Обробка зображень і мультимедіа

Олег Гутік

Лекція 10: Стиснення зображень, IV. Інтуїтивні методи

Стиснення зображень. Інтуїтивні методи

Легко вигадати прості інтуїтивні методи стиснення зображень. Ми наведемо їх тут лише з ілюстративною метою. На практиці, звичайно, використовуються витонченіші (і ефективніші) методи компресії зображень.

Стиснення зображень. Інтуїтивні методи

Легко вигадати прості інтуїтивні методи стиснення зображень. Ми наведемо їх тут лише з ілюстративною метою. На практиці, звичайно, використовуються витонченіші (і ефективніші) методи компресії зображень.

Стиснення зображень. Інтуїтивні методи

Легко вигадати прості інтуїтивні методи стиснення зображень. Ми наведемо їх тут лише з ілюстративною метою. На практиці, звичайно, використовуються витонченіші (і ефективніші) методи компресії зображень.

Стиснення зображень. Інтуїтивні методи

Легко вигадати прості інтуїтивні методи стиснення зображень. Ми наведемо їх тут лише з ілюстративною метою. На практиці, звичайно, використовуються витонченіші (і ефективніші) методи компресії зображень.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Підвибірка, можливо, є найпростішим методом стиснення зображення. Найпростіший спосіб підвибірки — це просто відкинути деякі пікселі. Кодер може, наприклад, ігнорувати кожен другий рядок і кожен другий стовпець зображення і записувати пікселі, що залишилися, в стислий файл. Це становитиме 25% від вихідного. Декодер вводить стислі дані та використовує кожен піксель для створення чотирьох однакових пікселів реконструйованого зображення. Це, звичайно, призводить до втрати багатьох деталей зображення. Такий метод рідко призводить до задовільних результатів. Зауважимо, що для такого методу коефіцієнт стиснення відомий заздалегідь.

Більш прийнятні результати можна отримати, якщо записувати в стислий файл середнє кожного блоку із 4 пікселів вихідного образу. При цьому жоден піксель не ігнорується, але метод, як і раніше, вкрай примітивний, оскільки хороший метод стиснення повинен відкидати деталі, непомітні для ока.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Підвибірка

Найкращі результати (але гірше стиснення) виходять, якщо кольорове зображення образу трансформується зі звичайного (наприклад, RGB) у зображенні з компонентами світимості і кольоровості. Кодер робить вибірку у двох компонентах кольоровості, а компоненту яскравості залишає недоторканою. Якщо вважати, що всі компоненти використовують одну й ту кількість біт, то дві компоненти кольоровості займають $2/3$ розміру вихідного файла. Зробивши вибірку, скорочуємо цей розмір до 25% від $2/3$, тобто до $1/6$. Тоді розмір стисненого файлу складатиметься з $1/3$ (для стиснутої компоненти світності) плюс $1/6$ (для двох компонентів кольоровості), що дорівнює $1/2$ початкового розміру.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

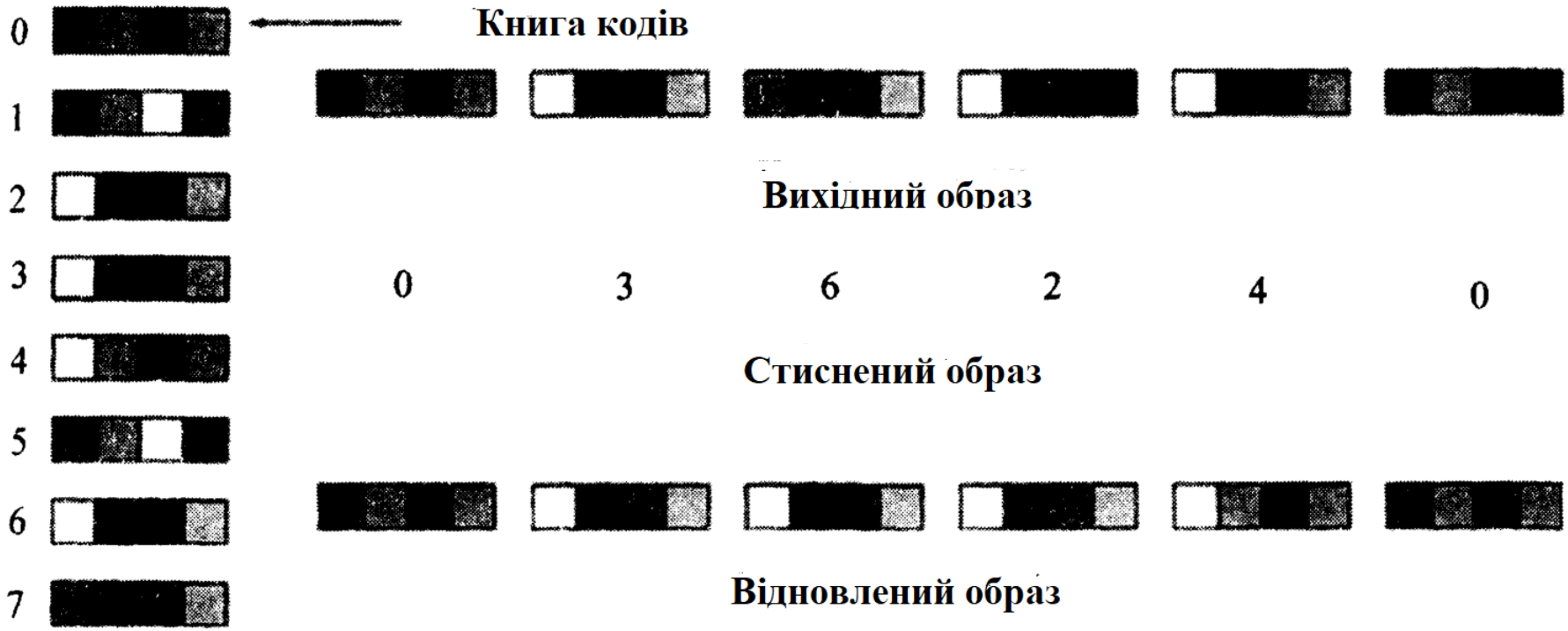
Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування

Термін “квантування” при використанні в стисненні даних означає заокруглення дійсних чисел до цілих або перетворення цілих чисел у менші цілі. Існує два види квантування, скалярне та векторне. Скалярне квантування є інтуїтивним методом, у якому завжди втрачається лише незначна інформація. При використанні другого методу можна досягти кращих результатів, тому ми наводимо його нижче.

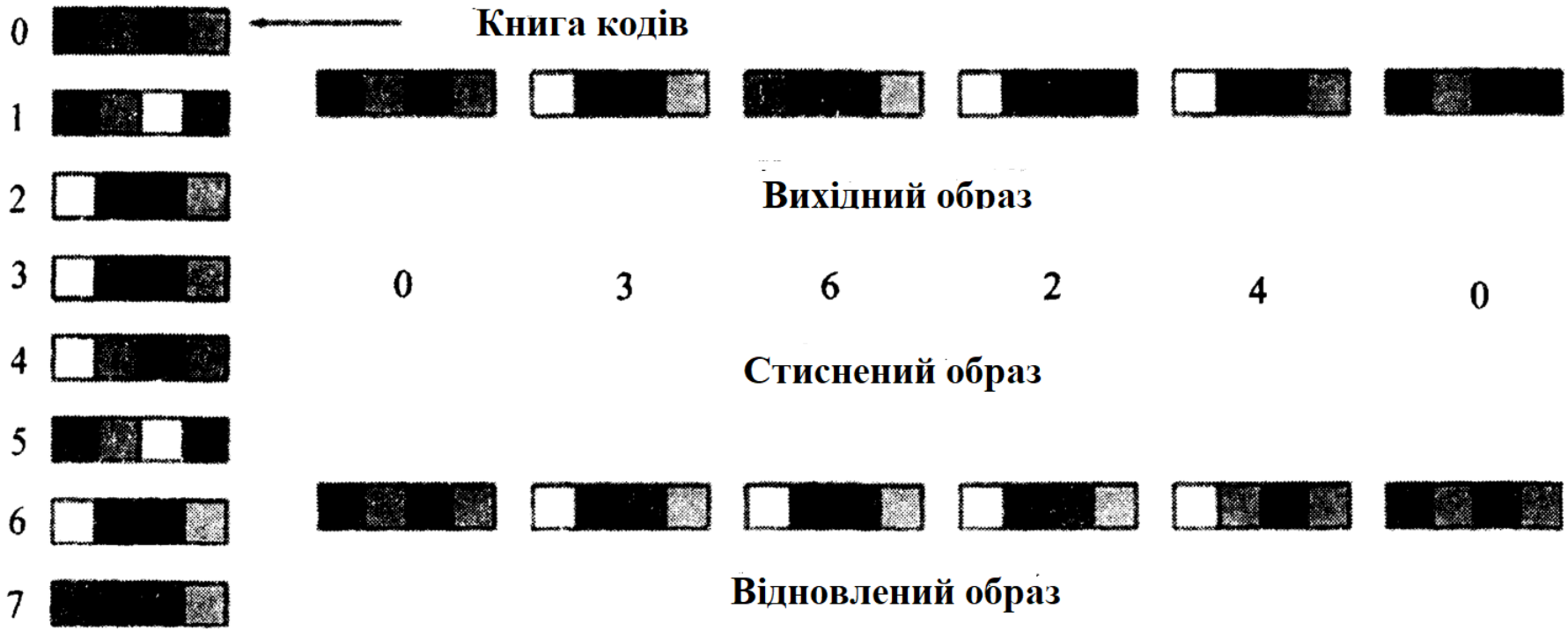
Зображення ділиться на рівні блоки пікселів, які називаються *векторами*, а кодер має список таких ж блоків, який називається *книгою кодів*. Кожен блок B зображення порівнюється з усіма блоками з кодової книги та розташований “найближчий” до блоку B блок C. Після чого у вихідний файл записується покажчик на блок C. Якщо розмір покажчика менше розміру блоку, то досягається стиск.

Стиснення зображень. Квантування



На рис. зображено приклад такої діяльності.

Стиснення зображень. Квантування



На рис. зображено приклад такої діяльності.

Дякую за увагу!