НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

«На правах рукопису»	«До захисту допущено»
УДК __004.051___	Завідувач кафедри _____ __Ткач М.М.__ (підпис) (ініціали, прізвище) “___”_____2015 р.

Магістерська дисертація

зі спеціальності __ 8.05020102 “Комп’ютеризовані та робототехнічні системи”__
(код та назва напряму спеціальності)

на тему: _____Покращення характеристик алгоритму стиснення JPEG з метою підвищення швидкості завантаження сайтів_____

Виконала: студентка _VI_ курсу, групи ___ІК-31м___
(шифр групи)

_____Бугаєнко Ольга Сергіївна _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник _____к.т.н. доц. Крилов Є.В._____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант_охорона праці __к.т.н. доц. Праховнік Н.А.____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

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

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2015 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

Факультет (інститут)____Інформатики та обчислювальної техніки____
(повна назва)

Освітньо-кваліфікаційний рівень _____«Магістр»_____
(назва ОКР)

Напрям підготовки _____6.050201 “Системна інженерія”_____
(код і назва)

Спеціальність _ 8.05020102 “Комп’ютеризовані та робототехнічні системи”
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ __Ткач М.М.__
(підпис) (ініціали, прізвище)

«___»_____2015 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
_____Бугаєнко Ользі Сергіївні _____
(прізвище, ім’я, по батькові)

1. Тема проекту (роботи) _____Покращення характеристик алгоритму стиснення JPEG з метою підвищення швидкості завантаження сайтів___

Науковий керівник дисертації_____Крилов Є.В._____к.т.н. _доцент_____,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «10» березня 2015 р. №292/2С

2. Строк подання студентом дисертації_____

3. Об’єкт дослідження_____Графічні зображення_____

4. Предмет дослідження_____Оптимізація за розміром та якістю_____

5. Перелік завдань які необхідно розробити_____ огляд сучасних методів стиснення зображень, модифікація алгоритму стиснення зображень JPEG, реалізація модифікованого алгоритму стиснення зображень, аналіз та порівняння результатів стиснення зображень модифікованим та звичайним алгоритмами, аналіз впливу зображень, стиснитих модифікованим алгоритмом, на швидкість завантаження сайту, аналіз виробничого приміщення

6. Орієнтовний перелік ілюстративного матеріалу блок схеми, результуючі зображення досліджень, схема виробничого приміщення

_

7. Орієнтований перелік публікацій Покращення характеристик алгоритму стиснення JPEG з метою підвищення швидкості завантаження сайтів / Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління».— Київ: Національний технічний університет України “Київський політехнічний інститут”. – 2015_

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Розділ охорони праці	Праховнік Н.А., доцент		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Ознайомлення із завданням		
2	Вивчення літератури		
3	Модифікація алгоритму		
4	Реалізація модифікованого алгоритму		
5	Проведення дослідження		
6	Аналіз результатів дослідження		
7	Документування		
8	Захист		

Студентка _____ Бугаєнко О.С.
(підпис) (ініціали, прізвище)

Керівник проекту (роботи) _____ Крилов Є.В.
(підпис) (ініціали, прізвище)

ВІДОМІСТЬ МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	А4		Завдання на дипломний проект	2	
2	А4		Пояснювальна записка	90	
3	А4		Код програми	12	
4	А3		Основні етапи стиснення цифрових зображень	1	
4	А3		Класифікація методів стиснення зображень	1	
4	А3		Основні етапи процедури стиснення за стандартом JPEG	1	
4	А3		Залежність якості зображення від кількості коефіцієнтів усереднення	1	
4	А3		Залежність якості зображення від ступеню стиснення	1	
4	А3		Швидкість завантаження сайту із зображеннями, стиснутими звичайним та модифікованим алгоритмами	1	

ABSTRACT

Work structure and scope. Explanatory note diploma consists of 4 sections, 90 pages, contains 34 figures, 10 tables, 1 application, 19 sources.

Master's research focuses on modifying image compression algorithm JPEG.

Research has show upgrade algorithm image compression JPEG, which will enable to increase the speed of download sites. The algorithm has been modified by aid agencies wavelet transform and implemented by the software MATLAB.

The work was conducted:

- review existing algorithms for image compression,
- analysis of the impact of the number of coefficients averaging image quality with wavelet transformation,
- modification JPEG image compression algorithm using wavelet transform,
- implementation of the modified algorithm
- analysis of the algorithms and modified JPEG JPEG,
- analysis of the impact of images compressed modified algorithm to speed the download site
- analysis of production facility.

The research was the implementation of the modified method of compressing JPEG images in MATLAB.

Key words:

Image compression, JPEG, wavelet transformation, modification.

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з 4 розділів, 90 сторінок, містить 34 рисунка, 10 таблиць, 1 додаток, 19 джерел.

Магістерська дисертація присвячена дослідженню модифікації алгоритму стиснення зображень JPEG.

Дослідження має показати покращення характеристик алгоритму стиснення зображень JPEG, що дасть можливість підвищити швидкість завантаження сайтів. Алгоритм було модифіковано за домогою вейвлет-перетворення та реалізовано за допомогою програмного забезпечення MATLAB.

У ході роботи було проведено:

- огляд існуючих алгоритмів стиснення зображень,
- аналіз впливу кількості коефіцієнтів усереднення на якість зображення при вейвлетному перетворенні,
- модифікацію алгоритму стиснення зображень JPEG за допомогою вейвлетного перетворення,
- реалізацію модифікованого алгоритму,
- аналіз результатів роботи алгоритмів JPEG та модифікованого JPEG,
- аналіз впливу зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту
- аналіз виробничогоприміщення.

Результатом дослідження стала реалізація модифікованого методу стиснення зображень JPEG у MATLAB.

Ключові слова:

Зображення, стиснення, JPEG, вейвлет-перетворення, модифікація.

Пояснювальна записка
до магістерської дисертації

на тему: Покращення характеристик алгоритму стиснення JPEG з метою підвищення швидкості завантаження сайтів

Київ – 2015 року

Зміст

АНОТАЦІЯ 6

Вступ 10

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ ЗОБРАЖЕНЬ 11

1.1. Представлення цифрових зображень 11

1.2. Класифікація методів стиснення. Основні характеристики 13

1.3. Алгоритми стиснення зображень без втрат 21

1.3.1. Стиснення способом кодування серій (RLE) 21

1.3.2. Стиснення за методом Хаффмана 22

1.3.3. Алгоритм Лемпеля-Зива (LZ-compression) 23

1.3.4. Алгоритм Лемпеля-Зива-Велч (Lempel-Ziv-Welch -LZW) 24

1.3.5. Алгоритм JBIG 24

1.3.6. Алгоритм Lossless JPEG 25

1.4. Алгоритми стиснення з втратами 26

1.4.1. Метод усіченого блочного кодування 26

1.4.2. Стиснення за стандартом JPEG 28

1.4.3. Стиснення за методом WIC (Wavelet Image Compression) 36

1.4.4. Фрактальне стиснення зображень 42

1.5. Постановка задачі 55

Висновки до розділу 55

РОЗДІЛ 2. МОДИФІКАЦІЯ АЛГОРИТМУ JPEG 57

2.1. Модифікований метод вейвлетного перетворення 57

2.2. Модифікований алгоритм JPEG 59

Висновки до розділу 61

РОЗДІЛ 3. ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ МОДИФІКАЦІЇ АЛГОРИТМУ JPEG 63

3.1. Аналіз результатів роботи модифікованого алгоритму 63

3.2. Вплив зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту 68

Висновки до розділу 70
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ 71
4.1. Загальна характеристика приміщення 71
4.2. Освітлення приміщення 73
4.3. Параметри мікроклімату 76
4.4. Шум та вібрація 78
4.5. Електромагнітне і іонізуюче випромінювання 80
4.6. Електробезпека 81
4.7. Протипожежна безпека 82
4.8. Дії при виникненні надзвичайних ситуацій 83
Висновки до розділу 86
ВИСНОВКИ 87
Список літератури 88
Додатки 90
Додаток А 91
Лістинг реалізації модифікованого алгоритму стиснення JPEG 91

ВСТУП

На сьогодняшній день одним з важливих факторів частого відвідування сайту користувачами є швидкість його завантаження. Особливу увагу слід приділити зображенням, оскільки в сучасному Інтернет вони складють основну частину контенту. Зображення повинні бути достатньо високої якості, але в той же час мати невеликий розмір файлу. Вимога до розміру файлу виникає із сучасних тенденцій серед користувачів – все частіше використовується мобільний інтернет, який має малу швидкість.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ ЗОБРАЖЕНЬ

1.1. Представлення цифрових зображень

Комп'ютерне зображення в його цифровому представленні є набором значень інтенсивностей світлового потоку, розподілених по кінцевої площі.

Для простоти розглянемо спочатку монохромні зображення. Інтенсивність випромінюваної світлової енергії з одиниці поверхні в точці з координатами (ξ, η) зображення можна представити деяким числом $B(\xi, \eta)$. Одиничний елемент зображення, що характеризується певним значенням (ξ, η), називається пікселем, а величина $z = f(\xi, \eta)$ – яскравістю[7].

Перш ніж розглянути алгоритми стиснення зображень, необхідно визначити що надалі буде розумітися під зображенням.

З математичної точки зору, зображення в градаціях сірого можна представити як дійсну функцію I двох дійсних змінних x і y. Функція $I(x, y)$ зображення в загальному випадку визначається в прямокутній області, але для зручності досліджень в роботі всі зображення визначаються в квадратних областях, тобто $x \in [0; W]$, а $y \in [0; H]$, де W – ширина зображення, а H – висота зображення і $W = H$.

Всі зображення можна поділити на дві групи: з палітрою і без неї. У зображень з палітрою у пікселі зберігається число – індекс в деякому одновимірному векторі кольорів, званому палітрою. Палітри зазвичай бувають 8, 16 і 256 кольорів[7].

Зображення без палітри зазвичай бувають в певній системі представлення кольору або в градаціях сірого. В градаціях сірого значення кожного з пікселів визначається як яскравість точки. Найбільш часто зустрічаються зображення з 2-ма, 16-ма і 256-ю рівнями сірого.

Якщо зображення відображається в якійсь системі представлення кольору, то кожен її піксель є структурою, що описує компоненти кольору. Найбільш поширеною системою представлення кольору, що використовується в

електронних та комп’ютерних системах, є система RGB. У цій системі колір визначається як комбінація червоного, зеленого та синього кольору. І на кожну з складових припадатиме по одному байту. Існують і інші системи представлення кольору, такі як CMYK, CIE, YUU і YCrCb[4].

Для того, щоб коректніше оцінити ступінь стиснення зображення та застосовності того чи іншого алгоритму стиснення до даного зображення вводиться поняття класу зображення.

Під класом цифрового зображення розуміється сукупність зображень, застосування до яких алгоритму стиснення дає якісно однаковий результат. Наприклад, для одного класу алгоритм стиснення дає чудовий коефіцієнт стиснення, а для іншого класу зображень навпаки, збільшує об'єм стискуваного файлу[11].

Умовно можна виділити наступні класи зображень:

- зображення з невеликою кількістю кольорів і великими областями, заповненими одним кольором. У зображенні відсутні плавні переходи кольорів. До таких класах зазвичай відноситься ділова графіка, науково-технічна, інженерна або плакатна графіка;
- зображення з плавними переходами кольорів, побудовані на комп'ютері: графіка презентацій та віртуальні моделі;
- фотореалістичні зображення, отримані після цифрової фотозйомки, сканування, а також постобробка цих зображень.

Можна виділити і специфічні класи зображень, такі як рентгенівські знімки, томаграфічні зображення, радіолокаційні плани місцевості і т.д. Але для порівняння алгоритмів стиснення зображень завжди необхідно визначати клас зображень, з якими вони працює.

У процесі роботи із зображеннями додатки, які здійснюють обробку, висувають різні вимоги до алгоритмів стиснення зображень. Через специфіку додатків такі вимоги іноді можуть суперечити один одному. У загальному випадку можна виділити наступні вимоги до алгоритмів стиснення зображень:

- високий ступінь компресії;
- висока якість стисненого зображення (дана вимога суперечить виконанню попереднього вимоги, тому завжди доводиться шукати компроміс між ступенем стиснення і якістю відновленого зображення);
- висока швидкість компресії (дана вимога актуально для додатків, що займаються кодуванням зображень в реальному масштабі часу: цифрових фотоапаратів, відеокамер);
- висока швидкість декомпресії (дана вимога актуально майже для всіх додатків);
- можливість показати приблизне зображення, не чекаючи повної його завантаження (дана вимога актуально для мережевих додатків і для додатків, що займаються передачею великих зображень);
- врахування специфіки зображення (дану вимога реалізують алгоритми стиснення, засновані на визначенні «області особливого призначення» (ROI - regions of interest)).

1.2. Класифікація методів стиснення. Основні характеристики

Всі методи стиснення інформації засновані на простому припущенні, що набір даних завжди містить надлишкові елементи. Стиснення досягається за рахунок пошуку та кодування надлишкових елементів.

Потік даних про зображення має істотну кількість зайвої інформації, яка може бути усунена практично без помітних для ока спотворень. При цьому розрізняють два типи надмірності.

Статистична надлишковість пов'язана з кореляцією і передбачуваністю даних. Ця надмірність може бути усунена без втрати інформації, вихідні дані при цьому можуть бути повністю відновлені. Найбільш відомі методи ефективного кодування символів засновані на знанні частоти кожного символу присутнього в повідомленні. Знаючи ці частоти, будують таблицю кодів, що володіє наступними властивостями:

- різні коди можуть мати різну кількість біт;
- коди символів, які зустрічаються з більшою частотою, мають менше біт, ніж коди символів з меншою частотою;
- хоча коди мають різну бітову довжину, вони можуть бути відновлені єдиним чином, тобто коди будуються як префіксні.

Цими властивостями володіє відомий алгоритм Хаффмана[7].

Візуальна (суб'єктивна) надлишковість, яку можна усунути з частковою втратою даних, мало впливають на якість відтворюваних зображень; це – інформація, яку можна вилучити з зображення, не порушуючи візуальне сприйняття якості зображень.

Усунення візуальної надмірності зображень є основним резервом скорочення інформації, що передається[7,11]. Для оптимізації процесу кодування з метою забезпечення передачі найменшого обсягу інформації необхідно, з одного боку, не передавати надлишкову інформацію, а з іншого, – не допустити надмірної втрати якості зображення.

Досі не існує простої та адекватної моделі візуального сприйняття зображень, придатної для оптимізації їх кодування.

Завдання стиснення зображення складається з двох основних частин: кодування і декодування. Якщо декодоване зображення завжди в точності відповідає кодованому зображенню, то такий алгоритм кодування-декодування називається алгоритмом стиснення без втрат. Якщо декодувати зображення відрізняється від кодованого, то подібний алгоритм називають алгоритмом стиснення з втратами. Загальна схема процесу стиснення зображення представлена на рис 1.1.

Розглянемо етапи процедури стиснення даних в загальному вигляді. Будь-який метод стиснення реалізує три основних етапи (рис.1.1.):

- кодування або первинне стиснення;
- вторинне стиснення;
- декодування або відновлення зображення.

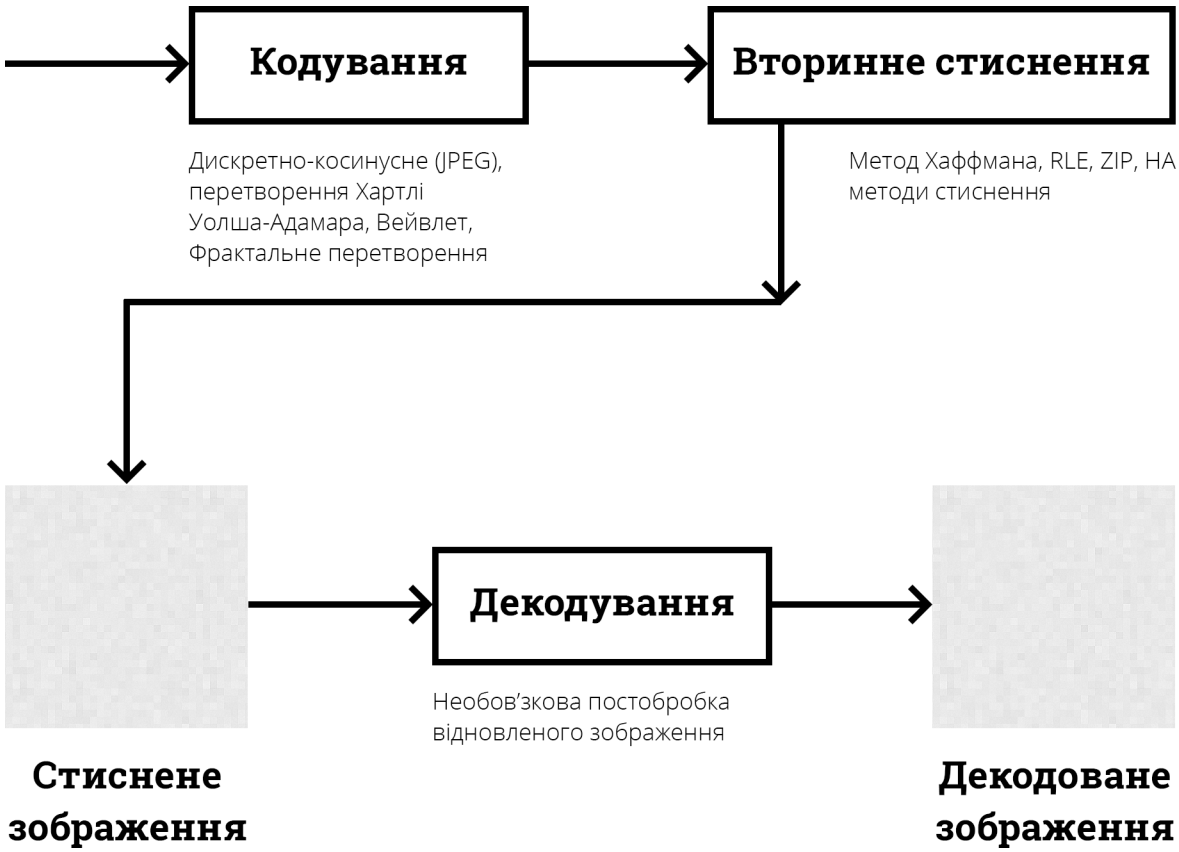

Рис.1.1. Основні етапи стиснення цифрових зображень

На першому етапі виконується перетворення вихідних даних з однієї форми подання в іншу. Зокрема, при стисненні зображень залежно від виду алгоритму стиснення може бути виконаний перехід від початкового зображення до наступних видів подання (табл.1.1.).

На другому етапі компоненти перетворення квантуються і зводяться до виду зручного для статистичного кодування, а потім кодуються. На цьому етапі забезпечується ущільнення інформаційного потоку.

Існує декілька різних підходів до проблеми стиснення інформації. Одні мають досить складну теоретичну математичну базу, інші засновані на властивостях інформаційного потоку і алгоритмічно досить прості. Будь-який спосіб, який реалізує стиснення даних, призначений для зниження обсягу вихідного потоку інформації за допомогою перетворення без втрат або перетворення з втратами. Тому всі способи стиснення можна розділити на дві категорії: стиснення без втрат та з втратами. Методи стиснення цифрових

зображень можна класифікувати за їх основним характеристикам, таким як: точність відновлення, симетричність основного перетворення і тип використовуваного перетворення[9].

Таблиця 1.1. Переходи від початкового зображення

Вихідне зображення	*Перетворене зображення*
Матриця пікселів (значення інтенсивності)	Матриця компонент спектру (спектральні перетворення)
	Набір коефіцієнтів перетворення (фрактальне стиснення)
	Опис об’єктів перетворення (стиснення з розпізнаванням)

Класифікація найбільш поширених методів стиснення наведена на рис.1.2.

Стиснення без втрат. Результатом стиснення без втрат завжди є зниження обсягу вихідного потоку інформації без зміни його інформативності, тобто без втрати інформаційної структури. Більше того, з вихідного потоку, за допомогою відновлюючого алгоритму, можна отримати вхідний[7].

Стиснення з втратами. Під стисненням з втратами мається на увазі таке перетворення вхідного потоку даних, при якому вихідний потік, що базується на певному форматі інформації, представляє досить схожий за зовнішніми характеристиками на вхідний потік об'єкт, проте відрізняється від нього об'ємом. Ступінь подібності вхідного і вихідного потоків визначається ступенем відповідності деяких властивостей об'єктів, що представляється даним потоком інформації.

