

V. V. Zhebka

MODELING OF THE MARKIV RANDOM FIELD FOR THE PURPOSE OF ITS FURTHER OPTIMIZATION AND APPLICATION

Models of the Markov random field are investigated. The main improvements of the Markov random field model are investigated. If we consider Markov models of random fields with binary conditional distributions, which include stochastic evolution in time, which is based on the autoregression structure for a large-scale model, these models retain the flexibility of static Markov random field models to reproduce the representation of spatial dependence in a small-scale model. Bayesian estimation in this case is achieved through the use of a so-called algorithm that requires the generation of auxiliary random fields, but does not require the use of ideal samples.

Markov random fields are a powerful tool in machine learning. It is often necessary to model such fields between dissimilar objects, which leads to the fact that the nodes in the graph belong to different types of data. To model inhomogeneous areas using graphical models, it is necessary to assign different types of distributions (binary, Gaussian, Poisson, exponent, exponential, etc.) to the model nodes.

The concept of conditional random fields is considered in the article, their features, advantages and disadvantages are established. The application of binary data in Markov models of random fields is considered, which generates a class of models of binary Markov random fields. It is established that the discrete nature of Markov random fields allows a wider range of possible values of dependence, ie negative dependence.

The model, loss function and distribution of the Markov random field function are investigated. Strengthening of Markov random fields is proposed. The pairwise exponential Markov random field is considered.

Keywords: Markov random field; optimization; loss function; distribution; pairwise exponential Markov random field.

УДК 004.7:519.87(043.3):621.391

DOI: 10.31673/2412-9070.2020.052226

Г. Я. Кіс, аспірант,

Державний університет телекомунікацій, Київ

ПІДХІД ДО ФРАКТАЛЬНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ

Запропоновано удосконалений фрактальний підхід до стиснення зображень. Розглянуто таку проблему фронтальної компресії, як повільна швидкість стиснення. Проаналізовано кілька наявних методів її вирішення. Ґрунтуючись на базових алгоритмах розпізнавання зображень та комп'ютерного бачення, а саме на побудові особливих точок зображень, обчисленні дескрипторів та методів пошуку найближчого сусіда в багатовимірному просторі, розроблено схему пошуку подібних частин зображення та запропоновано алгоритм стиснення. Цей алгоритм має складність $O(N \log N)$ за кількістю пікселів зображення, але якщо зафіксувати кількість доменних блоків, можливо досягти майже лінійної складності. Подано обґрунтування на основі попередніх досліджень щодо можливості такого зменшення. Було створено програму для перевірки цього алгоритму. Надано результати чисельного експерименту та його аналіз. Запропонований підхід придатний для стиснення як зображень, так і відео. Підхід має перспективи для подальшого підвищення швидкості та якості стиснення та потребує більш докладного вивчення.

Ключові слова: фрактальне стиснення зображень; метод комп'ютерного зору.

ВСТУП

Фрактальне стиснення зображень та відео [1] є альтернативою для досягнення високого ступеня стиснення з реконструйованим вихідним відео вищої якості, ніж наявні стандарти відео (MPEG, H.263, H.264). Нещодавно запропоновані схеми, наприклад [2], перевищують протокол H.264 за швидкістю та ступенем стиснення. Підхід засновано на ідеї безпошукового фрактального стиснення зображень [3]. Деякі властивості зображень, зокрема краї об'єктів та однорідні ділянки, не змінюються після масштабування. Цей тип надмірності ігнорується трансформаційними кодерами. Локальна масштабна інваріантність є однією з основних властивостей фрактального кодування, яка дуже корисна для деталізації зображень. Завдяки високій компресії та масштабній інваріантності корпорація Майкрософт прийняла його для стиснення зображень у своїй мультимедійній енциклопедії Encarta.

Метою цієї статті є розгляд фрактального стиснення зображень та подальше вдосконалення цього підходу на основі методів комп'ютерного зору.

ОСНОВНА ЧАСТИНА

Як відомо, стиснення даних за своєю суттю є проблемою оптимізації: знайти найкоротший опис даних, що задовольняють певні обмеження якості, або, навпаки, відшукати найкраще подання зображення для заданого розміру вихідних даних. Фрактальний підхід дає нові можливості для реалізації цієї концепції.

© Г. Я. Кіс, 2020

Фрактальну компресію вперше запропонували М. Барнслі та А. Слоан [1]. Трохи пізніше, 1989 року, А. Жаквін, який був докторантом у Барнслі, першим запрограмував та опублікував успішну схему фрактального стиснення зображення [4].

