TRANSFORM CODING

A way to use scalar quantizers to efficiently encode dependent sources.
Let $X_1, X_2, \ldots$ be a stationary or WSS random process.
Instead of quantizing each $X_i$ with a scalar quantizer

1) Parse data sequence into blocks of some length $k$,

$$X_1, X_2, \ldots = X_1, X_2 \ldots$$

where $X_i = (X_{(i-1)k+1}, \ldots, X_{ik})$; encode each $X_i$ as follows.

2) Find and scalar quantize the coefficients of an orthonormal expansion of

$X$ (=$X_i$) with respect to some orthonormal basis $\{t_1, \ldots, t_k\}$.

a) Find coefficients $U_1, \ldots, U_k$ such that

$$X = U_1 t_1 + \ldots + U_k t_k$$

b) Scalar quantize each $U_i$ with a quantizer that is optimized for it.

The quantized values are

$$V_1 = Q_1(U_1), \ldots, V_k = Q_k(U_k)$$

The encoder output is

$$B = e_1(U_1), \ldots, e_k(U_k)$$

3) The decoder receives $e_1(U_1), \ldots, e_k(U_k)$ and produces

$$Y = V_1 t_1 + \ldots + V_k t_k$$

Key Ideas:

A: Choose $t_1, \ldots, t_k$ so that $U_1, \ldots, U_k$ are uncorrelated.

(If they were correlated, then decorrelating would seem to do better.)

B: Choose $t_1, \ldots, t_k$ so that relatively few $U_i$'s are large, and use higher rate
scalar quantizers on $U_i$'s that are larger on the average, than on $U_i$'s that
are smaller on the average. (This is sometimes called "energy compaction".)

It remains to be seen that more rate is saved on the small $U_i$'s, than is
spent on the large $U_i$'s.

We will show: $A \Leftrightarrow B$. 
MATHEMATICAL DETAILS (FROM LINEAR ALGEBRA)

Definitions:

• $\mathbb{R}^k$ = set of real-valued k-dimensional vectors of the form $x = (x_1, \ldots, x_k)$. We often think of vectors as column vectors.

• Inner or dot product: $(x, y) = (y, x) = x'y = y'x = \sum_{i=1}^{k} x_i y_i$

• Length of $x$: $||x|| = \sqrt{\sum_{i=1}^{k} x_i^2} = \sqrt{(x, x)} = \sqrt{x'x}$

Fact: $||x + y||^2 = (x+y)'(x+y) = ||x||^2 + 2(x, y) + ||y||^2$

• Orthogonality: $x$ and $y$ are orthogonal if $x'y = 0$.

• Linear independence: $x_1, \ldots, x_m$ are linearly independent if there exist no coefficients $u_1, \ldots, u_m$ such that $u_1t_1 + \ldots + u_m t_k = 0$, except $u_1 = u_2 = \ldots = u_m = 0$.

• Basis: A basis for $\mathbb{R}^k$ is a linearly independ. set of vectors $\{t_1, \ldots, t_m\}$ in $\mathbb{R}^k$ that span $\mathbb{R}^k$. That is, every vector $x \in \mathbb{R}^k$ is a linear combination of the members of the set; i.e. there are coefficients $u_1, \ldots, u_m$ such that $x = u_1t_1 + \ldots + u_m t_k$

• Orthonormal basis for $\mathbb{R}^k$: $\{t_1, \ldots, t_k\}$ such that that members of the basis are:
  
  orthogonal: $t_i't_j = 0$ when $i \neq j$, and normalized: $||t_i|| = 1$ for each $i$

Facts:

• Every basis for $\mathbb{R}^k$ has exactly $k$ elements.