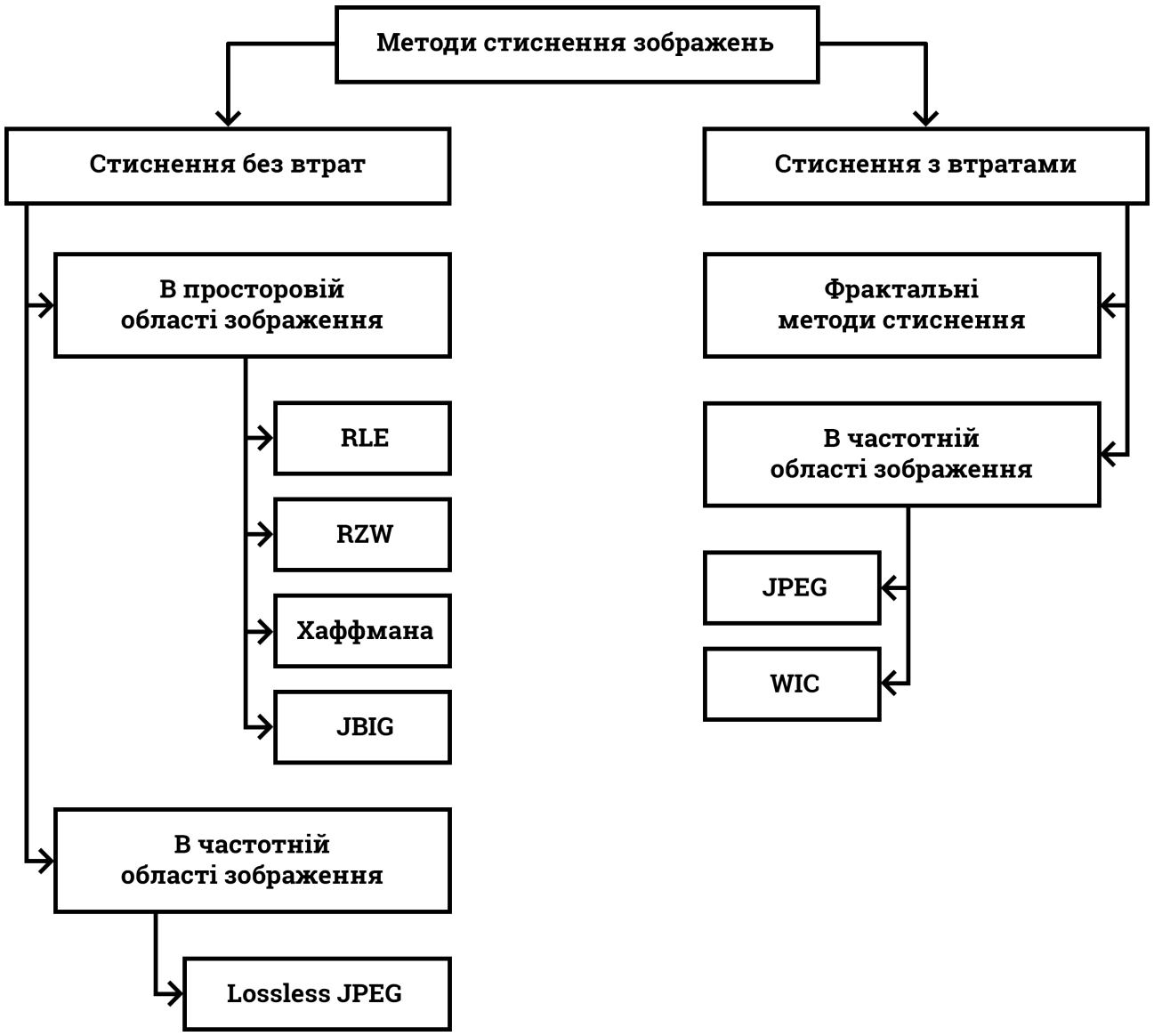

Рис.1.2. Класифікація методів стиснення зображень

Такі алгоритми використовуються для стиснення, наприклад даних растрових графічних файлів з низьким ступенем повторюваності байтів в потоці. При такому підході використовується властивість структури формату графічного файлу і можливість представити графічну картинку приблизно схожу за якістю відображення (для сприйняття людським оком) декількома способами. Тому, крім ступеня або величини стиснення, в таких алгоритмах виникає поняття якості, тому вихідне зображення в процесі стиснення змінюється. Під якістю можна розуміти ступінь відповідності вихідного і результуючого зображення. Для графічних файлів таку відповідність

визначається візуально, хоча є і відповідні формалізовані методики та оцінки[4]. Стиснення з втратами неможливо застосовувати в областях, в яких необхідно мати точну відповідність інформаційної структури вхідного і вихідного потоків.

Методи стиснення без втрат використовуються в основному в наукових і медичних прикладних програмах, коли втрата інформації неприпустима або самі шуми зображення є головною інформацією, наприклад, в системах оцінки якості оптико-електронних систем. Коефіцієнт стиснення досягається цими методами не більше 1,5. Методи стиснення з втратами дозволяють отримати істотно більші коефіцієнти стиснення.

Однак при цьому відбувається спотворення вихідного зображення, погіршення його якості. У зв'язку з цим при порівнянні різних методів стиснення крім коефіцієнта стиснення потрібно враховувати якість відновлення зображення.

Для симетричних методів стиснення процедури стиснення і відновлення однотипні. Час стиснення і відновлення для таких методів порівнянні. Для несиметричних методів процедура стиснення відрізняється від процедури відновлення і зазвичай займає більше машинного часу.

Визначимо основні величини, що характеризують метод стиснення.

1. Коефіцієнт стиснення ($К_{ст}$).

$$K_{\text{ст}} = \frac{V_1}{V_2} \qquad (1.1)$$

Цей параметр визначає у скільки разів файл, що зберігає стиснене зображення, менше файлу, що зберігає вихідне зображення. Величини V_1 і V_2 виражаються в байтах. $K_{\text{ст}}$ – величина безрозмірна.

2. Оцінка якості декодованого зображення.

Одна з проблем машинної графіки полягає в тому, що до цих пір не знайдений адекватний критерій оцінки втрат якості зображення[7].

Якість втрачається при оцифровуванні, при переведенні в обмежену палітру кольорів або в інший простір кольорів, а також при стисненні зображень з втратами.

Нехай є два зображення: *f (x, y)* – оригінал, і $\hat{f}(x,y)$ – відновлене зображення розміром *M* × *N*, тоді одним з простих критеріїв оцінки втрати якості є середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу:

$$d(x,y) = \sqrt{\frac{\sum_{x,y}^{M,N}(f(x,y)-\hat{f}(x,y))^2}{M \cdot N}} \quad (1.2)$$

За цим критерієм зображення буде сильно зіпсоване при зміні яскравості всього на 5%.

Іншим критерієм є максимальне відхилення від оригіналу:

$$d(x,y) = \max_{x,y}\left|f(x,y) - \hat{f}(x,y)\right| \quad (1.3)$$

Дана міра вкрай чутлива до биття окремих пікселів, тобто в зображенні може змінитися тільки один піксель, і цей критерій визнає зображення сильно зіпсованим.

На практиці використовуваною мірою якості зображення є критерій співвідношення сигнал/шум (PSNR).

$$d(x,y) = 10\log_{10}\frac{255^2 \cdot M \cdot N}{\sum_{x,y}^{M,N}(f(x,y)-\hat{f}(x,y))^2} \quad (1.4)$$

Ця міра аналогічна середньоквадратичному відхиленню, але користуватися нею зручніше через логарифмічний масштаб шкали. Найкраще втрати в якості оцінює людське око. Стиснення зображення можна вважати відмінним, якщо на око неможливо відрізнити оригінал від стисненого

зображення. Але на практиці при стисненні з втратами в зображення завжди вносяться які-небудь спотворення помітні при порівнянні оригіналу та стисненого зображення. До інших найбільш уживаним критеріям оцінки якості зображення належать:

Середня різниця:

$$AD = \sum_{x=1}^{M} \sum_{y=1}^{N} \left[f(x,y) - \hat{f}(x,y) \right] / (MN) \tag{1.5}$$

Коефіцієнт крос-кореляції:

$$K = \sum_{x=1}^{M} \sum_{y=1}^{N} f(x,y) \cdot \hat{f}(x,y) / \sum_{x=1}^{M} \sum_{y=1}^{N} [f(x,y)]^2 \tag{1.6}$$

Вірність зображення (image fidelity):

$$AD = 1 - \left(\sum_{x=1}^{M} \sum_{y=1}^{N} \left[f(x,y) - \hat{f}(x,y) \right]^2 / \sum_{x=1}^{M} \sum_{y=1}^{N} [f(x,y)]^2 \right) \tag{1.7}$$

3. Час перетворення

Розрізняють дві величини – час стиснення ($t_{ст}$) і час відновлення ($t_{відн}$).

Час стиснення складається з часу роботи основного перетворення ($t_{оп}$) і часу упаковки ($t_{уп}$). Час відновлення складається з часу розпакування ($t_{розп}$) і часу роботи зворотного перетворення ($t_{зп}$):

$$\begin{aligned} t_{\text{ст}} &= t_{\text{оп}} + t_{\text{уп}} \\ t_{\text{відн}} &= t_{\text{розп}} + t_{\text{зп}} \end{aligned} \tag{1.8}$$

Для того, щоб коректно оцінювати алгоритми стиснення-відновлення необхідно задати певні критерії:

- найгірший, середній і кращий коефіцієнт стиснення. (Інакше кажучи, розкид коефіцієнта стиснення, якщо вихідні дані будуть найгіршими; якийсь середньостатистичний коефіцієнт для того класу зображень, на який орієнтований алгоритм; і, нарешті, кращий коефіцієнт, причому останній має лише теоретичне значення, оскільки показує ступінь стиснення найкращого зображення);
- клас зображень, на який орієнтований алгоритм (іноді вказують, чому на інших класах зображень виходять гірші результати);
- симетричність – характеризує ресурсомісткість процесів кодування і декодування (при цьому найбільш важливим є відношення часу кодування до часу декодування);
- втрати якості (у більшості алгоритмів стиснення з втратою інформації існує можливість зміни коефіцієнта стиснення);
- характерні особливості алгоритму і зображень, до яких його застосовують[11].

1.3. Алгоритми стиснення зображень без втрат

1.3.1. Стиснення способом кодування серій (RLE)

Найбільш відомий і простий алгоритм стиснення інформації без втрат – це кодування серій послідовностей (Run Length Encoding – RLE)[9]. Суть даного підходу полягає в заміні ланцюжків або серій повторюваних байтів або їх послідовностей на один кодуючий байт і лічильник числа їх повторень. Проблема всіх аналогічних методів полягає лише у визначенні способу, за допомогою якого алгоритм розпакування міг би відрізнити в результуючому потоці байтів кодовану серію від інших – некодованих послідовностей байтів. Рішення проблеми досягається зазвичай простановкою міток на початку кодованих ланцюжків. Такими позначками можуть бути, наприклад, характерні значення бітів в першому байті кодованої серії, значення першого байта

кодованої серії і т.п. Кращий, середній і найгірший коефіцієнти стиснення - 1/32, 1/2, 2/1.

Дані методи, як правило, досить ефективні для стиснення растрових графічних зображень (BMP, PCX, TIFF), так як останні містять досить довгі серії повторюваних послідовностей байтів.

Недоліком методу RLE є досить низька ступінь стиснення або вартість кодування файлів з малим числом серій і, що ще гірше — з малим числом повторюваних байтів в серіях. До позитивних сторін алгоритму, мабуть, можна віднести тільки те, що він не вимагає додаткової пам'яті при роботі, і швидко виконується. Цікава особливість групового кодування у форматі PCX полягає в тому, що ступінь архівації для деяких зображень може бути істотно підвищений всього лише за рахунок зміни порядку кольорів у палітрі зображень.

1.3.2. Стиснення за методом Хаффмана

Кодування Хаффмана частіше за інших вживається для текстових файлів, яке полягає в тому, що символи тексту замінюються ланцюжками біт різної довжини. Методика Хаффмана гарантує однозначну побудову коду з найменшим для даного розподілу ймовірностей середнім числом символів на букву[7].

Щодо стиснення зображень, то в основі такого методу лежить облік частоти появи однакових байт в зображенні. При цьому пікселям вихідного зображення, які зустрічаються більше число разів, зіставляється код меншої довжини, а пікселям, що зустрічаються рідко – код більшої довжини (тобто формується префіксний код змінної довжини). Для збору статистики потрібно два проходи по файлу – один для перегляду та збору статистичної інформації, другий – для кодування. Коефіцієнти стиснення: 1/8, 2/3, 1[11].

При використанні такого методу потребується запис у файл і таблиці відповідності кодованих пікселів та кодуючих ланцюжків. Таке кодування

застосовується в якості останнього етапу архівації в JPEG. Методи Хаффмана дають досить високу швидкість і помірно добру якість стиснення.

Основним недоліком даного методу є залежність ступеню стиснення від близькості ймовірностей символів до величини 2^{-M}, оскільки кожен символ кодується цілим числом біт. Так, при кодуванні даних з двохсимвольним алфавітом стиснення завжди відсутнє, тому незважаючи на різні ймовірності появи символів у вхідному потоці алгоритм фактично зводить їх до 1/2. Такий алгоритм реалізований у форматі TIFF.

1.3.3. Алгоритм Лемпеля-Зива (LZ-compression)

Суть даного алгоритму полягає в наступному: пакувальник постійно зберігає деяку кількість останніх оброблених символів у буфері. У міру обробки вхідного потоку новоприйняті символи потрапляють в кінець буфера, зсуваючи попередні символи і витісняючи найстаріші. Розміри цього буфера, званого також ковзаючим словником, варіюються в різних реалізаціях кодуючих систем. Потім, після побудови хеш таблиць, виділяють (шляхом пошуку в словнику) найдовший початкову підрядок вхідного потоку, який збігається з одним з підрядків в словнику, і видають на вихід пару (length, distance), де length – довжина знайденого в словнику підрядка, а distance – відстань від нього до вхідного підрядка (тобто фактично індекс підрядка в буфері, вирахований з його розміру). Якщо такий підрядок не знайдено, у вихідний потік просто копіюється черговий символ вхідного потоку[9].

Існує досить велике сімейство LZ-подібних алгоритмів, які розрізняються, наприклад, методом пошуку повторюваних ланцюжків. Один з досить простих варіантів цього алгоритму, наприклад, припускає, що у вхідному потоці йде або пара <лічильник, зсув щодо поточної позиції>, або просто <лічильник> байт “що пропускаються ” і самі значення байтів.

При розархівації для пари <лічильник, зміщення> копіюються <лічильник> байт з вихідного масиву, отриманого в результаті розархівації, на <зміщення>

байт раніше, а <лічильник> (тобто число рівне лічильнику) значень байт “що пропускаються” просто копіюються у вихідний масив з вхідного потоку.

Даний алгоритм є несиметричним по часу, оскільки вимагає повного перебору буфера при пошуку однакових підрядків.

До переваг LZ можна віднести надзвичайну простоту алгоритму декомпресії.

1.3.4. Алгоритм Лемпеля-Зива-Велч (Lempel-Ziv-Welch -LZW)

Даний алгоритм відрізняється високою швидкістю роботи як при упаковці, так і при розпакуванні, досить скромними вимогами до пам'яті і простою апаратною реалізацією. Недолік – низький ступінь стиснення у порівнянні зі схемою двоступеневого кодування. Алгоритм перетворює потік символів на вході в потік індексів комірок словника на виході. Існує досить велике сімейство LZW-подібних алгоритмів, що розрізняються, наприклад, методом пошуку повторюваних ланцюжків[9].

Коефіцієнти стиснення: 1/1000, 1/4, 7/5. Коефіцієнт 1/1000 досягається тільки на одноколірних зображеннях розміром більше 4 Мб. Ситуація, коли алгоритм збільшує зображення, зустрічається вкрай рідко. Стиснення забезпечується за рахунок однакових підланцюжків у потоці. Алгоритм є майже симетричним, за умови оптимальної реалізації операції пошуку рядка в таблиці. LZW універсальний – саме його варіанти використовуються в звичайних архіваторах. Він реалізований у форматах GIF, TIFF і TGA.

1.3.5. алгоритм JBIG

Алгоритм розроблений групою експертів ISO (Joint Bi-level Experts Group) спеціально для стиснення однобітних чорно-білих зображень (наприклад, для факсів або відсканованих документів). В принципі може застосовуватися і до 2-х, і до 4-х бітових зображень. При цьому алгоритм розбиває їх на окремі бітові площини. JBIG дозволяє керувати такими параметрами, як порядок розбиття зображення на бітові площини, ширина смуг

в зображенні, рівні масштабування. Остання можливість дозволяє легко орієнтуватися в базі великих за розмірами зображень, переглядаючи спочатку їх зменшені копії. Налаштовуючи ці параметри, можна використовувати цікавий ефект при отриманні зображення по мережі або по будь-якому іншому каналу, пропускна здатність якого мала в порівнянні з можливостями процесора. Розпаковуватися зображення на екрані буде поступово, як би повільно "проявляючись". При цьому людина починає аналізувати картинку задовго до кінця процесу розархівації. Алгоритм побудований на базі Q-кодувальника, патентом на який володіє IBM. Q-кодер також, як і алгоритм Хаффмана, використовує для символів, що частіше з'являються, короткі ланцюжки, а для тих, що з'являються рідше, довгі. Однак, на відміну від нього, в алгоритмі використовуються і послідовності символів. Характерною особливістю JBIG є різке зниження ступеню стиснення при підвищенні рівня шумів вхідного зображення[11].

1.3.6. Алгоритм Lossless JPEG

Цей алгоритм розроблений групою експертів в області фотографії (Joint Photographic Expert Group). На відміну від JBIG, Lossless JPEG орієнтований на повнокольорові 24-бітові зображення[7].

Коефіцієнти стиснення: 1/20, 1/2, 1.

Стандарт стиснення зображень JPEG включає два способи стиснення: перший призначений для стиснення без втрат, другий - стиснення з втратою якості. Метод стиснення без втрат, використовуваний в стандарті lossless JPEG заснований на методі різницевого (диференціального) кодування. Основна ідея диференціального кодування полягає в наступному. Зазвичай зображення характеризуються сильною кореляцією між точками зображення. Цей факт враховується при різницевому кодуванні, а саме, замість стиснення послідовності точок зображення $x_1, x_2, ... x_N$, стисненню піддається послідовність різниць $y_i = x_i - x_{i-1}$, $i = 1,2, ... N$, $x_0 = 0$. Числа y_i називають

помилками передбачення x_i. У стандарті lossless JPEG передбачене формування помилок передбачення з використанням попередніх закодованих точок в поточному рядку та/або в попередньому рядку.

Lossless JPEG рекомендується застосовувати в тих додатках, де необхідна побітова відповідність вихідного і розархівованого зображень.

Спробуємо зробити деякі узагальнення. З одного боку, наведені алгоритми досить універсальні і покривають всі типи зображень, з іншого – вони, за сьогоднішніми мірками, забезпечують занадто маленький коефіцієнт архівації. Використовуючи один з алгоритмів без втрат, можна забезпечити коефіцієнт архівації зображення приблизно в два рази. У той же час стиснення з втратами оперують з коефіцієнтами 10-200 разів. Крім можливості модифікації зображення, одна з основних причин подібної різниці полягає в тому, що традиційні алгоритми орієнтовані на роботу з ланцюжком. Вони не враховують так звану "когерентність областей" в зображеннях. Ідея когерентності областей полягає в малій зміні кольору і структури зображення на невеликій ділянці. Всі алгоритми, про які йтиметься нижче, були створені пізніше спеціально для стиснення графіки і використовують цю ідею.

1.4. Алгоритми стиснення з втратами

1.4.1. Метод усіченого блочного кодування

Назва методу відображає той факт, що зображення розбивається на невеликі прямокутні ділянки однакового розміру, які називають блоками. Цей метод на відміну від більшості інших підлаштовує параметри кодування не під деяку усереднену характеристику всього зображення, а під локальні особливості в межах кожного блоку. Це дозволяє зберегти дрібні деталі зображень. Метод не призводить до розмивання меж, що є характерним для деяких інших алгоритмів. Метод УБК порівнянний з більшістю інших методів по ефективності стиснення даних і за обсягом обчислень, необхідних для кодування, але не має конкурентів за простотою декодування.

Базовий алгоритм УБК будується наступним чином[9]. Зображення, представлене M × N – матрицею $\|bij\|$ яскравості пікселів, розбивається на невеликі прямокутні блоки $m \times n$ елементів. Кожен такий блок обробляється незалежно від інших, тому опишемо алгоритм обробки одного блоку.

Обробка блоку починається з обчислення порога і двох рівнів квантування, потім проводиться квантування блоку на два рівня, після чого слідує упаковка проквантованного блоку. Для визначення рівнів квантування спочатку обчислюються два перших вибіркових моменти – середнє значення C і середній квадрат E:

$$C = \frac{1}{m \times n} \sum_i \sum_j b_{ij}\,,\ E = \frac{1}{m \times n} \sum_i \sum_j b_{ij}^2 \tag{1.9}$$

(де підсумовуються елементи зображення в межах блоку) і дисперсія

$$\sigma^2 = E - c^2 \tag{1.10}$$

Порогова величина квантування d покладається рівною середньому C. Верхній a і нижній b рівні квантування обчислюються за такими формулами:

$$a = C - \sigma\sqrt{q/(p-q)},\ b = C + \sigma\sqrt{(p-q)/q} \tag{1.11}$$

де $p = m \times n$ – число елементів блоку, q – число елементів, що перевищують поріг d.

