EE-597 Class Notes – Transform Coding

Phil Schniter

June 11, 2004

1 Transform Coding

1.1 Background and Motivation

- In transform coding (TC), blocks of N input samples are transformed to N transform coefficients which are then quantized and transmitted. At the decoder, an inverse transform is applied to the quantized coefficients, yielding a reconstruction of the original waveform. By designing individual quantizers in accordance with the statistics of their inputs, it is possible to allocate bits in a more optimal manner, e.g., encoding the "more important" coefficients at a higher bit rate.

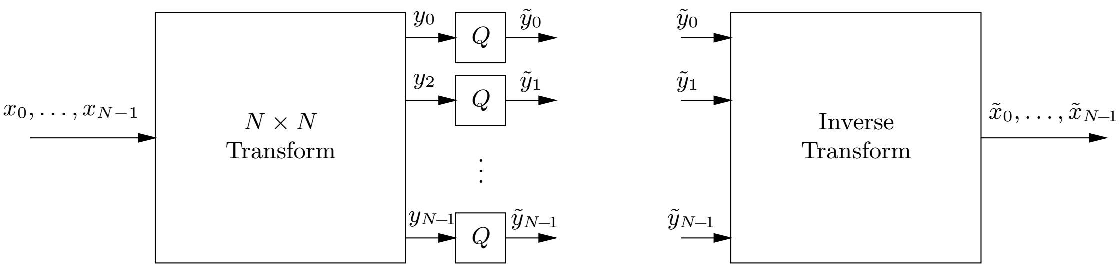

Figure 1: $N \times N$ Transform Coder/Decoder with Scalar Quantization

- *Orthogonal Transforms:* From our perspective, an $N \times N$ "transform" will be any real-valued linear operation taking N input samples to N output samples, or transform coefficients. This operation can always be written in matrix form

$$\mathbf{y}(m) = \mathbf{T}\mathbf{x}(m), \quad \mathbf{T} \in \mathbb{R}^{N \times N}$$

where $\mathbf{x}(m)$ and $\mathbf{y}(m)$ are vectors representing $N \times 1$ blocks of input/output elements:

$$\begin{aligned} \mathbf{x}(m) &= \big(x(mN),\ x(mN-1),\ \ldots,\ x(mN-N+1)\big)^t \\ \mathbf{y}(m) &= \big(y(mN),\ y(mN-1),\ \ldots,\ y(mN-N+1)\big)^t. \end{aligned}$$

Intuition comes from considering the transform's *basis vectors* $\{\mathbf{t}_k\}$ defined by the rows of the matrix

$$\mathbf{T} = \begin{pmatrix} - \mathbf{t}_0^t - \\ - \mathbf{t}_1^t - \\ \vdots \\ - \mathbf{t}_{N-1}^t - \end{pmatrix}$$

since the coefficient $y_k = \mathbf{t}_k^t \mathbf{x}$ can be thought of as the result of a "comparison" between the k^{th} basis vector and the input $\mathbf{x}$. These comparisons are defined by the inner product $<\mathbf{t}_k, \mathbf{x}>= \mathbf{t}_k^t \mathbf{x}$ which has a geometrical interpretation involving the angle θ_k between vectors $\mathbf{t}_k$ and $\mathbf{x}$.

$$<\mathbf{t}_k, \mathbf{x}> \;=\; \cos(\theta_k)\, \|\mathbf{t}_k\|_2\, \|\mathbf{x}\|_2.$$

When the vectors $\{\mathbf{t}_k\}$ are mutually orthogonal, i.e., $\mathbf{t}_k^t\mathbf{t}_\ell = 0$ for $k \neq \ell$, the transform coefficients represent separate, unrelated features of the input. This property is convenient if the transform coefficients are independently quantized, as is typical in TC schemes.

Example 1.1 (2×2 Transform Coder):

Say that stationary zero-mean Gaussian source $x(m)$ has autocorrelation $r_x(0) = 1$, $r_x(1) = \rho$, and $r_x(k) = 0$ for $k > 1$. For a bit rate of R bits per sample, uniformly-quantized PCM implies a mean-squared reconstruction error of

$$\sigma_r^2\big|_{\text{PCM}} = \frac{\Delta^2}{12}\bigg|_{\substack{\Delta = 2x_{\max}/L \\ L = 2^R}} = \underbrace{\frac{1}{3}\frac{x_{\max}^2}{\sigma_x^2}}_{\gamma_x}\sigma_x^2 2^{-2R} = \gamma_x \sigma_x^2 2^{-2R}.$$

For transform coding, say we choose linear transform

$$\mathbf{T} = \begin{pmatrix} \mathbf{t}_0^t \\ \mathbf{t}_1^t \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Setting $\mathbf{x}(m) = \big(x(2m) \quad x(2m-1)\big)^t$ and $\mathbf{y}(m) = \mathbf{T}\mathbf{x}(m)$, we find that the transformed coefficients have variance

$$\sigma_{y_0}^2 = \mathrm{E}\{|\mathbf{t}_0^t\mathbf{x}(m)|^2\} = \frac{1}{2}\mathrm{E}\left\{|x(2m) + x(2m-1)|^2\right\} = \frac{1}{2}\big(2r_x(0) + 2r_x(1)\big) = 1 + \rho$$

$$\sigma_{y_1}^2 = \mathrm{E}\{|\mathbf{t}_1^t\mathbf{x}(m)|^2\} = \frac{1}{2}\mathrm{E}\left\{|x(2m) - x(2m-1)|^2\right\} = \frac{1}{2}\big(2r_x(0) - 2r_x(1)\big) = 1 - \rho$$

and using uniformly-quantized PCM on each coefficient we get mean-squared reconstruction errors

$$\begin{aligned} \sigma_{q_0}^2 &= (1+\rho)\gamma_x 2^{-2R_0} \\ \sigma_{q_1}^2 &= (1-\rho)\gamma_x 2^{-2R_1}. \end{aligned}$$

We use the same quantizer performance factor γ_x as before since linear operations preserve Gaussianity.

For orthogonal matrices $\mathbf{T}$, i.e., $\mathbf{T}^{-1} = \mathbf{T}^t$, we can show that the mean-squared reconstruction error σ_r^2 equals the mean-squared quantization error:

$$\begin{aligned} \sigma_r^2 &:= \frac{1}{N}\sum_{k=0}^{N-1}\mathrm{E}\big\{(\tilde{x}(Nm-k) - x(Nm-k))^2\big\} \qquad \text{(here } N = 2\text{)} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\tilde{\mathbf{x}}(m) - \mathbf{x}(m)\|^2\right\} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\mathbf{T}^{-1}\tilde{\mathbf{y}}(m) - \mathbf{x}(m)\|^2\right\} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\mathbf{T}^{-1}\left(\mathbf{y}(m) + \mathbf{q}(m)\right) - \mathbf{x}(m)\|^2\right\} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\mathbf{T}^{-1}\mathbf{T}\mathbf{x}(m) + \mathbf{T}^{-1}\mathbf{q}(m) - \mathbf{x}(m)\|^2\right\} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\mathbf{T}^{-1}\mathbf{q}(m)\|^2\right\} \\ &= \frac{1}{N}\mathrm{E}\big\{\mathbf{q}^t(m)\underbrace{(\mathbf{T}^{-1})^t\mathbf{T}^{-1}}_{\mathbf{I}}\mathbf{q}(m)\big\} \\ &= \frac{1}{N}\mathrm{E}\left\{\|\mathbf{q}(m)\|^2\right\} \\ &= \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{q_k}^2. \end{aligned}$$