• There exists an orthonormal basis for every $k$, e.g. $\{(1,0,\ldots,0), (0,1,0,\ldots,0), \ldots, (0,\ldots,0,1)\}$

• Given a basis $\{t_1, \ldots, t_m\}$ and a vector $x$ there is exists one and only one set of coefficients $u_1, \ldots, u_m$ such that $x = u_1t_1 + \ldots + u_m t_k$

• Given an orthonormal basis $t_1, \ldots, t_k$ and a vector $x$ the coefficients $u_1, \ldots, u_m$ such that $x = u_1t_1 + \ldots + u_m t_k$ are

  $u_i = t_i'x$, $i = 1, \ldots, k$

  Proof: If $x = u_1t_1 + \ldots + u_k t_k$, then

  $t_i'x = t_i'(u_1t_1 + \ldots + u_k t_k) = u_1t_i't_1 + \ldots + u_k t_i't_k = u_i$ since $t_i$'s orthonormal

• If $u = (u_1, \ldots, u_k)$ is the vector of coefficients of the expansion of $x$ with respect to orthonormal basis $\{t_1, \ldots, t_m\}$, then

  $||u||^2 = ||x||^2$

  Proof: $||x||^2 = ||u_1t_1 + \ldots + u_m t_k||^2 = (u_1t_1 + \ldots + u_m t_k)'(u_1t_1 + \ldots + u_m t_k)$

  $= u_1^2 + \ldots + u_k^2$ because $t_i't_j = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$

  $= ||u||^2$
Recall Step 2a of transform coding

2) Given orthonormal basis \( \{ t_1, \ldots, t_k \} \) \( \Rightarrow \) \( \{ (t_{i1}, \ldots, t_{ik})^T \} \)

(a) Find coefficients \( U_1, \ldots, U_k \) such that \( X = U_1 t_1 + \ldots + U_k t_k \)

Therefore, \( U_i = t_i^T X = \sum_{j=1}^k t_{ij} X_j \)

Equivalently, we can compute the coefficients via a matrix multiply

\[
\begin{pmatrix}
U_1 & \ldots & U_k
\end{pmatrix} = (U_1, \ldots, U_k) T X
\]

where \( T \) is \( k \times k \) matrix

\[
T = \begin{bmatrix}
-\frac{t_{11}}{\|t_1\|} & \ldots & -\frac{t_{1k}}{\|t_1\|} \\
\ldots & \ldots & \ldots \\
-\frac{t_{k1}}{\|t_k\|} & \ldots & -\frac{t_{kk}}{\|t_k\|}
\end{bmatrix}
\]

More Definitions:

- In this context the matrix \( T \) is called a transform, and \( T X \) is called a transformation of \( X \). \( U \) is the vector of transform coefficients.
- A matrix with orthonormal rows, such as the ones we are interested in, is called "orthogonal". ("Othonormal" would be a better name, but "orthogonal" is what the mathematicians use.)

Facts: The following are equivalent statements

(a) \( T \) is an orthogonal matrix, (b) Rows of \( T \) are orthonormal
(c) Columns of \( T \) are orthonormal, (d) \( \|T X\| = \|X\| \) for all \( X \)
(e) \( T^{-1} = T^t \), (f) \( T^{-1} \) is orthogonal.

Proofs:

(a) \( \Leftrightarrow \) (b): By definition of an orthogonal matrix

\[
T T^t = \begin{bmatrix}
-\frac{t_{11}}{\|t_1\|} & \ldots & -\frac{t_{1k}}{\|t_1\|} \\
\ldots & \ldots & \ldots \\
-\frac{t_{k1}}{\|t_k\|} & \ldots & -\frac{t_{kk}}{\|t_k\|}
\end{bmatrix} \begin{bmatrix}
| & | & | \\
| & | & | \\
| & | & | \\
\end{bmatrix} = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0 \\
0 & \ldots & 0 & 1
\end{bmatrix}
\]

if and only if the rows of \( T \) are orthonormal. Thus \( T^t \) is the inverse of \( T \) if and only if its rows are orthonormal. (Note that by definition of the inverse \( T^{-1} T = I = T T^{-1} \).)

(c) \( \Leftrightarrow \) (e): By a similar argument \( T^t T = I \) iff the columns of \( T \) are orthonormal.

