводу даних HEIF у інші графічні формати, але можливість для переводу інших графічних форматів у HEIF значно зменшена (а той сайт, який все-таки надає таку послугу – неякісно

переводить у HEIF, оскільки 1) для простих зображень з низькою кореляцією спостерігається зниження, зменшення яскравості фону, а для більш складних зображень отримується такий же коефіцієнт стиснення, як і у WebP, при значно менших обчислювальних ресурсах у WebP).

По-друге, ми будемо аналізувати дані тестування й порівняння як наших досліджень, так і опублікованих й доступних для оцінки.

По-третє, для тестування ми будемо використовувати допоміжне програмне забезпечення, як розроблене в рамках виконання даного дипломного проекту, так і загальновідоме. З метою тестування було розроблене допоміжне ПЗ, яке дозволяє: обчислювати різницю між двома зображеннями (початковим та відновленим), і представляти цю різницю як у вигляді зображення (для наочного сприйняття, оцінки різниць, див. рисунок 1.3 наприклад) та у вигляді середнього значення різниці, похибки на один піксель (сума всіх значень похибки поділена на кількість пікселів).

Задачі тестування.

В першу чергу необхідно отримати відповіді на наступні питання:

- оцінка впливу різних складових кодека на кінцеве стиснення;
- вибір розмірності перетворення та вплив розмірності на стиснення;
- вибір простору кольорів RGB, YcbCr (і різних режимів YcbCr);
- різні типи зображень (або областей зображень) та алгоритми стиснення для них;
- вплив типу перетворення на стиснення і якість відновлення зображення (ступінчастих, слент, косинусно-синусних та інш.);
- альтернативні критерії вибору розмірності перетворення та коефіцієнту стиснення.

Ці питання також розглядалися в попередньому розділі, але на підставі теоретичних досліджень, а в рамках даного розділу очікується експериментальне підтвердження теоретичних висновків та рішень, та порівняння й аналіз різних стратегій стиснення зображень.

Найбільш важливі питання аналізу та тестування розглянуті в окремих підрозділах (розмірність та вибір перетворення, коефіцієнт стиснення з втратами і вплив на загальне стиснення різних алгоритмів), а менш важливі – розглянуті у висновках до розділу, без деталізації.

3. 2. Розмірність та адаптивний вибір перетворення

Розглянемо вплив кореляції між пікселями зображення та пікселями блоку на коефіцієнт стиснення зображень (або відео – оскільки вплив однаковий). На рисунку 3.1. показано результати тестування відеопослідовності PeopleOnStreet (1920x1080, 30 кадрів в сек.), а на рисунку 3.2 – результати тестування іншої відеопослідовності, Traffic.

