

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Метод стиснення зображень для цифрових баз даних

Виконав(ла): студент(ка) 6 курсу, групи РРм-61
спеціальності 172 Телекомунікації та радіотехніка

(шифр і назва спеціальності)

(підпис)

Грималовський Р.В.

(прізвище та ініціали)

Керівник

(підпис)

Дедів І.Ю.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Хвостівська Л.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Дунець В.Л.

(прізвище та ініціали)

Рецензент

(підпис)

Стрембіцький М.О.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра радіотехнічних систем
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Дунець В.Л.

(підпис)

(прізвище та ініціали)

« »

2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня магістр

(назва освітнього ступеня)

за спеціальністю 172 Телекомунікації та радіотехніка

(шифр і назва спеціальності)

студенту Грималовському Руслану Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Метод стиснення зображень для цифрових баз даних

Керівник роботи Дедів Ірина Юріївна, к.т.н., доц.

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

Затверджені наказом ректора від «30» листопада 2021 року № 4/7-1019

2. Термін подання студентом завершеної роботи 6 грудня 2021 року

3. Вихідні дані до роботи Технічне завдання, методи стиснення зображень

4. Зміст роботи (перелік питань, які потрібно розробити)

Задача обробки зображень. Типи надлишковості зображень.

Основи стиснення зображень.

Кодова надмірність. Коди Хаффмана. Моделювання усунення надлишковостей. Стандарт стиснення JPEG. Модифікований метод JPEG. Метод стиснення зображень.

Охорона праці. Безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання на кваліфікаційну роботу		
2	Написання розділу 1		
3	Написання розділу 2		
4	Написання розділу 3		
5	Написання розділу 4		
6	Попередній захист		
7	Захист		

Студент _____
(підпис)

Грималовський Руслан Віталійович
_____ (прізвище та ініціали)

Керівник роботи _____
(підпис)

Дедів Ірина Юрїївна
_____ (прізвище та ініціали)

АНОТАЦІЯ

Метод стиснення зображень для цифрових баз даних // Кваліфікаційна робота магістра // Грималовський Руслан Віталійович // ТНТУ, ФПТ, група РРм-61 // Тернопіль, 2021.

Ключові слова: ЗОБРАЖЕННЯ, НАДЛИШКОВІСТЬ, КОД ЗМІННОЇ ДОВЖИНИ.

В кваліфікаційній роботі магістра розглянуто метод стиснення зображень для цифрових баз даних. Розглянуто задачу стиснення зображень для усунення або зниження рівня міжпіксельної, кодової чи візуальної надлишковості в системах зберігання чи передачі даних. Розглянуто способи стиснення зображень зокрема з використанням кодів змінної довжини. Змодельовано модифікований метод стиснення зображень в середовищі Matlab, який ґрунтується на використанні вейвлет-перетворення, на відміну від інших стандартів стиснення, які використовують коди змінної довжини. Показано виграш та переваги змодельованого методу.

ANNOTATION

Image compression method for digital databases // Master's qualification work
// Grimalovsky R.V. // TNTU, FPT, group RRm-61 // Ternopil, 2021.

Keywords: IMAGE, REDUNDANCY, VARIABLE LENGTH CODE.

In the qualification work of the master the method of image compression for digital databases is considered. The problem of image compression to eliminate or reduce the level of interpixel, code or visual redundancy in storage or data transmission systems is considered. Methods of image compression in particular using variable length codes are considered. A modified method of image compression in the Matlab environment, which is based on the use of wavelet transform, is modeled, unlike other compression standards that use variable length codes. The benefits and advantages of the simulated method are shown.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА.....	9
1.1. Задача обробки зображень.....	9
1.2. Типи надлишковості зображень.....	11
1.3. Висновки до розділу 1	19
РОЗДІЛ 2. ОСНОВНА ЧАСТИНА.....	21
2.1 Основи стиснення зображень.....	21
2.2. Кодова надмірність.....	24
2.3. Коди Хаффмана.....	28
2.4 Висновки до розділу 2	31
РОЗДІЛ 3. НАУКОВО-ДОСЛІДНА ЧАСТИНА	33
3.1. Моделювання усунення надлишковостей.....	33
3.2. Стандарт стиснення JPEG.....	38
3.3. Модифікований метод JPEG.....	45
3.4. Метод стиснення зображень.....	49
3.5 Висновки до розділу 3	52
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	53
4.1 Охорона праці.....	53
4.2 Безпека в надзвичайних ситуаціях.....	57
4.3 Висновки до розділу 3.....	62
ВИСНОВКИ.....	63
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ	

ВСТУП

Актуальність роботи. Розробка систем обробки зображень завжди зв'язана з великим обсягом тестування та експериментальної роботи, який необхідно виконати для досягнення прийняттого вирішення поставленого завдання. Актуальною сьогодні є задача стиснення зображень, що викликана значною надлишковістю зображень та необхідністю у значних обсягах пам'яті баз даних для їхнього зберігання чи значних швидкостей передачі даних.

Проаналізовано основні типи надлишковості зображень в телекомунікаційних системах, такі як структурна, статична та візуальна (психофізична) надлишковість, та методи стиснення зображень для зменшення таких надлишковостей. Існують різні способи стиснення сигналів зображення. Насамперед, стиснення статичних і динамічних зображень може бути здійснено двома основними методами — із втратами або без втрат. У процесі стиснення із втратами частина інформації втрачається.

Встановлено, що для кодування статичних зображень особливо поширеним сьогодні є застосування методу кодування, який полягає у застосуванні кодів з мінливою довжиною кодового слова. Суть такого кодування полягає у присвоєнні словам із найбільшою ймовірністю появи коротких кодових комбінацій, а словам, ймовірність появи яких мала – довших кодових комбінацій. Найвідоміший із таких кодів — код Хаффмана. Однак його використання передбачає виконання значної кількості обчислень, що вимагає певних обчислювальних потужностей та затрат часу.

В роботі проводиться розробка методу стиснення зображень для цифрових баз даних.

Мета і завдання дослідження. Розроблення методу стиснення зображень для цифрових баз даних. Задачі:

- аналіз джерел по темі дослідження;
- аналіз відових принципів, підходів та методів кодування статичних та динамічних зображень;
- аналіз структури кодера та декодера зображень;
- моделювання методу стиснення зображень та аналіз його ефективності.

Об'єкт дослідження: процес стиснення зображень для цифрових баз даних..

Предмет дослідження: метод стиснення зображень для цифрових баз даних.

Практичне значення отриманих результатів. Отримані результати можуть бути використані при розробленні додатків швидкого стиснення зображень для цифрових баз даних.

Наукова новизна. Вперше застосовано метод кодування контрастності поряд із методом стиснення зображень на основі вейвлет-перетворення.

Апробація результатів дослідження. Участь в ІХ науково-технічній конференції «Інформаційні моделі, системи та технології» (м. Тернопіль, 2021 р.).

РОЗДІЛ 1

АНАЛІТИЧНА ЧАСТИНА

1.1 Задача обробки зображень

Розробка систем обробки зображень завжди зв'язана з великим обсягом тестування та експериментальної роботи, який необхідно виконати для досягнення прийняттого вирішення поставленого завдання. Ця особливість означає, що вміння будувати прототипні моделі можливих рішень відіграє важливу роль зменшення грошових і тимчасових витрат і отримання конкурентоспроможної реалізації необхідної системи.

Зір є найдосконалішим з наших органів чуття, тому зорові образи відіграють таку важливу роль у людському сприйнятті. Однак, машинне опрацювання зображень включає практично весь діапазон хвиль від гамма-випромінювання до радіохвиль. Причому оброблювані зображення можуть походити від таких джерел, які людям навіть важко асоціювати зі звичними зображеннями, що спостерігаються.

Немає межі, яка б розділяла область обробки зображень та інші суміжні дисципліни. Це відокремлення виробляється з припущення про наявність зображень на вході та виході цієї системи. Однак таке визначення є нам невиправдано обмежувальним і штучним. Насправді, при такому підході навіть таке просте завдання, як визначення середньої інтенсивності по всьому полю зображення (потрібне знайти одне однини), не буде вважатися операцією з обробки зображення. Попри те, є такі галузі, як машинний зір, що включає здатність до навчання, логічним висновкам та вчиненню певних дій на основі інформації, що оглядається. Такі завдання відносяться вже до галузі ШІ для копіювання та імітація інтелектуальної діяльності. Зазначимо, що ця галузь

штучного інтелекту перебуває лише на початковій стадії розвитку, і доводиться визнати, що прогрес у цьому напрямі йде значно повільніше, ніж передбачалося раніше. Аналіз зображення (який іноді називають "інтерпретацією" або "розумінням" зображень) знаходиться десь посередині між обробкою зображень та машинним зором.

У всьому різноманітті завдань виділяють низькорівневі, середньорівневі та високорівневі процеси. В першому випадку проводиться підвищення контрастності або покращення різкості зокрема. Ці процеси характеризуються тим, що на вхід та вихід надходять зображення. В другому випадку розглядаються такі задачі, як сегментація (розподіл зображень на області та виділення в них об'єктів), опис об'єктів та їх стиснення для надання їм зручної форми для подальшої комп'ютерної обробки, а також класифікація (розпізнавання) виділених об'єктів. У середньорівневих процесах на вході є зображення. Вихідними величинами є певні ознаки, витягнуті з вихідних зображень. Нарешті, в третьому випадку проводиться «осмислення» безлічі розпізнаних об'єктів, як це робиться в аналізі зображень, а ще далі стоять операції, пов'язані з когнітивними функціями, які зазвичай асоціюються з людським зором.

Маючи на увазі перелічені вище зауваження, видно, що природним полем перетину та переходу від опрацювання зображень до їх аналізу є область розпізнавання окремих фігур або об'єктів на зображеннях. Діями з цифрової обробки зображень є зчитування області зображення з текстом, попередня обробка цього зображення, виділення (сегментація) окремих літер та знаків тексту, опис цих символів у зручній комп'ютерній формі та, нарешті, розпізнавання кожного індивідуального символу тексту. Якщо ж говорити про осмислення вмісту, то це завдання вже можна віднести до області аналізу зображень чи машинного зору залежно від рівня складності інформації, очікуваної від прочитаного тексту. Цифрова обробка інформації у тому вигляді,

як вона була визначена вище, успішно застосовується у широкому колі областей, що мають важливе соціально-економічне значення.

Задача стиснення зображень викликана значною надлишковістю зображень та необхідністю у значних обсягах баз даних для їхнього зберігання чи значних швидкостей передачі даних.

1.2 Типи надлишковості зображень

Розрізняють такі типи надлишковості:

- структурну;
- статичну;
- візуальну (психофізичну).

Структурна надмірність (надлишковість) визначається стандартом. При цифровій передачі зображення немає необхідності використання кадрових і малих гасящих імпульсів, тобто обсяг інформації, що передається в секунду, можна скоротити за рахунок виключення пасивної частини повного сигналу.

Статистична надмірність зображення пов'язана з кореляцією, а отже, і передбачуваністю даних між пікселями в одному рядку, у сусідніх рядках і кадрах. Така надлишковість вилучається без втрати інформації, а вихідне зображення повністю відновлено. Так, більшість зображень одного кадру займає задній план, де яскравість постійна чи мало змінюється у просторі. Зображення в сусідніх кадрах також маловиразні навіть за наявності зображень, що рухаються. Знаючи розподіл яскравості в одному рядку (кадрі) можна передбачити розподіл яскравості в наступному.

Візуальна (психофізична) надмірність пов'язана з використанням властивостей зору, що дозволяють усунути частину інформації про зображення, що несуттєво впливає на його сприйняття. Так, шуми квантування розрізняються оком на великих деталях зображення як хибні контури, проте

малопомітні на дрібних деталях і різких перепадах яскравості. Це дозволяє вводити більш грубе квантування відеосигналу в околицях переходів яскравості та високочастотних компонентів.

Існують різні способи стиснення сигналів зображення. Насамперед, стиснення статичних і динамічних зображень може бути здійснено двома основними методами — із втратами або без втрат. У процесі стиснення із втратами частина інформації втрачається.

До групи методів, що забезпечують компресію із втратами, відносяться:

- кодування з передбаченням, у якому використовуються різні варіанти диференціальної ІКМ, зокрема і адаптивні ІКМ;
- лінійне кодування з перетворенням;
- міжкадрове кодування;
- проріджування відліків та кадрів;
- кодування із розчленуванням зображення на фрагменти;
- гібридне, або комбіноване, що представляє собою різні поєднання перерахованих методів кодування з перетворенням, та деякі інші.

До групи методів статичного стиснення без втрат входять:

- статичні коди, наприклад, код Хаффмана та його модифікації, код Джелінека, арифметичні коди;
- векторне квантування та деякі інші.

У пристроях стиснення відео зазвичай використовуються різні комбінації з першої та другої груп.

Крім описаних вище двох груп методів, застосовується ще й фрактальний метод стиснення зображення.

З метою компресії зображень використовуються одновимірні, двовимірні (просторова) та тривимірні (просторово-часова) обробка даних. При одновимірній обробці використовуються різні варіанти диференціальної ІКМ та статистичного кодування, що ґрунтуються на обрахунку кореляції між

сусідніми елементами зображення по горизонталі (по рядку). При двовимірній обробці враховується кореляція між сусідніми елементами по обох напрямках - по горизонталі та вертикалі. При тривимірній обробці враховується також кореляція між елементами з однаковими координатами сусідніх кадрів (часова кореляція).

У всіх стандартах компресії статичних і динамічних зображень використовуються комбінації різних методів стиснення. При цьому основу внутрішньокадрової компресії складають методи кодування з перетворенням, для міжкадрової компресії застосовують різні варіанти диференціальної ІКМ, сформований потік даних додатково піддається кодуванню статистичними кодами. В результаті ступінь стиснення зображень (відношення об'єму або швидкості передачі даних на вході компресора до вихідного об'єму або швидкості) може досягати десятків і навіть сотень разів при досить хорошій якості зображення, що відтворюється.

Дамо коротку характеристику методів кодування цифрового потоку з метою стиснення зображення, що застосовуються замість ІКМ.

1. Статичне кодування зображень.

Методи статичної надмірності дають стиснення зображення трохи більше 3/1, проте забезпечують точне відновлення зображень.

До статичного методу кодування відноситься застосування кодів з довжиною кодового слова. Суть такого кодування полягає у присвоєнні словам із найбільшою ймовірністю появи коротких кодових комбінацій, а словам, ймовірність появи яких мала – довших кодових комбінацій. Найвідоміший із таких кодів — код Хаффмана. Наприклад, необхідно закодувати чотири літери алфавіту - а, в, т, ц. Якщо закодувати їх 3-розрядним кодом, то отримаємо код а - 001; в - 010; т - 011; ц - 100. Для передачі слова, що складається з цих літер, необхідно 12 біт. Однак ймовірність появи літери а - 0,8, в - 0,5, т і ц - 0,1. Застосуємо цей спосіб кодування. Надамо букві а значення 0, в - 01, т - 010 і ц -

011. При такому кодуванні для передачі слова в середньому необхідно $1 \cdot 0,8 + 2 \cdot 0,5 + 2 \cdot 3 \cdot 0,1 = 2,4$ біта, що значно скоротило кількість бітів, отриманих у попередньому варіанті. За своєю ефективністю код наближається до теоретичної межі витрати бітів і тому називається ентропійним. Адаптивна версія коду Хаффмана застосовується у разі, коли ймовірність появи кодівих слів змінюється у процесі передачі.