Квантування проводиться за правилом:

$$s_{ij} = \begin{cases} a, \text{якщо } b_{ij} < d \\ b, \text{якщо } b_{ij} \geq d \end{cases} \tag{1.12}$$

де s_{ij} – елементи зображення після квантування.

Після квантування виходить блок, що містить тільки рівні *a* та *b*. Неважко показати, що середнє значення і середній квадрат вихідного і проквантованного блоків збігаються. Практично для зручності подальшої упаковки замість *a* записується нуль, замість *b* – одиниця. Рівні *a* і *b* записуються окремо.

Упаковка полягає в тому, що блок, який містить лише нулі і одиниці, інтерпретується як двійкове число, що має $m \times n$ розрядів. Відновлення закодованого зображення також проводиться по блоках і полягає в розпакуванні і зворотній підстановці.

З алгоритму видно, що ступінь стиснення безпосередньо залежить від розмірів блоку. Найбільш задовільні результати, як за ступенем стиснення, так і за якістю відновленого зображення були отримані при використанні блоків розміром 4×4[11].

Описаний вище спосіб визначення порогу і рівнів квантування НЕ є єдиним. Існує ряд інших критеріїв. Важливо, щоб критерій відповідав цілям подальшої обробки зображення і її конкретним особливостям.

1.4.2. Стиснення за стандартом JPEG

JPEG – один з найпоширеніших і досить потужних алгоритмів, являє собою метод стиснення зображень, що реалізується різними способами[7,9]. Працює він як на чорно-білих, так і на повнокольорових зображеннях.

Стандарт JPEG (Joint Photographic Experts Group - Об'єднана експертна група по фотографії) – формат зберігання фотографічних зображень, що відрізняється гарною якістю відновленого зображення. Процес стиснення зображення JPEG досить складний і часто для досягнення прийнятної продуктивності вимагає спеціальної апаратури. Схема процедури стиснення зображень за стандартом JPEG наведена на рис.1.3.

Кодування зображень за стандартом JPEG зазвичай починається з перетворення колірного простору з RGB в YUV (відоме також під назвою

YCbCr). Кольорове зображення традиційно може розглядатися як результат складання трьох компонент:

$$X_{ijC} = a_1 X_{ij(R)} + a_2 X_{ij(G)} + a_3 X_{ij(B)} \tag{1.13}$$

У цьому виразі a_1, a_2, a_3 — калориметричні коефіцієнти.

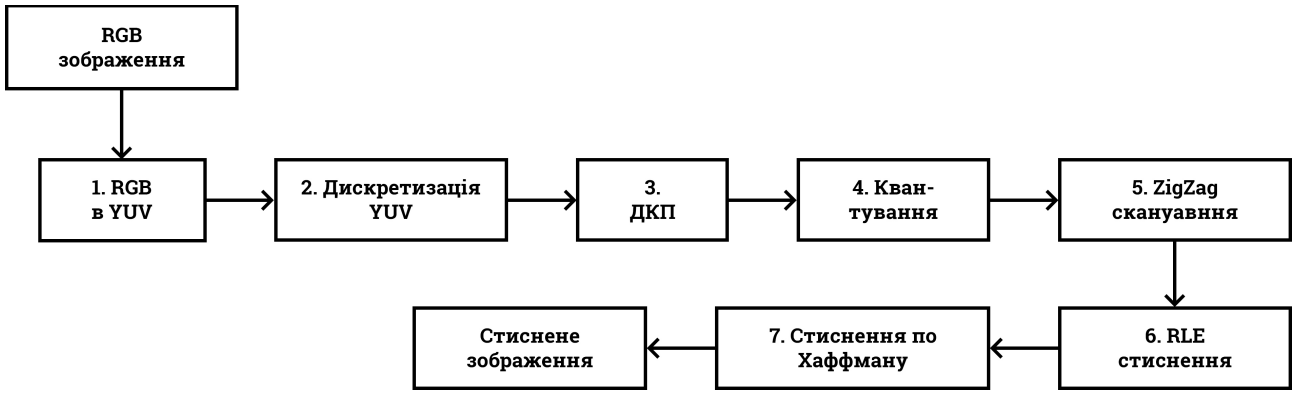

Рис.1.3. Основні етапи процедури стиснення за стандартом JPEG

У типових зображеннях у форматі RGB є істотна кореляція між кольоровими компонентами і з точки зору стиснення зображення цей формат є завідомо надлишковим. Як відомо, в стандартах телевізійного мовлення використовується інше представлення зображень, при якому також використовуються 3 компоненти сигналу, але при цьому ці компоненти майже некорельовані один з одним. Компоненти *R, G і B* перетворюються в яскравісну компоненту *Y* і дві різнокольорові компоненти *U* і *V*, формату YUV.

Перетворення форматів виконується за формулами:

Для перетворення $RGB \rightarrow YUV$:

$$\begin{vmatrix} Y \\ U \\ V \end{vmatrix} = \begin{vmatrix} 0.299 & 0.578 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \tag{1.14}$$

і для перетворення $YUV \rightarrow RGB$:

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{vmatrix} * \left(\begin{vmatrix} Y \\ U \\ V \end{vmatrix} - \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \right) \quad (1.15)$$

У форматі YUV компоненти слабо корельовані. Більше того, оскільки більша частина інформації зосереджена в компоненті яскравості, то буде втрачено мало інформації, якщо виконати проріджування компонент *U* і *V* з коефіцієнтом 2. При такому проріджуванні 4 сусідні точки (квадрат 2x2) описуються 4 значеннями компонентів *Y* та по одному значенню компонент *U* і *V*.

Результатом є стандартний формат YUV 4: 1: 1, який, як правило, є вхідним для більшості відеокодерів. Таким чином, виходить стиснення в 2 рази без скільки-небудь помітного спотворення зображення.

Стандарт не зобов'язує виконувати цю операцію, проте такий підхід дозволяє підвищити ефективність стиснення.

Далі вихідне зображення розбивається на матрицю клітин однакового розміру (найчастіше 8 × 8 пікселів) .Такий розмір обрано з наступних причин [9]:

1) З точки зору апаратної й програмної реалізації розмір блоку 8x8 не накладаються істотних обмежень на розмір необхідної пам'яті.
2) Обчислювальна складність ДКП для блоку 8x8 також є прийнятною для більшості обчислювальних платформ.

Наступний етап процедури стиснення даних полягає в перетворенні невеликих блоків зображення за допомогою двовимірного косинусного перетворення (ДКП). Обробка ведеться блоками 8x8 пікселів. Вибір ДКП в якості стандартного рішення диктується наступними причинами:

- Для зображень з сильно корельованими відліками (коефіцієнт кореляції >0,7) ефективність ДКП в сенсі компактності представлення даних близька до перетворення Карунена-Лоева (це перетворення є

оптимальним в тому сенсі, що воно ортонормоване і гарантує некорельованість коефіцієнтів перетворення – елементів *Y*).

- ДКП являє собою ортогональное сепарабельне перетворення, незалежне від зображення, тому його обчислювальна складність невелика.

Обробка кожного блоку виконується незалежно і полягає у виконанні ДКП по рядках і стовпцях блоку, яке має вигляд:

$$F(u,v) = \frac{c(u,v)}{4} \sum_{x=0}^{1} \sum_{y=0}^{1} f(x,y) \cos\left(\frac{2x+1}{16} ux\right) \cos\left(\frac{2y+1}{16} vx\right) \qquad (1.16)$$

У виразі (1.16) множник *C (u, v)* є нормуючим і дорівнює *1 / √2* при *u = 0* або *v = 0* і дорівнює одиниці для інших значень індексів.

Загальним недоліком дискретних ортогональних перетворень є їх висока обчислювальна складність. У зв'язку з цим використовуються так звані швидкі алгоритми виконання косинусного перетворення. Відомі з літератури алгоритми швидких перетворень в базисах косинусних функцій, хоча і відрізняються меншим числом операцій множення, але вимагають додаткових перекомпоновок після кожної ітерації алгоритму. Одномірне косинусное перетворення може бути обчислено через одномірне перетворення Хартлі. При цьому спочатку проводиться перестановка елементів вектора вихідних даних таким чином, що першу половину послідовності складають непарні елементи, а другу половину послідовності – парні елементи в порядку зростання номерів, а потім виконується одномірне перетворення Хартлі над модифікованим вектором[6]:

$$H_k = \sum \hat{X_m} cas\left[\pi\left(\frac{2\pi km}{N}\right)\right] \qquad (1.17)$$

(де *cas [...] = cos [....] + sin [...]),* яке може бути обчислено по швидкому алгоритму; після чого виконуються додаткові обчислення з елементами вектора результатів перетворення Хартлі:

$$\hat{c_k} = \left\{H_k \left[\cos\left(\frac{\pi k}{2N}\right) - \sin\left(\frac{\pi k}{2N}\right)\right] + H_{N-k}\left[\cos\left(\frac{\pi k}{2N}\right) + \sin\left(\frac{\pi k}{2N}\right]\right\}\right. \quad (1.18)$$

де $k = [1, ..., N-1]$, причому $\hat{c}_0 = H_0$.

Процедура зворотного косинусного перетворення відрізняється від прямого алгоритму іншою послідовністю обчислень – спочатку виконуються перетворення, подібні виразу (1.18), після чого виконується перетворення Хартлі і лише потім проводиться перестановка елементів вектора результатів.

До незручностей косинусного перетворення слід віднести:

- неподільність ядра перетворення для двовимірного варіанта;
- негнучкі алгоритми для різного розміру ядра;
- несиметричний алгоритм зворотного косинусного перетворення

В результаті виконання ДКП формується 64 частотних компоненти фрагмента. В результаті вихідний фрагмент представлений в області просторових частот. Цей крок ще не призводить до стиснення зображення. Однак, при його виконанні вважається, що в переважній більшості зображень близькі за своїм координатам пікселі мають і близькі значення. Тому, при переході від фрагмента до його частотному поданням велика частина енергії сигналу зосереджується в області низьких частот, тобто компоненти з меншим значенням індексу k у виразі (1.12) мають великі значення.

При виконанні цієї операції 64 вихідних пікселів перетворюються в матрицю з 64 коефіцієнтів, які характеризують "енергію" вихідних пікселів. Найважливішою особливістю цієї матриці коефіцієнтів є те, що перший коефіцієнт передає переважну частину "енергії", а кількість "енергії", переданої іншими коефіцієнтами, дуже швидко убуває. Тобто, більша частина інформації вихідної матриці 8x8 пікселів представляється першим елементом матриці, перетвореної за способом ДКП.

На цьому етапі відбувається деяка втрата інформації, пов'язана з принциповою неможливістю точного зворотного перетворення (на етапі

відновлення зображення). Однак ця втрата інформації досить незначна в порівнянні з втратами на наступному етапі.

Перетворена матриця з 64 пікселів потім проходить операцію квантування, яка застосовується для скорочення розрядності коефіцієнтів. Процес квантування, який веде до стиснення коефіцієнтів ДКП, виражається наступним чином:

$$z_{kl} = round\left(\frac{y_{kl}}{q_{kl}}\right) = \left\lfloor (y_{kl} \pm \left\lfloor \frac{q_{kl}}{2} \right\rfloor)/q_{kl} \right\rfloor, \quad k,l = 0,1,\dots,7 \qquad (1.19)$$

де q_{kl} - ваговий множник матриці квантування Q розміру 8x8 з номером kl позначає найбільше ціле менше або рівне x.

Іноді для формування матриці квантування може використовуватися спеціальна вагова функція, що дозволяє сформувати коефіцієнти квантування, що звертають в 0 найбільше число високо і середньо частотних коефіцієнтів. Елементи матриці Q лінійно зростають пропорційно сумі індексів елемента матриці, наприклад на рис.1.4. [9]:

В результаті квантування відбулося обнулення багатьох коефіцієнтів y_{kl}. Вибір матриці Q визначається необхідним коефіцієнтом стиснення.

Саме тут відбувається найбільша значна втрата інформації – відкидаються малі зміни коефіцієнтів. Тому в процесі відновлення зображення після операції зворотного ДКП виходять вже інші параметри пікселів. Квантування також забезпечує можливість подальшого ефективного стиснення даних за допомогою будь-якого способу стиснення без втрат.

Оскільки для *U*- і *V*- компонентів квантування може бути більш грубим, ніж для Y- компонента, на останньому етапі процесу стиснення *U*- і *V*- компонентів відбувається більшою мірою.

Після квантування компоненти спектра всіх оброблених фрагментів «витягуються» в послідовність чисел за допомогою алгоритму діагонального сканування. В основі такого сканування лежить прийом, що дозволяє досягти

більшого ущільнення і грунтується також на характерному вигляді спектра зображень реальних сцен.

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Рис.1.4. Приклад матриці квантування

Статистично доведено, що для реальних багаторівневих зображень двовимірний квантований спектр являє собою матрицю трикутного виду. Більшість значень знизу і праворуч – нулі. Побудова елементів матриці в ланцюжок проводиться так, як це показано на рис.1.5. При цьому в послідовність включаються тільки елементи від першого до останнього ненульового. Після нього в послідовність включається спеціальний стоп - код. Це дозволяє виключити з послідовності зустрічаються нулі.

Далі зазвичай застосовується метод однопрохідного кодування Хаффмана. Спочатку аналізується вся послідовність символів. Часто повторюваним серіям бітів присвоюються короткі позначення (маркери). Різниця розмірів маркерів і представляємих ними бітових серій і визначає досягаємий ступінь стиснення

Відновлення. При відновленні зображень перераховані вище кроки виконуються в зворотному порядку. Декодування починається з відновлення з отриманого бітового потоку закодованих нерівномірним кодом довжин серій нулів і значущих елементів матриці *Z*. Відновлення коефіцієнтів розкладання $y^{\wedge}_{kl}$ по квантованим значенням *Z* виконується за формулою:

$$\hat{y_{kl}} = z_{kl} q_{kl} \qquad (1.20)$$

Далі виконується зворотне ДКП:

$$f(x,y) = \frac{1}{4}\sum_{x=0}^{1}\sum_{y=0}^{1} c(u,v)F(u,v)cos\left(\frac{2x+1}{16}ux\right)cos\left(\frac{2y+1}{16}vx\right) \quad (1.21)$$

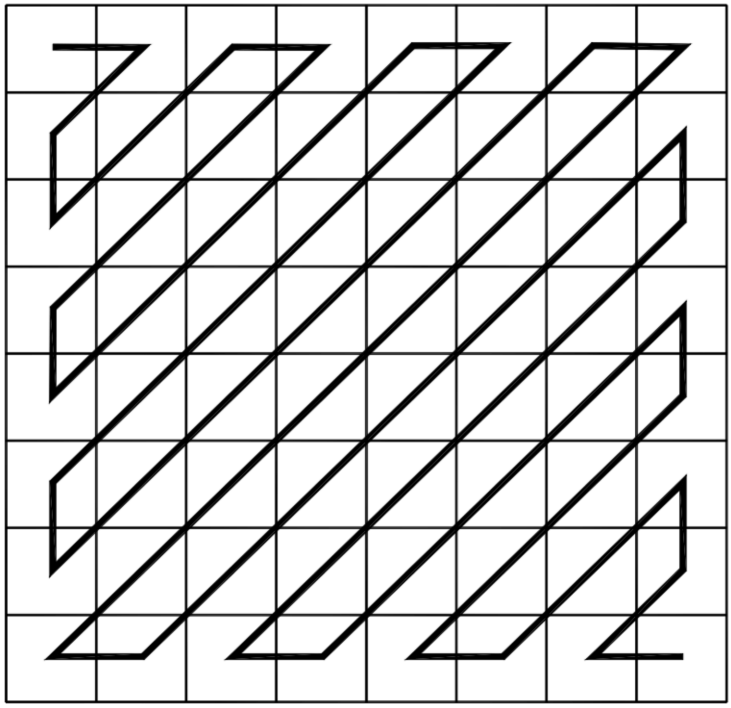

Рис.1.5. Діагональне “зіг-заг” сканування спектральних компонент

Після цього колірний простір даних зображення можна перетворити в початковий вигляд. Втрати при зворотному ДКП також не великі в порівнянні з втратами квантування.

Коефіцієнт архівації в JPEG може змінятся в межах від 2 до 200 разів (на практиці коефіцієнт стиснення не перевищує 20-25)[11].

Як і будь-якого іншого алгоритму з втратами, у JPEG є свої особливості. Найбільш відомі "ефект Гіббса" і дроблення зображення на квадрати 8x8. Перший виявляється близько різких меж предметів, утворюючи своєрідний "ореол". Розбиття на квадрати відбувається, коли задається занадто великий відсоток архівації для даного конкретного зображення. Недоліком методу JPEG є також те, що нерідко горизонтальні і вертикальні смуги на дисплеї абсолютно

не видно, і можуть проявиться тільки при друку у вигляді муарового візерунка. Він виникає при накладенні похилого растру друку на смуги зображення. З цієї причини JPEG не рекомендується використовувати в поліграфії при високих коефіцієнтах стиснення.

1.4.3. Стиснення за методом WIC (Wavelet Image Compression)

Припустимо, що ми маємо послідовність, що складається з 2^n точок $[x_1, x_2, \dots, x_{2^n}]$ для деякого цілого n > 0. Ми можемо порівняти цю послідовність з наступною функцією з векторного простору функцій V^n

$$f(t) = x_1\phi_{n,0} + \cdots + x_1\phi_{n,2^n-1}(t) \tag{1.22}$$

Першим кроком обчислень вейвлет-перетворення послідовності $[x_1, x_2, \dots, x_{2^n}]$ буде розклад $f(t)$ по альтернативному базису простору V^n, половину якого складають вейвлети:

$$f(t) = a_{n-1,0}\phi_{n-1,0} + \cdots + a_{n-1,2^{n-1}-1}\phi_{n-1,2^{n-1}-1}(t) + d_{n-1,0}\psi_{n-1,0} + \cdots + d_{n-1,2^{n-1}-1}\psi_{n-1,2^{n-1}-1}(t) \tag{1.23}$$

Коефіцієнти $d_{n-1,0}, \dots, d_{n-1,2^{n-1}-1}$ при базисних вейвлет-функціях складають половину коефіцієнтів вейвлет-перетворення, тому їх значення зберігаються. Наступним кроком процесу перетворення є застосування такого ж базисного перетворення до інших членів рівняння:

$$g_{n-1}(t) = a_{n-1,0}\phi_{n-1,0} + \cdots + a_{n-1,2^{n-1}-1}\phi_{n-1,2^{n-1}-1}(t) \tag{1.24}$$

Таким чином g_{n-1} – це елемент V^{n-1} и тому може розкладатись по альтернативному базису, який складається з масштабуючи функцій $\phi_{n-2,j}$ та

вейвлетів $\psi_{n-2,j}$.