Фрактальний підхід засновано на гіпотезі Барнслі, що самоподібність притаманна зображенням (наприклад, на рис. 1 позначено подібні ділянки), та на ідеї подання зображення у вигляді стискальних перетворень. Тобто зображення подається не яскравістю пікселів, а коефіцієнтами деяких перетворень, за допомогою яких можна відновити це зображення.

Перетворення W називають стискальним, якщо для будь-яких двох точок P_1, P_2 відстань (Хаусдорфова міра) $d(W(P_1), W(P_2)) < sd(P_1, P_2)$, де $s < 1$.

Із цієї формули випливає, що перетворення простору наближає довільні його точки. Теорема Банаха (1922) стверджує про наявність нерухомої точки стискального перетворення — будь-яке стискальне відображення повного метричного простору в себе має єдину нерухому точку. Якщо відомо перетворення, нерухому точку можна знайти методом послідовних наближень, починаючи з будь-якої точки цього простору.

Отже, якщо побудувати відображення, яке має за нерухому точку дане зображення, то ми побудуємо фрактальне подання цього зображення. Визначення такого відображення W називається оберненою задачею.

Застосовуючи це перетворення ітеративно, можливо реконструювати початкове зображення із заданою точністю. Цей процес відповідає розкиданню стисненого зображення.

Зрозуміло, що оптимальне подання зображення мінімальною кількістю коефіцієнтів перетворень є складною алгоритмічною проблемою. У [5] доведено, що ця задача є NP-повною. Пошук наближення до істинного перетворення є більш простим завданням.

Реальні зображення мають інший тип самоподібності, ніж такі класичні фрактали, як сніжинка Коха, де частина зображення є подібною до всього фрактала. Замість того, щоб зображення формувалось цілими копіями оригіналу за відповідних трансформацій, зображення буде формуватися копіями належним чином трансформованих частин оригіналу, так званім колажем. Цей підхід до побудови стискального відображення як суми більш простих перетворень був запропонований Жаквіном і дістав назву алгоритму колажу. Сума стискальних перетворень теж є стискальним перетворенням. Теорема колажу говорить про те, у який спосіб можна побудувати наближення до оригіналу. Якщо відстань d між оригіналом X та колажем $W(X)$ менша за ϵ , то відстань між оригіналом та нерухомою точкою X_w перетворення W менша за $\epsilon/(1-s)$.

Якщо побудувати перетворення колажу W так, що $W(X)$ наближається до X , то його нерухома точка також близька до X . Це є фундаментальним результатом. Дійсно, бачимо що для заданого блока зображення достатньо знайти параметри, які мінімізують квадратичне відхилення з іншим блоком, та перевірити, що воно буде стискальним.

На практиці вибір перетворень у формі

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \quad (1)$$

є достатнім для подання зображень (афінні перетворення площини). Вони є лінійними і задача лінійної регресії має аналітичний розв'язок. Наступною задачею є визначення відповідного стартового блока для перетворення.

Для подальших розміркувань розглянемо класичну схему блокового фрактального стиснення, запропоновану Жаквіном [4]. Припустимо, що ми маємо зображення 128×128 , на якому кожен піксель може бути одним із 256 рівнів сірого. Його називають ранговим зображенням (Range Image). Потім зменшуємо його вдвічі усередненням зображення до 64×64 пікселів. Це нове зображення має назву доменного зображення (Domain Image). Потім обидва зображення ділять на блоки, наприклад розміром 4×4 пікселі (рис. 2).

До кожного доменного блока застосовується таке афінне перетворення:

$$T(D_{i,j}) = sD_{i,j} + t,$$

де $s \in [0,1]$ та $t \in [-255, 255]$ — ціле число.



Рис. 1. Позначення подібних частин зображення прямокутниками

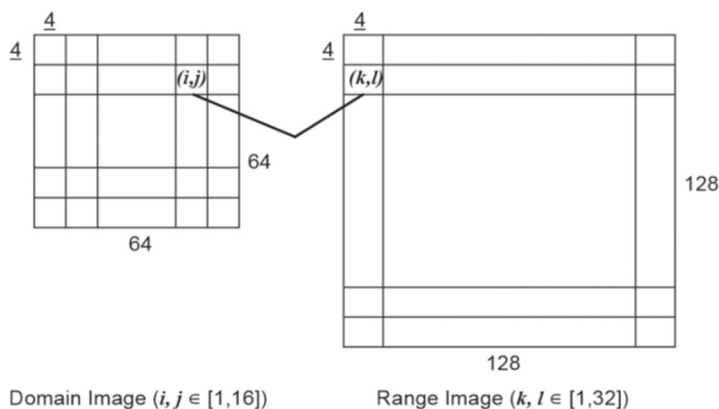


Рис. 2. Доменне та рангове зображення з блоками розміром 4 × 4

Зауважимо, що це кодування не використовує просторові геометричні перетворення, а тільки перетворення яскравості.