До групи ентропійних належить арифметичний код. Сутність даного кодування полягає в тому, що всій сукупності символів повідомлення відповідає певний інтервал. Цей інтервал розбивається на ділянки, що відповідають новим ймовірностям появи символів, і це розбиття повідомляє декодер. Після надходження чергового символу інтервал перераховується в нові межі, відповідні ймовірності появи цього символу, і знову розбивається пропорційно вихідним ймовірностям. З надходженням кожного нового символу розмір інтервалу зменшується, причому у суворій відповідності до ймовірностей символів. Найчастіше зустрічаються символи, що менше звужують інтервал, ніж рідкісні, і додають менше бітів коду інтервалу. Після закінчення циклу кодування формується деякий вузький інтервал, що однозначно характеризує послідовність символів, що передається, і вона легко може бути відновлена в декодері за будь-яким числом з цього інтервалу.

2. Кодування із перетворенням.

Застосовується кілька варіантів кодування з перетворенням, що засновані:

- на спектральному перетворенні з використанням перетворення Фур'є;
- дискретному косинусному перетворенні (ДКП);
- вейвлет-перетворенні.

Розглянемо принцип кодування, що базується на спектральному перетворенні з використанням перетворення Фур'є.

ДКП є модифікацією дискретного перетворення Фур'є. При використанні ДКП обробка проводиться блоками 8×8 пікселів. У середньому розмір блоку відповідає інтервалу кореляції елементів зображення. В результаті виконання ДКП формується матриця з 64 коефіцієнтів, що характеризують просторові частоти (двовимірні – x, y) функції яскравості зображення.

Після операції ДКП коефіцієнти можуть набувати не цілі значення, проте подальша обробка вимагає квантованого представлення цих коефіцієнтів. Процедура квантування коефіцієнтів проводиться з урахуванням психофізичних особливостей зору. Як зазначалося, зорове сприйняття допускає більший рівень шумів і помилок квантування у сфері зображень із великим рівнем високочастотних компонент. Це означає, що коефіцієнти цих компонентів можна квантувати на малу кількість рівнів. Постійна складова та амплітуди низькочастотних компонентів квантуються на велику кількість рівнів і передаються з високою точністю.

Фінальною операцією при квантуванні є Z -упорядкування, при якому коефіцієнти, що залишилися, вибудовуються в послідовності зростання просторових частот. Якщо просторові частоти однакові, попереду ставляться коефіцієнти для менших вертикальних частот. З низки коефіцієнтів утворюються пари чисел, одне з яких рівне ненульовому коефіцієнту, інші — числу попередніх цьому елементу нулів. Далі ці поєднання чисел кодуються кодом Хаффмана. Спрощена структурна схема алгоритму внутрішньокадрового кодування та декодування на основі ДКП наведена на рис. 1.1.

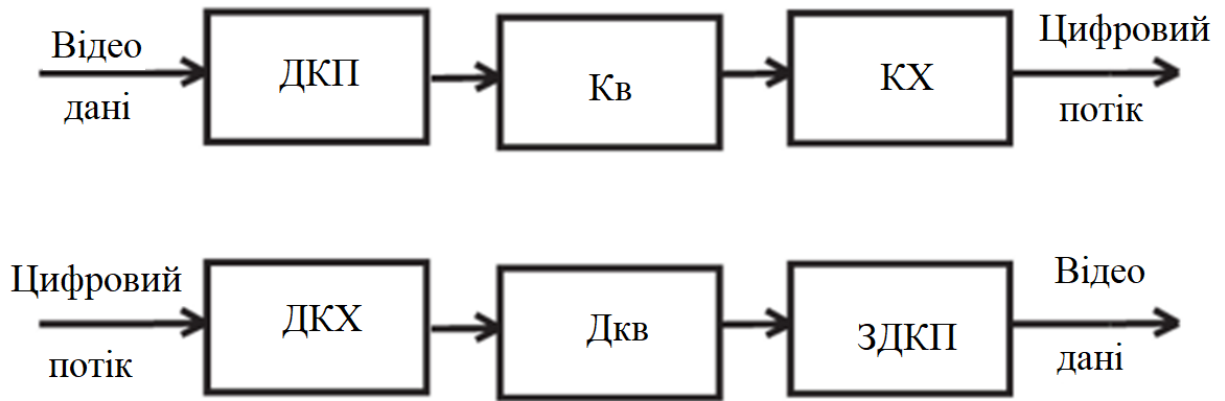


Рис. 1.1. Структура алгоритму внутрішньокадрового кодування та декодування на основі ДКП

Кв - квантувач; КХ - кодер Хаффмана; ДКХ - декодер Хаффмана; Дкв – деквантувач; ЗДКП - зворотне ДКП

При Wavelet-перетворенні (рис. 1.2) так само, як і при ДКП, здійснюється перехід з площини зображення в двовимірну частотну область. На відміну від ДКП, цей перехід здійснюється за допомогою гребінки цифрових фільтрів. Блокова структура шумів квантування найяскравіше виявляється при високих ступенях стиснення, що становлять найбільший інтерес у додатках до спеціальної техніки.

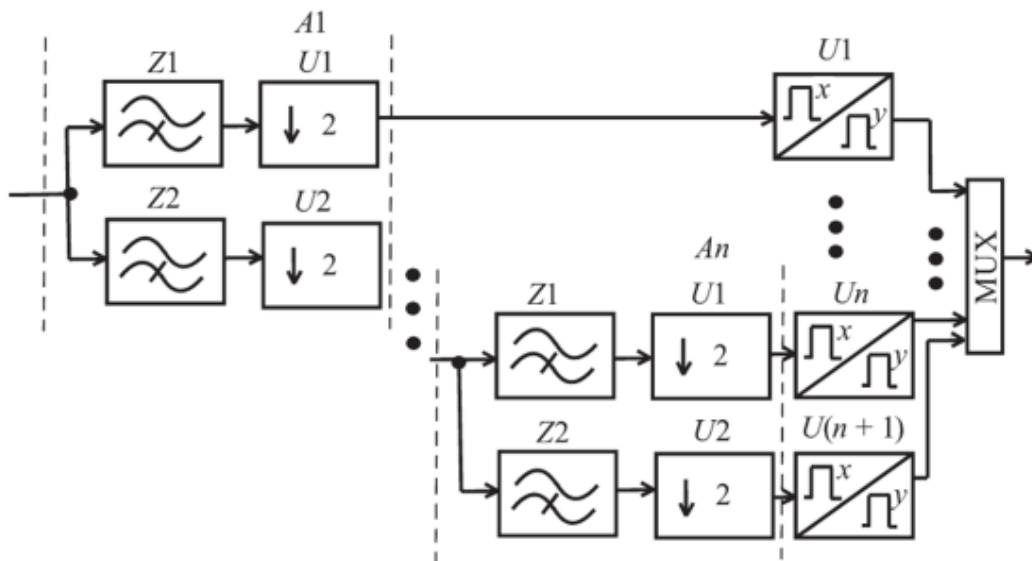


Рис. 1.2. Блок Wavelet-перетворення: A_1 - A_n субсмугові фільтри

Сигнал зображення розділяється за спектром на дві рівні частини за допомогою фільтрів нижніх та верхніх частот. Оскільки НЧ- і ВЧ-компоненти мають смугу вдвічі меншу, ніж у вихідного сигналу, їх частота дискретизації може бути знижена вдвічі. Вхідний сигнал є цифровим, тому після цифрової НЧ і ВЧ-фільтрації проводиться децимація (виключення кожного другого відліку).

Після кожного з трьох етапів перетворення зображення ділиться на одну низькочастотну та три високочастотні компоненти. Перша високочастотна компонента відображає горизонтальні високочастотні складові зображення. Друга високочастотна компонента відображає вертикальні складові, третя пов'язана з діагональними просторовими частотами та відображає яскраві переходи.

Оскільки після кожної процедури фільтрації кількість відліків на виході фільтра зменшується вдвічі, то результуюча кількість відліків на виході всієї гребінки фільтрів дорівнює кількості відліків у вихідному зображенні. Таким чином, при Wavelet-перетворенні кадр зображення переводиться в частотну область, де описується такою ж кількістю відліків, як і до перетворення. Для кодування спектральних відліків зображення, отриманих в результаті Wavelet-перетворення, використовуються ті самі принципи, що і за ДКП.

Для відновлення вихідного зображення кожна компонента розтягується, тобто після кожного відліку вставляється нульовий відлік.

Потім здійснюється аналогічна розглянутій вище НЧ- та ВЧ-фільтрація, в результаті якої нульові відліки замінюються на інтерпольовані.

Структура побудови схеми внутрішньокadresного стиснення на основі Wavelet-перетворення показана на рисунку 1.3.

Процедури квантування коефіцієнтів та статичного кодування на основі кодів Хаффмана виконуються так само, як і за ДКП.

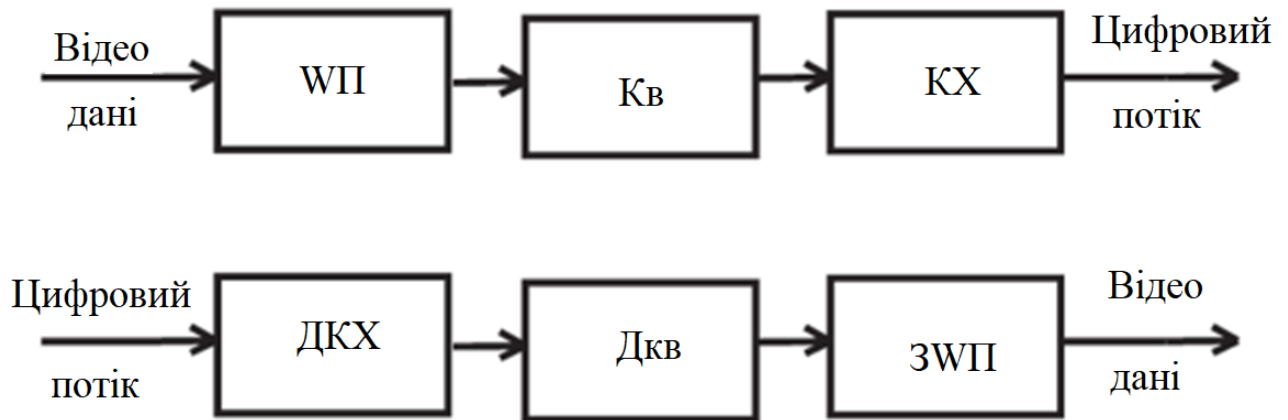


Рис. 1.3. Структурна схема алгоритму внутрішньокадрового стиску на основі Wavelet-перетворення: ВП - Wavelet-перетворення; ОВП - зворотне Wavelet-перетворення

3. Кодування із передбаченням.

За наявності сильних корекційних зв'язків між розташованими елементами просторової дискретизації можна обмежитися передачею низки елементів зображення, проте інші передбачити, тобто обчислити їх з допомогою спеціальних технічних пристроїв. Для використання такого методу застосовують диференціальну ІКМ. При цьому замість передачі значення X яскравості n -го пікселя передається різниця між X_n та його оцінкою (рис. 1.4). Величина E_n називається помилкою передбачення:

$$E_n = X_n - \hat{X}_n,$$

де \hat{X}_n - передбачене значення яскравості.



Рис. 1.4. Схема, яка пояснює алгоритм диференціальної імпульсно-кодової модуляції

Оцінка формується з використанням передбачувача, алгоритм роботи якого повинен бути узгоджений зі статистикою зображення, що передається. Найбільшого поширення набули методи лінійного передбачення.

Для кодування простих зображень використовується дельта-модуляція. При цьому передбачення проводиться тільки по одному пікселю, а квантування помилки передбачення проводиться на два рівні, тобто кожен піксель кодується одним бітом замість 8.

1.3 Висновки до розділу 1

В розділі проаналізовано задачу стиснення зображень, що викликана значною надлишковістю зображень та необхідністю у значних обсягах баз даних для їхнього зберігання чи значних швидкостей передачі даних. Проаналізовано такі типи надлишковості зображень в телекомунікаційних системах, як структурну, статичну та візуальну (психофізичну).

Існують різні способи стиснення сигналів зображення. Насамперед, стиснення статичних і динамічних зображень може бути здійснено двома основними методами — із втратами або без втрат. У процесі стиснення із втратами частина інформації втрачається.

Детальніше розглянуто статичний метод кодування, застосовний до зображень, які зберігатимуться в базах даних. До статичного методу кодування відноситься застосування кодів з довжиною кодового слова. Суть такого кодування полягає у присвоєнні словам із найбільшою ймовірністю появи коротких кодових комбінацій, а словам, ймовірність появи яких мала – довших кодових комбінацій. Найвідоміший із таких кодів — код Хаффмана.

РОЗДІЛ 2

ОСНОВНА ЧАСТИНА

2.1 Основи стиснення зображень

Системи стиснення зображень включають два структурних блоки: кодер і декодер. Зображення $f(x,y)$ подається на вхід кодера, який переводить вхідні дані на певний набір символів і використовує його для представлення вихідного зображення. Нехай n_1 і n_2 позначають кількість елементарних носіїв інформації (наприклад, бітів) вихідного та закодованого зображення. Тоді міру стиснення можна висловити кількісно за допомогою коефіцієнта стиснення $C_R = n_1/n_2$. Коефіцієнт стиснення 10 вказує на те, що вихідне зображення має 10 елементів зберігання інформації (тобто бітів) на кожен елемент зберігання в стислому наборі даних. У MATLAB приватне число біт для зберігання та представлення двох файлів зображень та/або змінних, можна обчислювати за допомогою наступної М-функції, скрипт якої наведено в додатку А.

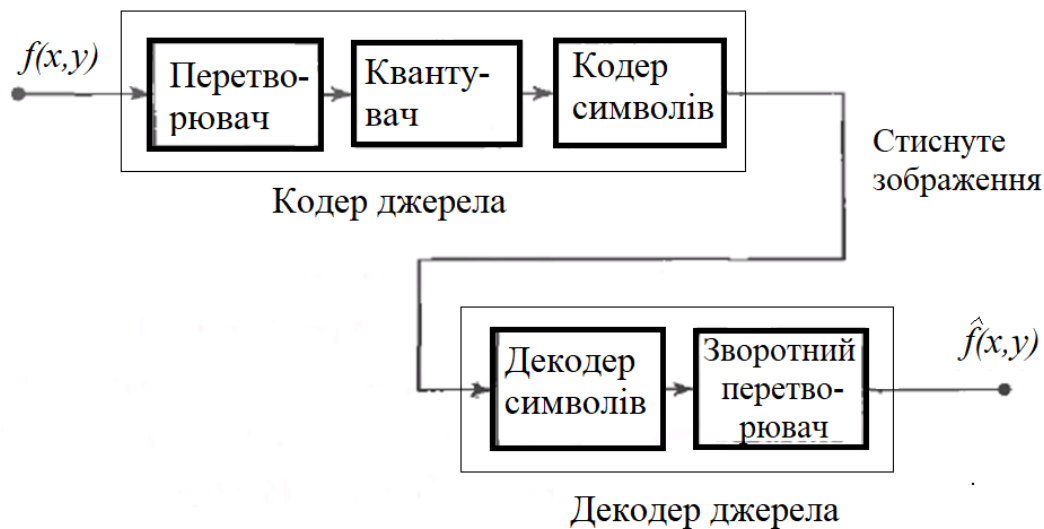


Рис. 2.1. Блок-схема модельної системи стиснення зображень

Наприклад, коефіцієнт стиснення зображення при кодуванні JPEG можна обчислити за допомогою команди, скрипт якої наведено нижче.

```
r=imratio(imread('bubbles25.jpg'),'bubbles25.jpg')
r = 35.1612
```

Зауважимо, що у `imratio` зовнішня функція `b = bytes (f)` написана так щоб вона повертала число байтів: (1) у файлі, (2) у структурній змінній, та/або (3) у змінній, що не є структурною.

