

РОЗДІЛ 4

СТИСНЕННЯ ВІДЕОЗОБРАЖЕНЬ, ЩО МІСТЯТЬ ВИМІРЮВАЛЬНУ ІНФОРМАЦІЮ, НА ОСНОВІ ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ

4.1. Метод стиснення цифрових відеозображень на основі вейвлет-перетворення

На основі вейвлет-перетворення існує стандартний метод стиснення цифрових відеозображень JPEG-2000 [145, 204]. Метод стиснення відеозображень JPEG-2000 на сьогоднішній день вважається одним із кращих і по ефективності стиснення помітно перевершує найпоширеніший метод JPEG. Цей метод заснований на вейвлет-перетворенні цифрового відеозображення і обробці отриманих вейвлет-коефіцієнтів, які стискаються значно краще, ніж відліки початкового відеозображення. Разом з тим вейвлет-перетворення і обробка коефіцієнтів досить складні та вимагають великої обчислювальної потужності комп'ютера. Тому JPEG-2000 застосовують в тих випадках, коли потрібно одержати як можна кращу якість стиснутого відеозображення, а час виконання стиснення є не критичним. В зв'язку з цим великий інтерес представляють способи, що дозволяють поліпшити часові характеристики вейвлет-перетворень.

Суть алгоритму перетворення цифрового відеозображення на основі методу JPEG-2000 полягає в наступному. До початкового цифрового відеозображення застосовується дискретне вейвлет-перетворення із заданим числом рівнів декомпозиції. Дискретне вейвлет-перетворення – це фільтрація рядка (або стовпця) початкового відеозображення двома вейвлет-фільтрами. Це КІХ-фільтри: низькочастотний фільтр L та високочастотний фільтр H .

Оскільки після фільтрації частотний діапазон відеосигналу скорочується у два рази, профільтовані дані, відповідно до теореми Котельникова, можна прорідити удвічі, тобто виключити всі парні коефіцієнти. Потім з отриманих компонент складається новий рядок, як показано на рис. 4.1.

$$\begin{array}{c}
 \begin{array}{c} \text{X} \\ \boxed{x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6} \end{array} \\
 \\
 l_i = \sum_j L_j x_{j-1} \quad h_i = \sum_j H_j x_{j-1} \\
 \\
 \begin{array}{cc} \boxed{l_1 \quad l_2 \quad l_3} & \boxed{h_1 \quad h_2 \quad h_3} \end{array} \\
 \\
 \begin{array}{cc} L & H \\ \boxed{l_1 \quad l_2 \quad l_3} & \boxed{h_1 \quad h_2 \quad h_3} \end{array}
 \end{array}$$

Рис. 4.1. Вейвлет-перетворення одновимірного сигналу

Для двовимірного сигналу існує окреме двовимірне вейвлет-перетворення. Але його досить складно реалізувати і на практиці його звичайно замінюють двома одновимірними. Така заміна не погіршує стиснення, але значно спрощує фільтрацію. В одновимірному випадку спочатку фільтрують

рядки, потім стовпці. Відеозображення при цьому розділяється на 4 частини (рис. 4.2).

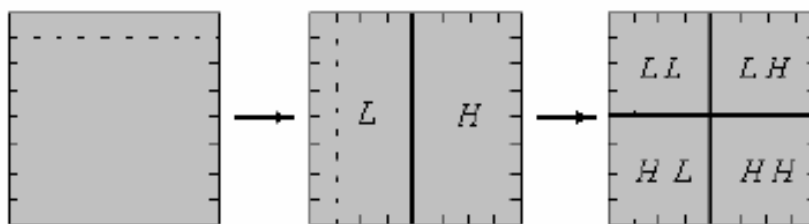


Рис. 4.2. Фільтрація двовимірного відеосигналу цифрового відеозображення

Потім до частини, що профільтована тільки низькочастотними фільтрами, застосовують ту ж саму процедуру. Такий алгоритм зветься алгоритмом Малла (рис. 4.3).

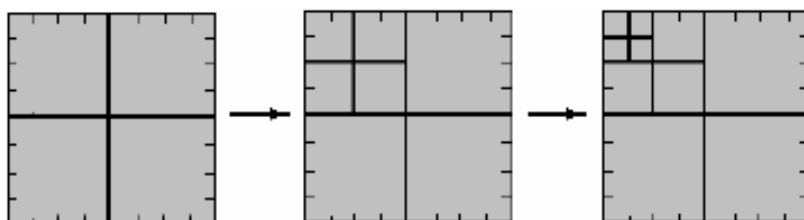


Рис. 4.3. Алгоритм Малла

Результат вейвлет-перетворення представлено на рис. 4.4.



Рис. 4.4. Результат вейвлет-перетворення

В даному випадку наведено дворівневе вейвлет-перетворення. Частина відеозображення у верхньому лівому куті профільтована тільки низькочастотними фільтрами та представляє собою згладжену зменшену копію оригіналу. Вся інша область відеозображення профільтована як низькочастотними, так і високочастотними фільтрами. На ній помітні великі ділянки близьких до нуля вейвлет-коефіцієнтів і окремі яскраві елементи, які означають різкі перепади яскравості на початковому відеозображенні. Ця частина дуже добре підлягає для стиску навіть звичайними алгоритмами.

Власне, одержання великої кількості нульових коефіцієнтів і є основною метою вейвлет-перетворення, саме ж перетворення стиску не дає.

Оптимальним за критерієм «якість-складність реалізації» є дворівневе вейвлет-перетворення. Значне збільшення числа рівнів не приводить до помітного підвищення ефективності стиску, а складність алгоритму істотно зростає.

Після перетворення здійснюється квантування коефіцієнтів. Саме на етапі квантування виникають основні інформаційні втрати, і саме за рахунок квантування можливо істотне зменшення об'єму подання відеозображення. Квантування може бути або рівномірним скалярним, або більш складним, наприклад, векторним. У випадку використання рівномірного скалярного квантування його параметри можуть мінятися залежно від квантуемого діапазону.

Далі застосовується послідовне кодування всіх частотних блоків LL2, LH2 тощо. Кодування здійснюється арифметичним методом за допомогою імовірнісної моделі і проводиться в наступній послідовності:

1) визначається розрядність коефіцієнтів блоку за формулою

$$N_{\max} = \text{ceil}(\log_2(C_{\max})),$$

де C_{\max} - максимальний за модулем коефіцієнт.

