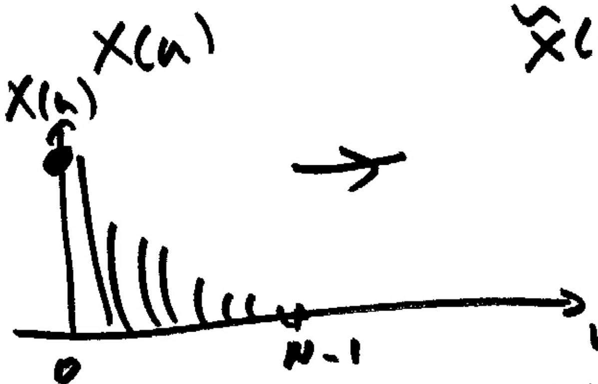
April 26, 06

Discrete Cosine Transform & its use in Image Compression

* Why DCT has better compaction properties than DFT?

DFT: $x(n)$ = finite length N pt seq.

$x(n)$ 

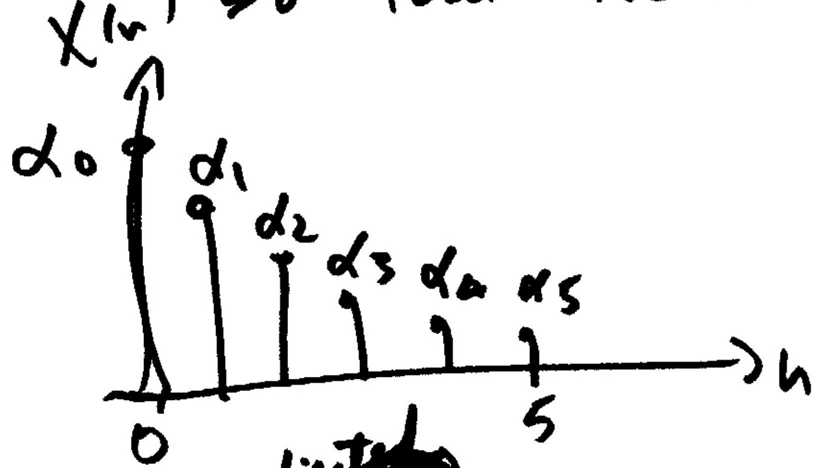
$\rightarrow$ $\tilde{x}(n) \rightarrow$ 

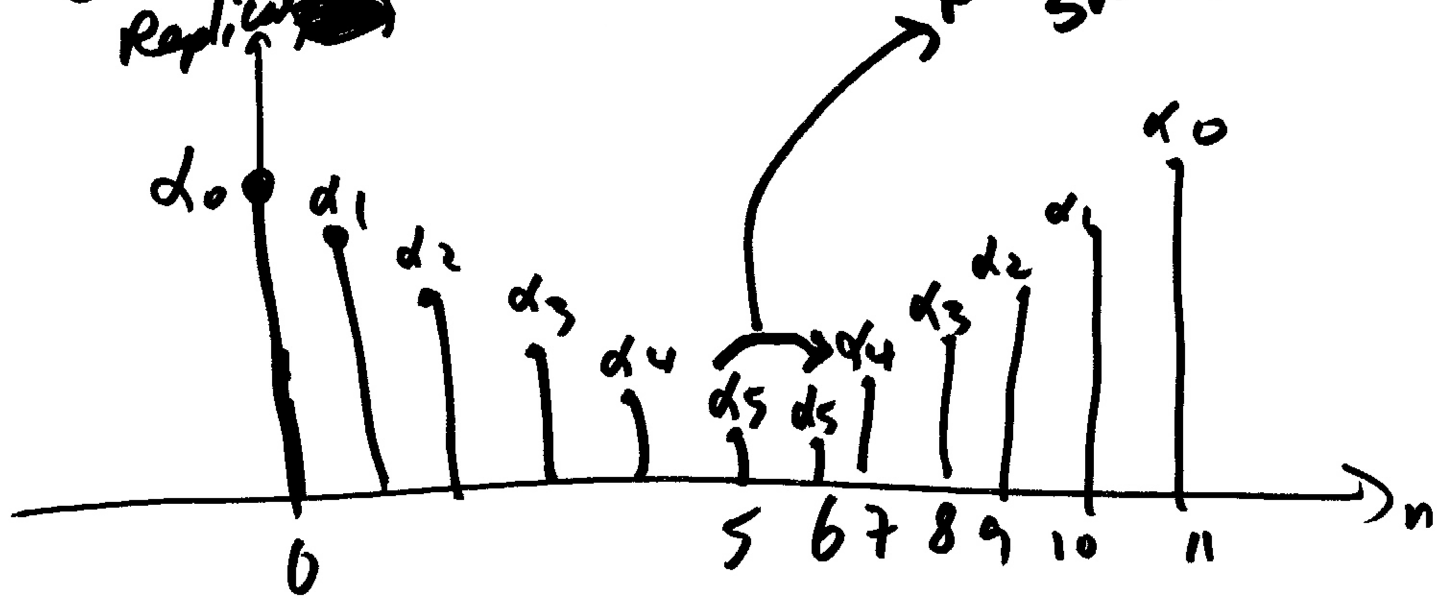
sharp discontinuities $\Rightarrow$ result in unnecessary high freq. coeffs in DFT.

