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 перший (бакалаврський)
Дослідження алгоритмів керування швидкістю при стисненні зображень
(тема)

Виконав:
здобувач 4 року навчання,
групи ТРІМІ-21-1
Молотков Євгеній
(власне ім'я, прізвище)

Спеціальність 172 Телекомунікації та радіотехніка
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Інформаційно-мережна інженерія
(повна назва освітньої програми)

Керівник ст. викл. Олексій Федоров
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ІМІ _____ Валерій Безрук
(підпис) (власне ім'я, прізвище)

2025 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____ / *Молотков Є.С.* /
(підпис) (прізвище та ініціали)

Керівник _____ / *Федоров О.В* /
(підпис) (прізвище та ініціали)

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій

Кафедра Інформаційно мережної інжерепії

Рівень вищої освіти перший (бакалаврський)

Спеціальність 172 Телекомунікації та Радіотехніка
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Молоткову Євгенію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження алгоритмів керування швидкістю при стисненні зображень

затверджена наказом університету від 23 травня 2025 р. № 410Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 24 червня 2025 р.

3. Вихідні дані до роботи Дослідити роботу алгоритмів керування швидкістю при стисненні зображень. Особливу увагу приділити залежності бітової їдкості від співвідношення кількості нульових ДКП коефіцієнтів після квантування. Провести експериментальні дослідження з використанням тестових зображень.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1 Загальні відомості про формат JPEG
2 Особливості сприйняття кольорів людиною
3 Стиснення зображень із втратами
4 Критерії якості зображень
5 Ентропія та ефективна швидкість передавання інформації

___6 Керування швидкістю при стисненні зображень_____
___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 MS Power Point_____
_____
_____
_____
_____
_____
_____
_____
_____

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1.	Ознайомлення із завданням. Уточнення ТЗ.	26.05.2025	вик.
2.	Підбір літератури за темою роботи	26.05 – 30.05.2025	вик.
3.	Виконання розділу 1	28.05 – 05.06.2025	вик.
4.	Виконання розділу 2	30.05 – 9.06.2025	вик.
5.	Виконання розділу 3	07.06 – 15.06.2025	вик.
6.	Виконання розділу 4	09.06 – 13.06.2025	вик.
7.	Виконання розділу 5	10.06 – 12.06.2025	вик.
8.	Виконання розділу 6	02.06 – 17.06.2025	вик.
9.	Оформлення презентаційного матеріалу,	18.06 – 23.06.2025	вик.
	підготовка до захисту в ЕК		

Дата видачі завдання _26_ ___травня____2025р.

Здобувач _____
(підпис)

Керівник роботи _____ __ст. викл. Олексій Федоров_
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 72 с., 12 рис., 8 таб., 23 джерел, 1 додаток.

Об’єктом дослідження є алгоритми керування швидкістю при стисненні зображень.

Мета кваліфікаційної роботи – дослідження алгоритмів керування швидкістю при стисненні зображень. Виявлення зв’язку між кількістю квантованих до нуля ДКП коефіцієнтів і бітовою швидкістю зображення.

В роботі демонструється той факт, що в будь-якій типовій системі кодування з перетворенням існує лінійна залежність між швидкістю кодування та процентним вмістом нулів серед квантованих коефіцієнтів перетворення.

Лінійна модель джерела, доповнена адаптивною схемою оцінки коефіцієнта пропорційності, призводить до уніфікованого алгоритму управління швидкістю для різних стандартних систем фото і відеокодування, таких як JPEG, MPEG-2, H.263, MPEG-4 та багатьох інших.

ЛІНІЙНА МОДЕЛЬ, ШВИДКІСТЬ, УПРАВЛІННЯ, СПОТВОРЕННЯ, КОМПРЕССІЯ ЯКІСТЬ, КОДУВАННЯ, ВІДЕОКОДУВАННЯ

ABSTRACT

Explanatory note: 72 p., 12 fig., 8 tabs, 23 sources, 1 app.

Object of study is algorithms to control the speed of image compression.

Purpose of the qualification work is to study speed control algorithms for image compression. Determine the relationship between the number of the DCT coefficients quantized to zero and the bit rate of the image.

It is demonstrated that in any typical transform coding system there is a linear relationship between the coding rate and the percentage of zeros among the quantized transform coefficients.

The linear source model, complemented by an adaptive scheme to estimate the slope coefficient, leads to a unified rate control algorithm for various standard photo and video coding systems such as JPEG, MPEG-2, H.263, MPEG-4, and many others.

LINEAR MODEL, RATE, CONTROL, DISTORTION, QUALITY, COMPRESSION, CODING, VIDEO CODING

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 8
ВСТУП 9
1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ФОРМАТ JPEG 10
2 ОСОБЛИВОСТІ СПРИЙНЯТТЯ КОЛЬОРІВ ЛЮДИНОЮ 18
2.1 Яскравість 18
2.2 Зорова система людини 20
3 СТИСНЕННЯ ЗОБРАЖЕНЬ ІЗ ВТРАТАМИ 23
3.2 Квантування 24
3.3 Кодування 28
4 КРИТЕРІЇ ЯКОСТІ ЗОБРАЖЕНЬ 35
4.1 Критерій візуального сприйняття 35
4.2 Середньоквадратичний критерій 36
4.3 Критерій максимальної помилки (рівномірного наближення) 38
4.4 Імовірнісно-зональний критерій 38
4.5 Похибки дискретного подання зображень 40
5 ЕНТРОПІЯ ТА ЕФЕКТИВНА ШВИДКІСТЬ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ 45
5.1 Ентропія 45
5.2 Невизначеність та ефективна швидкість передавання інформації 47
6 КЕРУВАННЯ ШВИДКІСТЮ ПРИ СТИСНЕННІ ЗОБРАЖЕНЬ 50
6.1 Загальні міркування. Алгоритми керування швидкістю 50
6.2 Лінійна модель джерела та схема керування швидкістю 54
6.3 Аналіз швидкості в –області 55
ВИСНОВКИ 58
ПЕРЕЛІК ПОСИЛАНЬ 59
ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ 62

ПЕРЕЛІК СКОРОЧЕНЬ

ДКП – дискретне косинусне перетворення

ДПФ – дискретне перетворення Фур’є

QC - quantization coefficient

RLE - run-length encoding

SPD - spectral power distribution

CIE - Commission Internationale de l’Éclairage

EoB - End of Block

DC – direct current

AC – alternating current

ВСТУП

Вихідний бітрейт (швидкість) відеокодера значно змінюється з часом через активність у сцені. У візуальному зв'язку через вузькосмугові канали або канали зі змінними в часі характеристиками керування швидкістю є дуже важливим для забезпечення успішної передачі закодованих відеоданих через канали зв'язку. В системах відеозв'язку в реальному часі, таких як відеоконференції, відеотелефонія, інтерактивні класи та веб-трансляції в реальному часі, наскрізна затримка передачі відео має бути дуже малою, що вимагає більш точного та надійного керування швидкістю.

В даній кваліфікаційній роботі розглядаються більш спеціалізовані по відношенню до задач відео компресії питання управління керування швидкістю при стисненні зображень. З іншого боку, оскільки відеопослідовності являють собою низку окремих зображень (кадрів) розглянуті методи можуть бути застосовані і в задачах стиснення відеоданих.

1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ФОРМАТ JPEG

JPEG — це складний метод стиснення з втратами або без втрат для кольорових або монохромних нерухомих зображень (не відео). Він не дуже добре обробляє двоколірні (чорно-білі) зображення. Також він найкраще працює із зображеннями з неперервними тонами, де сусідні пікселі мають схожі кольори. Важливою особливістю JPEG є використання багатьох параметрів, що дозволяє користувачеві регулювати обсяг даних, що втрачаються (а отже, і коефіцієнт стиснення) у дуже широкому діапазоні. Часто око не може помітити погіршення якості зображення навіть при коефіцієнтах стиснення 10 або 20. Існує два режими роботи: з втратами (він є базовим) і без втрат (який зазвичай дає коефіцієнт стиснення близько 0,5, тобто зменшення розміру вдвічі). Більшість реалізацій підтримують лише режим з втратами. Цей режим включає прогресивне та ієрархічне кодування [1, 2, 5].

JPEG метод стиснення, а не повний стандарт для представлення зображень. Саме тому він не визначає такі характеристики зображення, як співвідношення сторін пікселів, колірний простір або чергування рядків растрового зображення [5].

JPEG був розроблений як метод стиснення для зображень з неперервним тоном. Основні хфпфктеристики стиснення JPEG такі [5]:

1. Високі коефіцієнти стиснення, особливо в випадках, коли якість зображення оцінюється як дуже хороша або відмінна.

2. Використання багатьох параметрів, що дозволяє досвідченим користувачам експериментувати та досягати бажаного компромісу між стисненням та якістю.

3. Отримання хороших результатів з будь-яким типом зображення з неперервним тоном, незалежно від розмірів зображення, колірних

просторів, співвідношення сторін (зазвичай розміри вказуються в пікселях) або інших характеристик зображення.

4. Складний, але не надто комплексний метод стиснення, що дозволяє створювати програмні та апаратні реалізації на багатьох платформах.

5. Кілька режимів роботи [5]:

а) послідовний режим, в якому кожен компонент зображення (колір) стискається за один прохід зліва направо, зверху вниз;

б) прогресивний режим, в якому зображення стискається в декількох блоках (відомих як «проходи»), щоб переглядати його від грубих до дрібних деталей;

в) режим без втрат, який є важливим у випадках, коли користувач вирішує, що пікселі не повинні втрачатися (компромісом є низький коефіцієнт стиснення порівняно з режимами зі втратами); та

г) ієрархічний режим, в якому зображення стискається з кількома роздільними здатностями, що дозволяє переглядати блоки з нижчою роздільною здатністю без попереднього розпакування наступних блоків з вищою роздільною здатністю.

Назва JPEG є акронімом, що означає Joint Photographic Experts Group (Спільна група експертів з фотографії). Це була спільна робота CCITT (International Consultative Committee for Telegraphy and Telephony, Міжнародного консультативного комітету з телеграфії та телефонії) та ISO (the International Standards Organization, Міжнародної організації зі стандартизації), яка розпочалася в червні 1987 року і завершилася створенням першого проекту пропозиції JPEG у 1991 році. Стандарт JPEG виявився успішним і став широко використовуватися для стиснення зображень, особливо на веб-сторінках [5].

Основні етапи стиснення JPEG наведені нижче, і кожен крок детально описаний далі [5].

1. Кольорові зображення перетворюються з RGB у колірний простір яскравості/колірність (цей етап пропускається для монохромних зображень). Око чутливе до невеликих змін яскравості, але не до змін хроматичності, тому частина хроматичності може пізніше втратити багато даних і, таким чином, бути сильно стиснута без значного погіршення загальної якості зображення. Цей крок є необов'язковим, але важливим, оскільки решта алгоритму працює з кожним кольоровим компонентом окремо. Без перетворення колірного простору жоден з трьох кольорових компонентів не витримає значних втрат, що призведе до погіршення якості стисненого зображення [5].

2. Кольорові зображення піддаються пониженню роздільної здатності шляхом створення пікселів з низькою роздільною здатністю з оригінальних (цей крок використовується тільки при виборі ієрархічного стиснення; він завжди пропускається для зображень у відтінках сірого). Пониження роздільної здатності не проводиться для компонента яскравості. Зменшення розміру зображення виконується або у співвідношенні 2:1 як по горизонталі, так і по вертикалі (так зване 2h2v або 4:1:1 дискретизація), або у співвідношенні 2:1 по горизонталі та 1:1 по вертикалі (2h1v або 4:2:2 дискретизація). Оскільки це робиться для двох із трьох колірних компонентів, 2h2v зменшує зображення до $1/3 + (2/3)\cdot(1/4) = 1/2$ його оригінального розміру, а 2h1v зменшує його до $1/3 + (2/3)\cdot(1/2) = 2/3$ його оригінального розміру. Оскільки компонент яскравості не змінюється, помітної втрати якості зображення не відбувається. Зображення в відтінках сірого не проходять цей крок [5].

3. Пікселі кожного колірного компонента організовані в групи по 8x8 пікселів, які називаються одиницями даних, і кожна одиниця даних стискається окремо. Якщо кількість рядків або стовпців зображення не кратна 8, нижній рядок і крайній правий стовпець дублюються стільки разів, скільки необхідно. У режимі без чергування кодер обробляє всі одиниці даних першого компонента зображення, потім одиниці даних

другого компонента і, нарешті, одиниці даних третього компонента. У режимі чергування кодер обробляє три верхні ліві одиниці даних трьох компонентів зображення, потім три одиниці даних праворуч від них і так далі. Той факт, що кожна одиниця даних стискається окремо, є одним із недоліків JPEG. Якщо користувач вимагає максимального стиснення, розпаковане зображення може мати артефакти у вигляді кластерів блоків через відмінності між блоками [4].