Якщо `f` не є структурною змінною, функція `whos` повертає розмір цієї змінної в байтах. Якщо `f` - це ім'я файлу, то аналогічну дію виконує функція `dir`. У формі синтаксису функція `dir` повертає структуру, яка має поля `name`, `date`, `bytes` і `isdir`. У цих полях записані відповідно ім'я файлу, дата його останньої зміни, розмір у байтах та інформація про те, чи є `f` папкою (директорією) (`isdir = 1`, якщо «так», інакше `isdir = 0`). Нарешті, якщо `f` є структурою, то функція `bytes` викликає рекурсивно себе для складання числа байтів, використуваних при зберіганні кожного окремого поля структури. Це дозволяє виключити обсяг службової інформації, пов'язаної з структурною змінною (124 байта зберігання на одне поле), коли повертається тільки число байтів, необхідних для зберігання самих даних, записаних у полях. Функція `f ieldename1` використовується для отримання списку полів `f`, а оператори

```
for k=1:length(fields)
b = b + bytes (f .(fields{k}));
```

здійснюють рекурсію. Варто звернути увагу на використання динамічних імен полів структур у рекурсивних викликах функції `bytes`. Якщо `S` – це структура, а `F` – змінна символьного рядка, що містить ім'я поля, то оператори

$$S.(F) = \text{foo}; \text{field} = S.(F);$$

Використовують синтаксис динамічних імен полів структур привласнення та/або прочитання вмісту поля `F` структури `S`.

Щоб побачити та/або використовувати стиснуте (тобто закодоване) зображення, його необхідно подати на декодер, який побудує реконструйоване зображення $\hat{f}(x, y)$. У загальному випадку, зображення $\hat{f}(x, y)$ може співпадати з вихідним зображенням $f(x, y)$ або може від нього відрізнитись. У першому випадку система називається вільною від помилок або системою кодування або стиснення без втрати інформації. В іншому випадку в реконструйованому зображенні присутні певні спотворення та втрати і система називається кодуванням або стисненням із втратою інформації. Помилку $e(x, y)$ між $f(x, y)$ і $\hat{f}(x, y)$ можна обчислити для будь-яких пікселів (x, y) за формулою $e(x, y) = \hat{f}(x, y) - f(x, y)$, а величина загальної помилки між двома зображеннями дорівнює

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]$$

Середньоквадратичне відхилення e_{rms} між $f(x, y)$ і $\hat{f}(x, y)$ рівне квадратному кореню із середньої квадратичної помилки по всій матриці $M \times N$, тобто

$$e_{\text{rms}} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\hat{f}(x, y) - f(x, y))^2 \right]^{1/2}.$$

Наступна М-функція обчислює e_{rms} і будує (якщо $e_{\text{rms}} \neq 0$) зображення $e(x, y)$ та її гістограму. Оскільки матриця $e(x, y)$ може містити як позитивні, і негативні величини, замість `imhist` (яка працює лише з зображенням) використовується функція `hist` для побудови гістограми.

Зауважимо, що кодер (рис. 2.1) відповідає за скорочення всіх трьох типів надмірності (кодової, міжпіксельної та візуальної) вихідного зображення. На першому кроці кодування перетворювач трансформує вхідне зображення деякий (невізуальний) формат, пристосований для їзнення міжпіксельної надмірності. На другій стадії блок квантувач знижує точність перетворювача відповідно до раніше обумовленого критерію точності. На цій стадії відбувається скорочення візуальної надмірності, тобто деяке огрублення зображення, майже непомітне, однак, для ока. Ця процедура є незворотною, тому не виконується, якщо необхідно здійснити стиснення без втрати інформації. На третій та останній стадії процесу кодер символів будує оптимальний код (який скорочує кодову надмірність) для виходу квантувача і перетворює вихідну послідовність відповідно до побудованого коду.

Декодер на рис. 2.1 має лише дві компоненти: декодер символів та зворотний перетворювач. Виконуються ними дії, зворотні діям блоку кодера символів та блоку перетворювача. Оскільки процедура квантування не є оборотною, блок зворотного квантувача відсутня у загальній схемі декодера.

2.2. Кодова надмірність

Припустимо, що величина r_k при $k = 1, 2, \dots, L$ з ймовірностями $p_r(r_k)$ відповідає розподілу рівнів напівтонового зображення, загальна кількість яких

дорівнює L . Передбачається, що r_1 відповідає нульовому рівню (індекси масивів у MATLAB починаються з одиниці). Нехай також

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 1, 2, \dots, L,$$

де n_k дорівнює кількості появи на зображенні пікселів r_k -го рівня сірого кольору, а n - загальна кількість пікселів зображення. При кількості бітів, які застосовуються у цифровому поданні кожної величини r_k , рівній $l(r_k)$, то середня кількість бітів, необхідне для представлення кожного пікселю, дорівнює

$$L_{avg} = \sum_{k=1}^L l(r_k) p_r(r_k).$$

Таким чином, довжина кодових слів, котрі призначаються різним значенням рівнів сірого кольору, є підсумовуванням добутків числа бітів уявлення кожного рівня на ймовірність виникнення такого рівня. Значить, загальна кількість бітів для кодування зображення розмірами $M \times N$ дорівнює MNL_{avg} .

Таблиця 2.1. Ілюстрація кодової надмірності: $L_{avg} = 2$ для Коду 1 і $L_{avg} \simeq 1.81$ для Коду 2

r_k	$p_r(r_k)$	Код 1	$l_1(r_k)$	Код 2	$l_2(r_k)$
r_1	0.1875	00	2	011	3
r_2	0.5000	01	2	1	1
r_3	0.1250	10	2	010	3
r_4	0.1875	11	2	00	2

Якщо всі рівні зображення представлені m -бітовим двійковим кодом, то права частина останньої рівності, очевидно, зводиться до бітів. Справді, $l(r_k) = m$ для будь-якого, r_k і цей множник виноситься за знак підсумовування, а сума $p_r(r_k)$ стає рівною одиниці, і в результаті $L_{avg} = m$. З табл. 2.1 слід, що кодова надмірність майже завжди є присутньою, коли лінії градації кольору кодуються звичайними двійковими кодами фіксованої довжини. У цій таблиці дані обидві схеми кодування пікселів чотирирівневого зображення, розподіл рівнів якого дано у другому стовпці. Кодування кожного рівня двома бітами (Код 1 третьому стовпці таблиці) дає усереднену величину кодового слова, рівну 2 бітам. Середня довжина коду пікселя при кодуванні Кодом 2 (п'ятий стовпець таблиці) дорівнює

$$L_{avg} = \sum_{k=1}^4 l(r_k) p_r(r_k) = 3 \cdot 0.1875 + 1 \cdot 0.5 + 3 \cdot 0.125 + 2 \cdot 1.875 = 1.8125,$$

і досягнутий коефіцієнт стиснення стає рівним $C_R = 2/1.8125 \simeq 1.103$. Ефект стиснення при використанні Коду 2 досягається за рахунок того, що кодові слова мають змінну довжину, і це дозволяє призначати короткі коди значенням пікселів, що частіше за інших з'являються на зображенні.

Тоді виникає природне питання: скільки бітів потрібно для оптимального представлення рівнів пікселів. Тобто яка найменша кількість даних є достатньою для повного опису зображення без втрати інформації? З математичної погляду, відповіді такі питання дає наука теорія інформації. Основна послідовність цієї теорії у тому, що інформаційний потік можна моделювати як імовірнісного процесу, вимір якого добре узгоджується з нашою інтуїцією.

Відповідно до цього положення припускається, що стохастична величина E , що спостерігається з ймовірністю $P(E)$, несе

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

одиниць інформації. Якщо $P(E) = 1$ (подія завжди настає), то $I(E) = 0$ тут немає жодної інформації. Справді, у разі немає невизначеності у настанні цієї події, і тому немає необхідності повідомляти чи передавати інформацію у тому, що подія мала місце. При такому підході, що рідше настає подія, то вона «цінніше» і, отже, несе у собі велику інформацію. Маючи джерело випадкових подій з дискретної родини можливих результатів та відповідний набір ймовірностей цих результатів, середня інформація, що припадає на один вихід джерела, звана ентропією цього джерела, обчислюється по формулі

$$H = - \sum_{j=1}^J P(a_j) \log P(a_j).$$

Якщо інтерпретувати зображення як вибірку, що породжується деяким «напівтоновим джерелом», можна моделювати ймовірності символів цього джерела за допомогою гістограми спостерігається напівтонового зображення. Тоді будується число, яке називається оцінкою першого порядку \tilde{H} у випадку ентропії джерела:

$$\tilde{H} = - \sum_{j=1}^L p_r(r_k) \log p_r(r_k).$$

Цю оцінку можна обчислювати за допомогою наступної М-функції припущення, що кожен рівень сірого кольору кодується незалежно. У цьому випадку відомо, що ентропія задає нижню межу стиснення, яку можна досягти шляхом видалення кодової надмірності.

Варто звернути увагу на використання функції `find` з MATLAB для знаходження індексів ненульових елементів гістограми `h`. Оператор `find(x)` еквівалентний команді `find(x ~= 0)`. Функція `entropy` використовує `find` при побудові індексного вектора `i` гістограми `h`, який буде використовуватися надалі для видалення всіх нульових значень формули для обчислення ентропії в останньому рядку функції. Якщо цього не зробити, функція `log2` видасть для `h` значення NaN (результатом команди `0*-inf` є не число), коли ймовірність символу дорівнює нулю.

2.3 Коди Хаффмана

При кодуванні рівнів напівтонового зображення або виходу деякої операції напівтонового відображення (різниці пікселів, довжин серій тощо) коди Хаффмана призначають символам джерела найменшу кількість кодових символів (наприклад, бітів) за умови, що символи джерела (наприклад, пікселі) кодуються окремо.

Перший крок методу Хаффмана полягає у побудові серії редукованих джерел шляхом упорядкування ймовірностей символів даного джерела та об'єднання символів в один, котрий замінюватиме їх на наступному кроці. Цей процес проілюстровано на рис. 2.2 а) для розподілу рівнів з табл. 2.1. У двох лівих колонках вихідний набір символів проставлені за зниженням ймовірностей. При першій редукції два символи з мінімальними ймовірностями об'єднуються в один «складовий символ». Такий символ і його ймовірність поміщаються в список першого редукованого джерела, який також упорядковується від максимальної ймовірності до меншої [див. третю колонку Рис. 2.2, а)]. Цей процес повторюється до того часу, доки утворюється редукований джерело лише з двома символами.

Вихідне джерело		Редуковане джерело	
Символ	Ймовірність	1	2
a_2	0.5	0.5	0.5
a_4	0.1875	0.3125	0.5
a_1	0.1875	0.1875	
a_3	0.125		

а)

Вихідне джерело		Редуковане джерело				
Символ	Ймовірність	Код	1	2		
a_2	0.5	1	0.5	1	0.5	1
a_4	0.1875	00	0.3125	01	0.5	0
a_1	0.1875	011	0.1875	00		
a_3	0.125	010				

б)

Рис. 2.2. Метод кодування Хаффмана: а) Редукування джерела; б) Призначення кодів слів

Другий крок процедури Хаффмана полягає в кодуванні кожного редукованого джерела, починаючи з джерела з найменшою кількістю символів і рухаючись до джерела. Найменший двійковий код для джерела із двома символами складається, звичайно, із символів 0 та 1. З рис. 2.2, б) видно, що ці символи приписуються до двох символів джерела праворуч (порядок присвоєння не має значення - перестановка 0 і 1 дасть абсолютно той самий результат). Оскільки для отримання другого символу

редукованого джерела з ймовірністю 0.5 було проведено об'єднанням пари символів попереднього редукованого джерела (розташованого зліва від нього), кодовий символ 0 приписується кожному з об'єднаних символів, після чого коди цих символів доповнюються ліворуч символами 0 і 1 (довільно) для відзнаки їх одне від одного. Потім ця операція повторюється для редукованих джерел всіх рівнів до вихідного джерела. Остаточні коди для символів вихідного джерела наведено у третій колонці на рис. 2.2 б).

Код Хаффмана на рис. 2.2 б) (і з табл. 2.1) миттєво і однозначно декодується блоковим кодом. Код називається блоковим через те, що кожен символ зображується у фіксованій послідовності символів коду та миттєво декодується, так як кожне кодове слово в закодованій послідовності можна декодувати незалежно від наступних кодових символів. Тому, що у кожному коді Хаффмана кожне кодове слово перестав бути (початком) жодного іншого кодового слова. З цієї причини код Хаффмана є однозначно декодованим, Таким чином, будь-яку послідовність кодованих по Хаффман символів можна декодувати, аналізуючи її зліва направо. Зображення розміром 4x4, розглянуте в прикладі 2.1, кодується по Хаффману в порядку рядків зверху вниз і стовпчиками зліва направо за допомогою кодів з рис. 2.2 б), в результаті чого виходить послідовність з 29 кодових символів 10101011010110110000011110011. Оскільки ми використовували миттєво і однозначно декодується блоковий код, немає необхідності ставити розділові символи між закодованими пікселями. Аналіз цієї кодової послідовності зліва направо виявляє, що першим допустимим кодовим словом є 1, яке є кодом символу П2, що відповідає рівню сірого кольору 109. Наступне допустиме кодове слово - це 010, яке відповідає рівню 123. декодоване зображення, яке еквівалентне зображенню f з прикладу.

2.4 Висновки до розділу 2

В розділі проаналізовано основи стиснення зображень. Встановлено, що системи стиснення зображень включають два структурних блоки: кодер і декодер. Проаналізовано типову структуру та призначення блоків кодера та декодера.

Зображення подається на вхід кодера, який переводить вхідні дані на певний набір символів і використовує його для представлення вихідного зображення.

Щоб побачити або використовувати стиснуте (тобто закодоване) зображення, його необхідно подати на декодер, який побудує реконструйоване зображення. У загальному випадку, зображення може співпадати з вихідним зображенням або може від нього відрізнитись. У першому випадку система називається вільною від помилок або системою кодування або стиснення без втрати інформації. В іншому випадку в реконструйованому зображенні присутні певні спотворення та втрати і система називається кодуванням або стисненням із втратою інформації.

Кодер відповідає за скорочення всіх трьох типів надмірності (кодової, міжпіксельної та візуальної) вихідного зображення. На першому кроці кодування перетворювач трансформує вхідне зображення в деякий (невізуальний) формат, пристосований для зниження міжпіксельної надмірності. На другій стадії блок квантувач знижує точність перетворювача відповідно до раніше обумовленого критерію точності. На цій стадії відбувається скорочення візуальної надмірності, тобто деяке огрублення зображення, майже непомітне для ока. Ця процедура є незворотною, тому не виконується, якщо необхідно здійснити стиснення без втрати інформації. На третій та останній стадії процесу кодер символів будує оптимальний код (який

скорочує кодову надмірність) для виходу квантувача і перетворює вихідну послідовність відповідно до побудованого коду.

Декодер має лише дві компоненти: декодер символів та зворотний перетворювач. Виконуються ними дії, зворотні діям блоку кодера символів та блоку перетворювача.

РОЗДІЛ 3

НАУКОВО-ДОСЛІДНА ЧАСТИНА

3.1 Моделювання усунення надлишковостей

3.1.1. Міжпиксельна надлишковість

Розглянемо два зображення на рис. 3.1, а) та в), які мають практично однакові гістограми. На цих гістограмах позначають три визначальні моди, які означають присутність на зображеннях трьох домінуючих інтервалів сірих тонів. Оскільки тут рівні сірого кольору не є рівномірно розподіленими, можна використовувати коди змінної довжини. Внаслідок цього стає можливим скорочення кодової надмірності, яка буде присутня при кодуванні пікселів за звичайною схемою кодами фіксованої довжини (Додаток В).