$$\tilde{x}(n) \xrightarrow{DFS} \tilde{X}(k) \xrightarrow{\text{one period}} X(k) = DFT\{x(n)\}$$

How does DCT work

- Replicate signal in a "better" way.
  so that there is no sharp discontinuities





No longer any sharp discontinuities.

DCT Type II

$$X^{C_2}(k) = 2 \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

analysis ↗

synthesis ↘

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \beta(k)\, X_k^{C_2}(k) \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

$$\beta(k) = \begin{cases} \frac{1}{2} & k = 0 \\ 1 & 1 \le k < N \end{cases}$$

Relation Between DCT & DFT

Proposal 1

1. $x(n)$ N pT real seq.
2. 2N pT DFT $\longrightarrow X(k)$
3. $X^{c2}(k) = 2\,\mathrm{Re}\left\{X(k)\, e^{-j\frac{2\pi k}{N}}\right\}$

Can show

Proposal 2:

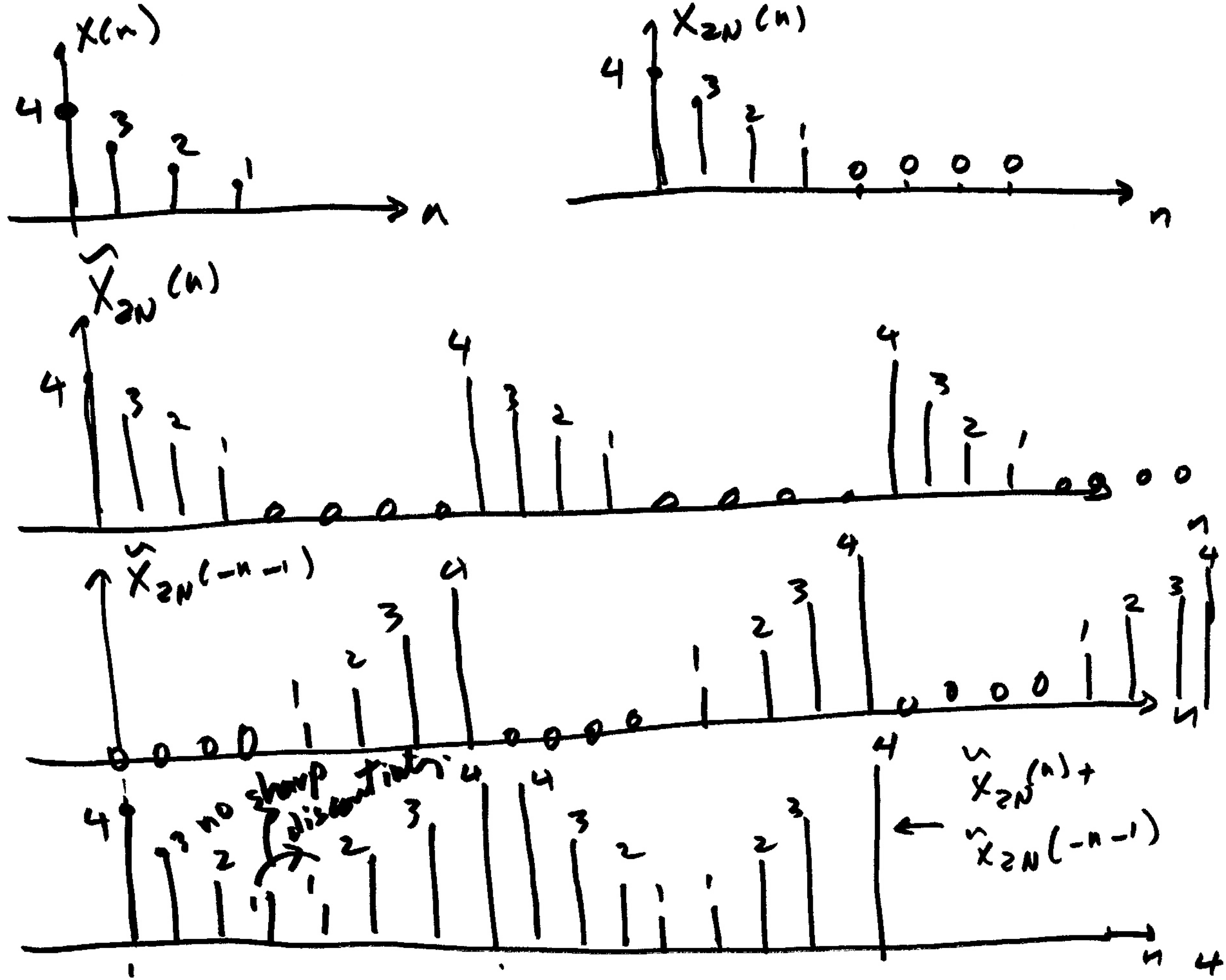
1. start N pT real seq. $x(n)$
2. Pad it with N zeros $x_{2N}(n)$
3. Form a periodic seq.

$$\tilde{x}_2(n) = \tilde{x}_{2N}(n) + \tilde{x}_{2N}(-n-1)$$

