ITCT Lecture 10.2: Discrete Cosine Transform (DCT)

Cited from Internet!

Good for self-learner!

- DCT is a Fourier-related transform similar to DFT, but using only real numbers. It is equivalent to a DFT roughly twice the transform length, operating on real data with "Even symmetry" (since Fourier transform of a real and even function is real and even). The most common variant of DCT is the type-II DCT and its inverse is the type-III DCT.

- Two related transform are the discrete sine transform (DST), which is equivalent to a DFT of “Real and odd” functions, and the modified DCT (MDCT), which is based on a DCT of overlapping data.
- The other interesting transform is the discrete Hartley transform (DHT) in which

    Even part of DHT = Real part of DFT

    Odd part of DHT = Imaginary part of DFT

- DCT is often used in signal and image processing, especially for lossy data compression, because it has a strong “Energy Compaction” property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT, approaching the KLT for signals based on certain limits of Markov processes.
- DCT is used in JPEG image compression, MJPEG , MPEG, H.264, and HEVC video compression.
- MDCT is used in MP3, AAC, etc. audio compression.

- Formulation:
  - DCT is a linear and invertible function
  - F: $R^n \rightarrow R^n$ (where R denotes the set of real numbers.), or equivalently on nxn matrix.
- DCT-I:

$$f_j = \frac{1}{2}(x_0 + (-1)^j x_{n-1}) + \sum_{k=1}^{n-2} x_k \cos[\frac{\pi}{n-1} jk]$$

- A DCT-I of n=5 real numbers abcde is exactly equivalent to a DFT of 8 real numbers abcdedcb (even symmetry), here divided by 2. (In contrast, DCT-II ~ IV involve a half-sample shift in the equivalent DFT.)
- Note that DCT-I is not defined for n less than 2. (All other DCT types are defined for any positive n.)

- DCT-II:

$$f_j = \sum_{k=0}^{n-1} x_k \cos[\frac{\pi}{n} j(k + \frac{1}{2})]$$

- Some authors further multiply the f_o term by $1/\sqrt{2}$ (see below for the corresponding change in DCT-III) . This makes the DCT-II matrix orthogonal (up to a scale factor), but breaks the direct correspondence with a real-even DFT of half-shifted input.

- DCT-III:

$$f_j = \frac{1}{2} x_0 + \sum_{k=1}^{n-1} x_k \cos[\frac{\pi}{n}(j + \frac{1}{2})k]$$

- Some authors further multiply the x_0 term by $1/\sqrt{2}$, this makes the DCT-III matrix orthogonal (up to a scale factor), but breaks the direct correspondence with a real even DFT of half-shifted output.

- DCT-IV:

$$f_j = \sum_{k=0}^{n-1} x_k \cos[\frac{\pi}{n}(j+\frac{1}{2})(k+\frac{1}{2})]$$

- DCT-IV matrix is orthogonal (up to a scale factor).
- MDCT is based on DCT-IV with overlapped data.

- DCT V-VIII:
  - DCT types I-IV are equivalent to real-even DFTs of even order; therefore, there are 4 additional types of DCT corresponding to real-even DFTs of logically odd order, which have factors of (n + 1/2) in the denominators of the cosine arguments. These variants seem to be rarely used in practice.

- Inverse Transforms:
  - IDCT-I is DCT-I multiplied by 2/(n-1).
  - IDCT-IV is DCT-IV multiplied by 2/n.
  - IDCT-II is DCT-III multiplied by 2/n (and versa).

- Computation
  - Direct application of the above formulas would require $O(n^2)$ operations, as in the FFT it is possible to compute the same thing with only O(nlogn) complexity by factorizing the computation. (One can also compute DCTs via FFTs combined with O(n)) pre- and post-processing steps.)

- References:

  - 1. Rao and Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications; Academic Press, Boston, 1990.
  - 2. Arai, Agui, Nakajima, A Fast DCT-SQ scheme for Images, Trans. On IEICE-E, 71(11), 1095, Nov. 1998.
  - 3. Tseng and Millen, On Computing the DCT, IEEE Trans. On Computers, pp. 966-968, Oct. 1978.
  - 4. Frigo and Johnson, The Design and Implementation of FFTW3, IEEE Proceedings, vol. 93, no. 2, pp. 216-231, 2005.

