

УДК 004

В.Мікіньов, магістр гр. КН-22М-2*Центральноукраїнський національний технічний університет*

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СТИСНЕННЯ РАСТРОВИХ ЗОБРАЖЕНЬ БЕЗ ВТРАТИ ЯКОСТІ

У статті розроблено програмне забезпечення, яке призначено для системи стиснення растрових зображень без втрати якості. Метою розробки є дослідження та програмна реалізація системи стиснення растрових зображень без втрати якості. Об'єктом дослідження є процес стиснення растрових зображень без втрати якості. Предметом дослідження є методи стиснення растрових зображень без втрати якості. Методи дослідження базуються на методах комп'ютерної графіки, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи стиснення растрових зображень без втрати якості. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. Зображення важливі для веб-сайтів, документів і особистих колекцій. Однак нестиснуті файли можуть мати великі розміри, спричиняючи повільне завантаження, проблеми зі зберіганням і труднощі з обміном ними в Інтернеті. Тут на допомогу приходить стиснення зображення. Стиснення зображення зменшує його розмір, не впливаючи на якість зображення. Використовуючи різні методи, він відкидає зайві дані у файлі зображення. Це призводить до меншого розміру, який займає менше місця для зберігання, швидше завантажується на веб-сторінках і легше ділитися електронною поштою або в соціальних мережах. У цій публікації ми розглянемо різні інструменти та методи, доступні для стиснення зображень. Таким чином ви можете оптимізувати своє цифрове зображення без шкоди для необхідної якості.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи стиснення растрових зображень без втрати якості.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи стиснення растрових зображень без втрати якості.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем стиснення растрових зображень без втрати якості.
- Дослідження системи стиснення растрових зображень без втрати якості.
- Програмна реалізація системи стиснення растрових зображень без втрати якості.

Об'єктом дослідження є процес стиснення растрових зображень без втрати якості.

Предметом дослідження є методи стиснення растрових зображень без втрати якості.

Методи дослідження базуються на методах комп'ютерної графіки, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Працюючи над веб-сайтом, який містить високоякісні зображення, важливо враховувати його розмір. Завантаження цих великих зображень на ваш сайт займає багато часу для завантаження сторінок. Ось у чому річ. Повільне завантаження через великі зображення може призвести до того, що відвідувачі покинуть сайт, навіть не побачивши ваш вміст. Це призводить до вищого показника відмов, що є показником, який відстежує, скільки відвідувачів залишає сайт після перегляду лише однієї сторінки. Щоб

уникнути цього, найкраще зменшити розмір зображення та використовувати JPEG як формат.

Не кажучи вже про те, що зменшення розміру вашого зображення має вирішальне значення з кількох причин. По-перше, менші розміри файлів призводять до швидшого часу завантаження. Це мінімізує час очікування та продовжує зацікавленість відвідувачів вашим сайтом. Крім того, зображення зі зменшеним розміром економлять місце для зберігання. Це важливо, оскільки цифрові файли накопичуються та займають значний простір для зберігання. Крім того, файлами меншого розміру набагато легше ділитися електронною поштою або через соціальні мережі. Часто ці платформи мають обмеження щодо розміру файлів для завантаження або вкладень.

Фрактальний стиск

Ідея фрактального стиску ґрунтується саме на властивості самоподоби. Але існують дві проблеми:

- по-перше, ніщо не гарантує наявності властивості самоподоби в довільного зображення;
- по-друге, якщо навіть об'єкт i є фрактальним, як виділити ту область (або області), на основі яких будується зображення.

Отже, необхідний якийсь теоретичний базис, що дозволяє вирішувати ці проблеми. Такий базис існує й наведений нижче.

Чорно-білі зображення

Метричні простори

Метрикою d у просторі X називається функція двох аргументів $d(x,y)$ така, що:

$$1) d(x,y) = d(y,x)$$

$$2) d(x,x) = 0$$

$$3) d(x,y) \leq d(x,z) + d$$

Тоді (X,d) – метричний простір. Послідовність точок:

$$\{x_n\} \subset X$$

називається збіжною до точки x , якщо:

$$\forall \varepsilon > 0 \exists N > 0 : \forall n \geq N \Rightarrow d(x_n, x) < \varepsilon$$

Метричний простір, де кожна послідовність Коші сходиться до точки цього простору, називається повним метричним простором.

