2D-DCT

313553014 Yi-Cheng Liao

1 Introduction

1.1 Two-Dimensional Discrete Cosine Transform (2D-DCT)

The two-dimensional Discrete Cosine Transform (2D-DCT) converts an image from the spatial domain to the frequency domain by representing it as a weighted sum of cosine basis functions. For an input image $f(x, y)$ of size $N \times N$, the 2D-DCT is mathematically defined as:

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right],$$

where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}}, & w = 0, \\ 1, & \text{otherwise.} \end{cases}$$

The coefficients $F(u, v)$ represent the frequency content of the image. Low-frequency components (small u, v) are concentrated in the top-left corner of the DCT matrix, while high-frequency components are located toward the bottom-right. In image compression applications such as JPEG, many of the high-frequency coefficients can be discarded without introducing significant perceptual degradation.

The inverse transform (2D-IDCT) reconstructs the spatial-domain image from the DCT coefficients as:

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right].$$

1.2 Two 1D DCT

The two-dimensional Discrete Cosine Transform (2D-DCT) is a *separable transform*, which means that it can be expressed as a combination of two successive one-dimensional DCTs (1D-DCTs) applied along the rows and columns of the image. This property allows the 2D transform to be computed more efficiently without changing its mathematical result.

Starting from the 2D-DCT definition:

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right],$$

we observe that the cosine terms in x and y are separable. Thus, the computation can be decomposed into two sequential 1D transforms.

First, apply a 1D-DCT along each row of the image:

$$g(u,y)=\sqrt{\frac{2}{N}}\,C(u)\sum_{x=0}^{N-1}f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right),$$

and then apply another 1D-DCT along the columns of the intermediate result:

$$F(u,v)=\sqrt{\frac{2}{N}}\,C(v)\sum_{y=0}^{N-1}g(u,y)\cos\left(\frac{(2y+1)v\pi}{2N}\right).$$

By substituting $g(u,y)$ into the second equation, we obtain:

$$F(u,v)=\left(\sqrt{\frac{2}{N}}\,C(u)\right)\left(\sqrt{\frac{2}{N}}\,C(v)\right)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right),$$

which is identical to the direct 2D-DCT formula.

Hence, the 2D-DCT can be computed equivalently as:

$$F=DCT_{\text{cols}}(DCT_{\text{rows}}(f))\,.$$

Similarly, the inverse transform (2D-IDCT) can be expressed using two successive 1D-IDCTs:

$$h(x,v)=\sqrt{\frac{2}{N}}\sum_{u=0}^{N-1}C(u)\,F(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right),$$

$$f(x,y)=\sqrt{\frac{2}{N}}\sum_{v=0}^{N-1}C(v)\,h(x,v)\cos\left(\frac{(2y+1)v\pi}{2N}\right).$$

where the normalization coefficient $C(u)$ is defined as:

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}}, & u=0,\\ 1, & \text{otherwise.}\end{cases}$$

Both methods — direct 2D-DCT and two sequential 1D-DCTs — yield identical frequency coefficients. However, the two-step implementation is computationally more efficient, reducing the complexity from $\mathcal{O}(N^4)$ for the naive 2D computation to $\mathcal{O}(N^3)$ for the separable version, or further to $\mathcal{O}(N^2\log N)$ when using fast DCT algorithms.

2 Results

2.1 Comparison of Runtime Between Direct 2D-DCT and Two 1D-DCT

To reduce the computational load during transformation, the input *Lena* image was resized from 512×512 pixels to 128×128 pixels before performing the Discrete Cosine Transform (DCT). This experiment compares the runtime and reconstruction quality between the conventional direct 2D-DCT and the separable Two 1D-DCT implementations.

Table 1 summarizes the performance comparison between the direct 2D-DCT and the separable Two 1D-DCT methods. The direct 2D-DCT produces extremely high numerical precision, achieving

Figure 1: Results of the direct 2D-DCT. The left image shows the original Lena image, the middle image visualizes the DCT coefficients in the logarithmic domain, and the right image represents the reconstructed result obtained after the inverse 2D-DCT.

