ECE6258 Lecture 21

The Discrete Cosine Transform (DCT)

Discrete cosine transform (DCT)

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

$$x[n_1,n_2] = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} C[k_1]C[k_2]X_c[k_1,k_2]\cos\left(\frac{\pi(2n_1+1)k_1}{2N_1}\right)\cos\left(\frac{\pi(2n_2+1)k_2}{2N_2}\right)$$

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

- Separable transform
- This is one of a family of DCTs

DCT and Image/Video Compression

- The DCT is part of most of the image and video compression standards
  - JPEG, MPEG1, MPEG2, MPEG4, H.261, H.263, H.264
- DCT coefficients are real.
- DCT has excellent energy compaction properties
  - It is the KLT for a 1st-order Markov process with high correlation coefficient.

The 1-D DCT

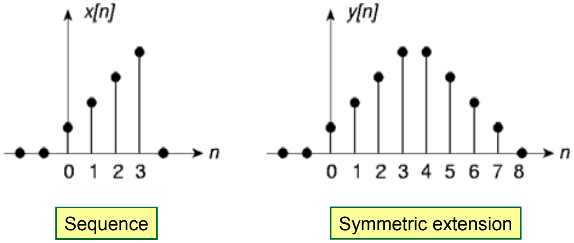
- The 1-D DCT of a sequence *x*[*n*] is defined by

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

- It is related to the DFT of the **periodically extended** sequence *y*[*n*].

$$y[n] = x[n] + x[2N-1-n]$$

Symmetric Extension



The DCT and the DFT

- Consider the 2*N*-point DFT of *y*[*n*]…

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

Algorithm #1 for the DCT

1. Set *y*[*n*] = *x*[*n*] + *x*[2*N*-1-*n*].
2. Calculate *Y*[*k*] = DFT{*y*[*n*]} using a 2*N*-point DFT.
3. Set $X_c[k]$ = exp(-*jπk*/(2*N*)) *Y*[*k*] for 0 ≤ *k* ≤ *N*-1.

- To compute the inverse DCT reverse the steps.
  - Steps 1 and 2 are trivial; only issue concerns step 3.

Inverting Step #3

- **Issue:** $X_c[k]$ known for 0 ≤ *k* ≤ *N*-1, but *Y*[*k*] needed for 0 ≤ *k* ≤ 2*N*-1.
- **But:** *Y*[*k*] is the DFT of a real sequence.
- **Therefore:**

  Y[*k*] = *Y**[(2*N*-*k*) mod 2*N*]

Inverting Step #3 (continued)

- Using the formula for $Y[k]$ and substituting into the preceding relation gives

$Y[k] = -Y[2N-k]$ for $k = 1,2,\ldots,N-1$

- Furthermore, because $y[n]$ is symmetric

$$Y[N] = \sum_{n=0}^{2N-1} (-1)^n y[n] = 0$$

Algorithm #1 for the Inverse DCT

1. Set $Y[k] = \exp(j\pi k/(2N))X_c[k]$ for $0 \le k \le N-1$
2. Set $Y[2N-k] = -Y[k]$ for $0 \le k \le N-1$
3. Set $Y[N]=0$
4. Calculate $y[n] = \text{IDFT}\{Y[k]\}$
5. Set $x[n] = y[n]$, for $0 \le n \le N-1$.

An Alternative

- Algorithm #1 requires computing a $2N$-point real DFT.
- Algorithm #2 effectively requires only an N-point real DFT.
- Consider the 1-D DFT sum…

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

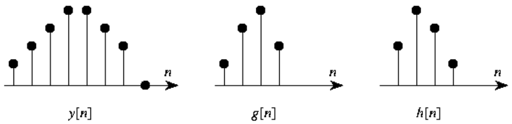
Derivation of alternative (continued)

- Divide this sum into two components—one over the even samples and one over the odd samples.

$$\begin{aligned} Y[k] &= \sum_{r=0}^{N-1} y[2r]W_{2N}^{2rk} + \sum_{r=0}^{N-1} y[2r+1]W_{2N}^{(2r+1)k} \\ &= \sum_{r=0}^{N-1} \underbrace{y[2r]}_{g[n]} W_N^{rk} + W_{2N}^{k} \sum_{r=0}^{N-1} \underbrace{y[2r+1]}_{h[n]} W_N^{rk} \end{aligned}$$

- Since each of these sums is only required for $k=0,1,\ldots,N-1$, each can be evaluated using an N-point real FFT algorithm.

Derivation of alternative (continued)



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

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

- Notice that

$$\boxed{h[n] = g[N-1-n],\ n=0,1,\ldots,N-1}$$

- In the DFT domain

$$\boxed{H[k] = W_N^{-k}\, G[(N-k) \bmod N\,] = W_N^{-k}\, G^*[k]}$$

Derivation of alternative (continued)

- Therefore,

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

- and the DCT becomes…

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

Algorithm #2 for computing the DCT

1. Set
   - $g[n] = x[2n]$, for $0 \le n \le N/2-1$
   - $g[N-1-n] = x[2n+1]$, for $0 \le n \le N/2-1$
2. Calculate the N-point DFT of $g[n]$.
3. Set $X_c[k] = 2\,\mathrm{Re}\{\exp(-j\pi k/(2N))G[k]\}$.

Algorithm #2 for the inverse DCT

1. Set
   - $G[0] = 0.5\, X_c[0]$
   - $G[k] = 0.5\{ \exp(-j\pi k/(2N))\, X_c[k] + \exp(-j\pi(N-k)/(2N))\, X_c[N-k]$
2. Calculate the N-point inverse DFT of $G[k]$.
3. Set

$$x[n] = \begin{cases} g[n/2], & n \text{ even} \\ g[N-(n+1)/2], & n \text{ odd} \end{cases}$$

Properties of the DCT

- Linearity:

$$ax[n_1,n_2]+bv[n_1,n_2] \longleftrightarrow aX_c[k_1,k_2]+bV_c[k_1,k_2]$$

- Separability

$$f[n_1]g[n_2] \longleftrightarrow F[k_1]G[k_2]$$

- Reflections

$$x[N_1-1-n_1,n_2] \longleftrightarrow (-1)^{k_1}X_c[k_1,k_2]$$
$$x[n_1,N_2-1-n_2] \longleftrightarrow (-1)^{k_2}X_c[k_1,k_2]$$