**СТИСНЕННЯ ЗОБРАЖЕНЬ В КОМП’ЮТЕРНІЙ ГРАФІЦІ**

**Сти́снення зобра́жень** — використання [алгоритмів](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) [стиснення даних](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B5%D0%B7_%D0%B2%D1%82%D1%80%D0%B0%D1%82) до зображень, що зберігаються в [цифровому виді](https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D1%84%D0%B0%D0%B9%D0%BB%D1%83). В результаті стиснення зменшується розмір зображення, що зменшує час передачі зображення по мережі і економить простір для зберігання. Стиснення зображень розділяють на [стиснення з втратами](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7_%D0%B2%D1%82%D1%80%D0%B0%D1%82%D0%B0%D0%BC%D0%B8) якості і [стиснення без втрат](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B5%D0%B7_%D0%B2%D1%82%D1%80%D0%B0%D1%82). Стиснення без втрат більш підходить для штучно побудованих зображень, таких як [графіки](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D1%96%D0%BA), іконки програм, або для спеціальних випадків, наприклад, якщо зображення призначені для подальшої обробки алгоритмами розпізнавання зображень. Алгоритми стиснення з втратами при збільшенні степені стиснення, як правило, породжують добре помітні людському оку [артефакти](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%82%D0%B5%D1%84%D0%B0%D0%BA%D1%82).

**Стиснення:**

Растрової графіки Векторної графіки

Статичних нерухомих зображень Відеопослідовностей

Зображень для візуального сприйняття

Зображень з вимірювальною інформацією

Стиснення без втрат інформації Стиснення з втратою частини інформації

Стиснення та кодування

**Актуальність стиснення зображень:**

1. Компактне зберігання колекцій та каталогів зображень

 Зображення для візуального сприйняття

 Зображення для оформлення поліграфічної продукції, веб-дизайну, реклами, відеороліків

 Зображення з вимірювальною інформацією

 Набори еталонних зразків об’єктів та промислових виробів

2. Передача зображень по комп’ютерним мережам, по інтерфейсам передачі даних в інф-вим системах

**Послідовність роботи з зображеннями з використанням процедур стиснення**

Формування зображення цифровою відеокамерою

(масив цифрових даних – матриця, містить відліки яскравості та кольору у вузлах прямокутної сітки)

|

Стиснення зображень (програмне або апаратне)

(графічний файл стиснутого формату або область у пам’яті компютера)

|

Зберігання стиснутого зображення на запам’ятовуючих пристроях

|

Передача стиснутого зображення по комп’ютерним мережам або інтерфейсам ІВС

|

Відновлення зображення після стиснення (програмне або апаратне)

(масив **відновлених** цифрових даних – матриця, містить відліки яскравості та кольору у вузлах прямокутної сітки)

Може містити викривлення та втрати частини інформації

|

Програмно-алгоритмічна обробка відновленого зображення

1. Візуалізація відновленого зображення

2. отримання вимірювальною інформації про об’єкти (колір, геометричні параметри, параметри руху)

|

1. Візуальна оцінка вмісту зображення

2. Результати вимірювання параметрів об’єктів

**Критерії оцінки якості відновленого зображення**

*Викривлення на відновленому зображенні обумовлені:*

Обчислювальні похибки при стисненні без втрат

(округлення дійсних чисел, переповнення розрядної сітки від 0 до 255)

Втрати інформації в результаті стиснення

*Критерії:*

1. Візуальна якість зображення (метод експертних оцінок)

2. Амплітудні похибки (середньоквадратичне значення різниці між початковим та відновленим після стиснення зображеннями)

Обчислюють оцінку середньоквадратичного значення похибки відтворення амплітуди цифрового відеозображення [161]:

,

де  – початкове цифрове зображення розміром MxN точок.

Співвідношення сигнал/шум в телевізійній техніці



Тоді на основі визначення співвідношення сигнал/шум в телевізійній техніці

,

де  – максимальне значення амплітуди цифрових відеозображень (при кодуванні амплітуди за допомогою 8-ми розрядів двійкового коду  дискретних рівнів (д.р.)),  – співвідношення сигнал/шум для ПФВЗ.