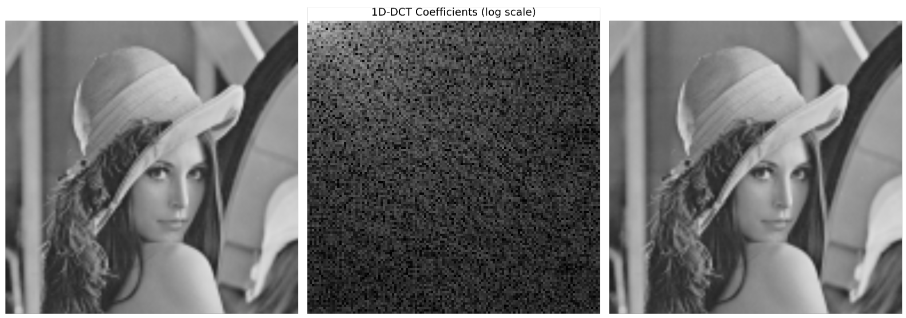

Figure 2: Results of the separable Two 1D-DCT implementation. The left image shows the original Lena image, the middle image visualizes the intermediate coefficients in the log domain, and the right image shows the reconstructed result obtained after applying two successive 1D-DCTs (row-wise and column-wise).

a PSNR of 285.61 dB. However, it is computationally expensive, taking 475.93 seconds to process a 128×128 image, mainly due to the $\mathcal{O}(N^4)$ complexity arising from the double summation over all pixel coordinates.

Table 1: Performance comparison between the direct 2D-DCT and separable Two 1D-DCT implementations.

Method	**Computation Time (s)**	**PSNR (dB)**
Direct 2D-DCT	475.93	285.61
Two 1D-DCT (Separable)	2.40	157.35

In contrast, the Two 1D-DCT approach takes advantage of the separability property of the cosine basis functions. By performing two sequential 1D transforms along the rows and columns, the computational complexity is reduced to $\mathcal{O}(N^3)$. This optimization decreases the runtime to only 2.40 seconds — over 200 times faster than the direct 2D-DCT implementation. Although the PSNR slightly decreases to 157.35 dB, the reconstructed image remains visually indistinguishable from the original, confirming that the separable DCT provides a much more efficient and practical implementation without noticeable degradation in quality.

Overall, the results demonstrate that exploiting the separable nature of the DCT greatly enhances computational efficiency while maintaining acceptable reconstruction quality. This approach serves as the foundation for most practical implementations of DCT in modern image compression systems such as JPEG.

2.2 Supplementary: Validation Using OpenCV DCT/IDCT

To verify the correctness of the manual implementations of both 2D-DCT and Two 1D-DCT, the same experiment was repeated using OpenCV's built-in `cv2.dct` and `cv2.idct` functions. These functions perform type-II DCT and its inverse (IDCT-II) through highly optimized FFT-based routines implemented in C++.

The input image was first converted to `float32` and normalized to the range [0, 1]. Subsequently, the 2D-DCT and IDCT were applied sequentially to obtain the reconstructed image, and the runtime as well as the PSNR were recorded for comparison.

The OpenCV implementation achieved a DCT computation time of approximately **0.001 seconds** and an IDCT time of **0.0003 seconds**, resulting in a reconstruction PSNR of **143.81 dB**. Compared with the manual implementations (475.93 seconds for direct 2D-DCT and 2.40 seconds for Two 1D-DCT), the OpenCV version is several orders of magnitude faster due to vectorized matrix operations and hardware-level optimizations.

Although the PSNR is slightly lower than that of the manually implemented methods, the reconstructed image remains visually identical to the input, confirming that both custom DCT implementations are numerically correct and consistent with the standard OpenCV reference. This validation confirms that the separable Two 1D-DCT produces mathematically equivalent results to the direct 2D-DCT formulation.

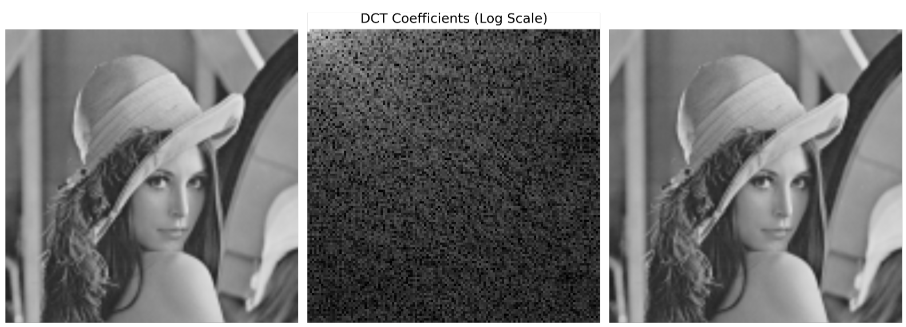

Figure 3: opencv result