

DOI: <https://doi.org/10.36910/6775-2524-0560-2023-53-32>

УДК 004.021

Старченко В'ячеслав Володимирович, старший викладач

<https://orcid.org/0000-0002-7039-2332>

Чорноморський національний університет імені Петра Могили, м. Миколаїв, Україна

АЛГОРИТМИ ФРАКТАЛЬНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ

Старченко В.В. Алгоритми фрактального стиснення зображень. Робота присвячена розгляду ключових аспектів та проблем застосування методів фрактального стиснення двовимірних зображень. Розглянуті особливості алгоритмів фрактальної компресії та відновлення зображень. Визначені шляхи підвищення швидкості роботи алгоритмів за рахунок оптимізації обмеження глибини рекурсії.

Ключові слова: фрактал, топологічна розмірність, рекурсія, L-система, система ітеруючих функцій, афінне перетворення, аттрактор, домен, ранг.

Starchenko V. Images fractal compression algorithms. The work is devoted to the consideration of key aspects and problems of applying methods of fractal compression of two-dimensional images. Features of fractal compression and image restoration algorithms are considered. Ways to increase the speed of the algorithms by optimizing the recursion depth limit are determined.

Keywords: fractal, topological dimension, recursion, L-system, Iterated Function Systems, affine transformation, attractor, domain, rank.

Вступ.

Фрактали зустрічаються скрізь, де закінчуються правильні форми евклідової геометрії. Все, що створено людиною, обмежено площинами. Якщо зустрічається природний об'єкт, то з першого погляду видно, що усвідомити та описати його форму з усіма дрібними деталями можна лише приблизно. Тут на допомогу приходять фрактали [1].

Термін "фрактал" (від латинського слова "fractus" – дріб) введений бельгійським математиком Бенуа Мандельбротом і позначає множину, що має дрібну фрактальну розмірність. Для пояснення фрактальної розмірності необхідно запровадити поняття топологічної розмірності [2]. Під топологічною розмірністю Dt множини у лінійному просторі розуміють число лінійно незалежних координат у просторі. Наприклад, коло та лінія мають топологічну розмірність 1; коло та квадрат – 2; куля і куб – 3. Фрактальна розмірність множини D – розмірність того простору, який повністю заповнюється множиною [3]. Для зв'язку фрактальної та топологічної розмірностей використовують показник Херста H , який обчислюється за формулою: $H = D - Dt$. Фракталом називають множину, фрактальна розмірність якої не збігається з топологічною [4]. Наприклад, для кривих Пеано (криві, що заповнюють площину), $Dt = 1$, $D = 2$. Існують два основні способи побудови фракталів. Перший спосіб – використання L-систем (від імені Lindenmaayer), другий спосіб – застосування системи IFS (англ. Iterated Function Systems).

Постановка наукової проблеми.

Об'єктом дослідження є методи підвищення ефективності алгоритмів фрактального стиснення двовимірних зображень.

Предметом дослідження є алгоритми фрактального стиснення двовимірних зображень та способи обмеження глибини рекурсії для прискорення їх роботи.

Метою роботи є визначення основних перспективних напрямків прискорення алгоритмів фрактального стиснення зображень за допомогою оптимального вибору обмежень на їх рекурсивну складову.

Аналіз досліджень

За допомогою фракталів можна стискати довільні зображення з деякою втратою якості, аналогічно стиску за алгоритмом JPEG. При цьому, фрактальне стиснення дає найкращі результати [5]. Методи компресії, засновані на алгоритмах RLE, Huffman або LZW, не враховують природи даних, що стискаються, і тому дають незадовільні результати при обробці зображень. При фрактальному стисненні зображення переводиться у формат IFS, який значно економніший, ніж просто BITMAP або LZW. Стиснення JPEG засноване на дискретному синусно-косинусному перетворенні (дискретне перетворення Фур'є), яке переводить зображення в амплітудно-фазову форму. Спотворення в області високих частот не сильно впливають на якість вихідних зображень, тому їх частково відкидають, а частково фільтрують низькочастотним фільтром з наступним стисненням за допомогою RLE. Тому метод JPEG дає погані результати на зображеннях, що містять

тонкі лінії (текст або креслення). Для того щоб отримати прямокутний сигнал, а тонка лінія саме таким сигналом і є, необхідно використовувати область високих частот, які у випадку JPEG фільтруються, піддаючись при цьому спотворенню [6].