- The implementation of the 2D-IDCT
- Let the 8-point 1-D DCT of input data f(x) be:

$$S_8(u) = \frac{C_u}{2} \sum_{x=0}^{7} f(x) \cos \frac{(2x+1)\pi u}{16} \quad (1)$$

- First the 1-D DCT is applied to all the rows of the 2-D input f(y,x):

$$S_{8r}(y,u) = \frac{C_u}{2} \sum_{x=0}^{7} f(y,x) \cos \frac{(2x+1)\pi u}{16} \quad (2)$$

- Then the 1-D DCT is applied to the columns of the results of (2):

$$S(v,u) = \frac{C_u}{2} \sum_{y=0}^{7} S_{8r}(y,u) \cos \frac{(2y+1)\pi v}{16} \qquad (3)$$

- By substitution we get the formulation of the 2-D DCT:

$$S(v,u) = \frac{C_v}{2} \frac{C_u}{2} \sum_{y=0}^{7} \sum_{x=0}^{7} f(y,x) \cos \frac{(2x+1)\pi u}{16} \cos \frac{(2y+1)\pi v}{16} \qquad (4)$$

- Fast 1-D DCT Algorithms
  - Define $\alpha = \frac{2\pi u x}{16}, \beta = \frac{\pi u}{16}, \text{and } H = \alpha + \beta.$
  - Eqn. (1) can be written as:

$$S_8(u) = \frac{C_u}{2} \sum_{x=0}^{7} f(x) \cos \frac{(2x+1)\pi u}{16} = \frac{C_u}{2} \sum_{x=0}^{7} f(x) \cos(\alpha + \beta)$$

  - Since

$$2\cos H \cos\beta = 2\cos(\alpha+\beta)\cos\beta \underset{\substack{\uparrow \\ (HW)}}{=} \cos\frac{2x\pi u}{16} + \cos\frac{2(15-x)\pi u}{16}$$

$$\frac{4}{C_u}\cos\frac{\pi u}{16}S_8(u)=\sum_{x=0}^{7}f(x)[\cos\frac{2x\pi u}{16}+\cos\frac{2(15-x)\pi u}{16}] \quad (5)$$

- If we constitute a sequence of elements f(k), k = 0, 1, …, 15 with

$$f(k)=\begin{cases} f(k), & \forall k<8 \\ f(15-k), & \forall k\geq 8 \end{cases}$$

- We can re-write (5) as

$$\frac{4}{C_u}\cos\frac{\pi u}{16}S_8(u)=\sum_{k=0}^{15}f(k)\cos\frac{2k\pi u}{16}=\mathrm{Re}\{\sum_{k=0}^{15}f(k)e^{-j\frac{2k\pi u}{16}}\}$$

- When $j=\sqrt{-1}$
- Because the 16-point DFT is defined by

$$F_{16}(u)=\sum_{k=0}^{15}f(k)e^{-j\frac{2k\pi u}{16}}$$

- We have the following DCT v.s. DFT relationship

$$\frac{4}{C_u}\cos\frac{\pi u}{16}S_8(u)=\mathrm{Re}\{F_{16}(u)\}$$

: only the first 8 values are needed.

- Instead of performing an IDCT an IDFT of twice the length is performed. IDFT can be implemented by IFFT with complexity O(NlogN)!

- The multiplication by $\frac{4}{C_u}\cos\frac{\pi u}{16}$ seems not so efficient. However, bear in mind that the last operation before the IDCT is the "Quantization". That means every value is to be multiplied with a certain constant (1/Quantization factor) depending on its position in DCT matrix. So we can merge the multiplications by $\frac{4}{C_u}\cos\frac{\pi u}{16}$ and the multiplication by Quantizer dependent constant together. As a result, the Quantization and DCT-DFT transform can be performed in one step.

