Ch. 12 Transform Coding

Goal: Transform the signal (e.g. DFT, etc.) into a new domain where compression can be done either better and/or easier.

Often (but not always!) done on a block-by-block basis:
- non-overlapped blocks (most common)
- overlapped blocks

Block Diagram of Transform Coding

"Fig. A"
We'll view transforms as operators on a vector space (finite dimensional):

\[
X = \begin{bmatrix}
x_0 \\
x_1 \\
\vdots \\
x_{n-1}
\end{bmatrix} \quad Y = \begin{bmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{bmatrix} \quad A = \text{operator}
\]

\[X \xrightarrow{A} Y \implies Y = AX\]

\[\text{Matrix corresponding to operator } A \quad (N \times N \text{ matrix})\]

Need an \underline{invertible} \( A \):

```
X → A → Quantize & Code → Decode → A⁻¹ → X
```

\underline{Usefulness of TC}

1. \underline{Info Theory Advantages}:
   - try to make \( Y \) have uncorrelated elements
   - try to concentrate energy into just a few elements of \( Y \)

2. \underline{Perceptual Distortion Advantages}:
   - transform domain often better suited for exploitation
3. Efficient Implementation
   - TC framework provides simple way to achieve #1 & #2
   - "Extra" cost of transform is not prohibitively large

Need ON Transforms

Using theory of quantization it is easy to assess transform-domain distortion:
\[ d(y, \hat{y}) = \frac{1}{N} \sum_{n=0}^{N-1} (y_n - \hat{y}_n)^2 \]

But what is signal-domain distortion?
\[ d(x, \hat{x}) = ? \]

If transform A is ON then
\[ d(x, \hat{x}) = d(y, \hat{y}) \]

⇒ simplifies understanding of impact of quantization choices in the transform domain
Recall the matrix $A$ for ON transform $A$ has:

- columns that are ON vectors
  \[ a_i^T a_j = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases} \]

- $A^{-1} = A^T \Rightarrow \begin{bmatrix} x = A^T y \\ \hat{x} = A^T \hat{y} \end{bmatrix}$

So... if transform is ON the signal distortion is:

\[
D = E \left\{ (x - \hat{x})^T (x - \hat{x}) \right\}
\]

\[
= E \left\{ (y - \hat{y})^T A A^T (y - \hat{y}) \right\}
\]

\[
= \sum \left\{ (y_n - \hat{y}_n)^2 \right\}
\]

\[
= \sum_{n=0}^{N-1} D_n \quad \text{Distortion of } n^{th} \text{ Transform Coefficient}
\]
Big Picture Result: \[ D = \sum_{n=0}^{N-1} D_n \]

If ON transform then Trans-Domain distortions add to give total distortion in signal domain.

Bit Allocation to TC Quantizers

In "Fig. A" we have \( N \) quantizers operating on the transform coefficients.

Q: How do we decide how many bits each of these should use?

Bit Allocation Problem:

Let

\[ R_b = \text{total Bit Budget} \]

\( R_b \) "Budget"

\[ R_i = \text{# bits allocated to } i^{th} \text{ Quant.} \]

\[ D_i(R_i) = \text{Dist. of } i^{th} \text{ Quant. when allocated } R_i \text{ bits} \]

\[ R = \sum_{i=0}^{N-1} R_i \]

\( R \) Total bits used
Assume distortions are additive: \( D = \sum_{i=0}^{N-1} D_i \)  (true for ON case)

\[ \Rightarrow D(R) = \sum_{i=0}^{N-1} D_i(R_i) \]

**Goal**: Allocate bits \( \{ R_i \}_{i=0}^{N-1} \) to

\[
\text{minimize } D(R) = \sum_{i=0}^{N-1} D_i(R_i)
\]

constrained by \( \sum_{i=0}^{N-1} R_i \leq R_B \)

(Alternate Goal: minimize \( R \) subject to \( D \leq D_B \))

\( D(R) \) vs. Different Allocation Schemes
Aspects of Bit Allocation

1. Theoretical View
   - Theory Drives Algo's

2. Algorithms
   - Average R-D Approach
   - Operational R-D Approach

Bit Allocation Theory

Given known functions $D_i(R_i)$

based on some appropriate signal/quantizer model

Solve the constrained optimization problem for the optimal allocation

$\mathbf{r} = \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_{N-1} \end{bmatrix}$

Interpret result to understand general characteristics
Aside: Constrained Optimization via Lagrange Multipliers


(see Sect. III)

Constrained Minimization:
\[
\min_{B_{\text{ES}}} H(B) \text{ subj. to } R(B) \leq R_c
\]