4. 2N pT DFT of one period of

$$\tilde{x}_2(n) \longrightarrow X_2(k) \overset{??}{=} X^{c2}(k)\, e^{\frac{j\pi k}{2N}}$$

5. $X_e^{c2}(k) = X_2(k)\, e^{-j\frac{\pi k}{2N}}$ (3.1)



x(n)
n
$x_{2N}(n)$
$\hat{x}_{2N}(n)$
$\hat{x}_{2N}(-n-1)$
no sharp discontinuity
$\hat{x}_{2N}(n) + \hat{x}_{2N}(-n-1)$

8 PT, 2N PT DFT of This seq.

Show proposal 2 works

How is $X_2(k)$ related to $\hat{X}_2(k)$?

$x(n) \rightarrow X(k) =$ 2N PT DFT of $x(n)$

Also 2N PT DFT of $x_{2N}(n)$

$$X_2(k) = X(k) + e^{j\frac{2\pi k}{2N}} X^*(k)$$

$$X_2(k) = e^{j\frac{\pi k}{2N}} \left\{ X(k) e^{-j\frac{\pi k}{2N}} + e^{j\frac{\pi k}{2N}} X^*(k) \right\}$$

5

$$X_2(k) = e^{j\frac{\pi k}{2N}} \underbrace{\left\{ 2\,\mathrm{Re}\left\{ X(k)\, e^{-j\frac{\pi k}{2N}} \right\} \right\}}_{X^{C_2}(k)}$$

$$X_2(k) = e^{j\frac{\pi k}{2N}} \, X^{C_2}(k)$$

$$\boxed{X^{C_2}(k) = X_2(k)\, e^{-j\frac{\pi k}{2N}}}$$

DFT of the original seq + reflected version.