2) у всіх коефіцієнтів кодуються біти на місці N_{\max} , потім $N_{\max} - 1$ тощо.

Такий спосіб називається кодуванням поверхів бітів (bit plane), в алгоритмі JPEG-2000 кожний поверх бітів кодується в три стадії, які називаються significance pass, refinement pass, cleanup pass (рис. 4.5). Ймовірнісна модель для арифметичного кодування будується на тому факті, що великі та малі коефіцієнти блоку не перемішані рівномірно, а концентруються в окремих областях, і це дозволяє судити про значення кодуемого біта, розглядаючи величину найближчих вейвлет-коефіцієнтів.

Відмінною рисою JPEG-2000 є наявність так званого методу оптимізації довжини коду блоків (optimized truncation). Біти послідовно кодуються від значення N_{\max} аж до молодшого біта (LSB, least significant bit), причому закодовані дані відразу не посилають у вихідний потік, а зберігаються в пам'яті. Після того, як закодовані всі блоки, у пам'яті залишається набір закодованих даних (для дворівневого вейвлет-перетворення цей набір складається з 7 елементів). Задача методу – оптимізувати довжину елементів таким чином, щоб забезпечити необхідний ступінь стиснення та одночасно так підібрати співвідношення довжин елементів, щоб середньоквадратична похибка стиснення була найменшою.

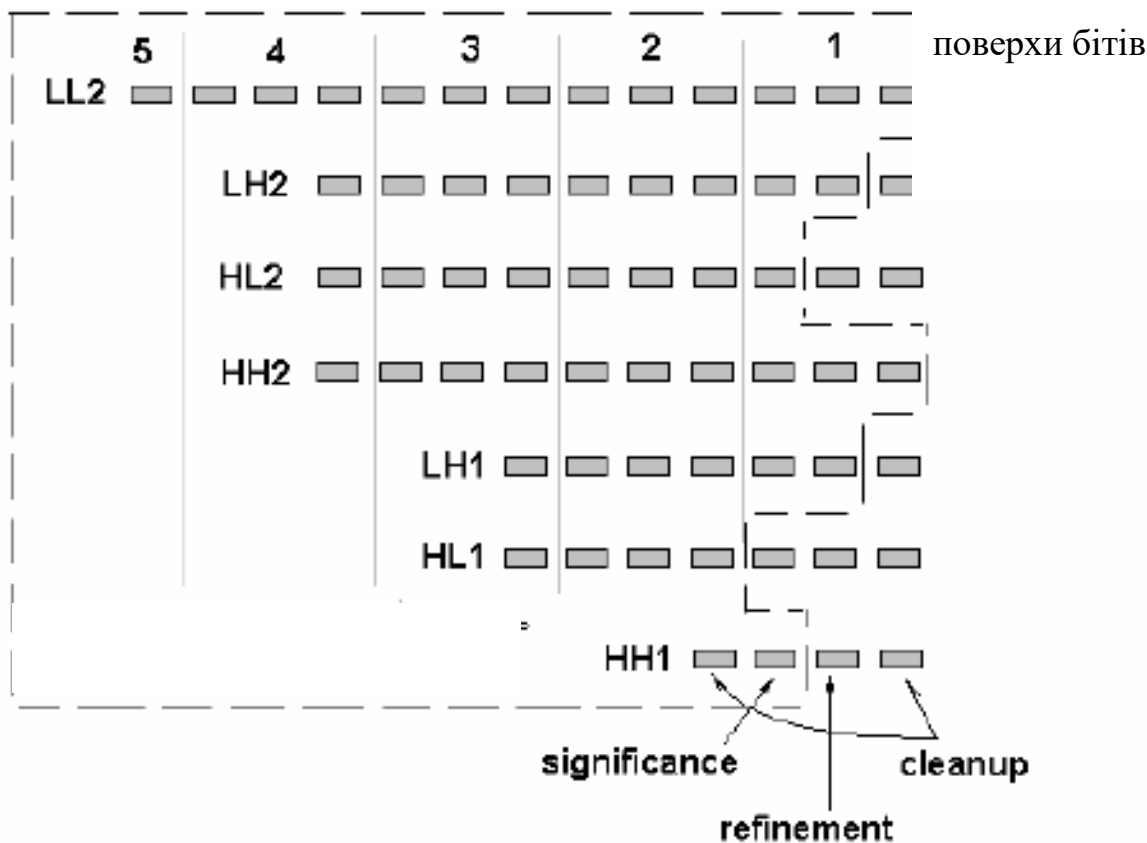


Рис. 4.5. Оптимізація довжин закодованих даних

У пам'яті залишаються тільки ті дані, які забезпечують кращу якість стиснуто відеозображення при заданому ступені стиснення.

Подання інформації, отриманої в результаті вейвлет-перетворення, дуже зручно тим, що воно забезпечує можливість одержання приблизних копій відеозображення без здійснення повного зворотного перетворення. Зворотне перетворення здійснюється в порядку, зворотному до порядку прямого перетворення. Роблячи обмежене число зворотних декомпозицій (об'єднань частотних діапазонів) та вважаючи, що всі не залучені в перетворення частотні діапазони містять винятково нульові елементи, ми легко можемо одержати або копію відеозображення в зменшеному масштабі, або відновлене відеозображення, але більш низької якості в порівнянні з відеозображенням, отриманим в результаті повного зворотного перетворення. З огляду на той факт, що блоки вейвлет-коефіцієнтів кодуються незалежно один від одного, ми одержуємо можливість часткового декодування не тільки на рівні перетворення, але і на рівні інтерпретації коду. Для одержання приблизної копії відеозображення досить декодувати лише частину інформації, а потім зробити часткове зворотне перетворення. Таким чином, формат зберігання відеозображення забезпечує масштабованість як за розподільчою здатністю, так за якістю.