Точка x називається граничною точкою множини X , якщо існує послідовність точок з X , що сходиться до точки x .

Множина у метричному просторі називається замкнутою, якщо вона містить всі свої граничні точки. Множина B з (X, d) називається обмеженою, якщо існує точка x_0 з X і кінцеве значення $R > 0$, такі, що для будь-якого x з B виконується умова $d(x,x_0)$

Компактні множини й простір Хаусдорфа

Множина C у метричному просторі (X, d) називається компактною, якщо кожна нескінченна послідовність із C має збіжну в C підпослідовність.

Визначимо простір Хаусдорфа.

Нехай (X, d) – повний метричний простір. Визначимо $H(X)$ як простір, що складається з компактних підмножин множини X . Тобто, кожна точка в $H(X)$ – це компактна підмножина з X .

Розглянемо довільну точку x з X і довільне B з $H(X)$. Покладемо по визначенню:

$$d(x, B) = \min\{d(x, y) : y \in B\}.$$

У силу компактності множини B мінімум існує й обмежений. Розглянемо тепер довільні $A, B \in H(X)$ і визначимо відстань між ними як:

$$d(A, B) = \max\{d(x, B) : x \in A\}.$$

Компактність A забезпечує існування й кінцівка максимуму, але $d(A, B)$ не задає метрику, тому що, у загальному випадку $d(A, B) \in d(B, A)$.

Визначимо нову міру відстані $h(A, B)$:

$$h(A, B) = \max\{d(A, B), d(B, A)\}.$$

Тепер ми одержали метрику в просторі $H(X)$. Метрика h називається метрикою Хаусдорфа, а метричний простір $(H(X), h)$ – метричним простором Хаусдорфа.

Стискаючі відображення

Перетворення $f: X \in X$ у метричному просторі (X, d) називається стискаючим відображенням, якщо існує константа s , $0 \in s < 1$ така, що:

$$d(f(x_1), f(x_2)) \in s * d(x_1, x_2) \text{ для всіх } x_1, x_2 \in X.$$

Константа s називається коефіцієнтом стиску відображення f . Точка x_0 називається нерухливою точкою (аттрактором) стискаючого відображення f , якщо $f(x_0) = x_0$.

Теорема про стискаючі відображення

Нехай $f: X \in X$ стискаюче відображення на повному метричному просторі (X, d) . Тоді f має одну й тільки одну нерухливу точку $x_0 \in X$, і для будь-якого:

$$x_0 \in X f_n(x) \in x_0.$$

Системи ітеруємих функцій

Нехай $\{w_1, \dots, w_N\}$ – кінцевий набір стискаючих відображень в (X, d) з коефіцієнтами стиску s_1, \dots, s_N ... Визначимо відображення W , на компактні множини точок з X у такий спосіб:

$$W(B) = w_1(B), \dots, w_N(B) \text{ для кожного } B \in H(X).$$

У такий спосіб одержали стискаюче відображення $W: H(X) \in H(X)$ з коефіцієнтом стиску:

$$s = \max\{s_1, \dots, s_N\} \dots$$

Система ітеруємих функцій (IFS) складається з повного метричного простору (X, d) і кінцевої множини стискаючих відображень $w_n: X \in X$ з коефіцієнтами стиску s_n . Коефіцієнт стиску IFS визначається як:

$$s = \max\{s_1, \dots, s_N\} \dots$$

IFS будемо позначати як $\{X, w_n: n = 1, 2, \dots, N\}$...

Теорема колажу. Нехай L – точка простору $H(X)$. Задано деяке $e > 0$. Виберемо IFS $\{X, w_n: n = 1, 2, \dots, N\}$ з коефіцієнтом стиску s , $0 < s < 1$, так щоб $h(L, W(L)) \leq e$, Тоді $h(L, A) \leq h(L, W(L)) / (1-s) \leq e / (1-s)$, де A – аттрактор даної IFS.

Афінні перетворення

Афінні перетворення – перетворення виду:

$$W \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

Афінні перетворення здійснюють стиск/розтягання, перенос і поворот об'єкта.