4. Потім до кожного блоку даних застосовується дискретне косинусне перетворення (ДКП) для створення відображення частотних компонентів розміром 8x8. Вони представляють середнє значення пікселя та послідовні зміни вищих частот у групі. Це готує дані зображення до вирішального кроку втрати інформації. Оскільки ДКП включає трансцендентну функцію косинус, воно повинно включати певну втрату інформації через обмежену точність комп'ютерної арифметики. Це означає, що навіть без основного кроку з втратами (крок 5 нижче) відбудеться певна втрата якості зображення, але зазвичай вона невелика [5].

5. Кожен із 64 частотних компонентів у блоці даних ділиться на окреме число, яке називається його коефіцієнтом квантування (QC), а потім округлюється до цілого числа. Саме тут інформація безповоротно втрачається. Великі QC викликають більше втрат, тому високочастотні компоненти зазвичай мають більші QC. Кожен із 64 QC є параметром JPEG і, в принципі, може бути визначений користувачем. На практиці більшість реалізацій JPEG використовують таблиці контролю якості (QC), рекомендовані стандартом JPEG для компонентів яскравості та кольоровості зображення [1 - 3, 5].

6. Далі 64 квантовані частотні коефіцієнти (які тепер є цілими числами) кожного блоку даних кодуються за допомогою комбінації кодування RLE та кодування Хаффмана. Замість кодування Хаффмана можна використовувати варіант арифметичного кодування [5].

7. На останньому кроці додаються заголовки та всі необхідні параметри JPEG, а також виводиться результат. Стиснутий файл може бути в одному з трьох форматів [5]:

1) формат *обміну (interchange)*, в якому файл містить стиснуте зображення та всі таблиці, необхідні декодеру (здебільшого таблиці квантування та таблиці кодів Хаффмана);

2) *скорочений (abbreviated)* формат для стиснутих даних зображення, де файл містить стиснуте зображення і може не містити таблиць (або лише кілька таблиць), та

3) *скорочений* (*abbreviated)* формат для даних специфікації таблиці, де файл містить лише таблиці та не містить стиснутого зображення. Другий формат має сенс у випадках, коли використовується та сама пара кодер/декодер, і вони мають однакові вбудовані таблиці. Третій формат використовується у випадках, коли багато зображень було стиснуто одним і тим самим кодером з використанням тих самих таблиць. Коли ці зображення потрібно розпакувати, вони надсилаються до декодера, якому передує один файл із даними специфікації таблиці [5].

Декодер JPEG виконує зворотні кроки. (Таким чином, JPEG є симетричним методом стиснення.)

Прогресивний режим є варіантом JPEG. У цьому режимі коефіцієнти ДКП з вищою частотою записуються у стиснутий потік блоками, які називаються «сканами». Кожне сканування, яке зчитується та обробляється декодером, призводить до чіткішого зображення. Ідея полягає в тому, щоб використовувати перші кілька сканувань для швидкого створення низькоякісного, розмитого попереднього перегляду зображення, а потім або ввести решту сканувань, або зупинити процес і відхилити зображення. Компроміс полягає в тому, що кодер повинен зберегти всі коефіцієнти всіх блоків даних у буфері пам'яті, перш ніж вони

будуть надіслані в сканах, а також пройти всі кроки для кожного сканування, уповільнюючи прогресивний режим [5].

На рисунку 1a показано приклад зображення з роздільною здатністю 1024x512. Зображення поділено на 128 x 64 = 8192 блоки даних, і кожен з них обробляється за допомогою ДКП, перетворюючись на набір із 64 8-бітних чисел. На рисунку 1b показано блок, глибина якого відповідає 8 192 одиницям даних, висота якого відповідає 64 коефіцієнтам ДКП (коефіцієнт постійного струму – верхній, пронумерований як 0), а ширина якого відповідає восьми бітам кожного коефіцієнта [5].

Після підготовки всіх блоків даних у буфері пам'яті кодер записує їх у стиснутий потік одним із двох методів: спектральним вибором або послідовним наближенням (рис. 1c,d). Перше сканування в будь-якому методі – це набір коефіцієнтів постійного струму. Якщо використовується спектральний вибір, кожни наступний проход складається з кількох послідовних (діапазону) AC-коефіцієнтів. Якщо використовується послідовне наближення, друге сканування складається з чотирьох найбільш значущих бітів усіх коефіцієнтів змінного струму, і кожне з наступних чотирьох сканувань, номери від 3 до 6, додає ще один значущий біт (біти від 3 до 0 відповідно) [5].

В ієрархічному режимі кодер зберігає зображення кілька разів у вихідному потоці з різною роздільною здатністю. Однак кожна частина з високою роздільною здатністю використовує інформацію з частин вихідного потоку з низькою роздільною здатністю, тому загальний обсяг інформації менший, ніж той, що потрібен для окремого зберігання різних роздільних здатностей. Кожна ієрархічна частина може використовувати прогресивний режим [5].

Ієрархічний режим корисний у випадках, коли зображення з високою роздільною здатністю потрібно вивести з низькою роздільною здатністю. Старі матричні принтери можуть бути гарним прикладом пристрою виводу з низькою роздільною здатністю, який досі іноді використовується [5].

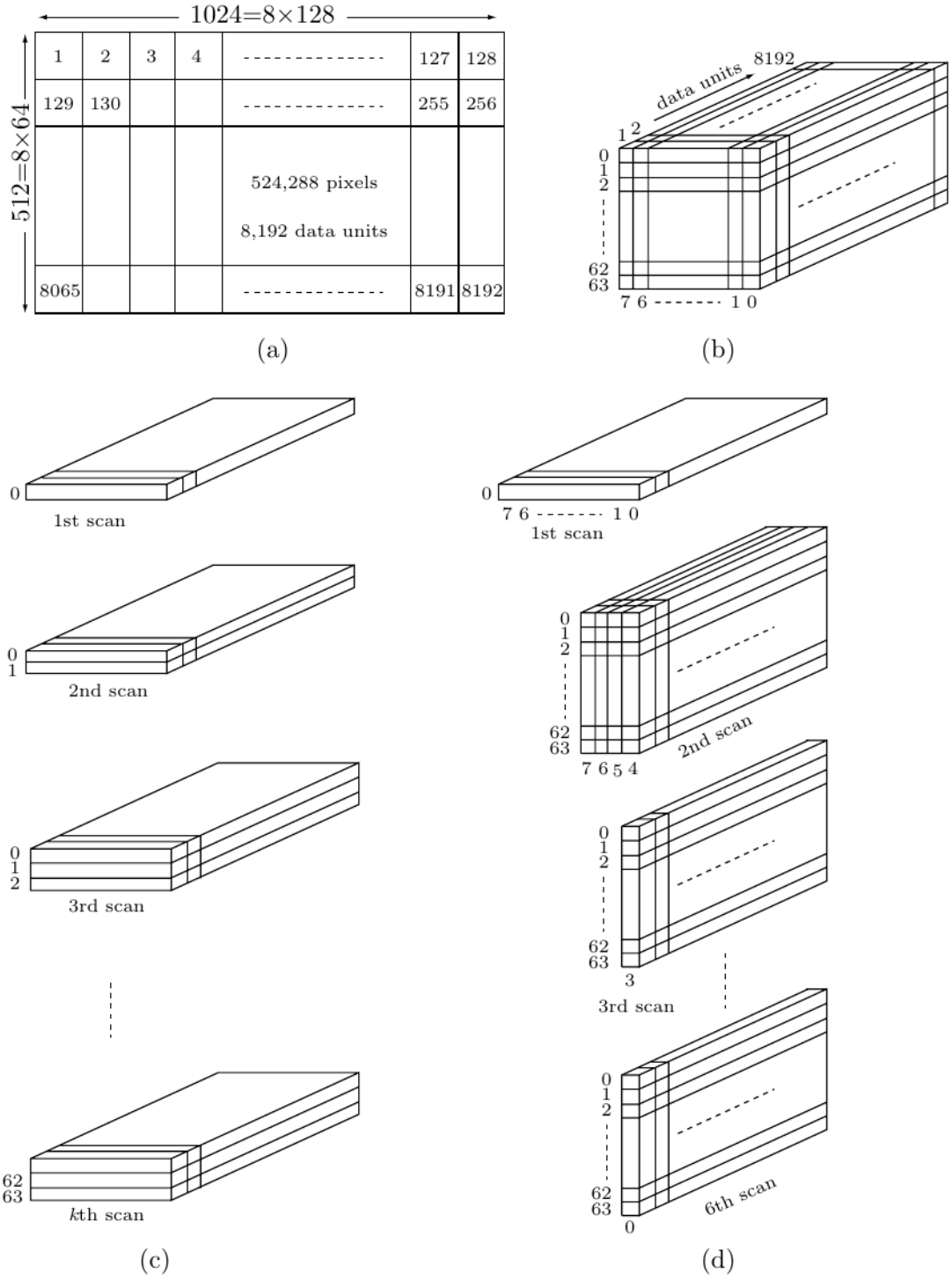


Рисунок 1.1 – Сканування в прогресивному режимі JPEG

Режим без втрат JPEG обчислює «прогнозоване» значення для кожного пікселя, генерує різницю між пікселем та його прогнозованим значенням і кодує різницю, використовуючи той самий метод (тобто кодування Хаффмана або арифметичне кодування), що використовується

на кроці 5 вище [5]. Прогнозоване значення обчислюється з використанням значень пікселів вище та ліворуч від поточного пікселя (пікселі, які вже були введені та закодовані). У наступних розділах ці кроки обговорюються більш детально.

2 ОСОБЛИВОСТІ СПРИЙНЯТТЯ КОЛЬОРІВ ЛЮДИНОЮ

2.1 Яскравість

Основною міжнародною організацією, яка займається світлом і кольором, є Міжнародний комітет з освітлення (Commission Internationale de l'Éclairage), скорочено CIE. Він відповідає за розробку стандартів та визначень у цій галузі. Одним із ранніх досягнень CIE була його діаграма кольоровості [3], розроблена в 1931 році. Вона показує, що для визначення кольору потрібні не менше трьох параметрів. Вираження певного кольору триплетом (x, y, z) подібне до позначення точки в тривимірному просторі, звідси й термін «кольоровий простір». Найпоширенішим колірним простором є RGB, де три параметри – це інтенсивність червоного, зеленого та синього у кольорі. При використанні в комп'ютерах ці параметри зазвичай знаходяться в діапазоні 0–255 (8 біт) [5].

CIE визначає колір як результат сприйняття світла у видимій області спектру з довжиною хвилі в діапазоні від 400 нм до 700 нм, що падає на сітківку ока (нанометр, нм, дорівнює 10^{-9} метрам). Фізична потужність (або випромінювання) виражається спектральним розподілом потужності (SPD, spectral power distribution), часто з 31 компонента, кожен з яких представляє 10-нм смугу [5].

CIE визначає яскравість як атрибут візуального відчуття, згідно з яким область випромінює більше або менше світла. Сприйняття яскравості мозком неможливо визначити, тому CIE визначає більш практичну величину, яка називається яскравістю. Вона визначається як потужність випромінювання, зважена функцією спектральної чутливості, характерною для зору (око дуже чутливе до зеленого, трохи менш чутливе до червоного та набагато менш чутливе до синього). Світлова ефективність стандартного спостерігача визначається CIE як додатна функція довжини

хвилі, яка має максимум приблизно на 555 нм. Коли спектральний розподіл потужності інтегрується з використанням цієї функції як зважувальної функції, результатом є яскравість за CIE, яка позначається Y. Яскравість є важливою величиною в галузі цифрової обробки та стиснення зображень [5].

Яскравість пропорційна потужності джерела світла. Вона подібна до інтенсивності, але спектральний склад яскравості пов'язаний із здатністю сприймати яскравість людським зором.

Око дуже чутливе до невеликих змін яскравості, тому корисно мати колірні простори, які використовують Y як один з трьох параметрів. Простий спосіб зробити це - відняти Y від синього та червоного компонентів RGB та використовувати три компоненти Y, $B-Y$ та $R-Y$ як новий колірний простір. Останні два компоненти називаються кольоровістю. Вони представляють колір з точки зору наявності або відсутності синього (Cb) та червоного (Cr) кольорів для заданої інтенсивності яскравості [5].