Іншою важливою перевагою даного методу є можливість доступу до окремих елементів відеозображення без повного декодування його представлення. Забезпечується така можливість по-перше, розбивкою початкового відеозображення на не перехресні області, які кодуються як окремі

відеозображення, а по-друге, представленням коду окремої області у вигляді частин (шарів), кожна з яких є сумарним кодом коефіцієнтів, що відповідають деякій області. Відзначимо, що шари у свою чергу діляться на так звані пакети, що містять код блоків коефіцієнтів на різних рівнях декомпозиції. Для того, щоб декодувати будь-яку область відеозображення, досить визначити її розташування в площині відеозображення і які шари, що відносяться до цієї області, містять код блоків коефіцієнтів, необхідних для відновлення даної області.

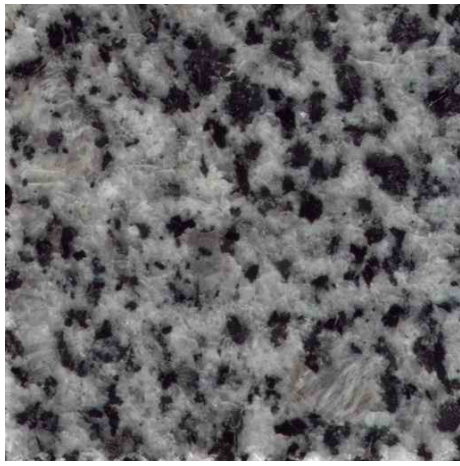
Безумовно, таке представлення відеозображення не може бути вигідним з погляду ефективності стиснення. Дійсно, зі зменшенням розміру структурних елементів ефективність стиснення дещо знижується. Стандарт методу JPEG-2000 у цьому випадку залишає вибір: з одного боку, ми маємо можливість одержувати інформаційні представлення, що дозволяють досить швидко вилучати і редагувати частини відеозображення, з іншого боку, стандарт не перешкоджає створенню інформаційних представлень, ефективних по об'єму стиснутих даних.

Кольорові цифрові відеозображення містить у собі відразу кілька компонентів. Найчастіше, вони складаються з трьох компонентів: червоного R , зеленого G і синього B кольору. Більшої ефективності стиснення можна досягти у випадку, коли кодуємо компоненти представлені в яскраво-колірній формі. Для перетворення відеозображення зі стандартного колірної представлення RGB у яскраво-колірне представлення $YCrCb$ (яскравість та кольорорізницеві сигнали). Стандартом методу JPEG-2000 передбачені обчислювальні процедури такого перетворення.

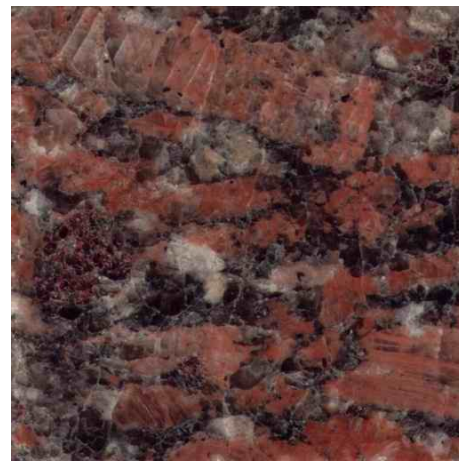
Для забезпечення завадостійкості інформаційного представлення та зручності доступу до даних в стандарті JPEG-2000 передбачена система маркерів і маркерних сегментів. Маркери відіграють роль розмежувачів інформаційного потоку; маркерні сегменти містять у собі параметри фрагментів інформації, обмежених маркерами. Дані, що починаються з маркера, як правило, можуть бути коректно проінтерпретовані без якої-небудь додаткової інформації, що забезпечує можливість часткового відновлення відеозображення, що було пошкоджено. Введення елементів завадостійкості забезпечує використання методу стиснення в телекомунікаційних додатках.

4.2. Розробка методики оцінювання похибок на стиснутих відеозображеннях

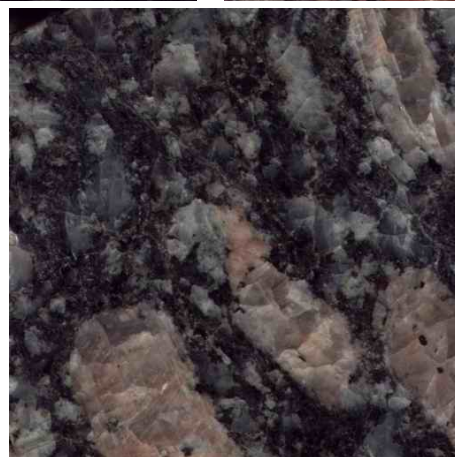
Роботу алгоритму стиснення JPEG-2000 розглянемо на конкретному прикладі, пов'язаному з стисненням цифрових відеозображень промислових виробів з природного каменю (рис. 4.6).



а) б)



в) г)



д)

Рис. 4.6. Відеозображення поверхні промислових виробів з природного каменю:
а) зображення №1; а) зображення №1; б) зображення №2;
в) зображення №3; г) зображення №4; д) зображення №5

Слід зазначити, що одним з найважливіших способів скорочення обсягу відеозображення (тобто його стиснення) є вибір оптимальних параметрів розподільчої здатності для яскравості, кольору та просторових ознак елементів відеозображення. Перевищення цих параметрів понад помітні оком людини значення приведе лише до збільшення обсягу відеозображення без поліпшення його візуальної якості. На відміну від цього, при вимірюванні ГП об'єктів потрібно обирати такі значення параметрів, що забезпечать необхідну точність вимірювальної відеоінформації.

Основним параметром алгоритму стиснення є таблиця квантування частотних коефіцієнтів, отриманих на основі вейвлет-перетворення. Змінюючи

елементи цієї таблиці, ми можемо змінювати коефіцієнт стиснення. Крім того, вона впливає і на точність відтворення відеозображення після стиснення.

Розглянемо критерії оцінювання похибок вимірювальної інформації, відновленої після стиснення. Оскільки при цьому може бути загублена інформація, що представляє інтерес, то бажано мати засоби кількісних оцінок характеру і величини втрат інформації. В основу такого визначення можуть бути покладені як суб'єктивні, так об'єктивні чисельні критерії вірності (точності) відтворення.