Кодування зображень

Для одержання коефіцієнтів потрібного афінного перетворення (у найпростіших випадках) досить вирішити 2 системи їх 3-х рівнянь із 3-ма невідомими. Як приклад розглянемо побудову Трикутника Серпинського. Це зображення описується 3-ма перетвореннями:

- 1-е переводить точки $\{1, 2, 3\}$ у точки $\{1, 4, 6\}$;
- 2-е: $\{1, 2, 3\}$ – в $\{4, 2, 5\}$;
- 3-е: $\{1, 2, 3\}$ – в $\{6, 5, 3\}$.

Унизу праворуч показане зображення трикутника Серпинського, отримане за допомогою імовірнісного алгоритму й перетворень знайдених зазначеним вище способом.

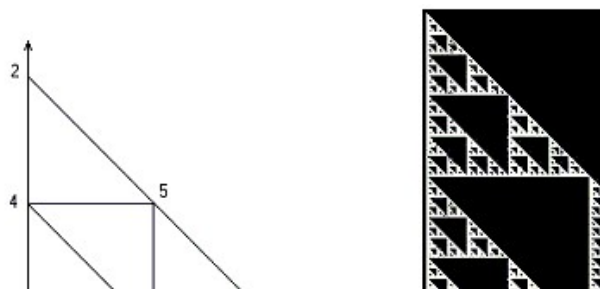


Рисунок 1 – Трикутник Серпинського

Декодування зображень

Детерміністичний алгоритм

Детерміністичний алгоритм для побудови зображення, що є аттрактором IFS, прямо застосовує теорему про стискаюче відображення до будь-якого початкового зображення B з $H(X)$. Алгоритм будує послідовність зображень A_n , багаторазово застосовуючи IFS відображення $W = \{w_1, \dots, w_N\}$. Якщо ми покладемо $A_0 = B$, то процес може бути записаний у вигляді $A_n = W(A_{n-1})$. По теоремі про стискаюче відображення, A_n сходиться до аттрактору даного IFS. Нижче наведений приклад роботи детерміністичного алгоритму – перші кілька ітерацій і кінцеве зображення, близьке до аттрактору.



Рисунок 2 – Приклад роботи детерміністичного алгоритму

Імовірнісний алгоритм

У той час як детерміністичний алгоритм є прямим застосуванням теореми про стискаючі відображення, що дозволяє спостерігати, як він діє на практиці, цей алгоритм виявляється занадто повільним і звичайно не використовується на практиці для побудови зображень – аттракторів.

Більше кращим є використання імовірнісного алгоритму: Імовірнісний алгоритм зв'язує з кожним перетворенням w_i з IFS імовірність p_i . Ці ймовірності визначають, наскільки щільно кожна частина зображення – аттрактора покрита точками. Імовірності перетворень можна обчислювати як відношення модуля визначника основної матриці перетворення до суми модулів визначників основних матриць всіх перетворень із IFS:

$$w_i(\mathbf{x}) = w_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} = T_i \mathbf{x} + \mathbf{t}_i,$$

$$p_i = \frac{|\det T_i|}{\sum_{j=1}^N |\det T_j|}$$

Алгоритм побудови зображення:

```

For n = 1 to number_of_number_of__in_image do
  (x,y) = random(B)
  For I=1 to number_of_ iterations do
    p = random(0,1)
    (x,y) =  $W_k(x,y)$  //де ймовірність p відповідає перетворенню  $W_k$ 
  end
end

```

Нижче наведений приклад роботи імовірнісного алгоритму.



Рисунок 3 – Приклад роботи імовірнісного алгоритму

Можна помітити, що для імовірнісного алгоритму існують особливі точки – чорні зображення. Якщо робота імовірнісного алгоритму починається з них, то аттрактор не буде отриманий ніколи. Звідси випливає наступне твердження: якщо в початковому зображенні присутня хоча б одна чорна точка, то існує відмінна від нуля ймовірність того, що в результаті роботи імовірнісного алгоритму вийде чисто чорне зображення, причому ця ймовірність тим більше, чим більше кількість чорних точок у зображенні.

Дійсно, нехай зображення має розміри $n \times m$. Тоді загальне число точок у ньому $N = n \times m$. Нехай число чорних точок дорівнює b , нехай число точок, використовуваних для побудови зображення дорівнює M . Ймовірність того, що випадково обрана точка виявиться чорної, дорівнює b/N .

