Ch. 13 Transform Coding

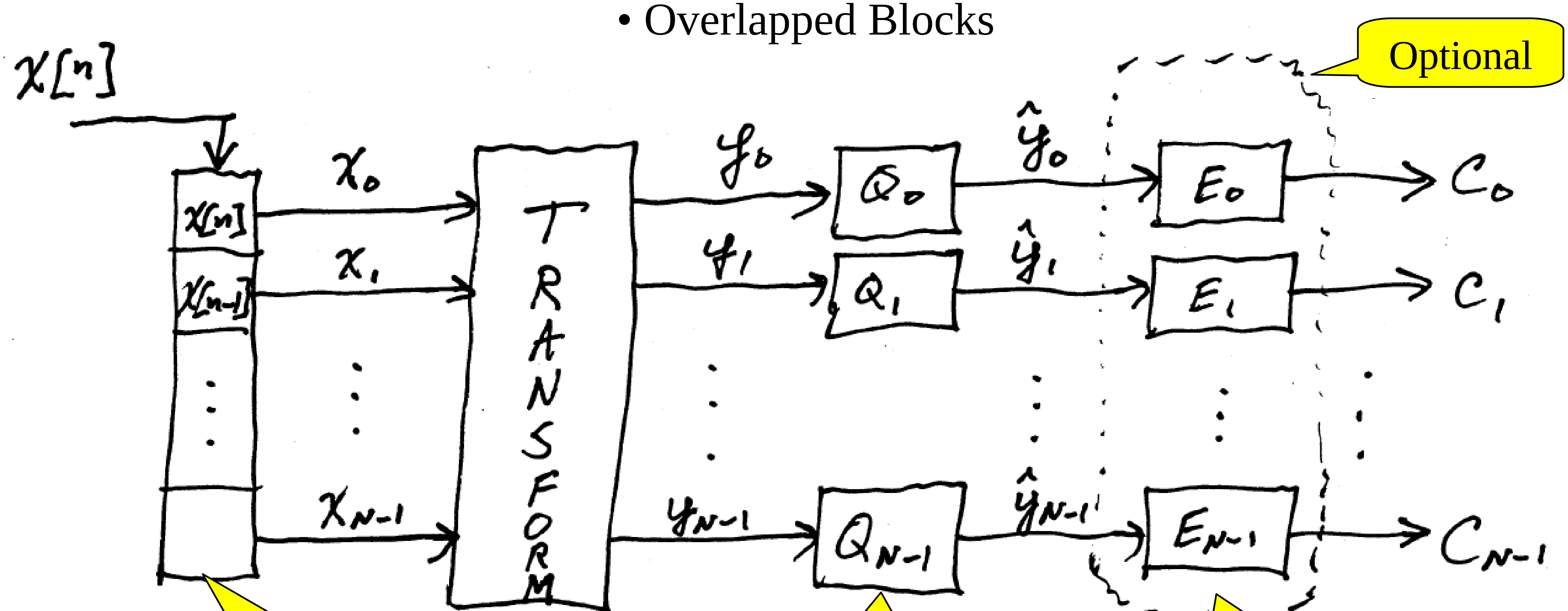
My Coverage is Different from the Book

Overview

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

Often (but not always!) done on a block-by-block basis:

- Non-Overlapped Blocks (most common)
- Overlapped Blocks



"Blocker"

Block Diagram of Transform Coding
"Fig. A"

Quantizers
Scalar
Vector
Differential

Entropy Coders
Huffmann
Arithmethic
Etc.

Transform as Linear Operator

We'll view transforms as linear operators on a vector space (finite dimensional):

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} \qquad \mathbf{x} \xrightarrow{\mathbf{A}} \mathbf{y} \quad \Rightarrow \quad \mathbf{y} = \mathbf{A}\mathbf{x}$$

A = Operator
… an $N \times N$ Matrix

Because at the decoder we need to undo the effect of this operator… we need matrix **A** to be invertible:

Usefulness of Transform Coding

1. Information Theory Advantages
   - Try to make **y** have uncorrelated elements
   - Try to concentrate energy into just a few elements of **y**
2. Perceptual Distortion Advantages
   - Transform domain is often better-suited for exploiting aspects of human perception: psychology of hearing and vision
3. Efficient Implementation
   - Transform Coding framework provides simple way to achieve #1 & #2
   - "Extra" cost of transform is usually not prohibitively large

Need for ON Transforms

Using theory of quantization it is easy to assess transform-domain distortion:

$$d(\mathbf{y},\hat{\mathbf{y}}) = \frac{1}{N}\sum_{n=0}^{N-1}\left(y_n - \hat{y}_n\right)^2$$