Основним є критерій середньоквадратичного відхилення (СКВ) різниці між початковим і відновленим відеозображеннями. Нехай $f(x, y)$ означає початкове відеозображення, а $\bar{f}(x, y)$ – його наближення, що отримується в результаті операцій стиснення і наступного відновлення. Для будь-яких x та y помилка (нев'язка) $e(x, y)$ для елементів відеозображень $f(x, y)$ і $\bar{f}(x, y)$ визначається як

$$e(x, y) = \bar{f}(x, y) - f(x, y)$$

а величина повної невязки двох відеозображень дорівнює

$$\|e(x, y)\| = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\bar{f}(x, y) - f(x, y) \right]^2,$$

де розміри відеозображення дорівнюють $M \times N$. Величина середньоквадратичного відхилення e_{CKB} різниці відеозображень $f(x, y)$ і $\bar{f}(x, y)$ буде дорівнює:

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

Іншим об'єктивним критерієм вірності відтворення є співвідношення «сигнал-шум» для відновленого відеозображення. Якщо розглядати відновлене відеозображення $\bar{f}(x, y)$ як суму початкового відеозображення $f(x, y)$ і шуму $e(x, y)$, доданого в результаті стиснення, то співвідношення «сигнал-шум» буде дорівнювати:

$$SNR_{CK} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\bar{f}(x, y) \right]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\bar{f}(x, y) - f(x, y) \right]^2}$$

4.3. Результати чисельного моделювання та експериментальних досліджень вейвлет-стиснення цифрових відеозображень

Стиснення тестових відеозображень (рис. 4.6) виконувалося за допомогою безкоштовної версії програми ACDSee. Використовувався метод стиснення JPEG 2000, заснований на wavelet-перетворенні.

Було обрано декілька рівнів стиснення з коефіцієнтами стиснення (відношенням об'єму цифрових даних початкового відеозображення до об'єму стиснутих даних) 100:1, 50:1, 25:1, 10:1, 8:1, 4:1, 2:1 та 1:1. При стисненні 100:1

візуально якість відеозображення значно погіршувалась, тому такий режим був прийнятий як режим максимального стиснення, який варто використовувати для подібних відеозображень. Стиснення 1:1 було прийнято до розгляду для оцінки максимально можливої якості, яка зберігається при вейвлет-перетворення цифрових відеозображень.

Стиснення проводилося для оригінальних кольорових відеозображень та для їх напівтонових (відтінки сірого) еквівалентів.

Для відновлених після стиснення відеозображень було виконано аналіз на наявність похибок відтворення кольору та геометричних параметрів об'єктів.

Похибки відтворення кольору об'єктів

Похибки відтворення кольору можна розглядати як загальну міру відхилення значень кольору стиснутого відеозображення відносно початкового:

$$\Delta = \sum_{i=1}^3 k_i \Delta_i,$$

де $i = 1, 2, 3$ – номер каналів для передачі кольору (в даному випадку: R (червоний), G (зелений), B (синій)), відповідно $k_1 = 0.3$, $k_2 = 0.59$, $k_3 = 0.11$. Значення Δ_i обчислюються за формулою:

$$\Delta_i = \sqrt{\frac{1}{M \cdot N} \sum_{x=1}^M \sum_{y=1}^N (Y_{vost.}(x, y, i) - Y_{orig.}(x, y, i))^2},$$

де $Y_{vost.}(x, y, i)$, $Y_{orig.}(x, y, i)$ – показники яскравості кольору i у точці (x, y) відновленого та початкового відеозображень відповідно.

Похибки відтворення кольору об'єктів обчислювалися для кольорових відеозображень та їх напівтонових еквівалентів окремо. Основною відмінністю між цими підрахунками було те, що у кольорового відеозображення 3 кольорових канали, а у напівтонового 1.

Текст програми для підрахунку похибки відтворення кольору об'єктів для кольорового відеозображення:

```
file_original = 'st5_512x512.bmp'
file_vost = 'st5_512x512_4.bmp'
inf_original = imfinfo(file_original);
inf_vost = imfinfo(file_vost);
image_original = imread(file_original);
image_vost = imread(file_vost);

koef = [0.3 0.59 0.11]

delta = 0;

for color_channel = 1:3
    channel_original = double(image_original(:, :, color_channel));
    channel_vost = double(image_vost(:, :, color_channel));
    channel_quad_summ = 0;
    for i = 1:inf_original.Width
        for j = 1:inf_original.Height
            channel_quad_summ = channel_quad_summ +
                (channel_vost(i, j) - channel_original(i, j))^2;
        end
    end
end
```



```

end
deltaT =
sqrt(channel_quad_summ/(inf_original.Width*inf_original.Height))
delta = delta + koef(color_channel) * deltaT;
end

```

В змінні `file_original`, `file_vost` заносимо назви відеозображень, які дсліджуємо. В даному прикладі це `st5_512x512.bmp` (файл початкового відеозображення), `st5_512x512_4.bmp` (файл відеозображення, стиснутого з коефіцієнтом 4:1).

```

inf_original = imfinfo(file_original);
inf_vost = imfinfo(file_vost);

```

В `inf_original` та `inf_vost` заносимо інформацію про відеозображення, використовуючи функцію `imfinfo()`. До цієї інформації належать: назва файлу, дата останнього редагування, розмір файлу у байтах, формат відеозображення, версія формату, висота, ширина, глибина кольору (кількість байт для зберігання кольору одного пікселя) та тип кольору.

```

image_original = imread(file_original);
image_vost = imread(file_vost);

```

За допомогою функції `imread()` читаємо безпосередньо самі початкове та відновлене відеозображення і записуємо отримані дані у змінні `image_original`, `image_vost`.

Далі в циклі знаходимо значення виразу для похибки $\sum_{x=1}^M \sum_{y=1}^N (A_{vost.}(x,y,i) - A_{orig.}(x,y,i))$ та поміщуємо його у змінну `channel_quad_summ`.

```

for i = 1:inf_original.Width
    for j = 1:inf_original.Height
        channel_quad_summ = channel_quad_summ +
(channel_vost(i,j) - channel_original(i,j))^2;
    end
end

```

Тут `channel_vost` та `channel_original` – це двовимірні масиви, елементами яких є значення для кожного пікселя відеозображення відповідного кольору (червоного, зеленого чи синього). Їх розмірність рівна розмірам відеозображення.

```

deltaT =
sqrt(channel_quad_summ/(inf_original.Width*inf_original.Height))

```

В змінну `deltaT` записуємо значення похибки для поточного кольору.

```

delta = delta + koef(color_channel) * deltaT;

```

В `delta` записуємо загальне значення похибки відтворення кольору об'єктів.