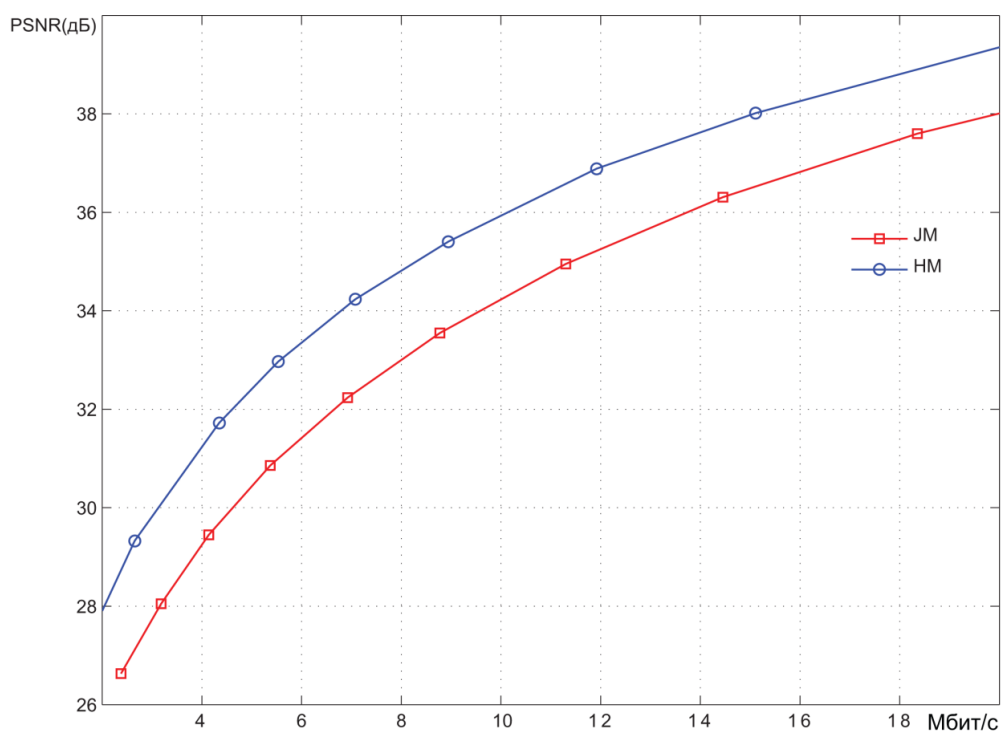


Рисунок 3.1 – Результати тестування відеопослідовності PeopleOnStreet

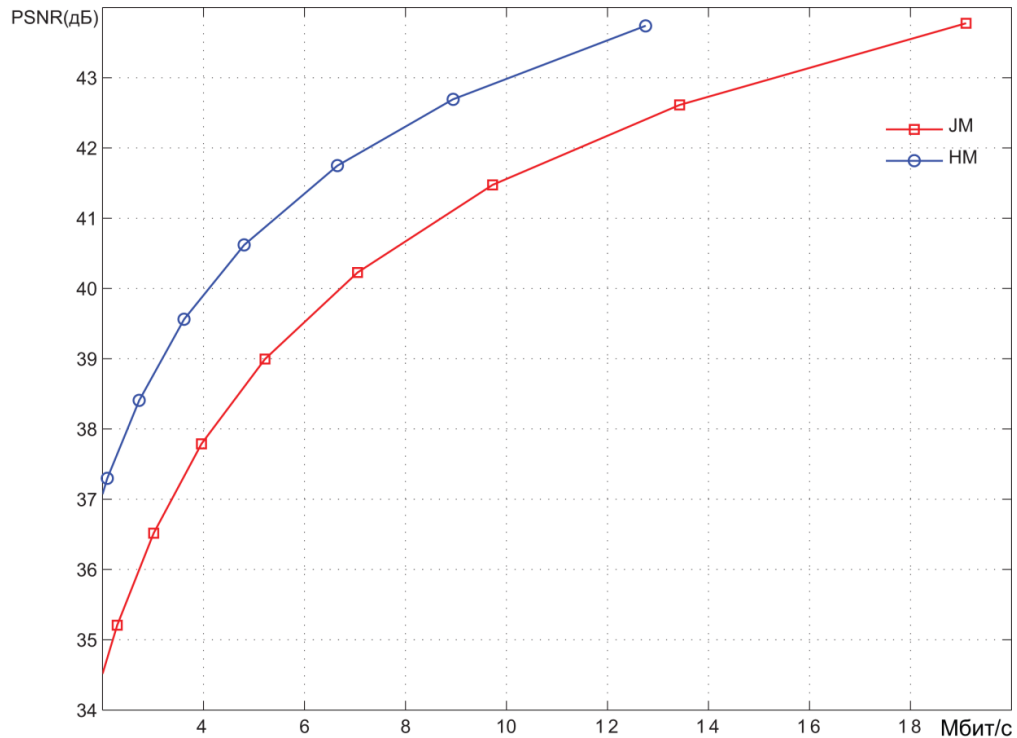


Рисунок 3.2 – Результати тестування відеопослідовності Traffic

Для нас важливо не порівняння H.264 та H.265, що показано на цих рисунках, а дані PSNR при однаковому бітрейті (мбіт/с), або навпаки – кількість мбіт/сек, що потрібно для досягнення певної якості відновленого зображення (тобто коли похибка не більше певного заданого рівня, значення). Відповідно, для досягнення цього заданого рівня якості об'єм даних (коефіцієнтів перетворення) може змінюватись в п'ять-шість раз: як бачимо, 39 дб. на рисунку 3.1 маємо при поточній швидкості 18 мегабіт в секунду, а ті ж самі 39 дб. на рисунку 3.2 отримуємо при швидкості 3 мегабіти в сек., що залежить лише від одного параметру: коефіцієнту кореляції даних, тобто пікселів блоку зображень та відеокадрів. Найбільш висока кореляція – це однакові пікселі, фон, а найбільш низька – штучно згенерований "білий шум", тобто хаотична послідовність значень пікселів (дивись рисунок 3.3)