But what is the resulting signal-domain distortion?? $d(\mathbf{x},\hat{\mathbf{x}}) = ???$

Fact: If transform **A** is ON then $d(\mathbf{x},\hat{\mathbf{x}}) = d(\mathbf{y},\hat{\mathbf{y}})$

➔ Simplifies understanding of impact of quantization choices in the transform domain

Recall: The matrix **A** for an ON transform has:

- Columns that are ON vectors: $\mathbf{a}_i^T \mathbf{a}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
- Inverse is the transpose: $\mathbf{A}^{-1} = \mathbf{A}^T$ ⇒ $\mathbf{x} = \mathbf{A}^T \mathbf{y}$, $\hat{\mathbf{x}} = \mathbf{A}^T \hat{\mathbf{y}}$

So… if the transform is ON then the signal distortion is:

$$D = E\left\{(\mathbf{x}-\hat{\mathbf{x}})^T(\mathbf{x}-\hat{\mathbf{x}})\right\}$$

$$= E\left\{(\mathbf{y}-\hat{\mathbf{y}})^T \underbrace{\mathbf{A}\mathbf{A}^T}_{=\mathbf{I}}(\mathbf{y}-\hat{\mathbf{y}})\right\}$$

$$= E\left\{(\mathbf{y}-\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}})\right\}$$

$$= E\left\{\sum_{n=0}^{N-1}(y_n-\hat{y}_n)^2\right\}$$

$$= \sum_{n=0}^{N-1} D_n$$

Distortion of n^{th} Transform Coefficient

<u>Big Picture Result</u>: If ON transform, then Transform-Domain distortions add to give total distortion in signal domain

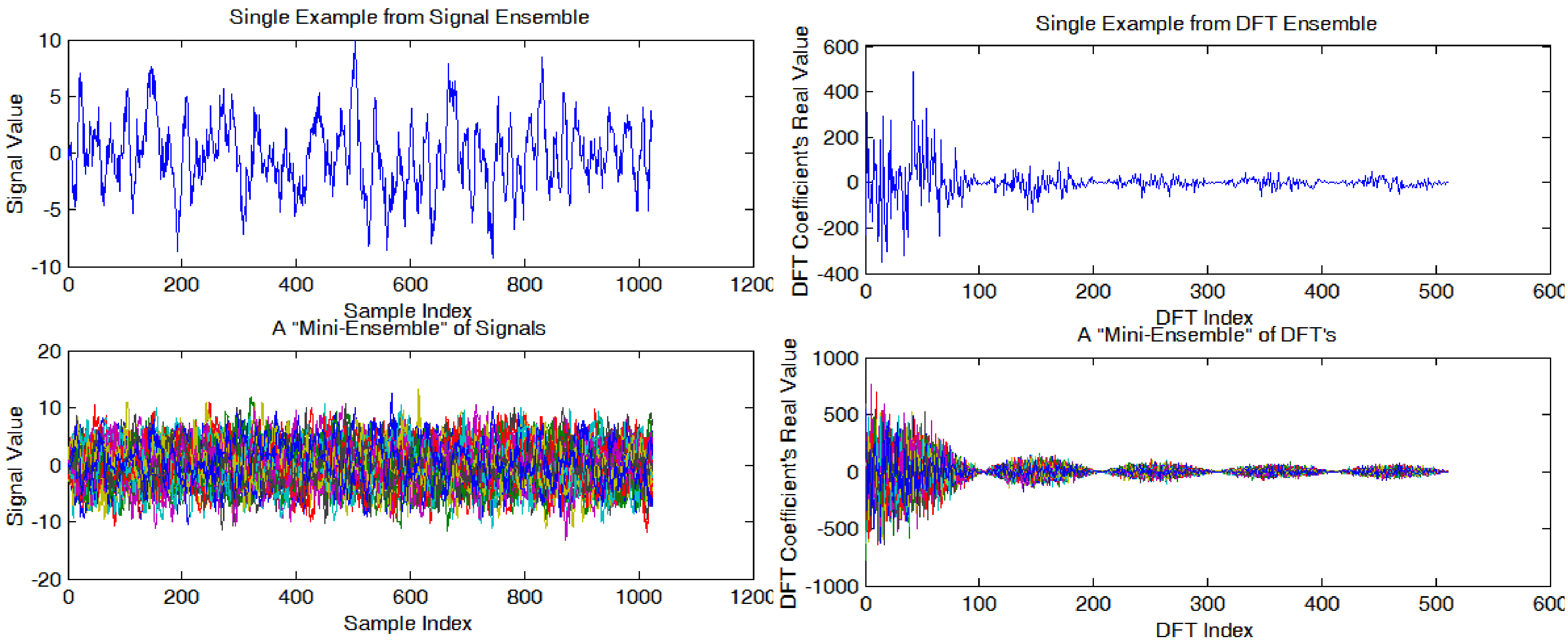
Bit Allocation to TC Quantizers

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?