- Let
  - X_M = (f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)): original data
  - F_M = ($\frac{1}{16}$ F(0), $\frac{1}{8}$ F(1), $\frac{1}{8}$ F(2), $\frac{1}{8}$ F(3), $\frac{1}{8}$ F(4), $\frac{1}{8}$ F(5), $\frac{1}{8}$ F(6), $\frac{1}{8}$ F(7)): scaled transformed data
- Define P(a,b) = $\cos\frac{2\pi a}{16} + \cos\frac{2\pi b}{16}$
- We can establish a matrix T_M:

$$
T_M = \begin{bmatrix}
\frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} & \frac{P(0,0)}{2} \\
P(0,15) & P(1,14) & P(2,13) & P(3,12) & P(4,11) & P(5,10) & P(6,9) & P(7,8) \\
P(0,30) & P(2,28) & P(4,26) & P(6,24) & P(8,22) & P(10,20) & P(12,18) & P(14,16) \\
P(045) & P(3,42) & P(6,39) & P(9,36) & P(12,33) & P(15,30) & P(18,27) & P(21,24) \\
P(0,60) & P(4,56) & P(8,52) & P(12,48) & P(16,44) & P(20,40) & P(24,36) & P(28,32) \\
P(0,75) & P(5,70) & P(10,65) & P(15,60) & P(20,55) & P(25,50) & P(30,45) & P(35,40) \\
P(0,90) & P(6,84) & P(12,78) & P(18,72) & P(24,66) & P(30,60) & P(36,54) & P(42,48) \\
P(0,105) & P(7,98) & P(14,91) & P(21,84) & P(28,77) & P(35,70) & P(42,63) & P(49,56)
\end{bmatrix}
$$

- $F_M = X_M \times (1/8)T_M$
- Because

$$\cos a = \cos(-a); \quad -\cos a = \cos(p - a) = \cos(p + a)$$

$$\cos(2pn + a) = \cos a;$$

- Define

$$k_2 = \cos\frac{\pi}{8}; k_4 = \cos\frac{2\pi}{8} = \cos\frac{\pi}{4} = \frac{\sqrt{2}}{2}; k_6 = \cos\frac{3\pi}{8} = \sin\frac{\pi}{8}$$

$$T_M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1+k_2 & k_2+k_4 & k_4+k_6 & k_6 & -k_6 & -k_6-k_4 & -k_4-k_2 & -k_2-1 \\ 1+k_4 & k_4 & -k_4 & -k_4-1 & -1-k_4 & -k_4 & k_4 & k_4+1 \\ 1+k_6 & k_6-k_4 & -k_4-k_2 & -k_2 & k_2 & k_2+k_4 & k_4-k_6 & -k_6-1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1-k_6 & -k_6-k_4 & -k_4+k_2 & k_2 & k_2 & -k_2+k_4 & k_4+k_6 & k_6-1 \\ 1-k_4 & -k_4 & k_4 & k_4-1 & -1+k_4 & k_4 & -k_4 & -k_4+1 \\ 1-k_2 & -k_2+k_4 & k_4-k_6 & -k_6 & k_6 & k_6-k_4 & -k_4+k_2 & k_2-1 \end{bmatrix}$$

- $X_M = F_M \times (8T_M^{-1}) = F_M \times L; (L = 8T_M^{-1})$

$$L = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & C_2 - 1 & 1 - C_2 + C_4 & C_6 - C_4 + C_2 - 1 & 1 - C_6 + C_4 - C_2 & -1 - C_4 + C_2 & 1 - C_2 & -1 \\ 1 & -1 + C_4 & -C_4 + 1 & -1 & -1 & -C_4 + 1 & -1 + C_4 & 1 \\ 1 & C_6 - 1 & 1 - C_6 - C_4 & -C_2 + C_4 + C_6 - 1 & 1 + C_2 - C_4 - C_6 & -1 + C_4 + C_6 & 1 - C_6 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -C_6 - 1 & 1 + C_6 - C_4 & C_2 + C_4 - C_6 - 1 & 1 - C_2 - C_4 + C_6 & -1 + C_4 - C_6 & 1 + C_6 & -1 \\ 1 & -1 - C_4 & C_4 + 1 & -1 & -1 & C_4 + 1 & -1 - C_4 & 1 \\ 1 & -C_2 - 1 & 1 + C_2 + C_4 & -C_6 - C_4 - C_2 - 1 & 1 + C_6 + C_4 + C_2 & -1 - C_4 - C_2 & 1 + C_2 & -1 \end{bmatrix}$$