У $B-Y$ та $R-Y$ використовуються різні діапазони чисел для різних застосувань. Діапазони $YPb\,\mathrm{Pr}$ оптимізовані для компонентного аналогового відео. Діапазони $YCbCr$ підходять для компонентного цифрового відео, такого як студійне відео, JPEG, JPEG 2000 та MPEG [5].

Колірний простір $YCbCr$ був розроблений як частина Рекомендації ITU-R BT.601 (раніше CCIR 601) під час розробки всесвітнього стандарту цифрового компонентного відео. Y визначено як діапазон від 16 до 235; Cb та Cr визначено як діапазон від 16 до 240, де 128 дорівнює нулю. Існує кілька форматів дискретизації $YCbCr$, таких як 4:4:4, 4:2:2, 4:1:1 та 4:2:0, які також описані в рекомендації [5].

Перетворення між RGB з діапазоном 16–235 та $YCbCr$ є лінійними, а отже, простими. Перетворення RGB в $YCbCr$ виконується таким чином (зверніть увагу на невелику вагу синього) [5]:

$$
\begin{aligned}
Y &= (77/256)R + (150/256)G + (29/256)B, \\
Cb &= -(44/256)R - (87/256)G + (131/256)B + 128, \\
Cr &= (131/256)R - (110/256)G - (21/256)B + 128;
\end{aligned}
\tag{2.1}
$$

тоді як протилежне перетворення є [5]

$$
\begin{aligned}
R &= Y + 1{,}371(Cr - 128), \\
G &= Y - 0{,}698(Cr - 128) - 0{,}336(Cb - 128), \\
B &= Y + 1{,}732(Cb - 128).
\end{aligned}
\tag{2.2}
$$

Під час перетворення *YCbCr* у RGB отримані значення RGB мають номінальний діапазон 16-235, з можливими випадковими значеннями 0 - 15 та 236 - 255 [5].

2.2 Зорова система людини

Ми бачимо світло, яке потрапляє в око і падає на сітківку, де є два типи світлочутливих клітин. Вони містять пігменти, що поглинають видиме світло і, отже, дають нам відчуття зору. Один тип - це палички, яких багато, вони розкидані по всій сітківці та реагують лише на світло та темряву. Вони дуже чутливі та можуть реагувати на один фотон світла. В оці приблизько від $110 \cdot 10^6$ до $125 \cdot 10^6$ паличок [4, 5]. Інший тип - це колбочки, розташовані в одній невеликій ділянці сітківки (фовеа). Їх кількість становить близько $6{,}4 \cdot 10^6$, вони чутливі до кольору, але потребують інтенсивнішого світла, порядку сотень фотонів. До речі, колбочки дуже чутливі до червоного, зеленого та синього (див. рис. 2.1), що є однією з причин, чому телебачення використовує ці кольори як основні [5].

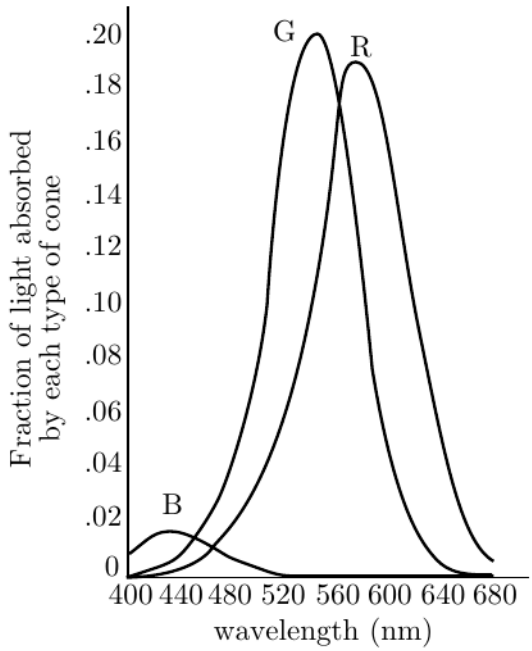


Рисунок 2.1 - Чутливість колбочок

Кожен зі світлових датчиків в оці, паличок і колбочок, надсилає світлове відчуття до мозку, яке по суті є пікселем, а мозок об'єднує ці пікселі в неперервне зображення. Таким чином, людське око схоже на цифрову камеру. Як тільки це усвідомлюємо, ми, природно, хочемо порівняти роздільну здатність ока з роздільною здатністю сучасної цифрової камери. Сучасні цифрові камери мають від 300 000 датчиків (для дешевої камери) до приблизно шести мільйонів датчиків (для високоякісної) [5].

Таким чином, око має набагато вищу роздільну здатність, але його ефективна роздільна здатність ще вища, якщо врахувати, що око може рухатися та перефокусуватися приблизно три-чотири рази на секунду. Це означає, що за одну секунду око може сприймати та надсилати до мозку близько півмільярда пікселів. Якщо припустити, що наша камера робить знімок раз на секунду, то співвідношення роздільних здатностей становить приблизно 100 [5].

Певні кольори, такі як червоний, помаранчевий та жовтий, психологічно асоціюються з теплом. Вони вважаються *теплими* та

змушують зображення виглядати більшим і ближчим, ніж воно є насправді. Інші кольори, такі як синій, фіолетовий і зелений, асоціюються з холодними речами (повітря, небо, вода, лід) і тому називаються *холодними* кольорами. Вони змушують зображення виглядати меншим і віддаленим [5].

3 СТИСНЕННЯ ЗОБРАЖЕНЬ ІЗ ВТРАТАМИ

3.1 Дискретне косинусне перетворення

Комітет JPEG обрав ДКП (дискретне косинусне перетворення) через його хорошу продуктивність і оскільки воно не висуває ніяких додаткових вимог щодо структури даних (ДПФ, дискретне перетворення Фур'є, наприклад, припускає, що дані, що перетворюються, є періодичними), а також тому, що існують способи його пришвидшити [5].

Стандарт JPEG вимагає застосування ДКП не до всього зображення, а до одиниць даних - блоків розміром 8x8 пікселів. Причини цього такі [5]:

1. Застосування ДКП до великих блоків включає багато арифметичних операцій і тому є повільним. Застосування ДКП до малих блоків даних є швидшим [5].

2. Досвід показує, що в зображеннях з неперервними тонами (напівтонових зображеннях) кореляції між пікселями є короткосяжними (обмежені близькою відстанню). Значення пікселя в такому зображенні (компонента кольору або відтінок сірого) близьке до значень його найближчих сусідів, але не має нічого спільного зі значеннями віддалених пікселів) [5].

В JPEG ДКП перетворення виконується наступним чином

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^{7} \sum_{y=0}^{7} p_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right); \quad 0 \le i, j \le 7, \qquad (3.1)$$

де $C_f = \begin{cases} 1/\sqrt{2}, & f = 0, \\ 1, & f > 0. \end{cases}$

ДКП є ключовим елементом JPEG для стиснення з втратами. Неважлива інформація зображення зменшується або видаляється шляхом

квантування 64 коефіцієнтів ДКП, особливо тих, що розташовані ближче до нижнього правого кута [5]. Якщо пікселі зображення корелюють (взаємопов'язані), квантування не сильно погіршує якість зображення. Для досягнення найкращих результатів кожен із 64 коефіцієнтів квантується шляхом ділення його на окремий коефіцієнт квантування (QC). Усі 64 QC є параметрами, якими, в принципі, може керувати користувач [5].

Декодер JPEG працює шляхом обчислення оберненого ДКП (ІДКП, інверсного ДКП) [5]:

$$p_{xy} = \frac{1}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} C_i C_j G_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right)\cos\left(\frac{(2y+1)j\pi}{16}\right); \quad 0 \le x, y \le 7. \qquad (3.2)$$

Він бере 64 квантовані коефіцієнти ДКП і обчислює 64 пікселі p_{xy}. Якщо коефіцієнти квантування (QC, quantization coefficients) підібрані правильно, нові 64 пікселі будуть дуже схожі на оригінальні. Математично, ДКП - це взаємно однозначне відображення 64-точкових векторів з простору зображення в частотний простір. ІДКП (інверсне ДКП) - це зворотне відображення. Якби ДКП та ІДКП можна було обчислити з нескінченною точністю, і якби коефіцієнти ДКП не квантувалися, оригінальні 64 пікселі були б точно відновлені [5].

3.2 Квантування

Після обчислення кожного 8×8 блоку коефіцієнтів ДКП $\|G_{ij}\|$ виконується квантування. Саме на цьому етапі відбувається втрата інформації (за винятком деяких неминучих втрат через обмежену точність обчислень на інших етапах). Кожне число в матриці коефіцієнтів ДКП ділиться на відповідне число з так званої «таблиці квантування», після чого результат округлюється до найближчого цілого. Для трьох колірних

компонент потрібні три окремі таблиці. Стандарт JPEG дозволяє використовувати до чотирьох таких таблиць, і користувач може обрати будь-яку з них для квантування кожної колірної компоненти [5].

Ті 64 числа, які складають кожну таблицю квантування, є параметрами JPEG. В принципі, всі вони можуть бути задані та точно налаштовані користувачем для максимального стиснення. На практиці, мало хто з користувачів має терпіння або досвід для експериментів з такою кількістю параметрів, тому програмне забезпечення JPEG зазвичай використовує наступні два підходи [5]:

1. Таблиці квантування за замовчуванням. Дві такі таблиці, для компонентів яскравості (градації сірого) та кольоровості, є результатом багатьох експериментів, проведених комітетом JPEG. Вони включені до стандарту JPEG та відтворені тут, у табл. 3.1 і 3.2. Легко побачити, як коефіцієнти (ширина кроків) квантування (QC) в таблиці загалом зростають, коли ми рухаємося від верхнього лівого кута до нижнього правого кута. Саме таким чином JPEG зменшує коефіцієнти ДКП з високими просторовими частотами [5].

Таблиця 3.1 –Рекомендована таблиця квантування яскравостей

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

Таблиця 3.2 – Рекомендована таблиця квантування хроматичних компонент *Cr* і *Cb*

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

2. Проста таблиця квантування $\|Q_{ij}\|$ обчислюється на основі одного параметра R, заданого користувачем. Простий вираз, такий як $Q_{ij} = 1 + (i + j) \cdot R$ гарантує, що QCs починаються з малого значення у верхньому лівому куті та збільшуються до нижнього правого кута. В табл. 3 наведено приклад такої таблиці коли $R = 2$ [5].

Таблиця 3.3 – Таблиця квантування $1 + (i + j) \cdot 2, \quad 0 \le i, j \le 7$

1	3	5	7	9	11	13	15
3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29

Якщо квантування виконано правильно, в матриці коефіцієнтів ДКП залишиться дуже мало ненульових чисел (елементів), і вони, як правило,

будуть сконцентровані у верхній лівій частині. Ці числа є результатом роботи алгоритма JPEG, але перед записом у вихідний потік вони додатково стискаються. В літературі по JPEG це стиснення називається “ентропійним кодуванням”. Ентропійне кодування використовує три операції для стиснення матриці цілих чисел розміром 8x8 [5]:

1. Елементи, що складають блок 8x8 (загалом 64 числа) перевпорядковуються шляхом сканування блока зигзагом (рис. 3.3). В результаті утворюється послідовність з 64 чисел, яка починається з кількох ненульових значень і зазвичай закінчується багатьма послідовними нулями. На вихід потрапляють лише ненульові числа (після їх подальшого стиснення), за якими слідує спеціальний код кінця блоку (EOB). Таким чином, немає необхідності виводити кінцеві нулі (можна сказати, що EOB - це кодування довжин серій (run-length encoding) для всіх кінцевих нулів) [5].

2. Ненульові числа стискаються за допомогою кодування Хаффмана.

3. Перше з цих чисел (DC-коефіцієнт) обробляється інакше, ніж інші (AC-коефіцієнти).