Рисунок 3.3. – "Білий" шум

Це ілюструє той факт, що дані з низькою кореляцією дуже слабо піддаються стисненню, тому при порівнянні різних перетворень важливо попередньо оцінювати кореляцію даних, що стискаються перетворенням. Адже різниці, що отримуються після процесу передбачення значень пікселів (різниці між абсолютними значеннями та отриманими прогнозованими) мають низьку кореляцію (хоча й не таку низьку, як білий шум), і, відповідно, вони 1) слабо стискаються перетворенням; 2) приблизно однаково стискаються перетвореннями різної розмірності.

Порівнюючи кодеки різних поколінь та різних технологій (наприклад, WebP використовує перетворення розмірності 4, а HEIF – від 4 до 32), ми бачимо, що різниця між ними незначна [6] (дивись рисунок 3.4), хоча це в деякій мірі залежить від вибору системи оцінювання (PSNR, SSIM та інш.).

Відповідно, порівнюючи два кодека, які нас найбільше цікавлять (з точки зору розмірності перетворення та порівняння ефективності перетворень різної розмірності), а саме – WebP та HEIF, ми бачимо, що різниця між ними на графіку несуттєва (і складає приблизно декілька процентів).

З іншого боку, аналогічні результати отримані в процесі безпосереднього тестування розробленого кодеку зображень з різними

початковими перетвореннями, від 4 до 32 (перетворення взяті з [10]).