Коефіцієнти трансформації для заданого блока $R_{k,l}$ відшуковуються за допомогою мінімізації середнього квадратичного відхилення $E_{i,j} = \sum (R_{k,l} - T(D_{i,j}))^2$, де сума йде по всім пікселям рангового блока. Якщо зафіксувати i, j, k, l , то можна обчислити s, t за формулами лінійної регресії, прирівнюючи відповідні похідні до нуля:

$$\frac{\partial E}{\partial s} = 0, \quad \frac{\partial E}{\partial t} = 0.$$

Отже, для кожного рангового блока (k,l) розглядаємо всі доменні блоки (i, j) та знаходимо той, для якого значення E буде мінімальним. Набір усіх цих коефіцієнтів називається фрактальною кодовою книгою. Бачимо, що пошук доменного блока має лінійну складність, а складність кодування буде квадратичною за кількістю пікселів. У подальшому, кодова книга стискається будь-яким статистичним методом, наприклад арифметичним стисненням.

Ця базова схема з математичного погляду повністю розв’язує задачу, але з практичного боку дуже повільна.

Велика низка робіт присвячена прискоренню побудови перетворень. Було запропоновано кілька покращень. Зрозуміло, якщо зменшити кількість доменних або рангових блоків, то кількість пошуків зменшиться. Так, Саупе [6] вилучив доменні блоки з найменшою дисперсією. Він дослідив, що найчастіше використовуються структуровані блоки з високою варіацією інтенсивності. Та якщо видалити неістотні блоки, то якість та швидкість зростає, але трішки зменшується ступінь компресії (рис. 3).

Використання блоків довільного розміру, наприклад у вигляді квадродерев [7], також зумовлює зменшення кількості блоків. Починаючи з усього зображення, воно поділиться на чотири частини, якщо немає доменного блока, який наближає його з необхідною точністю. Ця схема застосовується рекурсивно для кожного нового рангового блока. Достатньо повний опис алгоритму з математичним обґрунтуванням надано у [8].

Зауважимо що у [9] запропоновано навпаки об’єднувати блоки, якщо для кожного з четвірки знайдено перетворення, але цей підхід суперечить ідеї зменшення кількості блоків (рис. 4).

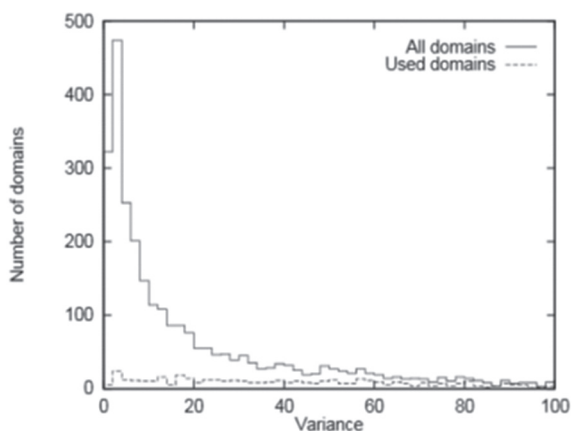


Рис. 3. Гістограма доменних блоків за варіацією згідно з [6] (більшість блоків із малою варіацією не використовуються)

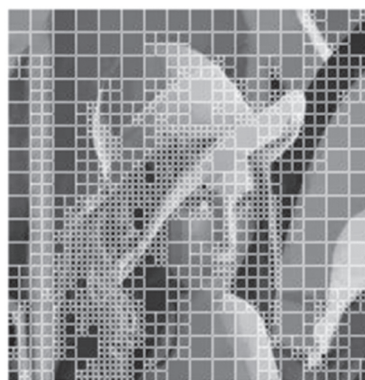


Рис. 4. Розбиття зображення Lena, якщо поріг на середнє квадратичне відхилення 80 [9] (кількість рангових блоків значно менша за просте розбиття)

У [10] запропоновано зменшити кількість доменних блоків, використовуючи ідею локальної подібності. Так, у доменному зображенні беруться тільки референтні блоки. Блок, який дає відстань, меншу за певний поріг із сусідніми блоками в деякому околі, буде вибрано як еталонний (референтний) блок для регіону. Доменний пул формується тільки референтними блоками.