Since our 2×2 matrix is indeed orthogonal, we have mean-squared reconstruction error

$$\sigma_r^2\big|_{\text{TC}} = \frac{1}{2}\Big((1+\rho)\gamma_x 2^{-2R_0} + (1-\rho)\gamma_x 2^{-2R_1}\Big)$$

at bit rate of $R_0 + R_1$ bits per two samples. Comparing TC to PCM at equal bit rates (i.e. $R_0 + R_1 = 2R$),

$$\frac{\sigma_r^2\big|_{\mathsf{TC}}}{\sigma_r^2\big|_{\mathsf{PCM}}} \;=\; \frac{1}{2}\frac{(1+\rho)\gamma_x 2^{-2R_0} + (1-\rho)\gamma_x 2^{-2(2R-R_0)}}{\gamma_x 2^{-2R}} \;=\; (1+\rho)2^{2(R-R_0)-1} + (1-\rho)2^{2(R_0-R)-1}.$$

Fig. 2 shows that (i) allocating a higher bit rate to the quantizer with stronger input signal can reduce the average reconstruction error relative to PCM, and (ii) the gain over PCM is higher when the input signal exhibits stronger correlation ρ. Also note that when $R_0 = R_1 = R$, there is no gain over PCM—a verification of the fact that $\sigma_r^2 = \sigma_q^2$ when $\mathbf{T}$ is orthogonal.

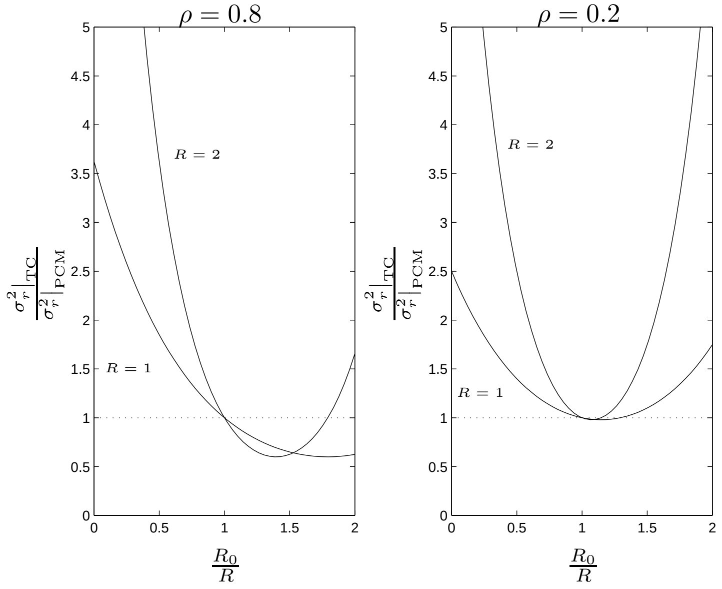

Figure 2: Ratio of TC to PCM mean-squared reconstruction errors versus bit rate R_0 for two values of ρ.

1.2 Optimal Bit Allocation

- *Motivating Question:* Assuming that $\mathbf{T}$ is an $N \times N$ orthogonal matrix, what is the MSE-optimal bitrate allocation strategy assuming independent uniform quantization of the transform outputs? In other words, what $\{R_\ell\}$ minimize average reconstruction error for fixed average rate $\frac{1}{N}\sum_\ell R_\ell$?
- Say that the k^{th} element of the transformed output vector $\mathbf{y}(n)$ has variance $\{\sigma_{y_k}^2\}$. With uniform quantization, Example 1.1 showed that the k^{th} quantizer error power is

$$\sigma_{q_k}^2 \;=\; \gamma_{y_k}\sigma_{y_k}^2 2^{-2R_k}, \tag{1}$$

where R_k is the bit rate allocated to the k^{th} quantizer output and where γ_{y_k} depends on the distribution of y_k. From this point on we make the simplifying assumption that γ_{y_k} is independent of k. As shown in Example 1.1, orthogonal matrices imply that the mean squared reconstruction error equals the mean squared quantization error, so that

$$\sigma_r^2 \;=\; \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{q_k}^2 \;=\; \frac{\gamma_y}{N}\sum_{k=0}^{N-1}\sigma_{y_k}^2 2^{-2R_k}.$$

Thus we have the constrained optimization problem

$$\min_{\{R_k\}}\sum_{k=0}^{N-1}\sigma_{y_k}^2 2^{-2R_k} \quad \text{s.t.} \quad R = \frac{1}{N}\sum_{k=0}^{N-1}R_k.$$

Using the Lagrange technique, we first set

$$\frac{\partial}{\partial R_\ell}\left(\sum_k \sigma_{y_k}^2 2^{-2R_k} - \lambda\frac{1}{N}\sum_k R_k\right) = 0 \qquad \forall\ \ell.$$

Since $2^{-2R_k} = (e^{\ln 2})^{-2R_k} = e^{-2R_k \ln 2}$, the zero derivative implies

$$0 \;=\; -2\ln 2\cdot 2^{-2R_\ell}\cdot\sigma_{y_\ell}^2 - \frac{\lambda}{N} \quad\Rightarrow\quad R_\ell \;=\; -\frac{1}{2}\log_2\left(\frac{\lambda}{-2N\sigma_{y_\ell}^2\ln 2}\right) \quad \forall\ell. \tag{2}$$

Hence

$$R \;=\; \frac{1}{N}\sum_k R_k \;=\; -\frac{1}{2N}\sum_k \log_2\left(\frac{\lambda}{-2N\sigma_{y_k}^2\ln 2}\right) \;=\; -\frac{1}{2}\log_2\left(\frac{\lambda}{-2N\ln 2}\right) + \frac{1}{2N}\sum_k \log_2\sigma_{y_k}^2$$

so that

$$-\frac{1}{2}\log_2\left(\frac{\lambda}{-2N\ln 2}\right) \;=\; R - \frac{1}{2}\log_2\left(\prod_k \sigma_{y_k}^2\right)^{1/N}.$$

Rewriting (2) and plugging in the expression above,

$$\begin{aligned}
R_\ell &= -\frac{1}{2}\log_2\left(\frac{\lambda}{-2N\ln 2}\right) + \frac{1}{2}\log_2\sigma_{y_\ell}^2 \\
&= R - \frac{1}{2}\log_2\left(\prod_k \sigma_{y_k}^2\right)^{1/N} + \frac{1}{2}\log_2\sigma_{y_\ell}^2 \\
&= \boxed{R + \frac{1}{2}\log_2\left(\frac{\sigma_{y_\ell}^2}{\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N}}\right)}.
\end{aligned} \tag{3}$$

- The optimal bitrate allocation expression (3) is meaningful only when $R_\ell \geq 0$, and practical only for integer numbers of quantization levels 2^{-2R_ℓ} (or practical values of R_ℓ for a particular coding scheme). Practical strategies typically
  - set $R_\ell = 0$ to when (3) suggests that the optimal R_ℓ is negative,
  - round positive R_ℓ to practical values, and
  - iteratively re-optimize $\{R_\ell\}$ using these rules until all R_ℓ have practical values (as in the homework).
- Plugging (3) into (1), we find that optimal bit allocation implies

$$\sigma_{q_\ell}^2 \;=\; \gamma_y 2^{-2R}\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N} \qquad \forall\ \ell,$$

which means that, with optimal bit allocation, each coefficient contributes equally to reconstruction error. (Recall a similar property of the Lloyd-Max quantizer.)

