The discrete cosine transform (DCT) is used in several compression applications (e.g., JPEG image compression and MPEG video compression). The DCT is defined by

\[ X(k) = \frac{2}{\sqrt{N}} \alpha(k) \sum_{n=0}^{N-1} x(n) \cos \left( \frac{(2n+1)k\pi}{2N} \right), k = 0, ..., N - 1 \]

where \( \alpha(0) = \frac{1}{\sqrt{2}} \) and \( \alpha(k) = 1, k \neq 0 \). The inverse transform is

\[ x(n) = \frac{2}{\sqrt{N}} \sum_{k=0}^{N-1} \alpha(k) X(k) \cos \left( \frac{(2n+1)k\pi}{2N} \right), n = 0, ..., N - 1 \]

The forward transform can be written as \( X = Ax \), where \( x \) and \( X \) are column vectors and \( A \) is an \( N \) by \( N \) real-valued matrix. The rows of this matrix are interpreted as the basis vectors for the DCT. For \( N = 8 \), the matrix \( A \) is listed below.

\[
\begin{bmatrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\
0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\
0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\
0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\
0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\
0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\
0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \\
\end{bmatrix}
\]

The matrix \( A \) is unitary, meaning that the rows and columns are unit energy (unit norm) and that \( A^{-1} = A^T \). One consequence is that the energy in \( x \) is the same as the energy in \( X \).

For highly positively correlated random processes (a reasonable model for voiced speech or the rows or columns in a typical natural image scene) the DCT has a remarkable property: it allows significant decorrelating of the data. For example, let \( x = [1, a, a^2, a^3, ..., a^{N-1}]^T \), where \( 0 \leq a \leq 1 \). Then
applying the DCT yields $y = Ax$, where most of the energy in $x$ is localized into just a few components of $y$. For example, letting $a = 0.8, 0.9, \text{ and } 0.95$ yields the following

$$x^T = \begin{bmatrix} 1.0000 & 0.8000 & 0.6400 & 0.5120 & 0.4096 & 0.3277 & 0.2621 & 0.2097 \end{bmatrix}$$

$$y^T = (Ax)^T = \begin{bmatrix} 1.4712 & 0.7079 & 0.1512 & 0.0945 & 0.0359 & 0.0288 & 0.0115 & 0.0073 \end{bmatrix};$$

$$x^T = \begin{bmatrix} 1.0000 & 0.9000 & 0.8100 & 0.7290 & 0.6561 & 0.5905 & 0.5314 & 0.4783, \end{bmatrix}$$

$$y^T = (Ax)^T = \begin{bmatrix} 2.0136 & 0.4771 & 0.0490 & 0.0530 & 0.0111 & 0.0159 & 0.0035 & 0.0040; \end{bmatrix}$$

$$x^T = \begin{bmatrix} 1.0000 & 0.9500 & 0.9025 & 0.8574 & 0.8145 & 0.7738 & 0.7351 & 0.6983, \end{bmatrix}$$

$$y^T = (Ax)^T = \begin{bmatrix} 2.3800 & 0.2772 & 0.0139 & 0.0294 & 0.0031 & 0.0088 & 0.0010 & 0.0022 \end{bmatrix};$$

In each case, $||x||^2 = ||y||^2$, so the energy in a vector is preserved by the transform, but it has been focused into a small number of transform coefficients.

For length $N = 8$, the 8 basis functions in the DCT (the eight rows of the matrix previously listed) are shown below. Note the increase in frequency as the DCT index, $k$, increases. The DCT coefficients, $X(k)$, can then be interpreted as representing different frequency components in the original signal. In particular, since the basis signal for $k = 0$ is constant, the $X(0)$ coefficient is typically interpreted as being the “dc” part of the signal.
One application of the DCT is in image and video coding. In JPEG image compression, the image is partitioned into 8x8 blocks, and a two-dimensional DCT applied to each block. (The two-dimensional DCT is separable, and corresponds to computing a one-dimensional DCT along each row, and then computing the one-dimensional DCT along each resulting column.)

As an example, consider the 512x512 24-bit (RGB) image shown below (mandrill). The image has bright colors and much detail and texture, and requires a fairly large bit-rate to encode with acceptable quality.

A color image typically consists of three separate 2-dimensional arrays, one each for the red, green, and blue color components. Each array is of size N rows by M columns, and each array entry is referred to as a picture element (pixel) with unsigned integer value. An 8-bit color plane then has pixel values in the range 0 to 255, with the largest value corresponding to the highest intensity of the color, and the smallest value (zero) corresponding to a complete absence of the color. A monochrome, or grayscale, image is generated by combining equal amounts of the red, green, and blue; that is, for each image pixel, by simply averaging the corresponding red, green, and blue values. The 512x512, 8-bit monochrome (mandrill) image is shown below.
To simplify the display, let’s focus on the 256x256 portion of the mandrill image shown below, denoted here as “mandeyes”.

Next, let’s compute the 8x8 DCT on each block of size 8x8 in the image. The resulting block DCT image is shown below.

The DCT coefficients have been rescaled for 8-bit (grayscale) display in the image above. After examining the image for a while, the general outline of the mandrill eyes and nose can be seen in the image. The dc coefficients of each 8x8 DCT block contain the average gray-level of the block, and can be used to extract a low-resolution of the image. These dc coefficients are the light pixels in the top left corner of each 8x8 block.
For comparison, full-size mandrill image after 8x8 block DCT (above) and reconstruction using dc coefficients only (below).
Returning to the “mandeyes” image, extracting the dc coefficients and assembling as a 32x32 image, with rescaling to the range [0, 255], the resulting image is shown above at right. The DCT supports low-resolution “thumbnail” image reconstruction.