Пошук відповідності до рангових блоків здійснюється спочатку в своєму регіоні. Та якщо немає відповідного референтного блока, кодується тільки його середнє значення. Усі референтні блоки теж кодується звичайними методами. Автори стверджують, що ця техніка дає якість (PSNR) 35 дБ при відношенні стиснення майже 10.

У [3] пішли ще далі та запропонували безпошуковий алгоритм побудови доменного блока. Статистичні дослідження авторів показують, що для більшості рангових блоків імовірність визначення подібного блока максимальна в доменному блоці, центр якого збігається з відповідним ранговим блоком (рис. 5). Грунтуючись на цьому факті, доменний блок взагалі не відшуковують. Експеримент авторів показав, що це спрощення істотно не зменшує якість відтворення (результати унаочнює рис. 6).

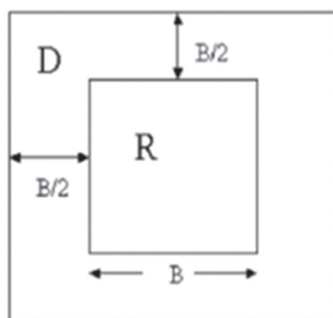


Рис. 5. Розміщення доменного блока відносно рангового згідно з [3]



Рис. 6. Результати фрактального кодування з роботи [3] Lena PSNR 34,02 дБ, час кодування 0,062 с; Baboon PSNR 24,2 дБ, час кодування 0,1 с для ПК 2,8 GHz

Альтернативою підходу зменшення кількості блоків було впровадження схем швидкого пошуку, наприклад за допомогою класифікації блоків [7]. Саупе розвинув цей підхід, описуючи блок багатовимірним вектором (ключем) із використанням методів кластеризації [11]. Незважаючи на те, що задача пошуку найближчого сусіда має логарифмічну складність, метод виявився достатньо повільним.

Більш вдалу схему до фрактального стиснення зображення запропонували в [12], використавши 36-вимірні HOG (гістограма градієнтів) дескриптори блоків (рис. 7). Гістограми градієнтів добре передають структуру блоків і вперше їх було застосовано для ефективного розв'язання задачі розпізнавання пішоходів [3].

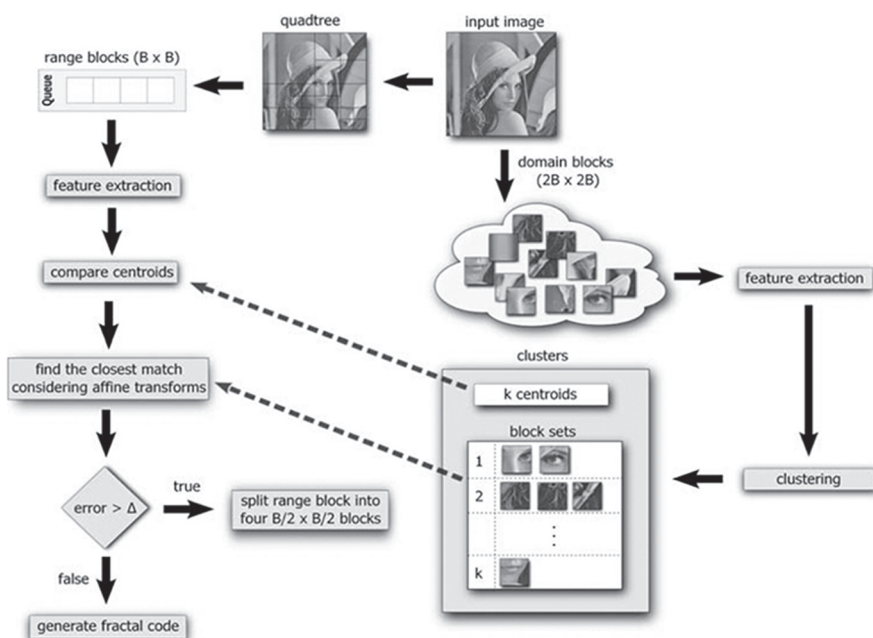


Рис. 7. Фрактальне стиснення зображення з використанням кластеризації [12]

За заданого рівня розкладання квадродерева рангові блоки $V \times V$ пікселів та доменні блоки $2V \times 2V$ пікселів добираються з вхідного зображення. Далі використовують вектори ознак як дескриптор для кожного блока. Алгоритм кластеризації (k -середніх) застосовується для поділу блоків доменів на K кластерів.