3. Похибки вимірювальної інформації (передача кольору, передача геометричних параметрів)

**Можливість стиснення** обумовлена наявною інформаційною надлишковістю зображень.

*Стиснення без втрат інформації*

1. Статистична надлишковість

2. Кореляційні зв’язки між сусідніми точками зображення

*Стиснення з втратами частини інформації*

3. Особливості візуального сприйняття зображень людиною

4. Допустима величина похибок вимірювання параметрів об’єктів

Для цього вводиться поняття швидкості створення інформації в ПФВЗ  [161], що фактично визначає кількість біт, необхідну для передачі інформації про одну д.т. цифрового відеозображення. Цифрове відеозображення можна вважати двовимірним марковським джерелом з експоненційною кореляційною функцією і нормальним законом розподілу амплітуди. Якщо , то згідно [Претт ЦОИ 1982] маємо:

,

де  – дисперсія амплітуди відеозображення,  і  – коефіцієнти кореляції відліків у рядках і стовпцях відеозображення, =1 біт/(д.т.) – коефіцієнт пропорційності, що визначає розмірність  для випадку двійкового кодування цифрового відеозображення.



Гонсалес Вудс ЦОИ 2005



Претт ЦОИ 1982 т.2 стр 767





## Класи додатків, що використовують стиснення[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=1) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=1)]

В загальному виділяють 3 класи додатків, що використовують алгоритми компресії:

* **Клас 1.** Характеризуються високими вимогами до часу архівації та розархівації. Нерідко потрібно виконати перегляд зменшеної копії зображення і пошук у базі даних зображень.
* **Клас 2.** Характеризується високими вимогами до ступеня архівації та часу розархівації. Час архівації ролі не грає.
* **Клас 3.** Характеризується дуже високими вимогами до ступеня архівації.

Можна навести безліч більш вузьких класів додатків. Так, своє застосування машинна графіка знаходить і в різних [інформаційних системах](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Наприклад, вже стає звичним досліджувати ультразвукові та [рентгенівські знімки](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D1%82%D0%B3%D0%B5%D0%BD%D1%96%D0%B2%D1%81%D1%8C%D0%BA%D0%B8%D0%B9_%D0%B7%D0%BD%D1%96%D0%BC%D0%BE%D0%BA) не на папері, а на екрані монітора. Поступово в електронний вигляд переводять та історії хвороб. Зрозуміло, що зберігати ці матеріали логічніше в єдиній картотеці. При цьому без використання спеціальних алгоритмів більшу частину архівів займуть фотографії. Тому при створенні ефективних алгоритмів вирішення цього завдання потрібно врахувати специфіку рентгенівських знімків — переважання розмитих ділянок. У геоінформаційних системах — при зберіганні аерофотознімків місцевості — специфічними проблемами є великий розмір зображення і необхідність вибірки лише частини зображення на вимогу. Крім того, може знадобитися масштабування. Це неминуче накладає свої обмеження на алгоритм компресії. В електронних картотеках і досьє різних служб для зображень характерно подібність між фотографіями в профіль, і подібність між фотографіями в фас, яке також необхідно враховувати при створенні алгоритму архівування. Подібність між фотографіями спостерігається і в будь-яких інших спеціалізованих довідниках. Як приклад можна навести енциклопедії птахів чи квітів.

## Вимоги до алгоритмів компресії[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=2) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=2)]

Характер використання зображень визначає ступінь важливості наведених нижче суперечливих вимог до алгоритму:

* **Високий ступінь компресії.**
* **Висока якість зображень.** Виконання цієї вимоги напряму суперечить виконанню попередньої.
* **Висока швидкість компресії.** Ця вимога для деяких алгоритмів з втратою інформації є взаємовиключною з першими двома. Що більше часу необхідно витратити на аналіз зображення з метою отримання найвищої ступені компресії, тим кращим буде результат. І, відповідно, чим менше буде витрачено на компресію (аналіз), тим нижчою буде якість зображення і більшим його розмір.
* **Висока швидкість декомпресії.**
* **Масштабування зображень.** Дана вимога передбачає легкість зміни розмірів зображення до розміру вікна активного додатку. Одні алгоритми дозволяють легко масштабувати зображення під час декомпресії, в той час як інші не тільки не дозволяють легко масштабувати, але і збільшують вірогідність появи «неприємних» артефактів після застосування стандартних алгоритмів масштабування до декомпресованих зображень. Приміром можна навести приклад «поганого» зображення для алгоритму [JPEG](https://uk.wikipedia.org/wiki/JPEG) — це зображення з досить дрібним регулярним малюнком (піджак в дрібну клітку). Характер внесених алгоритмом JPEG спотворень такий, що зменшення або збільшення зображення може дати неприємні ефекти.
* **Стійкість до помилок.** Дана вимога означає локальність порушень у зображенні при пошкодженні або втраті фрагмента переданого файлу. Дана вимога суперечить вимозі високого ступеня архівації, оскільки необхідно вводити надлишкову інформацію. Однак для різних алгоритмів обсяг цієї надлишкової інформації може істотно відрізнятися.
* **Редагованість.** Під редагованістю розуміється мінімальна ступінь погіршення якості зображення при його повторному збереженні після редагування. Багато алгоритмів з втратою інформації можуть істотно пошкодити зображення за декілька [ітерацій](https://uk.wikipedia.org/wiki/%D0%86%D1%82%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F) редагування.
* **Незначна вартість апаратної реалізації. Ефективність програмної реалізації.** Ці вимоги до алгоритму пред'являють виробники багатьох інформаційних систем. Так, декомпресор фрактального алгоритму дуже ефективно реалізується з використанням технології MMX і розпаралелюванням обчислень, а стиснення за стандартом CCITT Group 3 легко реалізується апаратно.

## Критерії порівняння алгоритмів[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=3) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=3)]

Характеристики алгоритму залежать від конкретних умов, в які буде поставлений алгоритм. Так, ступінь компресії залежить від того, на якому класі зображень алгоритм тестується. Аналогічно, швидкість компресії нерідко залежить від того, на якій платформі реалізований алгоритм. Виділяють кілька *критеріїв порівняння* алгоритмів компресії:

* **Найгірший, середній і найкращий коефіцієнти стиснення.** Тобто частка, на яку зросте зображення, якщо вихідні дані будуть найгіршими; середньостатистичний коефіцієнт для класу зображень, на який орієнтований алгоритм, і, найкращий коефіцієнт. Останній необхідний лише теоретично, оскільки показує ступінь стиснення найкращого (як правило, абсолютно чорного) зображення, іноді фіксованого розміру.
* **Клас зображень, на який орієнтований алгоритм.**
* **Симетричність.** Співвідношення характеристики алгоритму кодування до аналогічної характеристики при декодуванні. Характеризує ресурсомісткість процесів кодування і декодування.
* **Втрати якості.**
* **Характерні особливості алгоритму і зображень, до яких його застосовують.** Тут можуть зазначатися найважливіші для алгоритму властивості, які можуть стати визначальними при його виборі.

## Алгоритми архівації без втрат[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=4) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=4)]

### Алгоритм RLE[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=5) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=5)]

**Алгоритм RLE** *(Run Length Encoding)* — один з найстаріших і найпростіших алгоритмів архівації графіки. Зображення в ньому витягується в ланцюжок [байт](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82) по рядках [растра](https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%82%D1%80). Саме стиснення в RLE відбувається за рахунок того, що у вихідному зображенні зустрічаються *ланцюжки однакових байт*. Алгоритм орієнтований на зображення з невеликою кількістю кольорів: ділову та наукову графіку. До позитивних сторін алгоритму можна віднести тільки те, що він не вимагає додаткової пам'яті при архівації та розархівації, а також швидко працює.