L - системи

L-система – це граMATика деякої мови (досить простої), яка описує ініціатор і перетворення, що виконується над ним. Розглянемо класичний приклад фрактальної множини – триадну криву Коха (Рис. 1). Побудова кривої починається з одиничного відрізка, який називається ініціатором і є передфракталом 0-го порядку. Далі ініціатор замінюється на утворюючий елемент – криву з чотирьох прямолінійних ланок, кожна з яких має довжину $1/3$. Так утворюється передфрактал 1-го порядку. Його довжина дорівнює $4/3$. Для побудови передфрактала наступного порядку кожна ланка замінюється зменшеним утворюючим елементом. В результаті отримуємо криву, що складається з $4 \times 4 = 16$ ланок, кожна з яких має довжину $(1/3) / 3 = 1/9$, загальна довжина дорівнює $16/9$. Довжина передфрактала n -го порядку дорівнює $(4/3)^n$ у ступені n . Очевидно, що межа довжини кривої при n , що прагне до нескінченності, дорівнює нескінченності.

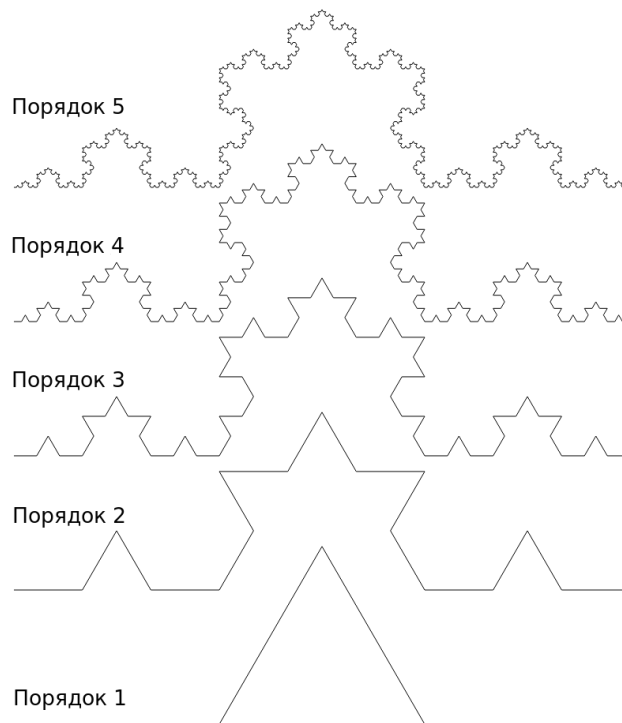


Рис. 1. Фрактал Коха

L-системи призначені для генерування передфракталів заданого порядку. Ця властивість відрізняє їх від IFS, призначених для побудови самих фракталів.

Системи ітеруючих функцій (IFS)

Система ітеруючих функцій – це сукупність афінних перетворень, що стискають. Як відомо, афінні перетворення включають масштабування, поворот і паралельне перенесення. Афінне перетворення вважається таким, що стискає, якщо коефіцієнт масштабування під час його виконання менше одиниці [7].

Розглянемо докладніше побудову кривої Коха із використанням афінних перетворень. Кожен новий елемент кривої містить чотири ланки, отримані з утворюючого елемента використанням масштабування, повороту та перенесення:

- 1) для отримання першої ланки достатньо стиснути вихідний відрізок утричі. Слід зазначити, що те саме масштабування застосовується для всіх ланок;
- 2) наступна ланка будується з використанням всіх можливих перетворень, а саме: стиснення в три рази, поворот на мінус 60° і паралельне перенесення на $1/3$ по вісі X .
- 3) третя ланка будується аналогічно другій: стиснення в три рази, поворот на 60° , паралельне перенесення на $2/3$ по вісі X ;
- 4) остання ланка: стиснення втричі, паралельне перенесення на $2/3$ по вісі X .