Після визначення для блока діапазону найближчого кластера, відшукується доменний блок із цього кластера з найкращим збігом. Потім, як кращий блок домену знайдено, оцінюється помилка відновлення. Якщо вона менша, ніж поріг, фрактальний код генерується, в іншому разі блок діапазону ділиться на чотири субблока розміром $V/2 \times V/2$ пікселів, які будуть розглянуті на наступному рівні розкладання дерева.

Автори дослідили, що цей підхід не прискорюється істотно, якщо кількість кластерів більш ніж 20. Згідно з авторами, якість відтворення краща за метод Фішера [7]. Недоліком є те, що всі доменні блоки кластера обробляються. Можна спробувати покращити цей метод із застосуванням пошуку найближчого сусіда в багатовимірному просторі.

Схема побудови доменних блоків

Пропонується схема вибору доменних блоків на основі методів розпізнавання зображень.

У комп'ютерному зорі та обробленні зображень виявлення ознак стосується методів, спрямованих на обчислення абстракцій інформації зображення. Отримані ознаки будуть підмножинами ділянки визначення зображення, зазвичай у вигляді ізольованих точок, неперервних кривих або зв'язаних ділянок.

Найчастіше використовуються особливі точки (interest point), які належать до точкових ознак зображення, що мають локальну двовимірну структуру.

Найбільш поширеними підходами до детектування особливих точок є алгоритми SIFT [15] D. Lowe (2004), FAST [16] E. Rosten and T. Drummond (2006), SURF [17] Herbert Bay et al. (2006). Особливі точки можуть мати такі ознаки, як орієнтація та розмір.

Після виявлення ознак кожне зображення абстрагується кількома локальними регіонами (патчами, *patch*). Методи представлення особливостей працюють над тим, як подати патчі у вигляді числових векторів. Ці вектори називаються дескрипторами ознак. Хороший дескриптор повинен мати можливість бути інваріантним до освітлення, обертання, масштабування і афінної варіації до певної міри. Один із найвідоміших дескрипторів, інваріантний до масштабу та повороту, — це SIFT. SIFT перетворює кожен блок у 128-вимірний вектор. Після цього кроку кожне зображення являє собою набір векторів тієї самої розмірності (128 для SIFT), де порядок різних векторів не має значення.

Останнім кроком моделювання візуального документу є перетворення векторного подання патчів як «кодових слів» (аналог слів у текстових документах). Кодове слово може розглядатися як представник кількох подібних патчів. Один простий спосіб отримання словника «кодових слів» — це застосування k -середніх (K -means) кластеризації над усіма векторами, аналогічно [12]. Кодові слова потім визначаються як центри вивчених кластерів. Кількість кластерів — це розмір кодової книги.

Отже, кожний фрагмент зображення відображається в певній кодовій комбінації в процесі кластеризації, тому зображення може бути подано як гістограма кодових слів. Кодові книги можна розглядати як стисле уявлення зображення.

Звідси випливає можливість застосування підходу кодової книги для стиснення зображень.

Аналізуючи [14], де автори розглядають таке подання зображення для пошуку подібних зображень, можна побачити, що підхід кодової книги може бути також застосовний для виявлення самоподібності і належних перетворень, оскільки всі потенційно подібні блоки будуть входити до того самого кластера.

Послугуючись висновками Саупе [6], спробуємо побудувати доменний пул, скориставшись тільки структурованими блоками, побудованими в околі особливих точок. А отже, блоки з малою варіацією будуть проігноровані. Рангові блоки будуються за базовою схемою. Виявлення відповідних доменних блоків можливо здійснити за допомогою пошуку у кодовій книзі, що буде значно прискорювати визначення відповідних блоків.

Алгоритм побудови доменних блоків

У схемі, що пропонується, замість кадродерев застосовано простий поділ на фіксовані рангові блоки для спрощення розрахунків.

Припустимо, що маємо вхідне зображення $I(x, y)$ у градаціях сірого розміром (N, N) .

Побудова доменів.

1. Формуємо зменшене зображення $(N/2, N/2)$. Відшукуємо особливі точки за алгоритмом FAST.

2. Знаходимо геометричні властивості (домінантний напрям α_{domain} , середня яскравість I_{domain} , дисперсія $\text{std}_{\text{domain}}$) особливих точок за околom розміру $(8, 8)$. Домінантний напрям визначається як напрям середнього градієнта в околі точки.

3. Будуємо дескриптор — було вибрано дескриптор у вигляді 64-розмірного вектора дійсних чисел, що є нормованою яскравістю пікселів в околі точки. Яскравість відраховується від середнього значення та нормується на дисперсію $i(x,y) = (I(x,y) - I_{\text{domain}}) / \text{std}_{\text{domain}}$. Окіл розгортається вздовж домінуючого напрямку. У таким спосіб досягається деяка інваріантність значень дескрипторів відносно обертання та зміни освітлення.