Ймовірність того, що всі обрані точки виявляться чорними, дорівнює:

$$\left(\frac{b}{N} \right)^M$$

Таким чином, для того, щоб алгоритм спрацював коректно з ймовірністю більшої, ніж $1-e$, де $e \in (0, 1)$, потрібне виконання нерівності:

$$\log_{b/N} e \in M.$$

Розглянемо приклад. Нехай $e = 0,01$. Тоді для коректної роботи імовірнісного алгоритму при початковому зображенні розміру 300×300 з єдиною білою точкою потрібно $M > \log_{89999/90000} 0,01 > 414463$.

Цікаво, що, хоча імовірнісний алгоритм широко використовується в різних наукових працях, ніхто з авторів жодного разу не звернув уваги на даний факт.

Зображення в градаціях сірого**Метричний простір**

Зображення в градаціях сірого можна розглядати як речовинні функції $f(x,y)$, визначені на одиничному квадраті:

$$I^2 = I \times I.$$

На цих функціях можна ввести метрику в такий спосіб:

$$d_2(f, g) = \left[\iint |f(x, y) - g(x, y)|^2 \right]^{1/2}.$$

Нехай F – простір речовинні функції, інтегровальних із квадратом на I^2 з уведеною метрикою. Тоді (F, d) – повний метричний простір і в ньому виконується теорема про стискаючі відображення.

Оскільки працювати ми будемо із цифровими зображеннями, які по суті є матрицями фіксованих значень функції $f(x,y)$, узятих у фіксованих точках (x_i, y_j) , у цьому випадку можна користуватися середньоквадратичною метрикою (rms):

$$d_{rms}(f, g) = \left[\sum_{i=1}^n \sum_{j=1}^m |f(x_i, y_j) - g(x_i, y_j)|^2 \right]^{1/2}.$$

PIFS і афінні перетворення зображень у градаціях сірого

У цьому розділі використовуються IFS спеціального виду – системи ітеруємих кусочно-визначених функцій (PIFS). PIFS складається з повного метричного простору X , набору підобластей D_i з X і набору стискаючих відображень $w_i : D_i \in X, i=1, 2, \dots, n...$

Афінні перетворення

Нехай w_{0i} – афінне перетворення, що переводить у себе одиничний квадрат I^2 . $w_{0i}(x,y) = A_i(x,y)T + b_i$, де A_i – деяка матриця розміру 2×2 , b_i – вектор розміру 2×1 . Нехай D_i – деяка підобласть I^2 , і нехай R_i – область значень перетворення $w_{0i} : w_{0i}(D_i) = R_i$. Визначимо відображення $w_i : F \in F$, що діє на зображення $f(x,y)$:

$$w_i(f(x, y)) = s_i \cdot f(w_{0i}^{-1}(x, y)).$$

(якщо w_{0i} – оборотно й (x,y) з R_i).

Константа s_i управляє контрастністю, o_i управляє яскравістю. w_i – це базове афінне перетворення зображень у градаціях сірого.

Стискаючі відображення

Для того, щоб відображення w_i було стискаючої зажадаємо виконання умови:

$$d_2(w_i(f), w_i(g)) \leq s \cdot d_2(f, g).$$

Теорема про стискаюче відображення

Нехай $\{R_i\}$ утворять покриття множини I^2 , і при цьому попарно не перетинаються.

Нехай v_i – PIFS виду: $v_i : D_i \in R_i$ для деякої множини доменних областей D_i . Для кожного v_i визначимо відповідний стиск w_i на просторі зображень F зазначеним вище способом. Нехай $W(f(x,y)) = w_i(f(x,y))$ для (x,y) з R_i . Тоді W є стиском на F , має єдину нерухливу точку f і $W_n(f) \in f$ при $n \in I^2$ для будь-яких зображень f .

Теорема колажу

Нехай задане зображення в градаціях сірого f . Нехай W – стискаюче відображення таке, що $d_2(f, W(f)) \leq s$. Тоді $d_2(f, f) \leq s/(1-s)$, де s – коефіцієнт стиску відображення W , f_w – аттрактор PIFS.

Фрактальне кодування

При фрактальному кодуванні зображень ми намагаємося знайти множини стискаючих відображень, які відображають доменні блоки в множини рангових блоків.

Базовий алгоритм кодування виглядає в такий спосіб:

1. Розбиваємо вихідне зображення на непересічні рангові блоки.

2. Покриваємо зображення послідовністю доменних блоків, що можливо перетинаються (домени можуть бути різних розмірів і звичайно їхня кількість обчислюється тисячами).