Next, let’s extract a small portion of the mandrill image and study the effect of quantization of the DCT coefficients. A 32x32 portion of the image (just below the eye) is shown below. The 8x8 block DCT is shown next to it.

![8x8 block](image)

The DCT coefficients for the top, right 8x8 block of pixels in the image above are listed below.

\[
\begin{array}{cccccccc}
461.375 & -95.0127 & -7.3506 & 54.6144 & -64.8750 & 33.0798 & 15.8981 & -14.2220 \\
-30.3943 & 19.9429 & -0.9922 & 24.4397 & 15.8965 & 15.1920 & 2.9058 & -0.4091 \\
25.5334 & 8.9818 & 1.2920 & -2.8692 & -8.3986 & 0.5153 & -2.3825 & 15.0044 \\
\end{array}
\]

Note that the dc coefficient (upper left) is much larger in magnitude than the other values. Also, the larger magnitude coefficients tend to be localized in the upper left of the 8x8 array.

Moving the 8x8 block to the left (so, the second 8x8 block in the top portion of the 32x32 image) yields the DCT coefficients listed below.

\[
\begin{array}{cccccccc}
-413.5498 & 4.8619 & 25.5318 & -0.0452 & 23.2702 & 27.3147 & -2.0194 & -3.3765 \\
\end{array}
\]
This portion of the image is much lighter in gray-level than the first, and so the dc coefficient is significantly larger.

Now, continuing with this second 8x8 block, suppose these DCT coefficients are quantized using a uniform (odd number of levels) scalar quantizer (with odd number of quantization levels, so there is a zero level in the set of possible outputs) and select a step size of 25. The resulting levels are listed below.

\[
\begin{array}{cccccccc}
950 & 100 & 25 & 0 & 0 & 25 & 25 & 25 \\
-425 & 0 & 25 & 0 & 25 & 25 & 0 & 0 \\
50 & -50 & 0 & 0 & 0 & 0 & 0 & 25 \\
75 & 25 & 0 & 0 & 0 & 0 & 0 & 0 \\
25 & 50 & 25 & 0 & 0 & 25 & 0 & 0 \\
25 & 50 & 0 & 25 & 0 & 0 & 0 & 0 \\
0 & 25 & 0 & 0 & 0 & 0 & 25 & 0 \\
-25 & 25 & 0 & -25 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Normalizing these values by the quantizer step size yields the following.

\[
\begin{array}{cccccccc}
38 & 4 & 1 & 0 & 0 & 1 & 1 & 1 \\
-17 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
2 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \\
3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

In an image compression application (e.g., JPEG) the normalized quantized coefficients are the signed integers, with the dc coefficient tending to be large, and the larger values of the other coefficients tending to be in the upper left corner of the 8x8 block of quantized coefficients. The coefficients are typically accessed in a zig-zag order, and the sequence of signed integers losslessly encoded. (JPEG uses a combination of Huffman and runlength coding.)
As the quantization step size increases, the quantized block coefficients tend to have few non-zero values, and long strings of zeros can be efficiently encoded using a runlength code. For example, increasing the quantizer step size to 50 results in the normalized quantized block DCT coefficients shown below.

\[
\begin{array}{cccccccc}
19 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\
-8 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

Using a zig-zag scan pattern, the sequence of 64 quantization indices to be encoded is (19, 2, -8, 1, 0, 0, 1, -1, 2, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, …, 0). Typically, the dc coefficient is encoded separately, and a special symbol, “EOF”, is used immediately after the last non-zero coefficient, to denote that only zeros remain in the sequence.

It is insightful to visually examine the effect of quantizing using different step sizes, as well as to examine the signal-to-noise ratio resulting from different step sizes. The peak signal-to-noise ratio for a quantized image is computed as follows. Let \(x(n,m)\) denote the original image pixel values, and \(y(n,m)\) the corresponding quantized image pixel values. Then for an image of size \(N\) by \(M\), the mean-squared error is defined as

\[
MSE = \frac{1}{N \times M} \sum_{n=1}^{N} \sum_{m=1}^{M} (x(n,m) - y(n,m))^2
\]

The peak signal-to-noise ratio (PSNR) is then defined as

\[
PSNR = 10 \log_{10} \frac{255^2}{MSE}
\]

For small distortion (small enough step size) we expect a \(\text{mse}\) of approximately \(\frac{\Delta^2}{12}\) for a uniform scalar quantizer of step size \(\Delta\). So, if the quantizer step size is increased by a factor of two, we expect a change in \(\text{mse}\) of approximately 4, and hence a change in \(\text{SNR}\) of about \(10 \log_{10} 4 = 6.02\) dB.
The figures below show the 256x256 portion of the mandrill image with no quantization (the original), and reconstructed images for quantization step sizes of 25, 50 and 100. Also listed are the corresponding PSNR values.

The PSNR does not change by 6 dB for each doubling of the quantizer step size because rather large step sizes are required for the quantization effects to be perceptually significant. For a quantization step size of 25, there is little noticeable difference between the original and the quantized image. As the step size is doubled, and then doubled again, the distortion becomes increasingly significant. One characteristic of the distortion in the lower right image is the “blockiness” of the coding artifacts. This is typical of JPEG (or MPEG) compression when the compression ratio is large and hence there is large distortion (and relatively small PSNR).