Тепер можна знайти систему функцій IFS для опису кривої Коха. Для цього треба зробити суперпозицію афінних перетворень – масштабування, повороту та паралельного перенесення.

З курсу лінійної алгебри відома формула обчислення нових координат x' , y' за афінними перетвореннями:

$$\begin{aligned}x' &= a \times x - b \times y + e, \\y' &= c \times x - d \times y + f, \\a &= \text{scale}_x \times \cos(\alpha), \\b &= \text{scale}_x \times \sin(\alpha), \\c &= \text{scale}_y \times \sin(\alpha), \\d &= \text{scale}_y \times \cos(\alpha), \\e &= \text{move}_x, \\f &= \text{move}_y,\end{aligned}$$

де scale_x – масштабування по вісі X ;

scale_y – масштабування по вісі Y ;

α – кут оберту;

move_x – паралельне перенесення вздовж вісі X ;

move_y – паралельне перенесення вздовж вісі Y .

Отримані коефіцієнти a , b , c , d , e , f для кожної ланки складуть необхідну систему функцій, що ітерують.

Коефіцієнти афінного перетворення IFS усіх ланок кривої Коха зведені у таблиці 1.

Таблиця 1. Коефіцієнти афінного перетворення IFS усіх ланок кривої Коха

№ ланки	a	b	c	d	e	f
1	0,3333	0,0000	0,0000	0,3333	0,0000	0,0000
2	0,1667	-0,2887	0,2887	0,1667	0,3333	0,0000
3	-0,1667	0,2887	0,2887	0,1667	0,6666	0,0000
4	0,3333	0,0000	0,0000	0,3333	0,6666	0,0000

Афінне перетворення для однієї ланки є масштабуванням на коефіцієнт $1/3 = 0,3333$. Коефіцієнти для першої та останньої ланок кривої Коха практично ідентичні та відрізняються лише паралельним перенесенням по вісі X (коефіцієнт e).

Друге і третє перетворення включають не тільки масштабування і перенесення, а й оберт на 60° і мінус 60° . Тут коефіцієнти обчислюються так:

$$\begin{aligned}0,1667 &= 1/3 \times \cos(60); \\-0,2887 &= 1/3 \times [-\sin(60)].\end{aligned}$$

Для синтезу фракталу вибирається початкова точка, до якої застосовується випадково обране з IFS-перетворення, в результаті чого точка переміщується в інше місце пристрою відображення. Ця операція повторюється багато разів (достатньо 100 ітерацій), і через деякий час точка починає блукати по атрактору (атрактор – множина всіх можливих траєкторій), який і буде зображенням фракталу. Кожне нове положення точки фарбується кольором, відмінним від фону. Існує теорема, що доводить, що отриманий атрактор буде замкненим. Якщо вибрати початкову точку так, щоб вона відразу опинилася на атракторі, то вона починає блукати в області цього атрактора, не потрапляючи у інші частини пристрою відображення. Розглядаючи кожне перетворення окремо, можна помітити, що після кількох ітерацій, точка перестане рухатися. Точка зупинки називається нерухомою точкою – це рішення системи лінійних рівнянь двох змінних, яке шукається методом простої ітерації. Нерухома точка кожного перетворення входить до складу атрактора. Тому за початкову точку при побудові фракталу можна взяти нерухома точку першого перетворення з набору IFS.

Збільшуючи роздільну здатність пристрою відображення або масштаб, можна побачити нові частини фракталу, які будуть схожі на весь фрактал. Це найважливіша ознака фракталу – самоподібність.