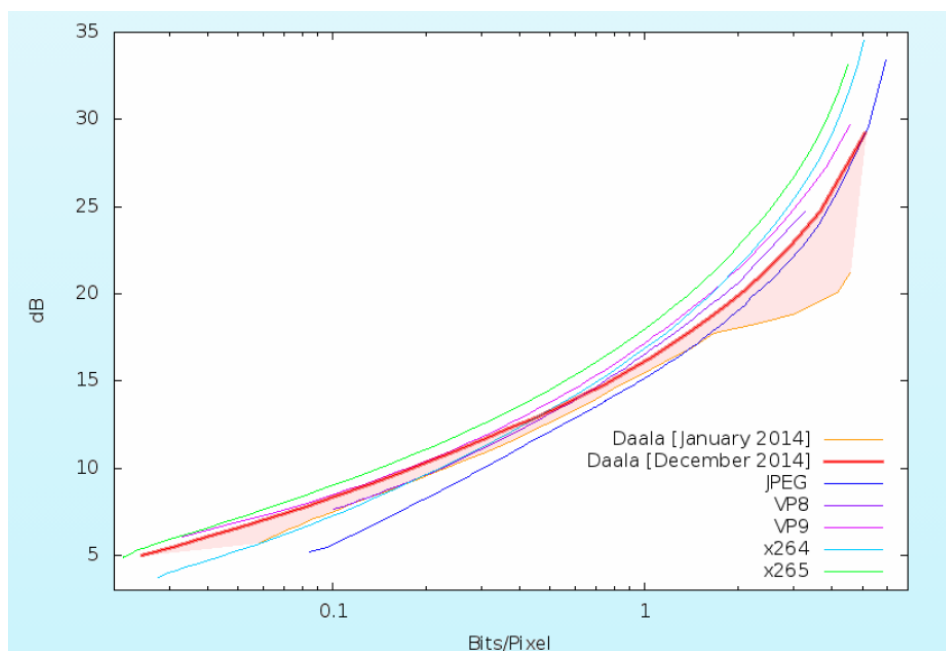


Рисунок 3.4 – Порівняння кодеків зображень

Якщо порівнювати перетворення без попереднього обчислення різниць (тобто без використання передбачення), то результати будуть іншими, оскільки тоді декореляційні властивості перетворень будуть значно кращими (внаслідок високої кореляції перетворюваних даних). При цьому спостерігається типова картина, коли найбільші коефіцієнти перетворення скупчуються у верхньому лівому кутку, і навпаки – у нижньому правому куті матриці коефіцієнтів перетворення спостерігаються найменші коефіцієнти (та нулі). Відповідно, враховуючи такий розклад, розподіл коефіцієнтів, доцільним виглядає алгоритм зчитування коефіцієнтів за схемою "зігзаг" та подібним (дивись рисунок 3.5). Крім того, використання матриці квантування, в якій коефіцієнти в лівому верхньому кутку ділять на значно менший коефіцієнт квантування, ніж у нижньому правому (див. рисунок 2.5 – матриця квантування для JPEG) призводить до того, що в з середини і до кінця послідовності

коефіцієнтів за схемою "зігзаг" спостерігаються нулі, які краще кодуються алгоритмами ентропійного кодування.

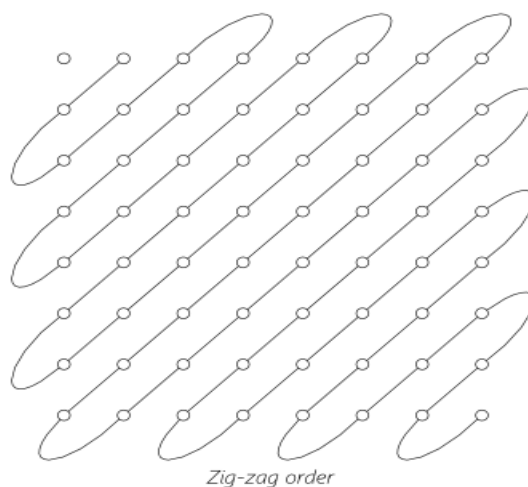


Рисунок 3.5 – Алгоритм сканування коефіцієнтів перетворення

Але коли кодується зображення з низькою кореляцією, в тому числі – різниці після етапу передбачення, то коефіцієнти перетворення розподіляються в матриці коефіцієнтів перетворення хаотично, і використання алгоритму "зігзаг" не дає додаткового стиснення (при умові, що використовується один параметр квантування для всіх коефіцієнтів).

Таким чином, стратегія використання одного перетворення, реалізована в JPEG та пізніше в WebP – досить обгрунтована, оскільки використання перетворень більшої розмірності дає незначний приріст стиснення, і лише на певних ділянках/зонах зображення з високою (або середньою) кореляцією значень пікселів між собою. З іншого боку, враховуючи тенденцію подальшого збільшення роздільної здатності та розміру зображень, і, відповідно, збільшення розмірів і кількості областей зображення з високою і середньою кореляцією, доцільним виглядає й застосування перетворень більших розмірностей при адаптивному виборі розмірності перетворення.

3. 3. Коефіцієнт стиснення з втратами

Враховуючи вищезазначені аргументи та зауваження щодо впливу на коефіцієнт стиснення кореляції початкових даних (значень пікселів), можна зазначити, що при наявності високих вимог до рівня допустимих похибок, відхилень від початкових даних досягнути великих коефіцієнтів стиснення дуже складно, особливо для зображень з низькою кореляцією.

Крім того, очікуване збільшення кількості біт на піксель з 8 до 12 та навіть 16 також впливає на подальше підвищення рівня якості зображень та зменшення рівня стиснення. Причому при використанні до 8-бітного зображення малих коефіцієнтів стиснення отримуємо приблизно таку ж високу якість, як і в результаті середнього стиснення 12-бітного зображення (тобто немає підстав взагалі використовувати 12 біт на піксель для початкового зображення, якщо потім буде використовуватись середній або великий коефіцієнт стиснення – для збереження початкової якості доведеться використовувати лише малі коефіцієнти стиснення).

Виникає питання: а наскільки доцільно взагалі збільшувати розміри зображень та відео до 8К чи 16К та кількість біт на піксель? Експериментально досліджено, що зір людини досягає 120 мегапікселей по роздільній здатності, тому сучасні тенденції ставити відеокамеру на 48 мегапікселей (у смартфонах) досить обґрунтовані (так само як і збільшення кількості кадрів в секунду для відео з 25 до 60 і навіть до 120).