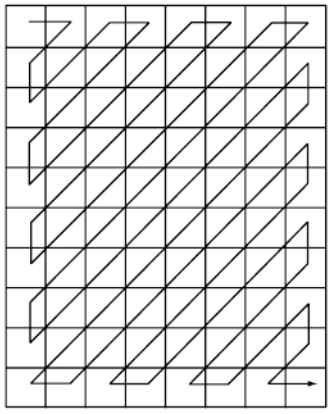


Рисунок 3.1 – Сканування матриці зигзагами

3.3 Кодування

Спочатку розглянемо пункт 3 з останнього переліку попереднього підрозділу. Кожна матриця квантованих коефіцієнтів ДКП розміром 8x8 містить один DC-коефіцієнт (у позиції (0, 0), верхній лівий кут) та 63 AC-коефіцієнти. DC-коефіцієнт є мірою середнього значення 64 оригінальних пікселів, що складають блок даних. Досвід показує, що в зображенні з неперервними тонами сусідні блоки даних пікселів зазвичай корелюють у тому сенсі, що середні значення пікселів у сусідніх блоках даних близькі. Відомо, що DC-коефіцієнт (постійна складова, direct current) блоку даних 8x8 пікселів є кратним середньому значенню 64 пікселів, що складають блок. Це означає, що DC-коефіцієнти сусідніх блоків даних не сильно відрізняються. JPEG зберігає перший з них (в закодованому вигляді), а потім повертаються різниці (також закодовані) DC-коефіцієнтів послідовних блоків даних [5].

Наведемо приклад. Якщо значення перших трьох блоків даних 8x8 зображення мають квантовані коефіцієнти постійної напруги, що дорівнюють 1118, 1114 та 1119 відповідно, то в вихідний JPEG-файл з першого блоку даних потрапить 1118 (але закодоване за Хаффманом, див. нижче), а потім 63 (закодовані) коефіцієнти змінної напруги цього блоку даних. Вихідний сигнал для другого блоку даних буде 1114 − 1118 = −4 (також закодований за Хаффманом), а потім 63 (закодовані) коефіцієнти змінної напруги цього блоку даних, а вихідний сигнал для третього блоку даних буде 1119 − 1114 = 5 (також закодований за Хаффманом), а потім знову 63 (закодовані) коефіцієнти змінної напруги цього блоку даних. Такий спосіб обробки коефіцієнтів постійної напруги вартий додаткових зусиль, оскільки різниці невеликі [5].

Кодування різниць DC-коефіцієнтів виконується за допомогою таблиці 3.4. Отже спочатку скажемо декілька слів про цю таблицю. Кожен рядок має номер (зліва), унарний код для рядка (справа) та декілька

стовпців між ними. Кожен рядок містить більші за модулем числа (а також більше чисел), ніж попередній, але не містить чисел з попередніх рядків. Рядок i містить діапазон цілих чисел $[-(2^i-1),+(2^i-1)]$ але без середнього діапазону $[-(2^{i-1}-1),+(2^{i-1}-1)]$. Таким чином, рядки стають дуже довгими, що означає, що простий двовимірний масив не є підходящею структурою даних для цієї таблиці [5]. Насправді, немає потреби зберігати ці цілі числа в структурі даних, оскільки програма може визначити, де в таблиці має знаходитися будь-яке задане ціле число x аналізуючи біти x.

Таблиця 3.4 – Кодування різниць DC-коефіцієнтів

0:	0										0
1:	-1	1									10
2:	-3	-2	2	3							110
3:	-7	-6	-5	-4	4	5	6	7			1110
4:	-15	-14	…	-9	-8	8	9	10	…	15	11110
5:	-31	-30	-29	…	-17	-16	16	17	…	31	111110
6:	-63	-62	-61	…	-33	-32	32	33	…	63	1111110
7:	-127	-126	-125	…	-65	-64	64	65	…	127	11111110
…	…	…	…	…	…	…	…	…	…	…	…
14:	-16383	-16382	-16381	…	-8193	-8192	8192	8193	…	16383	111111111111110
15:	-32767	-32766	-32765	…	-16385	-16384	16384	16385	…	32767	1111111111111110
16:	32767										1111111111111111

Перший DC-коефіцієнт, який потрібно закодувати в нашому прикладі, - 1118. Він знаходиться в рядку 11, стовпці 930 таблиці (нумерація стовпців починається з нуля), тому він кодується як 111111111110|01110100010 (унарний код для рядка 11, за яким слідує 11-бітне двійкове значення 930). Друга різниця DC-коефіцієнта це -4. Вона знаходиться в рядку 3, стовпці 3 таблиці 3.4, тому кодується як 1110|011 (унарний код для рядка 3, за яким слідує 3-бітне двійкове значення 3) [5].

Пункт 2 вище стосується точного способу стиснення 63 AC-коефіцієнтів блоку даних. Він використовує комбінацію RLE (run length encoding, кодування довжин серій) та кодування Хаффмана або

арифметичного кодування. Ідея полягає в тому, що послідовність AC-коефіцієнтів зазвичай містить лише декілька ненульових чисел, з серіями нулів між ними, та довгою серією нулів в кінці. Для кожного ненульового числа x кодер [3]:

1) знаходить кількість Z послідовних нулів, що передують x;

2) знаходить x в таблиці 3.4 та визначає номери його рядка та стовпця (R and C);

3) пара (R, Z) (саме (R, Z), а не (R, C)) використовується як номери рядка та стовпця для таблиці 3.5; і

4) код Хаффмана, знайдений у цій позиції в таблиці, об'єднується з C (де C записується як R-бітне число), і результат (нарешті) є кодом, що видається кодером JPEG для AC-коефіцієнта x та всіх послідовних нулів, що йому передують.

Таблиця 3.5 – Кодування AC-коефіцієнтів

R Z:	0	1	···	15
0:	1010			11111111001(ZRL)
1:	00	1100	···	1111111111110101
2:	01	11011	···	1111111111110110
3:	100	1111001	···	1111111111110111
4:	1011	111110110	···	1111111111111000
5:	11010	11111110110	···	1111111111111001
···	···			

Коди Хаффмана в таблиці 3.5 не є рекомендованими стандартом JPEG. Стандарт рекомендує використання таблиць 3.6 та 3.7 і стверджує, що кодек JPEG може використовувати до чотирьох таблиць коду Хаффмана, за винятком того, що базовий режим може використовувати лише дві такі таблиці. Таким чином, фактичні коди в таблиці 3.5 є довільними [5].

Таблиця 3.6 – Рекомендовані коди Хаффмана для AC-коефіцієнтів яскравості

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	00 1111000	01 11111000	100 1111110110	1011 1111111110000010	11010 1111111110000011
1	1100 1111111110000100	11011 1111111110000101	11110001 1111111110000110	111110110 1111111110000111	11111110110 1111111110001000
2	11100 111111110001010	11111001 111111110001011	1111110111 111111110001100	111111110100 111111110001101	111111110001001 111111110001110
3	111010 1111111110010001	111110111 1111111110010010	111111110101 1111111110010011	1111111110001111 1111111110010100	1111111110010000 1111111110010101
4	111011 1111111110011001	1111111000 1111111110011010	1111111110010110 1111111110011011	1111111110010111 1111111110011100	1111111110011000 1111111110011101
5	1111010 1111111110100001	11111110111 1111111110100010	1111111110011110 1111111110100011	1111111110011111 1111111110100100	1111111110100000 1111111110100101
6	1111011 1111111110101001	111111110110 1111111110101010	1111111110100110 1111111110101011	1111111110100111 1111111110101100	1111111110101000 1111111110101101
7	11111010 1111111110110001	111111110111 1111111110110010	1111111110101110 1111111110110011	1111111110101111 1111111110110100	1111111110110000 1111111110110101
8	111111000 1111111110111001	111111111000000 1111111110111010	1111111110110110 1111111110111011	1111111110110111 1111111110111100	1111111110111000 1111111110111101
9	111111001 1111111111000010	1111111110111110 1111111111000011	1111111110111111 1111111111000100	1111111111000000 1111111111000101	1111111111000001 1111111111000110
A	111111010 1111111111001011	1111111111000111 1111111111001100	1111111111001000 1111111111001101	1111111111001001 1111111111001110	1111111111001010 1111111111001111
B	1111111001 1111111111010100	1111111111010000 1111111111010101	1111111111010001 1111111111010110	1111111111010010 1111111111010111	1111111111010011 1111111111011000
C	1111111010 1111111111011101	1111111111011001 1111111111011110	1111111111011010 1111111111011111	1111111111011011 1111111111100000	1111111111011100 1111111111100001
D	11111111000 1111111111100110	1111111111100010 1111111111100111	1111111111100011 1111111111101000	1111111111100100 1111111111101001	1111111111100101 1111111111101010
E	1111111111101011 1111111111110000	1111111111101100 1111111111110001	1111111111101101 1111111111110010	1111111111101110 1111111111110011	1111111111101111 1111111111110100
F	1111111001 1111111111111001	1111111111110101 1111111111111010	1111111111110110 1111111111111011	1111111111110111 1111111111111101	1111111111111000 1111111111111110

Читач повинен звернути увагу на код EOB у позиції (0, 0) та код ZRL у позиції (0, 15). Перший вказує на кінець блоку, а другий – це код, що формується для 15 послідовних нулів, коли кількість послідовних нулів перевищує 15. Ці коди рекомендовані для коефіцієнтів змінного струму (AC-коефіцієнтів) яскравості таблиці 3.6. Значення кодів EOB та ZRL, рекомендовані для кольорових AC-коефіцієнтів з таблиці 3.7, – це 00 та 11111111010 відповідно [5].

Таблиця 3.7 – Рекомендовані коди Хаффмана для АС-коефіцієнтів, що відповідають хроматичним компонентам

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	01 111000	100 1111000	1010 111110100	11000 1111110110	11001 111111110100
1	1011 111111110101	111001 111111110001000	11110110 111111110001001	111110101 111111110001010	11111110110 111111110001011
2	11010 1111111110001100	11110111 1111111110001101	1111110111 1111111110001110	111111110110 1111111110001111	111111111000010 1111111110010000
3	11011 1111111110010010	11111000 1111111110010011	1111111000 1111111110010100	111111110111 1111111110010101	1111111110010001 1111111110010110
4	111010 1111111110011010	111110110 1111111110011011	1111111110010111 1111111110011100	1111111110011000 1111111110011101	1111111110011001 1111111110011110
5	111011 1111111110100010	1111111001 1111111110100011	1111111110011111 1111111110100100	1111111110100000 1111111110100101	1111111110100001 1111111110100110
6	1111001 1111111110101010	11111110111 1111111110101011	1111111110100111 1111111110101100	1111111110101000 1111111110101101	1111111110101001 1111111110101110
7	1111010 1111111110110010	11111111000 1111111110110011	1111111110101111 1111111110110100	1111111110110000 1111111110110101	1111111110110001 1111111110110110
8	11111001 1111111110111011	1111111110110111 1111111110111100	1111111110111000 1111111110111101	1111111110111001 1111111110111110	1111111110111010 1111111110111111
9	111110111 1111111111000100	1111111111000000 1111111111000101	1111111111000001 1111111111000110	1111111111000010 1111111111000111	1111111111000011 1111111111001000
A	111111000 1111111111001101	1111111111001001 1111111111001110	1111111111001010 1111111111001111	1111111111001011 1111111111010000	1111111111001100 1111111111010001
B	111111001 1111111111010110	1111111111010010 1111111111010111	1111111111010011 1111111111011000	1111111111010100 1111111111011001	1111111111010101 1111111111011010
C	111111010 1111111111011111	1111111111011011 1111111111100000	1111111111011100 1111111111100001	1111111111011101 1111111111100010	1111111111011110 1111111111100011
D	11111111001 1111111111101000	1111111111100100 1111111111101001	1111111111100101 1111111111101010	1111111111100110 1111111111101011	1111111111100111 1111111111101100
E	11111111100000 1111111111110001	1111111111101101 1111111111110010	1111111111101110 1111111111110011	1111111111101111 1111111111110100	1111111111110000 1111111111110101
F	111111111000011 1111111111111010	111111111010110 1111111111111011	1111111111110111 1111111111111100	1111111111111000 1111111111111101	1111111111111001 1111111111111110

Як приклад [5] розглянемо послідовність

$$1118, 2, 0, -2, \underbrace{0,\ldots,0}_{13}, -1, 0,\ldots$$

Перший АС-коефіцієнт, який дорівнює 2, не має нулів перед собою, тому Z = 0. Він знаходиться в таблиці 3.4 в рядку 2, стовпці 2, отже R = 2 та C = 2. Код Хаффмана в позиції (R, Z) = (2, 0) таблиці 3.5 дорівнює 01, тому остаточний код, що видається для 2, є 01|10. Наступний ненульовий коефіцієнт, -2, має один нуль перед собою, тому Z = 1. Він знаходиться в

таблиці 3.4 у рядку 2, стовпці 1, отже R = 2 та C = 1. Код Хаффмана в позиції (R, Z) = (2, 1) таблиці 3.5 дорівнює 11011, тому остаточний код, що видається для 2, є 11011|01.