Якщо для побудови фракталу використовувати систему ітеруючих функцій, то можна отримати зображення, деталізація якого обмежена лише роздільною здатністю пристрою відображення, на відміну від побудови, що ґрунтується на L-системі, де точність залежить від заданого порядку передфракталу. Щоб отримати високу роздільну здатність з використанням L-

систем, необхідно задавати великий порядок передфракталу, але оскільки побудова заснована на рекурсивному алгоритмі, відповідно збільшується глибина рекурсії і, як наслідок, уповільнення процесу побудови.

Фрактальне стиснення зображень

Вперше можливість застосування теорії IFS до проблеми стиснення зображення було досліджено Майклом Барнслі. Арнольд Жакин (в 1992 році – аспірант Майкла Барнслі) представив метод фрактального кодування, в якому використовуються системи доменних та рангових блоків зображення (англ. domain and range subimage blocks) квадратної форми, що покривають усе зображення. Цей підхід став основою більшості методів фрактального кодування [8]. Згідно з цим методом зображення повинно бути розбите на велику кількість рангових зображень R , що не перекриваються, також визначається множина неперетинних доменних зображень D . Для кожного рангового блоку алгоритм кодування знаходить найбільш оптимальний доменний блок та афінне перетворення, яке перетворює цей доменний блок на даний ранговий блок. Структура зображення відображається в системі рангових блоків, доменних блоків та перетворень. Блок-схема типового алгоритму фрактальної компресії наведена на Рис. 2. Ідея методу полягає в наступному: припустимо, що вихідне зображення є нерухомою точкою якогось стискаючого відображення. Тоді можна замість самого зображення запам'ятати це відображення, а для відновлення досить багаторазово застосувати це відображення до будь-якого стартового зображення. За теоремою Банаха [9], такі ітерації завжди призводять до нерухомої точки, тобто до початкового зображення [10].

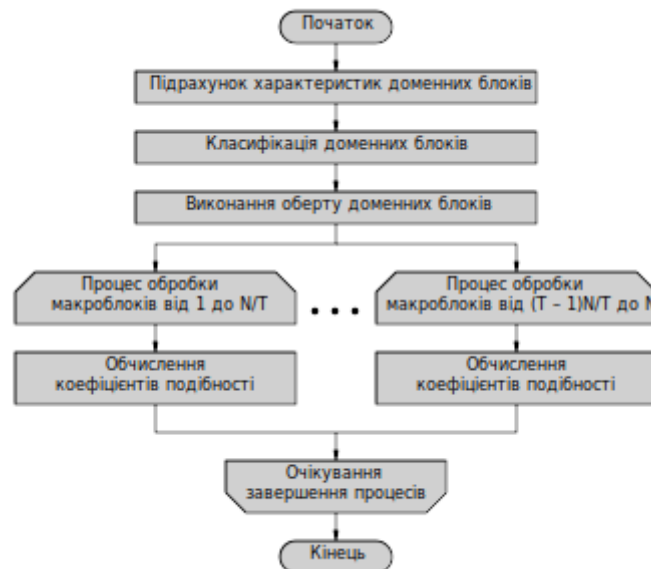


Рис. 2. Алгоритм фрактальної компресії

Стиснення відбувається таким чином. Зображення розбивається на частини множиною R і покривається множиною D . Причому елементи множини D більші за площею, ніж R . Для кожного елемента $R(i)$ перебираються всі елементи множини D і будуються афінні перетворення $R(i) \rightarrow D(j)$. Зі всіх перетворень вибирається одне, що відбувається з найменшою похибкою. Афінне перетворення виконується за трьома координатами – X, Y, C (Колір). Наведемо це перетворення:

$$\begin{vmatrix} x \\ y \\ c \end{vmatrix} = \begin{vmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{vmatrix} * \begin{vmatrix} x \\ y \\ c \end{vmatrix} + \begin{vmatrix} e \\ f \\ 0 \end{vmatrix}$$

де a, b, c, d, e, f — коефіцієнти перетворення на площині;
 s — контрастність;
 o — яскравість.