Тестуючи версії розробленого кодеку з перетвореннями різної розмірності на тестових зображеннях «фото» та «пейзаж», отримали наступні результати: при $QP=14$ похибка для всіх перетворень коливалась в межах 2,5-3,2 одиниць яскравості на біт для зображення «пейзаж» і коло 1,1-1,5 для зображення «фото», що близько 1% (від діапазону в 8 біт, тобто 256 значень яскравості) для кольорового простору RGB.

Після переходу від RGB до YcbCr спостерігаються наступні зміни:

збільшення похибки на 5-10% (тобто замість 2,5-3,2 похибка складала 2,7-3,5) при одночасному зменшенні на 5-7% розміру файлу коефіцієнтів (в режимі YcbCr 4:4:4, тобто без прорідження кольорових складових). Відповідно, режим YcbCr 4:2:2, а тим більше режим 4:2:0 взагалі не тестувались – оскільки якщо навіть без прорідження похибка по каналу В, синього кольору складала на 20-30% більше похибки по каналам G (зеленого) та R (червоного), то після прорідження похибка мала б збільшитись ще більше.

При фіксованому QP=24 похибка для перетворень всіх розмірностей коливалась в межах 4,0-5,2 одиниць яскравості на біт для зображення «пейзаж» і коло 1,9-2,3 для зображення «фото», що складає близько 2% від всього діапазону для кольорового простору RGB.

Також було проведене порівняння відомих кодеків WebP та JPEG для цих самих тестових зображень, з виконанням необхідних вимог (початкове зображення було у форматі BMP, відновлене зображення також переводилось в BMP, і обчислювалась середня похибка на піксель). Початкове зображення «пейзаж» мало розмір 620 кб. у форматі JPEG, після переводу в BMP та стиснення JPEG зробили декілька варіантів зображення з різною якістю, і кожне було порівняно з оригіналом (аналогічно – для WebP). В результаті було підтвержені дані, що викладені в [7] – стиснення WebP має на 30% кращий результат, ніж у JPEG.

Зображення у JPEG з розміром файлу 410 кб. мало наступні похибки по каналам RGB: 3,0 – R, 2,6 – G, 4,0 – B, і в середньому – 3,2 одиниць яскравості на піксель. Аналогічні значення похибки для WebP були отримані при розмірі файлу 315 кб. (середнє значення – 3,1 на піксель).

При більшому коефіцієнті стиснення WebP з розміром файлу 225 кб. продемонструвало середню похибку в 5,3 одиниць яскравості на піксель (5,1 – червоний колір, 4,5 – зелений, 6,5 – синій). Як бачимо, YcbCr дає найменшу похибку для каналу зеленого кольору, і найбільшу – для синього

кольору (що обґрунтовано особливостями сприйняття кольорів людським оком). JPEG зображення показало середню похибку в 4,2 на піксель при розмірі зображення в 350 кб., і похибку в 7,4 при 200 кб.

Таким чином, можна підсумувати, що кодеки JPEG і WebP застосовують коефіцієнти стиснення приблизно рівні QR=14 для високої якості і QR=24 для більшого стиснення. Відповідно, для базового режиму розробленого кодеку пропонується використовувати QR=14, і для адаптивного режиму QR=14 і QR=24 в залежності від діапазону перепадів значень пікселів в кодованому блоці.

3. 4. Вплив на загальне стиснення різних алгоритмів

Порівнюючи кодеки JPEG і WebP, можна приблизно оцінити вплив вагу на загальне стиснення окремих етапів, алгоритмів кодування. Найпростіше отримати оцінку покращенню ентропійного кодування – внаслідок використання замість алгоритму кодування Хаффмана алгоритму арифметичного кодування отримується 7-10% додаткового стиснення [7].

Начебто інша частина з 30% переваги WebP складається з впливу алгоритмів передбачення, але при порівнянні двох відновлених зображень (різними кодеками) спостерігається значні різниці в палітрі, і якщо у зображенні, стиснутого JPEG, гістограма близька до оригінального BMP файлу, то у WebP видно зменшення абсолютних значень пікселів та скачкоподібні коливання, переходи (дивись рисунок 3.6).

