Обробка зображень і мультимедіа

Олег Гутік

Лекція 22: Стиснення зображень, XVI. JPEG. DCT

Стиснення зображень. JPEG. DCT

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого) близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого) близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Стиснення зображень. JPEG. DCT

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градації сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градації сірого), близькі до значень найближчих пікселів, але далекі пікселі майже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градації сірого), близькі до значень найближчих пікселів, але далекі пікселі вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Стиснення зображень. JPEG. DCT

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градації сірого) близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Стиснення зображень. JPEG. DCT

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Дискретне косинус-перетворення (DCT) вже обговорювалося нами у попередніх лекціях. Комітет JPEG зупинив свій вибір саме на цьому перетворенні через його добрі властивості, а також через те, що в ньому не робиться жодних обмежень на структуру даних, що стискаються. Крім того, є можливості для прискорення DCT.

Стандарт JPEG застосовує DCT не до всього зображення, а до одиниць даних (блоків) розміру 8×8 пікселів.

(1) Справа в тому, що застосування DCT до всього зображення використовує велику кількість арифметичних операцій і тому виконується повільно. Застосування DCT до одиниць даних обчислюється значно швидше.

(2) З дослідів відомо, що у неперервно-тонових зображеннях кореляція пікселів зберігається у малих областях. Пікселі такого зображення мають величини (компоненти кольору або градацію сірого), близькі до значень навколишніх пікселів, але далекі сусіди вже не мають кореляції.

Отже, застосування DCT до всього зображення не зробить кращий стиск.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \dots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \dots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT.

Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \dots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \tag{1}$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT.
Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG “викидає” незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Однак слід зазначити, що використання малих одиниць даних має зворотний бік. Застосування DCT до всього зображення (з наступним скороченням інформації та декомпресією) створює картину, більш приємну для ока. Виконання DCT над одиницям даних робиться істотно швидше, але після компресії та декомпресії можлива поява блокових артефактів (штучних спотворень), особливо на стиках блоків, оскільки різні блоки по-різному взаємодіють з етапом квантування, який слідує за DCT. Формули DCT для JPEG збігаються з (1). Ми їх повторюємо для зручності викладу:

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (1)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Перетворення DCT є основою стиснення втрати інформації в стандарті JPEG. Метод JPEG "викидає" незначну частину інформації із зображення з допомогою] поділу кожного з 64 коефіцієнтів DCT (особливо ті, які розташовані в правій нижній частині блоку) на коефіцієнт квантування QC. У звичайному випадку кожен коефіцієнт DCT ділиться на особливий коефіцієнт QC, але всі 64 параметри QC можуть змінюватися на розсуд користувача.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \dots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Декодер JPEG обчислює обернене перетворення DCT (IDCT) з допомогою рівнянь (2), які ми тут повторюємо

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2n}\right)\cos\left(\frac{(2x+1)i\pi}{2n}\right), \qquad (2)$$

де

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & \text{якщо } f = 0; \\ 1, & \text{якщо } f = 1, 2, \ldots, 7. \end{cases}$$

Тут використовуються квантовані коефіцієнти DCT, а в результаті виходять значення пікселів p_{xy}. Кажучи мовою математики, перетворення DCT є взаємно однозначним лінійним відображенням 64-вимірного векторного простору зображення в простір частот тієї ж розмірності. IDCT є оберненим відображенням. Якщо були б можливі обчислення з абсолютною точністю, то після застосування DCT та IDCT (без квантування) результат збігся б із вихідним блоком. На практиці, звичайно, використовується квантування, але якщо його робити акуратно, то результатом цих перетворень буде блок, який дуже близький до вихідного блоку зображення.

Дякую за увагу!