1.3 Gain Over PCM

- With an orthogonal transform and the optimal bit allocation (3), the total reconstruction error equals

$$\sigma_r^2\big|_{\text{TC}} \;=\; \frac{1}{N}\sum_{\ell=0}^{N-1}\sigma_{q_\ell}^2 \;=\; \gamma_y 2^{-2R}\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N}. \tag{4}$$

We can compare to uniformly quantized PCM, where $\sigma_r^2\big|_{\text{PCM}} = \gamma_x \sigma_x^2 2^{-2R}$. Since an orthogonal transform implies

$$\sigma_x^2 = \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{y_k}^2,$$

we have the following gain over PCM:

$$\boxed{G_{\text{TC}} \;=\; \frac{\sigma_r^2\big|_{\text{PCM}}}{\sigma_r^2\big|_{\text{TC}}} \;=\; \frac{\gamma_x}{\gamma_y}\frac{\frac{1}{N}\sum_{k=0}^{N-1}\sigma_{y_k}^2}{\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N}}}.$$

Note that the gain is proportional to the ratio between arithmetic and geometric means of the transform coefficient variances. (Note similarities to the spectral flatness measure.)

The factor γ_y/γ_x accounts for changes in distribution which affect uniform-quantizer efficiency. For example, if $\mathbf{T}$ caused uniformly distributed x to become Gaussian distributed y_k, γ_y/γ_x would contribute a 7 dB loss in TC-to-PCM performance. If, on the other hand, x was Gaussian, then y_k would also be Gaussian and $\gamma_y/\gamma_x = 1$.

1.4 Optimal Orthogonal Transform

- Ignoring the effect of transform choice on uniform-quantizer efficiency γ_y, (4) suggests that TC reconstruction error can be minimized by choosing the orthogonal transform $\mathbf{T}$ that minimizes the product of coefficient variances. (Recall that orthogonal transforms preserve the arithmetic average of coefficient variances.)

Aside 1.1 (Eigen-Analysis of Autocorrelation Matrices):
Say that $\mathbf{R}$ is the $N\times N$ autocorrelation matrix of a real-valued, wide-sense stationary, discrete time stochastic process. The following properties are often useful:

1. $\mathbf{R}$ is symmetric and Toeplitz. (A symmetric matrix obeys $\mathbf{R} = \mathbf{R}^t$, while a Toeplitz matrix has equal elements on all diagonals.)
2. $\mathbf{R}$ is positive semidefinite or PSD. (PSD means that $\mathbf{x}^t\mathbf{R}\mathbf{x} \geq 0$ for any real-valued $\mathbf{x}$.)
3. $\mathbf{R}$ has an eigen-decomposition
$$\mathbf{R} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^t,$$
where $\mathbf{V}$ is an orthogonal matrix ($\mathbf{V}^t\mathbf{V} = \mathbf{I}$) whose columns are eigenvectors $\{\mathbf{v}_i\}$ of $\mathbf{R}$:
$$\mathbf{V} = (\mathbf{v}_0\,\mathbf{v}_1\,\cdots\,\mathbf{v}_{N-1}),$$
and $\boldsymbol{\Lambda}$ is a diagonal matrix whose elements are the eigenvalues $\{\lambda_i\}$ of $\mathbf{R}$:
$$\boldsymbol{\Lambda} = \text{diag}(\lambda_0\,\lambda_1\,\cdots\,\lambda_{N-1}).$$
4. The eigenvectors $\{\lambda_i\}$ of $\mathbf{R}$ are real-valued and non-negative.
5. The product of the eigenvectors equals the determinant ($\prod_{k=0}^{N-1}\lambda_k = |\mathbf{R}|$) and the sum of the eigenvalues equals the trace ($\sum_{k=0}^{N-1}\lambda_k = \sum_k[\mathbf{R}]_{k,k}$).

- *The KLT:* Using the outer product,

$$\mathbf{y}(m)\mathbf{y}^t(m) \;=\; \begin{pmatrix} y_0^2(m) & y_0(m)y_1(m) & \cdots & y_0(m)y_{N-1}(m) \\ y_1(m)y_0(m) & y_1^2(m) & \cdots & y_1(m)y_{N-1}(m) \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1}(m)y_0(m) & y_{N-1}(m)y_1(m) & \cdots & y_{N-1}^2(m) \end{pmatrix}$$

Using $\big[\mathbf{A}\big]_{k,k}$ to denote the k^{th} diagonal element of a matrix $\mathbf{A}$, matrix theory implies

$$\begin{aligned}
\prod_{k=0}^{N-1} \sigma_{y_k}^2 &= \prod_{k=0}^{N-1} \big[\mathrm{E}\{\mathbf{y}(m)\mathbf{y}^t(m)\}\big]_{k,k} \\
&\geq \big|\mathrm{E}\{\mathbf{y}(m)\mathbf{y}^t(m)\}\big| \\
&= \big|\mathbf{T}\,\mathrm{E}\{\mathbf{x}(m)\mathbf{x}^t(m)\}\mathbf{T}^t\big| \\
&= \big|\underbrace{\mathbf{T}^t\cdot\mathbf{T}}_{\mathbf{I}}\underbrace{\mathrm{E}\{\mathbf{x}(m)\mathbf{x}^t(m)\}}_{\mathbf{R}_x}\underbrace{\mathbf{T}^t\cdot\mathbf{T}}_{\mathbf{I}}\big| \quad \text{since} \quad |\mathbf{T}^t\mathbf{A}| = |\mathbf{A}| = |\mathbf{A}\mathbf{T}| \quad \text{for orthogonal } \mathbf{T} \\
&= \prod_{k=0}^{N-1} \lambda_k(\mathbf{R}_x),
\end{aligned}$$

thus minimization of $\prod_k \sigma_{y_k}^2$ would occur if equality could be established above. Say that the eigen-decomposition of the autocorrelation matrix of $x(n)$, which we now denote by $\mathbf{R}_x$, is

$$\mathbf{R}_x = \mathbf{V}_x\mathbf{\Lambda}_x\mathbf{V}_x^t$$

for orthogonal eigenvector matrix $\mathbf{V}_x$ and diagonal eigenvalue matrix $\mathbf{\Lambda}_x$. Then choosing $\boxed{\mathbf{T} = \mathbf{V}_x^t}$, otherwise known as the *Karhunen-Loeve transform* (KLT), results in the desired property:

$$\mathrm{E}\{\mathbf{y}(m)\mathbf{y}^t(m)\} = \mathrm{E}\{\mathbf{V}_x^t\mathbf{x}(m)\mathbf{x}^t(m)\mathbf{V}_x\} = \mathbf{V}_x^t\mathbf{R}_x\mathbf{V}_x = \mathbf{V}_x^t\mathbf{V}_x\mathbf{\Lambda}_x\mathbf{V}_x^t\mathbf{V}_x = \mathbf{\Lambda}_x.$$

To summarize:

1. the orthogonal transformation matrix $\mathbf{T}$ minimizing reconstruction error variance has rows equal to the eigenvectors of the input's $N\times N$ autocorrelation matrix,
2. the variances of the optimal-transform outputs $\{\sigma_{y_k}^2\}$ are equal to the eigenvalues of the input autocorrelation matrix, and
3. the optimal-transform outputs $\{y_0(m), \ldots, y_{N-1}(m)\}$ are uncorrelated. (Why? Note the zero-valued off-diagonal elements of $\mathbf{R}_y = \mathrm{E}\{\mathbf{y}(m)\mathbf{y}^t(m)\}$.)