Зауважимо, що ентропійні оцінки першого порядку обох зображень близькі один до одного (7.4253 і 7.3505 біт/піксел), і ці зображення однаково стискаються функцією `mat2huff` (з коефіцієнтами стиснення 1.0704 проти 1.0821). З цього порівняння видно, що коди змінної довжини не використовують при стиску очевидну структурну перевагу рис. 3.1, в), на якому сірники лежать рівним рядом. На цьому зображенні є явна міжпиксельна кореляція, проте ця кореляція присутня також і на рис. 3.1, а), тому що значення пікселів можна значною мірою передбачити за значеннями їхніх сусідів. Інформація, яку має кожен окремий піксел, досить мала. Більшість візуального вкладу індивідуальних пікселів є надмірною, і її можна відновити, знаючи внесок інших близьких пікселів. Ця залежність і є основою для міжпиксельної надмірності.

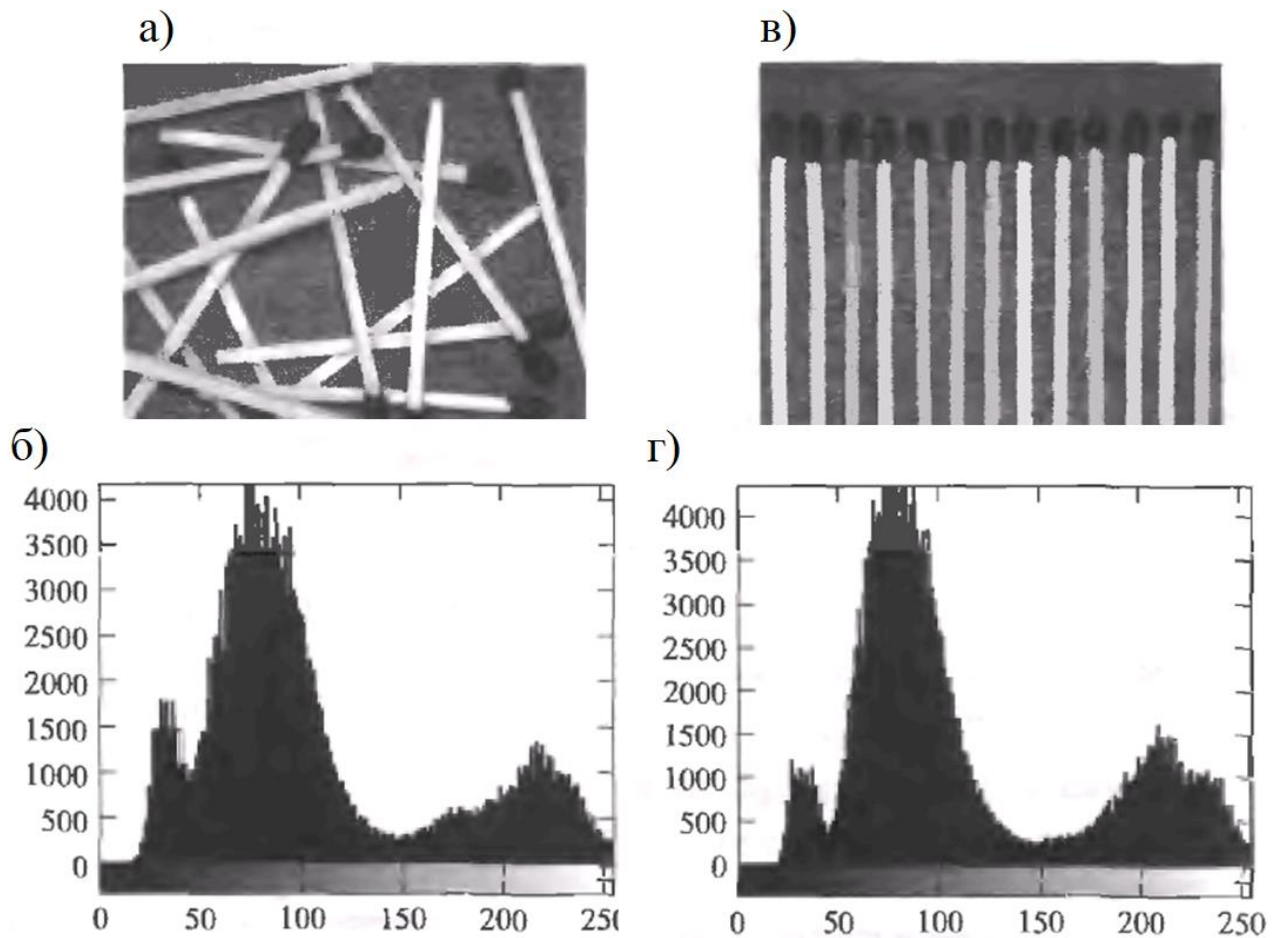


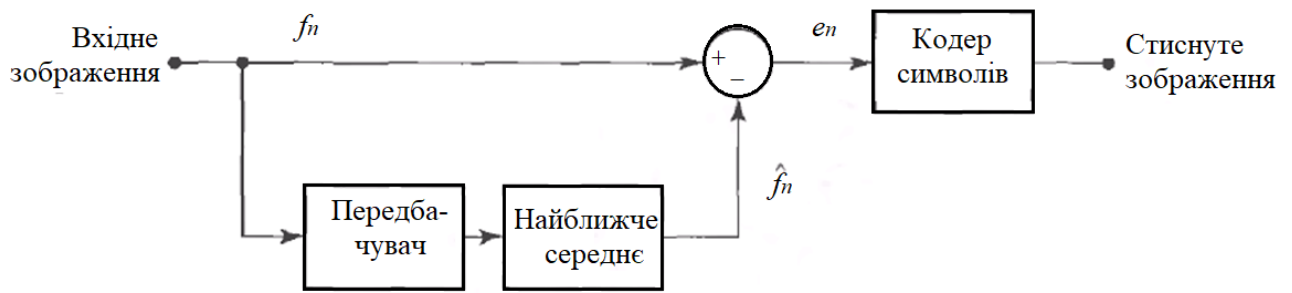
Рис. 3.1. Два зображення та їх напівтонові гістограми

Для того, щоб скоротити межпиксельную надмірність, двовимірні масиви, що використовуються для представлення зорових образів, необхідно перетворити на більш ефективний (але «невізуальний») формат. Так, для подання зображень можна використовувати різницю пікселів, що примикають один до одного. Ці перетворення (вони видаляють міжпиксельну надмірність) прийнято називати відображеннями. Відображення називаються оборотними, якщо елементи вихідного зображення можна однозначно відновити (реконструювати) за багатьма відображеними даними.

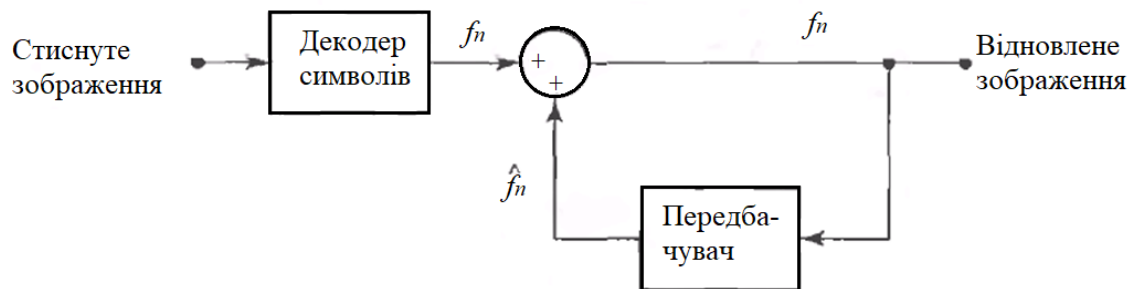
Проста процедура відображення показана на рис. 3.2. Цей підхід, званий кодуванням з передбаченням без втрат, видаляє межпиксельную надмірність

за допомогою віднімання та кодування лише нової (додаткової інформації) пікселів. Нова інформація пікселів отримується у вигляді різниці між реальним значенням пікселом та величиною його передбачення. Як видно, система включає до свого складу кодер і декодер, і кожен з них має той самий блок провісник. Коли черговий піксел вхідного зображення, що позначається f_n , поступає до кодера, передбачувач будує прогноз (оцінку значення цього пікселя), що базується на деякому наборі попередніх вхідних пікселів. Далі значення на виході передбачувача заокруглюється до цілого значення \hat{f}_n , яке використовується для знаходження різниці або помилки передбачення

$$e_n = f_n - \hat{f}_n.$$



а)



б)

Рис. 3.2. Спосіб кодування з передбаченням без втрати інформації:

а) кодер; б) декодер

Ця помилка кодується кодом змінної довжини (кодером символів), і цим генерується черговий елемент стисненого потоку даних. Декодер на рис. 3.2; б) відновлює значення e_n за прийнятим кодовим словом змінної довжини і здійснює зворотне перетворення

$$f_n = e_n + \hat{f}_n.$$

Для побудови передбачення \hat{f}_n можуть використовуватися різні локальні, глобальні чи адаптивні методи. Однак у більшості випадків прийнято обчислювати пророцтво у вигляді лінійної комбінації m попередніх пікселів:

$$\hat{f}_n = \text{round} \left[\sum_{i=1}^m \alpha_i f_{n-i} \right],$$

де m — це порядок лінійного передбачення, операція $\text{round} [\]$ означає заокруглення до цілого (на зразок функції round з MATLAB), а α_i при $i = 1, 2, \dots, m$ — коефіцієнти передбачення. Для одномірного кодування з передбаченням запишемо

$$\hat{f}(x, y) = \text{round} \left[\sum_{i=1}^m \alpha_i f(x, y - i) \right],$$

де кожна індексована змінна тепер виражена у вигляді функції просторових координат x і y . Зазначимо, що за такого підходу передбачають $\hat{f}(x, y)$ залежить від значень пікселів одного поточного рядка, що обробляється.

M-функції `mat21pc` та `Ipc2mat` реалізують процедуру кодування/декодування з передбаченням, описану вище (з відрахуванням кроків символного кодування/декодування). Функція кодування `mat21pc` використовує цикл `for` для побудови одночасного прогнозу всіх пікселів входу x . На кожному кроці ітерації масив x_s , який на початку дорівнює x , зсувається на одну позицію вправо (з нульовим заповненням зліва), множиться на відповідні коефіцієнти передбачення і додається до підсумовує масиву p . Оскільки зазвичай число коефіцієнтів лінійного прогнозу мало, процедура працює досить швидко. Зазначимо, що з виконанні наступної програми, якщо фільтр передбачення f не заданий на вході, використовується найпростіший фільтр, що з одного числа 1 (Додаток Г).

Функція декодування `Ipc2mat` робить дії, обернені перетворенням функції кодування `mat21pc`. Ця програма здійснює n ітерацій циклу `for`, де n позначає число стовпців закодованої вхідної матриці. На кожній ітерації обчислюється лише один стовпець виходу декодера, оскільки для цього потрібно знати всі попередні стовпці. Для прискорення обчислень циклу `for` матриця розширюється до максимально необхідного розміру перед початком циклу і заповнюється нулями. Зауважте також, що передбачення обчислюються тут у тому порядку, що у функції `Ipc2mat`. У цьому випадку вдається уникнути помилок заокруглення у обчисленнях з плаваючою точкою (Додаток Д).

3.1.2. Візуальна надмірність

В порівнянні із кодовою та міжпіксельною надлишковостями, візуальна надлишковість пов'язана зі справжньою візуальною інформацією, що піддається кількісному виміру. Її видалення бажано, оскільки ця інформація сама по собі не є істотною для звичайного візуального сприйняття. Оскільки зменшення візуальних надлишкових даних призводить до втрати частини кількісної інформації, то це називається квантуванням. Така термінологія добре узгоджується зі звичайним

змістом цього слова, що означає уявлення широкого набору величин за допомогою обмеженої множини допустимих значень. Ця процедура перестав бути оборотною (після її скоєння відбувається непоправна втрата частини інформації), тобто. квантування призводить до втрати частини даних, що стискаються.

3.2 Стандарт стиснення JPEG

При трансформаційному кодуванні використовується деяке лінійне перетворення, або дискретне косинусне перетворення, яке задається рівняннями

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \alpha_M(u) \alpha_N(v) \cos \left[\frac{(2x+1)u\pi}{2M} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right],$$

де

$$\alpha_M(u) = \begin{cases} \sqrt{\frac{1}{M}} & \text{при } u = 0, \\ \sqrt{\frac{2}{M}} & \text{при } u = 1, 2, \dots, M-1 \end{cases}$$

При цьому зображення $f(x, y)$ відображається в деяку кількість коефіцієнтів, які потім кодуються. Для переважної кількості природних зображень значна частина коефіцієнтів перетворення має мінімальну амплітуду, та його молено грубо квантувати (або навіть зовсім відкинути), вніши цим у зображення незначні спотворення, майже непомітні оку.

Одним із найвідоміших і широко застосовуваних універсальних стандартів компресії зображень є стандарт JPEG. У базовій системі кодування JPEG, заснованої на дискретному косинусному перетворенні і придатною для більшості додатків стиснення, вхідні та вихідні зображення обмежені 8-бітним форматом представлення

компонент яскравості і кольоровості, а довжина представлення коефіцієнтів DCT дорівнює 11 біт. Як видно на спрощеній блок-схемі рис. 3.3 а), саме стиснення здійснюється за чотири кроки: вилучення подзображень розмірами 8x8, обчислення DCT, квантування і присвоєння кодів змінної довжини, тобто кодування.

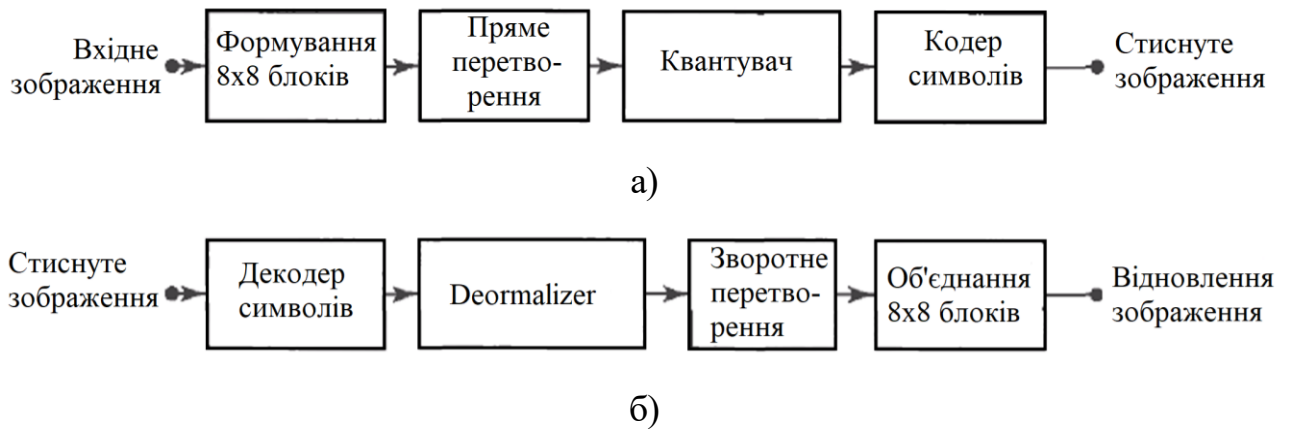


Рис. 3.3 Блок-схема JPEG: а) кодер; б) декодер

Перший крок процесу стиснення JPEG полягає в розділенні вхідного зображення на блоки пікселів, що не перетинаються, розмірами 8x8. Кожен блок 8x8 (зображення) піддається певній обробці. Всі його 64 піксели зрушуються відніманням числа 2^{m-1} , де 2^m - Це кількість рівнів яскравості зображення, після чого обчислюється дискретне косинусне перетворення блоку. Отримані коефіцієнти нормуються та квантуються за правилом

$$\hat{T}(u, v) = \text{round} \left[\frac{T(u, v)}{Z(u, v)} \right],$$

де $\hat{T}(u, v)$ при $u, v = 0, 1, \dots, 7$ позначають нормовані та квантовані коефіцієнти, $T(u, v)$ - Коефіцієнти DCT поточного блоку 8x8 зображення $f(x, y)$, а $Z(u, v)$ - нормуюча матриця перетворення, запропонована стандартом та наведена на рис. 3.4 а). Якщо міняти елементи матриці $Z(u, v)$ пропорційно деякому числу, це дозволяє

варіювати коефіцієнт досягається стиснення, що відбивається на якості реконструйованого зображення.