4. Якщо особливі точки мають майже рівні дескриптори (а отже, подібні), у цьому разі використовуємо тільки точки з найбільшою дисперсією яскравості. Усі інші точки знищуються, оскільки не мають додаткової інформації.

Побудова регіонів.

1. Вхідне зображення розбивається на квадратні блоки (8,8).

2. У кожному блоці визначаються геометричні властивості.

3. Розгортаємо кожний блок та будуємо дескриптори.

Знаходження відповідностей.

За допомогою алгоритму FLANN (*Fast Library for Approximate Nearest Neighbors*), який застосовує k -D дерева для пошуку близьких векторів, визначаємо три найкращих домена для кожного блока. Щоб відображення було стискальним, беремо той із них, який має максимальну дисперсію, і вона повинна бути меншою за дисперсію регіону.

Побудова афінних перетворень.

Знаючи розміри, домінуючі напрями та розташування доменних та рангових блоків, дуже легко побудувати афінні перетворення,

де $\alpha = \alpha_{\text{range}} - \alpha_{\text{domain}}$, R_α — матриця обертання 2×2 на кут α ;

$s = s_{\text{range}} / s_{\text{domain}}$, $s < 1$ — коефіцієнт стиснення; s_{domain} , s_{range} — розміри блоків у пікселях.

Припустимо, що $X(x,y)$ — довільна точка доменного блока, тоді маємо афінне перетворення

$$T(X) = s \cdot R_\alpha (X - x_{\text{domain}}) + x_{\text{range}},$$

де x_{domain} , x_{range} — центр відповідно доменного та рангового блока.

Також кожен піксель має бути трансформований у просторі яскравості:

$$I_{\text{range}} = I_{\text{domain}} \cdot a + b;$$

$$a = \text{std}_{\text{range}} / \text{std}_{\text{domain}};$$

$$b = I_{\text{range}} - I_{\text{domain}} \cdot a.$$

Таким чином, для кожного рангового блока необхідно зберегти кут α , коефіцієнти перетворення яскравості a і b та індекс відповідного доменного блока. Для кожного доменного блока потрібно зберегти ще положення його центра. Усі інші значення або відомі, або легко обчислюються.

Результати експериментальних досліджень

З метою проведення експериментальних досліджень було розроблено алгоритм на мові Python з використанням бібліотеки комп'ютерного бачення OpenCV. Результат роботи подано для зображення Lena (рис. 8) та Baboon (рис. 9) $512 \times 512 \times 8$.



Рис. 8. Оригінальне та реконструйоване зображення (10 ітерацій) Lena PSNR 29 дБ, час кодування 1 с

На процесорі Intel 2,33 GHz розкодування потребує 0,125 с на одну ітерацію. Досягнена якість psnr ~28,8 дБ (Lena), швидкість кодування < 1 с, ~21 дБ (Baboon), швидкість кодування < 2 с.

Залежність якості реконструкції запропонованим методом від кількості ітерацій унаочнює рис. 10.



Рис. 9. Оригінальне та реконструйоване зображення Baboon $512 \times 512 \times 8$, PSNR 21 дБ, час кодування 2 с

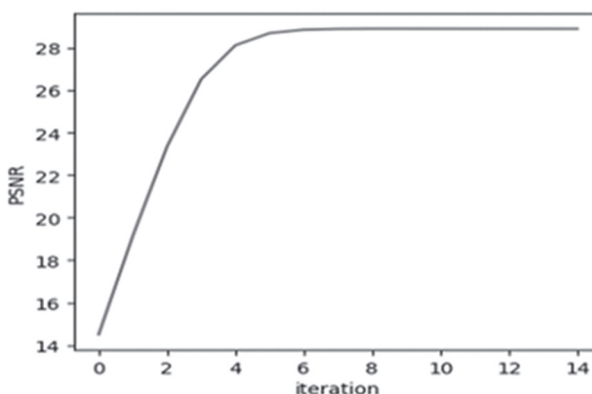


Рис. 10. Залежність якості реконструкції запропонованим методом від кількості ітерацій

На зображенні (рис. 11) можна бачити знайдені доменні блоки за алгоритмом FAST та рангові блоки фіксованого розміру. Доменних блоків набагато менше, ніж рангових, що дає змогу значно прискорити пошук.