### Алгоритм LZW[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=6) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=6)]

Назву [**алгоритм LZW**](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D0%B5%D0%BC%D0%BF%D0%B5%D0%BB%D1%8F-%D0%97%D1%96%D0%B2%D0%B0) отримав за першими літерами прізвищ його розробників — Lempel, Ziv і Welch. Стиснення в ньому, на відміну від RLE, здійснюється за рахунок *однакових ланцюжків байт*. Процес стиснення виглядає досить просто. Ми зчитуємо послідовно символи вхідного потоку і перевіряємо, чи є у створеній нами таблиці рядків такий рядок. Якщо рядок є, то ми зчитуємо наступний символ, а якщо рядка немає, то ми заносимо в потік код для попередньої знайденої рядки, заносимо рядок в таблицю і починаємо пошук знову. *Алгоритм LZW* орієнтований на 8-бітові зображення. *LZW* є універсальним алгоритмом — саме його варіанти використовуються у звичайних архіваторах.

### Алгоритм Хаффмана[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=7) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=7)]

*Докладніше:* [*Код Хаффмана*](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0)

[Алгоритм Хаффмана](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0) — один з класичних алгоритмів, відомих з 60-х років. Використовує тільки частоту появи однакових байт в зображенні. Зіставляє символам вхідного потоку, які зустрічаються більше число раз, ланцюжок біт меншої довжини. І, навпаки, тим, які зустрічається рідко — ланцюжок більшої довжини. Для збору статистики вимагає двох проходів по зображенню. *Алгоритм Хаффмана* практично не застосовується до зображень у чистому вигляді. Зазвичай використовується як один з етапів стиснення в складніших схемах. Цей алгоритм реалізований у форматі [TIFF](https://uk.wikipedia.org/wiki/TIFF). Він є надзвичайно простим у реалізації, швидким і може бути легко реалізований апаратно.

### JBIG[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=8) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=8)]

**Алгоритм JBIG** розроблений групою експертів [ISO](https://uk.wikipedia.org/wiki/ISO) *(Joint Bi-level Experts Group)* спеціально для стиснення однобітних чорно-білих зображень[[1]](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C#cite_note-1), наприклад, факсів або відсканованих документів. Може також застосовуватися і до 2-х, і до 4-х бітових зображень. При цьому алгоритм розбиває їх на окремі бітові площини. JBIG дозволяє управляти такими параметрами, як порядок розбиття зображення на бітові площини, ширина смуг в зображенні, рівні масштабування.

### Lossless JPEG[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=9) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=9)]

**Алгоритм Lossless JPEG** розроблений групою експертів в області фотографії *(Joint Photographic Expert Group)*. На відміну від JBIG, Lossless JPEG орієнтований на повнокольорові 24-бітні або 8-бітові зображення в градаціях сірого без палітри. Він являє собою спеціальну реалізацію JPEG без втрат. Lossless JPEG рекомендується застосовувати в тих додатках, де необхідно побітова відповідність вихідного і декомпресованого зображень.

## Алгоритми архівації з втратами[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=10) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=10)]

### Алгоритм JPEG[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=11) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=11)]

[**JPEG**](https://uk.wikipedia.org/wiki/JPEG) — один з найновіших і досить потужних алгоритмів. Практично він є стандартом де-факто для повнокольорових зображень[[2]](https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C#cite_note-2). Оперує алгоритм областями 8х8, на яких яскравість і колір змінюються порівняно плавно. Внаслідок цього, при розкладанні матриці такої області в подвійний ряд косинусів значущими виявляються тільки перші коефіцієнти. Таким чином, стиснення в JPEG здійснюється за рахунок плавності зміни кольорів у зображенні. Алгоритм розроблений групою експертів в області фотографії спеціально для стиснення 24-бітових зображень. JPEG — *Joint Photographic Expert Group* — підрозділ у рамках ISO — [Міжнародної організації зі стандартизації](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%B6%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D0%B0_%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F_%D0%B7%D1%96_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%97). В цілому алгоритм заснований на [дискретному косинусному перетворенні](https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B5_%D0%BA%D0%BE%D1%81%D0%B8%D0%BD%D1%83%D1%81%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F) (надалі ДКП), що застосовується до матриці зображення для отримання деякої нової матриці коефіцієнтів. Для отримання початкового зображення застосовується зворотне перетворення.