Нарешті, послідовність кінцевих нулів кодується як 1010 (EOB), тому вихід для наведеної вище послідовності AC-коефіцієнтів становить 011011011101110101010. Ми бачили раніше, що DC-коефіцієнт кодується як 111111111110|01110100010, тому остаточний вихід для всього блоку даних з 64 пікселів — це 46-бітне число

1111111111100111010001001101101110111010101010.

Ці 46 біт кодують одну колірну компоненту 64 пікселів блоку даних. Припустимо, що дві інші колірні компоненти також кодуються у 46-бітні числа. Якщо кожен піксель спочатку складається з 24 біт, то це відповідає коефіцієнту стиснення $64 \times 24 / (46 \times 3) \approx 11{,}13$; дуже вражаюче!

(Зауважте, що DC-коефіцієнт 1118 вніс 23 з 46 біт. Наступні блоки даних кодують різниці своїх DC-коефіцієнтів, що може зайняти менше 10 біт замість 23. В результаті вони можуть мати значно вищі коефіцієнти стиснення.)

Ті ж самі таблиці (таблиці 3.6 та 3.7), що використовуються кодером, звичайно, повинні використовуватися і декодером. Таблиці можуть бути попередньо визначені та використовуватися кодеком JPEG за замовчуванням, або вони можуть бути спеціально розраховані для даного зображення на спеціальному проході, що передує фактичному стисненню. Стандарт JPEG не визначає жодних кодових таблиць, тому будь-який кодек JPEG повинен використовувати власні [5].

Деякі варіанти JPEG використовують певну версію арифметичного кодування, яка називається QM-кодером і визначена у стандарті JPEG. Ця версія арифметичного кодування є адаптивною, тому вона не потребує таблиць 3.6 та 3.7. Вона адаптує свою поведінку до статистики зображення в міру просування. Використання арифметичного кодування може забезпечити на 5–10% краще стиснення, ніж код Хаффмана, для типового

зображення з неперервним тоном. Однак його реалізація складніша, ніж кодування Хаффмана, тому на практиці рідко можна знайти JPEG-кодек, який його використовує.

4 КРИТЕРІЇ ЯКОСТІ ЗОБРАЖЕНЬ

При обробці та аналізі зображень завжди доводиться ставити питання про їхню якість. Якість такого складного об'єкта, як зображення, є дуже важливим, але водночас і доволі нечітким поняттям. Вона оцінюється різними способами та у зв'язку з різними завданнями [6].

Іноді якість розглядається як характеристика самого зображення і визначається його власними властивостями (статистичними, структурними, семантичними). Відповідні критерії є або суб'єктивними, або спираються на об'єктивні характеристики зображення: форму та параметри розподілу яскравості, ширину просторового спектра тощо. Такі безвідносні критерії мають досить обмежене застосування і майже ніде не використовуються [6].

При іншому підході якість розглядається як міра близькості двох зображень: реального та деякого ідеального, або вхідного та перетвореного. Такий підхід є більш конструктивним, він дозволяє оцінювати кількісні зміни значень яскравості, рівень спотворень зображень при їх перетвореннях (фільтрації, стисненні даних тощо), тобто, по суті, якість самого засобу перетворення - алгоритму чи системи. Саме це є дуже важливим при побудові алгоритмів і систем обробки зображень та оцінці якості алгоритмів [6].

Розглянемо показники якості зображень, що використовуються найчастіше.

4.1 Критерій візуального сприйняття

Він базується на результатах експертизи. Звичайна процедура оцінки якості полягає в пред'явленні набору пар зображень (аналізовані та ідеальні) експертам-спостерігачам, які висловлюють судження на рівні:

«спотворення непомітні», «помітні, але не погіршують», «погіршують, але не заважають», «трохи заважають» тощо. Індивідуальні оцінки обробляються та усереднюються. Існують спеціальні прийоми, що виключають «звикання» експертів у процесі експериментів, їхні упередження до конкретних сюжетів і т.д. [6].

Проведення подібної експертизи - завжди складне завдання, і її результати є вельми приблизними. Для спеціальних зображень (які, наприклад, отримують при дистанційному зондуванні) експерти мають бути фахівцями з вирішення відповідних прикладних завдань аналізу відеоінформації [6].

Але головний недолік суб'єктивного критерію - відсутність кількісних оцінок. Він не дозволяє вирішувати завдання оптимізації систем обробки зображень в просторі параметрів, що безперервно змінюються. Тут можливий лише перебір варіантів, і то не дуже великий.

Бажано, щоб критерій мав просту аналітичну форму і просто обчислювався за наданими зображеннями. Цій вимозі відповідає низка критеріїв, що розглядаються нижче [6].

4.2 Середньоквадратичний критерій

Нехай зображення $f(n_1, n_2)$ та $g(n_1, n_2)$ описуються моделями однорідних випадкових полів. Мірою відповідності реального зображення ідеальному може слугувати середнє значення квадрата їх різниці [6]:

$$\varepsilon_{кв} = E[(f - g)^2] \tag{4.1}$$

ця величина буде постійною по всьому полю аргументів, тому аргументи (однакові для f, g) для стислості не вказуємо.

Якщо математичні сподівання f і g рівні, то різниця має нульове середнє, і величина $\varepsilon_{кв}$ набуває сенсу дисперсії різниці (а значення $\varepsilon_{кв}$ - середньоквадратичного відхилення g від f) двох зображень.

Для стаціонарної моделі зазвичай вважається виконаною умова ергодичності, за якої усереднення за ансамблем реалізацій може бути замінене на усереднення за однією реалізацією. Тоді для неперервних зображень, заданих при $|x_1| < L_1$, $|x_2| < L_2$, маємо [6]

$$\varepsilon_{\text{кв}}^2 \approx \frac{1}{4L_1L_2} \int_{-L_1}^{L_1} \int_{-L_2}^{L_2} [f(x_1,x_2) - g(x_1,x_2)]^2 \, dx_1 \, dx_2 , \qquad (4.2)$$

а для дискретних, заданих при $0 \le n_1 \le N_1 - 1$, -1, $0 \le n_2 \le N_2 - 1$, маємо [6]

$$\varepsilon_{\text{кв}}^2 \approx \frac{1}{N_1N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} [f(n_1,n_2) - g(n_1,n_2)]^2 . \qquad (4.3)$$

Зауважимо, що в задачах порівняльного аналізу варіантів та оптимізації постійні коефіцієнти в (4.2) та (4.3) можуть бути відкинуті.

Вирази (4.2) і (4.3) дозволяють обчислювати середньоквадратичну помилку і для пари довільних зображень, які не обов'язково описуються стаціонарними полями. Так часто і робиться. Однак у цьому випадку слід мати на увазі, що значення $\varepsilon_{\text{кв}}^2$ характеризуватиме «середню» якість зображення в цілому, а на різних його фрагментах помилки, в принципі, можуть відрізнятися [6].

Перевага середньоквадратичного критерію - його простота. При його використанні багато завдань аналізу та оптимізації алгоритмів обробки зображень легко вирішуються аналітично. Тому він дуже часто застосовується.

При обробці зображень слід враховувати, що даний критерій погано узгоджується із критерієм суб'єктивного сприйняття [6].

4.3 Критерій максимальної помилки (рівномірного наближення)

У неперервному випадку:

$$\varepsilon_{\max} = \max_{(x_1,x_2)} |f(x_1,x_2) - g(x_1,x_2)|, \tag{4.4}$$

і дискретному

$$\varepsilon_{\max} = \max_{(n_1,n_2)} |f(n_1,n_2) - g(n_1,n_2)|. \tag{4.5}$$

Це дуже жорсткий критерій. Він використовується в тих випадках, коли висувається вимога високої точності представлення не зображення в цілому, а кожної його точки (відліку). Це необхідно у відповідальних випадках, при отриманні цінних, унікальних зображень [6].

Однак даний показник має серйозний недолік - складність теоретичної оцінки і, відповідно, використання його в процедурах оптимізації (принаймні для загальноприйнятих моделей зображення).

4.4 Імовірнісно-зональний критерій

Цей критерій є модифікацією (та узагальненням) попереднього. У разі використання критерію максимальної помилки вважається, що всі значення різницевого сигналу (поточної помилки) $\varepsilon = f - g$ лежать у діапазоні $[-\varepsilon_{\max}, \varepsilon_{\max}]$, тобто розподіл імовірностей для ε має, наприклад, вигляд, показаний на рис. 4.1 [6].

Однак на практиці в багатьох випадках це не виконується. Найпростішим прикладом є ситуація, коли зображення спотворене адитивним гаусовим шумом $g = f + v$, що має щільність розподілу, яка ніде не дорівнює нулю (див. рис. 4.2) [6]:

$$p_v(v) = \frac{1}{2\pi\sigma_v^2} e^{-\frac{v^2}{2\sigma_v^2}} \tag{4.6}$$

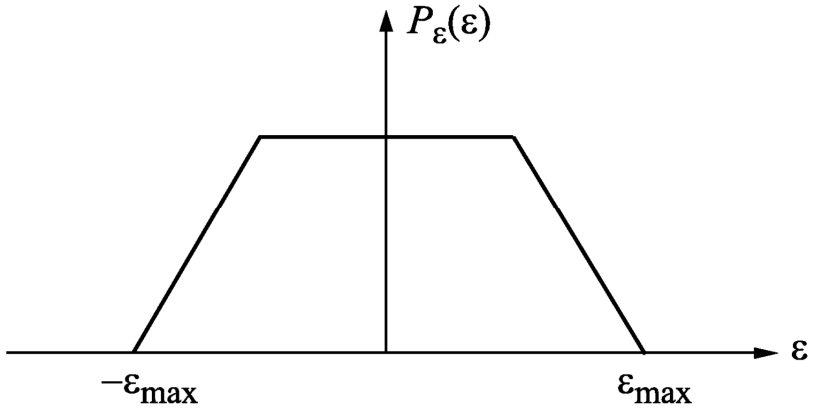


Рисунок 4.1 – Приклад розподілу імовірностей різницевого сигналу

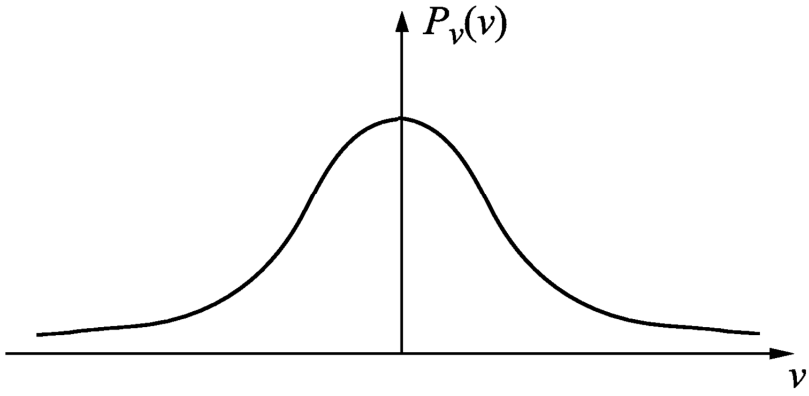


Рисунок 4.2 – Щільність імовірності розподілу гаусового шуму

Різниця $\varepsilon = f - g = -v$ - має такий самий розподіл. Тут можна оцінити максимальну помилку лише з певною довірчою ймовірністю p. Імовірнісно-зональний критерій визначається парою чисел $(\varepsilon_{\max}, p)$ [6].

Сенс цього критерію виражається формулою [6]

$$\int_{-\varepsilon_{\max}}^{\varepsilon_{\max}} P_{\varepsilon}(\varepsilon)\, d\varepsilon = p \qquad (4.7)$$

і ілюструється на рис. 4.3.

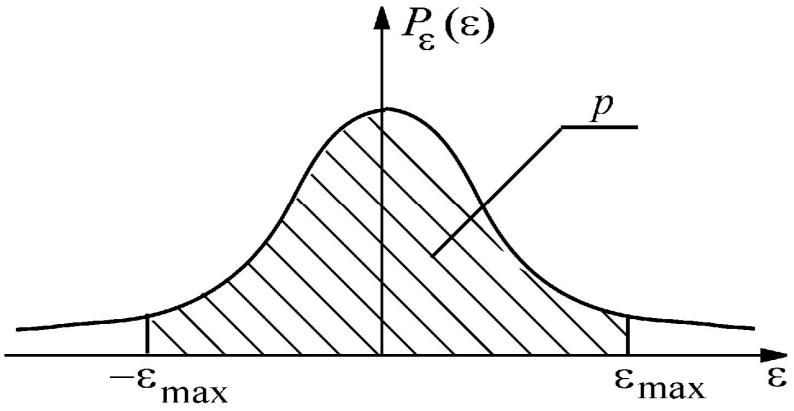


Рисунок 4.3 – Імовірнісно-зональний критерій

Тут, як і в попередньому випадку, часто виникають складнощі при теоретичній оцінці [6]. Значення такого показника якості отримують експериментально, в результаті аналізу гістограми розподілу помилки ε.