Theorem: For any \(\lambda \geq 0\), the solution \(B^*(\lambda)\) to the unconstrained problem \(B^*\)

\[
\min_{B_{\text{ES}}} \{H(B) + \lambda R(B)\}
\]

is also the solution to the constrained problem with constraint \(R_c = R(B^*(\lambda)) \equiv R^*(\lambda)\)
Proof: Since $B^*$ is a solution to the unconst. problem:

$$H(B^*) + \lambda R(B^*) \leq H(B) + \lambda R(B)$$

for all $B \in S$

Re-arranging gives:

$$H(B^*) - H(B) \leq \lambda \left[ R(B) - R(B^*) \right] \quad (A)$$

Since this is true for all $B \in S$, it is also true on the subset $S^*$ of $B$ s.t. $R(B) = R(B^*)$

$$\Rightarrow S^* = \{B \mid R(B) = R(B^*)\}$$

Note: $R(B) - R(B^*)$ is negative for all $B \in S^*$

Since $\lambda$ is positive:

$$\Rightarrow H(B^*) - H(B) \leq 0 \quad \forall B \in S^*$$

$$\Rightarrow H(B^*) \leq H(B)$$

$$\Rightarrow H(B^*) \text{ is minimum over all } B \text{ s.t. } R(B) \leq R^*(\lambda)$$

(End of Proof)
What does theorem say?  

To each \( \lambda \geq 0 \) there is

- a constrained problem with constraint \( R_c = R^*(\lambda) \)
  & solution \( B^*(\lambda) \) \( \text{depends on } \lambda \)

- The unconstrained problem \( \min \{ H(B) + \lambda R(B) \} \)
  also has same solution \( B^*(\lambda) \)

So... if we can find closed-forms for \( B^*(\lambda) \) & \( R^*(\lambda) \) as function of \( \lambda \)

then "adjust" \( \lambda = \lambda_c \) so \( R^*(\lambda_c) = R_c \)

we get... \( B^*(\lambda_c) \) solves the constrained problem with our desired constraint!

---

End Aside
The "Aside Theorem" says to:

\[ \text{minimize: } J_\lambda = D(R) + \lambda R \quad (\text{for each } \lambda \geq 0) \]

Using "Additive Distortion" & "Additive Rate"

\[ \Rightarrow J_\lambda = \sum_{i=0}^{N-1} D_i(R_i) + \lambda \sum_{i=0}^{N-1} R_i \]

To find minimum: need \( \frac{\partial J_\lambda}{\partial R_j} = 0 \) \( \forall j \)

\[ \frac{\partial J_\lambda}{\partial R_j} = \frac{\partial D_j(R_j)}{\partial R_j} + \lambda \]

\[ \text{set } = 0 \]

\[ \Rightarrow \frac{\partial D_j(R_j)}{\partial R_j} = -\lambda \quad \forall j \]

〈Aha! Insight!〉 \Rightarrow All the quantizers must operate at an R-D point that has the same slope. "Equal Slopes Requirement"
Intuitive View of "Equal Slopes"

Consider $N=2$

```
\begin{array}{cc}
  \text{Case} & \text{Case} \\
  \begin{array}{c}
   \text{D}_1 \\
   \text{D}_2 \\
   \text{D}_3 \\
   \text{D}_4 \\
  \end{array} & \begin{array}{c}
   \text{R}_1 \\
   \text{R}_2 \\
   \text{R}_3 \\
   \text{R}_4 \\
  \end{array}
\end{array}
```

"Proof" by Contradiction

1. Assume an optimal operating pt. $(R_1^*, D_1^*)$ & $(R_2^*, D_2^*)$ with non-equal slopes

   \[ S_i^* = \frac{dD_i}{dR_i} \bigg|_{R_i=R_i^*} \]

   with $S_1^* \neq S_2^*$

   WLOG: $S_1 = S_2 - \Delta$ w/ $\Delta > 0$

2. Because optimal: $R_1^* + R_2^* = R_B$ (meets budget)

3. Now imagine increasing rate $R_1$:

   \[ R_1^* \rightarrow R_1^* + \varepsilon, \quad \varepsilon > 0 \]
4. To stay within bit budget we must decrease $R_2$ by $E$

$$R_2^* \rightarrow R_2^* - E, \quad E > 0 \text{ (same } E)$$

5. Find New Distortions (approximate using Taylor Series)

$D_1^*$ decreases to $\approx D_1^* + S_1 E$ 

$$= D_1^* + (S_2 - \Delta) E$$

$D_2^*$ increases to $\approx D_2^* - S_2 E$

6. Check New Total Distortion:

$$D_{\text{new}} = [D_1^* + (S_2 - \Delta) E] + [D_2^* - S_2 E]$$

$$= D_1^* + D_2^* - \Delta E$$

old \quad \text{dist} \quad > 0$$

$\Rightarrow$ $D_{\text{new}} < D_{\text{old}}$

(Non-Equal Slopes)

$\Rightarrow$ Original Allocation Not Optimal

$\Rightarrow$ Need Equal Slopes
OK - So we need Equal Slopes!
But... Which Slope?

Here are two cases w/ equal slopes
Q: Which one should we use?

Note: slope #1 gives lower total Rate than slope #2

⇒ choose slope that causes Total Rate = Budget

Recall: All slopes = \(-\lambda\)

⇒ choose \(\lambda\) to meet bit budget

Recall "Aside Theorem": must find \(\lambda=\lambda_c\) that gives \(R(\lambda_c)=R_c\)

⇒ set \(\lambda\) so that unconstrained solution solves our desired constrained problem