Для отримання коефіцієнтів рівності (1.23) з коефіцієнтів рівності (1.22) використовується ортогональність. Кожна $\phi_{n-1,j}$ ортогональна кожній $\phi_{n-1,k}$ так же як і всім $\psi_{n-1,j}$ и аналогічно кожен вейвлет $\psi_{n-1,j}$ ортогональний іншим вейвлетам $\psi_{n-1,k}$ і всім масштабуючим функціям $\phi_{n-1,j}$. Також, кожна $\phi_{n-1,j}$ и кожний $\psi_{n-1,j}$ є нормованими. Отже, з викладеного вище отримаємо:

$$\int_0^1 f(t)\phi_{n-1,j}(t)dt = a_{n-1,j} \tag{1.25}$$

З огляду на ортогональність в правій частині залишається тільки один член, а нормування призводить до відсутності коефіцієнта при $a_{n-1,j}$. Тепер, підставив праву частину рівності (1.22) замість $f(t)$ у (1.25) і прийнявши $j = 0$, отримаємо:

$$\int_0^{1/2^n} x_1\sqrt{2^n}\sqrt{2^{n-1}}dt + \int_{1/2^n}^{2/2^n} x_2\sqrt{2^n}\sqrt{2^{n-1}}dt = (x_1 + x_2)\left(\frac{1}{\sqrt{2}}\right)2^n\left(\frac{1}{2^n}\right) = \frac{x_1+x_2}{\sqrt{2}} \tag{1.26}$$

Комбінуючи (1.25) і (1.26) при $j = 0,$ отримаємо:

$$a_{n-1,0} = \frac{x_1+x_2}{\sqrt{2}} \tag{1.27}$$

Квадратний корінь в коефіцієнті з’являється за рахунок нормування. У випадку використання ненормованих базисних функцій результатом буде двоточкове середнє значення. Інші коефіцієнти $a_{n-1,j}$ $j = 1, \dots, 2^n - 1$ обчислюються аналогічно. Таким чином:

$$a_{n-1,j} = \frac{x_{2j+1}+x_{2j+2}}{\sqrt{2}},\ \ j = 1, \dots, 2^{n-1} - 1 \tag{1.28}$$

Аналогічно, використовуючи властивості ортогональності и нормованості функцій $\psi_{n-1,j}$ можна обчислити коефіцієнти $d_{n-1,j}$ за наступною формулою:

$$d_{n-1,j} = \frac{x_{2j+1} - x_{2j+2}}{\sqrt{2}}, \ \ j = 1, \dots, 2^{n-1} - 1 \tag{1.29}$$

Рівняння (1.28) і (1.29) можна представити у вигляді одного матричного рівняння:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & . & . & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & . & . & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ . \\ . \\ . \\ . \\ . \\ . \\ x_{2^n} \end{bmatrix} = \begin{bmatrix} a_{n-1,0} \\ . \\ . \\ a_{n-1,2^{n-1}-1} \\ d_{n-1,0} \\ . \\ . \\ d_{n-1,2^{n-1}} \end{bmatrix} \tag{1.30}$$

Це квадратна матриця с 2^n строками и 2^n столбцями. Визначимо

$$A_{\mathrm{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & . & . & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{1.31}$$

і

$$D_n \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & . & . & . & . & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{1.32}$$

де A_n и D_n - матриці розмром $2^{n-1}\times 2^n$, и A_n можна розглядати як оператор усереднення и D_n- оператор різниці. Введемо також векторний запис

$$x = \begin{bmatrix} x_1 \\ . \\ . \\ x_{2^n} \end{bmatrix}, a_{n-1} = \begin{bmatrix} a_{n-1,0} \\ . \\ . \\ a_{n-1,2^n-1} \end{bmatrix}, d_{n-1} = \begin{bmatrix} d_{n-1,0} \\ . \\ . \\ d_{n-1,2^n-1} \end{bmatrix} \quad (1.33)$$

де x - це вектор-стовбець с 2^n елементами, а a_{n-1} і d_{n-1} - вектори-стовбці з 2^{n-1} елементами. Тоді можна представити це як вираз

$$\begin{bmatrix} A_n \\ D_n \end{bmatrix} x = \begin{bmatrix} a_{n-1} \\ d_{n-1} \end{bmatrix} \quad (1.34)$$

Матриця в лівій частині (1.34) – це єдина матриця $2^n\times 2^n$, а вектор в правій частині - єдиний вектор-стовбець $2^n\times 1$. На кожному кроці процесу вейвлет-перетворення зберігаються деталізуючі коефіцієнти и обробляються коефіцієнти усереднення. У випадку, що розглядається, вейвлет-перетворення буде мати 2^n компонентів. Половину з них ми отримуємо з рівняння (1.34) у якості деталізуючих коефіцієнтів в d_{n-1}. Зберігаємо ці коефіцієнти як половину вейвлет-перетворення.[3] Наступний крок вейвлет-перетворення полягає в застосування до a_{n-1} операцій усереднення и віднімання на наступному, більш низькому, рівні роздільної здастності:

$$\begin{bmatrix} A_{n-1} \\ D_{n-1} \end{bmatrix} a_{n-1} = \begin{bmatrix} a_{n-2} \\ d_{n-2} \end{bmatrix} \quad (1.35)$$

Тут A_{n-1} і D_{n-1} - це $2^{n-1}\times 2^n$ матриці, а a_{n-2} и d_{n-2} - это вектори-стовбці розмірності 2^{n-2}. Щоб побудувати частину вейвлет-перетворення, ми збережемо d_{n-2} разом з d_{n-1}. Далі процес полягає в подальшому застосуванні

операцій усереднення і віднімання до a_k зберігаючи отримані деталізуючі коефіцієнти як частину вейвлет-перетворення. На останньому кроці зберігається середнє значення a_0, яке є однокомпонентним вектором с єдиним елементом $a_{0,0}$. Результуюче вейвлет-перетворення, яке можна представити як єдиний вектор-стовбець з $1 + 1 + 2 + \cdots + 2^{n-1} = 2^n$ елементами, буде мати вигляд:

$$\begin{bmatrix} a_0 \\ d_0 \\ d_1 \\ . \\ . \\ . \\ . \\ d_{n-1} \end{bmatrix} \qquad (1.36)$$

Вейвлет-стиснення – це стиснення з використанням сплесків. На відміну від Фур'є аналізу та ДКП сплески визначені лише на частині області задання аргументу і можуть розглядатися як репліки єдиної базової функції, що відрізняються за масштабом та розміщенням. Така базова функція називається материнською.

Найпростіша ідея алгоритму полягає в тому, щоб зберігати в файл різницю - число між середніми значеннями сусідніх блоків у зображенні, які зазвичай приймають значення близькі до 0.

Так два числа a_{2i} і a_{2i+1} можна представити у вигляді: $b^1_i = (a_{2i} + a_{2i+1}) / 2$ і $b^2_i = (a_{2i} - a_{2i+1}) / 2$, аналогічно послідовність a_i може бути попарно переведена в послідовність $b^{1,2}_i$. Дане перетворення можна застосовувати послідовно кілька разів до результатів попереднього перетворення.

Для двомірного випадку (квадрата з 4-х точок з яскравостями $a_{2i,2j}$, $a_{2i,2j+1}$, $a_{2i+1,2j}$, $a_{2i+1,2j+1}$) найпростішим перетворенням може бути перетворення такого вигляду:

$$
\begin{aligned}
b_{i,i}^{1} &= (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1})/4 \\
b_{i,i}^{2} &= (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1})/4 \\
b_{i,i}^{3} &= (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1})/4 \\
b_{i,i}^{4} &= (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1})/4
\end{aligned}
\tag{1.37}
$$

Таким чином, використовуючи дане перетворення, вихідне зображення перетвориться в 4 субзображення, до кожного з яких, в принципі, дане перетворення можна застосувати повторно.

Отже, основна ідея вейвлетного перетворення сигналу полягає в ієрархічному розкладанні вхідного сигналу на послідовності так званих базових компонент з розрідженням, яке послідовно зменшується, і пов'язаних з ними компонент деталей. На кожному рівні розкладу базова компонента і компонента деталей містять інформацію, необхідну для відновлення базового сигналу на наступному рівні.

При виконанні двовимірного вейвлет-аналізу з використанням перетворення Хаара (вираз 1.26) для розкладу зображення спочатку виконується розкладання по рядках, а потім за стовпцями. Результатом розкладання є 4 матриці HH0, HL0, LH0, LL0, відповідні фільтрації фільтром *h1(n)* по рядках і стовпцях, фільтром *h1(n)* по рядках і фільтром *h0(n)* за стовпцями, фільтром *h0(n)* по рядках і фільтром *h1(n)* за стовпцями, і фільтром *h0(n)* по рядках і стовпцях. Далі низькочастотна матриця LL0 піддається вейвлет розкладанню. Його результатом є матриці HH1, HL1, LH1, LL1. Таке розкладання повторюється *r* раз, як показано на рис.1.6. Результатом розкладання є набір з *3r + 1* матриць зменшуваною розмірності. Кожна матриця піддається скалярному або векторному квантуванню і подальшому кодуванню. Вибір числа рівнів квантування або кроку квантування проводиться виходячи з необхідного стиснення і відповідного розподілу бітів між матрицями.

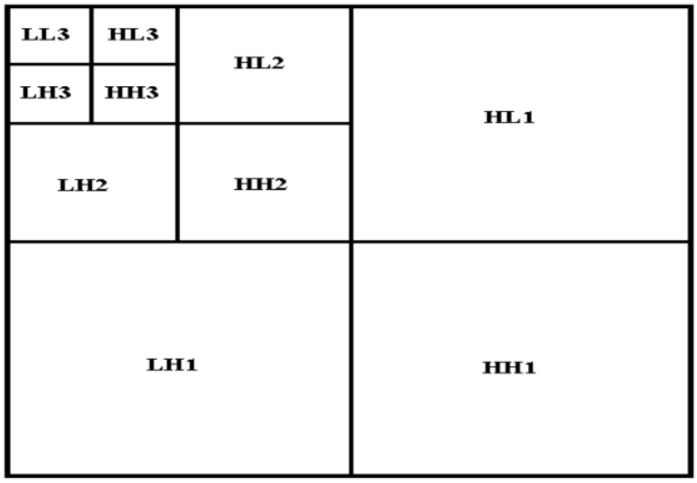

Рис.1.6. Трьохрівневе вейвлет претворення зображення

Високочастотні матриці (матриці деталей), як правило, після квантування містять велику кількість нульових елементів. Одним із широко використовуваних методів кодування матриць деталей є кодування довжин серій нулів і наступних за ними значень кодом Хаффмана. Низькочастотні матриці безпосередньо кодуються кодом Хаффмана.

Сьогодні у фахівців не викликає сумніву, що ці алгоритми кодування мають ряд переваг в порівнянні з алгоритмами, побудованими на основі дискретного косинусного перетворення Фур'є, яке використовується в кодеках JPEG, MPEG-1, MPEG-2[11].

1.4.4. Фрактальне стиснення зображень

Багато в чому недоліки в якості відновленого зображення по стислому форматі згідно JPEG пов'язано з тим, що при стисненні ніяк не враховуються специфіка зображення, тобто не виявляється його структура, характерні ділянки і т.д. Саме врахування специфіки зображення лежить в основі фрактального методу, який запропонував в 1988 році Мандельброт. Згідно Мандельброту, фрактал – це структура, виділена при аналізі зображення, і володіє схожою формою незалежно від її розмірів. Наприклад, у зображенні крони дерева, фрактал – зображення аркуша. Тому зображення можна як би збирати з фракталів, тобто зображення в термінах фрактального підходу є суперпозиція

фракталів. У свою чергу, окремий фрактал може бути описаний деяким стандартним чином.

Основу фрактального підходу становить постулат про те, що зображення реального світу мають афінну надмірність[7]. Інакше кажучи, існує набір афінних коефіцієнтів, що описують обертання, стиснення, розширення, спотворення форми, зрушення об'єктів зображення.

На початку 80-х років Майкл Барнслі висунув ідею отримання заздалегідь заданого зображення як аттрактора хаотичного процесу. Барнслі намагався відповісти на запитання: чи можливо для даного зображення побудувати хаотичну систему, яка бути для нього дивним аттрактором. Він використовував систему ітерованих функцій (Iterated Function System – IFS).

Найбільш поширеним прикладом фрактальної зображення, згенерованого за допомогою IFS є зображення папороті (рис.1.7.), використане для створення даного зображення, складається з 4-х афінних перетворень. Кожне перетворення кодується ліченими байтами, хоча вихідне зображення може бути будь-якого розміру. Таким чином, можна зробити висновок, що фрактальна компресія – це пошук самоподібних областей та визначення для них параметрів афінних перетворень.

Проста IFS, що породила зображення папороті не придатна для стиснення довільних зображень, оскільки вони зазвичай подаються у градаціях сірого, а не з значень 0 або 1 і тому прості IFS використовуються тільки для самоподібних зображень, тобто коли зображення будуються з елементів, що є копією цілого зображення.

Рис.1.7. Зображення папороті, що згенероване за допомогою IFS

В основі фрактального стиснення лежать кілька основних визначень і теорем. Розглянемо їх коротко.

Перетворення. Перетворення зіставляє точці в одному просторі точку в іншому просторі, можливо в тому ж самому просторі. Перетворення називається відображенням f і записується як: $f: X_1 \to X_2$ якщо воно переводить простір X_1 в простір X_2.

Перетворення в метричному просторі (X, d) називається стискуючим відображенням, якщо існує константа s, $0 \le s < 1$ така що:

$$d(f(x_1), f(x_2)) \le s \cdot d(x_1, x_2) \qquad (1.38)$$

де $d(x_1, x_2)$ - відстань від точки x_1 до точки x_2 в просторі X.

Константа s називається коефіцієнтом стиснення відображення f.

На рис.1.8. показаний приклад стиснення відображення (R_2,d_2), застосованого до безлічі точок в R_2. Тут дане перетворення застосовувалося більше одного разу: спочатку $f(x)$ обчислювалося для точки x, а потім перетворення f застосовувалося до результату перетворення: $f(f(x))$. Таке послідовне, багаторазове застосування перетворення називають ітераціями і позначають як: $f^{\circ n}$, тобто $f((... f(x) ...))$, де f застосовується n раз.

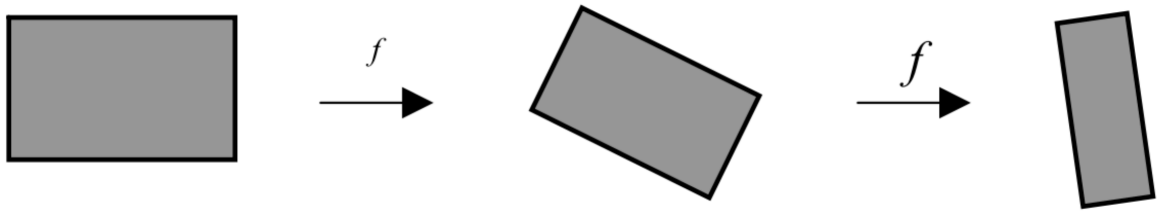

Рис.1.8. Стискуюче відображення для точок у просторі R^2

Система ітеруючих функцій (IFS) складається з повного метричного простору (X, d) і кінцевого безлічі стискають відображень w_n: $X_1 \rightarrow X_2$ з коефіцієнтами стиснення s_n. Таким чином, IFS можна позначити таким чином: $\{X, w_n: n = 1,2, .., N\}$, або якщо розглядати простір точок в R_2, то просто $\{w_n\}$.

У теорії фрактального стиснення доводиться теорема колажу, яку можна сформулювати наступним чином.

Нехай є двійкове зображення $L \subset R_2$ і нехай є стискають відображення, такі що: $U^N_n = {}_1w_n\ (L)$

Покривають L майже точно. Можна вважати таке $w_n\ (L)$ зменшеною копією L. Тоді теорема колажу стверджує, що аттрактор A (аттрактором IFS називається зображення, яке є єдиною нерухомою точкою IFS) системи $\{w_n\}$ близький до L. «колаж» є набір областей $w_n\ (L)$.

Так як аттрактор A – це результат нескінченного числа ітерацій IFS, то він є за своєю суттю фракталом.

Щоб на практиці застосувати теорему колажу (рис.1.9), необхідно вибрати перетворення, які будуть стискаючими відображеннями. Одним з таких перетворень є аффінне перетворення: $T: R^2 \rightarrow R^2$:

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \ddot{x} \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \qquad (1.39)$$

де $a, b, c, d, e, f \in R$. Афінні перетворення можуть здійснювати поворот,переміщення й масштабування (рис.1.10.).

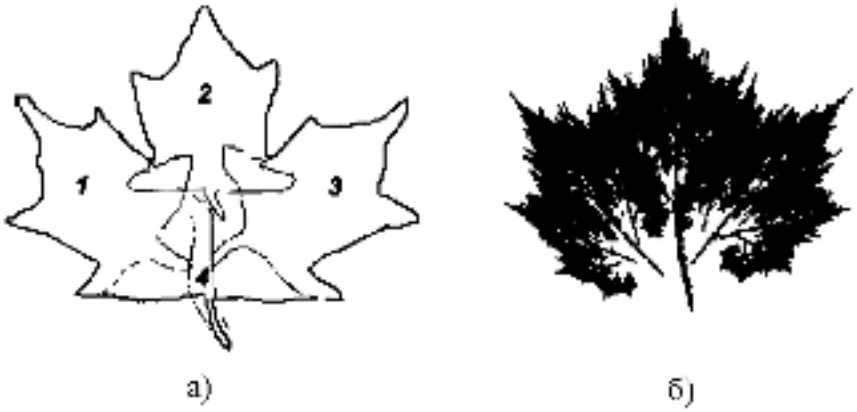

Рис.1.9. Ілюстрація теореми колажу.
(а) вихідне зображення і чотири фрагменти зображення; (б) зображення-аттрактор

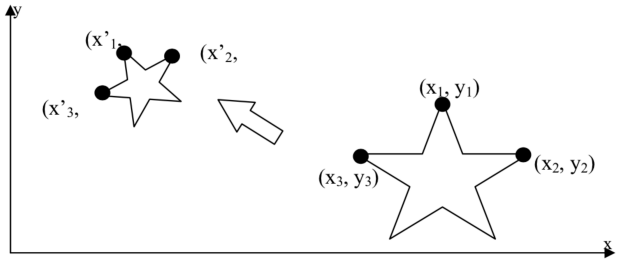

Рис.1.10 Дія афінного перетворення на точки в просторі R^2

Щоб створити IFS зображення – потрібно знайти хоча б деяку приблизну самоподібність в зображенні. Потім потрібно виділити точки і перетворення IFS для кожної з фігур подібності на зображенні. Коли точки і перетворення для IFS визначені, то обчислюються коефіцієнти афінного перетворення, використовуючи систему рівнянь, засновану на виразі (1.39).

Існує два алгоритми побудови зображення-аттрактора за допомогою IFS. Один з них це пряме застосування теореми про стискають відображеннях, а інший - застосування так званої «гри хаосу». Детермінований алгоритм для

побудови зображення є аттрактором IFS, до будь-якого початкового зображенню B застосовує теорему про стискають відображеннях. Алгоритм будує послідовність зображень An, багаторазово застосовуючи IFS відображення *W = {w1, w2, ..., wn}*:

$$A_n = W^{on}(B) \tag{1.40}$$

Детермінований алгоритм корисний з точки зору навчання, оскільки дозволяє бачити результат перетворення на кожному кроці ітерації (рис1.11.). Імовірнісний алгоритм пов'язує з кожним аффінним перетворенням w_i в IFS ймовірність p_i. Ці ймовірності визначають, наскільки щільно кожна частина зображення-аттрактора покрита точками. Імовірнісний алгоритм створює зображення-атрактори високої якості швидше, ніж детермінований алгоритм. Це відбувається не тільки через те, що імовірнісний алгоритм на кожному кроці ітерації виконує меншу роботу, а й сама виконана робота дає кращий результат.

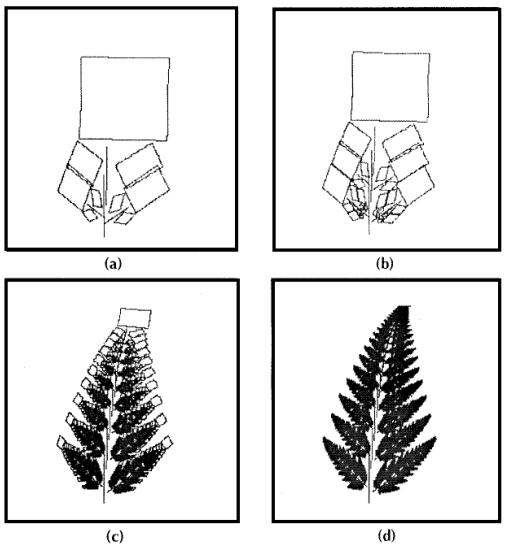

Рис.1.11. Детермінований алгоритм, застосований до IFS папороті. Вид зображення A_n після: a) 2-х, b) 3-х, c) 10-і, d) 30-й ітерацій.

Реальні зображення не володіють властивістю глобальної самоподібності, яка притаманна зображенням IFS. До того ж реальні зображення представлені в

градаціях сірого, а не в двійкових значеннях, як зображення IFS. Тому при фрактальному кодуванні реальних зображень намагаються знайти безліч стискають перетворень, які відображають доменні блоки (які можуть перекриватися) в безліч рангових блоків, які покривають зображення. Рангові блоки можуть бути однакового розміру, але частіше використовується адаптивне розбиття з перемінним розміром блоку (одним з поширених методів адаптивного розбиття є метод квадродерева).

Базовий алгоритм фрактального кодування зображення виконується таким чином (рис 1.12):

Рис.1.12. Відображення доменних блоків у рангові при фрактальному стисненні зображення

1. Зображення f розбивається на не перекриваються рангові блоки {Ri}. У самому простому випадку блоки можуть являти собою прямокутники, але можуть бути й інші форми.

2. Зображення покривається послідовністю доменних блоків, які можуть перетинатися. Домени можуть бути різного розміру, і їх кількість може обчислюватися сотнями і тисячами.

3. Для кожного рангового блоку знаходять домен і відповідне перетворення, яке найкращим чином покриває ранговий блок. Зазвичай це Афінний перетворення.

4. Якщо достатньо точної відповідності не вийшло, то розбиваємо рангові блоки на менші рангові блоки. Даний процес продовжується до тих пір поки,

не отримують прийнятного відповідності або розмір рангового блоку досягає певних значень.

В даний час відомі наступні різновиди такого алгоритму [9].

Алгоритм Фішера. Ідея алгоритму полягає в тому, щоб певним чином класифікувати *D*-блоки і *R*-блоки, а пошук близького *D*-блоку виробляти в тому ж класі, до якого відноситься ранговая область. Робиться це в такий спосіб [9].

Вихідні блоки розбиваються на чотири частини. Для кожної з частин підраховується середнє значення A_i і дисперсія V_i пікселів. Далі блоки класифікуються за наступним принципом. Визначимо три базових типи блоків:

$$\begin{aligned} &\text{тип 1: } A_1 \geq {}_2A \geq A \leq A_4, \\ &\text{тип 2: } A_1 \geq A_2 \geq A_4 \geq A_3, \\ &\text{тип 3: } A_1 \geq A_4 \geq A_2 \geq A_3. \end{aligned} \qquad (1.41)$$

Зрозуміло, що будь-який блок за допомогою відповідного афінного перетворення квадрата в квадрат можна привести до виду, відповідному одному з вказаних типів. Після того, як ми зафіксували три основні класи, блоки класифікуються за дисперсії. Таким чином, в кожному з трьох класів з'являються 24 підкласу, разом 72 класу. Пошук близького до *R*-блоку *D*-блоку проводиться перебором у відповідному класі.