This is the so-called “Bit Allocation Problem”… we have a constrained total # of bits… how do we allocate them across the N quantizers?

Q: Why not just allocate them evenly???



Bit Allocation Problem

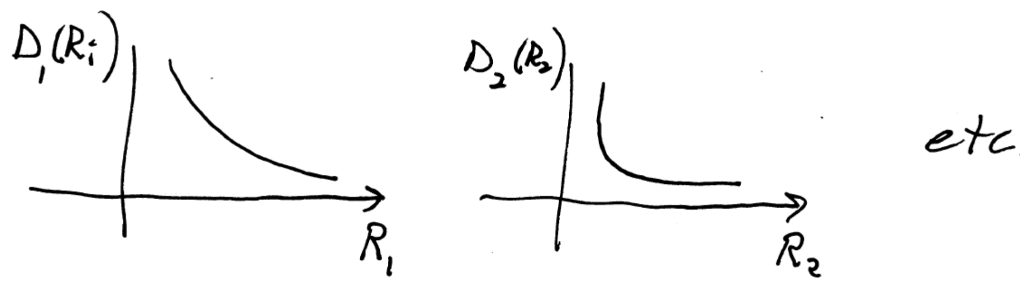
R_B = Total Rate Budget ("Bit Budget")

R_i = # of bits allocated to the i^{th} quantizer ➔ Total Bits Used: $R = \sum_{i=0}^{N-1} R_i$

$D_i(R_i)$ = Distortion of i^{th} quantizer when using R_i bits

Assume distortions are additive (true for ON transform): $D = \sum_{i=0}^{N-1} D_i(R_i)$

Each quantizer has its own R-D curve… depends on quantizer type and char. of the i^{th} transform coefficient



Goal: Allocate bits $\{R_i\}_{i=0}^{N-1}$

to minimize $D = \sum_{i=0}^{N-1} D_i(R_i)$ constrained by $R = \sum_{i=0}^{N-1} R_i \leq R_B$

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

Aspects of Bit Allocation

1. Theoretical View
2. Algorithms
   - Average R-D Approach
   - "Operational" R-D Approach

Theory Drives Algorithms

Models used here determine if we strive for
• Average R-D Solution
• Operational R-D Solution

Bit Allocation Theory

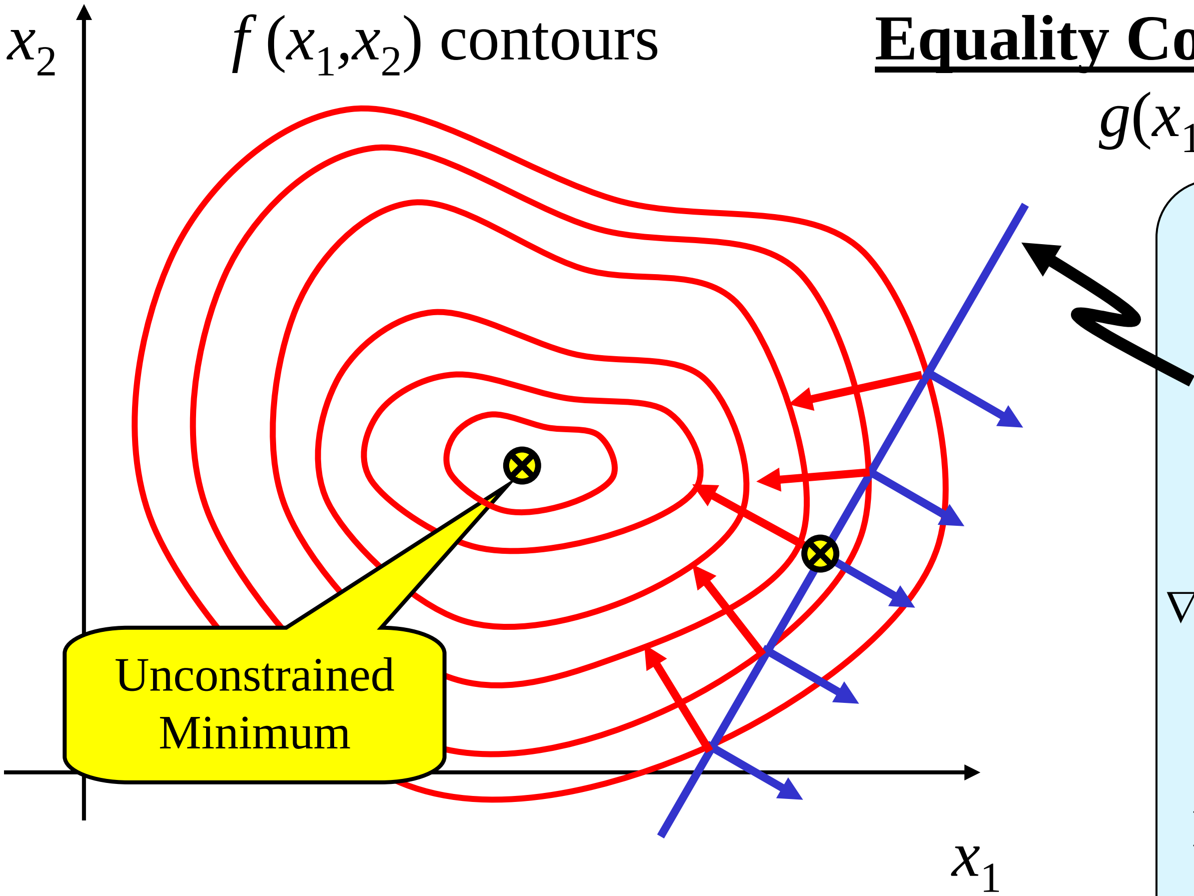
Given known functions $D_i(R_i)$
(Based on some appropriate signal & quantizer models)