4.5 Похибки дискретного подання зображень

Реальне «фізичне» зображення є функцією неперервних просторових координат - $f(x_1, x_2)$. У комп'ютері обробляється його дискретний аналог, матриця $f(n_1, n_2)$ - цифрове зображення. Воно лише наближено відповідає

неперервному. Невідповідність зумовлена похибками, що вносяться в дані в процесі перетворення в цифрову форму [6].

Усі результати цифрової обробки стосуються саме цифрового зображення. Зрозуміло, що така обробка має сенс лише в тому випадку, якщо цифрові зображення достатньо точно описують первинні, тобто похибка цифрового подання є малою. Треба вміти оцінювати цю похибку.

Така оцінка важлива ще й тому, що дозволяє визначити потенційні можливості процедур цифрової обробки (фільтрації, кодування тощо) з погляду точності [6].

Існують похибки двох основних типів, пов'язаних із процедурами перетворення зображень у цифрову форму - дискретизації (за просторовими координатами) та квантування за рівнем (яскравості). Нижче ми дамо оцінку похибки квантування параметра за рівнем [6].

Нехай величина (параметр) f що перетворюється, може приймати будь-які значення з діапазону $[f_{\min}, f_{\max}]$, який називається шкалою параметра.

При поданні параметра в цифровій формі в межах шкали фіксується (призначається) Q квантових рівнів: $f_0, f_1, \ldots, f_{(Q-1)}$. Поточне (фактичне) значення параметра ототожнюється з одним із рівнів квантування, і далі замість значення параметра використовується просто номер обраного рівня, що кодується двійковим кодом. Якщо використовується b-розрядний код, то є можливість пронумерувати $Q = 2^b$ рівнів квантування [6].

Розташування квантових рівнів на шкалі параметрів може бути різним. На практиці інтервали між рівнями квантування зазвичай беруться однаковими. При цьому крок квантування за рівнем: $\Delta_f = f_q - f_{q-1}$ для будь-яких $1 \le q \le Q-1$ є величиною постійною [6].

Рівномірне розташування Q рівнів на шкалі параметра показано на рис. 4.4. Тут крок квантування

$$\Delta_f = \frac{f_{\max} - f_{\min}}{Q} = \frac{f_{\max} - f_{\min}}{2^b}. \qquad (4.8)$$

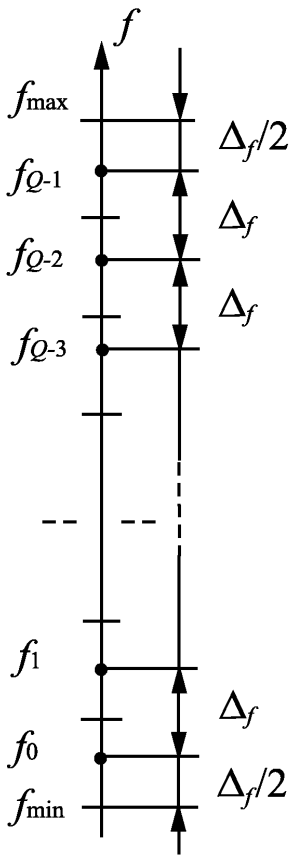


Рисунок 4.4 – Рівномірне розташування рівнів квантування на шкалі параметра

В даному випадку поточне значення параметра ототожнюється з найближчим рівнем квантування. Будемо розглядати саме такий варіант квантування [6].

Для кожного конкретного значення параметра f обирається свій квантовий рівень - f_q, при цьому похибка цифрового подання параметра (похибка квантування за рівнем) [6]

$$\varepsilon_f = f - f_q. \qquad (4.9)$$

Оскільки f - випадкова величина, то f_ε також є випадковою. Але можна визначити максимальне та середньоквадратичне значення похибки.

Максимальна похибка квантування за рівнем (для нашого варіанту квантування) [6]:

$$\varepsilon_{f\max} = \max|\varepsilon_f| = \Delta_f / 2. \qquad (4.10)$$

Зазвичай крок квантування Δ_f суттєво менший від шкали параметра (тобто $b > 1,\ Q >> 1,\ \Delta_f << f_{\max} - f_{\min}$). При цьому логічно припустити, що похибка ε_f - випадкова і має практично рівномірний розподіл у діапазоні $[-\Delta_f / 2, \Delta_f / 2]$ (см . рис. 4.5.) [6].

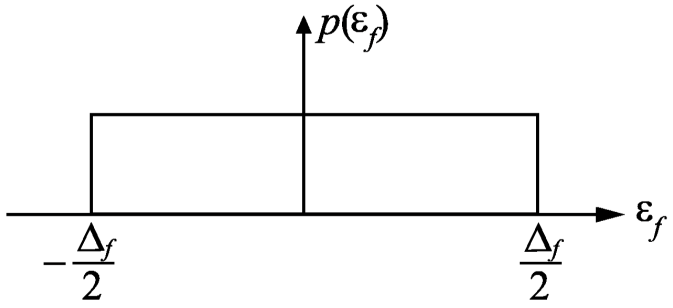


Рисунок 4.5 – Розподіл похибки

Дисперсія похибки [6]

$$\varepsilon^2_{f\text{кв}} = \int_{-\Delta_f}^{\Delta_f} \varepsilon_f^2 \cdot p(\varepsilon_f) d\varepsilon_f = \frac{\Delta_f^2}{12}, \qquad (4.11)$$

середньоквадратичне відхилення (корінь із дисперсії)

$$\varepsilon_{f\,\text{кв}} = \frac{\Delta_f}{2\sqrt{3}} = \frac{\varepsilon_{f\,\max}}{\sqrt{3}}. \qquad (4.12)$$

Врахуємо далі наступне. Якщо параметр f має нормальний (або близький до нормального) розподіл із дисперсією σ_f^2 та математичним сподіванням μ_f, то зазвичай прагнуть обрати шкалу так, щоб вона збігалася з «довірчим інтервалом» $[\mu_f - 3\sigma_f, \mu_f + 3\sigma_f]$ (усі значення f лежать в цьому інтервалі з імовірністю $\approx 0{,}997$). Тоді

$$f_{\max} - f_{\min} = 6\sigma_f; \qquad \Delta_f = \frac{6\sigma_f}{2^b}; \qquad (4.13)$$

і отримуємо, що

$$\varepsilon_{f\,\max} = \frac{3\sigma_f}{2^b}; \qquad \varepsilon_{f\,\text{кв}} = \frac{\sqrt{3}\sigma_f}{2^b}. \qquad (4.14)$$

5 ЕНТРОПІЯ ТА ЕФЕКТИВНА ШВИДКІСТЬ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ

5.1 Ентропія

Для розробки системи зв'язку з певною здатністю до обробки повідомлень потрібна метрика для вимірювання обсягу переданої інформації. Шеннон увів таку метрику, що називається ентропією джерела повідомлень (яке має n можливих вихідних значень). Ентропія (entropy) визначається як середня кількість інформації, що припадає на один вихід джерела, і виражається таким чином [7]:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \text{ біт/вихід джерела.} \qquad (5.1)$$

Тут p_i - імовірність i-го вихідного значення і $\sum_{i=1}^{n} p_i = 1$. Якщо повідомлення є двійковим або джерело має лише два можливих вихідних значення з імовірностями p та $q = 1 - p$, вираз для ентропії набуває такого вигляду [7]:

$$H = -(p \log_2 p + q \log_2 q). \qquad (5.2)$$

Залежність ентропії від p показана на рис. 5.1.

Величина H має низку особливостей:

1. Якщо логарифм у рівнянні (5.1) береться за основою 2, одиниця вимірювання H - середня кількість біт на подію. Тут одиниця вимірювання *біт* - це міра кількості інформації, і її не слід плутати з терміном "біт", що означає "двійкова цифра " (binary digit - bit).

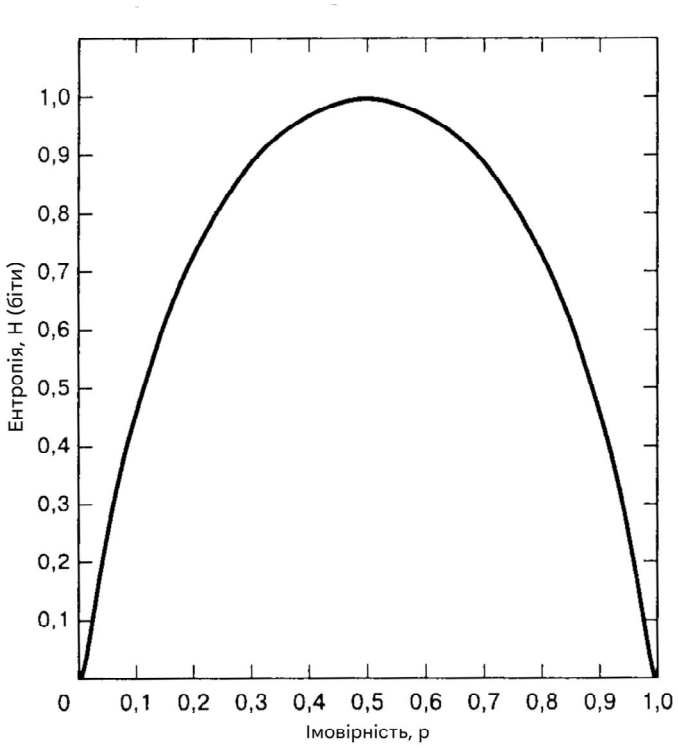


Рисунок 5.1 – Залежність ентропії від імовірності (дві події)

2. Сам термін "ентропія" має дещо невизначений сенс, що спричинено наявністю кількох формулювань у статистичній механіці. Для інформаційного джерела з двома рівноймовірними станами (наприклад, випадання симетричної монети) з рис. 5.1 видно, що невизначеність результату і, отже, середня кількість інформації є максимальними [7]. Щойно імовірності відходять від рівноймовірного стану, середня кількість інформації знижується. В граничному випадку, коли одна з ймовірностей стає нулем, H також дорівнює нулю. Результат відомий до того, як відбудеться подія, тож результат не несе в собі додаткової інформації [7].

3. Для ілюстрації зв'язку між кількістю інформації та апріорною ймовірністю (якщо апріорна ймовірність повідомлення на приймачі є нулем або одиницею, повідомлення можна не надсилати) розглянемо такий приклад [7]. Після дев'ятимісячної вагітності жінка опиняється в пологовій палаті. Чоловік із хвилюванням чекає в приймальні. Через деякий час до

нього підходить лікар і каже: "Прийміть мої вітання, ви стали батьком". Яку інформацію батько отримав від лікаря? Майже ніякої: батько практично достовірно знав, що дитина має народитися. Якби лікар сказав: "ви стали батьком хлопчика" або "ви стали батьком дівчинки", він передав би 1 біт інформації, оскільки існує 50% імовірність того, що дитина виявиться дівчинкою або хлопчиком.

5.2 Невизначеність та ефективна швидкість передавання інформації

Нехай двійковим симетричним каналом зі швидкістю 1000 двійкових символів/с відбувається передача інформації, а апріорна ймовірність передачі нуля або одиниці однакова. Припустимо також, що завади в каналі настільки значні, що, незалежно від переданого символу, ймовірність прийому одиниці дорівнює 1/2 (те саме - для нуля). У такому випадку половина прийнятих символів має випадково виявитися правильною, і може скластися враження, що система забезпечує швидкість 500 біт/с, хоча насправді ніякої інформації не передається. Однаково "гарний" прийом дає і використання "інформації", що надійшла з каналу, і генерація цієї "інформації" методом підкидання симетричної монети. Втраченою є інформація про правильність переданих символів. Для оцінки невизначеності в прийнятому сигналі Шеннон використовує поправочний коефіцієнт, який називає невизначеністю або ненадійністю (equivocation) [7].

