# Обробка зображень і мультимедіа

Олег Гутік


## Лекція 23: Стиснення зображень, XVII. JPEG. Практичне DCT

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

Рівняння (1)

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

легко перекладаються будь-якою мовою програмування високого рівня. Проте, є кілька можливостей істотного прискорення обчислення цих величин. Ці формули лежать у "серці" методу JPEG, а тому прискорення обчислень просто необхідне. Опишемо декілька корисних прийомів.

# Стиснення зображень. JPEG. Практичне DCT

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

1. Незалежно від розміру зображення, використовується лише $32$ значення функції косинус (див. наступний абзац). Їх можна один раз обчислити та використовувати багато разів в операціях над одиницями даних $8 \times 8$. Тоді обчислення виразу

$$p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

складатиметься всього з двох операцій множення. Подвійна сума (1) вимагатиме $64 \times 2 = 128$ множень та $63$ додавань.

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

Аргументи функції косинус, які у DCT, мають вигляд $\frac{(2x+1)i\pi}{16}$, де $x$ та $i$ — цілі числа з інтервалу $[0, 7]$. Їх можна записати у вигляді $\frac{n\pi}{16}$, де $n$ — ціле число з інтервалу $[0, 15 \cdot 7]$. Оскільки функція косинус періодична, то для неї $\cos(32\pi/16) = \cos(0\pi/16)$, $\cos(33\pi/16) = \cos(1\pi/16)$, тощо. У результаті буде потрібно лише $32$ значення $\cos(n\pi/16)$ при $n = 0, 1, \ldots, 31$.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $CPC^T$, де $P$ — вихідна $8 \times 8$ матриця пікселів, матриця $C$ визначається за формулами

$$C_{ij} = \begin{cases} \frac{1}{\sqrt{8}}, & \text{якщо } i = 0, \\ \frac{1}{2} \cos\left(\frac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 7, \end{cases}$$

і $C^T$ — транспонована матриця $C$.

У результаті, обчислення одного елемента матриці $P$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $C$ і $P$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $CP$ на $C^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множеннння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $P$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $C$ і $P$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $CP$ на $C^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

2. Аналіз подвійної суми (1) дозволяє переписати її у вигляді добутку матриць $\boldsymbol{CPC}^T$, де $\boldsymbol{P}$ — вихідна $8 \times 8$ матриця пікселів, матриця $\boldsymbol{C}$ визначається за формулами

$$C_{ij} = \begin{cases} \dfrac{1}{\sqrt{8}}, & \text{якщо } i = 1; \\ \dfrac{1}{2} \cos\left(\dfrac{(2j+1)i\pi}{16}\right), & \text{якщо } i = 1, \ldots, 31, \end{cases}$$

і $\boldsymbol{C}^T$ — транспонована матриця $\boldsymbol{C}$.

У результаті, обчислення одного елемента матриці $\boldsymbol{P}$ вимагає восьми множень і семи (ну нехай теж восьми, для простоти) додавань. Множення двох матриць $\boldsymbol{C}$ і $\boldsymbol{P}$ складається з $64 \cdot 8 = 8^3$ операцій множенння і стільки ж операцій додавання. Множення $\boldsymbol{CP}$ на $\boldsymbol{C}^T$ вимагатиме такої ж кількості операцій, а отже, одне DCT одиниці даних складається з $2 \cdot 8^3$ операцій множення (і додавання). Якщо вихідне зображення складається з $n \times n$ пікселів, причому $n = 8q$, де $q$ — кількість одиниць даних, то для обчислення DCT всіх цих одиниць знадобиться $2q^2 8^3$ множень (і стільки ж додавань). Для порівняння, обчислення одного DCT всього зображення вимагатиме

$$2n^3 = 2q^3 8^3 = (2q^2 8^3)q$$

операцій. За допомогою поділу зображення на одиниці даних ми скоротили загальну кількість множень (і додавань) у $q$ разів. На жаль, число $q$ не може бути надто великим, оскільки це зменшує розмір одиниці даних.

# Стиснення зображень. JPEG. Практичне DCT

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

Нагадаємо, що кольорове зображення складається з трьох компонентів (зазвичай це RGB, яке перетворюється на YCbCr або YPbPr). Кожна компонента перетворюється окремо, що дає загальну кількість операцій, що дорівнює $3 \cdot 2q^2 8^3 = 3072q^2$. Для зображення з роздільною здатністю $512 \times 512$ пікселів потрібно зробити

$$3072 \cdot 64^2 = 12582912$$

множень (і додавань).

3. Інший шлях прискорення DCT полягає у виконанні арифметичних обчислень над числами, представленими у формі з фіксованою точкою, а не у формі з плаваючою точкою. Для багатьох комп'ютерів операції над числами з фіксованою точкою робляться значно швидше за операції з плаваючою точкою.

# Стиснення зображень. JPEG. Практичне DCT

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього $54$ операцій множення та $468$ операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього $11$ операцій множення та $29$ операцій додавання.

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього 54 операцій множення та 468 операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього 11 операцій множення та 29 операцій додавання.

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього $54$ операцій множення та $468$ операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього $11$ операцій множення та $29$ операцій додавання.

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього $54$ операцій множення та $468$ операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього $11$ операцій множення та $29$ операцій додавання.

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього $54$ операцій множення та $468$ операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього $11$ операцій множення та $29$ операцій додавання.

Безперечно, найкращий алгоритм для DCT описаний у [E. N. Feig and E. Linzer, *Discrete cosine transform algorithms for image data compression*, in Proceedings Electronic Imaging'90 East, Boston, MA, 1990, pp. 84–87]. Для нього потрібно всього $54$ операцій множення та $468$ операцій додавання та зсувів. На сьогоднішній день є різні спеціалізовані мікросхеми, які виконують ці процедури дуже ефективно. Читач, що цікавиться, може познайомитися в [D. A. Leiewer and D. S. Hirschberg, *Data compression*, Computing Surveys **19** (1987), no. 3, 261–297] зі швидким алгоритмом одновимірного DCT, що використовує всього $11$ операцій множення та 29 операцій додавання.

# Дякую за увагу!