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

а)

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

б)

Рис. 3.4. а) Нормувальна матриця JPEG; б) Послідовність упорядкування коефіцієнтів зигзагу в JPEG

Після того, як коефіцієнти DCT всіх блоків будуть проквантовані, елементи двовимірних матриць $\hat{T}(u, v)$ переупорядковуються в зигзагоподібному порядку, показаному на рис. 3.4 б) послідовними числами 0,1,2,..., 63, утворюючи одномірний масив. Оскільки отримані одномірні вектори (квантованих коефіцієнтів) упорядковані за зростанням просторової частоти, то самі вони зменшуються в цьому напрямку, і кодер символів на рис. 3.3, а) оптимізований з урахуванням ймовірної появи довгих серій нулів, які зазвичай виникають після впорядкування зигзагом. Зокрема ненульові коефіцієнти AC [тобто всі $\hat{T}(u, v)$, крім випадку $u = v = 0$] кодуються з використанням кодів змінної довжини, які визначають величини цих коефіцієнтів та число наступних за ними нулів. Коефіцієнти DC [тобто $\hat{T}(0, 0)$]

кодуються різницевиими кодами стосовно коефіцієнтів DC попередніх під зображень. У стандарті є таблиці для кодів Хаффмана коефіцієнтів DC і AC, які використовуються при стисненні за замовчуванням, але користувач може побудувати власні таблиці, а також задати свою нормувальну матрицю. В останньому випадку всю цю інформацію доведеться додати до стислих даних у зарезервовану область параметрів.

Хоча повна реалізація стандарту JPEG є громіздкою, наступна M-функція моделює процес базового кодування (Додаток E).

Відповідно до блок-схеми на рис. 3.3. а), функція `im2jreg` обробляє окремі зображення або блоки 8×8 вхідного зображення x (попередньо блок за блоком, але не все зображення відразу). Для впорядкування обчислень використовуються дві функції обробки блоків `blkproc` та `im2col`. Функція `blkproc` має стандартний синтаксис

$$= \text{blkproc}(A, [M \ N], \text{FUN}, P1, P2, \dots).$$

Ця функція уніфікує весь процес побічної обробки зображення. Вона приймає вхідне зображення A разом із розмірами блоків $[M \ N]$, за якими потрібно обробляти зображення, а також функцію `FUN`, яка оброблятиме ці блоки, та деякий опціональний набір аргументів $P1, P2, \dots$ функції `FUN`. Потім функція `blkproc` розділяє на блоки $M \times N$ (використовуючи заповнення нулями там, де блок виходить за рамки зображення), викликає функцію `FUN` для кожного блоку з заданими параметрами $P1, P2, \dots, i$, нарешті, поміщає результат у вихідне зображення S_t .

Іншою функцією побічної обробки, яка використовується в процедурі `im2jreg`, є функція `im2col`. Якщо функція `blkproc` не підходить для реалізації специфічних побічних дій, функція `im2col` може використовуватися для переупорядкування входу так, щоб операції можна було зробити простіше та ефективніше (наприклад, використовуючи векторизацію). Виходом `im2col`

служить матриця, де кожен стовпець містить елементи одного блоку вхідного зображення. Її стандартизований формат має вигляд

$$B = \text{im2col}(A, [M N], 'distinct'),$$

де параметри A , B і $[M N]$ мають той сенс, що і в функції `blkproc`, а рядок `'distinct'` повідомляє `im2col` про те, що оброблювані блоки не перетинаються; інший рядок-параметр `'sliding'` сигналізує про те, що необхідно будувати один стовпець для кожного пікселя в A (як би блок ковзав по зображенню).

У процедурі `im2jreg` функція `blkproc` застосовується для полегшення обчислення DCT, нормування та квантування, а функція `im2col` спрощує впорядкування квантованих коефіцієнтів та відстеження серій нулів. На відміну від стандарту JPEG, функція `im2jreg` виявляє лише останню серію нулів у кожній упорядкованій послідовності коефіцієнтів квантованого блоку, замінюючи всю цю серію одним символом `eof` (закінчення блоку). Нарешті, відзначимо, що у системі MATLAB є ефективна утиліта, написана з урахуванням FFT, яка виконує DCT великих зображень (див. довідку функції `dct2`). Тим не менш, `im2jreg` використовує альтернативну матричну формулу

$$T = H F H^T,$$

де F - це блок 8×8 зображення $f(x, y)$, H - матриця перетворення DCT, яка будується командою `dctmtx`, а T - результат застосування DCT до F . Тут верхній індекс позначає операцію транспонування. За відсутності квантування зворотне перетворення DCT від T обчислюється за такою формулою

$$F = H^T T H.$$

Ця формула особливо ефективна при обробці малих квадратних зображень (подібно до DCT 8x8 в JPEG). Таким чином, команда

$$y = \text{blkproc}(x, [8 \ 8], 'P1*x*P2', h, h')$$

обчислює перетворення DCT зображення x блоків 8x8 з використанням матриці перетворення h і транспонованої матриці h' як параметрів $P1$ і $P2$, які є множниками у формулі матричного вироблення DCT $P1 * x * P2$, що грає роль функціонального параметра FUN.

Аналогічна поблочна обробка та перетворення на основі матричних добутків (див. рис. 3.3 б)) застосовуються в процесі декодування зображення, стисненого функцією `im2jreg`. Нижче наведена функція `jreg2im` виконує всю необхідну послідовність зворотних операцій (за винятком квантування). Вона використовує спільну функцію

$$A = \text{col2im}(B, [M \ N], [MM \ NM], 'distinct')$$

для відтворення двовимірного зображення зі стовпців матриці, де кожен стовпець по 64 елемента є блоком 8x8 реконструйованого зображення. Параметри A , B , $[M \ N]$ та `'distinct'` були визначені при описі функції `im2col`, а масив $[MM \ NM]$ означає розміри вихідного зображення A (Додаток Є).

На рис. 3.5, а) та б), наведено два кодовані та декодовані зображення JPEG, які є наближеннями одного і того ж вихідного чорно-білого зображення на рис. 3.5 а). Перший результат, що має коефіцієнт стиснення приблизно 18 до 1, отриманий прямим використанням стандартної матриці, що нормує, на рис. 3.4 а). Друге зображення, що має коефіцієнт стиснення 42 до 1, побудовано при множенні матриці, що нормує, на 4, що відповідає більш грубому квантуванню.

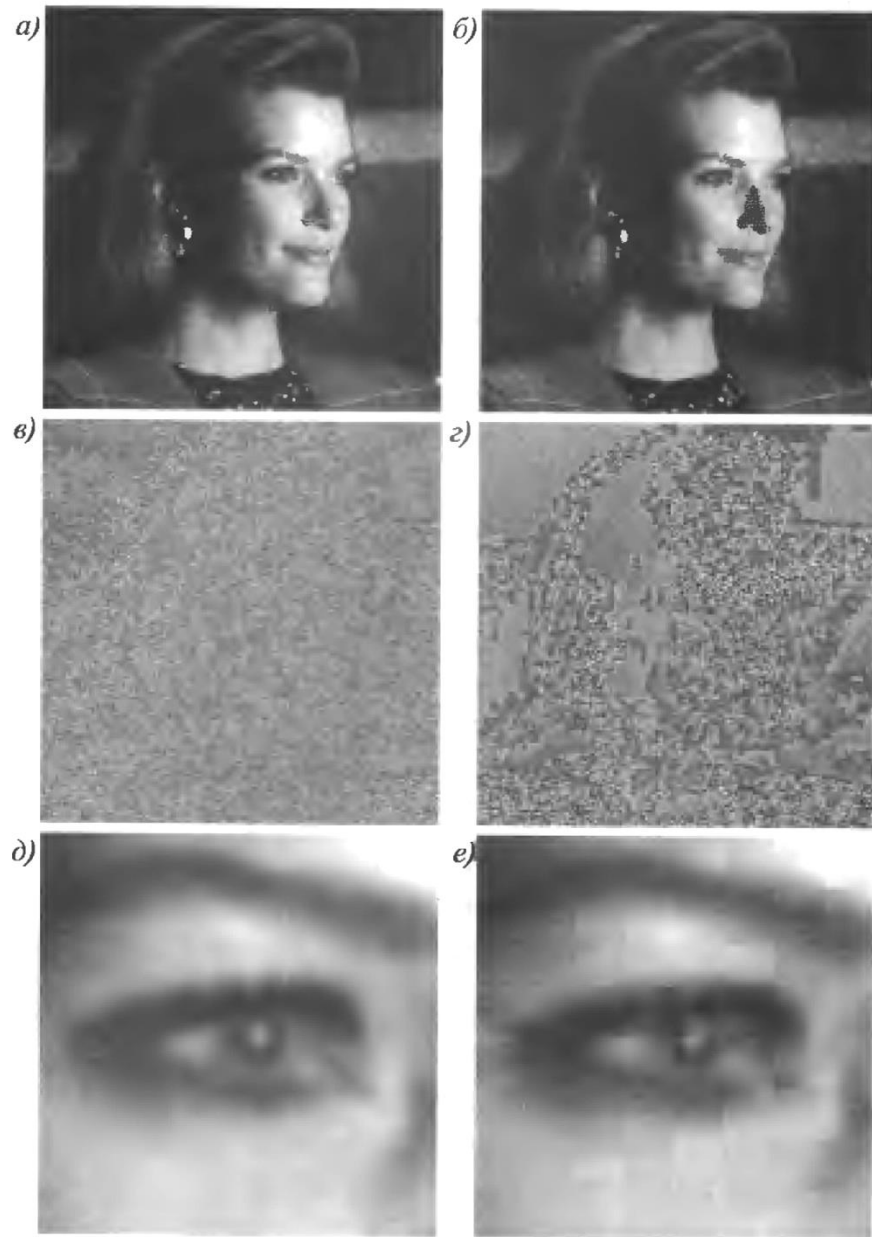


Рис. 3.5. Ліва колонка: наближення з використанням DCT та нормуючої матриці 3.4 а). Права колонка: аналогічні результати з нормуючою матрицею, збільшеною в 4 рази

Різниця між вихідним зображенням та відновленими зображеннями побудовані, відповідно, на рис. 3.5, в) та г). Тони обох зображень були посилені для кращої візуалізації відхилень. Відповідні середні помилки RMS дорівнюють 2.5 та 4.4 рівнів яскравості. Вплив цих помилок на якість

зображень видно чіткіше на збільшених ділянках зображень, наведених на рис. 3.5, д) та е). Ці зображення дозволяють краще оцінити тонкі відмінності між обома відновленими зображеннями

Зображення на рис. 3.5 отримані за допомогою наступної послідовності команд наведеної в додатку Ж.

Ці результати відрізняються від тих, які виходять при стисненні стандартом JPEG, оскільки розглянута програма `im2jpeg` лише моделює процес кодування Хаффмана, закладений в стандарт JPEG. Є дві важливі відмінності, що заслуговують на увагу:

(1) у стандарті всі серії нульових коефіцієнтів кодуються за Хаффманом, а `im2jpeg` кодується лише завершальна серія нулів кожного блоку;

(2) кодер і декодер стандарту використовують (за замовчуванням) запропонований код Хаффмана, тоді як `im2jpeg` отримує інформацію, необхідну реконструкції таблиці кодових слів із самого зображення.

Якщо використовувати повний стандарт JPEG, то коефіцієнт стиснення підвищиться приблизно вдвічі.

3.3 Модифікований метод JPEG

Як і початкова реалізація JPEG, алгоритм стиснення стандарту JPEG 2000 базується на тому, що коефіцієнти перетворення, яке декорелює пікселі зображення, можна кодувати ефективніше, ніж самі вихідні пікселі. Якщо базисні функції перетворення — вейвлети у разі JPEG 2000 — пакують найважливішу частину візуальної інформації відносно невелику кількість коефіцієнтів, то коефіцієнти, що залишилися, можна грубо проквантувати або зовсім обнулити, що призведе лише до малих спотворень стисненого зображення.

Рис. 3.9 показує спрощену систему кодування JPEG 2000 (у ній відсутні кілька опціональних операцій). Перший крок процесу кодування, як і у вихідному стандарті JPEG, полягає у зрушенні в нуль середнього рівня яскравості пікселів, яке досягається відніманням з усіх пікселів числа 2^{m-1} , де 2^m - загальна кількість можливих рівнів яскравості. Далше знаходиться одновимірне дискретне вейвлетне перетворення рядків та стовпців зображення. При стисканні без втрат використовується біортогональне вейвлетне перетворення 5-3². У додатках, що допускають втрату інформації, використовується вейвлетне перетворення 9-7. В обох випадках початковим результатом декомпозиції зображення є чотири піддіапазони - одне низькочастотне наближення і три приватні характеристики (деталі) по горизонталі, вертикалі та діагоналі.

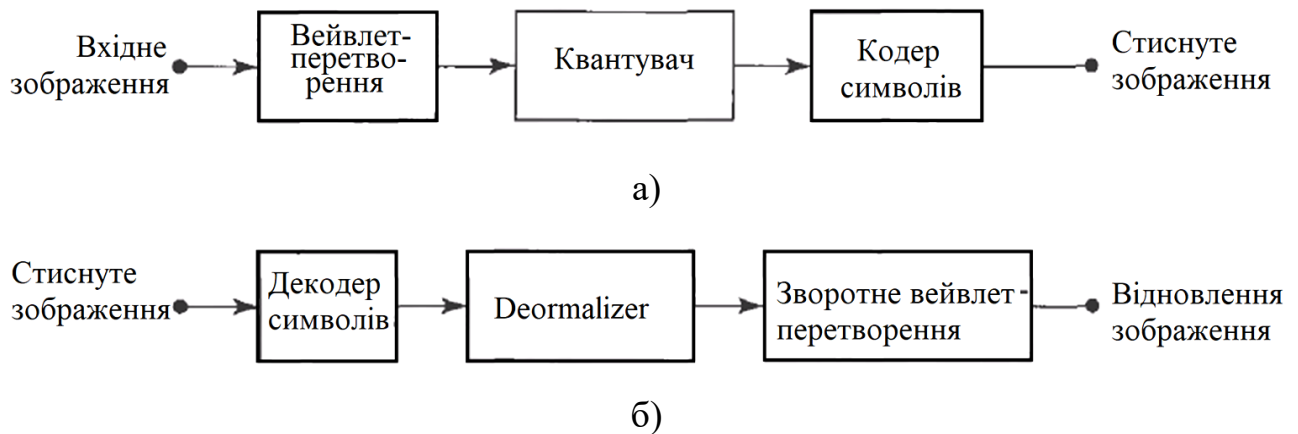


Рис. 3.6. Блок-схема JPEG 2000: а) кодер та б) декодер

Повторюючи процес декомпозиції NL разом із наступними ітераціями, що застосовуються до коефіцієнтів наближення попередньої декомпозиції, будується вейвлетне перетворення масштабу NL. Просторовий дозвіл сусідніх масштабів відрізняється вдвічі. Перетворення масштабу NL загального виду складається з $3N_L + 1$ піддіапазону, коефіцієнти яких позначаються індексованою літерою a_b , де $b = N_L LL, N_L HL,$

$N_{L}LH, N_{L}LL, \dots, 1HL, 1LH, 1HH$. Стандарт не обмежує кількість обчислюваних масштабів. Після того, як вейвлетне перетворення масштабу N_L обчислено, загальна кількість коефіцієнтів рівна числу відліків вихідного зображення, але головна візуальна інформація сконцентрована лише у невеликому числі коефіцієнтів. Для зменшення кількості бітів, потрібних для їх подання, коефіцієнт $a_b(u, v)$ піддіапазону b квантується на величину $q_b(u, v)$ по формулі

$$q_b(u, v) = \text{sign}(a_b(u, v)) \cdot \text{floor} \left[\frac{|a_b(u, v)|}{\Delta_b} \right],$$

де $\text{sign}(\cdot)$ — це знак числа, $\text{floor} [\cdot]$ — ціла частина числа (подібно до стандартних функцій sign3 і floor в MATLAB). Крок квантування Δ_b задається виразом

$$\Delta_b = 2^{R_b - \epsilon_b} \left(1 + \frac{\mu_b}{2^{11}} \right).$$

Тут R_b — номінальний динамічний діапазон піддіапазону b . Динамічний діапазон b дорівнює сумі числа бітів, що використовуються для представлення вихідного зображення, та кількості додаткових бітів аналізу піддіапазону b .