3. Для кожного рангового блоку знаходимо домен і відповідне перетворення, що щонайкраще покриває ранговий блок

4. Якщо досить точної відповідності не одержуємо, то розбиваємо рангові блоки на менші. Продовжуємо цей процес доти, поки не доможемося прийнятної відповідності, або розмір рангових блоків не досягне деякого заздалегідь певної межі.

Декодування зображень

Зображення декодується шляхом ітеративного застосування перетворення W до довільного початкового зображення g , де $W(g(x,y)) = w_i(g(x,y))$, для (x,y) з R_i . Якщо перетворення були обрані коректні, то ітерація $W_n(g)$ буде близька до вихідного значення f при деякому прийнятному значенні n . Важливо, що відповідно до теореми про стискаюче відображення процес буде сходиться незалежно від вибору початкового зображення.

Розробка структурної схеми

Структурна схема системи наведена на рисунку 4. З нього видно, що система складається з наступних структурних блоків:

- Вхідне зображення.
- Блок параметрів стиснення.
- Блок стиснення за допомогою фракталів.
- Кодер.
- Блок запису у файл стисненого зображення.
- Блок зберігання стисненого зображення.
- Блок читання з файлу.
- Декодер.
- Блок декомпресії за допомогою фракталів.
- Вихідне зображення.



Рисунок 4 – Структурна схема системи

Перед тим, як перейти до докладного опису алгоритмів фрактального кодування, коротко перелічимо особливості фрактального кодування:

– У технології фрактального кодування закладений великий потенціал, але вона не стандартизована.

- Відноситься до методу стиску із втратами (дані не відновлюються у вихідному виді).
- При стиску використовуються системи ітеруємих функцій.
- Компресія даних повільна, декодування – швидке.
- Ми можемо вибрати великий діапазон значень дозволу зображень, досягаючи теоретично високих показників стиску.
- Технологія запатентована.

На сьогодні відомо кілька основних алгоритмів фрактального кодування – базовий алгоритм, алгоритм кодування FE (з виділенням характеристичних особливостей і ряд інших алгоритмів).

Зображення (у градаціях сірого) інтерпретуються як речовинні функції $f(x,y)$, певні на одиничному квадраті $I^2 = I \times I$. Тобто, можна записати, що

$$f : I^2 \rightarrow \{1, 2, \dots, N\},$$

тут N – число відтінків сірого. Взагалі, робота із зображеннями представляється особливо трудомісткою, коли говоримо про кольорові зображення, оскільки в них точка представляється у вигляді сукупності трьох основних квітів (червоних, зелених і синього) або їхніх похідних. У зображеннях у градаціях сірого кожна точка характеризується відтінком сірого, яскравістю (діапазон від 0 до 255). Тому спочатку розглядають роботу з напівтоновими зображеннями, а потім переходять до кольорових зображень.

Можна ввести метрику на цих функціях (тому що ми працюємо з метричними просторами) у такий спосіб:

$$d_2(f, g) = \left(\int_{I^2} |f(x, y) - g(x, y)|^2 dx dy \right)^{1/2}.$$

Якщо функції безперервні, але оскільки ми будемо працювати з повним метричним простором F , певним на одиничному квадраті, а зображення є цифровими, то інтеграл убирається, співвідношення (9) записується в наступному вигляді:

$$d_{rms} = \left[\sum_{i=1}^n \sum_{j=1}^m |f(x_i, y_j) - g(x_i, y_j)|^2 \right]^{1/2}.$$

Цифрові зображення являють собою матрицю фіксованих значень функції $f(x,y)$, узятих у фіксованих точках $f(x_i, y_j)$. Наведене співвідношення називається середньоквадратичним відхиленням (root mean square). Цей показник (крім стиску зображень) може використовуватися як міра мінливості значень ознак, ступеня відхилення бажаних показників від спостережуваних. Він буде використовуватися в тому числі й для оцінки ефективності кодування зображень.

У фрактальному кодуванні використовується система ітеруємих функцій, більше загального виду. Вона називається кусочно-визначена система ітеруємих функцій (PIFS). Цей тип системи ітеруємих функцій складається з повного метричного простору X , набору підобластей і набору стискаючих відображень.