- Note that the presence of mutually uncorrelated transform coefficients supports our approach of quantizing each transform output independently of the others.

Example 1.2 (2×2 **KLT Coder):**

Recall Example 1.1 with Gaussian input having $\mathbf{R}_x = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$. The eigenvalues of $\mathbf{R}_x$ can be determined from the characteristic equation $\big|\mathbf{R}_x - \lambda\mathbf{I}\big| = 0$:

$$\begin{vmatrix} 1-\lambda & \rho \\ \rho & 1-\lambda \end{vmatrix} = (1-\lambda)^2 - \rho^2 = 0 \quad \Leftrightarrow \quad 1-\lambda = \pm\rho \quad \Leftrightarrow \quad \lambda = 1\pm\rho.$$

The eigenvector $\mathbf{v}_0$ corresponding to eigenvalue $\lambda_0 = 1+\rho$ solves $\mathbf{R}_x\mathbf{v}_0 = \lambda_0\mathbf{v}_0$. Using the notation $\mathbf{v}_0 = \left(\begin{smallmatrix} v_{00} \\ v_{01} \end{smallmatrix}\right)$ and $\mathbf{v}_1 = \left(\begin{smallmatrix} v_{10} \\ v_{11} \end{smallmatrix}\right)$,

$$\begin{aligned} v_{00} + \rho v_{01} &= (1+\rho)v_{00} \\ \rho v_{00} + v_{01} &= (1+\rho)v_{01} \end{aligned} \quad \Leftrightarrow \quad v_{01} = v_{00}.$$

Similarly, $\mathbf{R}_x\mathbf{v}_1 = \lambda_1\mathbf{v}_1$ yields

$$\begin{aligned} v_{10} + \rho v_{11} &= (1-\rho)v_{10} \\ \rho v_{10} + v_{11} &= (1-\rho)v_{11} \end{aligned} \quad \Leftrightarrow \quad v_{11} = -v_{10}.$$

For orthonormality,

$$v_{00}^2 + v_{01}^2 = 1 \quad \Rightarrow \quad \mathbf{v}_0 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$v_{10}^2 + v_{11}^2 = 1 \quad \Rightarrow \quad \mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Thus the KLT is given by $\mathbf{T} = \mathbf{V}_x^t = (\mathbf{v}_0\, \mathbf{v}_1)^t = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$.

Using the KLT and optimal bit allocation, the error reduction relative to PCM is

$$\frac{\sigma_r^2\big|_{\mathsf{TC}}}{\sigma_r^2\big|_{\mathsf{PCM}}} = \frac{\gamma_y}{\gamma_x} \cdot \frac{\sqrt{(1+\rho)(1-\rho)}}{\frac{1}{2}\big((1+\rho)+(1-\rho)\big)} = \sqrt{1-\rho^2}$$

since $\gamma_y = \gamma_x$ for Gaussian $x(n)$. This value equals 0.6 when $\rho = 0.8$, and 0.98 when $\rho = 0.2$ (compare to Fig. 2).

1.5 Asymptotic Optimal Gain

- For an $N \times N$ transform coder, (4) presented an expression for the reconstruction error variance $\sigma_r^2\big|_{\text{TC}}$ written in terms of the quantizer input variances $\{\sigma_{y_k}^2\}$. Noting the N-dependence on $\sigma_r^2\big|_{\text{TC}}$ in (4) and rewriting it as $\sigma_r^2\big|_{\text{TC},N}$, a reasonable question might be: What is $\sigma_r^2\big|_{\text{TC},N}$ as $N \to \infty$?

- When using the KLT, we know that $\sigma_{y_k}^2 = \lambda_k$ where λ_k denotes the k^{th} eigenvalue of $\mathbf{R}_x$. If we plug these $\sigma_{y_k}^2$ into (4), we get

$$\sigma_r^2\big|_{\text{TC},N} = \gamma_y 2^{-2R} \left(\prod_{k=0}^{N-1} \lambda_k\right)^{1/N}.$$

Writing $(\prod_k \lambda_k)^{1/N} = \exp(\frac{1}{N}\sum_k \ln \lambda_k)$ and using the Toeplitz Distribution Theorem [1]

$$\text{For any } f(\cdot), \quad \lim_{N\to\infty} \frac{1}{N}\sum_k f(\lambda_k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(S_x(e^{j\omega}))d\omega$$

with $f(\cdot) = \ln(\cdot)$, we find that

$$\begin{aligned} \lim_{N\to\infty} \sigma_r^2\big|_{\text{TC},N} &= \gamma_y 2^{-2R} \exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S_x(e^{j\omega})d\omega\right) \\ &= \gamma_y \sigma_x^2\, 2^{-2R}\, \text{SFM}_x \end{aligned}$$

where SFM_x denotes the spectral flatness measure of $x(n)$, redefined below for convenience:

$$\text{SFM}_x = \frac{\exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S_x(e^{j\omega})d\omega\right)}{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(e^{j\omega})d\omega}.$$

Thus, with optimal transform and optimal bit allocation, asymptotic gain over uniformly quantized PCM is

$$\boxed{G_{\text{TC},N\to\infty} = \frac{\sigma_r^2\big|_{\text{PCM}}}{\sigma_r^2\big|_{\text{TC},N\to\infty}} = \frac{\gamma_x \sigma_x^2\, 2^{-2R}}{\gamma_y \sigma_x^2\, 2^{-2R}\, \text{SFM}_x} = \frac{\gamma_x}{\gamma_y}\, \text{SFM}_x^{-1}}.$$

- Recall that, for the optimal DPCM system,

$$G_{\text{DPCM},N\to\infty} = \frac{\sigma_r^2\big|_{\text{PCM}}}{\sigma_r^2\big|_{\text{DPCM},N\to\infty}} = \frac{\sigma_x^2}{\sigma_e^2\big|_{\min}},$$

where we assumed that the signal applied to DPCM quantizer is distributed similarly to the signal applied to PCM quantizer and where $\sigma_e^2\big|_{\min}$ denotes the prediction error variance resulting from use of the optimal infinite-length linear predictor:

$$\sigma_e^2\big|_{\min} = \exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S_x(e^{j\omega})d\omega\right).$$

Making this latter assumption for the transform coder (implying $\gamma_y = \gamma_x$) and plugging in $\sigma_e^2\big|_{\min}$ yields the following asymptotic result:

$$\boxed{G_{\mathrm{TC},N\to\infty} \;=\; G_{\mathrm{DPCM},N\to\infty} \;=\; \mathrm{SFM}_x^{-1}}.$$

In other words, transform coding with infinite-dimensional optimal transformation and optimal bit allocation performs equivalently to DPCM with infinite-length optimal linear prediction.

- The fact that optimal transform coding performs as well as DPCM in the limiting case does not tell us the relative performance of these methods at practical levels of implementation, e.g., when transform dimension and predictor length are equal and $\ll \infty$. Below we compare the reconstruction error variances of TC and DPCM when the transform dimension *equals* the predictor length.