Приклад результат виконання програми:

```

>>
file_original =
st5_512x512.bmp

```

```

file_vost =
st5_512x512_4.bmp
koef =
    0.3000    0.5900    0.1100
deltaR =
    0.9533
deltaG =
    0.8644
deltaB =
    1.0118
delta =
    0.9072

```

Три значення deltaR , deltaG , deltaB відповідають похибкам для червоного, зеленого та синього кольорів відповідно, delta – загальне значення похибки відновленого кольорового відеозображення, deltaY – значення похибки відновленого напівтонового відеозображення.

За допомогою цієї програми було знайдено значення похибки відтворення кольору об'єктів для усіх п'яти відеозображень (див. рис. 4.6), стиснутих з усіма наведеними коефіцієнтами. Далі було побудовано графіки залежності похибки відтворення кольору об'єктів від коефіцієнту стиснення (рис. 4.7).

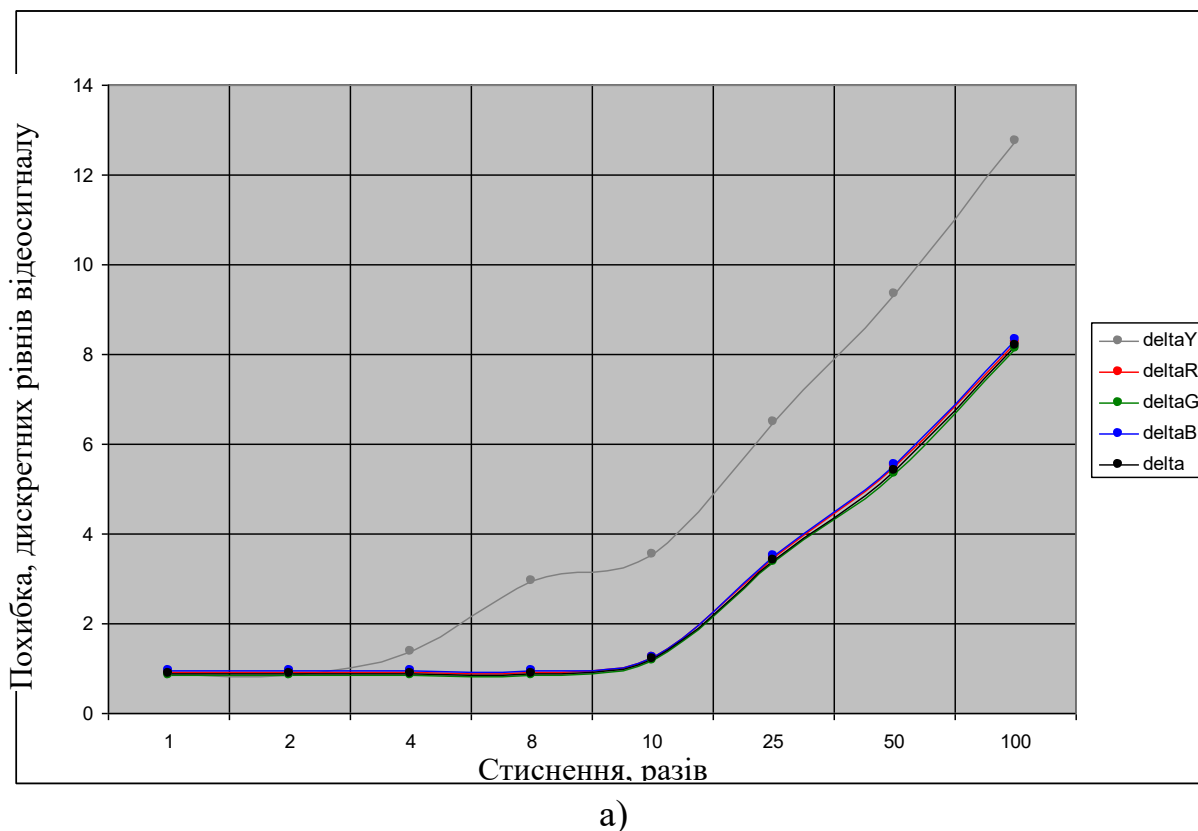
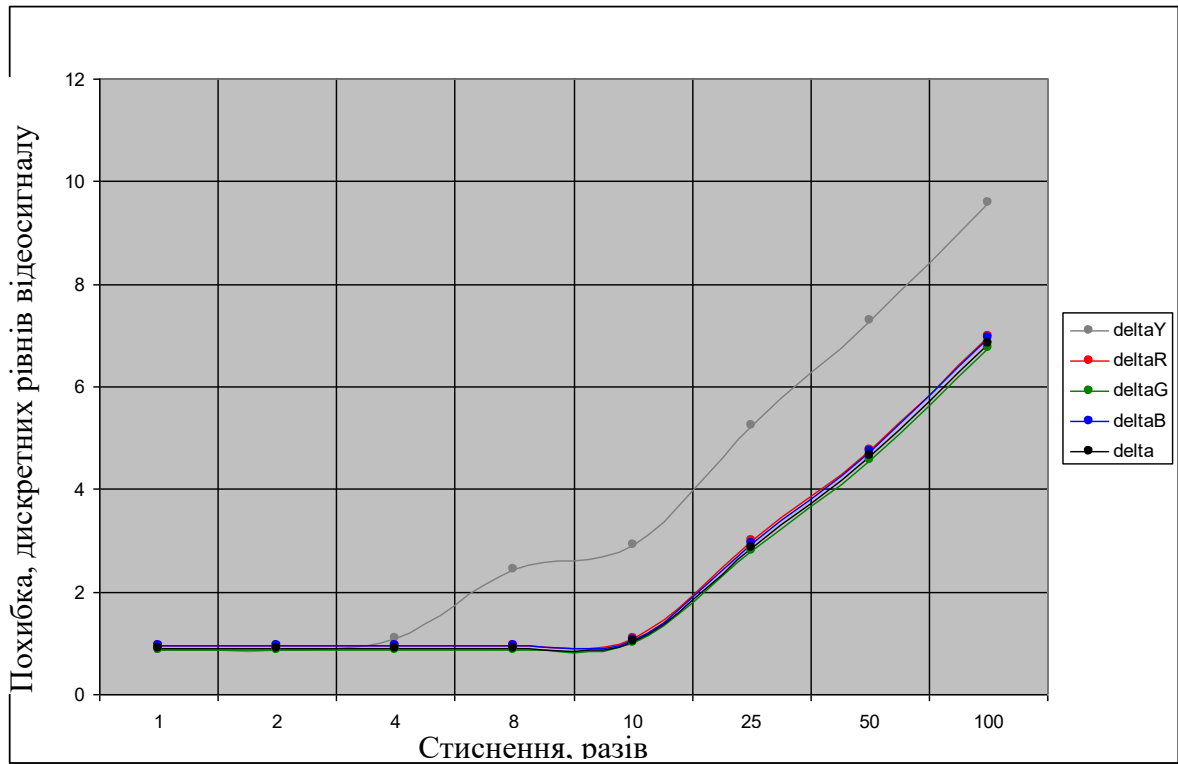
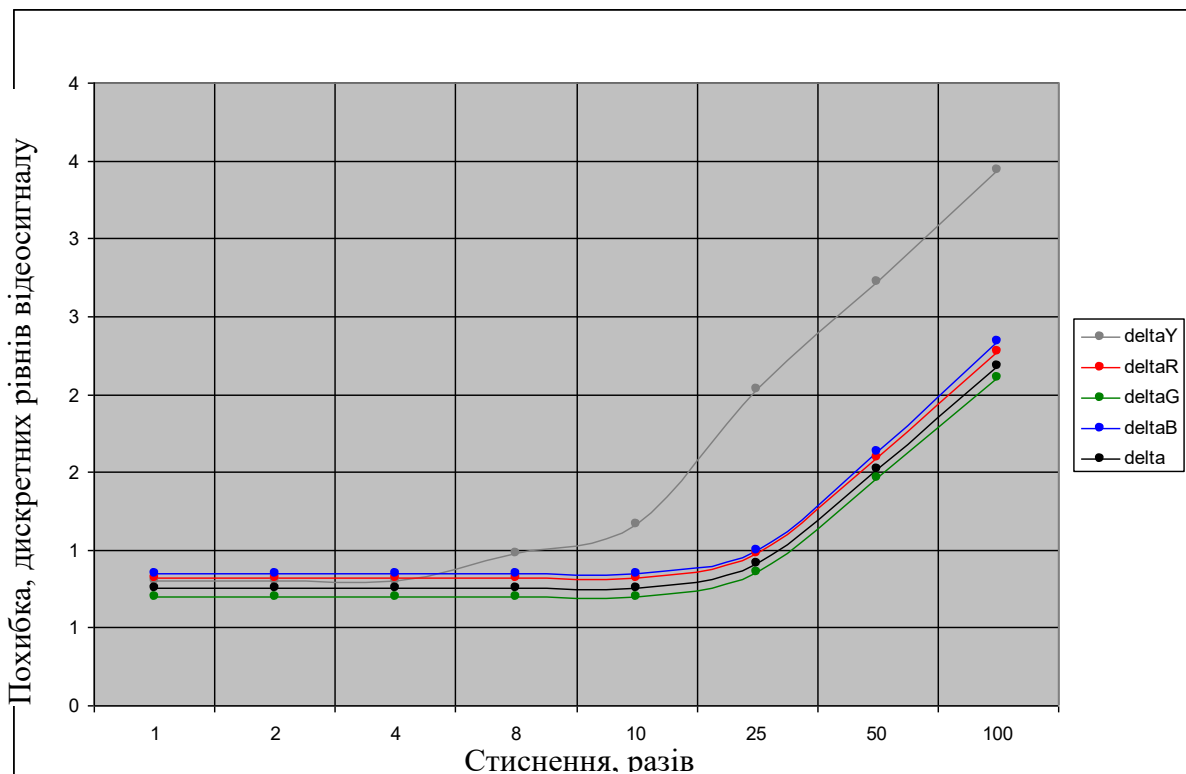