Для стиснення без втрат використовуються такі параметри: $\mu_b = 0$, $R_b = \epsilon_b$, $\Delta_b = 1$. Для необоротного стиску в стандарті не передбачено жодного конкретного кроку квантування. Натомість декодер повинен знати кількість бітів порядку та мантиси, яке або передається разом з кожним піддіапазоном, або тільки з піддіапазоном $N_{L}LL$ (Неявне квантування). Вважаючи, що $\epsilon_0 = \mu_0$ це відоме число бітів піддіапазону $N_{L}LL$, параметри для піддіапазону b обчислюються за формулами

$$\mu_b = \mu_0, \quad \varepsilon_b = \varepsilon_0 + nsd_b - nsd_0,$$

де nsd_b позначає число рівнів декомпозиції від вихідного зображення до піддіапазону d . Фінальним кроком процесу стиснення є кодування квантованих коефіцієнтів за допомогою арифметичного методу кодування на основі бітових площин.

Функція `im2_jpeg2k`, що наводиться далі, моделює процес кодування JPEG 2000 на рис. 3.9, а) за винятком етапу арифметичного символного кодування. Для простоти використовується кодування Хаффмана, розширене кодуванням довжин нульових серій (Додаток 3).

Декодер JPEG 2000 просто звертає описані вище операції. Після декодування арифметичних кодів коефіцієнтів відбувається відновлення вибраного користувачем числа піддіапазонів зображення. Хоча кодер може мати арифметично закодованих бітових площин для конкретного піддіапазону, користувач може вибрати для декодування тільки бітових площин. Це дорівнює квантуванню коефіцієнтів з використанням розміру кроку $2^{M_b - N_b}$ ■
At,. Всі біти, що не декодуються, прирівнюються нулю, і отримані коефіцієнти, що позначаються $\bar{q}_b(u, v)$, відновлюються за формулами

$$R_{q_b}(u, v) = \begin{cases} (\bar{q}_b(u, v) + 2^{M_b - N_b(u, v)}) \cdot \Delta_b & \text{при } \bar{q}_b(u, v) > 0, \\ (\bar{q}_b(u, v) - 2^{M_b - N_b(u, v)}) \cdot \Delta_b & \text{при } \bar{q}_b(u, v) < 0, \\ 0 & \text{при } \bar{q}_b(u, v) = 0, \end{cases}$$

де $R_{q_b}(u, v)$ позначає відновлені значення коефіцієнтів, $N_b(u, v)$ кількість декодованих бітових площин для $\bar{q}_b(u, v)$. Потім отримані коефіцієнти піддаються зворотному перетворенню, а результату додається величина 2^{n-1} ,

де p – число бітів яскравості пікселів. Функція `jpeg2k2im` моделює весь процес, звертаючи результат стиснення під дією функції наведеної в додатку И.

Принципова відмінність системи JPEG 2000 на основі вейвлетів від системи JPEG на базі перетворення DCT полягає у відсутності етапу поділу зображення на маленькі зображення. Оскільки вейвлетні перетворення одночасно є обчислювально ефективними та локальними за своєю природою (тобто їх базисні функції мають кінцеву тривалість), немає необхідності ділити стиснене зображення на підблоки. Відсутність стадії розподілу усуває артефакти блочності, які є невід'ємними супутниками наближення зображень на основі перетворення DCT при високому ступені стиснення.

3.4 Метод стиснення зображень

Пропонований метод ґрунтується на методі, який лежить в основі методу JPEG 2000 з наступною різницею. Оскільки JPEG 2000 працює на основі вейвлетів і в структурі відновлених зображень відсутній ефект блочності то пропонується перед вейвлет перетворенням кодування контрастності, що дасть можливість пропорційно до кількості рівнів кодування зменшити розмір зображення, вихідного до використання методу JPEG 2000. А власне ефект від застосування зворотного вейвлет перетворення дасть можливість відновити втрачені на першому етапі кодування рівні контрастності.

На рис. 3.7 наведено вигляд зображення до кодування.



Рис. 3.7. Тестове зображення

На рис. 3.8 наведено результат контрастного кодування зображення



Рис. 3.8. Результат контрастного кодування

На рис. 3.9 наведено вигляд кодованого зображення



Рис. 3.9. Результат кодування зображення

На рис. 3.10 наведено вигляд декодованого зображення.

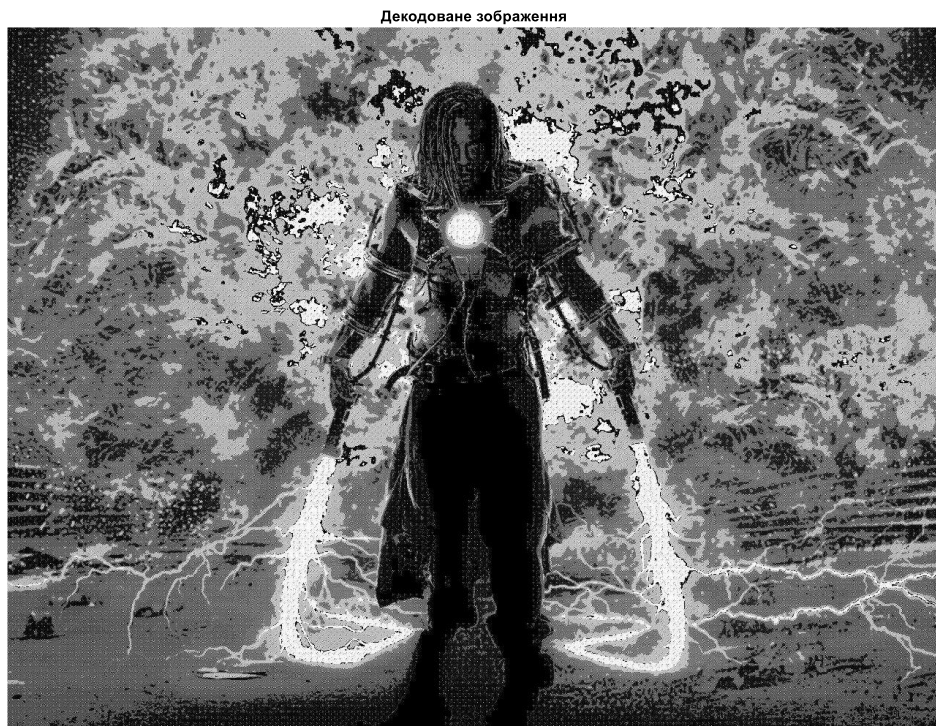


Рис. 3.10. Вигляд декодованого зображення

З рис. 3.10 видно, що результат декодування частково співпадає з вихідним за візуальною якістю при кращому коефіцієнті стиснення.

3.5 Висновки до розділу 3

В розділі проведено моделювання усунення надлишковостей, зокрема піжпиксельної та візуальної. Описано спосіб кодування з передбаченням без втрати інформації

Як основу проаналізовано та змодельовано в середовищі Matlab процес стиснення зображень, який використовується в стандарті JPEG, та ґрунтується на використанні кодування Хаффмана.

Змодельовано метод стиснення за стандартом JPEG, який ґрунтується на використанні вейвлет перетворення замість перетворення Хаффмана.

Принципова відмінність описаного методу стиснення на основі вейвлетів від системи JPEG полягає у відсутності етапу поділу зображення на маленькі зображення. Оскільки вейвлетні перетворення одночасно є обчислювально ефективними та локальними за своєю природою (тобто їх базисні функції мають кінцеву тривалість), немає необхідності ділити стиснене зображення на підблоки. Відсутність стадії розподілу усуває артефакти блочності, які є присутні у інших стандартах.

Запропоновано метод кодування зображень та перевірено якість його роботи.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Основні вимоги до побудови і функціонування системи управління охороною праці (СУОП). Забезпечення функціонування та побудова СУОП в організації. Положення про СУОП, структура та зміст його розділів.

В Україні функціонує багаторівнева СУОП, функціональними ланками якої є відповідні структури державної законодавчої і виконавчої влади різних рівнів, управлінські структури підприємств і організацій, трудових колективів.

Залежно від спрямування вирішуваних завдань всі ланки СУОП можна розділити на дві групи:

- ланки, що забезпечують вирішення законодавчо-нормативних, науково-технічних, соціально-економічних та інших загальних питань охорони праці;
- ланки, до функціональних обов'язків яких входить забезпечення безпеки праці в умовах конкретних організацій, підприємств.

До першої групи належать органи державної законодавчої ініціативи та органи державного управління охороною праці:

- Верховна Рада України;
- Кабінет Міністрів України;
- Державна служба гірничого нагляду та промислової безпеки України (Держгірпромнагляд України);
- міністерства та інші центральні органи державної виконавчої влади;
- Фонд соціального страхування від нещасних випадків і профзахворювань;
- місцева державна адміністрація, органи місцевого самоврядування.

Верховна Рада України зі своєї ініціативи у взаємодії з відповідними структурами державної виконавчої влади визначає державну політику в сфері охорони праці, вирішує питання щодо удосконалення і розвитку законодавчої бази охорони праці, соціальні питання, пов'язані зі станом умов і охорони праці.

Кабінет Міністрів України забезпечує реалізацію державної політики в сфері охорони праці, виходячи із стану охорони праці в державі, організує розробку загальнодержавних програм відповідно до поліпшення цього стану, затверджує ці програми і контролює їх виконання, визначає функції органів виконавчої влади щодо вирішення питань охорони праці і нагляду за охороною праці.

Для вирішення цих питань при Кабінеті Міністрів України функціонує Національна рада з питань безпечної життєдіяльності населення, яку очолює віце-прем'єр-міністр України.

Держгірпромнагляд України здійснює комплексне управління охороною праці на державному рівні, реалізує державну політику в цій сфері, розробляє за участі відповідних органів державної програми в сфері охорони праці, координує роботу державних органів і об'єднань підприємств із питань безпеки праці, розробляє і переглядає разом з компетентними органами систему показників і обліку умов і безпеки праці, здійснює міжнародне співробітництво з питань охорони праці і нагляд за охороною праці в державі тощо.

Рішення Держгірпромнагляду України, що відносяться до її компетенції, обов'язкові для виконання всіма міністерствами, іншими центральними органами державної виконавчої влади, місцевими державними адміністраціями, місцевими радами народних депутатів і підприємствами.

Фонд соціального страхування від нещасних випадків здійснює профілактику нещасних випадків і профзахворювань, а також координацію всієї страхової діяльності, пов'язаної з охороною праці.

Міністерство праці і соціальної політики України здійснює також державну експертизу умов праці, визначає порядок і здійснює контроль за якістю проведення атестації робочих місць згідно з їх відповідністю нормативним актам про охорону праці, бере участь у розробці нормативних документів про охорону праці.

Інші міністерства і центральні органи державної виконавчої влади як ланки системи управління охороною праці визначають науково-технічну політику галузі з питань охорони праці, розробляють і реалізують комплексні заходи щодо поліпшення безпеки праці, здійснюють методичне керівництво діяльністю підприємств галузі з охорони праці, співробітничать з галузевими профспілками щодо вирішення питань безпеки праці, організують у встановленому порядку навчання і перевірку знань правил і норм охорони праці керівниками і фахівцями галузі, створюють, у разі необхідності, професійні воєнізовані аварійно-рятувальні формування, здійснюють внутрішній контроль за станом охорони праці.

Для забезпечення виконання перелічених функцій в апаратах міністерств і інших центральних органів державної виконавчої влади створюються служби охорони праці.

Місцеві державні адміністрації й органи місцевого самоврядування в межах підвідомчої їм території забезпечують реалізацію державної політики в сфері охорони праці, формують за участі профспілок місцеві програми заходів щодо поліпшення безпеки, гігієни праці і виробничого середовища, здійснюють контроль за дотриманням нормативних актів про охорону праці. Для забезпечення виконання названих функцій при місцевих органах державної виконавчої влади створюються відповідні структурні підрозділи.

Управлінські структури підприємств забезпечують в умовах конкретних виробництв реалізацію вимог законодавчих і нормативних актів про охорону праці з метою створення безпечних і нешкідливих умов праці, попередження

виробничого травматизму і професійних захворювань, вирішують весь комплекс питань з охорони праці, пов'язаних з даним виробництвом. У своїй діяльності стосовно охорони праці управлінські структури підприємств взаємодіють з комісією з питань охорони праці підприємства (за наявності такої), з профспілками підприємства та уповноваженими трудових колективів.

СУОП в умовах конкретної організації, на конкретному об'єкті завжди є багаторівневою системою управління, у якій верхнім рівнем є державне управління, а нижнім - управління охороною праці на конкретному об'єкті. Як проміжні рівні управління можуть виступати відомче, регіональне управління, а також управління в об'єднанні, тресті тощо.

Слід зазначити, що вихідні параметри СУОП визначаються, виходячи з вимог норм, правил, проектної документації, аналізу фактичного стану виробничої ситуації і ряду факторів виробничого середовища, тому СУОП варто віднести до категорії звичайних, багатоконтурних систем, які піддаються програмуванню. Багатоконтурність систем управління в даному випадку пояснюється складністю об'єкта управління, його великою інерційністю, складністю і інерційністю реалізації управлінських впливів.

Правовою основою СУОП є: Конституція України, Кодекс законів про працю України, Закони України «Про охорону праці» і «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві і професійного захворювання, які спричинили втрату працездатності», накази і розпорядження Президента України, розпорядження і постанови Кабінету Міністрів, Держгірпромнагляд, Міністерства охорони здоров'я, Міністерства праці і соціальної політики, а також інших директивних органів України з питань охорони праці (органи Державного управління охороною праці).

Позитивна дія впровадження систем управління охороною праці (СУОП) на рівні організації як на зниження небезпек і ризиків, так і на продуктивність, нині визнана урядами, роботодавцями і працівниками.

4.2 Безпека в надзвичайних ситуаціях

Надзвичайні екологічні ситуації та екологічний ризик

Особливу роль у житті людини відіграють надзвичайні ситуації, що виникають під час стихійних лих або техногенних катастроф. Разом із соціальними та економічними збитками надзвичайні ситуації завдають також екологічної шкоди, що відображається в руйнуванні й деградації природних систем, забрудненні повітря, водойм і ґрунтів. У результаті виникають надзвичайні екологічні ситуації. Надзвичайні екологічні ситуації — ті ситуації, що виникають унаслідок раптових природних лих або техногенних аварій і супроводжуються великими збитками. Характерними особливостями цих ситуацій є велика гострота прояву, значні відхилення показників навколишнього середовища від норми (перевищення граничнодопустимих концентрацій (ГДК) забруднювальних речовин у сотні, тисячі й навіть десятки тисяч разів); ураганні швидкості вітру; затоплення селітебних територій (населених пунктів); виникнення катастрофічних селевих потоків та ін.