  Recalling that

$$G_{\mathrm{DPCM},N-1} \;=\; \frac{\sigma_x^2}{\sigma_e^2\big|_{\min,N-1}}$$

  and

$$\sigma_e^2\big|_{\min,N-1} \;=\; \frac{|\mathbf{R}_N|}{|\mathbf{R}_{N-1}|}$$

  where $\mathbf{R}_N$ denotes the $N\times N$ autocorrelation matrix of $x(n)$, we find

$$G_{\mathrm{DPCM},N-1} = \sigma_x^2\frac{|\mathbf{R}_{N-1}|}{|\mathbf{R}_N|}, \quad G_{\mathrm{DPCM},N-2} = \sigma_x^2\frac{|\mathbf{R}_{N-2}|}{|\mathbf{R}_{N-1}|}, \quad G_{\mathrm{DPCM},N-3} = \sigma_x^2\frac{|\mathbf{R}_{N-3}|}{|\mathbf{R}_{N-2}|}, \quad \ldots$$

  Recursively applying the equations above, we find

$$\prod_{k=1}^{N-1} G_{\mathrm{DPCM},k} \;=\; (\sigma_x^2)^{N-1}\frac{|\mathbf{R}_1|}{|\mathbf{R}_N|} \;=\; \frac{(\sigma_x^2)^N}{|\mathbf{R}_N|}$$

  which means that we can write

$$|\mathbf{R}_N| \;=\; (\sigma_x^2)^N\left(\prod_{k=1}^{N-1} G_{\mathrm{DPCM},k}\right)^{-1}.$$

  If in the previously derived TC reconstruction error variance expression

$$\sigma_r^2\big|_{\mathrm{TC},N} \;=\; \gamma_y 2^{-2R}\left(\prod_{\ell=0}^{N-1}\lambda_\ell\right)^{1/N}$$

  we assume that $\gamma_y = \gamma_x$ and apply the eigenvalue property $\prod_\ell \lambda_\ell = |\mathbf{R}_N|$, the TC gain over PCM becomes

$$\begin{aligned} G_{\mathrm{TC},N} \;&=\; \frac{\sigma_r^2\big|_{\mathrm{PCM}}}{\sigma_r^2\big|_{\mathrm{TC},N}} \;=\; \frac{\gamma_x\sigma_x^2 2^{-2R}}{\gamma_x 2^{-2R}\cdot\sigma_x^2\left(\prod_{k=1}^{N-1}G_{\mathrm{DPCM},k}\right)^{-1/N}} \\ &=\; \left(\prod_{k=1}^{N-1} G_{\mathrm{DPCM},k}\right)^{1/N} \\ &<\; G_{\mathrm{DPCM},N}. \end{aligned}$$

  The strict inequality follows from the fact that $G_{\mathrm{DPCM},k}$ is monotonically increasing with k.

  To summarize, DPCM with optimal length-N prediction performs better than TC with optimal $N\times N$ transformation and optimal bit allocation for any finite value of N. There is an intuitive explanation for this: the propagation of memory in the DPCM prediction loop makes the *effective* memory of DPCM greater than N, while in TC the effective memory is exactly N.

1.6 Sub-Optimum Transforms

- *Goal:* Recall that the goal of the optimal orthogonal transform was to minimize the ratio of geometric to arithmetic output variances:

$$\frac{\left(\prod_{k=0}^{N-1}\sigma_{y_k}^2\right)^{1/N}}{\frac{1}{N}\sum_{k=0}^{N-1}\sigma_{y_k}^2}. \tag{5}$$

  The ratio (5) attains its maximum value (= 1) when $\sigma_{y_k}^2$ are equal for all k and takes on much smaller values when the difference between the $\sigma_{y_k}^2$ (sometimes called the dynamic range of $\{\sigma_{y_k}^2\}$) is large.

- *Problem with KLT:* The KLT, i.e., the orthogonal transform minimizing (5), is a function of the input signal statistics. Specifically, the KLT equals the eigenvector matrix of the input autocorrelation matrix. Unfortunately, realistic signals are non-stationary, requiring continual KLT redesign if optimality is to be preserved, and eigenvector computation is computationally intensive, especially for large N. Thus, the question becomes: Are there *fixed* orthogonal transforms that do a good job of minimizing the ratio (5) for "typical" input signals? As we will see, the answer is yes…

- *DFT Intuitions:* For the sake of intuition, lets first consider choosing $\mathbf{T}$ as an orthogonal DFT matrix. In this case, the coefficient variances $\{\sigma_{y_k}^2\}$ would be samples of the power spectrum and the dynamic range of $\{\sigma_{y_k}^2\}$ would be determined by the relative input power in different frequency bands.

  Recalling that asymptotic TC ($N \to \infty$) performance is determined by SFM_x, which has the same geometric-to-arithmetic-average structure as (5):

$$\mathrm{SFM}_x \;=\; \frac{\exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\ln S_x(e^{j\omega})d\omega\right)}{\frac{1}{2\pi}\int_{-\pi}^{\pi}S_x(e^{j\omega})d\omega} \;=\; \lim_{N\to\infty}\left.\frac{\left(\prod_{k=0}^{N-1}S(e^{j\omega_k})\right)^{1/N}}{\frac{1}{N}\sum_{k=0}^{N-1}S(e^{j\omega_k})}\right|_{\omega_k=2\pi k/N},$$

  we might intuit that the DFT is optimal as $N \to \infty$. The asymptotic optimality of the DFT can, in fact, be proven [2]. Of course, we don't have much reason to expect that the DFT would be optimal for finite transform dimension N. Still, for many signals, it performs well. (See Fig. 3.)

- *Other Transforms:* The most commonly used orthogonal transform in speech, image, audio, and video coding is the discrete cosine transform (DCT). The excellent performance of the DCT follows from the fact that it is especially suited to "lowpass" signals, a feature shared by most signals in the previously mentioned applications. Note that there are plenty of signals for which the DCT performs poorly—it just so happens that such signals are not frequently encountered in speech, image, audio, or video. We will describe the DCT and provide intuition regarding it's good "lowpass" performance shortly.

  Like the DFT, the DCT has fast algorithms which make it extremely practical from an implementation standpoint. Another reasonably popular orthogonal transform, requiring even less in the way of computation, is the discrete Hadamard transform (DHT), also described below.

  Fig. 3 compares DFT, DCT, DHT, and KLT for various transform lengths N along with asymptotic TC performance. Fig. 3(a) shows gain over PCM when using a lowpass autoregressive (AR) source $\{x(n)\}$ generated from white Gaussian noise $\{v(n)\}$ via:

$$X(z) \;=\; \frac{1}{1-0.8z^{-1}}V(z),$$

  while Fig. 3(b) shows the gain for highpass $\{x(n)\}$:

$$X(z) \;=\; \frac{1}{1+0.8z^{-1}}V(z).$$

  See Fig. 4 for the power spectra of these two processes.

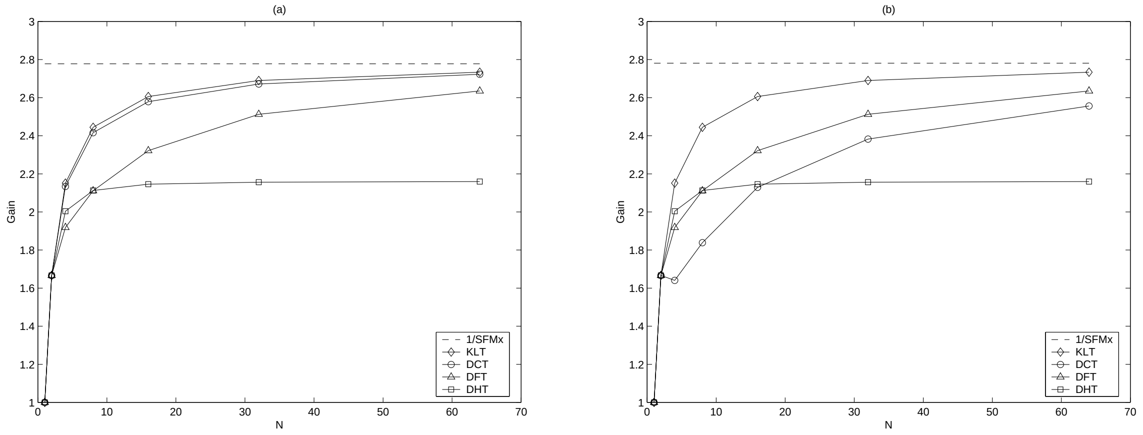

Figure 3: $G_{\mathrm{TC},N}$ for various transforms and various N on an AR(1) lowpass (left) and highpass (right) sources.