2N pT

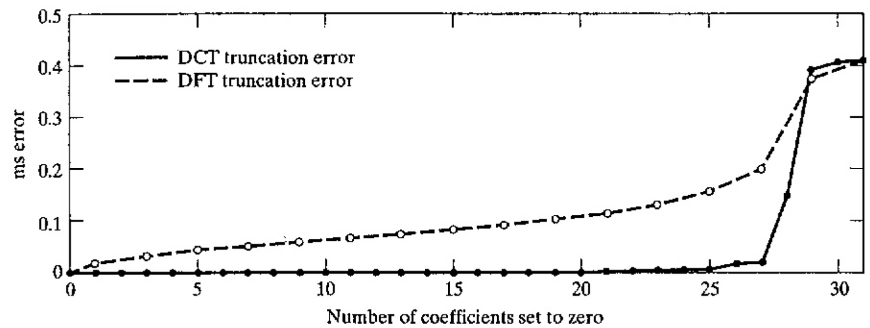


Figure 8.29 Comparison of truncation errors for DFT and DCT-2.

and

$$E^{\text{dct}}[m] = \frac{1}{N}\sum_{n=0}^{N-1} |x[n] - x_m^{\text{dct}}[n]|^2$$

to be the mean-squared approximation errors for the truncated DFT and DCT, respectively. These errors are plotted in Figure 8.29, with $E^{\text{dft}}[m]$ indicated with $\circ$ and $E^{\text{dct}}[m]$ shown with $\bullet$. For the special cases $m = 0$ (no truncation) and $m = N-1$ (only the DC value is retained), the DFT truncation function is $T_0[k] = 1$ for $0 \le k \le N-1$ and $T_{N-1}[k] = 0$ for $1 \le k \le N-1$ and $T_{N-1}[0] = 1$. In these cases, both representations give the same error. For values $1 \le m \le 30$, the DFT error grows steadily as m increases, while the DCT error remains very small up to about $m = 25$, implying that the 32 numbers of the sequence $x[n]$ can be represented with slight error by only seven DCT-2 coefficients.

Show picture in note

show 8.30 G/W.

The signal in Example 8.13 is a low frequency exponentially decaying signal with zero phase. We have chosen this example very carefully to emphasize the energy compaction property. Not every choice of $x[n]$ will give such dramatic results. Highpass signals and even some signals of the form of Equation (8.178) with different parameters do not show this dramatic difference. Nevertheless, in many cases of interest in data compression, the DCT-2 provides a distinct advantage over the DFT. It can be shown (Rao and Yip, 1990) that the DCT-2 is nearly optimum in the sense of minimum mean-squared truncation error for sequences with exponential correlation functions.

8.8.6 Applications of the DCT

The major application of the DCT-2 is in signal compression, where it is a key part of many standardized algorithms. (See Jayant and Noll, 1984 and Rao and Hwang, 1996.) In this application, the blocks of the signal are represented by their cosine transforms. The popularity of the DCT in signal compression is mainly due to its energy concentration property, which we demonstrated by a simple example in the previous section.

The DCT representations, being orthogonal transforms like the DFT, have many properties similar to those of the DFT that make them very flexible for manipulating