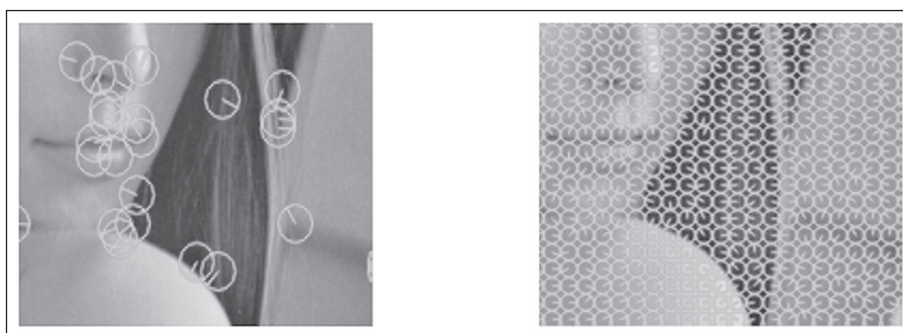


Рис. 11. Приклад доменних та рангових блоків з домінуючими напрямками, які було обчислено

Беручи до уваги, що мова Python є більш придатною для прототипування алгоритмів, можливо очікувати прискорення у кілька разів на мові C++.

Істотна частина обчислювального часу використовується для розрахунку дескрипторів та середнього стандартного відхилення кожного блока, що можна прискорити за допомогою інтегральних зображень [17] зі складністю $O(1)$.

Що стосується підвищення якості реконструкції, то можливо запровадити такі кроки. Оскільки позиція доменних блоків є дуже неточною відносно регулярної структури рангових блоків, її можна покращити, наприклад за допомогою оптичного потоку [18], максимізуючи збіг рангових та доменних блоків. Це буде потребувати додаткової пам'яті для збереження зсувів. Застосування квадродерев для розбиття рангового зображення також може підвищити якість та збільшити ступінь стиснення.

ВИСНОВКИ

Застосування методів комп'ютерного зору, зокрема особливих точок та їх порівняння на основі стандартних дескрипторів, має перспективу для подальшого вивчення, оскільки розроблено значну кількість типів точок, дескрипторів та методів їх порівняння, які працюють у реальному часі. Ці можливі

схеми потребують більш ґрунтовних досліджень щодо практичного використання у системах передавання відео. Підхід легко трансформувати для стиснення відео.

Найбільш відомими є методи фрактальної компресії відео на основі кубів та попереднього кадру [19]. У першому методі відео поділяється на групи кадрів. Кожну групу, у свою чергу, розділено на тривимірні (3D) блоки доменів та рангів. Однак цей метод має високу обчислювальну складність і низький ступінь стиснення.

У разі стиснення на основі кадру кожен кадр кодується з використанням попереднього кадру як пулу доменів. Зрозуміло, що запропонований алгоритм можливо застосувати також для відео, якщо використовувати дескриптори з попереднього кадру.

Список використаної літератури

1. *Michael F. Barnsley, Alan D. Sloan. Methods and apparatus for image compression by iterated function system. US4941193A, 1987.*
2. *Lima V., Schwartz W., Pedrini H. Fast Low Bit-Rate 3D Searchless Fractal Video Encoding. 10.1109/SIBGRAPI.2011.15. P. 189–196.*
3. *Furao S., Hasegawa O. A Fast No Search Fractal Image Coding Method // Signal Processing: Image Communication. 2004. Vol. 19, no. 5. P. 393–404.*
4. *Jacquin A. Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding: Doctoral Thesis, Georgia Institute of Technology, 1989.*
5. *Ruhl M., Hartenstein H. Optimal fractal coding is NP-hard: Proceedings DCC'97 Data Compression Conference / J. A. Storer, M. Cohn (eds.) // IEEE Computer Society Press, March 1997.*
6. *Dietmar Saupe. Lean domain pools for fractal image compression: Conference Proceedings of SPIE Electronic Imaging'96, Science and Technology, Still mage Compression II. San Jose, January 1996. Vol. 2669.*
7. *Fisher Y. Fractal image compression: theory and application. Springer-Verlag, New York, 1995.*
8. *Prem Melville, Vinhthuy Phan. Fractal Compression. 1997.*
9. *Faster fractal image compression using quadtree recomposition / D. Jackson, W. Mahmoud, W. A. Stapleton, P. Gaughan // Image and Vision Computing. 1997. Vol. 15, Issue 10. P. 759–767.*
10. *Abdul-Rahman Selim, Dessouky Hadhoud, Fathi Abd El-Samie. A simplified fractal image compression algorithm: International Conference on Computer Engineering and Systems // ICCES, 2008. P. 53–58. 10.1109/ICCES.2008.4772965, 2008*
11. *Saupe D. Accelerating fractal image compression by multi-dimensional nearest neighbor search: Proceedings DCC '95 Data Compression Conference. Snowbird, UT, USA, 1995. P. 222–231.*
12. *Schwartz W. R., Pedrini H. Improved Fractal Image Compression Based On Robust Feature Descriptors // International Journ. of Image and Graphics. 2011. Vol. 11, No. 4. P. 571–587.*
13. *Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection: International Conference on Computer Vision and Pattern Recognition. Jun. 2005. P. 886–893.*
14. *Feature-Based Sparse Representation for Image Similarity Assessment / Li-Wei Kang, Chao-Yung Hsu, Hung-Wei Chen [et al.] // IEEE Transactions On Multimedia. October 2011. Vol. 13, No. 5.*
15. *Lowe D. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision. 2004.*
16. *Rosten E., Drummond T. Machine learning for high-speed corner detection: European Conference on Computer Vision. 2006. Springer. P. 430–443. (The FAST corner detector).*
17. *Facciolo G., Limare N., Meinhardt E. Integral Images for Block Matching // Image Processing On Line. 2014.*
18. *Tomasi C., Kanade T. Detection and Tracking of Point Features // Carnegie Mellon University Technical Report CMU-CS-91-132. April 1991.*
19. *Lima V., Schwartz W., Pedrini H. Fast Low Bit-Rate 3D Searchless Fractal Video Encoding. P. 189–196. 10.1109/SIBGRAPI.2011.15.*