### Фрактальний алгоритм[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=12) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=12)]

*Докладніше:* [*Алгоритм фрактального стиснення*](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%84%D1%80%D0%B0%D0%BA%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D1%81%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F)

Фрактальна архівація заснована на тому, що зображення представляють в компактнішій формі — з допомогою коефіцієнтів [системи ітерованих функцій](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%96%D1%82%D0%B5%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%85_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%B9) (*Iterated Function System — IFS*). IFS являє собою набір тривимірних [афінних перетворень](https://uk.wikipedia.org/wiki/%D0%90%D1%84%D1%96%D0%BD%D0%BD%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F), які переводять одне зображення в інше. Перетворенню підлягають точки в тривимірному просторі (х\_координата, у\_координата, яскравість).

Найвідоміші два зображення, отриманих за допомогою IFS: [«трикутник Серпінського»](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D0%BA%D1%83%D1%82%D0%BD%D0%B8%D0%BA_%D0%A1%D0%B5%D1%80%D0%BF%D1%96%D0%BD%D1%81%D1%8C%D0%BA%D0%BE%D0%B3%D0%BE) і «папороть Барнслі». «Трикутник Серпінського» задається трьома, а «папороть Барнслі» чотирма афінними перетвореннями. Кожне перетворення кодується кількома байтами, в той час як зображення, побудоване з їх допомогою, може займати і декілька мегабайт.

Алгоритм використовують при стисненні повнокольорових 24 бітних зображень або зображень у градаціях сірого без різких переходів кольорів (фотографії). Бажано, щоб області більшої значимості (для сприйняття) були більш контрастними і різкими, а області меншої значимості — неконтрастними і розмитими.

### Рекурсивний (хвильовий) алгоритм[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=13) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=13)]

Англійська назва **рекурсивного стиснення** — [wavelet](https://uk.wikipedia.org/wiki/Wavelet). Цей вид архівації відомий досить давно і безпосередньо виходить з ідеї використання [когерентності](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B3%D0%B5%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C) областей. Орієнтований алгоритм на кольорові і чорно-білі зображення з плавними переходами. Ідеальний для картинок типу рентгенівських знімків. Коефіцієнт стиснення задається і варіюється в межах 5-100. До переваг цього алгоритму можна віднести те, що він дуже легко дозволяє реалізувати можливість поступового «проявлення» зображення при передачі зображення по мережі. На відміну від JPEG і фрактального алгоритму даний метод оперує блоками 2х2, 4х4, 8х8 і т. д. За рахунок того, що коефіцієнти для цих блоків зберігаються незалежно, можна досить легко уникнути дроблення зображення на «мозаїчні» квадрати.

## Параметри різних алгоритмів стиснення зображень[[ред.](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&veaction=edit&section=14) | [ред. код](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B8%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D1%8C&action=edit&section=14)]

| **Алгоритм** | **Коефіцієнти стиснення** | **Симетричність за часом** | **На що орієнтований** | **Втрати** | **Розмірність**  |
| --- | --- | --- | --- | --- | --- |
| **RLE** | 32, 2, 0.5 | 1 | 3,4-х бітні | Немає | 1D  |
| **LZW** | 1000, 4, 5 / 7 | 1.2-3 | 1-8 бітні | Немає | 1D  |
| **Хаффмана** | 8, 1.5, 1 | 1-1.5 | 8 бітними | Немає | 1D  |
| **CCITT-3** | 213 (3), 5, 0.25 | ~ 1 | 1-бітні | Немає | 1D  |
| **JBIG** | 2-30 разів | ~ 1 | 1-бітні | Немає | 2D  |
| **Lossless JPEG** | 2 рази | ~ 1 | 24-бітові, сірі | Немає | 2D  |
| **JPEG** | 2-20 разів | ~ 1 | 24-бітові, сірі | Є | 2D  |
| **Рекурсивний стиск** | 2-200 разів | 1.5 | 24-бітові, сірі | Є | 2D  |
| **Фрактальний** | 2-2000 разів | 1000-10000 | 24-бітові, сірі | Є | 2.5D  |