Ми також можемо визначити афінні перетворення, які переводять у себе одиничний квадрат $I^2 \rightarrow I^2$ у такий спосіб:

$$\bar{w}_i(x, y) = A_i \begin{pmatrix} x \\ y \end{pmatrix} + b_i.$$

Тут A_i – матриці перетворень розміром 2×2 , а b_i – вектора зрушення (словом, тут звичайне афінне перетворення). А тепер можна записати відображення $w_i: F \rightarrow F$ у загальному виді.

$$\overline{w}_i(f)(x, y) = s_i f(\overline{w}_i^{-1}(x, y)) + o_i,$$

за умови, що перетворення \overline{w}_i оборотне й $(x, y) \in R_i$. Константа s_i розширює (або звужує) діапазон значень функції f , тобто управляє контрастністю (для зображень у градаціях сірого). Величина o_i відповідає за зміну яскравості зображення. Перетворення \overline{w}_i називається просторовою складовою перетворення w_i . Задане співвідношення є базовим для зображень у градаціях сірого, ми його будемо використовувати при стиску.

У випадку використання кусочно-визначеної системи ітеруємих функцій фіксована точка (або аттрактор) є зображенням f , для якої виконується $W(f)=f$. Теорема про стискаючі відображення говорить, що W у результаті буде зображенням, а ми зможемо порахувати послідовності $W(f_0)$, $W(W(f_0))$, $W(W(W(f_0)))$, де f_0 – яке-небудь зображення. Оскільки ми знаємо, що відображення W – стискаюче на просторі зображень, то ми в результаті одержимо єдину фіксовану точку, що є якимсь зображенням.

Припустимо, що нам необхідно закодувати зображення f . Це означає, що нам необхідно визначити набір перетворень w_1, w_2, \dots, w_N , і при цьому:

$$W = \bigcup_{i=1}^N w_i, f = x_w$$

Нам потрібно, щоб f була нерухливою точкою перетворення W . Співвідношення для фіксованих точок показує, як нам можна цього досягти:

$$f = W(f) = w_1(f) \cup w_2(f) \cup \dots \cup w_N(f).$$

Ми намагаємося знайти розбивку f на області, у результаті якого кожна область є зменшеною копією цілого зображення.

Мінімізація цього рівняння означає наступне: ми, по-перше повинні добре підібрати область D_i таким чином, щоб між областями був достатній збіг. По-друге, необхідно знайти гарні значення контрасту і яскравості (співвідношення 12), тобто значення коефіцієнтів s_i і o_i , для перетворення w_i . Для кожної області D_i ми зможемо порахувати, використовуючи аналітичні методи, значення коефіцієнтів, у такий спосіб досягається як можна менше значення величини d_{rms} .

Приведемо базовий алгоритм фрактального кодування.

1. Розбиваємо зображення f на непересічні рангові блоки $\{R_i\}$. Рангові блоки являють собою прямокутники (у найпростішому випадку квадрати), але можуть використовуватися й інші способи розбивки, наприклад, як рангові області можуть використовуватися трикутники. Блоки можуть бути однаковими, а може використовуватися адаптивна розбивка – методом квадродерева.

2. Покриваємо зображення послідовністю доменних блоків, що можливо перекриваються. Домени можуть бути різних розмірів, звичайно їхня кількість обчислюється сотнями й тисячами.

Домени можуть перекриватися, а можуть розміщатися на деякій відстані друг від друга. Доменне зображення темніше, ніж вихідне, оскільки яскравість при перетвореннях змінюється. Афінне перетворення записується в такий спосіб (для зображень у градаціях сірого):

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

Величина s_i означає яскравість пікселя й діє подібно кнопці настроювання контрасту. Коли цей параметр дорівнює 0, те пікселі домену перетворюються в чорний колір, якщо параметр дорівнює 1, то домен залишається таким же, якщо параметр установлений у межах від 0 до 1 (найчастіше беруть 0.75), чим значення більше, тим більше контраст області. Параметр o_i відповідає за світність пікселя і його зміна аналогічно зміні настроюванню яскравості. Позитивні значення роблять блок пікселів більше світлим, негативні значення – більше темним. Коли істи можливість контролювати показники контрасту і яскравості, то можна здійснити розширене афінне перетворення, що відображає доменні блоки в рангові блоки.

Матрична частина перетворення відповідає за поворот і зміну яскравості зображення (коефіцієнти A_i, b_i, c_i, d_i і s_i), а вектор $[e_i, f_i, o_i]$ за зрушення й зміну контрасту.