Інші коефіцієнти дорівнюють 0, тому що немає потреби пов'язувати зміну кольору з перетворенням координат. Від числа перетворень, тобто потужності множини R , залежить ступінь компресії, адже на кожне перетворення потрібно 8 чисел. Множина таких перетворень і становить шукану систему ітеруючих функцій.

Розбиття на R та покриття D є найскладнішою частиною алгоритму. У найпростішому випадку це робиться так: зображення розбивають регулярною сіткою $R(i)$ по 8×8 пікселів. Для кожного $R(i)$ перебирають усі можливі $D(j)$ по 16×16 (причому є 8 варіантів для кожного квадрата – 4 повороти на 90° та дзеркальна симетрія). Для кожного $R(i)$ вводять своє найкраще перетворення, а потім шукають мінімальне покриття вихідного зображення з наявних $D(j)$. Залишають ті перетворення, які потрібні для мінімального покриття. Один з варіантів такого ділення наведено на Рис. 3.

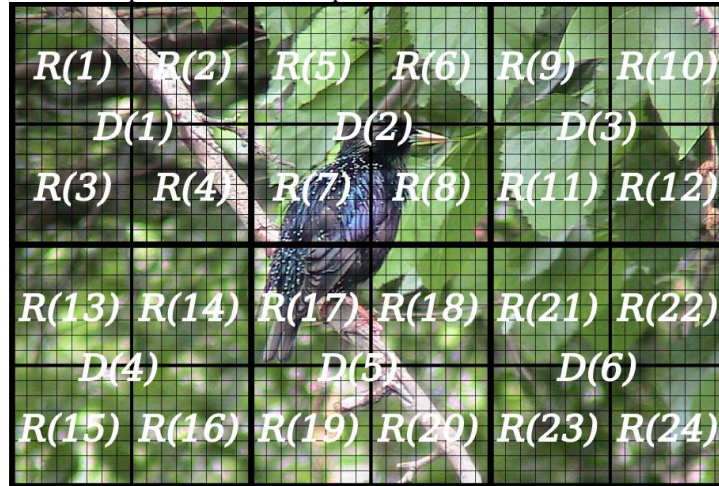


Рис. 3. Один з можливих варіантів розбиття зображення на доменні (D) та рангові (R) блоки

Декомпресія виконується таким чином. Обирається початкове зображення, з якого буде побудований оригінал. Від вибору початкового зображення залежить швидкість стабілізації зображення, тобто час декомпресії. До кожного елемента множини R (тобто квадрату 8×8 з ініціатора) застосовують випадково обране афінне перетворення з наявної системи ітеруючих функцій. В результаті такої дії елементи зображення копіюються в інші частини пристрою відображення зі зміною орієнтації, яскравості та контрасту. Після того, як усі $R(i)$ перебрані, починається усе спочатку. За теоремою про систему стискаючих афінних перетворень зображення прагнуть до стабільності. Зазвичай достатньо 10 – 20 ітерацій [11].

При компресії можна не зберігати оригінальні розміри зображення, достатньо запам'ятати їх співвідношення. А під час декомпресії – задавати ті розміри, які потрібні. Така можливість дозволяє вирішити завдання екстраполяції початкового зображення. При встановленні нових розмірів, що перевищують старі, до нового зображення додаються елементи, подібні до інших елементів зображення. І якщо обробляється природний об'єкт з регулярною текстурою, підміна не буде помітна.

Основна проблема фрактального стиску – це те, що декомпресія виконується швидко і однозначно, тоді як пряма процедура (компресія) вимагає від системи великих обчислювальних потужностей. Другою проблемою є отримання оптимального перетворення $R(i) \rightarrow D(j)$.

Основні напрямки щодо прискорення фрактального алгоритму стиснення є такими:

- удосконалення схеми розбиття зображення на ранги;
- прискорення порівняння рангу та домену;
- обмеження кількості доменів, що беруть участь у порівнянні з цим рангом шляхом класифікації рангових та доменних блоків.