Звичайно, такі відхилення тривають недовго — години, дні, десятки днів, іноді більше. Потім ступінь гостроти екологічного стану зменшується, хоча може залишатися досить високим. Отже, поняття надзвичайна екологічна ситуація та катастрофічна екологічна ситуація розрізняються тим, що перша триває порівняно недовго, але настає раптово та характеризується виключно високими відхиленнями стану навколишнього середовища від норми, а друга — досить тривала (як правило, роки), але має меншу гостроту прояву.

Надзвичайна ситуація за певних обставин може перетворитися на катастрофічну. Наприклад, ситуація у Чорнобильській зоні. Протягом майже місяця радіаційна обстановка в Чорнобилі була надзвичайною. Після спорудження саркофага викиди радіоактивних елементів різко зменшилися, але забруднення до того часу охопило великі території. Таке високе радіаційне забруднення продовжується вже понад два десятиріччя. За оцінкою спеціалістів, екологічна ситуація в Чорнобильській зоні є катастрофічною.

Таким чином, надзвичайні екологічні ситуації відображаються у порушенні нормального функціонування природних і природно-антропогенних систем, пов'язаних із раптовими природними або техногенними впливами (стихійні лиха, катастрофи, аварії), що супроводжуються соціальними, економічними та екологічними збитками і потребують для ліквідації особливих управлінських рішень (рис. 1.2). Збитки виявляються у загибелі та пораненні людей, погіршенні їх здоров'я, руйнуванні матеріальних об'єктів, структури природних і природно-антропогенних систем, втраті їх природно-ресурсного і екологічного потенціалу. Довготривала надзвичайна ситуація зумовлює формування зони екологічної катастрофи або екологічного лиха.

Надзвичайні екологічні ситуації виникають унаслідок дії трьох основних груп факторів:

- свідомого руйнування природного середовища, походження техніки, погіршення становища економічних об'єктів під час війн і диверсійних актів;
- руйнівних катастроф, які виникають у зв'язку з некомпетентними та помилковими технічними рішеннями (наприклад, Чорнобильська аварія);
- природних стихійних явищ. Той факт, що різко збільшилися їх частота та інтенсивність в останні десятиріччя, спеціалісти пов'язують з антропогенною стимуляцією, що спричинює посилення відхилень природних процесів від нормального рівня коливань.

Економічні збитки, завдані у зв'язку з несприятливими і небезпечними природними процесами та явищами, значно збільшилися. За деякими оцінками, вони зростають швидше, ніж показники світового валового продукту, тобто може бути досягнута межа просторового і технологічного розвитку виробництва за його здатністю компенсувати збитки, які збільшуються, від несприятливих і небезпечних явищ. Первинні процеси, що виникають у природному середовищі внаслідок цих факторів, посилюватимуться або послаблюватимуться залежно від природної обстановки (стійкість ландшафтів, погодні умови, фаза коливань екосистеми тощо) і соціально-економічних умов (психологічна готовність і неготовність населення до ліквідації наслідків надзвичайної ситуації, технічна оснащеність спеціальних служб, економічні можливості та ін.). Таким чином, надзвичайні екологічні ситуації в більшості випадків мають комплексну природу.

Заходи щодо запобігання надзвичайним екологічним ситуаціям або подолання їх наслідків можна згрупувати у три класи:

- організаційні, серед яких розрізняють планувальні та оперативні;
- інженерно-технічні;
- технологічні.

Отже, заходи, спрямовані на запобігання надзвичайним екологічним ситуаціям та подолання, їх можна поділити на два типи: заходи, спрямовані на зниження піддатливості об'єктів небезпечним впливам, і заходи, спрямовані на зниження чутливості об'єктів до небезпечних впливів. У першому випадку здійснюють заходи з метою зовнішнього захисту об'єктів, виключення тих чи інших територій з використання у виробничих цілях тощо. Зниження чутливості об'єктів до небезпечних впливів досягається, насамперед, за рахунок досконаліших технологій, шляхом регулювання технологічних режимів у зв'язку з природними циклами, створення системи дублювання об'єктів, інформаційних систем і систем швидкого реагування.

Основні функції щодо запобігання надзвичайним екологічним ситуаціям та подолання їх на державному рівні виконують міністерства з надзвичайних ситуацій.

Ризик — це об'єктивне поняття, він пов'язаний практично з будь-якою діяльністю людини. Уміння усвідомлювати ступінь ризику дає змогу людині оцінити власні можливості й вибрати напрями поведінки при цьому. Під сутністю терміна ризик розуміють імовірність, по-перше, будь-якої небезпечної події; по-друге, негативних наслідків від неї та обсягу очікуваних збитків. Одні ризики конкретні, інші — не мають такого визначення. Існують професійні ризики (наприклад, небезпека професійних захворювань) і такі, яких зазнає все населення (екологічний, економічний, геологічний, політичний ризики).

Предметом нашого дослідження є екологічний ризик, чіткого визначення якого досі немає. М.Ф. Реймерс вважає, що це ймовірність наслідків будь-яких (специфічних або випадкових, поступових або катастрофічних) антропогенних змін природних об'єктів і факторів^{*22}. З екологічним ризиком пов'язані поняття екологічної безпеки і небезпеки. Ці альтернативні категорії стосуються населення як реципієнта дії навколишнього середовища за його відповідно несприятливого чи сприятливого статусу.

Екологічний ризик пов'язаний із такими групами факторів: 1) техногенними; 2) природними; 3) військовими; 4) соціально-економічними; 5) політичними; 6) тероризмом.

Техногенний екологічний ризик виникає у зв'язку з аваріями на ЛЕС, аваріями танкерів, на небезпечних хімічних виробництвах, під час руйнування гребель водосховищ тощо. Причинами аварій є інтенсивність технологічних процесів та зв'язків, висока концентрація виробництва, ресурсомісткість і багатовідходність технологій, погана оснащеність очисними й утилізаційними пристроями.

Природний екологічний ризик пов'язаний із ймовірністю вияву багатьох несприятливих природних явищ, таких як землетруси, вулканізм, селі, повені, цунамі та ін. Потрібно враховувати особливості геологічної будови (властивості гірських порід, наявність або відсутність розламів тощо), рельєфу (наприклад, посилення ризику забруднення в улоговинах), ландшафтів (ступінь їх стійкості до техногенних навантажень). Варто також зважати на сусідство цінних та унікальних природних об'єктів, територій особливого режиму охорони. Екологічний ризик збільшується за високої густоти населення, а також залежить від характеру сприйняття населенням подій, що відбуваються. Відомо, що катастрофічні наслідки аварій і стихійних природних явищ різко зростають у результаті психологічної неготовності населення до таких подій.

Особливу групу факторів виникнення екологічного ризику становлять воєнні дії, які зумовлюють різноманітні зміни навколишнього середовища та безпосередньо впливають на людину й інші суб'єкти. Екологічний ризик пов'язаний також із соціально-економічними факторами. Йдеться про ймовірність виникнення несприятливих екологічних ситуацій у разі прийняття рішень про будівництво тих чи інших небезпечних об'єктів у зв'язку з соціальною й економічною потребами такого будівництва. До цієї категорії належить будівництво багатьох АЕС, створення небезпечних хімічних виробництв, транспортних систем. У деяких випадках аналогічні рішення пов'язані з політичними факторами.

Нині є та розробляється велика кількість науково обґрунтованих постанов, нормативів, правил, державних стандартів, за якими регламентується господарська діяльність, встановлюються граничнодопустимі концентрації шкідливих і токсичних компонентів у ґрунтах, підземних і поверхневих водах тощо. На основі цих документів та екологічного законодавства в Україні розроблено систему заходів на державному, відомчих та об'єктних рівнях, що регламентують ведення екологічно безпечної господарської діяльності,

будівництво різних споруд, межі забруднення природного середовища в рамках не лише окремих локальних систем, а й великих регіонів, держави в цілому. Такі заходи можна об'єднати у три основні групи — соціально-організаційні, оцінювально-прогнозні та технічні. Усі види заходів взаємопов'язані і є основою для організації безпечної життєдіяльності. Якщо їх правильно дотримуватися, можна не тільки зберегти стан навколишнього середовища, а й поліпшити його, уникнути екологічно небезпечних явищ і катастроф, зумовлених антропогенно-техногенною діяльністю.

4.3 Висновки до розділу

В розділі «Охорона праці та безпека в надзвичайних ситуаціях» описано основні вимоги до побудови і функціонування системи управління охороною праці, а також надзвичайні екологічні ситуації та екологічний ризик.

ВИСНОВКИ

Проаналізовано задачу стиснення зображень, що викликана значною надлишковістю зображень та необхідністю у значних обсягах баз даних для їхнього зберігання чи значних швидкостей передачі даних. Проаналізовано такі типи надлишковості зображень в телекомунікаційних системах, як структурну, статичну та візуальну (психофізичну).

Проаналізовано статичний метод кодування, застосовний до зображень, які зберігатимуться в базах даних. До статичного методу кодування відноситься застосування кодів з довжиною кодового слова. Найвідоміший із таких кодів — код Хаффмана. Встановлено, що системи стиснення зображень включають два структурних блоки: кодер і декодер. Проаналізовано типову структуру та призначення блоків кодера та декодера. Типовий кодер відповідає за скорочення всіх трьох типів надмірності (кодової, міжпиксельної та візуальної) вихідного зображення.

Проведено моделювання усунення надлишковостей, зокрема піжпиксельної та візуальної. Описано спосіб кодування з передбаченням без втрати інформації. Як основу проаналізовано та змодельовано в середовищі Matlab процес стиснення зображень, який використовується в стандарті JPEG, та ґрунтується на використанні кодування Хаффмана.

Змодельовано метод стиснення за стандартом JPEG, який ґрунтується на використанні вейвлет перетворення замість перетворення Хаффмана. Встановлено, що перевагою описаного методу стиснення на основі вейвлетів від системи JPEG є відсутність етапу поділу зображення на маленькі зображення. Оскільки вейвлетні перетворення одночасно є обчислювально ефективними та локальними за своєю природою, немає необхідності ділити стиснене зображення на підблоки. Відсутність стадії розподілу усуває

артефакти блочності, які є присутні у інших стандартах. Запропоновано метод кодування зображень та перевірено якість його роботи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. У.Прэтт. Цифровая обработка изображений. Т.1, Т.2 - М.,“Мир”,1982.
2. В.Г.Потемкин. Система инженерных и научных расчетов MATLAB 5.x., т.т.1,2 – М.,Диалог-МИФИ, 1999.
3. Гонсалес Р. Вудс Р. Эддинс С. Цифровая обработка изображений в среде MATLAB. Москва: Техносфера, 2006. – 616 с. ISBN 5-94836-092-X.
4. Хусаинов Н.Ш. Руководство к набору лабораторных работ «Кодирование изображений с использованием пакета MATLAB» по дисциплине «Теория кодирования информации». – Таганрог: ТТИ ЮФУ, 2008, 60 с.
5. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. - М.: Мир, 1988.
6. Ярославский Л.П. Введение в цифровую обработку изображений. - М.: Сов. радио, 1979.
7. Методы передачи изображений. Сокращение избыточности / У.К.Прэтт,
8. Д.Д. Сакрисон, Х.Г.Д. Мусманн и др. Под ред. У.К.Прэтта. -М.: Радио и связь, 1983.
9. Джайн А.К. Сжатие видеоинформации // ТИИЭР.- Т.69. - № 3.

ДОДАТКИ

```

function cr = imration(f1, f2)
%IMRATIO Computes the ratio of the bytes in two images/variables.
%CR = IMRATIO(F1, F2) returns the ratio of the number of bytes in
%variables/files F1 and F2. If F1 and F2 are an original and
% compressed image, respectively, CR is the compression ratio.
error(nargchk(2, 2, nargin)); % Check input arguments
cr = bytes(f1) / bytes(f2); % Compute the ratio
function b = bytes(f)
% Return the number of bytes in input f. If f is a string, assume
% that it is an image filename; if not, it is an image variable.
if ischar(f)
info = dir(f); b = info.bytes;
elseif isstruct(f)
% MATLAB's whos function reports an extra 124 bytes of memory
% per structure field because of the way MATLAB stores
% structures in memory. Don't count this extra memory; instead,
% add up the memory associated with each field.
b = 0;
fields = fieldnames(f);
for k = 1:length(fields)
b = b + bytes(f.(fields{k}));
end
info = whos('f'); b = info.bytes;
end

```

```

function rmse = compare(f1, f2, scale)
%COMPARE Computes and displays the error between two matrices.
% RMSE = COMPARE(F1, F2, SCALE) returns the root-mean-square error
% between inputs F1 and F2, displays a histogram of the difference,
% and displays a scaled difference image. When SCALE is omitted, a
% scale factor of 1 is used.
% Check input arguments and set defaults.
error(nargchk(2, 3, nargin));
if nargin < 3
scale = 1;
end
% Compute the root-mean-square error.
e = double(f1) - double(f2);
[m, n] = size(e);
rmse = sqrt(sum(e(:).^2) / (m * n));
% Output error image & histogram if an error (i.e., rmse ~= 0).
if rmse
% Form error histogram,
emax = max(abs(e(:)));
[h, x] = hist(e(:), emax);
if length(h) >= 1
figure; bar(x, h, 'k');
% Scale the error image symmetrically and display
emax = emax / scale;
e = mat2gray(e, [-emax, emax]);
figure; imshow(e);
end
end
end

```

```
function h = entropy(x, n)
%ENTROPY Computes a first-order estimate of the entropy of a matrix.
%H = ENTROPY (X, N) returns the first-order estimate of matrix X
% with N symbols (N = 256 if omitted) in bits/symbol. The estimate
% assumes a statistically independent source characterized by the
% relative frequency of occurrence of the elements in X.
error(nargchk(1, 2, nargin)); % Check input arguments
if nargin < 2
    n = 256; % Default for n.
end
x = double(x); % Make input double
xh = hist(x(:), n); % Compute N-bin histogram
xh = xh / sum(xh(:)); % Compute probabilities
% Make mask to eliminate 0's since log2(0) = -inf.
i = find(xh);
h = -sum(xh(i).* log2(xh(i))); % Compute entropy
```

```
>> f1 = imread('Random Matches.tif');  
>> c1 = mat2huff(f1);  
>> entropy(f1)  
ans = 7.4253  
>> imratio(f1, c1)  
ans = 1.0704  
>> f2 = imread('Aligned Matches.tif');  
>> c2 = mat2huff(f2);  
>> entropy(f2)  
ans = 7.3505  
>> imratio(f2, c2)  
ans = 1.0821
```

```

function y = mat21pc(x, f)
%MAT2LPC Compresses a matrix using 1-D lossles predictive coding.
% Y = MAT2LPC(X, F) encodes matrix X using 1-D lossless predictive
% coding. A linear prediction of X is made based on the
% coefficients in F. If F is omitted, F = 1 (for previous pixel
% coding) is assumed. The prediction error is then computed and
% output as encoded matrix Y.
% See also LPC2MAT.
error(nargchk(1, 2, nargin)); % Check input arguments
if nargin < 2 % Set default filter if omitted
f = 1;
end
x = double(x); % Ensure double for computations
[m, n] = size(x); % Get dimensions of input matrix
p = zeros(m, n); % Init linear prediction to 0
xs = x; zc = zeros(m,1); % Prepare for input shift and pad
for j = 1:length(f) % For each filter coefficient ...
xs = [zc xs(:, 1:end -1)]; % Shift and zero pad x
p = p + f(j) * xs; % Form partial prediction sums
end
y = x - round(p); % Compute the prediction error