Г. Я. Кис

ПОДХОД К ФРАКТАЛЬНОМУ СЖАТИЮ ИЗОБРАЖЕНИЙ

Предложен усовершенствованный фрактальный подход к сжатию изображений. Рассмотрена такая проблема фрактальной компресии, как медленная скорость сжатия. Проанализированы несколько существующих методов ее решения. На основе базовых алгоритмов распознавания изображений и компьютерного видения, а именно построении особых точек изображений, исчисления дескрипторов и методов поиска ближайшего соседа в многомерном пространстве разработана схема поиска подобных регионов изображения и предложен алгоритм сжатия. Этот алгоритм имеет сложность $O(N \log N)$ по количеству пикселей изображения,

но если зафиксировать количество доменных блоков, можно достичь почти линейной сложности. Представлено обоснование на основе предыдущих исследований почему это уменьшение возможно. Была разработана программа для проверки этого метода. Представлены результаты численного эксперимента и его анализ. Предложенный подход предназначен для сжатия как изображений, так и видео. Подход перспективен для увеличения скорости и качества сжатия изображений видео и требует дальнейшего совершенствования.

Ключевые слова: фрактальное сжатие изображений; метод компьютерного зрения.

G. Ya. Kis

A FEATURE-BASED APPROACH TO FRACTAL IMAGE COMPRESSION

Recently a number of researches have demonstrated performance improvement in the video fractal compression compared to the current video transmission standards (MPEG, H.263, H.264). This article describes a current problem of relatively low fractal encoding speed. Indeed, high computational complexity is a sore point of fractal compression approach. It seems almost every paper on this subject touches the problem of encoding speed. Productive ideas and algorithms can be borrowed from the pattern recognition problem. In the course of recent decades feature points approach in computer vision has been demonstrating good performance in SLAM and pattern recognition. Technology of feature detection, description and tracking is being developed successfully and has effective applications in augmented reality like Android ARCore and IOS ARKit frameworks that are real-time engines. Similarities among parts of video frames are analyzed and used for both image registration and visual scene tracking therefore it fits highly to block matching task.

Statistic properties for domain/range blocks matching has been analyzed on the basis of previous investigation for fractal compression. As a result, a simple algorithm is proposed based on computer vision approach. The approach includes a visual feature points extraction, feature descriptors calculation and fast NN-search in descriptor space. The key idea of the proposed approach is as follows. Only a limited number of domain blocks around the most salient points are subject to selection. Other blocks are not essential for matching as transforms would have big Lipschitz constant and will have worse contractive properties. Salient points should be unique as well. Further the descriptors for feature point are calculated. The algorithm has $O(N \log N)$ complexity for pixel number in the frame image, however if the number of domain blocks is limited the complexity could be almost linear.

Python program for the algorithm test has been developed and shows that reconstruction result is acceptable in terms of encoding speed (< 2 s on 2 GHz CPU) and quality (PSNR) ~ 25 dB. The result of the proposed approach could be interesting for further improvement both for image and video compression. Further steps for quality increasing are also described.

Keywords: fractal image compression; computer vision method.