- Where $C_2 = 2\cos\frac{\pi}{8}; C_4 = 2\cos\frac{2\pi}{8} = \sqrt{2}$

$$C_6 = 2\cos\frac{3\pi}{8} = 2\sin\frac{\pi}{8}$$

- The matrix L can be factored as:

  L = B1 x M x A1 x A2 x A3

with:

$$B1=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}$$

$$M=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -C_2 & 0 & -C_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -C_6 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \qquad A2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- So the IDFT can be performed by 5 steps:

$$A3=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$
B1:\begin{cases}
a_0 = \dfrac{1}{16}F(0) \\
a_1 = \dfrac{1}{8}F(4) \\
a_2 = \dfrac{1}{8}F(2) - \dfrac{1}{8}F(6) \\
a_3 = \dfrac{1}{8}F(2) + \dfrac{1}{8}F(6) \\
a_4 = \dfrac{1}{8}F(5) - \dfrac{1}{8}F(3) \\
temp1 = \dfrac{1}{8}F(1) + \dfrac{1}{8}F(7) \\
temp2 = \dfrac{1}{8}F(3) + \dfrac{1}{8}F(5) \\
a_5 = temp1 - temp2 \\
a_6 = \dfrac{1}{8}F(1) - \dfrac{1}{8}F(7) \\
a_7 = temp1 + temp2
\end{cases}
$$

$$
M:\begin{cases}
b_0 = a_0 \\
b_1 = a_1 \\
b_2 = a_2 c_4 \\
b_3 = a_3 \\
b_4 = -(a_4 C_2 + a_6 C_6) \\
b_5 = a_5 C_4 \\
b_6 = (-a_4 C_6 + a_6 C_2) \\
b_7 = a_7
\end{cases}
$$

$$A1:\begin{cases} temp3 = b_6 - b_7 \\ n_0 = temp3 - b_5 \\ n_1 = b_0 - b_1 \\ n_2 = b_2 - b_3 \\ n_3 = b_0 + b_1 \\ n_4 = temp3 \\ n_5 = b_4 \\ n_6 = b_3 \\ n_7 = b_7 \end{cases}$$

$$A2:\begin{cases} m_0 = n_7 \\ m_1 = n_0 \\ m_2 = n_4 \\ m_3 = n_1 + n_2 \\ m_4 = n_3 + n_6 \\ m_5 = n_1 - n_2 \\ m_6 = n_3 - n_6 \\ m_7 = n_5 - n_0 \end{cases}$$

$$A3:\begin{cases} f(0) = m_4 + m_0 \\ f(1) = m_3 + m_2 \\ f(2) = m_5 - m_1 \\ f(3) = m_6 - m_7 \\ f(4) = m_6 + m_7 \\ f(5) = m_5 + m_1 \\ f(6) = m_3 - m_2 \\ f(7) = m_4 - m \text{[illegible]} \end{cases}$$

- In part M, a simplification can be made:

$$b_4 = -(a_4C_2 + a_6C_6) = -a_4C_2 - a_6C_6 - a_4C_6 + a_4C_6$$
$$= -C_6(a_4 + a_6) - a_4(C_2 - C_6)$$
$$b_2 = -a_4C_6 + a_6C_2 = -a_4C_6 + a_6C_2 - a_6C_6 + a_6C_6$$
$$= -C_6(a_4 + a_6) + a_6(C_2 + C_6)$$

- So define $Q = C_2 - C_6$ and $R = C_2 + C_6$

$$temp4 = C_6(a_4 + a_6)$$
$$b_4 = -Qa_4 - temp4$$
$$b_6 = Ra_6 - temp4$$

- If you count the operations you will get 29 additions and 5 multiplications.
- Because an 8x8 2-D DCT can be computed by applying 8-point 1-D DCT to each row and each column of the 2D data. Therefore, 2x8x29 = 464 additions and 2x8x5 = 80 multiplications are required in total.
- A more efficient DCT algorithm has been proposed by Feig and Winograd in IEEE Trans. on ASSP, pp. 2174-2193, 1992.