(b) \( \Rightarrow \) (d): Proved earlier.

(e) \( \Rightarrow \) (d): (e) \( \Rightarrow \) \( \|T X\|^2 = (T X, T X) = (T X)^t T X = x^t T^t T X = x^t X = \|X\|^2 \)

(d) \( \Rightarrow \) (e): Assuming \( \|T X\| = \|X\| \) for all \( X \), we have

\[
0 = \|T X\| - \|X\| = (T X, T X) - (X, X) = (T X)^t T X - x^t x = x^t T^t T X - x^t x = x^t (T^t T - I) x
\]

since this holds for all \( X \) it must be that \( T^t T = I \), i.e. \( T^{-1} = T^t \).

(e) \( \Leftrightarrow \) (f): Elementary.
Fact: For an orthogonal transformation $||T_x-T_y|| = ||x-y||$

Proof: $||T_x-T_y||^2 = (T_x-T_y, T_x-T_y) = (T_x-T_y)^t(T_x-T_y)$

$= (T(x-y))^tT(x-y) = (x-y)^tT^t(x-y)$

$= (x-y)^t(x-y) = ||x-y||^2$

Because an orthogonal transformation preserves the length of vectors and also the distance between vectors, we may view it as essentially a rotation. (There might also be axis flips.)

Examples of orthogonal transformations:

- DFT discrete Fourier transform
- DCT discrete cosine transform
- DWT discrete wavelet transform
- WHT Walsh-Hadamard
- Two-dimensional versions of the above
- KLT Karhunen-Loeve transform

**Example:** DCT Basis Vectors (One-Dimensional)

\[
\begin{align*}
&k = 8 \\
&k = 16
\end{align*}
\]
Example: (Real) FFT Basis Vectors (one-dimensional)

\[ k = 8 \quad k = 16 \]

---

**TRANSFORM CODING**

\[
\begin{array}{c}
X \\
\xrightarrow{\text{orthogonal transformation}} \\
T \\
\mapsto U \\
\xrightarrow{Q_1} V_1 \\
\xrightarrow{Q_2} V_2 \\
\vdots \\
\xrightarrow{Q_k} V_k \\
\xrightarrow{T^{-1}} Y
\end{array}
\]

Performance:

Rate: \[ R = \frac{1}{k} \sum_{i=1}^{k} R_i, \] where \( R_i \) is rate of \( Q_i \) (SQ-FL or SQ-VL).

The transform code is **fixed rate** if its scalar quantizers are fixed-rate (i.e. SQ-FL) and **variable rate**, if its scalar quantizers are variable-rate (i.e. SQ-VL).
Step 1: Optimal quantizers for a given transform.

Choose quantizers \( Q_1, \ldots, Q_k \) to minimize distortion assuming \( \|U - V\| \) are chosen.

Step 2: Optimal transform.

Choose transform \( T \) to minimize distortion assuming \( \|U - V\| \) and \( \|X - Y\| \) are chosen.

This is why we restrict to orthogonal transforms.

Where \( D_{ij} = \|U - V\| \) is the distortion applied to \( U_i \).

\[
D = \sum_{i=1}^{k} E_{i} (X_i - Y_i)^2 = \sum_{i=1}^{k} E_{i} (U_i - V_i)^2
\]

That is, let us find the \( \delta_{\text{OPTA}} \) function.

Let us find the least possible distortion.

The parameters of a transform code to be chosen: dimension \( k \), \( k \times k \) orthogonal matrix \( T \), and scalar quantizers \( Q_1, \ldots, Q_k \).

The derivation is done in two steps.

Optimal Design and the \( \delta_{\text{OPTA}} \) Function
Step 1(a): Given $k$, $T$ and $R_1, \ldots, R_k$ such that $\frac{1}{k} \sum_{i=1}^{k} R_i = R$, choose $Q_1, \ldots, Q_k$ to have rates $R_1, \ldots, R_k$, respectively, and to minimize distortion.