Constrained Opt → Lagrange Mult.

Solve the constrained optimization problem for the optimal allocation vector $\mathbf{r} = [R_0 \ R_1 \ \ldots \ R_{N-1}]$

Interpret the result to understand general characteristics

Constrained Optimization: Lagrange Multiplier



Equality Constraint: $g(x_1,x_2) = C$

$g(x_1,x_2) - C = h(x_1,x_2) = 0$

Ex. $ax_1 + bx_2 - c = 0$

$\Rightarrow x_2 = (-a/b)x_1 + c/b$

A Linear Constraint

$$\nabla h(x_1,x_2) = \begin{bmatrix} \dfrac{\partial h(x_1,x_2)}{\partial x_1} \\ \dfrac{\partial h(x_1,x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Ex. The grad vector has "slope" of $b/a \Rightarrow$ orthogonal to constraint line

Constrained Max occurs when:

$$\nabla f\,(x_1,x_2) = -\lambda \nabla h(x_1,x_2)$$

$$\Rightarrow \quad \nabla f\,(x_1,x_2) + \lambda \nabla h(x_1,x_2) = 0$$

$$\nabla_{x_1,x_2,\lambda}\left[\, f\,(x_1,x_2) + \lambda\left(g(x_1,x_2) - C\right)\right] = 0$$

Lagrange Multiplier Approach to Bit Allocation

See paper: Shoham & Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Sept. 1988, pp. 1445 – 1453. **(*See especially Sect. III*)**

Constrained Minimization: $\min_{B \in S} H(B)$ subject to $R(B) \leq R_c$

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

$$\min_{B \in S} \left\{ H(B) + \lambda R(B) \right\}$$

is also the solution to the constrained problem with constraint

$$\min_{B \in S} H(B) \quad \text{subj. to} \quad R(B) \leq \underbrace{R\left(B^*(\lambda)\right)}_{\triangleq R^*(\lambda)}$$

So… for each $\lambda \geq 0$ we find the λ-dependent solution to a λ-dependent unconstrained problem… this solution solves a particular version of the constrained problem, where the constraint is λ-dependent

Proof: Since $B^*(\lambda)$ is a solution to the unconstrained problem

$$H\left(B^*(\lambda)\right) + \lambda R\left(B^*(\lambda)\right) \le \underbrace{H(B) + \lambda R(B)}_{\forall B \in S}$$

Re-arranging this gives: $$H\left(B^*(\lambda)\right) - H(B) \le \lambda\left[R(B) - R\left(B^*(\lambda)\right)\right]$$

Since this is true for $B \in S$ it is true for $B \in S^* \subseteq S$ such that $R(B) \le R(B^*(\lambda))$

$$S^* = \underbrace{\left\{B \mid R(B) \le R(B^*)\right\}}$$

Set of all allocations B that satisfy
λ-dependent constraint $R(B^*(\lambda))$

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

Since λ is positive we have that $H\left(B^*(\lambda)\right) - H(B) \le 0$ ⇨ $H\left(B^*(\lambda)\right) \le H(B) \quad \forall B \in S^*$

⇨ $H\left(B^*(\lambda)\right)$ is minimum over all B s.t. $R(B) \le R(B^*(\lambda))$

⇨ $B^*(\lambda)$ solves the constrained problem with constraint $R(B^*(\lambda))$

<End of Proof>

What does this theorem say?