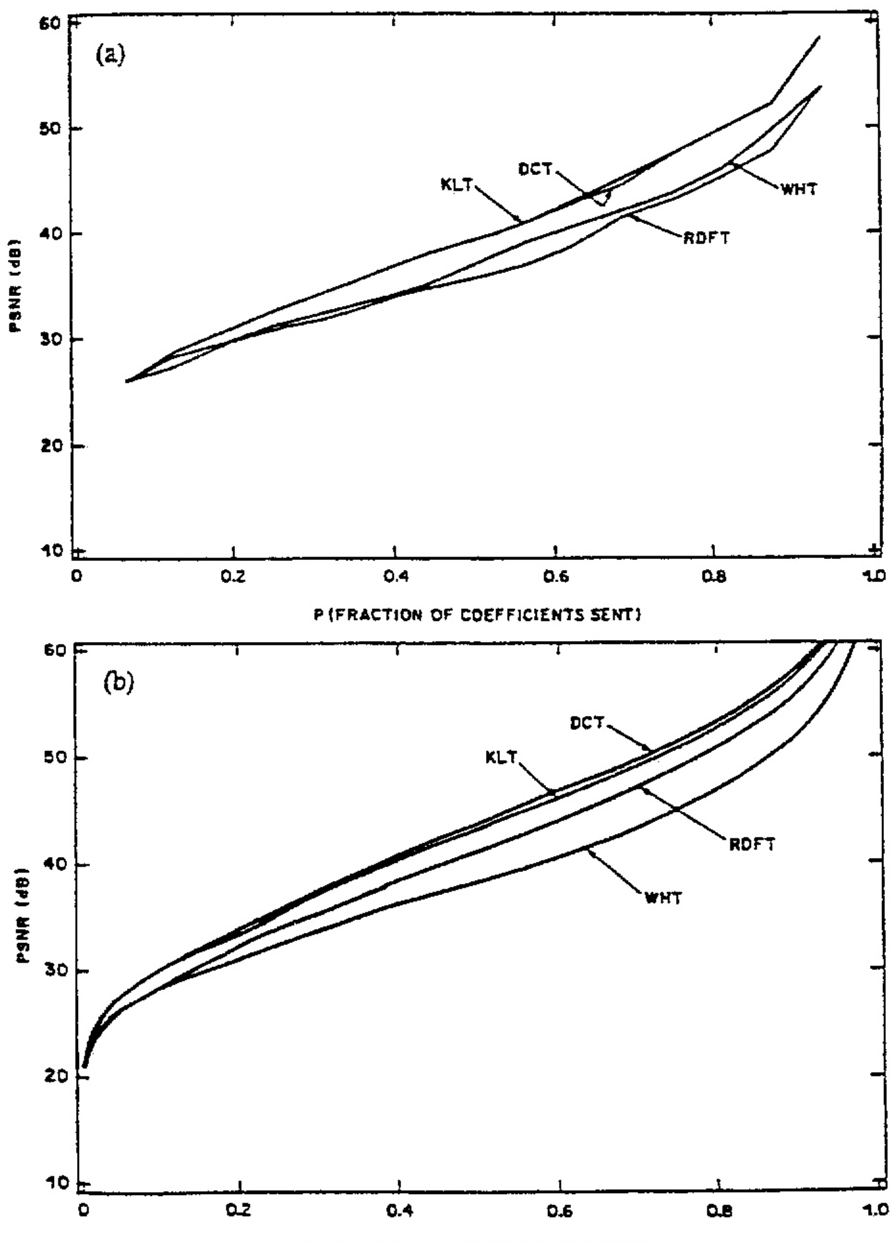


Fig. 5.3.21 Comparison of truncation errors using separable, two-dimensional blocks with the image "Karen". The coefficients having the largest MSV are transmitted. (a) 4×4 blocks, $N=16$. (b) 16×16 blocks, $N=256$.

8

9.