Неоднозначність визначається як умовна ентропія повідомлення X, зумовлена даним повідомленням Y, або

$$H(X \mid Y) = -\sum_{X,Y} P(X,Y) \log_2 P(X \mid Y) = \\ -\sum_{Y} P(Y) \sum_{X} P(X \mid Y) \log_2 P(X \mid Y), \qquad (5.3)$$

де X - повідомлення, передане джерелом, Y - прийнятий сигнал, $P(X,Y)$ - сумісна ймовірність X та Y, а $P(X|Y)$ - умовна ймовірність X при прийомі Y. Неоднозначність можна представити як невпевненість у передачі X за умови, що надійшло Y. Для каналу без помилок $H(X|Y)=0$, оскільки отримане повідомлення Y абсолютно точно визначає X. Водночас для каналу з ненульовою ймовірністю виникнення символьної помилки $H(X|Y)>0$, оскільки канал вносить певну невизначеність [7].

Розглянемо двійкову послідовність, для якої апріорні ймовірності джерела $P(X=0)=P(X=1)=1/2$ і де в середньому в прийняту послідовність зі 100 біт канал вносить одну помилку ($P_B=0{,}01$). Виходячи з рівняння (5.3), неоднозначність $H(X|Y)$ можна записати таким чином

$$H(X|Y)=-[(1-P_B)\log_2(1-P_B)+P_B\log_2 P_B]=$$
$$-[0{,}99\log_2 0{,}99+0{,}01\log_2 0{,}01]=0{,}081 \text{ біт/отриманий символ.}$$

Отже, в кожний прийнятий символ канал вносить 0,081 біт неоднозначності.

Шеннон показав, що значення середньої ефективної кількості інформації H_{eff} в приймачі отримується шляхом віднімання неоднозначності з ентропії джерела. Отже,

$$H_{eff}=H(X)-H(X|Y). \qquad (5.4)$$

Для системи, що передає рівноімовірні двійкові символи, ентропія $H(X)$ дорівнює 1 біт/символ. Якщо символи приймаються з $P_B=0{,}01$, невизначеність, як показано вище, дорівнює 0,081 біт/отриманий символ.

Тоді, використовуючи рівняння (5.4), можемо записати ефективну ентропію H_{eff} прийнятого сигналу

$$H_{eff} = 1 - 0{,}081 = 0{,}919 \text{ біт/(отриманий символ).}$$

Іншими словами, якщо, наприклад, за секунду передається $R = 1000$ двійкових символів, то R_{eff} можна виразити таким чином:

$$R_{eff} = R \cdot H_{eff} = (1000 \text{ символів}) \times (0{,}919 \text{ біт/символ}) = 919 \text{ біт/с.}$$

Зауважимо, що в граничному випадку $P_B = 0{,}5$

$$H(X \mid Y) = -[0{,}5 \log_2 0{,}5 + 0{,}5 \log_2 0{,}5] = 1 \text{ біт/символ.}$$

Звідки при $R = 1000$ символів/с, отримуємо

$$R_{eff} = (1000 \text{ символів/с}) \times (1 - 1) = 0,$$

що й слід було очікувати.

6 КЕРУВАННЯ ШВИДКІСТЮ ПРИ СТИСНЕННІ ЗОБРАЖЕНЬ

В даному розділі демонструється той факт, що в будь-якій типовій системі кодування з перетворенням існує лінійна залежність між швидкістю кодування та процентним вмістом нулів серед квантованих коефіцієнтів перетворення, що позначаються як ρ. В роботі [23] на основі теореми Шеннона про кодування джерела представлено теоретичне обґрунтування цієї лінійної моделі джерела. Також обговорюється фізичний зміст параметра моделі. Зокрема демонструється, що він прямо пов'язаний із вмістом зображення і є мірою його складності.

Лінійна модель джерела, доповнена адаптивною схемою оцінки коефіцієнта пропорційності, призводить до уніфікованого алгоритму управління швидкістю для різних стандартних систем фото і відеокодування, таких як JPEG, MPEG-2, H.263, MPEG-4 та багатьох інших [23].

6.1 Загальні міркування. Алгоритми керування швидкістю

Вихідний бітрейт відеокодера значно змінюється з часом через активність у сцені. У візуальному зв'язку через вузькосмугові канали або канали зі змінними в часі характеристиками керування швидкістю є дуже важливим для забезпечення успішної передачі закодованих відеоданих через канали зв'язку [23]. В системах відеозв'язку в реальному часі, таких як відеоконференції, відеотелефонія, інтерактивні класи та веб-трансляції в реальному часі, наскрізна затримка передачі відео має бути дуже малою, що вимагає більш точного та надійного керування швидкістю.

У стандартному кодуванні зображень та відео вихідний бітрейт R контролюється параметром квантування відеокодера. Позначимо параметр квантування як q. Для рівномірного квантувача, q представляє крок

квантування. Для перцептивного квантувача з матрицею квантування, як у кодуванні зображень JPEG [2] та відео MPEG [8], q представляє коефіцієнт масштабування квантування. Залежність між q та R описується функцією швидкість-квантування (R - Q), що позначається як $R(q)$. Коли $R(q)$ відома, для досягнення цільового бітрейту кодування R_t, ми просто обираємо відповідний параметр квантування $q_t = R^{-1}(R_t)$ [23].

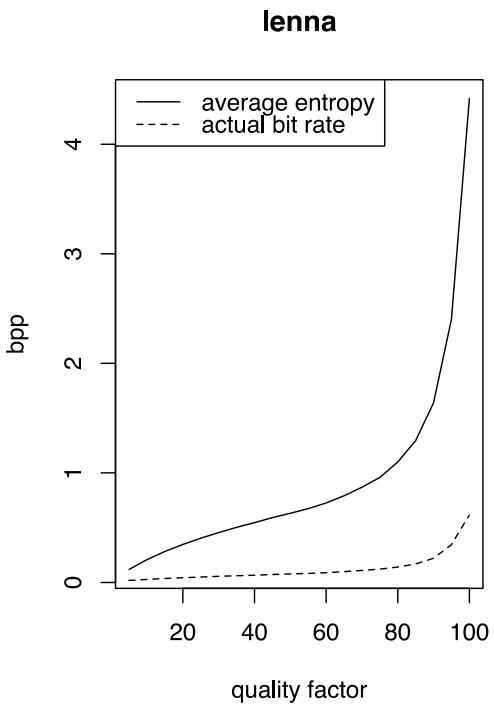
Формула залежності швидкість-спотворення (R - D) для простого квантувача була встановлена вже давно [9, 10]. Однак у типовій системі трансформаційного кодування, наприклад, у кодуванні JPEG [2, 5], такий тип аналітичної формули ентропії не працює, особливо при низьких бітрейтах [11, 23]. На рис. 6.1 показано тестові зображення “Lena” і “Pepper” формату pgm розміру 512x512 пікселів, що їх було використано для проведення експерименту. В свою чергу на рис. 6.2 показано фактичні бітрейти кодування JPEG та ентропії квантованих коефіцієнтів дискретного косинусного перетворення (ДКП) для зображень “Lena” і “Pepper” при різних масштабах квантування (показниках якості зображення). Можна бачити, що відносна похибка між ними є дуже великою. Для зручності порівняння, на рис. 6.3 також показані ті самі залежності, що і на рис. 6.2, але для кожного із зображень окремо.

У класичній формулі R - D, єдиним параметром, що описує вхідне джерело, є дисперсія вхідних даних. Добре відомо, що самої лише дисперсії далеко не достатньо для характеристики вхідних даних джерела та для визначення кінцевого бітрейту кодування. Крім того, аналітична формула ентропії не враховує поведінку кодування конкретного алгоритму. Наприклад, алгоритм стиснення зображень на основі вейвлетів, запропонований у [12] є набагато ефективнішим, ніж кодування JPEG [2]. Тому, навіть для того самого зображення, різні алгоритми кодування мають зовсім різні функції $R – D$ [23].

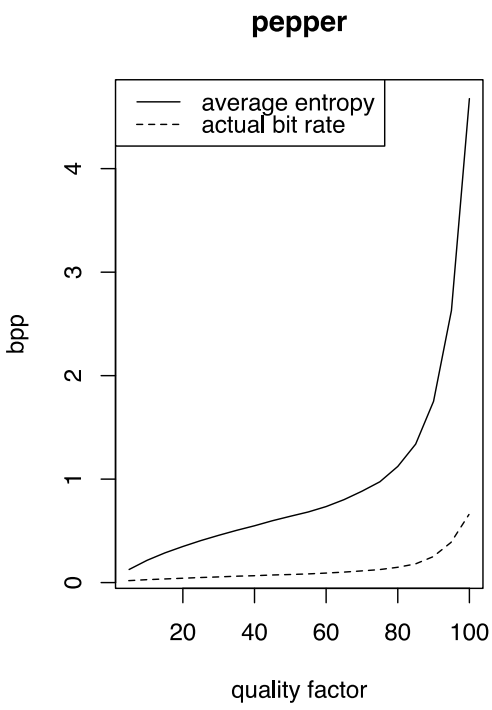
а)

б)

Рисунок 6.1 – Тестові зображення розміру 512x512 пікселів: а) Lenna.pgm, б) pepper.pgm

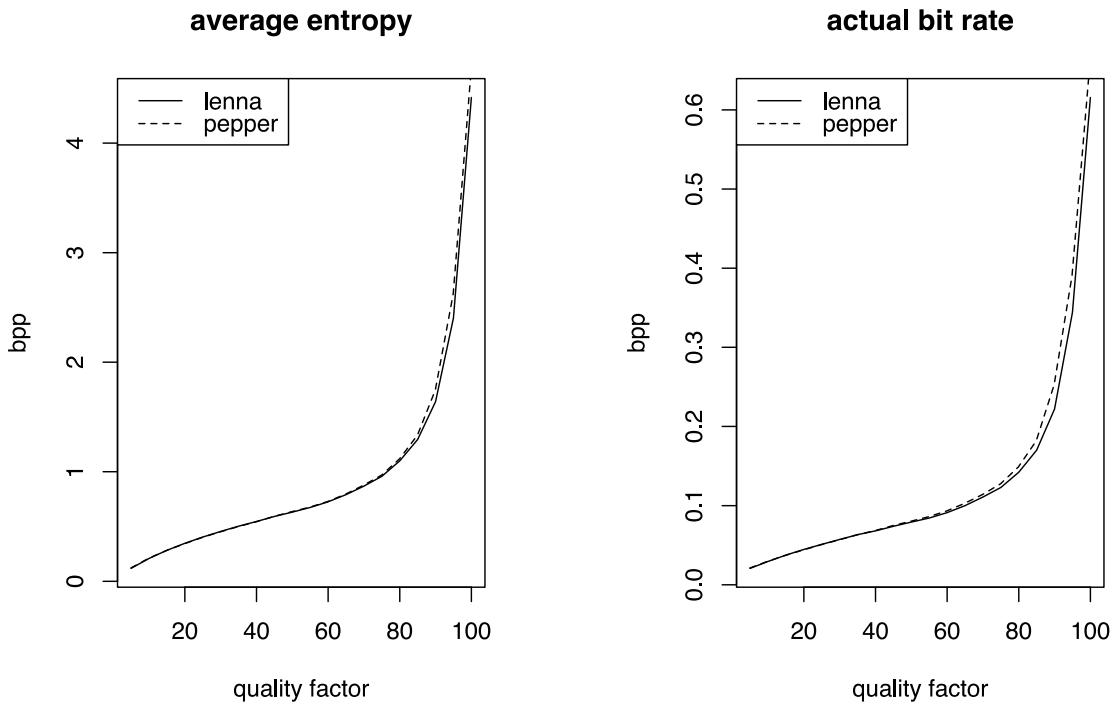


а)



б)

Рисунок 6.2 – Різні підходи до визначення бітової швидкості: середня вибіркова ентропія та фактичний бітрейт, а) Lenna.pgm; б) pepper.pgm



а) б)

Рисунок 6.3 – Ті самі залежності, що і на рис. 6.2, але для кожного із зображень окремо: а) Lenna.pgm; б) pepper.pgm

Оскільки аналітична формула ентропії не дуже добре працює в практичних застосунках кодування, в літературі були розроблені та використані більш досконалі формули швидкості, які застосовувалися для керування швидкістю у відеокодуванні [11]. Для точнішої оцінки функції швидкості також було запропоновано деякі операційні підходи. В [13], крива R - D (швидкість-спотворення) моделюється за допомогою експоненційної формули в поєднанні з кількома параметрами керування. Параметри моделі потім оцінюються на основі статистики кодування, отриманої шляхом повторного кодування вхідного відео. Легко бачити, що цей алгоритм має дуже високу обчислювальну складність. У відеокодуванні статистика кодування попередніх кадрів може бути використана для оцінки параметрів моделі поточного кадру, як в алгоритмі керування швидкістю VM7 [14, 15]. Таким чином зменшується обчислювальна складність. На жаль, це не так при змінах сцени. Тому цей

підхід часто страждає від погіршення продуктивності при змінах сцени [16]. Емпірична оцінка параметра моделі також може проводитися на рівні макроблоків (МБ), як в алгоритмі керування швидкістю TMN8 [16, 17]. Завдяки адаптивній схемі квантування на рівні МБ, алгоритм TMN8 має кращу продуктивність керування швидкістю порівняно з керуванням швидкістю у VM7 [23]. Однак через обмежену точність своєї моделі джерела, він також страждає від погіршення продуктивності для відео з високою динамікою руху.