Генетичний алгоритм. Генетичний алгоритм (ГА) являє собою алгоритмічний підхід до вирішення екстремальних задач однокритеріального вибору, заснований на моделюванні основних факторів еволюційного розвитку популяції [9].

При використанні ГА для пошуку оптимальних рішень кожен елемент $x \in X$ простору оптимізації повинен бути представлений як вектор $b \in B$ з N символів двійкового алфавіту $A = \{0,1\}$, де $B = A^N$. Необхідно також, щоб простір оптимізації X складався із скінченного числа елементів.

Популяцією $П = (\chi^1, \chi^2, ..., \chi^M)$ чисельності M вважається вектор простору B^M, координати якого називаються генотипами особин даної популяції.

Кроком ГА є перехід від поточного покоління до наступного, тобто отримання нової популяції $П_{t+1}$ з $П_t$. У побудові чергової особини нової популяції беруть участь оператори кросинговеру (схрещування), мутації і випадковий оператор відбору, *Select*: $B^M \rightarrow \{1, ..., M\}$ дія якого полягає у виборі номера особини батька при породженні чергового нащадка.

Для визначення ГА необхідно задати оператор кросинговеру *Cross*: $B \times B \rightarrow B \times B$ і оператор мутації *Mut*: $B \rightarrow B$.

Дія кросинговеру $(\chi', \tau') = Cross\ (\chi, \tau)$ полягає у виборі випадковим чином деякої позиції *j*, рівномірно розподіленої від 1 до *N-1*, після чого результат формується у вигляді

$$\chi' = (\chi_1, \chi_2, ..., \chi_j, \tau_{j+1}, ..., \tau_N),\ \tau' = (\tau_1, \tau_2, ..., \tau_j, \chi_{j+1}, ..., \chi_N). \qquad (1.42)$$

Вплив кросинговеру регулюють за допомогою ймовірності P_{Cross} спрацьовування цього оператора (в іншому випадку все залишається без змін).

Оператор мутації в кожній позиції аргументу із заданою вірогідністю P_{mut} замінює її вміст на випадковий елемент двійкового алфавіту A, обраний відповідно з рівномірним розподілом (в іншому випадку все залишається без змін).

Цільова функція вихідної задачі, замінюється в ГА на функцію придатності генотипу $Ф\ (\chi)$, де $\chi \in B$.

Процес роботи алгоритму є послідовна зміна поколінь, на кожному кроці якої популяція $П_{t+1}$ наповнюється парами нащадків від особин популяції $П_t$ за формулою:

$$(\chi^{t+1}_k, \chi^{t+1}_{k+1}) = Mut\ (Cross\ (\chi^t_{Select(П^t)}, \chi^t_{Select(П^t)})) \qquad (1.43)$$

де (χ^{t+1}_k, χ^{t+1}_{k+1}) – особини з найменшою придатністю популяції П. Тобто індивіди витягуються попарно з $П^t$ і після кросинговеру і мутації поміщаються в $П^{t+1}$. Зміна ймовірностей мутації і кросинговеру дозволяє регулювати роботу ГА і налаштовувати його на конкретні завдання.

Модифікований генетичний алгоритм. Опишемо схему ГА у застосуванні до задачі фрактального стиснення [7]. В якості генотипу ГА зручно взяти вектор, компонентами якого будуть піксельні координати області $D_{j(i)}$ вихідного зображення, визначеного на тороідальній поверхні, і число кодує афінне перетворення W_i. Є вісім способів афінного перетворення квадрата в квадрат: поворот на чотири сторони або дзеркальне відображення і поворот на чотири сторони. Отже, на кодування цього перетворення достатньо трьох біт. Функцію придатності покладемо рівною

$$\Phi = \frac{1}{1+\sum([F(\xi,\eta)-F_i(\xi,\eta)]^2:(\xi,\eta)\in R_i\cap Z^2)} \qquad (1.44)$$

де в нижній частині під знаком суми – евклідова відстань між вихідним і перетвореним блоком. Ця функція задовольняє вимоги ГА (невід'ємна) і адекватна для оператора рулеточної селекції, при якій кожен індивід $\chi^{i,t}$ популяції $П_t$ виявляється батьком при формуванні черговий особини $\chi^{i,t+1}$ популяції $П_{t+1}$ з імовірністю

$$P_{select}(\chi^{i,t}) = \frac{\Phi(\chi^{i,t})}{\sum_j \Phi(\chi^{j,t})} \qquad (1.45)$$

При такому поданні хромосом, що визначають даний генотип, будь-який вектор простору рішень завжди допустимий і має ненульову придатність.

Оператор мутації для даного алгоритму – стандартний, а оператор кросинговеру модифікований таким чином:

- позиція кросинговеру може розташовуватися тільки в місцях стикування двійкового представлення координат.
- виконується тільки для близькоспоріднених особин, тобто, якщо відстань $\rho(\chi, \tau)$ мала, де ρ – деяка метрика.

Збільшення швидкодії алгоритму. Недоліком даного методу стиснення є дуже великий обсяг обчислень для кодування зображень. Для програм реалізують даний метод, застосовуються як звичайні методи збільшення швидкості програм: цілочисельна арифметика, асемблерна реалізація та інші, так і деякі специфічні прийоми: використання афінних перетворень тільки одного типу, різні варіанти розбиття зображення на області R_i.

Наведемо схеми двох алгоритмів, які для деяких класів зображень можуть значно зменшити обсяг обчислень. Параметрами першого алгоритму служать рівень втрат при кодуванні і мінімальний розмір областей R_i. Цей алгоритм забезпечує рівномірну якість кодування всього зображення. Параметром другого алгоритму є кількість областей R_i, використовуваних для кодування зображення, що прямо впливає на обсяг обчислень, але він не забезпечує достатньої точності кодування окремих фрагментів зображення [9].

Алгоритм 1

• Оберемо допустимий рівень втрат при кодуванні e.

• $R_1 = \Omega$ і помітимо його як необроблений фрагмент.

• Поки є необроблений фрагмент R_i виконувати:

1. Знайти $D_{j(i)}$, W_i і F_i, які найкращим чином наближають R_i (на яких досягається мінімум $\sigma^2_{Ri}(W_i, D_{j(i)}, F_i)$).

2. Якщо $\sigma^2_{Ri}(W_i, D_{j(i)}, F_i)) < e$ або розмір $R_i < min$, то позначити R як оброблене. Інакше, розбити R_i на більш дрібні фрагменти і позначити їх як необроблені.

Алгоритм 2

• Оберемо максимальне число N фрагментів R_i.

• Додамо фрагмент $R_1 = \Omega$ в список перетворень і позначимо його як

необроблений.

• Поки є необроблені фрагменти в списку виконувати:

1. Для кожного необробленого фрагмента знайти відповідні W_i, $D_{j\,(i)}$, F_i.

2. Знайти в списку фрагмент R_i найбільшого розміру і з найбільшою метрікою $\sigma^2_{Ri}\ (W_i, D_{j\,(i)}, F_i))$.

3. Якщо число фрагментів у списку менше N, тоді розбити фрагмент R_i на більш дрібні і занести їх в список як необроблені. Викреслити з списку фрагмент R_i.

Алгоритм відновлення зображення:

1.Створити два зображення однакового розміру А і Б. Розмір зображень може не дорівнювати розміру початкового зображення. Початковий рисунок областей А і Б не має значення. Це можуть бути випадкові дані.

2. Дані з області А перетворяться в область Б. Для цього спочатку зображення Б ділиться на домени також як і на першій стадії процесу стиснення (розташування доменів описано в заголовку файлу). Тепер для кожного домена області Б проводиться відповідне афінне перетворення рангових областей зображення А, описане коефіцієнтами з стисненого файлу. Результат поміщається в область Б. На цій стадії створюється зовсім нове зображення.

3. Дані з області Б перетворюються в область А. Цей крок ідентичний попередньому, тільки зображення А і Б помінялися місцями, тобто тепер область А ділиться на блоки і рангові області зображення Б відображаються на ці блоки.

4. Процедури 2 і 3 повторюються до тих пір, поки зображення А і Б не стануть такими, що неможливо відрізнити.

Прямий і зворотний хід (стиснення і відновлення) не еквівалентні за витратами. Пряме перетворення (стиснення) значно довше, зворотне перетворення (відновлення) набагато швидше. Алгоритм фрактального

стиснення — несиметричний алгоритм. Коефіцієнт симетричності (відношення часу архівації до часу розархівації) коливається в межах 1000-10000.

До достоїнств фрактального методу можна віднести:

- Високі коефіцієнти стиснення.
- Високу швидкість зворотного перетворення.
- Можливість подальшого структурного аналізу зображення [4].

При цьому фрактальний метод володіє наступними недоліками:

- Залежністю результатів роботи методу від принципів відбору базових елементів і доменів.
- Коефіцієнт стиснення залежить від повторюваності базових елементів.

Алгоритм орієнтований на повнокольорові зображення і зображення в градаціях сірого кольору. Фрактальне стиснення реалізовано у форматі FIF.

В даний час існує досить багато алгоритмів стиснення зображень. Основою будь-яких методів стиснення даних є використання природної надмірності вихідної інформації. Стиснення зображень здійснюється або в просторової або в частотній областях зображення. Найбільш яскравими прикладами просторового стиснення зображень є алгоритми PCX, GIF, а частотного стиснення – JPEG.

В просторової області не можна домогтися досить великого коефіцієнта стиснення, тому в даний час більшість алгоритмів стиснення працюють в частотній області зображення, оскільки найбільш явно виражена надмірність вихідних даних проявляється при їх спектральному аналізі, як результату розкладання в деякому ортогональному базисі.

У подібних алгоритмах можна виділити три основні кроки:

1. Застосування оборотних дискретних ортогональних перетворень до зображення.
2. Вибір найбільш значущих частотних коефіцієнтів.
3. Вторинне стиск обраних коефіцієнтів, наприклад арифметичним або статистичними алгоритмом стиснення.

1.5. Постановка задачі

Оскільки метою є підвищення швидкості завантаженя сайтів, то недоцільно буде використання алгоритмів стиснення зображень без втрат, натомість краще використовувати алгоритми стиснення з втратами. Серед алгоритмів з втратами найбільш використовуваним в вебі є алгоритм стиснення JPEG. Виходячи з цього логічним буде саме його модифікація.

Вейвлетне перетворення дає досить гарні результати для сприйняття людським оком, а ткож є досить легким в обчисленнях і не перевищує складність обчислень дискретного косинусного перетворення.

Перед модифікацією не обхідно дослідити вплив кількості коефіцієнтів уереднення на якість зображення. На основі результатів модифікувати алгоритм стиснення JPEG.

Для коректної оцінки спотворень зображень необхідно використати критерій якості. Використання двох критеріїв якості дасть можливість краще зрозуміти масштаби спотровень. Зручно використати середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу (1.2), так як воно є найпростішим критерієм та критерій PSNR (1.4), оскільки він найчастіше застосовується на практиці.

Отже задачі, які потрібно вирішити: дослідити вплив кількості коефіцієнтів усереднення вейвлетного перетворення на якість зображення; модифікувати алгоритм стиснення JPEG; реалізувати модифікований алгоритм стиснення; провести експериментальне дослідження результатів роботи модифікованого алгоритму; дослідити вплив зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту.

Висновки до розділу

Методи стиснення зображення поділяються на дві великі групи – методи стиснення без втрат та методи стиснення з втратами.

Методи стиснення без втрат дозволяють отримати відновлене зображення індентичне вихідному. Такі методи стиснення використовують тоді, коли ідентичність вхідного та вихідного забражень є критичною або коли самі шуми зображення є об’єктом дослідження – наприклад, в медицині та науковий дияльності. Методи стиснення без втрат дають в результаті невеликі коефіцієнти стиснення.

Методи стиснення з втратами використовують у випадках, коли зміни зображення, які непомітні людському оку є прийнятними – фото- та відеокамери, особливо веб. Такі зміни відновленого зображення дають змогу досягти значно більших коефіцієнтів стиснення, ніж при стисненні без втрат.

Проаналізовано сучасні методи стиснення зображень та виконано постановку задачі.

РОЗДІЛ 2. МОДИФІКАЦІЯ АЛГОРИТМУ JPEG

2.1. Модифікований метод вейвлетного перетворення

Класичне вейвлет-перетворення, що застосовується у алгоритмах стиснення зображень, працює тільки з одним коефіцієнтом усереднення. Це спричиняє необхідність використовувати додаткові попередні методи для зведення інформації про піксель зображення в системі RGB до одного значення. Також, для зворотного перетворення потрібно зберігати як саме зводилося це число, і за якими формулами його можна повернути [2].

Розглянемо результати вейвлет-перетворень (з 90% втрати інформації) починаючи з 2х коефіцієнтів усереднення (рис.2.1. – 2.6.). Для їх оцінки використаємо критерії наступні якості:

- середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу

$$d(x,y) = \sqrt{\frac{\sum_{x,y}^{M,N}(f(x,y) - \hat{f}(x,y))^2}{M \cdot N}}$$

- критерій якості PSNR

$$d(x,y) = 10\log_{10}\frac{255^2 \cdot M \cdot N}{\sum_{x,y}^{M,N}(f(x,y) - \hat{f}(x,y))^2}$$

Як бачимо зображення з трьома коефіцієнтами мають більш плавні переходи та чіткіші деталі, ніж у зображення з двома коефіцієнтами Те ж саме можна сказати про зображення з чотирма коефіцієнтами. Зображення, що мають більше чотирьох коефіцієнтів усереднення, для людського ока не відрізняються від зображень з чотирма коефіцієнтами усереднення. Також, спираючись на значення кофіцієнтів якості (табл.2.1., рис.2.7. – 2.8.) бачимо, що немає сенсу

брати більше, ніж 4 коефіцієнти усереднення. Отже, для модифікації алгоритму JPEG обрано вейвлет-перетворення з чотирма коефіцієнтами усереднення.

Таким чином коефіцієнти перетворення будуть мати вигляд:

$$a_{i,j} = \frac{x_{8j+i} + x_{8j+(i+4)}}{\sqrt{2}}, \quad j = 1,\ldots,2^{n-1} - 1, i = 0..3 \tag{2.1}$$

$$d_{i,j} = \frac{x_{8j+1} - x_{8j+(i+4)}}{\sqrt{2}}, \quad j = 1,\ldots,2^{n-1} - 1, i - 0..3 \tag{2.2}$$

Можемо спростити метод до одномірного випадку, в результаті чого отримаємо матрицю наступного вигляду:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ d_{0,0} & d_{1.0} & d_{2,0} & d_{3,0} \\ . & . & . & . \\ . & . & . & . \\ d_{0,n-1} & d_{1,n-1} & d_{2,n-1} & d_{3,n-1} \end{bmatrix} \tag{2.3}$$

В класичному вейвлет-перетворенні після отримання коефіцієнту

Рис.2.1. вейвлет-перетворення з 2ма коефіцієнтами усереднення

Рис.2.2. вейвлет-перетворення з 3ма коефіцієнтами усереднення

Рис.2.3. вейвлет-перетворення з 4ма коефіцієнтами усереднення

Рис.2.4. вейвлет-перетворення з 5ма коефіцієнтами усереднення

Рис.2.5. вейвлет-перетворення з 6ма коефіцієнтами усереднення

Рис.2.6. вейвлет-перетворення з 7ма коефіцієнтами усереднення

усереднення та коефіцієнтів різниці, частина других відкидається. Це зменшує обсяг файлу, але збільшує втрати графічної інформації при зворотному перетворенні.

2.2. Модифікований алгоритм JPEG

Спочатку виконується перехід від кольорової схеми RGB до YCbCr наступним чином:

$$\begin{vmatrix} Y \\ U \\ V \end{vmatrix} = \begin{vmatrix} 0.299 & 0.578 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \quad (2.4)$$

Далі виконується модифіковане вейвлетне перетворення з коефіцієнтами наведеними вище (2.1) та (2.2).

Таблиця 2.1. Значення критеріїв якості

	Кількість коефіцієнтів усереднення						
	2	3	4	5	6	7	8
Середньокв. відхилення	1.0051	1.0046	1.0046	1.0046	1.0046	1.0046	1.0046
PSNR	0.0779	0.0739	0.0736	0.736	0.736	0.736	0.736

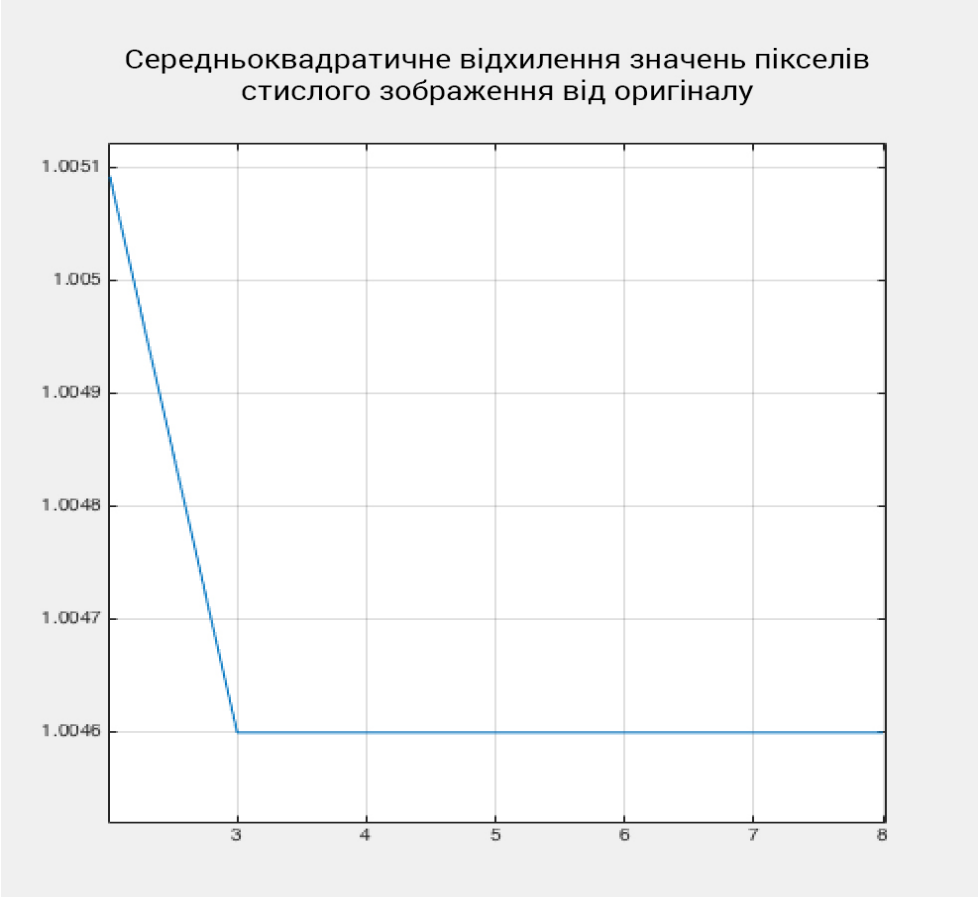

Рис.2.7. Середньоквадратичне відхилення значень пікселів стиснутого зображення від оригіналу

Отримавши матрицю виду (2.3) виконуємо квантування коефіцієнтів. Коефіцієнти діляться на заздалегідь задане число – ступінь стиснення. При збільшенні цього числа знижується динамічний діапазон коефіцієнтів, вони стають ближче до 0, і ми отримуємо великий ступінь стиснення.

Після проведення квантування проводиться “зіг-заг” сканування та застосовується метод однопрохідного кодування Хаффмана.