Рис. 4.7. Похибки відтворення кольору об'єктів в залежності від величини стиснення: а) відеозображення №1; б) відеозображення №2; в) відеозображення №3; г) відеозображення №4; д) відеозображення №5; deltaR , deltaG , deltaB – відповідають похибкам для червоного, зеленого та синього кольорів відповідно, delta – загальне значення похибки відновленого кольорового відеозображення, deltaY – значення похибки відновленого напівтонового відеозображення

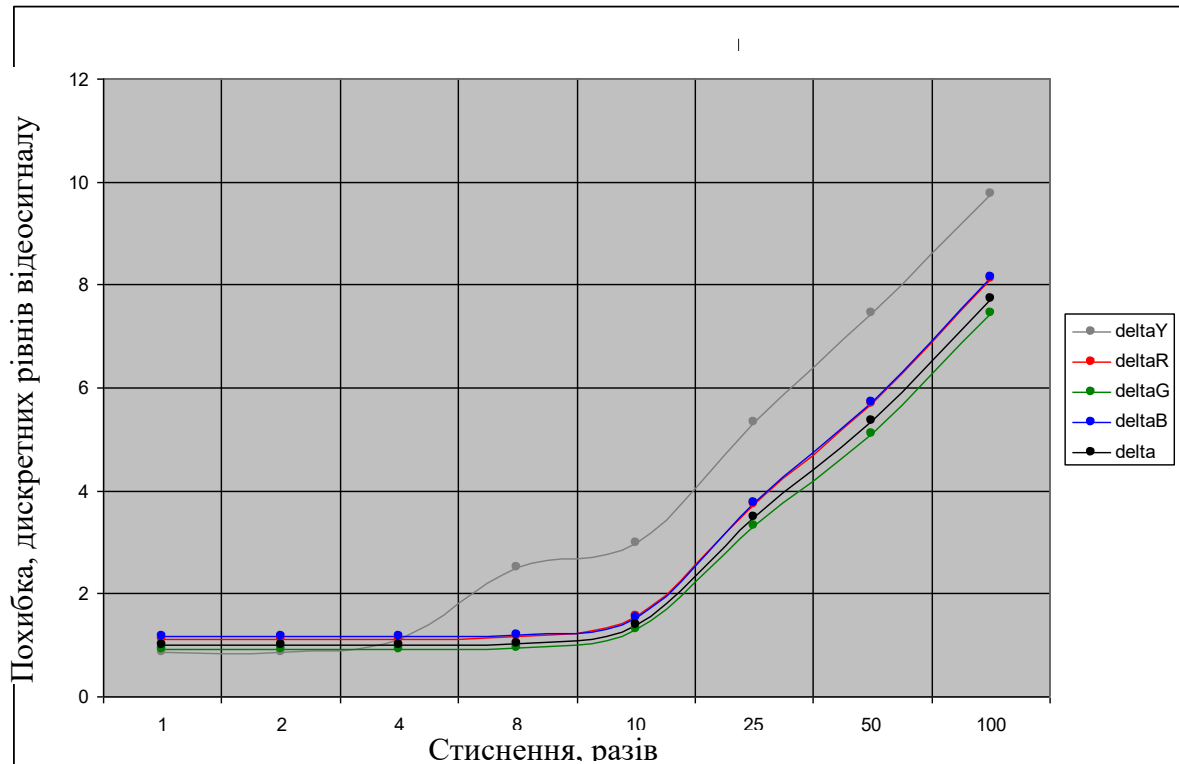


б)

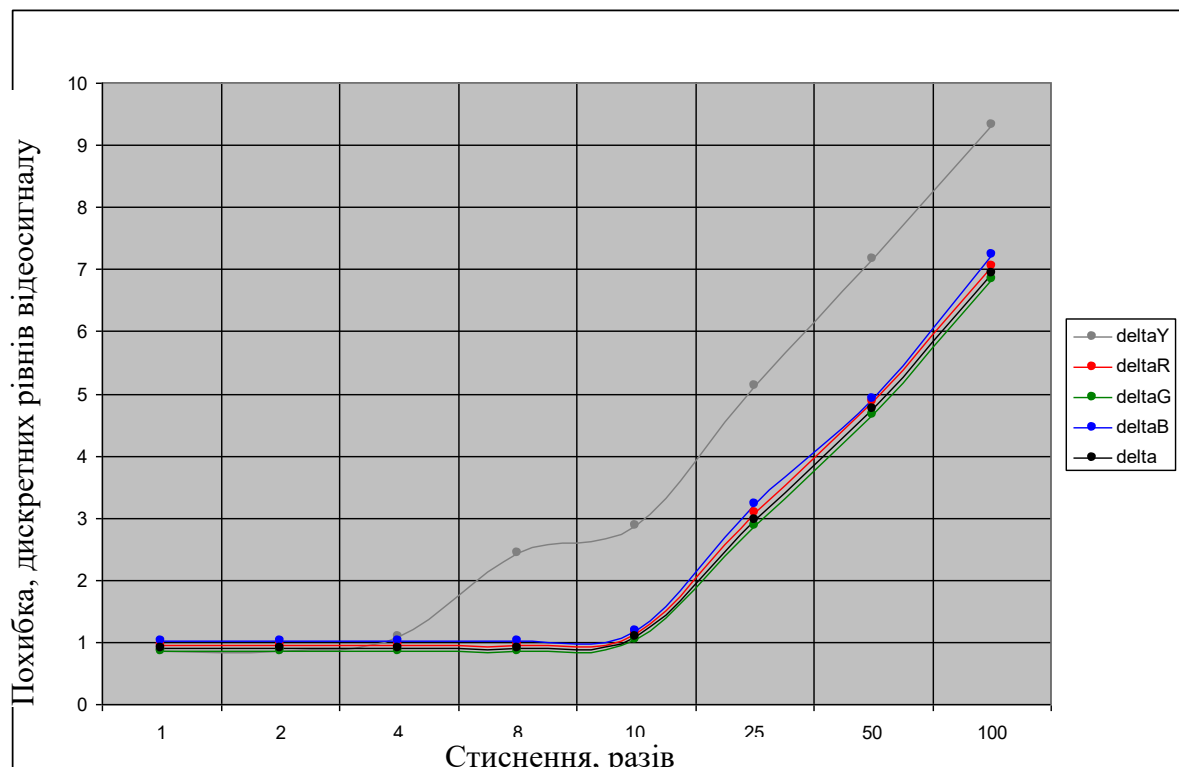


в)

Рис. 4.7 (продовження)



Г)



Д)

Рис. 4.7 (продовження)

На основі отриманих результатів можна зробити висновок про ефективність застосування вейвлет-перетворення та відповідного методу стиснення до цифрових відеозображень з вимірювальною інформацією про ГП

та параметри руху об'єктів. При стисненні кольорових та напівтонових відеозображень з коефіцієнтом 10:1 похибки майже непомітні. Прийнятним коефіцієнтом стиснення для вимірювань ГП є значення 50:1.

Похибки координат об'єктів на відновлених відеозображеннях

Оцінка похибок координат основана на аналізі ділянок відеозображення, дискретні точки яких мають однакові ознаки. Ці ділянки відповідають об'єктам вимірювань та їх отримано шляхом сегментації тестових відеозображень. Ознакою сегментації була величина яскравості дискретних точок цифрового відеозображення. Для відеозображень №1, №2, №4, №5 (див. рис. 4.6) точки, для яких яскравість належить діапазону [128, 255] дискретних рівнів відеосигналу, вважались світлими, інші з проміжку [0, 127] дискретних рівнів – темними. Для відеозображення №3 такими діапазонами були [51, 255] та [0, 50], що пояснюється низькою яскравістю самого відеозображення.

Міра координатного викривлення – це величина, яка знаходиться за формулою:

$$\Delta_{coord.} = \sqrt{\frac{1}{L} \sum_{i=1}^L (R_{compr.}(i) - R_{orig.}(i))^2},$$

де $R_{compr.}(i)$, $R_{orig.}(i)$ – координат контурних точок об'єкта, L – сумарна кількість контурних точок.

В даному випадку було обрано 20 точок, 10 з яких досліджувались на зміщення координати x , інші 10 – на зміщення координати y (рис. 4.8).

