Discrete Cosine Transform

Lecture and Notes by Prof. Brian L. Evans (UT Austin)

Scribe: Clint Slatton (UT Austin)

Based on notes by Prof. Russell Mersereau (Georgia Tech)

Introduction

The DFT is not the only transform that is widely used in applications

Published standards for image and video coding (compression) make use of the DCT

1. JPEG (1989)
2. MPEG1, 2, and 4
   2.1. MPEG1 (1992): video CD players, storage and retrieval of moving pictures and audio on storage media
   2.2. MPEG2 (1994): HDTV, DVD, standard for Digital TV (cable)
   2.3. MPEG3, originally targeted for HDTV, was incorporated into MPEG2
   2.4. MPEG4 (late 1998): standard for multimedia applications, targeted for wireless video
3. H.261, H.263
   3.1. H.261 (circa 1993): video conferencing
   3.2. H.263 (circa 1995): wireless video

These standards provide instructions for decoding the signal, but there is often considerable freedom for encoding the signal. For more information go to http://drogo.cselt.it/mpeg/.

Compression

Two classes of compression algorithms try to reduce the number of bits required to represent a signal.

- *lossless*: compression ratios around 2-3:1 for data files
- *lossy*: compression ratios up to 1000:1 for video

For wireless video, need compression ratios up to 1000:1. Can get *near lossless* video compression at 8:1 with little degradation.

Compression algorithms work by removing redundancy in the signal. In video signals, the redundancy can be of three forms.

- statistical: (e.g. Huffman codes, arithmetic, Lempel-Ziv)
- spatial: (e.g. vector quantization, DCT, subband coders, wavelets)
- temporal: (e.g. motion compensation)

Wavelet compression is used in JPEG-2000, MPEG4, and H.263+

Transform coders decompose a frame into blocks, typically 8 x 8. In MPEG2, they are called macroblocks and divide the frame into luminance (intensity) and chrominance (color) images (YUV).

- luminance image: one 16 x 16 macroblock or four 8 x 8 macroblocks (Y)
- chrominance image: two 8 x 8 blocks (UV)

A 2-D DCT of each block is computed and the transform coefficients are quantized. Quantized coefficients are coded losslessly. The choice of quantization affects the transmission rate and distortion.

Advantages of the DCT (relative to the DFT)

- real-valued
- better energy compaction (much of the signal energy can be represented by only a few coefficients)
- coefficients are nearly uncorrelated
- experimentally observed to work well

2-D DCT

$$X_c[k_1, k_2] = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} 4x[n_1, n_2] \cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right) \cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right)$$

$$x[n_1, n_2] = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} C[k_1] C[k_2] X_c[k_1, k_2] \cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right) \cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right)$$

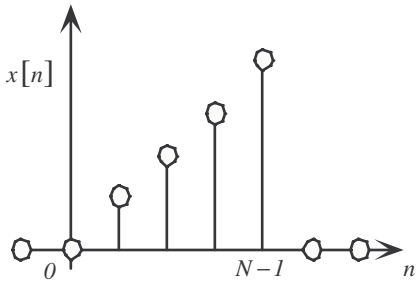
$$C[k] = \begin{cases} 1/2, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

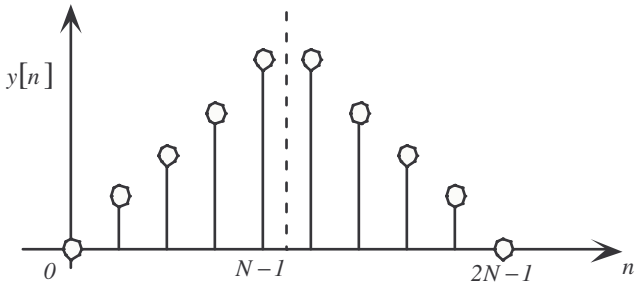
1. The DFT is related to the Fourier Series coefficients of a *periodically* extended sequence.
2. The DCT is related to the Fourier Series coefficients of a *symmetrically* extended sequence.
3. The 2-D DCT is a separable transform. It can be evaluated using a row-column decomposition. For a 8 x8 DCT, we need 16 1-D DCTs. Calling 16 functions may lead to unacceptable overhead for 8-point DCTs.

1-D DCT

$$X_c[k] = \sum_{n=0}^{N-1} 2x[n]\cos\left(\frac{\pi(2n+1)k}{2N}\right)$$

Define the symmetric extension of $x[n]$ as $y[n] = x[n] + x[2N-1-n]$ for $n = 0, 1, \ldots, N-1$.





Now consider the 2*N*-point DFT of $y[n]$.

$$\begin{aligned}
Y[k] &= \sum_{n=0}^{2N-1} y[n] e^{-j\frac{2\pi}{2N}kn} \\
&= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{2N}kn} + \sum_{n=N}^{2N-1} x[2N-1-n] e^{-j\frac{2\pi}{2N}kn} \\
&= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{2N}kn} + \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{2N}k(2N-1-n)} \\
&= \sum_{n=0}^{N-1} x[n] \left\{ e^{-j\frac{2\pi}{2N}kn} + e^{j\frac{2\pi}{2N}kn} e^{j\frac{2\pi}{2N}k} \right\} \\
&= e^{j\frac{\pi}{2N}k} \cdot \sum_{n=0}^{N-1} x[n] \left\{ e^{-j\frac{\pi}{2N}k} e^{-j\frac{2\pi}{2N}kn} + e^{j\frac{2\pi}{2N}kn} e^{j\frac{\pi}{2N}k} \right\} \\
&= e^{j\frac{\pi}{2N}k} \cdot \sum_{n=0}^{N-1} 2x[n] \cos\left(\frac{\pi k}{2N}(2n+1)\right)
\end{aligned}$$

Algorithm #1 for 1-D DCT

1. Set $y[n] = x[n] + x[2N - 1 - n]$
2. Calculate $Y[k]$ using a 2*N*-point DFT
3. Set $X_c[k] = exp\left(-j\frac{\pi}{2N}k\right)Y[k]$, for $k = 0, 1, \ldots, N - 1$.