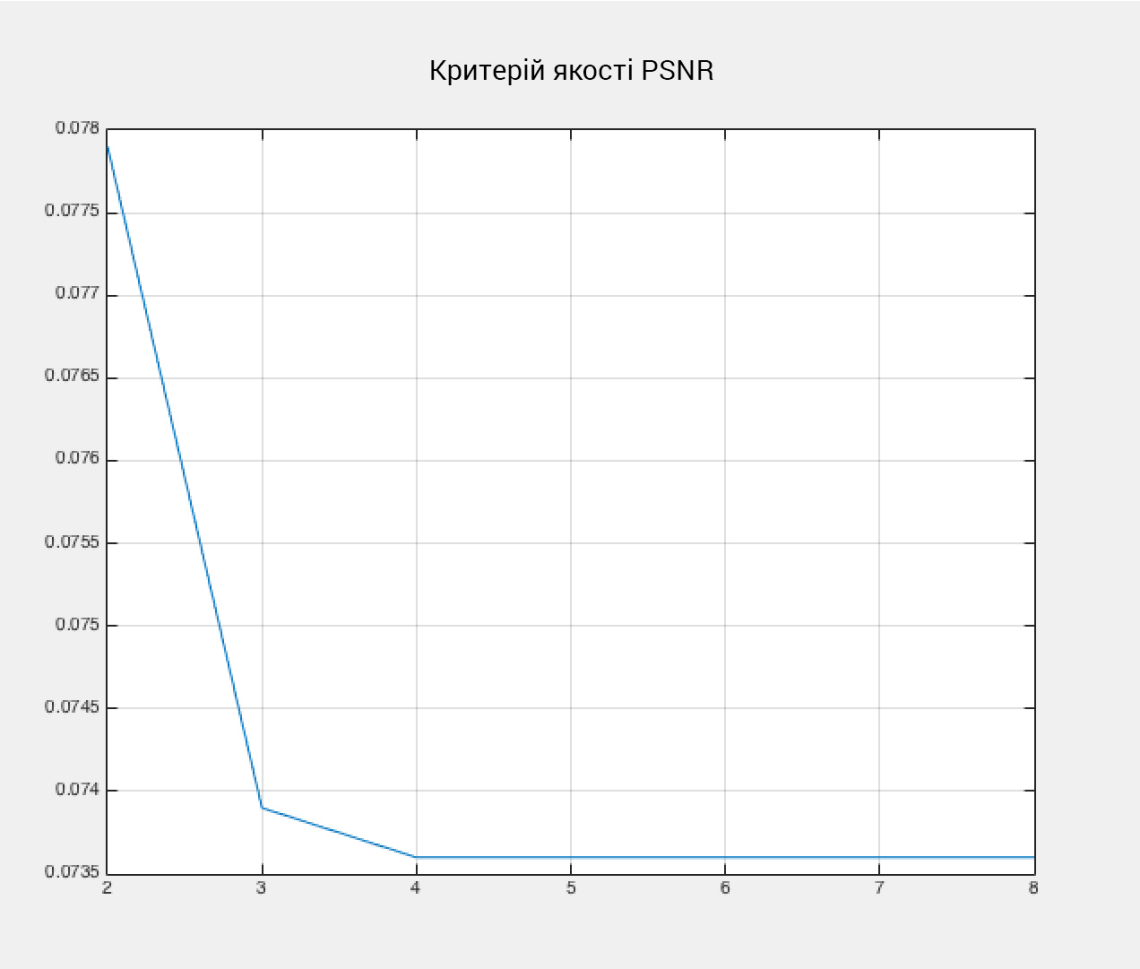

Рис.2.8. Критерій якості PSNR

Висновки до розділу

Проведено експериментальне дослідження випливу кількості коефіцієнтів усеренення на якість зображення при вейвлет-перетворенні. За результатами

дослідження обрано 4 коефіцієнти. На основі цього запропонована модифікація алгоритму стиснення зображень JPEG.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ МОДИФІКАЦІЇ АЛГОРИТМУ JPEG

3.1. Аналіз результатів роботи модифікованого алгоритму

Алгоритм було реалізовано за допомогою програмного забезпечення “MATLAB”.

Порівняємо різні ступені стиснення зображень за допомогою алгоритму JPEG та модифікованого JPEG. На рис.3.1. – 3.2. наведено зображення з 50% коефіцієнтом стиснення.

Як бачимо, зображення стиснуте за допомогою модифікованого алгоритму мало відрізняється від вихідного зображення, в той час як на зображенні стиснутому за допомогою оригінального алгоритму добре видно спотворення. Критерії якості наведені у табл.3.2. та їх іллюстрація на рис.3.9. – 3.10.

Рис.3.1. Вихідне зображення

Розглянемо результати більшого ступеню стиснення (рис.3.4. – 3.5.).

При ступеню стиснення у 80% оригінальний алгоритм дає у результаті значні спотворення зображення. Модифікований алгоритм також дає спотровення, але набагато менші.

Рис.3.2. зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 50%

Рис.3.3. зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 50%

При стисненні в 90% (рис.3.6 – 3.7) модифікований агоритм також дає значні спотворення зображення, але значно прийнятніші для ока, ніж у результаті стисненя оригінальним алгоритмом. Також модифікований алгоритм дає можливість позбутися розбиття зображення на квадрати.

Рогзгянемо як алгоритми впливають на розміри файлів табл.3.1 та рис.3.8. Як бачимо, модифікований алгоритм дає змогу отримати менший розмір файлу, ніж оригінальний алгоритм.

Рис.3.4. зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 80%

Рис.3.5. зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 80%

Рис.3.6. зображення стиснуте за допомогою алгоритму JPEG з коефіцієнтом стиснення 90%

Рис.3.7. Зображення стиснуте за допомогою модифікованого алгоритму JPEG з коефіцієнтом стиснення 90%

Таблиця 3.1. Порівняння розмірів файлів

	Алгоритм стиснення	
Ступінь стиснення	*JPEG*	*Мод. JPG*
50%	76 KB	66 KB
80%	68 KB	60 KB
90%	57 KB	55 KB

Початковий розмір файлу складає 84 KB.

Таблиця 3.2. Значення критеріїв якості

		Ступінь стиснення								
		10%	20%	30%	40%	50%	60%	70%	80%	90%
JPEG	Середньокв. відхилення	1.0111	1.0361	1.0949	1.1776	1.3151	1.5413	2.0153	3.5573	33.0291
	PSNR	0.1302	0.3417	0.8212	1.4537	2.4128	3.7918	6.1209	11.0564	30.4119
Мод. JPEG	Середньокв. відхилення	2.5175	7.2188	934.9723	502.9834	1823.3	8434	5525.3	37415	16788000
	PSNR	8.0535	17.2033	59.45	54.0651	65.2509	78.5546	94.8811	91.495	164.5337

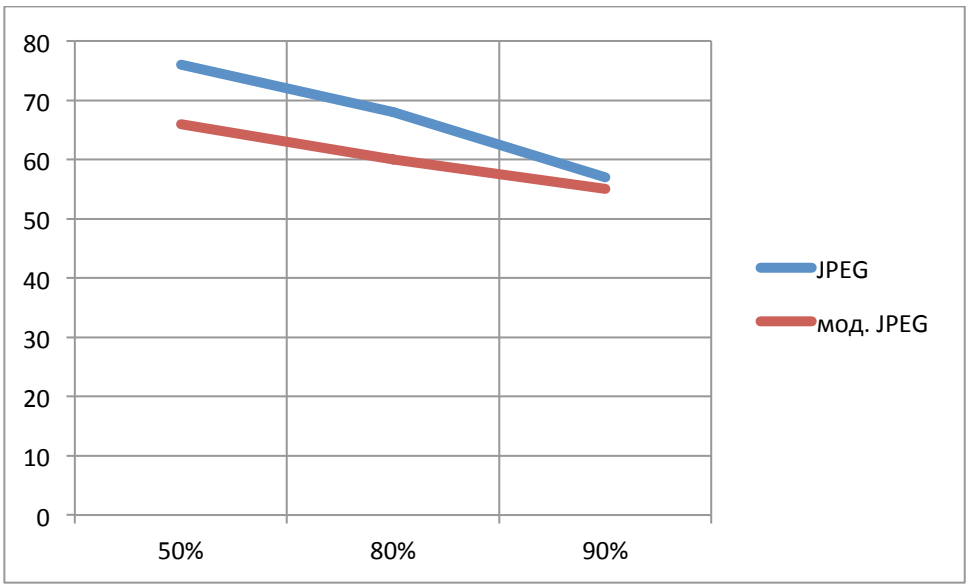

Рис.3.8. Залежність розміру файлу зображенння від коефіцієнта стиснення

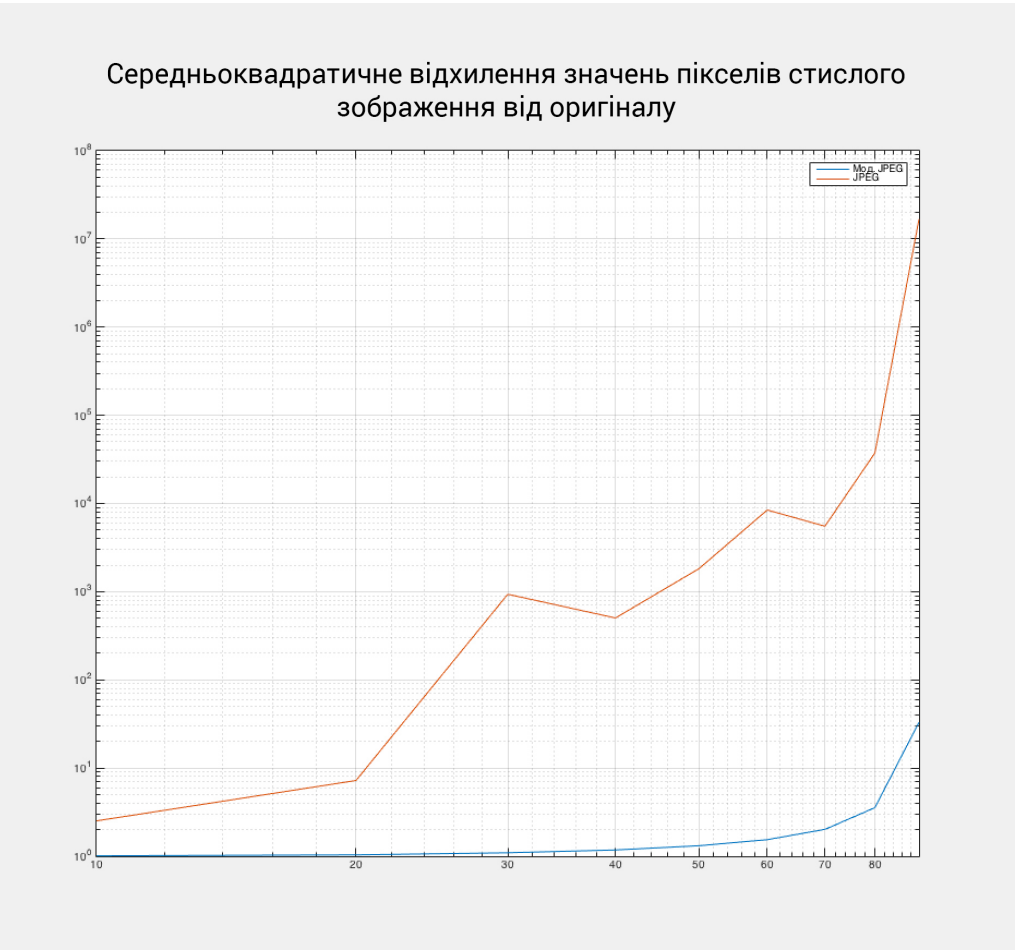

Рис.3.9. – Середньоквадратичне відхилення значень пікселів стислого зображення від оригіналу

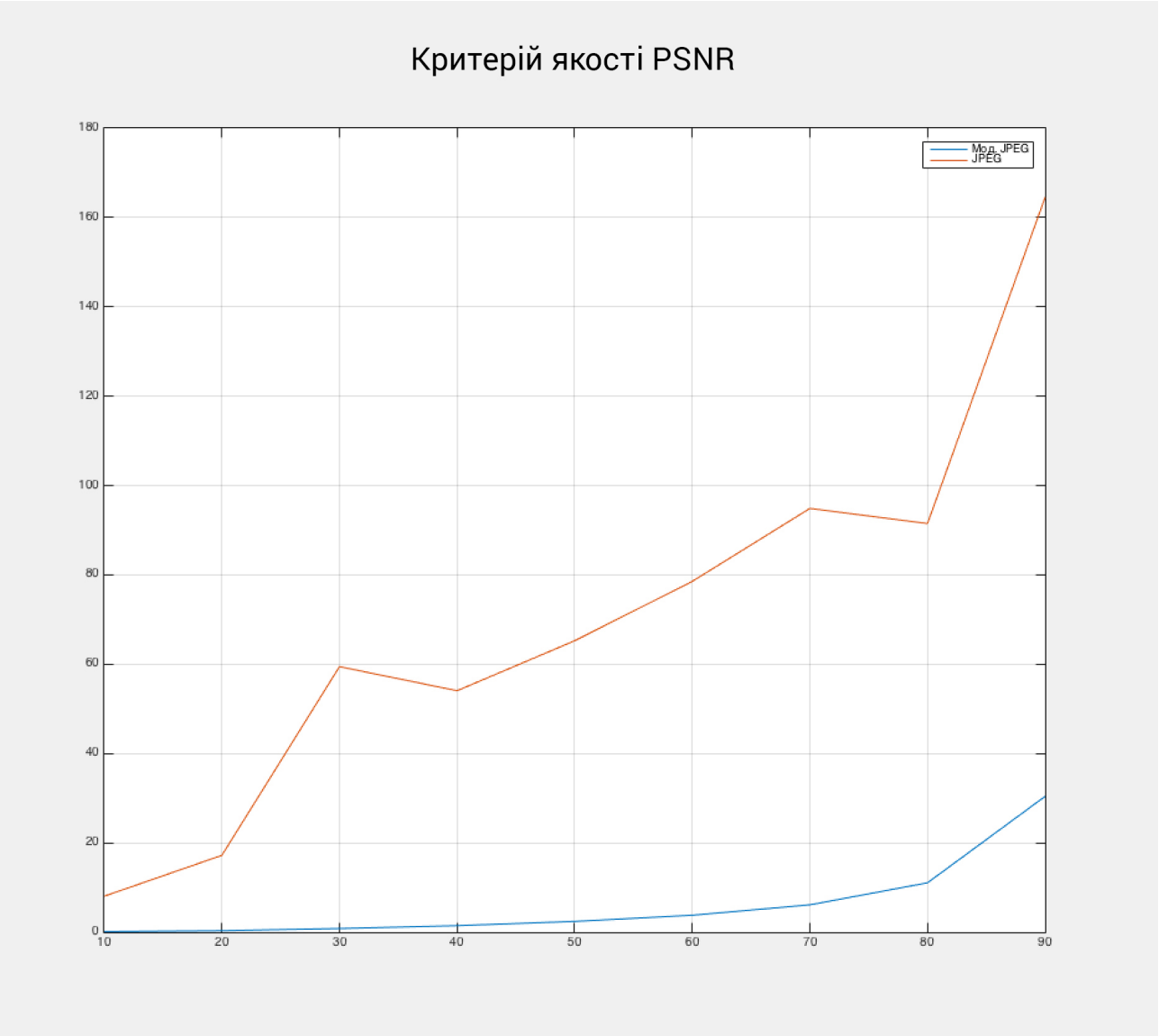

Рис.3.10. Критерій якості PSNR

3.2. Вплив зображень, стиснутих модифікованим алгоритмом, на швидкисть завантаження сайту

Вплив зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту було розглянуто на прикладі сайту http://kurspc.com.ua/. Роглянуто головну сторінку сайту, де міститься найбільша кількість зображень та найбільші зображення за розміром.

Як бачимоо з рис.3.11. та рис.3.12., швидкість завантаження сайту із зображеннями, стиснутими модифікованим методом складає 2,50с, а зі

звичайнимі зображеннями – 8,37с. Отже, мета була досягнута – модифікований алгоритм стиснення зображень JPEG дає змогу підвищити швидкість завантаження сайтів.

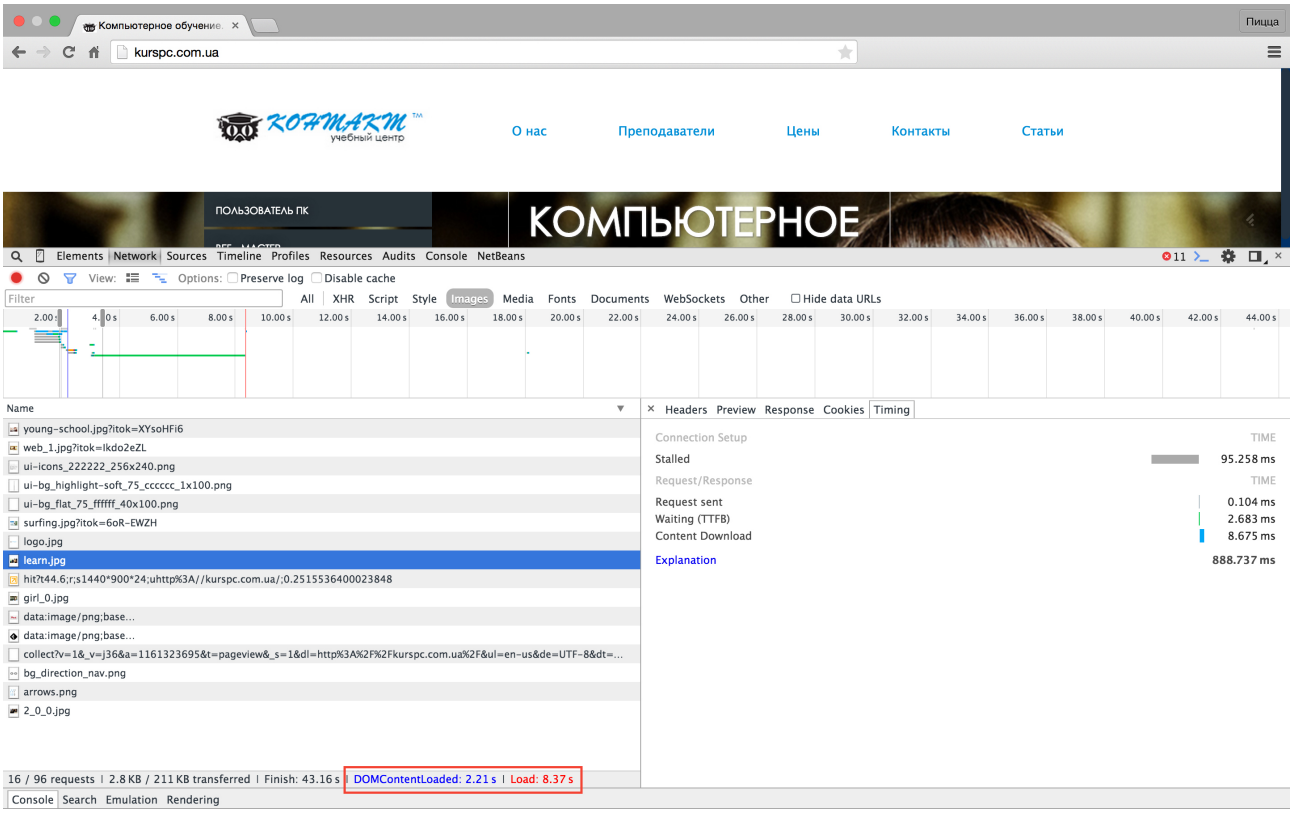

Рис.3.11. звичайна швидкість завантаження сайту

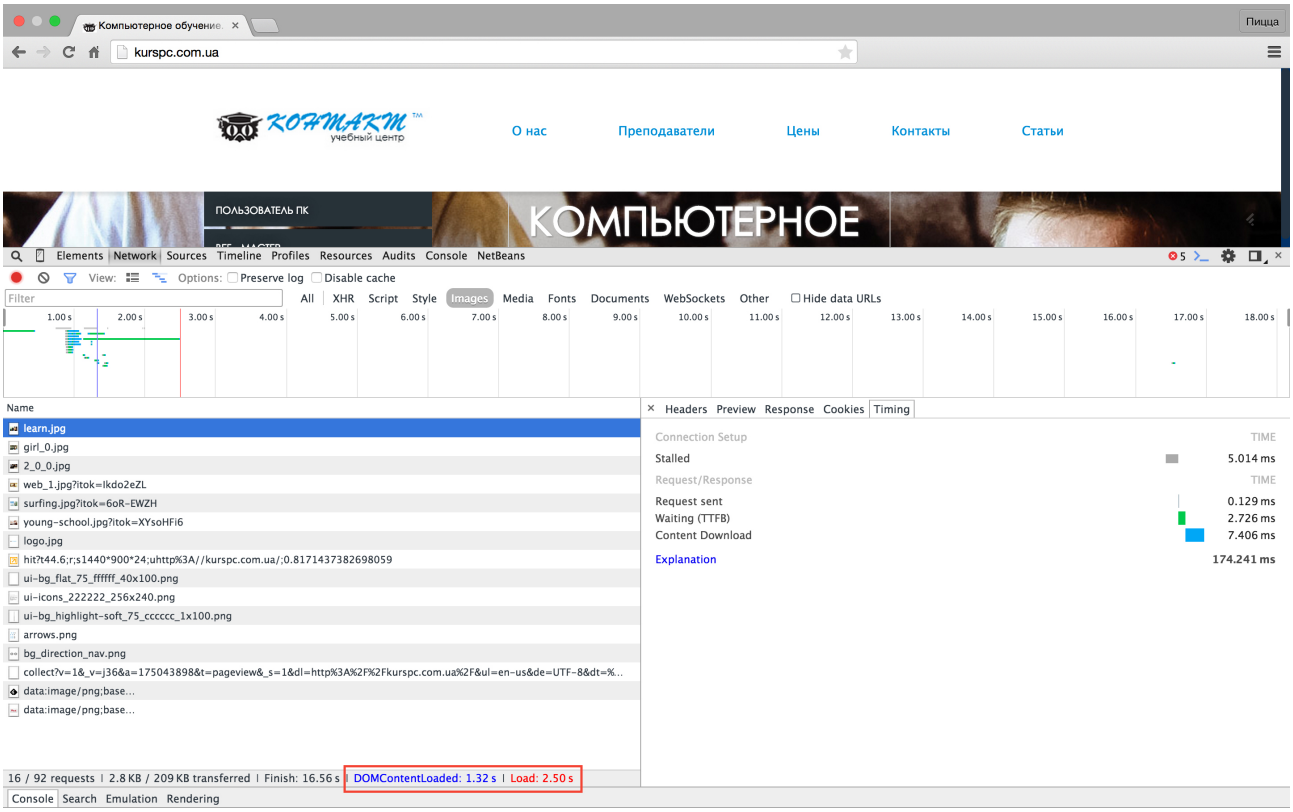

Рис.3.12. швидкість завантаження сайту із зображеннями, стиснутими модифікованим алгоритмом

Висновки до розділу

Проведено аналіз результатів оригінального та модифікованого алгоритмів стиснення зображень JPEG. Модифікований алгоритм показує значну перевагу. Оптимальним відношенням якість/розмір файлу є стиснення зображення до 80%.