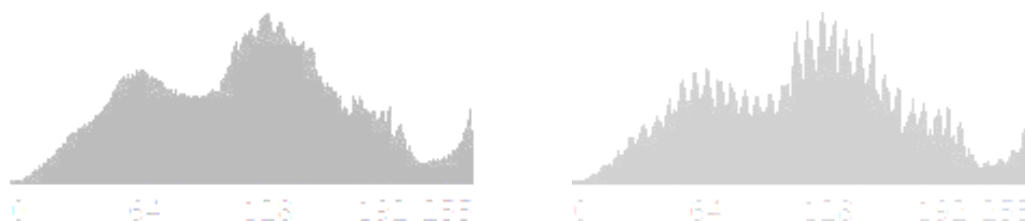


Рисунок 3.6 – Гістограми зображень, стиснутих JPEG і WebP

Таким чином, WebP використовує адаптивну палітру кольорів, яка краще кодується алгоритмами ентропійного кодування за рахунок збільшення частоти одних значень та зменшення частоти інших значень пікселів. У [7] вказано, що WebP використовує адаптивну палітру кольорів, але не зроблені висновки та оцінки про її роль, вплив у збільшенні загального коефіцієнту стиснення зображення, а основна перевага WebP перед JPEG обгрунтовується наявністю алгоритмів передбачення.

Враховуючи, що після передбачення на етап кодування перетворенням попадають дані зі значно меншою кореляцією, які гірше піддаються стисненню, можемо зробити висновок, що адаптивна палітра кольорів вносить приблизно таку ж частку у збільшення стиснення, як і алгоритми передбачення значень пікселів, тобто коло 10-12%.

Таким чином, отримуємо наступні результати оцінки впливу різних алгоритмів стиснення на загальне стиснення: алгоритми передбачення – коло 10%, алгоритми формування адаптивної палітри – коло 10%, алгоритми ентропійного кодування – 7-10%, інші алгоритми – 3-5%.

Висновки до розділу

З метою тестування різних кодеків (відомих і розроблених тестових версій) було розроблено допоміжне програмне забезпечення для оцінки різниці між оригінальним та відновленими зображеннями.

В якості тестових версій були вибрані, програмно реалізовані та протестовані наступні варіанти кодеків:

- Без алгоритму передбачення значень пікселів (аналог JPEG);
- З алгоритмом передбаченням значень пікселів (три варіанти: один відомий і два запропонованих);
- Без алгоритму прогресивної передачі даних;
- З запропонованим алгоритмом прогресивної передачі даних;
- З фіксованими розмірностями перетворень (чотири версії для

чотирьох перетворень розмірностей 4, 8, 16 і 32);

- З адаптивним вибором перетворень 8 і 16;
- З фіксованим вибором коефіцієнту квантування (QP=14,24);
- З адаптивним вибором коефіцієнту квантування;
- Без алгоритму зчитування блоку коефіцієнтів «зигзаг»;
- З алгоритмом зчитування блоку коефіцієнтів «зигзаг»;
- З використанням простору RGB;
- З використанням простору YcbCr у режимі 4:4:4;
- З алгоритмами 1D і 2D перетворення даних.

В результаті тестування, аналізу отриманих результатів і порівняння з відомими результатами тестувань зроблені наступні висновки:

Перехід від простору RGB до YcbCr у режимі 4:4:4 дає 5-10% стиснення за рахунок звуження діапазону кольорів і 5-7% похибки.

В просторі RGB у каналі В (синій колір) спостерігається найменша похибка відновлення з усіх трьох каналів, а при переході до YcbCr вона стає більшою за інші похибки на 20-30%.

Інші режими YcbCr дають значно більшу похибку кольорів, і поступово вийдуть з обігу після переходу до 12 біт на піксель.

Алгоритм зчитування блоку коефіцієнтів «зигзаг» має сенс при використанні матриці квантування коефіцієнтів з різною вагою для низьких і високих частот.

Вплив розмірності перетворення на загальний коефіцієнт стиснення зображення незначний (2-3%), внаслідок значного зменшення кореляції даних після етапу передбачення.

Використання одного універсального алгоритму передбачення дає додаткове стиснення на 10-15% (в залежності від типу даних).