Для покращення схеми розбиття зображення її організація повинна здійснюватися у два етапи:

- 1) розділити покриття зображення ранговими блоками, тобто сформувати грубу версію схеми розбиття;
- 2) відкинути області, що повністю перекриваються сусідніми рангами.

У сучасних алгоритмах фрактального стиснення найчастіше використовуються такі схеми розбиття:

- 1) квадродререво — схема розбиття у вигляді дерева, вузли якого відповідають квадратній області зображення. Кожен вузол містить у собі чотири підвузли, які відповідають чотирьом підобластям цього квадрата. Схема не дозволяє додавати довільну кількість блоків, тому блоки меншого розміру додаються лише групами по чотири. Блоки найбільшого розміру утворюють сітку,

до якої прив'язана вся схема розбиття зображення на рангові блоки, утворену блоками найбільшого розміру;

2) мозаїка — схема розбиття початкового зображення за якою воно ділиться на блоки різного розміру, прив'язані до сітки, що складається з блоків найменшого розміру. У порівнянні з алгоритмом квадродерева це гарантує найбільш гнучке та раціональне покриття об'єкта блоками. За рахунок такого покриття зображення кількість блоків зменшиться.

Для прискорення порівняння рангу та домену на алгоритм стиснення можна накласти такі обмеження:

- обмежити кількість перетворень, наперед забезпечивши ступінь стиснення не нижче фіксованої величини;
- вимагати, щоб у ситуації, коли різниця між обробленим фрагментом і найкращим його наближенням буде вищою за певне порогове значення, цей фрагмент обов'язково дробився на частини;
- заборонити дробити фрагменти розміром менше, наприклад, чотирьох пікселів.

Змінюючи порогові значення та пріоритет цих обмежень, можна дуже гнучко керувати коефіцієнтом компресії зображення: від попиксельної відповідності до будь-якого ступеня стиснення.

Висновки та перспективи подальшого дослідження

Значна складність та повільність процесу стиснення є причиною того, що фрактальні методи найкраще пристосовані для застосунків архівування, таких як цифрові енциклопедії, в яких кодування необхідне лише одного разу, а декодування відбувається багато разів. Тому переважна більшість досліджень у галузі фрактального стиску зараз спрямована на зменшення часу архівації, необхідного для отримання якісного зображення. На даний момент відомо багато алгоритмів оптимізації перебору, що виникає при фрактальному стиску. Для збільшення швидкості кодування проводиться робота за двома напрямками. У рамках першого напрямку розглядаються задачі класифікації доменів (англ. classification of domains), у яких за рахунок зменшення кількості доменів, серед яких ведеться пошук, скорочується кількість обчислень. Другий напрямок ґрунтується на методах виділення особливостей зображення (англ. feature extraction). При цьому підході збільшення швидкості кодування відбувається за рахунок порівняння доменних та рангових блоків.