6.2 Лінійна модель джерела та схема керування швидкістю

Переважна більшість моделей джерел, що були описані в літературі [11, 13, 14, 16] намагаються знайти найкращий вираз для бітрейту кодування R у термінах параметра квантування q. Іншими словами, функція швидкості визначається та моделюється в області q. Для підвищення точності моделі джерела вираз $R(q)$ стає все більш складним [23].

Добре відомо, що нулі відіграють ключову роль у трансформаційному кодуванні зображень та відео. Високий коефіцієнт стиснення в трансформаційному кодуванні в основному досягається за рахунок ефективного кодування нулів [2, 5, 8, 12, 18, 20]. Позначимо через ρ відсоток нулів серед квантованих коефіцієнтів перетворення. Зауважимо, що ρ монотонно зростає з q, що означає наявність взаємно однозначного відображення між ними [23]. Отже, математично, бітрейт кодування R також є функцією від ρ, що позначається як $R(\rho)$. В роботі [23] автори пропонують досліджувати функцію швидкості в області ρ замість традиційної області q. Також автори роботи [23] на основі багатьох чисельних симуляцій та теоретичного обґрунтування виявили, що у всіх типових системах трансформаційного кодування, таких як стиснення

зображень на основі вейвлетів [12, 21, 22], JPEG кодування зображень [2, 5], MPEG-2 [8], H.263 [18], та MPEG-4 відеокодування [20], $R(\rho)$ завжди є лінійною функцією

$$R(\rho) = \theta \cdot (1 - \rho), \tag{6.1}$$

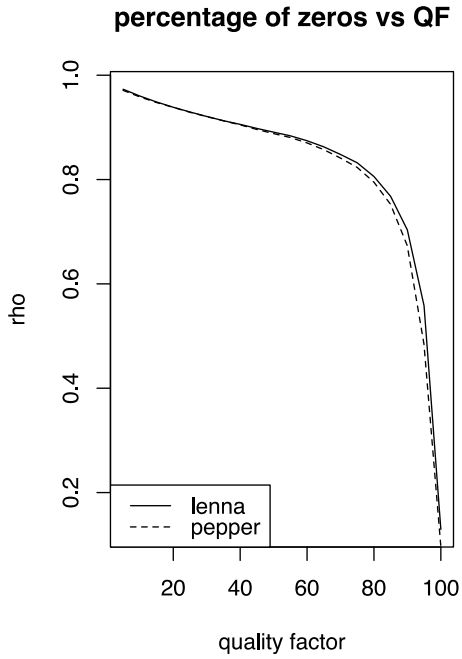
де θ — константа. Це призводить до уніфікованої моделі джерела для всіх типових систем трансформаційного кодування. Також в [23] відмічається, що єдиний параметр моделі θ безпосередньо пов'язаний зі змістом зображення. Визначивши значення параметра θ криву швидкості можна побудувати за допомогою (6.1). Як наслідок, в [23] пропонується уніфікований лінійний алгоритм керування швидкістю для відеокодування MPEG-2, H.263, та MPEG-4. Запропонований алгоритм є концептуально простим і має низьку обчислювальну складність і забезпечує досить точне та надійне керування швидкістю.

6.3 Аналіз швидкості в ρ-області

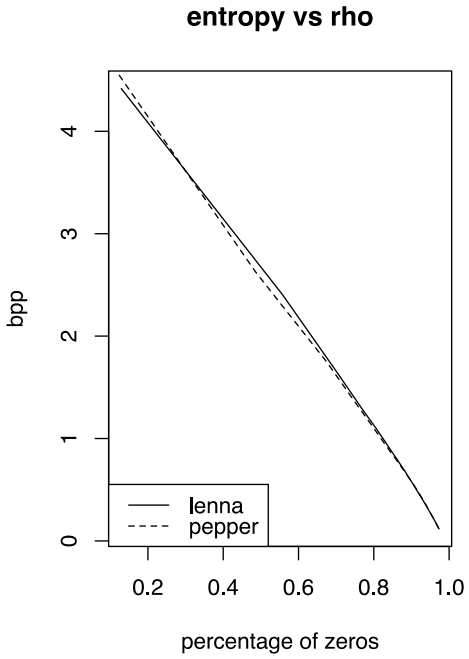
Як обговорювалося в попередніх підрозділах, якщо припустити, що коефіцієнти перетворення мають додатний розподіл, то, очевидно, існує взаємно однозначне відображення між q та ρ. Отже, будь-яка функція в q області може бути відображена в ρ області і навпаки. Це відображення можна легко обчислити з розподілу коефіцієнтів перетворення [23].

Для іллюстрації зазначених принципів побудуємо залежності (див. рис. 6.4). Аналіз залежностей вказує на те, що відсоток нульових ДКП коефіцієнтів залишається майже тим самим для кожного з двох тестових зображень. Крім того залежності лінійного типу, подібні до (6.1) мають місце як для середньої вибіркової ентропії, так і для значення фактичного

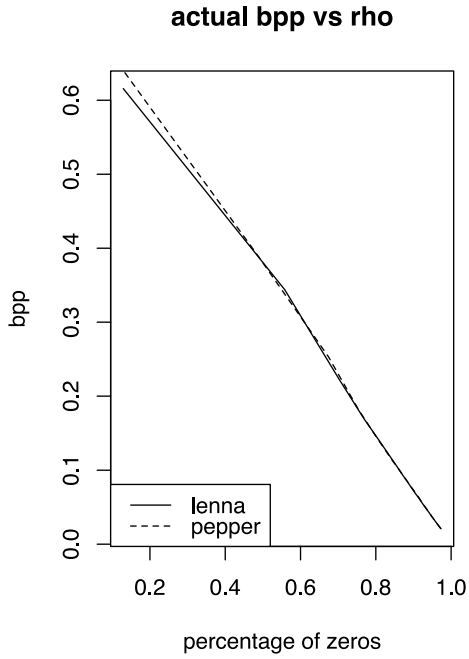
бітрейту, хоча залежності на рис. 6.2 на перший погляд свідчать про протилежне.



а)



б)



в)

Рисунок 6.4 – Залежності для тестових зображень Lenna.pgm і pepper.pgm: а) величини ρ від показника якості, б) і в) відповідно середньої ентропії та фактичного бітрейту від ρ

Схоже на те, що саме залежності, подібні до тих, які наведені на рис: 6.2, наштовхнули авторів роботи [23] на помилковий висновок про те, що середня вибіркова ентропія не може бути використана в ефективних алгоритмах управління керуванням швидкістю при стисненні зображень. Це підтверджують залежності на рис. 6.4 б) та в). Додатково зазначимо, що залежність на рис. 6.4 а) за формою нагадує залежності з рис. 6.2 та 6.3, але після дзеркального відображення відносно уявної горизонтальної вісі.

В таблиці 6.1 наведено значення параметру θ із співвідношення (6.1), визначеного за методом найменших квадратів для залежностей з рис. 6.4 б) та в), що відповідають різним тестовим зображенням.

Таблиця 6.1 - Значення параметру θ для залежностей з рис. 6.4 б) та в)

Зображення	Середня вибіркова ентропія	Фактичний бітрейт
Lenna.pgm	5,100	0,7195
Pepper.pgm	5,134	0,7426

Насамкінець зазначимо, що відповідні скрипти були написані на мовах програмування R і C++. Останній використовувався для спряження з бібліотекою libjpeg-turbo, що була задіяна для отримання значень фактичного бітрейту (тобто після квантування та кодування за Хаффманом).

ВИСНОВКИ

В будь-якій типовій системі кодування з перетворенням існує лінійна залежність між швидкістю кодування та процентним вмістом нулей серед квантованих коефіцієнтів перетворення, що позначаються як ρ. Спираючись на теорему Шеннона про кодування джерела можна надати теоретичне обґрунтування цієї лінійної моделі джерела.

Лінійна модель джерела, доповнена адаптивною схемою оцінки коефіцієнта пропорційності, призводить до уніфікованого алгоритму управління швидкістю для різних стандартних систем фото і відеокодування, таких як JPEG, MPEG-2, H.263, MPEG-4 та багатьох інших.

Схоже на те, що саме залежності, подібні до тих, які наведені на рис: 6.2, наштовхнули авторів роботи [23] на помилковий висновок про те, що середня вибіркова ентропія не може бути використана в ефективних алгоритмах управління керуванням швидкістю при стисненні зображень. Отримані в роботі залежності (див. рис. 6.4) спростовують це, оскільки можуть слугувати своєрідними контрприкладами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Pennebaker, William B., and Joan L. Mitchell (1992) JPEG Still Image Data Compression Standard, New York, Van Nostrand Reinhold.

2. Wallace, Gregory K. (1991) “The JPEG Still Image Compression Standard,” Communications of the ACM, 34(4):30–44, April.

3. Salomon, David (1999) Computer Graphics and Geometric Modeling, New York, Springer.

4. Osterberg, G. (1935) “Topography of the Layer of Rods and Cones in the Human Retina,” Acta Ophthalmologica, (suppl. 6):1–103.

5. Salomon, David (2007) Data Compression. The Complete Reference, London, Springer.

6. Сойфер В.А. (Ред.) (2003). Методы компьютерной обработки изображений. ФИЗМАТЛИТ.

7. Sklar, B. (2021). Digital communications: fundamentals and applications. Pearson.

8. Legall, D. (1993). A video compression standard for multimedia applications. Commun. ACM, 34, 226-252.

9. Gish, H., & Pierce, J. (1968). Asymptotically efficient quantizing. IEEE Transactions on Information Theory, 14(5), 676-683.

10. Berger, T. (2003). Rate□distortion theory. Wiley Encyclopedia of Telecommunications.

11. Hang, H. M., & Chen, J. J. (1997). Source model for transform video coder and its application. I. Fundamental theory. IEEE Transactions on Circuits and Systems for Video Technology, 7(2), 287-298.

12. Said, A., & Pearlman, W. A. (1996). A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on circuits and systems for video technology, 6(3), 243-250.

13. Ding, W., & Liu, B. (1996). Rate control of MPEG video coding and recording by rate-quantisation modeling. IEEE transactions on circuits and systems for video technology, 6(1), 12-20.

14. Chiang, T., & Zhang, Y. Q. (1997). A new rate control scheme using quadratic rate distortion model. IEEE Transactions on circuits and systems for video technology, 7(1), 246-250.

15. Lee, H. J., Chiang, T., & Zhang, Y. Q. (2000). Scalable rate control for MPEG-4 video. IEEE Transactions on Circuits and Systems for Video Technology, 10(6), 878-894.

16. Ribas-Corbera, J., & Lei, S. (1999). Rate control in DCT video coding for low-delay communications. IEEE Transactions on circuits and systems for video technology, 9(1), 172-185.

17. Ribas-Corbera, J., & Lei, S. (1997). Contribution to the rate control Q2 experiment: A quantizer control tool for achieving target bit rates accurately. Coding of Moving Pictures and Associated Audio MPEG96/M1812 ISO/IEC JTC/SC29/WG11.

18. Cote, G., Erol, B., Gallant, M., & Kossentini, F. (1998). H. 263+: Video coding at low bit rates. IEEE Transactions on circuits and systems for video technology, 8(7), 849-866.

19. Codec, T. E. L. N. O. R. (1995). Video codec test model, TMN5. ITU-T/GS-15.

20. Sikora, T. (1997). The MPEG-4 video standard verification model. IEEE Transactions on circuits and systems for video technology, 7(1), 19-31.

21. Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. IEEE Transactions on signal processing, 41(12), 3445-3462.

22. Tsai, M. J., Villasenor, J. D., & Chen, F. (1996). Stack-run image coding. IEEE Transactions on Circuits and Systems for Video technology, 6(5), 519-521.

23. He, Z., & Mitra, S. K. (2002). A linear source model and a unified rate control algorithm for DCT video coding. IEEE transactions on Circuits and Systems for Video Technology, 12(11), 970-982.