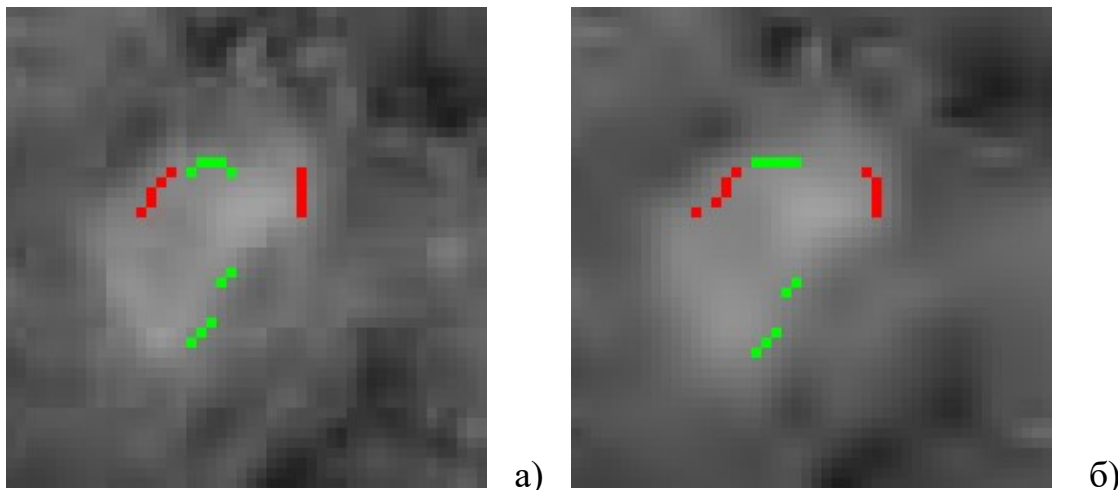


Рис. 4.8. Зміщення контурних точок при стисненні напівтонового еквівалента відеозображення №2: а) початкове відеозображення, б) відновлене відеозображення після стиснення в 50 разів

Похибка $\Delta_{coord.}$ знаходилася за допомогою наступної програми:

```
dx_dy = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0]
L = 20
summ = 0;
for i = 1:L
    summ = summ + (dx_dy(i))^2;
end
```

```
deltaCoord = sqrt(summ/L)
```

В масиві dx_dy зберігаються зміщення координат контурних точок.

В циклі

```
for i = 1:L  
    summ = summ + (dx_dy(i))^2;  
end
```

знаходиться значення виразу $\sum_{i=1}^L (R_{compr.}(i) - R_{orig.}(i))^2$, що присутній в формулі для розрахунку похибки визначення координат.

deltaCoord = sqrt(summ/L) – середньоквадратична похибка визначення координат контурних точок об'єкта.

Результат виконання програми:

```
>>
```

```
L =
```

```
    20
```

```
deltaCoord =
```

```
    0.2236
```

```
>>
```

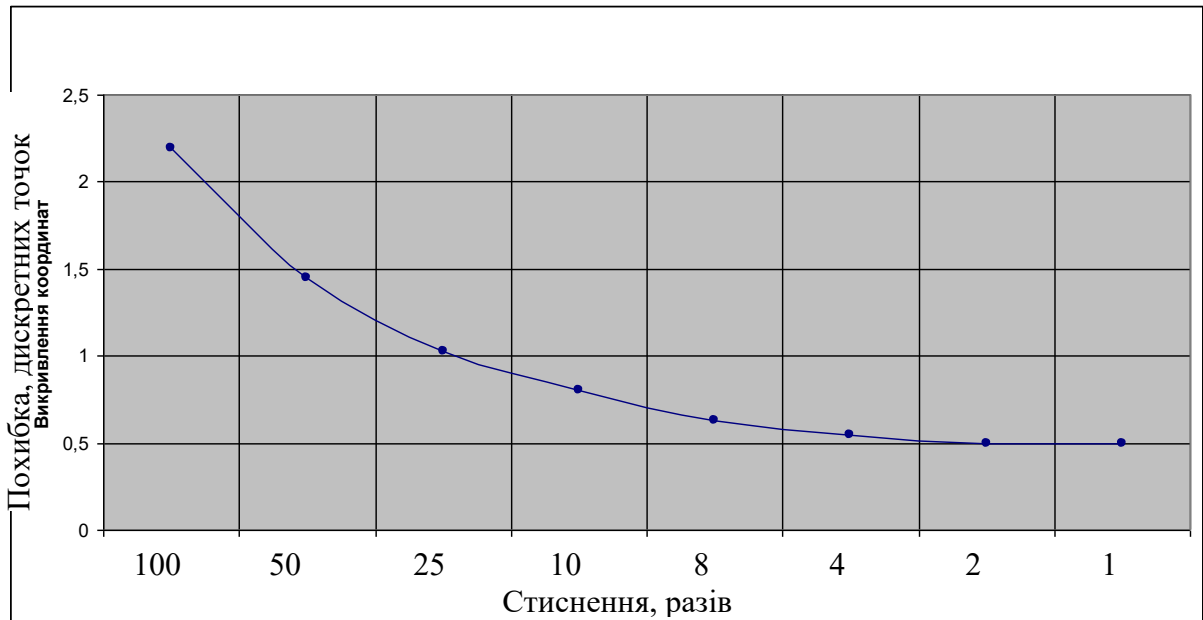
Так, було знайдено середньоквадратичні похибки координат контурних точок об'єктів для усіх напівтонових відеозображень з усіма коефіцієнтами стиснення.

Також було оцінено похибку визначення площі об'єктів на відеозображеннях, відновлених після стиснення:

$$\Delta_S = S_{compr.} - S_{orig.},$$

де $S_{compr.}$, $S_{orig.}$ – площі об'єктів на відновленому та початковому відеозображеннях. На практиці $S_{compr.}$, $S_{orig.}$ визначаються як кількість дискретних точок відеозображення, що належать об'єкту та знайдені шляхом сегментації.

На основі отриманих даних побудовано графіки залежності похибок визначення координат контурних точок та площі об'єктів в залежності від величини стиснення цифрового відеозображення (рис. 4.9, 4.10).



а)

Рис. 4.9. Похибки визначення координат контурних точок об'єктів в залежності від величини стиснення: а) відеозображення №1; б) відеозображення №2; в) відеозображення №3; г) відеозображення №4; д) відеозображення №5