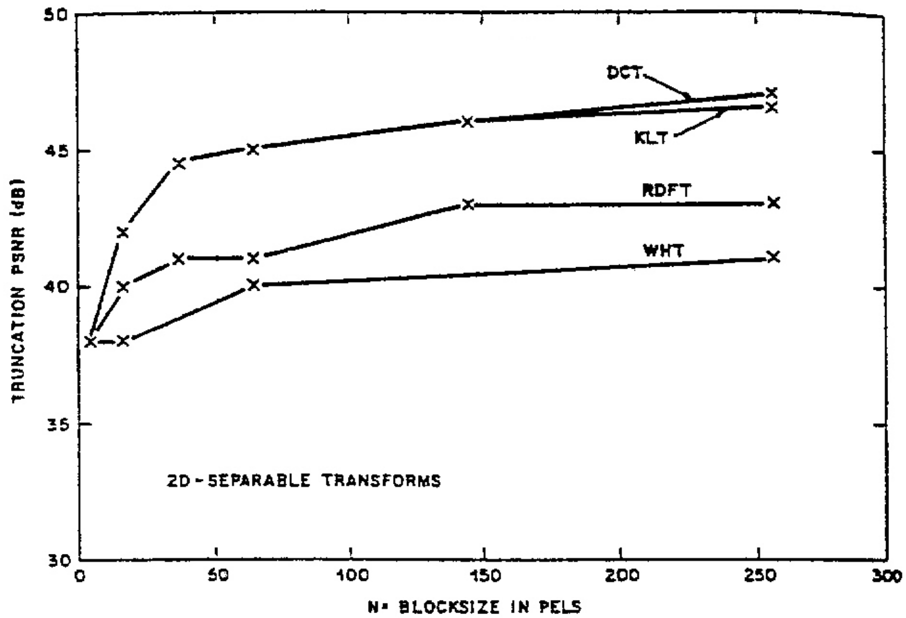


Fig. 5.3.22 Truncation PSNR versus block size for separable transforms with the image "Karen" when 60 percent of the coefficients are kept (p=0.6).

size, the higher the energy compaction achieved by the transform. Also two-dimensional blocks achieve more compaction than one-dimensional blocks. Experience has shown that over a wide range of pictures there is not much improvement in average energy compaction for two dimensional block sizes above 8×8 pels. However, individual pictures with higher nonstationary statistics can always be found for which this rule of thumb is violated (for example, compare the KLT curves of Fig. 5.3.17 and Fig.5.3.19). Also, considerable correlation may remain between blocks, even though the correlation between pels within a block is largely removed.[5.3.21] We shall return to this point in a later section.

5.3.1f Miscellaneous Transforms

Several other transforms have been studied. For example, the Haar Transform[5.3.8] can be computed from an orthogonal (but not orthonormal) matrix T that contains only +1's, −1's and zeros as shown in Fig. 5.3.23. This enables rather simple and speedy calculation, but at the expense of energy compaction performance.

Fig 8.30 — [illegible]

8.31

8.32

8.33

8.34

G/W

Walsh Hadamard Transform

$$T(u,v) = \sum_{x,y}^{N-1} \sum_{0} f(x,y)\, g(x,y,u,v)$$

Walsh Hadamard:

$$g(x,y,u,v) = \frac{1}{N} (-1)^{\sum_{i=0}^{m-1} [b_i(x) p_i(u) + b_i(y) p_i(v)]}$$

$N = 2^m$, summation in exponent is modulo 2.

$b_k(z)$ = kth bit from right to left in binary representation of z.

e.g $m=3$ $\quad z = 6 \longrightarrow (110)$

$b_0(z) = 0 \quad b_1(z) = 1 \quad b_2(z) = 1$

$P_i(u)$:

$$P_0(u) = b_{m-1}(u)$$

$$P_1(u) = b_{m-1}(u) + b_{m-2}(u)$$

$$P_2(u) = b_{m-2}(u) + b_{m-3}(u)$$

$$\vdots$$

$$P_{m-1}(u) = b_1(u) + b_0(u)$$

Scalar Quantization of a vector Source

8

8 

64 coeff.

X bits to divy up among the coef. How do I do "optimal" bit allocation.

- Assume N scalars f_i $1 \le i \le N$
- Each f_i is quantized to L_i reconstruction levels.
- Total of B bits to encode N scalar

- Optimum bit allocation strategy depends on
  (a) error criteria (b) pdf of each of random variables.

- Error criteria:

  minimize $MSE \triangleq \sum_{i=1}^{N} E\left[(f_i - f_i')^2 \right]$

  ~~subject~~ with respect B_i

  $B_i \triangleq$ bits allocated to ith scalar f_i

- Assume pdf of all f_i same except different variances.

- Assume Lloyd / max quantizer.

Lagrangian Optimization:

Can show:

① $$B_i = \frac{B}{N} + \frac{1}{2} \log \frac{\sigma_i^2}{\left(\prod_{j=1}^{N} \sigma_j^2 \right)^{1/N}}$$

where σ_i is variance of f_i

②

L_i = # of recon levels for f_i

$$L_i = \frac{\sigma_i}{\left[\prod_{j=1}^{N} \sigma_j\right]^{1/N}} \, 2^{B/N}$$