Використання адаптивної палітри є перспективним алгоритмом стиснення зображень на 10-15% (в залежності від типу даних).

Зменшення коефіцієнту стиснення та підвищення якості зображень зменшить похибки кодування і потребу у фільтрації.

Використання алгоритму 1D дає на 2-3% менше стиснення, але у 2 рази зменшує кількість операцій для обчислення перетворення.

Для вибору адаптивного коефіцієнта стиснення можна використовувати спрощений критерій: розмір діапазону значень.

					ІАЛЦ.467200.004 ПЗ	Лист
Зм	Лист	№ докум.	Підп.	Дата		52

ВИСНОВОК

Проведено огляд існуючих форматів графічних даних, аналіз відомих алгоритмів стиснення зображень та вибір найбільш ефективних за швидкістю для високошвидкісного кодека зображень.

Розглянуто, протестовано та проаналізовано альтернативні алгоритми та підходи стиснення зображень.

Проведено теоретичні та експериментальні дослідження й порівняння окремих складових загального коефіцієнта стиснення зображень.

Запропоновано та обгрунтовано основні параметри та складові швидкодіючого кодеку зображень.

Запропоновано новий формат прогресивної передачі зображень, що не має недоліків існуючих версій прогресивного кодеку зображень (розмитість першої ітерації зображення, надмірна обчислювальна складність).

Запропоновано й протестовано два нових алгоритма передбачення значень пікселів, один з яких органічно поєднується з новим алгоритмом прогресивної передачі даних.

Розроблено програмне забезпечення нового кодеку зображень, оптимального по швидкодії.

Розроблені адаптивні режими для стиснення різних типів зображень та різних типів областей зображень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Москва: Техносфера, 2005. 1072 с.
2. Сэломон Д. Сжатие данных, изображений и звука. М.: Техносфера, 2004. 368 с.
3. ISO/IEC 23008-12:2017 – Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 12: Image File Format. URL: <https://www.iso.org/standard/66067.html>
4. HEIF – High Efficiency Image File Format. URL: <https://nokiatech.github.io/heif/>
5. BPG Image Format. URL: <https://bellard.org/bpg/>
6. AV1 Image File Format (AVIF). URL: <https://aomediacodec.github.io/av1-avif/>
7. A new image format for the Web | WebP | Google Developers. URL: <https://developers.google.com/speed/webp/>
8. FLIF – Free Lossless Image Format. URL: <http://flif.info/>
9. JPEG. URL: <https://jpeg.org>
10. ITU-T Rec. H.265/ISO/IEC 23008-2: 2013. Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency Video Coding, 2013
11. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.
12. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)", IEEE Trans.

Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1669-1684, Dec. 2012.

13. F. Bossen, B. Bross, K. Sühring, and D. Flynn, "HEVC Complexity and Implementation Analysis," IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012.
14. Ричардсон Ян. Видеокодирование. H.264 и MPEG-4 – стандарты нового поколения. Москва.–Техносфера.–2005.– 368 с.
15. Versatile Video Coding (VVC) | JVET. URL: <https://jvet.hhi.fraunhofer.de/>
16. Луц В.К. Просте цілочисельне косинусне перетворення порядку 16 для кодування сигналів і зображень. Штучний інтелект. 2014. № 4. С. 31–43
17. Гнатив Л.А. Целочисленные косинусные преобразования: методы построения новых быстрых преобразований порядка 8,16 и их применение. Кибернетика и системный анализ. 2014. Том 50, №6. С. 104-121.
18. Гнатив Л. А., Кожуховский А. Д., Литвин А. И. Алгоритмы быстрых преобразований для кодов Ридда-Маллера и Хэмминга. Автоматика. 1990. № 4. С. 74-77.
19. Шевчук Б.М. Завадостійка передача захищених пакетів даних в інформаційно-ефективних радіомережах. Математичне та комп'ютерне моделювання. Серія: Технічні науки. 2017. Вип. 15. С. 255 – 261.
20. Шевчук Б.М. Эффективные по быстродействию алгоритмы передачи и приема высокоинформативных пакетов в радиосетях. Кибернетика и системный анализ. 2016. Т. 52. № 2. С. 174 – 183.