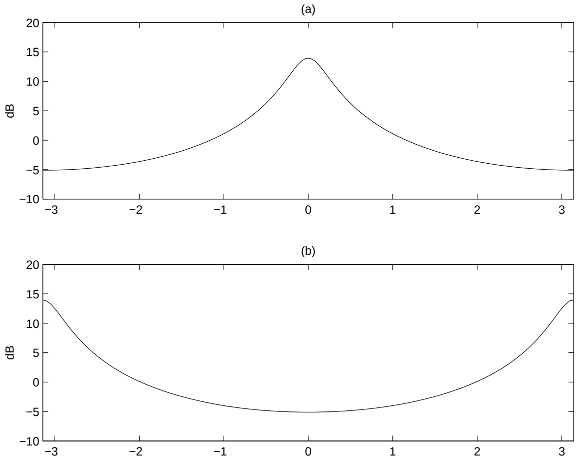

Figure 4: Power spectra of AR(1) sources used in the transform matrix comparisons of Fig. 3.

- *The DHT:* The $N \times N$ DHT is defined below for power-of-two N:

$$\mathbf{H}_2 \;=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \mathbf{H}_{2N} \;=\; \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{pmatrix}$$

Note that $\mathbf{H}_N$ is orthogonal[1], i.e., $\mathbf{H}_N \mathbf{H}_N^t = \mathbf{I}$. As an example

$$\mathbf{H}_4 \;=\; \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Fig. 5 illustrates DHT basis vectors for the case $N = 8$.

The primary advantage of the DHT is that its implementation can be accomplished very efficiently. Fig. 3 suggests that the DHT performs nearly as well as the KLT for $N = 2$ and 4, but its performance falls well short of optimal for larger N.

[1] Caution: outputs of the Matlab command `hadamard` must be scaled by $1/\sqrt{N}$ to produce *orthogonal* DHT matrices!

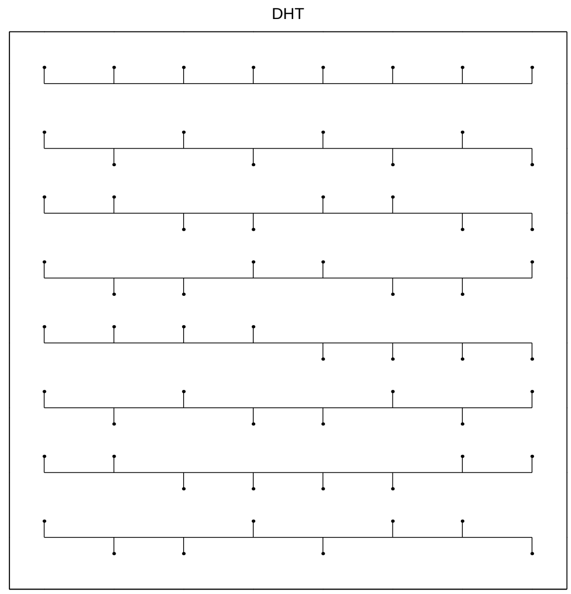

Figure 5: 8×8 DHT basis vectors.

- *The DFT:* The normalized[2] DFT from $\{x_n\}$ to $\{y_k\}$ is defined below along with its inverse.

$$\begin{aligned} y_k &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}; \quad k = 0 \dots N-1, \\ x_n &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{j\frac{2\pi}{N}kn}; \quad n = 0 \dots N-1. \end{aligned}$$

  The normalized DFT can be represented by a symmetric unitary[3] matrix $\mathbf{W}_N$:

$$\big[\mathbf{W}_N\big]_{k,n} = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi}{N}kn}; \quad k,n = 0 \dots N-1.$$

  By unitary, we mean that $\mathbf{W}_N \mathbf{W}_N^{*t} = \mathbf{I}$, where $(\cdot)^*$ denotes complex conjugation. Note that a unitary matrix is the complex-valued equivalent of an orthogonal matrix.

  In practice, the $N \times N$ DFT is implemented using the fast Fourier transform (FFT), which requires $\approx \frac{N}{2} \log_2 N$ complex multiply/adds when N is a power of two.

- *The Real-Valued DFT:* Since we assume real-valued x_n, complex-valued DFT outputs y_k might seem problematic since transmitting both real and imaginary components would decrease our transmission efficiency. For a real-valued DFT input, however, the DFT outputs exhibit conjugate symmetry which allows us to represent the N complex valued outputs with only N real-valued numbers. More precisely, real-valued DFT input $\{x_n\}$ implies that DFT output $\{y_k\}$ has the property

$$y_k = y_{N-k}^*, \qquad k = 1, 2, \dots, N/2,$$

  which implies

$$\begin{aligned} \mathrm{Re}\{y_k\} &= \mathrm{Re}\{y_{N-k}\} \qquad k = 1, 2, \dots, N/2, \\ \mathrm{Im}\{y_k\} &= -\mathrm{Im}\{y_{N-k}\} \qquad k = 1, 2, \dots, N/2, \\ \mathrm{Im}\{y_0\} &= \mathrm{Im}\{y_{N/2}\} = 0. \end{aligned}$$

[2] Due to the norm-preserving scale factor $1/\sqrt{N}$, the DFT definitions above differ from those given in most digital signal processing textbooks.

[3] Outputs of the Matlab command `dftmtx` must be scaled by $1/\sqrt{N}$ to produce *unitary* DFT matrices.

A good method by which to select non-redundant components of the DFT output is:

1. Compute complex-valued $\{y_k\}$ using the standard DFT.
2. Construct real-valued $\{y'_k\}$ from $\{y_k\}$ as follows:

$$\begin{aligned}
y'_0 &= y_0\ (\in \mathbb{R}) \\
y'_1 &= \sqrt{2}\,\mathrm{Im}\{y_1\} \\
y'_2 &= \sqrt{2}\,\mathrm{Re}\{y_1\} \\
&\vdots \\
y'_{N-3} &= \sqrt{2}\,\mathrm{Im}\{y_{N/2-1}\} \\
y'_{N-2} &= \sqrt{2}\,\mathrm{Re}\{y_{N/2-1}\} \\
y'_{N-1} &= y_{N/2}\ (\in \mathbb{R}).
\end{aligned}$$