This requires $N+N\log_2(2N)$ complex multiplies. Another algorithm can be developed that requires fewer multiplies by using a shorter DFT.

Review of 1-D Decimation-in-Time (DIT) FFT

Consider the 1-D DFT

$$Y[k] = \sum_{n=0}^{2N-1} y[n]W_{2N}^{nk}$$

Divide the sum into two components, one over the even samples and one over the odd samples.

$$\begin{aligned} Y[k] &= \sum_{r=0}^{N-1} y[2r]W_{2N}^{2rk} + \sum_{r=0}^{N-1} y[2r+1]W_{2N}^{(2r+1)k}, \quad k = 0,1,...,2N-1 \\ &= \sum_{r=0}^{N-1} y[2r]W_N^{rk} + W_{2N}^k \sum_{r=0}^{N-1} y[2r+1]W_N^{rk}, \quad k = 0,1,...,2N-1 \\ &= G[k] + W_{2N}^k H[k] \end{aligned}$$

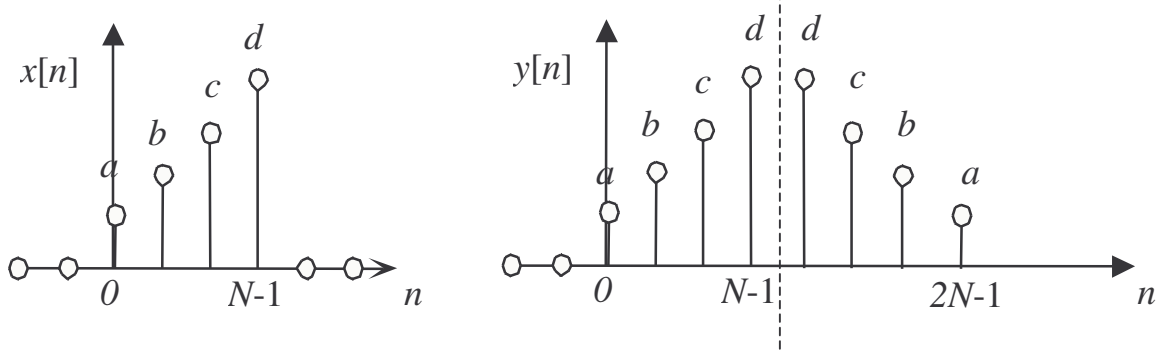
Note that $W_{2N}^{2rk} = \exp(-2\pi\frac{2rk}{2N}) = \exp(-2\pi\frac{rk}{N}) = W_N^{rk}$

G[*k*] and *H*[*k*] are N-point DFTs:

Set

$$g[n] = y[2n], \quad n = 0, 1, \ldots, N - 1$$

$$h[n] = y[2n + 1], \quad n = 0, 1, \ldots, N - 1$$



$$g[n] = \{a, c, d, b\}$$

$$h[n] = \{b, d, c, a\} = g[N-1-n]$$
$$\Rightarrow H[k] = W_N^{-k} G[((N-k))_N] = W_N^{-k} G^*[k]$$

Therefore,

$$Y[k] = G[k] + e^{j\frac{\pi k}{2N}} e^{-j\frac{\pi k}{N}} G^*[k], \quad k = 0,1,\ldots,2N-1$$

$$Y[k] = G[k] + e^{-j\frac{\pi k}{2N}} G^*[k], \quad k = 0,1,\ldots,2N-1$$

$$X_c[k] = \mathrm{Re}\left(e^{-j\frac{\pi k}{2N}} \left(G[k] + e^{-j\frac{\pi k}{2N}} G^*[k] \right) \right), \quad k = 0,1,\ldots,2N-1$$

$$= \mathrm{Re}\left(e^{-j\frac{\pi k}{2N}} G[k] + e^{-j\frac{2\pi k}{2N}} G^*[k] \right)$$

$$= \mathrm{Re}\left(e^{-j\frac{3\pi k/2}{2N}} \left(e^{j\frac{\pi k/2}{2N}} G[k] + e^{-j\frac{\pi k/2}{2N}} G^*[k] \right) \right)$$

$$\boxed{X_c[k] = 2\cos\left(\frac{3\pi k}{4N}\right)\cos\left(\frac{\pi k}{4N}\right)\mathrm{Re}(G[k]), \quad k = 0,1,\ldots,N-1}$$

Algorithm #2 for 1-D DCT

1. Set

   $$g[n] = x[2n], \quad n = 0,1,\ldots,\frac{N}{2}-1$$

   $$g[N-1-n] = x[2n+1], \quad n = 0,1,\ldots,\frac{N}{2}-1$$

2. Calculate the *N*-point DFT of $g[n]$

3. Set

   $$X_c[k] = 2\cos\left(\frac{3\pi k}{4N}\right)\cos\left(\frac{\pi k}{4N}\right)\mathrm{Re}(G[k])$$

The coefficients in front of $G_{real}[k]$ can be pre-computed for different values of *k* for a given *N*.

Note: Special algorithms exist for computing DFTs of real arrays.