```

```

function x = Ipc2mat(y, f)
%LPC2MAT Decompresses a 1-D lossless predictive encoded matrix.
% X = LPC2MAT(Y, F) decodes input matrix Y based on linear
% prediction coefficients in F and the assumption of 1-D lossless
% predictive coding. If F is omitted, filter F = 1 (for previous
% pixel coding) is assumed.
% See also MAT2LPC.
error(nargchk(1, 2, nargin)); % Check input arguments
if nargin < 2 % Set default filter if omitted
f = 1;
end
f = f(end:-1:1); % Reverse the filter coefficients
[m, n] = size(y); % Get dimensions of output matrix
order = length(f); % Get order of linear predictor
f = repmat(f, m, 1); % Duplicate filter for vectorizing
x = zeros(m, n + order); % Pad for 1st 'order' column decodes
% Decode the output one column at a time. Compute a prediction based
% on the 'order' previous elements and add it to the prediction
% error. The result is appended to the output matrix being built.
for j = 1:n
jj = j + order;
x(:, jj) = y(:, j) + round(sum(f(:, order:-1:1) .* ...
x(:, (jj - 1):-1:(jj - order)), 2));
end
x = x(:, order + 1:end); % Remove left padding

```



```

function y = im2jpeg(x, quality)
%IM2JPEG Compresses an image using a JPEG approximation.
% Y = IM2JPEG(X, QUALITY) compresses image X based on 8 x 8 DCT
% transforms, coefficient quantization, and Huffman symbol
% coding. Input QUALITY determines the amount of information that
% is lost and compression achieved. Y is an encoding structure
% containing fields:
% Y.size      Size of X
% Y.numblocks Number of 8-by-8 encoded blocks
% Y.quality   Quality factor (as percent)
% Y.huffman   Huffman encoding structure, as returned by MAT2HUFF
% See also JPEG2IM.
error(nargchk(1, 2, nargin)); % Check input arguments
if ndims(x) ~= 2 | ~isreal(x) | ~isnumeric(x) | ~isa(x, 'uint8')
error('The input must be a UIKT8 image.');
```

```

end
if nargin < 2
quality = 1; % Default value for quality, end
m = [16 11 10 16 24 40 51 61    % JPEG normalizing array
     12 12 14 19 26 58 60 55    % and zig-zag reordering
     14 13 16 24 40 57 69 56    % pattern.
     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] * quality;
order = [1 9 2 3 10 17 25 18 11 4 5 12 19 26 33 ...
```

```

41 34 27 20 13 6 7 14 21 28 35 42 49 57 50 ...
43 36 29 22 15 8 16 23 30 37 44 51 58 59 52 ...
45 38 31 24 32 39 46 53 60 61 54 47 40 48 55 ...
62 63 56 64];

[xm, xn] = size(x);    % Get input size.
x = double(x) - 128;  % Level shift input
t = dctmtx(8);        % Compute 8x8 DCT matrix
% Compute DCTs of 8x8 blocks and quantize the coefficients.
y = blkproc(x, [8 8], 'PI * x * P2', t, t);
y = blkproc(y, [8 8], 'round(x ./ PI)', m);
y = im2col(y, [8 8], 'distinct'); % Break 8x8 blocks into columns
xb = size(y, 2);      % Get number of blocks
y = y(order, :);     % Reorder column elements
eob = max(x(:) + 1;  % Create end-of-block symbol
r = zeros(numel(y) + size(y, 2), 1);
count = 0;
for j = 1:xb          % Process 1 block (col) at a time
i = max(find(y(:, j))); % Find last non-zero element
if isempty(i)        % No nonzero block values
i = 0;
end
p = count + 1; q = p + i;
r(p:q) = [y(1:i, j); eob]; % Truncate trailing 0's, add EOB,
count = count + i + 1;  % and add to output vector
end
r((count + 1):end) = []; % Delete unused portion of r
y.size = uint16([xm xn]);
y.numblocks = uint16(xb);
y.quality = uint16(quality * 100);
y.huffman = mat2huff(r);

```

```

function x = jpeg2im(y)
%JPEG2IM Decodes an IM2JPEG compressed image.
% X = JPEG2IM(Y) decodes compressed image Y, generating
% reconstructed approximation X. Y is a structure generated by
% IM2JPEG.
% See also IM2JPEG.
error(nargchk(1, 1, nargin)); % Check input arguments
m = [16 11 10 16 24 40 51 61 % JPEG normalizing array
     12 12 14 19 26 58 60 55 % and zig-zag reordering
     14 13 16 24 40 57 69 56 % pattern.
     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];
order = [1 9 2 3 10 17 25 18 11 4 5 12 19 26 33 ...
         41 34 27 20 13 6 7 14 21 28 35 42 49 57 50 ...
         43 36 29 22 15 8 16 23 30 37 44 51 58 59 52 ...
         45 38 31 24 32 39 46 53 60 61 54 47 40 48 55 ...
         62 63 56 64];
rev = order; % Compute inverse ordering
for k = 1:length(order)
    rev(k) = find(order == k);
end
m = double(y.quality) / 100 * m; % Get encoding quality,
xb = double(y.numblocks); % Get x blocks,
sz = double(y.size);
xn = sz(2); % Get x columns,

```

```

xm = sz (1); % Get x rows.
x = huff2mat(y.huffman); % Huffman decode,
eob = max(x(:)); % Get end-of-block symbol
z = zeros(64, xb); k = 1; % Form block columns by copying
for j = 1:xb % successive values from x into
for i = 1:64 % columns of z, while changing
if x(k) == eob % to the next column whenever
k = k + 1; break; % an EOB symbol is found.
else
z(i, j) = x(k); k = k + 1;
end end end
z = z(rev, :); % Restore order
x = col2im(z, [8 8], [xm xn], 'distinct'); % Form matrix blocks
x = blkproc(x, [8 8], 'x .* P1', m); % Denormalize DCT
t = dctmtx(8); % Get 8x8 DCT matrix
x = blkproc(x, [8 8], 'P1 * x * P2', t, t); % Compute block DCT-1
x = uint8(x + 128); % Level shift

```

```
>> f = imread(Tracy.tif);
>> c1 = im2jpeg(f);
>> f1 ~ jpeg2im(c1);
>> imratio(f, c1)
ans =18.2450
>> compare(f, f1, 3)
ans =2.4675
>> c4 = im2jpeg(f, 4);
>> f4 = jpeg2im(c4);
>> imratio(f, c4)
ans =41.7826
>> compare(f, f4, 3)
ans =4.4184
```

```

function y = im2jpeg2k(x, n, q)
% IM2JPEG2K Compresses an image using a JPEG 2000 approximation.
% Y = IM2JPEG2KCX, N, Q) compresses image X using an N-scale JPEG
% 2K wavelet transform, implicit or explicit coefficient
% quantization, and Huffman symbol coding augmented by zero
% run-length coding. If quantization vector Q contains two
% elements, they are assumed to be implicit quantization
% parameters; else, it is assumed to contain explicit subband step
% sizes. Y is an encoding structure containing Huffman-encoded
% data and additional parameters needed by JPEG2K2IM for decoding.
% See also JPEG2K2IM.
global RUNS
error(nargchk(3, 3, nargin)); % Check input arguments
if ndims(x) ~= 2 | ~isreal(x) | ~isnumeric(x) | ~isa(x, 'uint8')
error('The input must be a UINT8 image.');
```

```

end
if length(q) ~= 2 & length(q) ~= 3 * n + 1
error('The quantization step size vector is bad.');
```

```

end
% Level shift the input and compute its wavelet transform,
x = double(x) - 128;
[c, s] = wavefast(x, n, 'jpeg9.7');
% Quantize the wavelet coefficients,
q = stepsize(n, q);
sgn = sign(c); sgn(find(sgn == 0)) = 1; c = abs(c);
for k = 1:n
qi = 3 * k - 2;
c = wavepaste('h', c, s, k, wavecopyC'h', c, s, κ) / q(qi));
c = wavepaste('v', c, s, k, wavecopyCv', c, s, k) / q(qi + 1));
c = wavepasteC'd', c, s, k, wavecopyOd', c, s, k) / q(qi + 2));
end
c = wavepasteCa', c, s, k, wavecopyCa', c, s, k) / q(qi + 3));
c = floor(c); c = c .* sgn;

```

```

% Run-length code zero runs of more than 10. Begin by creating
% a special code for 0 runs ('zrc') and end-of-code ('eoc') and
% making a run-length table.
zrc = min(c(:)) - 1; eoc = zrc - 1; RUNS = [65535];
% Find the run transition points: 'plus' contains the index of the
% start of a zero run; the corresponding 'minus*' is its end + 1.
z = c == 0; z = z - [0 z(1:end - 1)];
plus = find(z == 1); minus = find(z == -1);
% Remove any terminating zero run from 'c'.
if length(plus) ~= length(minus)
c (plus (end) :end) = [] ; c = [c eoc];
end
% Remove all other zero runs (based on 'plus' and 'minus') from 'c'.
for i = length(minus):-1:1
run = minus(i) - plus(i);
if run > 10
ovflo = floor(run / 65535); run = run - ovrflo * 65535;
c = [c(1:plus(i) - 1) repmat([zrc 1], 1, ovrflo) zrc ...
runcode(run) c(minus(i):end)];
end
end
% Huffman encode and add misc. information for decoding.
y.runs = uint16(RUNS);
y.s = uint16(s(:));
y.zrc = uint16(-zrc);
y.q = uint16(100 * q);
y.n = uint16(n);
y.huffman = mat2huff(c);
function y = runcode(x)
% Find a zero run in the run-length table. If not found, create a
% new entry in the table. Return the index of the run.
global RUNS
y = find(RUNS == x);

```

```

if length(y) ~= 1
RUNS = [RUNS; x];
y = length(RUNS);
end
function q = stepsize(n, p)
% Create a subband quantization array of step sizes ordered by
% decomposition (first to last) and subband (horizontal, vertical,
% diagonal, and for final decomposition the approximation subband).
if length(p) == 2 % Implicit Quantization
q = [];
qn = 2^(8 - p(2) + n) * (1 + p(1) / 2^1);
for k = 1:n
qk = 2^-k * qn;
q = [q (2 * qk) (2 * qk) (4 * qk)];
end
q = [q qk];
else % Explicit Quantization
q = p;
end
q = round(q * 100) / 100; % Round to 1/100th place
if any(100 * q > 65535)
error('The quantizing steps are not UINT16 representable.');
```

```

end
if any(q == 0)
error('A quantizing step of 0 is not allowed.');
```

```

end

```


im2jpeg2k.

```
function x = jpeg2k2im(y)
```

```
%JPEG2K2IM Decodes an IM2JPEG2K compressed image.
```

```
% X = JPEG2K2IM(Y) decodes compressed image Y, reconstructing an
```

```
% approximation of the original image X. Y is an encoding
```

```
% structure returned by IM2JPEG2K.
```

```
% See also IM2JPEG2K.
```

```
error (nargchkd, 1, nargin));    % Check input arguments
```

```
% Get decoding parameters: scale, quantization vector, run-length
```

```
% table size, zero run code, end-of-data code, wavelet bookkeeping
```

```
% array, and run-length table.
```

```
n = double(y.n);
```

```
q = double(y.q) / 100;
```

```
runs = double(y.runs);
```

```
rln = length(runs);
```

```
zrc = -double(y.zrc);
```

```
eoc = zrc - 1;
```

```
s = double(y.s);
```

```

s = reshape(s, n + 2, 2);
% Compute the size of the wavelet transform.
cl = prod(s(1, :));
for i = 2:n + 1
cl = cl + 3 * prod(s(i, :));
end
% Perform Huffman decoding followed by zero run decoding,
r = huff2mat (y.huffman);
c = [] ; zi = find(r == zrc) ; i = 1;
for j = 1:length(zi)
c = [c r(i:zi(j) - 1) zeros(1, runs(r(zi(j) + 1)))];
i = zi(j) + 2; end
zi = find(r == eoc); % Undo terminating zero run
if length(zi) == 1 % or last non-zero run.
c = [c r(i:zi - 1)] ;
c = [c zeros(1, cl - length(c))];
else
c = [c r(i:end)];
end
% Denormalize the coefficients,
c = c + (c > 0) - (c < 0);
for k = 1:n
qi = 3 * k - 2;
c = wavepasteC'h', c, s, k, wavecopyCh', c, s, k) * q(qi);
c = wavepasteC'v', c, s, k, wavecopy('v>', c, s, k) * q(qi + D);
c = wavepaste('d', c, s, k, wavecopyCd', c, s, k) * qCqi + 2));
end
c = wavepasteCa', c, s, k, wavecopyC'a*, c, s, k) * q(qi + 3));
% Compute the inverse wavelet transform and level shift.
x = waveback(c, s, 'jpeg9.7', n);
x = uint8(x + 128);

```

УДК 621.391.837:621.397.13

Денис Сіренко, Руслан Грималовський, Ірина Дедів, к.т.н., доцент
Тернопільський національний технічний університет імені Івана Пулюя, Україна

СПОСІБ ПІДСИЛЕННЯ ВІЗУАЛЬНОГО СПРИЙНЯТТЯ ЗОБРАЖЕНЬ НА ОСНОВІ СТОХАСТИЧНОГО РЕЗОНАНСУ

Denis Sirenko, Ruslan Hrymalovsky, Iryna Dediv, Ph.D., Assoc. Prof.

THE METHOD OF ENHANCING VISUAL PERCEPTION OF IMAGES ON THE BASIS OF STOCHASTIC RESONANCE

Поширеною в області оброблення, перетворення та зберігання даних є задача цифрового оброблення зображень для їх зберігання, передачі та подання в автономних системах машинного зору з однієї сторони та для підвищення якості зображень з метою покращення їх візуального сприйняття людиною. При цьому, актуальною сьогодні є друга група задач, оскільки при прийомі та передачі зображень, їхньому кодуванні/декодуванні часто погіршується якість візуального сприйняття, а також зростає кількість людей із вадами зору та порушеннями об'єктивного зорового сприйняття.

Суть підвищення якості зображень полягає в такій їхній обробці, щоб результат виявився більш відповідним з точки зору візуального сприйняття людиною. При цьому різні підходи до покращення якості зображень можна умовно розділити на дві групи, а саме: методи обробки в просторовій області (просторові методи) та методи обробки в частотній області (частотні методи). У випадку застосування методів першої групи застосовуються підходи з перетворення параметрів та координат окремих пікселів цифрового зображення. Підходи з другої групи полягають в модифікації сигналу, що формується шляхом застосування до зображення перетворення Фур'є. Перспективними також є методи покращення якості зображень у вигляді комбінації різних підходів цих двох груп. Варто зазначити, що загальної та єдиної теорії покращення якості зображень немає. Метою обробки зображень, які призначені для безпосереднього спостереження людиною, є підвищення візуальної інтерпретації. Однак, візуальне оцінювання якості зображення людиною є суб'єктивним процесом, який буде індивідуальним для кожної окремо взятої людини і визначатиметься її індивідуальними особливостями зорового сприйняття. Коли ж обробка зображення виконується для задач машинного сприйняття, завдання оцінювання якості отриманого результату обробки стає об'єктивним із можливістю застосування значної кількості об'єктивних методів. Тим не менше, навіть у ситуації, коли проблема дозволяє встановити чіткі критерії якості, потрібна певна кількість спроб тестування, доки буде обрано конкретний підхід до покращення зображень.

В дослідженнях проводиться розроблення методу покращення зображень для кращого візуального їхнього сприйняття людиною на основі застосування ефекту стохастичного резонансу, що полягає в підсиленні або збільшенні кількості інформації, що міститься в ослабленному чи зашумленному зображенні при дії на нього оптимального рівня шуму.

Література

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
2. Gammaitoni L. «Stochastic resonance»: Reviews of Modern Physics/ Gammaitoni L., Hanggi P., Jung P., Marchesoni F..-T.70 - №1