Зміна яскравості доменного зображення відбувається в такий спосіб: після його одержання яскравість всіх пікселів множиться на деяку величину, розповсюджене значення – 0.75.

3. Для кожного рангового блоку знаходимо домен і відповідне перетворення, що щонайкраще перекриває ранговий блок. Настроюються параметри перетворення – яскравість і контраст, що забезпечує найкращу відповідність.

При використанні даного підходу до кодування розмір доменних блоків завжди вдвічі більше, ніж розмір регіонів. Якщо ранговий блок має розмір 8×8 , то домен завжди 16×16 .

У результаті кодування у файл записується код зображення:

1. Кількість регіонів по горизонталі й вертикалі.
2. Розмір регіону.
3. Коефіцієнти афінних перетворень:
 - Координати домену.
 - Номер афінного перетворення.
 - Різницю усередненої яскравості між регіоном і доменом.

У файл, таким чином, зберігаються тільки числові коефіцієнти, а не саме зображення. Чисел досить для розпакування (виконання зворотних перетворень, при декодуванні 2 і 4 перетворення міняються місцями).

Одним із ключових параметрів є зсув доменів (домени можуть перекриватися, а можуть розташовуватися друг від друга на деякій відстані). Процес фрактального стиску полягає в пошуку самоподібних областей зображення. У цьому випадку послідовно перебираються всі регіони, і для кожного регіону підбирається найбільш схожий на нього доменний блок. Таким чином, застосовується сукупність перетворень: порівняння блоків по пікселям і афінні перетворення. Найпоширеніші наступні:

1. Поворот на 0 градусів.
2. Поворот на 90 градусів.
3. Поворот на 180 градусів.
4. Поворот на 270 градусів.
5. Симетрія щодо осі X .
6. Симетрія щодо осі Y .
7. Симетрія щодо головної діагоналі.
8. Симетрія щодо побічної діагоналі.

Від характеру вихідного зображення залежить, які перетворення можна виконувати. Можна вибрати всі перетворення, а можна – виконати тільки перетворення повороту або тільки симетрію. Але рекомендується виконати всі 8 перетворень.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів стиснення растрових зображень без втрати якості. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем стиснення растрових зображень без втрати якості; Досліджена система стиснення растрових зображень без втрати якості; На основі отриманих результатів досліджень створена програмна реалізація системи стиснення растрових зображень без втрати якості. Розроблені

під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання стиснення растрових зображень без втрати якості. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
2. Smirnov, O., Neskorođieva, T., Fedorov, E., Rudakov, K., Neskorođieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,
3. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Driev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland) Volume 22, Issue 16*, 6223, 2022.
4. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021*. P. 414-418.
5. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021*. P. 255-260.
6. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
7. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
8. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.
9. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14*.
10. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки. № 2(33)*. с. 161-172, 2019.
11. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» *Комп'ютерні науки та кібербезпека. № 4*. С. 30-37. 2019.
12. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. *Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.*
13. Smirnov, O., Kuznetsov, A., Kuznetsova., K. *Synthesis of Discrete Signals with Improved Correlation Properties*. Монографія: In.: *ISCI'2019: Information Security in Critical Infrastructures*. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
14. Смірнов О.А., Дреєва Г.М. *Метод генерування фрактального трафіку за допомогою моделі генератора на графі*. Монографія: *Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139*
15. Дреєва Г.М., Смірнов О.А., Дреєв О.М. *Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі*. *Центральноукраїнський науковий вісник. Технічні науки. № 1(32)*. с. 173-183, 2019.
16. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. *Мережні інформаційні технології*. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.
17. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. *Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах*. *Збірник наукових праць "Системи обробки інформації"*. - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.
18. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. *Комп'ютерні мережі*. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.
19. Смірнов О.А., Дреєв О.М. *Порівняння бітових щільностей при використанні різних методів кодування інформації*. *Збірник наукових праць "Системи обробки інформації"*. - Випуск 2 (118). т.2. - Х.: ХУПС -

2014. - С. 64-67

20. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції "Проблеми та перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014р. – Харків: ХНСУ. - 2014. - С. 240.
21. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лешко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.
22. Смірнов О.А., Дреєв О.М., Доренський О.П. «Дослідження впливу стиснення зображень на оперативність їх доставки у телекомунікаційній системі. Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.
23. Смірнов О.А., Доренський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.
24. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.