The method above is convenient because (i) it preserves the frequency ordering of the DFT and (ii) it preserves the norm of the DFT output vector, i.e., $\|\mathbf{y}'\| = \|\mathbf{y}\|$. Using the conjugate symmetry property of DFT outputs, we can write the transformation from $\{y_k\}$ from $\{y'_k\}$ as a matrix operation $\mathbf{U}_N$:

$$\mathbf{y}' = \underbrace{\begin{pmatrix}
1 & 0 & 0 & 0 & \cdots & & & & \cdots & 0 & 0 & 0 \\
0 & -j/\sqrt{2} & 0 & 0 & \cdots & & & & \cdots & 0 & 0 & j/\sqrt{2} \\
0 & 1/\sqrt{2} & 0 & 0 & \cdots & & & & \cdots & 0 & 0 & 1/\sqrt{2} \\
0 & 0 & -j/\sqrt{2} & 0 & \cdots & & & & \cdots & 0 & j/\sqrt{2} & 0 \\
0 & 0 & 1/\sqrt{2} & 0 & \cdots & & & & \cdots & 0 & 1/\sqrt{2} & 0 \\
\vdots & & & & & & & & & & & \vdots \\
0 & & & \cdots & 0 & -j/\sqrt{2} & 0 & j/\sqrt{2} & 0 & \cdots & & 0 \\
0 & & & \cdots & 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 & \cdots & & 0 \\
0 & & & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & & 0
\end{pmatrix}}_{\mathbf{U}_N}
\underbrace{\begin{pmatrix}
\mathrm{Re}\{y_0\} \\
\mathrm{Re}\{y_1\} + j\mathrm{Im}\{y_1\} \\
\mathrm{Re}\{y_2\} + j\mathrm{Im}\{y_2\} \\
\vdots \\
\mathrm{Re}\{y_{N/2-1}\} + j\mathrm{Im}\{y_{N/2-1}\} \\
\mathrm{Re}\{y_{N/2}\} \\
\mathrm{Re}\{y_{N/2-1}\} - j\mathrm{Im}\{y_{N/2-1}\} \\
\vdots \\
\mathrm{Re}\{y_2\} - j\mathrm{Im}\{y_2\} \\
\mathrm{Re}\{y_1\} - j\mathrm{Im}\{y_1\}
\end{pmatrix}}_{\mathbf{y}}$$

The normalization feature guarantees that $\mathbf{U}_N$ is unitary (which is easily checked by inspection). Then $\mathbf{U}_N\mathbf{W}_N$, the product of two unitary matrices, is also unitary. Since $\mathbf{U}_N\mathbf{W}_N$ is actually real-valued (since it takes any real-valued $\mathbf{x}$ to a real-valued $\mathbf{y}'$) it should be referred to as orthogonal rather than unitary. Henceforth we rename $\mathbf{U}_N\mathbf{W}_N$ the *real-valued DFT matrix* $\mathbf{W}^{\mathrm{r}}_N$

$$\mathbf{W}^{\mathrm{r}}_N := \mathbf{U}_N\mathbf{W}_N.$$

It is easily checked that the basis vectors of $\mathbf{W}^{\mathrm{r}}_N$ are sampled sines and cosines at the uniformly spaced frequencies $\{2\pi k/N;\ k = 0,\ldots,N/2\}$. Fig. 6 gives an illustration of the real-valued DFT basis vectors for the case $N = 8$.

Example 1.3 (DFT and Real-valued DFT for $N = 4$):

$$\mathbf{W}_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \end{pmatrix}, \quad \mathbf{W}_4\mathbf{x} = \begin{pmatrix} 4 \\ -1/2 + j3/2 \\ 3 \\ -1/2 - j3/2 \end{pmatrix}.$$

$$\mathbf{W}^{\mathrm{r}}_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & -\sqrt{2} & 0 & \sqrt{2} \\ \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 1 & -1 & 1 & -1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 3 \\ -1 \\ 4 \\ 2 \end{pmatrix}, \quad \mathbf{W}^{\mathrm{r}}_4\mathbf{x} = \begin{pmatrix} 4 \\ 3/\sqrt{2} \\ -1/\sqrt{2} \\ 3 \end{pmatrix}.$$

Recall that when N is a power of 2, an N-dimensional complex-valued FFT requires $\approx \frac{N}{2}\log_2 N$ complex multiply/adds. When the input is real, however, an N-dimensional FFT may be computed using $\approx N\log_2 N$ real multiply/adds [3].

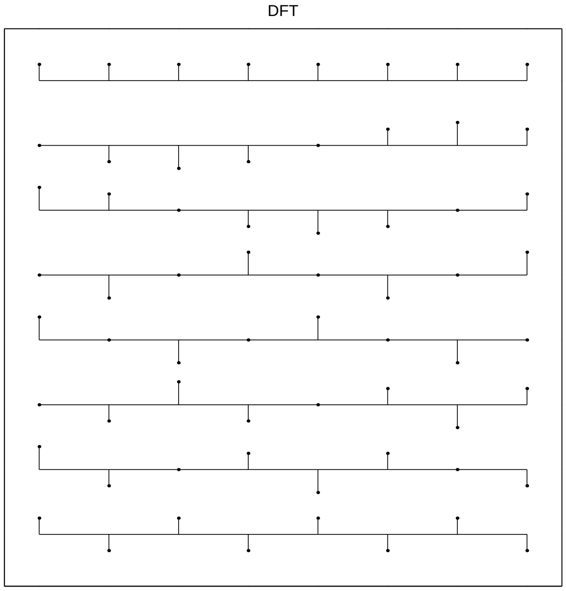

Figure 6: 8×8 Real-valued DFT basis vectors.

- *The DCT:* The DCT is defined below along with its inverse

$$\begin{aligned} y_k &= \sqrt{\frac{2}{N}}\,\alpha_k \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)k\pi}{2N}; \quad k=0\ldots N-1, \\ &\quad \text{for} \ \ \alpha_0 = 1/\sqrt{2}, \ \ \alpha_{k\neq 0}=1, \\ x_n &= \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha_k\, y_k \cos\frac{(2n+1)k\pi}{2N}; \quad n=0\ldots N-1, \end{aligned}$$

  The DCT can be represented by an orthogonal matrix $\mathbf{C}_N$:

$$\big[\mathbf{C}_N\big]_{k,n} = \sqrt{\frac{2}{N}}\,\alpha_k \cos\frac{(2n+1)k\pi}{2N}; \quad k,n=0\ldots N-1.$$

  See Fig. 7 for an illustration of DCT basis vectors when $N=8$.

- *A Fast DCT:* There are a number of fast algorithms to compute the DCT. The method presented below is based on the FFT and leads to intuition concerning the good “lowpass” performance of the DCT.

  1. Create a $2N$-length mirrored version of N-length $\{x_n\}$ (see Fig. 8(c)):

$$\bar{x}_n = \begin{cases} x_n & n=0,1,\ldots,N-1, \\ x_{2N-1-n} & n=N,N+1,\ldots,2N-1. \end{cases}$$

  2. Compute $\{\bar{y}_k\}$, the $2N$-point DFT of $\{\bar{x}_n\}$:

$$\bar{y}_k = \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} \bar{x}_n e^{-j\frac{2\pi}{2N}kn} = \sqrt{\frac{2}{N}} e^{j\frac{\pi}{2N}k} \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)k\pi}{2N}.$$

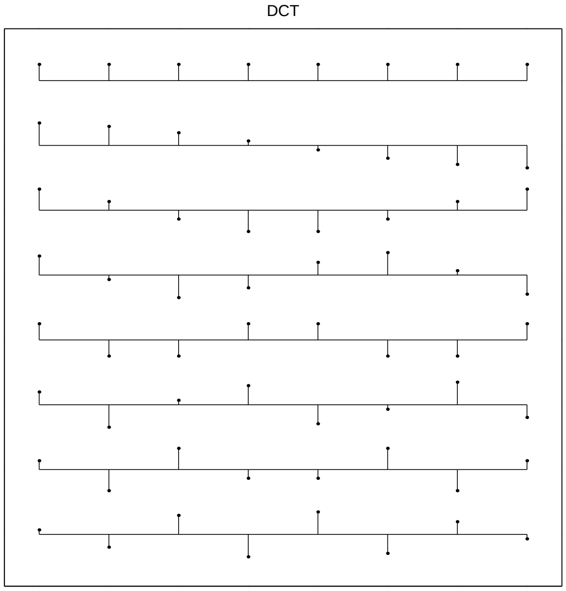

Figure 7: 8×8 DCT basis vectors.