Також проведено експерементальне дослідження впливу зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту. Результати дослідження показують прерогативу використання модифікованого алгоритму стиснення зображень.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даній роботі проводиться модифікація алгоритму стиснення зображень. Процесс модифікації алгоритму і розрахунку результатів потребує використання ПК.

При використані ПК, враховані усі вимоги по забезпеченню нормативних умов з використанням обчислювальної техніки НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних маши, НПАОП 0.00-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин .

Даний розділ дипломної роботи присвячений аналізу небезпечних і шкідливих факторів, які виникають при експлуатації ПК.

4.1. Загальна характеристика приміщення

Законом України «Про охорону праці», конституцією України, іншими нормативно-правовими актами закріплюється державна політика в області охорони праці. Вона базується на пріоритеті життя і здоров'я працівників, повної відповідальності роботодавця за створення безпечних і здорових умов праці, соціального захисту працівників, повного відшкодування збитків особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань, використання економічних методів управління, виконання нормативів охорони праці незалежно від форм власності і видів діяльності підприємства.

Приміщення розташоване на 5 поверсі 12-ти поверхової офісної будівлі. Це задовольняє НПАОП 0.00-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин, що не дозволяє розміщувати приміщення де використовується обчислювальна техніка на цокольних поверхах та у підвальних приміщеннях.

Робоче місце – це оснащений технічними засобами (засобами відображення інформації, органами управління, допоміжним обладнанням) простір, де здійснюється діяльність користувача (користувачів).

Організацією робочого місця називається система заходів щодо оснащення робочого місця засобами і предметами праці і розміщення їх у визначеному порядку.

Оператор ПК може стикатися з впливами таких факторів, як підвищений рівень шуму, відсутнє або погане природне чи штучне освітлення, підвищена чи низька температура середовища, а також ймовірність ураження струмом. В приміщенні встановлений комп'ютер asus vivobook s301la та один БФП samsung sl-m2070w.

Кількість робочих місць – 1. План приміщення розміщено на рис.4.1. Характеристики приміщення подані в табл.4.1.

Згідно НПАОП 0.00-1.31-2010 Правила охорони праці під час експлуатації електронно-обчислювальних машин та НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних маши при створенні робочих місць з ПЕОМ має враховуватися відстань між робочими столами з моніторами, яка має бути не менше 2 м, а відстань між бічними поверхнями моніторів – не менше 1,2 м. Приміщення, де знаходяться комп'ютери, повинні бути достатньо просторими і добре провітрюваними. Мінімальна площа на одне робоче місце – 6 $м^2$, мінімальний об'єм – 20 $м^3$.

Параметри приміщення задовольняють вимогам.

Кожне робоче місце включає в себе: системний блок, монітор, принтер та інші периферійні пристрої, які необхідні оператору. Під матричні принтери потрібно підкладати вібраційні килимки для гасіння вібрації та шуму.
Кожний програміст який працює за персональним комп'ютером, піддається впливу наступних факторів:
Освітлення;

Мікроклімат;

Шум і вібрації;

Електромагнітні випромінювання;

Статична електрика та ін.

Таблиця 4.1. Характеристики приміщення

Довжина	4,5 м
Ширина	4,5 м
Висота	3 м
Загальна площа	20,25 м^2
Загальний об’єм	60,75м^3
Кількість робочих місць	1
Площа на робоче місце	20,25 м^2
Об’єм на робоче місце	60,75м^3
Кількість вікон	3
Кількість батарей центрального водного опалення	1

4.2. Освітлення приміщення

Колір стін в приміщенні пастельно-голубий згідно ДБН-В.2.5-28-2006 Природне і штучне освітлення, що має коефіцієнт відбиття 40-50%. Необхідно провести розрахунок освітлення для кімнати площею 20,25 м ^{2,} ширина якої 4.5 м, довжина 4.5 м, висота – 3 м. Визначимо показники методом світлового потоку.

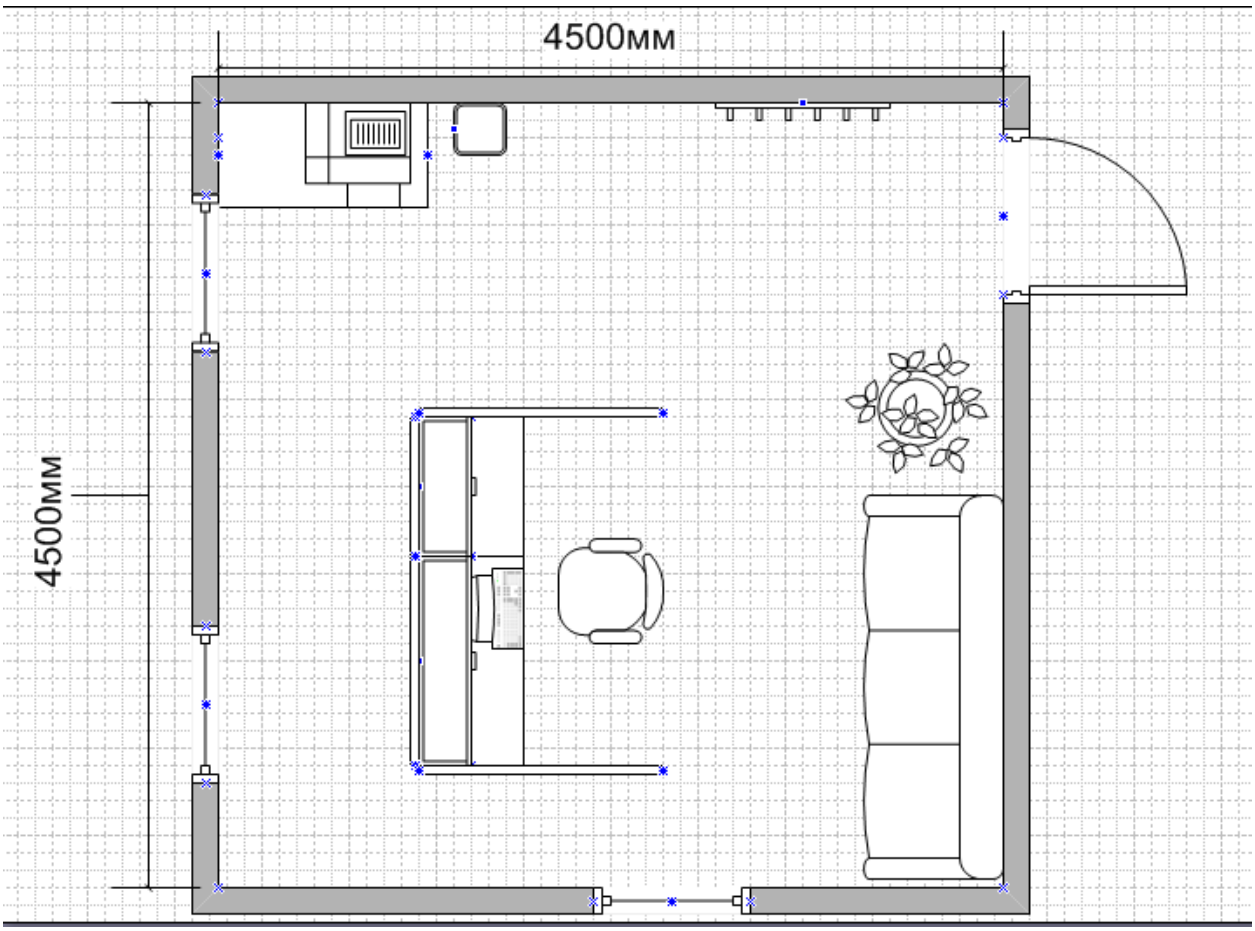

Рис.4.1 — Схема кімнати

Для визначення кількості світильників визначимо світловий потік, падаючий на поверхню за формулою:

$$F_{л} = \frac{E \cdot k \cdot s \cdot z}{N \cdot \eta} \qquad (4.1)$$

Де $F_{л}$ – розраховується світловий потік, лм;

E – нормована мінімальна освітленість;

s – площа освітлюваного приміщення;

z – відношення середньої освітленості до мінімальної ;

k – коефіцієнт запасу, враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації ;

n – коефіцієнт використання, (виражається відношенням світлового потоку, що

падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в частках одиниці; залежить від характеристик світильника, розмірів приміщення, фарбування стін і стелі, які характеризуються коефіцієнтами відображення від стін (p_c) і стелі (p_n)), значення коефіцієнтів p_c і p_n були зазначені вище: p_c = 50%, p_n = 75%. Значення n визначимо по таблиці коефіцієнтів використання різних світильників. Для цього обчислимо індекс приміщення по формулі :

$$i = \frac{s}{h \cdot (A + B)} \qquad (4.2)$$

Де s – площа приміщення;

h – розрахункова висота підвісу світильників над робочою поверхнею

A – ширина приміщення;

B – довжина приміщення;

$$h = H - h_p - h_c \qquad (4.3)$$

Де H – висота приміщення;

h_p = 0,7м – висота робочої поверхні від підлоги;

h_c = 0,6м – відстань від ламп до перекриття;

Таким чином,

h = 2,7-0,7-0,6 = 1,4м

Підставивши значення отримаємо:

$$i = \frac{32}{1.4 \cdot (8 + 4)} = 1{,}9 \qquad (4.4)$$

Знаючи індекс приміщення i, знаходимо n = 0.23.

Підставимо значення у формулу для визначення світлового потоку:

$$F_{\text{л}} = \frac{300 \cdot 1{,}5 \cdot 20{,}25 \cdot 1{,}1}{0{,}23 \cdot 6} = 7263.58 \text{ (Лм)}$$

Розрахуємо необхідну кількість ламп:

$$N = \frac{F_{\text{л}}}{F} \qquad (4.5)$$

Де N - обумовлений число ламп;

$F_{\text{л}}$ - світловий потік, $F_{\text{л}}$ = 7263,58 лм;

F - світловий потік лампи, F= 3050 лм.

$$N = \frac{7263{,}58}{3050} = 2{,}38$$

Число світильників – 4

Рекомендоване відношення відстаней між світильниками (l) до розрахованої висоти (h)

$$L = 1{,}6 \cdot 1{,}7 = 2{,}72 \Rightarrow 2$$

Габаритні розміри світильника (мм): 100x50x50. Схема розміщення на рис.4.2.

4.3. Параметри мікроклімату

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні. У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату (табл.4.2. – 4.3.). У санітарних

нормах ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень, встановлені величини параметрів мікроклімату, що створюють комфортні умови. Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт залежно від пори року і доби, чергування праці і відпочинку), так і технічні засоби (вентиляція, кондиціювання повітря, опалювальна система).

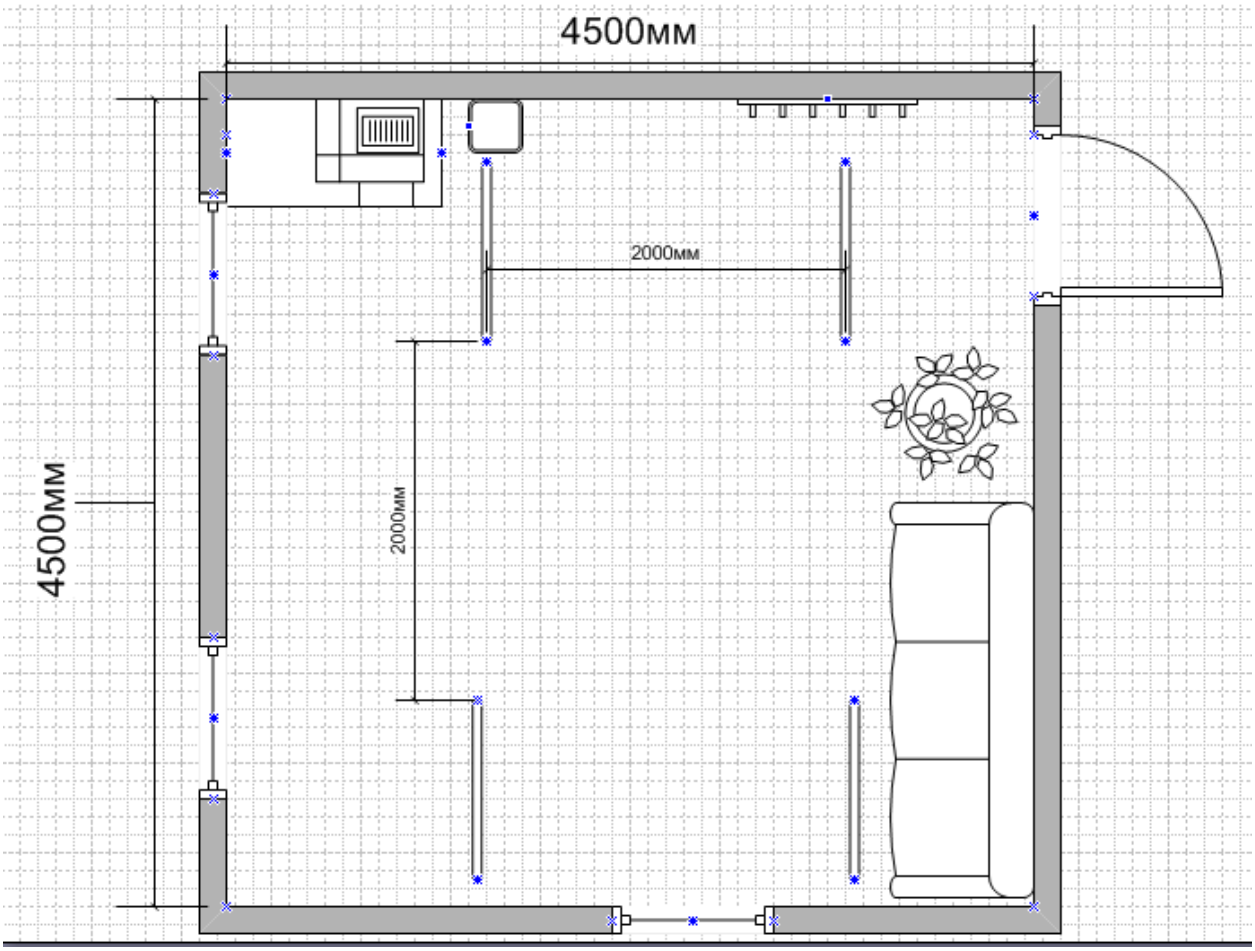

Рис.4.2. Розміщення світильників

У нашому випадку забезпечувати комфортні умови роботи фахівця буде вентиляція приміщення природного типу та кондиціонер. А в холодну сиснтема централізованого двотрубного опалення з автоматичними терморегуляторами.

Швидкість руху повітря у приміщенні приблизно 0.1 м/с, вологість -50%, температура повітря коливається від 21 ˚с до 23 ˚с. Шкідливі речовини у повітрі

відсутні. Виходячи з вищенаведеного і враховуючи, що робоче місце постійне, можна зробити висновок, що згідно ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень, мікрокліматичні умови знаходяться в оптимальних діапазонах.

Таблиця 4.2. Оптимальні параметри мікроклімату для приміщень, де встановлені комп'ютери.

Період року	*Параметр мікроклімату*	*Величина*
Холодний	Температура повітря в приміщенні відносна вологість швидкість руху повітря	22 ... 24 ° с 40 ... 60% до 0,1 м / с
Теплий	Температура повітря в приміщенні відносна вологість швидкість руху повітря	23 ... 25 ° с 40 ... 60% 0,1 ... 0,2 м / с

Таблиця 4.3. норми подачі свіжого повітря в приміщення, де розташовані комп'ютери.

Характеристика приміщення	Об'ємна витрата подається в приміщення свіжого повітря, м ^{3} / на одну людину в годину
Об'єм до 20м ^{3} на особу 20 ... 40м ^{3} на особу більш 40м ^{3} на особу	Не менше 30 не менше 20 природна вентиляція

4.4. Шум та вібрація

Рівень шуму на робочому місці програмістів і операторів відеоматеріалів не повинен перевищувати 50дба, а в залах обробки інформації на обчислювальних машинах - 65дба. Для зниження рівня шуму стіни і стеля приміщень, де

встановлені комп'ютери, облицьовуються звукопоглинальними матеріалами. Рівень вібрації в приміщеннях обчислювальних центрів знижується шляхом встановлення обладнання на спеціальні віброізолятори.

Рівні звукового тиску джерел шуму, що діють на оператора на його робочому місці представлені в табл.4.4.

Таблиця 4.4. - рівні звукового тиску різних джерел.

Джерело шуму	Рівень шуму, дба
Жорсткий диск	28
Кулер	26
Монітор	6
Клавіатура	12
Принтер	61
Сканер	34
Кондиціонер	38

Зазвичай робоче місце оператора оснащене наступним обладнанням: вінчестер в системному блоці, вентилятор систем охолодження ПК, монітор, клавіатура, принтер і сканер.

Підставивши значення рівня шуму для кожного виду обладнання у формулу, отримаємо:

$$l_\sigma = 10 \cdot lg\,(10^{2,9} + 10^{2,6} + 10^{0,6} + 10^{1,2} + 10^{6,1} + 10^{3,4} + 10^{3,8}) = 67,96 \text{ дб}$$

Отримане значення перевищує допустимий рівень шуму для робочого місця оператора, рівний 65 дб за ДСН 3.3.6-037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку, але якщо врахувати, що навряд чи такі периферійні пристрої як сканер і принтер будуть використовуватися одночасно, то ця цифра буде ще нижчою. Крім того, при роботі принтера безпосередню

присутність оператора необов'язково, тому що принтер обладнаний механізмом автоподачі аркушів.

4.5. Електромагнітне і іонізуюче випромінювання

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітора комп'ютера представлені в табл.4.5.

Максимальний рівень рентгенівського випромінювання на робочому місці оператора комп'ютера звичайно не перевищує 10мкбер / ч, а інтенсивність ультрафіолетового і інфрачервоного випромінювань від екрану монітора лежить в межах 10 ... 100мвт / м $^{2.}$

Таблиця 4.5. Допустимі значення параметрів неіонізуючих електромагнітних випромінювань відповідно до ДСН 239-96 Державні санітарні норми і правила захисту населення від впливу електромагнітних випромінювань

Найменування параметра	Допустимі
Напруженість електричної складової електромагнітного поля на відстані 50см від поверхні відео монітора	10в / м
Напруженість магнітної складової електромагнітного поля на відстані 50см від поверхні відео монітора	0,3 а / м
Напруженість електростатичного поля не повинна перевищувати: для дорослих користувачів для дітей дошкільних установ і що вчаться середніх спеціальних і вищих навчальних закладів	20кв / м 15кв / м

Яскравість екрану складає 200-225 кд/кв.м., що більше ніж мінімальні нормативні 100-120 кд/кв. Відношення яскравості екрану до яскравості

навколишніх поверхонь не перевищує 3:1.відстань від очей оператора до екрану складає 70 см в нормальному робочому положенні. Тривалість зосередженого спостереження для працюючих за ПЕВМ не перевищує 80 % усього часу спостереження.

Оператори ЕВМ роблять перерви 10-15 хвилин кожні 2 години одночасно з провітрюванням приміщення.

Усі перераховані параметри відповідають нормам.

4.6. Електробезпека

Приміщення відноситься до класу приміщень без підвищеної небезпеки для здоров'я працівників. Відносна вологість повітря не перевищує 60%, температура не перевищує 25°с. В приміщені присутні споживачі електроенергії: освітлювальні прилади та комп'ютери. В приміщені використовується прихована проводка, що виключає можливість дотику до оголених проводів. Проводка прокладена кабелем марки ввп 3*1,5 з мідними струмопровідними жилами; ізоляція виготовлена з пвх; кабель плоский з розділеною основою. Номінальна зміна напруги 220 в; частота до 50 гц. $I_н$=15 а, $i_ф$=15 а. Вмикачі штучного освітлення виконані з діелектричних матеріалів.

Для забезпечення електробезпеки застосовується занулення. Всі струмопровідні прилади ізольовані. В них використовується подвійне ізоляційне покриття (передня частина системного блоку виконана з пластику, що виключає дотик користувача до металічних частин). Блок живлення в корпусі комп'ютера знаходиться в окремому захисному кожусі, що виключає можливість доступу до високої напруги.

