

## **ДОСЛІДЖЕННЯ МЕТОДІВ ФРАКТАЛЬНОГО СТИСНЕННЯ ЗОБРАЖЕНЬ З ВИМІРЮВАЛЬНОЮ ІНФОРМАЦІЄЮ В КОМП'ЮТЕРИЗОВАНИХ СИСТЕМАХ УПРАВЛІННЯ**

Завдання збереження і передачі інформації в компактному вигляді завжди була актуальною в інформатиці. І якщо стосовно текстової інформації розроблені досить ефективні методи стиснення даних, то розробки якісної компресії статичних зображень і динамічної відеоінформації тільки набирають оберті. На сьогоднішній день існують і широко застосовуються стандартні методи стиснення. Однак потреба в зберіганні все більших обсягів інформації і бажання передачі її по каналах зв'язку з максимальною швидкістю зумовили інтерес до дослідження і розробки більш досконалих методів.

Всі існуючі алгоритми можна розділити на два великі класи:

- алгоритми стиснення без втрат;
- алгоритми стиснення з втратами.

Коли ми говоримо про стиснення без втрат, ми маємо на увазі, що існує алгоритм, зворотний алгоритму стиснення, що дозволяє точно відновити вихідне зображення. Для алгоритмів стиснення з втратами зворотного алгоритму не існує. Існує алгоритм, який відновлює зображення, яке не обов'язково точно збігається з вихідним. Алгоритми стиснення і відновлення підбираються так, щоб домогтися високого ступеня стиснення і при цьому зберегти візуальну якість зображення.

Алгоритм фрактального стиснення зображення відносять до алгоритмів архівації з частковою втратою інформації. Основа методу фрактального кодування – це виявлення самоподібних ділянок в зображенні. Ідея компресії зображення заснована на застосуванні систем ітераційних функцій (Iterated Function System або IFS).

Процес стиснення:

1. Поділ зображення на області, які неперекриваються (домени). Набір доменів повинен покривати все зображення повністю.
2. Вибір рангових областей. Рангові області можуть перекриватися і не покривати цілком все зображення.
3. Фрактальне перетворення: для кожного домена підбирається така рангова область, яка після афінного перетворення найбільш точно апроксимує домен.
4. Стиснення і збереження параметрів афінного перетворення. У файл записується інформація про розташування доменів і рангових областей, а також стислі коефіцієнти афінних перетворень.

Етапи відновлення зображення:

1. Створення двох зображень однакового розміру А і В. Розмір і зміст областей не мають значення.
2. Зображення В ділиться на домени так само, як і на першій стадії процесу стиснення. Для кожного домена області В проводиться відповідне афінне перетворення рангових областей зображення А, описане коефіцієнтами з стисненого файлу. Результат поміщається в область В. Після перетворення виходить зовсім нове зображення.
3. Перетворення даних з області В в область А. Цей крок повторює крок 3, тільки зображення А і В помінялися місцями.
4. Кроки 3 і 4 повторюються до тих пір, поки зображення А і В не стануть нерозрізненними.

Даний алгоритм стиснення потенційно здатний забезпечити найкраще співвідношення ступеня стиснення і якості відновленого зображення. Застосування фрактального алгоритму дозволяє інколи стискати зображення в сотні і навіть тисячі разів. Точність отриманого зображення залежить від точності афінного перетворення.

Відмінною особливістю фрактального стиснення є його яскраво виражена асиметрія. Алгоритми стиснення і відновлення істотно розрізняються (стиснення вимагає набагато більшої кількості обчислень).

Складність алгоритмів фрактального стиснення в тому, що використовується цілочисельна арифметика і спеціальні досить складні методи, що зменшують помилки округлення. Для знаходження відповідних доменних блоків, потрібен повний перебір. Оскільки при цьому переборі кожен раз повинні порівнюватися два масиви, дана операція виходить досить тривалою, що вимагає значних обчислювальних ресурсів.

В геоінформаційних системах, де проблемою є великі розміри зображень, а також в електронних картотеках і дось різних служб є характерна схожість між зображеннями. Тому практичне застосування даного алгоритму стиснення можна знайти в зберіганні різних баз даних із зображеннями, де без використання спеціальних алгоритмів більшу частину пам'яті займають фотографії.