Since

$$D = \frac{1}{k} \sum_{i=1}^{k} D_{U_i, Q_i},$$

for each $i$ we choose $Q_i$ to minimize $D_{U_i, Q_i}$ subject to $R(Q_i) \leq R_i$. Then

$$D_{U_i, Q_i} = \delta_{sq, U_i}(R_i) \overset{A}{=} \text{OPTA for scalar quantization of } U_i.$$

The OPTA for $U_i$ is either the fixed-rate or the variable-rate OPTA, depending on whether we want to optimize transform coding for fixed or variable rate coding.

It follows that the least distortion attainable for the given $k$, $T$, and $R_1, \ldots, R_k$ is

$$D = \frac{1}{k} \sum_{i=1}^{k} \delta_{sq, U_i}(R_i)$$

Step 1(b): Bit/rate allocation

We now seek to find the $R_1, \ldots, R_k$ that minimizes

$$\frac{1}{k} \sum_{i=1}^{k} \delta_{sq, U_i}(R_i)$$

subject to the constraint

$$\frac{1}{k} \sum_{i=1}^{k} R_i \leq R.$$

This is called a bit or rate allocation, i.e. an allocation of $kR$ bits among the scalar quantizers for the individual coefficients. If some are given rate larger than $R$, then others must be given rate less than $R$.

Recall the original idea of transform coding, namely, that some coefficients will be smaller on the average than others, and that these will be encoded at lower rates than those that are larger on the average. If there is to be a net gain, then it will depend on a successful bit allocation.

For fixed-rate coding, the values of $\delta_{sq, U_i}(R_i)$ can be tabulated for the various values of $R_i$ and a numerical optimization algorithm can be applied to find the best choice of the $R_i$'s. See Gersho and Gray, pp. 226-235. Similar methods may be used for variable-rate coding.

We now consider the high-resolution case. In this case, one may solve analytically for the best bit allocation.
HIGH-RESOLUTION ANALYSIS

Assume $R$ is so large that the optimal $R_i$'s are so large that we may use the approximations

$$
\delta_{sq,U_i}(R_i) \approx \frac{1}{T^2} \sigma_i^2 \alpha_i 2^{-2R_i},
$$

where $\sigma_i^2 = \text{variance of } U_i$, and for fixed-rate coding

$$
\alpha_i = \beta_i = \frac{1}{\sigma_i^2} \left( \int f_{U_i}^{1/3}(u) \, du \right)^3
$$

and for variable-rate coding

$$
\alpha_i = \frac{1}{\sigma_i^2} 2^{2h(U_i)}, \text{ where } h(U_i) = -\int_{-\infty}^{\infty} f_{U_i}(u) \log_2 f_{U_i}(u) \, du
$$

We now have

$$
D \approx \frac{1}{k} \sum_{i=1}^{k} \frac{1}{12} \sigma_i^2 \alpha_i 2^{-2R_i}
$$

And the least dist'n for dimen. $k$, transform $T$, and rate $R$ or less is approx'y

$$
\min_{R_1 \geq 0, \ldots, R_k \geq 0: \frac{1}{k} \sum_{i=1}^{k} R_i \leq R} \frac{1}{k} \sum_{i=1}^{k} \frac{1}{12} \sigma_i^2 \alpha_i 2^{-2R_i}
$$

A USEFUL OPTIMIZATION RESULT

**Lemma:** Let $g_1(R), \ldots, g_k(R)$ be functions that are continuous and strictly increasing functions, and let $R_1, \ldots, R_k$ minimize

$$
\sum_{i=1}^{k} g_i(R_i) \text{ subject to } \frac{1}{k} \sum_{i=1}^{k} R_i \leq R.
$$

Then

$$
g_i'(R_i) = g_i'(R_j) \text{ for all } i,j.
$$

That is, at the optimum choice of $R_1, \ldots, R_k$, the slopes of the $g_i$ functions are the same.

(This is the basis of the Lagrange multiplier method.)