Отже, в приміщенні використовуються наступні засоби захисту:

При нормальних умовах :

- Ізоляція;
- Недоступність токовивідних частин;

- Інформаційні позначення (використання маркувань окремих частин електричного обладнання, написи, попереджувальні знаки);

При аварійних умовах:

- Заземлення;
- Подвійна ізоляція;
- Блокування безпеки.

Для попередження ураження електричним струмом використовується ряд організаційно-технічних дій:

- Розташування проводів живлення поза зоною переміщення людей;
- Всі співробітники ознайомленні з правилами техніки безпеки і в приміщенні знаходяться наступні пам’ятки для співробітників:
- Перша допомога постраждалому від електричного струму;
- Інструкція по техніці безпеки для працюючих в класах персональних ЕОМ.

Усі перераховані параметри відповідають НПАОП 0.00-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин.

4.7. Протипожежна безпека

Дане приміщення за класом горіння речовин відноситься до класу Е (горіння електроустановок під напругою), так як основними горючими матеріалами є монітори, периферійні пристрої та системні блоки комп’ютерів. Також дане приміщення можна віднести до класу А (горіння твердих речовин, що супроводжується або не супроводжується тлінням), так як в ньому знаходяться дсп столи, шухляди, дошка, папір. Вибухонебезпечні речовин в приміщенні відсутні.

У приміщенні відсутні вибухонебезпечні речовини, а також гарячі/холодні речовини, що займаються в процесі взаємодії з водою/киснем/самі з собою. Тому за НАПБ б.03.002-2007 Норми визначення категорій приміщень, будинків

та зовнішніх установок за вибухопожежною та пожежною небезпекою, приміщення відноситься до категорії В.

За класифікацією пожежонебезпечних зон приміщення відноситься до класу П-IIА.

В даному приміщенні основним джерелом запалювання може виступати замикання електропроводки (також замикання в середині електроприладів: комп'ютера, периферійних пристроїв).

В приміщені є один вогнегасник оу-2 (об'єм 3 літрів, маса заряду – 2кг). Отже, на площі 15,75 м^2 є 1 вогнегасник, що задовольняє вимогам ППБУ-2004 Правила пожежної безпеки в Україні (на 20 квадратних метрів необхідно 2 вогнегасника).

Технічними засобами виявлення пожежі обрані димові оптико-електронні автономні автоматичні сповіщувачі діп-43м (іп-212-43м) у кількості двох штук, встановлені на стелі приміщення на відстані, обумовленій ДБН В2.5-13-98 Державні будівельні норми.

Біля дверей розміщено план евакуації з даного приміщення на вулицю.

За пожежною безпекою приміщення відповідає НАПБ А.01.001-2004 Правила пожежної безпеки в Україні.

4.8. Дії при виникненні надзвичайних ситуацій

Офісна будівля, у якій розташоване приміщення, знаходиться у одній з промислових зон міста Києва. Поруч із ним розташовані заводи та їх цехи. Тому виникає небезпека аварії на вибухонебезпечних об'єктах. Розглянемо ситуацію та проведемо оцінювання інженерної та пожежної обстановки під час такої ситуації.

Розрахунки:

$$Зона\ I:\ r1 = 17{,}5 \cdot \sqrt[3]{Q} = 17{,}5 * 7{,}94 = 138{,}95 \text{ м}$$

$$\Delta Pф = 30 \text{кПа}$$

$$Зона\ II:\ r2 = 1{,}7 * r1 = 1.7*138.95 = 236.22 \text{ м}$$

$$\Delta PII = 1300(r1/r0)^3+50=1300(138.95/500)^3+50=28.6+50=78.6$$

Висновок: об’єкт опиниться за межами цих зон, тобто у зоні повітряної ударної хвилі (зона III)

$$\Delta P_{\phi} = \frac{262}{\sqrt{1 + 7{,}66 \cdot 10^{-5} \cdot \frac{L^3}{Q} - 1}} = 30.1 \text{ кПа}$$

Ступінь руйнування будівлі – *Середня*

Характеристика руйнувань будівлі:

Руйнування даху;

Легких внутрішніх перегородок;

В капітальних стінах з’являються тріщини.

Ступінь руйнування контрольно-вимірювальної апаратури – сильні

Ступінь руйнування кабельних ліній – слабкі

Ступінь ураження людей від прямої дії УХ – слабкі

Характеристика уражень людей: Легка контузія організму, часткова втрата слуху, вивихи кінцівок.

Ступінь вогнестійкості – III ступінь вогнестійкості

Очікувана пожежна обстановка - для виробництва категорії пожежної небезпеки В, ступеня вогнестійкості будівель – III, при надмірному тику 25 кПа і щільності забудови більше 20% можна очікувати в перші 30 хвилин окремі пожежі з переростанням за 1…2 год в суцільну.

Небезпечна кількість вибухової речовини для уникнення руйнувань будівлі – 40т

Небезпечна кількість вибухової речовини для уникнення будь-яких руйнувань – менше 12 т

Загальні висновки і рекомендації:

На відстані 500 м від бізнес-центру стався вибух тротилу, що призвело до руйнувань будівлі, елементів бізнес-центру, постраждали люди. В першу чергу треба сповістити про НС службу порятунку, а також лікарів. Не дивлячись на

те, що загроза для людей є невисокою, слід забезпечити кожного першою невідкладною допомогою.

Таблиця 4.6. Вихідні дані

Вхідні дані		Одиниці вимірювання
Відстань від офісної будівлі до місця аварії (вибуху)	500	метри
Маса вибухової речовини (тротил)	500	тонни
Категорія виробництва з пожежної безпеки	В	
Щільність забудови об’єкту	20	%
Характеристики елементів будівлі бізнес-центру:		
Будівля	З легким металевим каркасом	
Верстати	відсутні	
Кабельні лінії	наземні	
Контрольно-вимірювальна апаратура	наявна	
Границі вогнетривкості несучих стін	0,5 годин	
Границі вогнетривкості перегородок	0,25 годин	

Серед рекомендацій, спрямованих на зменшення заподіяної шкоди та уражень людей, можуть бути такі:

Укріпити будівлю установленням додаткових колон, підкосів;

Створити 50% запас контрольно-вимірювальної апаратури;

Установити на вікнах захисні металеві сітки, щоб розбите скло не потрапляло в приміщення бізнес-центру;

Установити і регулярно контролювати стан вогнегасників та інших протипожежних систем;

Порушити питання перед відповідними органами про зменшення запасу вибухонебезпечної речовини до безпечної кількості.

Висновки до розділу

В даному розділі були розглянуті питання, що стосуються безпеки використання та правильності експлуатації комп’ютерної техніки. Мікроклімат приміщення, освітленість відповідають відповідним вимогам та нормам. Шум несуттєво перевищує задані норми, але не знаходиться на критичному рівні через специфічність техніки (принтер). Електробезпека та пожежна безпека також забезпечені в повній мірі. На підставі цього можна зробити висновок, що робочі місця в даному приміщенні задовольняють екологічним нормам і вимогам та створюють безпечні умови праці.

ВИСНОВКИ

1. У магістерській дисертації розглянуті методи стиснення зображень, проведено їх аналіз та порівняння. На основі аналізу виявлено, що можливо покращити алгоритм стиснення JPEG.
2. Зроблено постановку задачі: для модифікації обрано алгоритм стиснення JPEG та вейвлет-перетворення у якості модифікації.
3. Розглянуто вейвлет-перетворення та експерементальне дослідження впливу кількості усереднюючих коефіцієнтів на якість зображення. Проаналізовано результати та на їх основі запропонована модифікація алгоритму стиснення зображень JPEG вейвлет-перетворенням з чотирма коефіцієнтами усережнення.
4. Реалізовано модифікований алгоритм JPEG та проведено експерементальне дослідження його роботи. Проаналізовано результати дослідження та обрано оптимальне відношення якість/розмір файлу – стиснення зображення до 80%.
5. Проведено дослідження впливу зображень, стиснутих модифікованим алгоритмом, на швидкість завантаження сайту. Такий вплив є досить ефективним і може бути використаний для підвищення швидкості завантаження сайтів.
6. В даному розділі були розглянуті питання, що стосуються безпеки використання та правильності експлуатації комп’ютерної техніки. Місця в даному приміщенні задовольняють екологічним нормам і вимогам та створюють безпечні умови праці.

СПИСОК ЛІТЕРАТУРИ

1. Миано Дж. Форматы и алгоритмы сжатия изображений в действии: пер. С англ. / Дж. Миано — М.:Триумф, 2003. —335 с.
2. Яковлев А.Н. Введение в вейвлет-преобразовани. / А.Н. Яковлев — Новосибирск: НГТУ, 2003. - 104 с.
3. Уэлстид С. Фракталы и вейвлеты для сжатия изображений в действии: пер. с англ. / С. Уэлстид — М.:Триумф, 2003. —319 с.
4. Исследование вейвлетного метода сжатия изображений для повышения быстродействия веб приложений : науч.-техн. сб. Киев: Национальный технический университет Украины “Киевский политехнический институт” / Е.В. Крылов, В.К. Аникин, Е.В. Аникина – 2014.
5. Блаттер К. Вейвлет-анализ. Основы теории.// – М. – Техносфера. - 2006, - 279 с.
6. Брейсуэлл Р. Преобразование Хартли. Теория и приложения.// - М. - Мир, - 1990. – 192 с.
7. Гонсалес Р., Вудс Р. Цифровая обработка изображений.// Пер. с англ.- Москва.- Техносфера. – 2006. -1072 с.
8. Ковалгин Ю.А., Вологодин Э.И. Цифровое кодирование звуковых сигналов// – СПб. - КОРОНА-принт, 2004. – 240 с.
9. Сэломон Д. Сжатие данных, изображений и звука// – М. – Техносфера. - 2004. – 368 с.
10. Тропченко А.А., Молчанов В.А. Особенности сжатия цветных изображений JPEG-подобными алгритмами// Научно-технический вестник СПбГУ ИТМО/ - Вып. 32. - СПб. - СПбГУ ИТМО. – 2006. - с.22-26
11. Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие – СПб: СПбГУ ИТМО, 2009. – 108 с.

12. НПАОП 0.00-1.31-99 Правила охорони праці під час експлуатації електронно-обчислювальних машин.
13. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин
14. ДСН 3.3.6.042-99. Державні санітарні норми мікроклімату виробничих приміщень.
15. ДБН В.2.5-28-2006. Природне і штучне освітлення.
16. ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку.
17. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою
18. НАПБ А.01.001-2004 Правила пожежної безпеки в Україні
19. ДБН В.2.5-56:2010 Інженерне обладнання будинків і споруд. Системи протипожежного захисту

ДОДАТКИ

Додаток А

Лістинг реалізації модифікованого алгоритму стиснення JPEG

```
clc;
clear all;
I = imread('cameraman.tif');
I1=I;
[row coln P]= size(I);
I= double(I);

%----------------------------------------------------------
% Subtracting each image pixel value by 128
%----------------------------------------------------------
I = I - (128*ones(256));

quality = input('What quality of compression you require - ');

%----------------------------------------------------------
% Quality Matrix Formulation
%----------------------------------------------------------
Q50 = [ 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];

 if quality > 50
    QX = round(Q50.*(ones(8)*((100-quality)/50)));
    QX = uint8(QX);
 elseif quality < 50
```

```
    QX = round(Q50.*(ones(8)*(50/quality)));
    QX = uint8(QX);
 elseif quality == 50
    QX = Q50;
 end


%-------------------------------------------------------
% Jpeg Compression
%-------------------------------------------------------
restored = zeros(row,coln);
QX = double(QX);


%-------------------------------------------------------
% Jpeg Encoding
%-------------------------------------------------------

X = I;
iter = 2; % number of wavelet iterations
cf = 1-(quality/100);
M = row;
N = coln;



% Haar basis constants

avg = 1/sqrt(2)*[1;1];
diff = 1/sqrt(2)*[1;-1];
```

```
%-------------------------------------------------------------------------
% Compute the wavelet coefficients
%-------------------------------------------------------------------------

tic
   WX = X;
   for i=1:iter
        %       if strcmp(type,'Haar')

        % Generate the M/(2^(i-1)) mother and father Haar wavelets
        mother_m = [avg;zeros(M/(2^(i-1))-2,1)];
        father_m = [diff;zeros(M/(2^(i-1))-2,1)];

        % Generate the N/(2^(i-1)) mother and father Haar wavelets
        mother_n = [avg;zeros(N/(2^(i-1))-2,1)];
        father_n = [diff;zeros(N/(2^(i-1))-2,1)];

        % Compute wavelet transform of upper left quadrant
        fft_mother_n_tilde = fft([mother_n(1);mother_n(end:-1:2)]');
        fft_father_n_tilde = fft([father_n(1);father_n(end:-1:2)]');
        fft_mother_m_tilde = fft([mother_m(1);mother_m(end:-1:2)]);
        fft_father_m_tilde = fft([father_m(1);father_m(end:-1:2)]);
        for p = 1:P
           for j = 1:M/(2^(i-1))
              lowfreq = ifft(fft_mother_n_tilde.*fft(WX(j,1:N/(2^(i-1)),p)));
              detail = ifft(fft_father_n_tilde.*fft(WX(j,1:N/(2^(i-1)),p)));
             WX(j,1:N/(2^(i-1)),p) = [lowfreq(1:2:end) detail(1:2:end)];
           end
```

```
            for j = 1:N/(2^(i-1))
                lowfreq = ifft(fft_mother_m_tilde.*fft(WX(1:M/(2^(i-1)),j,p)));
                detail = ifft(fft_father_m_tilde.*fft(WX(1:M/(2^(i-1)),j,p)));
                WX(1:M/(2^(i-1)),j,p) = [lowfreq(1:2:end);detail(1:2:end)];
            end
        end


    end
    ForwardTransformTime = toc %#ok

%-------------------------------------------------------------------------

%-------------------------------------------------------------------------
% Compress image by zeroing out cf% of the wavelet coefficients
%-------------------------------------------------------------------------

WXc = WX;
if cf > 0
    % Zero out smallest cf% wavelet coefficients in absolute value
    coeffs = zeros(1,M*N*P);
    for p = 1:P
        for i = 1:M
            coeffs(M*N*(p-1) + ((i-1)*N+1:N*i)) = abs(WX(i,:,p));
        end
    end
    sortedcoeffs = sort(coeffs);
    thresh = sortedcoeffs(ceil(cf*M*N*P));
    indices = find((abs(WX) <= thresh));
    WXc(indices(randperm(length(indices),floor(cf*M*N*P)))) = 0;
```

```
elseif cf == -1
   % Keep the low frequency approximation and zero out everything else
   WXc = zeros(M,N,P);
   for p = 1:P
      WXc((M-M/(2^iter)+1):end,(N-N/(2^iter)+1):end,p) = WX((M-
M/(2^iter)+1):end,(N-N/(2^iter)+1):end,p);
   end
end

%-----------------------------------------------------------------------

in = WXc;

% Z-scanning    initializing the variables
%-----------------------------------------
h = 1;
v = 1;

vmin = 1;
hmin = 1;

vmax = size(in, 1);
hmax = size(in, 2);

i = 1;

output = zeros(1, vmax * hmax);
%---------------------------------

while ((v <= vmax) & (h <= hmax))
```

```
if (mod(h + v, 2) == 0)             % going up

   if (v == vmin)
      output(i) = in(v, h);        % if we got to the first line

      if (h == hmax)
     v = v + 1;
   else
        h = h + 1;
      end;

      i = i + 1;

   elseif ((h == hmax) & (v < vmax))   % if we got to the last column
      output(i) = in(v, h);
      v = v + 1;
      i = i + 1;

   elseif ((v > vmin) & (h < hmax))    % all other cases
      output(i) = in(v, h);
      v = v - 1;
      h = h + 1;
      i = i + 1;
 end;

else                                % going down

  if ((v == vmax) & (h <= hmax))       % if we got to the last line
      output(i) = in(v, h);
```

```
            h = h + 1;
            i = i + 1;

        elseif (h == hmin)                % if we got to the first column
            output(i) = in(v, h);

            if (v == vmax)
          h = h + 1;
         else
              v = v + 1;
            end;

            i = i + 1;

        elseif ((v < vmax) & (h > hmin))     % all other cases
            output(i) = in(v, h);
            v = v + 1;
            h = h - 1;
            i = i + 1;
         end;

    end;

    if ((v == vmax) & (h == hmax))          % bottom right element
        output(i) = in(v, h);
        break
    end;

end;
```

output

```
%-----------------------------------------------------------------------
% Reconstruct orignal image from compressed wavelet coefficients
%-----------------------------------------------------------------------
   tic
   Xhat = WXc;
   for i=iter:-1:1

        % Generate the M/(2^(i-1)) mother and father Haar wavelets
        mother_m = [avg;zeros(M/(2^(i-1))-2,1)];
        father_m = [diff;zeros(M/(2^(i-1))-2,1)];

        % Generate the N/(2^(i-1)) mother and father Haar wavelets
        mother_n = [avg;zeros(N/(2^(i-1))-2,1)];
        father_n = [diff;zeros(N/(2^(i-1))-2,1)];

        % Compute inverse wavelet transform of upper left quadrant
        fft_mother_m = fft(mother_m);
        fft_father_m = fft(father_m);
        fft_mother_n = fft(mother_n);
        fft_father_n = fft(father_n);
        for p = 1:P
           for j = 1:N/(2^(i-1))
              up1 = upsample(Xhat(1:M/(2^i),j,p),2);
              up2 = upsample(Xhat(M/(2^i)+1:M/(2^(i-1)),j,p),2);
              Xhat(1:M/(2^(i-1)),j,p) = ifft(fft_mother_m.*fft(up1(:))) +
ifft(fft_father_m.*fft(up2(:)));
           end
           for j = 1:M/(2^(i-1))
```

```
                up1 = upsample(Xhat(j,1:N/(2^i),p),2);
                up2 = upsample(Xhat(j,N/(2^i)+1:N/(2^(i-1)),p),2);
                Xhat(j,1:N/(2^(i-1)),p) = (ifft(fft_mother_n.*fft(up1(:))) +
ifft(fft_father_n.*fft(up2(:))))';
            end
        end

    end
    InverseTransformTime = toc %#ok

I2 = Xhat;

% ----------------------------------------------------
% Conversion of Image Matrix to Intensity image
%----------------------------------------------------

K=mat2gray(I2);

%----------------------------------------------------
% Formulation of forward DCT Matrix and inverse DCT matrix
%-------------------------------------------

DCT_matrix8 = dct(eye(8));
iDCT_matrix8 = DCT_matrix8';   %inv(DCT_matrix8);

%----------------------------------------------------
% Jpeg Compression
%----------------------------------------------------

dct_restored = zeros(row,coln);
```

```
QX = double(QX);

%-----------------------------------------------------
% Jpeg Encoding
%-----------------------------------------------------
%-----------------------------------------------------
% Forward Discret Cosine Transform
%-----------------------------------------------------

for i1=[1:8:row]
   for i2=[1:8:coln]
      zBLOCK=I(i1:i1+7,i2:i2+7);
      win1_dct=DCT_matrix8*zBLOCK*iDCT_matrix8;
      dct_domain(i1:i1+7,i2:i2+7)=win1_dct;
   end
end

%-----------------------------------------------------
% Quantization of the DCT coefficients
%-----------------------------------------------------

for i1=[1:8:row]
   for i2=[1:8:coln]
      win1_dct = dct_domain(i1:i1+7,i2:i2+7);
      win2_dct=round(win1_dct./QX);
      dct_quantized(i1:i1+7,i2:i2+7)=win2_dct;
   end
end
```

```
%-----------------------------------------------------------
% Jpeg Decoding
%-----------------------------------------------------------
% Dequantization of DCT Coefficients
%-----------------------------------------------------------

for i1=[1:8:row]
   for i2=[1:8:coln]
      win2_dct = dct_quantized(i1:i1+7,i2:i2+7);
      win3_dct = win2_dct.*QX;
      dct_dequantized(i1:i1+7,i2:i2+7) = win3_dct;
   end
end

%-----------------------------------------------------------
% Inverse DISCRETE COSINE TRANSFORM
%-----------------------------------------------------------

for i1=[1:8:row]
   for i2=[1:8:coln]
      win3_dct = dct_dequantized(i1:i1+7,i2:i2+7);
      win4_dct=iDCT_matrix8*win3_dct*DCT_matrix8;
      dct_restored(i1:i1+7,i2:i2+7)=win4_dct;
   end
end
I2_dct=dct_restored;



% -----------------------------------------------------
```

```
% Conversion of Image Matrix to Intensity image
%-------------------------------------------------------


K_dct=mat2gray(I2_dct);


%-------------------------------------------------------
%Display of Results
%-------------------------------------------------------

K1 = single(K);
K1_dct = single(K_dct);

saveas(gcf,'jpg.jpg')

figure(2);imshow(I1);

saveas(gcf,'orig.jpg')

figure(3);imshow(K1);

saveas(gcf,['wavelet-',num2str(quality),'.jpg'])

figure(4);imshow(K1_dct);

saveas(gcf,['dct-',num2str(quality),'.jpg'])
```