Список бібліографічного опису

1. Hill F. S. Computer graphics. Macmillan Publ., 1990. 754 p.
2. Адашевська І. Ю., Краєвська О. О. Самоподібність як характеристична властивість фракталу. Фрактальна (дробова) розмірність Хаусдорфа. Scientific Achievements of Modern Society : abstr. of 4th Intern. Sci. and Practical Conf., Liverpool, United Kingdom, 4–6 Dec. 2019. P. 603–612. URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/44138> (дата звернення: 14.05.2023).
3. Morris S. A. Topology without tears. Publ. June 28, 2020. 704 p. URL: <https://www.topologywithouttears.net/topbook.pdf> (Last accessed: 14.05.2023).
4. Rosenberg E. Fractal dimensions of networks. Springer, 2020. 524 p.
5. Fisher Yu. Fractal image compression. IEEE SIGGraph Course Notes. 1992. 21 p.
6. Foley J. D., van Dam A., Feiner S. K., Hughes J. F. Computer graphics: Principles and practice. Addison-Wesley Publ., 1990. 1006 p.
7. Amar A. B., O'Regan D. Topology and approximate fixed points. Springer, 2022. 247 p.
8. Liu S. T., Zhang Yo. P., Liu C. A. Fractal control and its applications. Springer, 2020. 352 p.
9. Agarwal P., Jleli M., Samet B. Fixed point theory in metric spaces. Springer, 2018. 165 p.
10. Subrahmanyam P.V. Elementary fixed point theorems. Springer, 2019. 297 p.
11. Miller G. The definition and rendering of terrain maps. Computer Graphics. 1986. Vol. 20, No. 4. P. 39–48.
12. Свинчук О. В., Барабаш О. В., Олімпієва Ю. І., Льїн О. Ю. Застосування фрактальних функцій для шифрування даних в системах захисту інформації. Телекомунікаційні та інформаційні технології. 2020. № 1 (66). С. 15–24. DOI: 10.31673/2412-4338.2020.011524.
13. Дахно Н. Б., Шевченко Г. В., Свинчук О. В. Фрактальні функції в телекомунікаційних мережах. Проблеми інформатизації : матеріали 13 Міжнар. наук.-техн. конф., 11–12 квітня 2019 р. Київ : Держ. ун-т телекомунікацій, 2019. С. 39–41.
14. Cruz-Cunha M. M., Mateus-Coelho N. R. Handbook of research on cyber crime and information. Hershey, PA : IGI Global, 2020. 739 p.
15. Li M. Fractal teletraffic modeling and delay bounds in computer. CRC Press, 2022. 238 p.

References

1. Hill F. S. Computer graphics. Macmillan Publ., 1990. 754 p.
2. Adashevskaya I. Yu., Kraevskaya O. O. Self-similarity as a characteristic property of a fractal. Hausdorff's fractal (fractional) dimension. Scientific Achievements of Modern Society : abstr. of 4th Intern. Sci. and Practical Conf.,

- Liverpool, United Kingdom, 4–6 Dec. 2019. P. 603–612. URL: <https://repository.kpi.kharkov.ua/handle/KhPI-Press/44138> (Last accessed: 05/14/2023).
3. Morris S. A. Topology without tears. Publ. June 28, 2020. 704 p. URL: <https://www.topologywithouttears.net/topbook.pdf> (Last accessed: 14.05.2023).
 4. Rosenberg E. Fractal dimensions of networks. Springer, 2020. 524 p.
 5. Fisher Yu. Fractal image compression. IEEE SIGGraph Course Notes. 1992. 21 p.
 6. Foley J. D., van Dam A., Feiner S. K., Hughes J. F. Computer graphics: Principles and practice. Addison-Wesley Publ., 1990. 1006 p.
 7. Amar A. B., O'Regan D. Topology and approximate fixed points. Springer, 2022. 247 p.
 8. Liu S. T., Zhang Yo. P., Liu C. A. Fractal control and its applications. Springer, 2020. 352 p.
 9. Agarwal P., Jleli M., Samet B. Fixed point theory in metric spaces. Springer, 2018. 165 p.
 10. Subrahmanyam P.V. Elementary fixed point theorems. Springer, 2019. 297 p.
 11. Miller G. The definition and rendering of terrain maps. Computer Graphics. 1986. Vol. 20, No. 4. P. 39–48.
 12. Svinchuk O.V., Barabash O.V., Olimpiyeva Yu.I., Ilyin O.Yu. Application of fractal functions for data encryption in information protection systems. Telecommunications and information technologies. 2020. No. 1 (66). P. 15–24. DOI: 10.31673/2412-4338.2020.011524.
 13. Dakhno N. B., Shevchenko G. V., Svinchuk O. V. Fractal functions in telecommunication networks. Problems of informatization: materials of the 13th International science and technology conference, April 11–12, 2019. Kyiv: Derz. University of Telecommunications, 2019. P. 39–41.
 14. Cruz-Cunha M. M., Mateus-Coelho N. R. Handbook of research on cyber crime and information. Hershey, PA : IGI Global, 2020. 739 p.
 15. Li M. Fractal teletraffic modeling and delay bounds in computer. CRC Press, 2022. 238 p.