3. Compute $\{y_k\}$, the N-point DCT outputs, from $\{\bar{y}_k\}$:

$$y_k \;=\; e^{-j\frac{\pi}{2N}k}\alpha_k \bar{y}_k; \quad k = 0, 1, \ldots, N-1.$$

Assuming a real-valued input, the scheme outlined above can be implemented using

$$\approx 2N + 2N \log_2 \frac{2N}{2} = 2N(1 + \log_2 N) \;\; \text{real-valued multiply/adds}$$

where we are assuming use of the real-FFT described previously.

- *DCT vs. DFT Performance for Lowpass Signals:* Fig. 3 suggests that DCT and DFT performance both equal KLT performance asymptotically, i.e., as transform dimension $N \to \infty$. Indeed, this can be proven [2]. A more practical question is: How do DCT and DFT performances compare for finite N? To answer this question, we will investigate the effects of input data block length on the DCT and DFT.

  To start, consider the DFT of an N-length input block $\{x_0, \ldots, x_{N-1}\}$:

$$y_k \;=\; \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}; \quad k = 0 \ldots N-1.$$

  It can be seen that the DFT outputs $\{X_0, \ldots, X_{N-1}\}$ are samples of the discrete-time Fourier transform (DTFT) at frequencies $\{\omega_k = \frac{2\pi}{N}k;\; k = 0 \ldots N-1\}$:

$$X(\omega) \;=\; \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-j\omega n}, \qquad y_k \;=\; X\left(\tfrac{2\pi}{N}k\right).$$

  Now lets consider a periodic extension of $\{x_0, \ldots, x_{N-1}\}$ which repeats this N-length sequence a total of L times:

$$x'_n \;=\; \begin{cases} \frac{1}{\sqrt{L}} x_{\langle n \rangle_N} & -\frac{NL}{2} \le n < \frac{NL}{2} \\ 0 & \text{else.} \end{cases}$$

Above, $\langle n \rangle_N$ denotes "n modulo N" and L is assumed even (see Fig. 8(a)-(b)). Here is the interesting point: *the DTFT of the NL-length periodic extension equals the DTFT of the original N-length data block when sampled at the frequencies $\{\omega_k\}$!*

$$\begin{aligned}
X'(\omega_k) &= \frac{1}{\sqrt{NL}} \sum_{n=-NL/2}^{NL/2-1} x'_n e^{-j\frac{2\pi}{N}kn} \\
&= \frac{1}{\sqrt{NL}} \sum_{\ell=-L/2}^{L/2-1} \sum_{m=0}^{N-1} x'_{m+\ell N}\, e^{-j\frac{2\pi}{N}k(m+\ell N)} \\
&= \frac{1}{L\sqrt{N}} \sum_{\ell=-L/2}^{L/2-1} \sum_{m=0}^{N-1} x_m e^{-j\frac{2\pi}{N}k(m+\ell N)} \\
&= \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} e^{-j\frac{2\pi}{N}km} \\
&= X(\omega_k).
\end{aligned}$$

This implies that the overall spectral shape of $X'(\omega)$ will be inherited by the DFT outputs $\{X_0, \ldots, X_{N-1}\}$. So what is the overall shape of $X'(\omega)$?

Say that $\{x_n\}$ is a lowpass process. Being lowpass, we expect the time-domain sequence $\{x_n\}$ to look relatively "smooth." If the starting and ending points of the N-block, are different, however, the periodic extension $\{x'_n\}$ will exhibit time-domain discontinuities (see Fig. 8(b)) that are uncharacteristic of the process $\{x_n\}$. These discontinuities imply that $X'(\omega)$ will contain high-frequency content not present in the power spectrum of the lowpass input process. Based on our previous findings, if artificial high-frequency content exists at $X'(\omega_k)$, it must also exist at $X(\omega_k) = X_k$. In conclusion, the periodic extension $\{x'_n\}$ provides an intuitive explanation of why short-block DFT analysis of lowpass signals often seems corrupted by "artificial" high-frequency content.

So why is this important? Recall that transform performance is proportional to the dynamic range of transform output variances. If the DFT outputs corresponding to otherwise low spectral energy are artificially increased due to short-window effects, the dynamic range of $\{\sigma^2_{y_k}\}$ will decrease, and DFT performance will fall short of optimal.

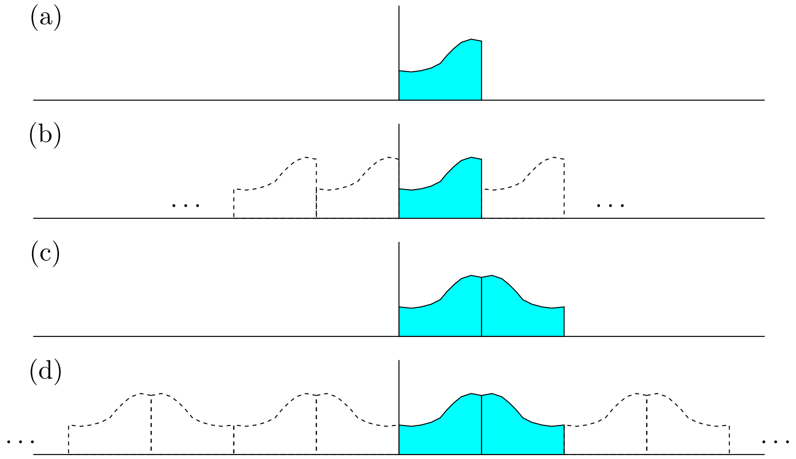

Figure 8: Illustration of periodic extensions inherent to DFT and DCT: (a) N-length DFT input block, (b) periodic extension inherent to DFT, (c) equivalent $2N$-length DFT-input-block for DCT, (d) periodic extension inherent to DCT.

Now lets consider the DCT. From derivation of the fast algorithm, we know that the N DCT output magnitudes from length-N input $\{x_0, \ldots, x_{N-1}\}$ are equal to the first N DFT output magnitudes from length-$2N$ input $\{\bar{x}_n\}$—a mirrored version of $\{x_0, \ldots, x_{N-1}\}$. (See Fig. 8(c).) Due to the mirroring effect, the periodic extension of $\{\bar{x}_n\}$ will not have the discontinuities present in the periodic extension of $\{x_n\}$, and so a $2N$-point DFT analysis of $\{\bar{x}_n\}$ will not have "artificial" high frequency enhancement. Assuming that the process from which $\{x_n\}$ was extracted is lowpass, the DCT outputs will exhibit large dynamic range, and an improvement over DFT coding performance is expected. This is confirmed by Fig. 3(a).

References

[1] U. Grenander and G. Szego, *Toeplitz Forms and Their Applications*, Berkeley, CA: University of California Press, 1958.

[2] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[3] H.V. Sorensen, D.L. Jones, M.T. Heideman, and C.S. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. ??, June, 1987.