To each $\lambda \geq 0$…

- there is a constrained problem with: constraint $R^*(\lambda)$ & solution $B^*(\lambda)$ — Both depend on λ
- the unconstrained problem $\min\{H(B)+\lambda R(B)\}$ also has the same solution

So…***if*** we can find closed-forms for $B^*(\lambda)$ & $R^*(\lambda)$ as functions of λ

then we can "adjust" $\lambda = \lambda_c$ so that $R^*(\lambda_c) = R_c$ — Our Actual Constraint

So we get that… $B^*(\lambda_c)$ solves the constrained problem w/ our desired constraint

Applying the Theorem to TC Bit Allocation

We want to minimize $D(R) = \sum_{i=0}^{N-1} D_i(R_i)$ constrained by $R = \sum_{i=0}^{N-1} R_i \le R_B$

The theorem says minimize: $J_\lambda = D(R) + \lambda R$ for arbitrary fixed λ (get results in terms of λ)

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

To minimize… we need to set: $\frac{\partial J_\lambda}{\partial R_j} = 0 \quad \forall j$ $\Rightarrow$ $\frac{\partial J_\lambda}{\partial R_j} = \underbrace{\frac{\partial D_j(R_j)}{\partial R_j} + \lambda}_{\text{set } = 0}$

$$\Rightarrow \quad \boxed{\frac{\partial D_j(R_j)}{\partial R_j} = -\lambda}$$

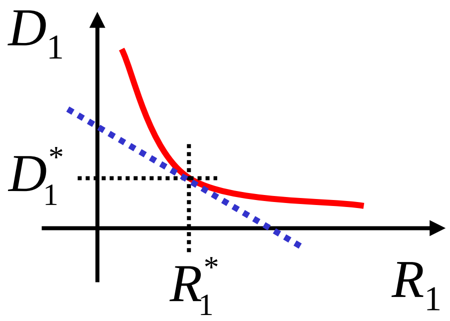
Aha!! Insight!!

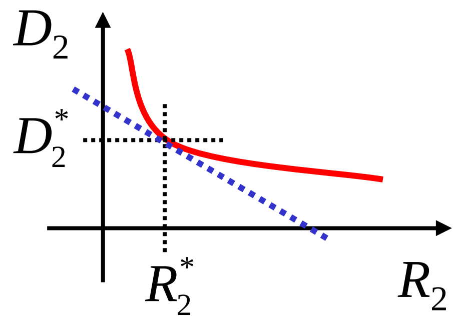
➔ **_All_ the quantizers _must_ operate at an R-D point that has the _same slope_**

"_Equal Slopes Requirement_"

Intuitive View of "Equal Slopes"

Consider N = 2 case





Intuitive "Proof" by Contradiction

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

$$S_i^* \triangleq \left.\frac{\partial D_i}{\partial R_i}\right|_{R_i = R_i^*} \qquad \text{with } S_1^* \neq S_2^* \qquad WLOG: \quad S_1^* = S_2^* - \Delta \quad w/ \quad \Delta > 0$$

2. Because assumed optimal: $R_1^* + R_2^* = R_B$ i.e., meets budget

3. Now… imagine small increase in R_1 for quantizer #1: $R_1^* \to R_1^* + \varepsilon, \quad \varepsilon > 0$

4. To keep the bit budget, must decrease rate R_2 by same small amount:

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

5. Find new distortions due to these rate changes… (Use Taylor series approximations… valid because rate changes were small)

$$D_1^* \text{ decreases to } \approx D_1^* + S_1^* \varepsilon = D_1^* + \left(S_2^* - \Delta\right)\varepsilon$$

$$D_2^* \text{ increases to } \approx D_2^* - S_2^* \varepsilon$$

6. Find new total distortion:

$$D_{New} = \left[D_1^* + \left(S_2^* - \Delta\right)\varepsilon\right] + \left[D_2^* - S_2^* \varepsilon\right]$$

$$= \underbrace{D_1^* + D_2^*}_{D_{Old}} - \underbrace{\Delta\varepsilon}_{>0} \quad \Rightarrow \quad D_{New} < D_{Old}$$

Contradiction!!! We assumed we were optimal (but with non-equal slopes)

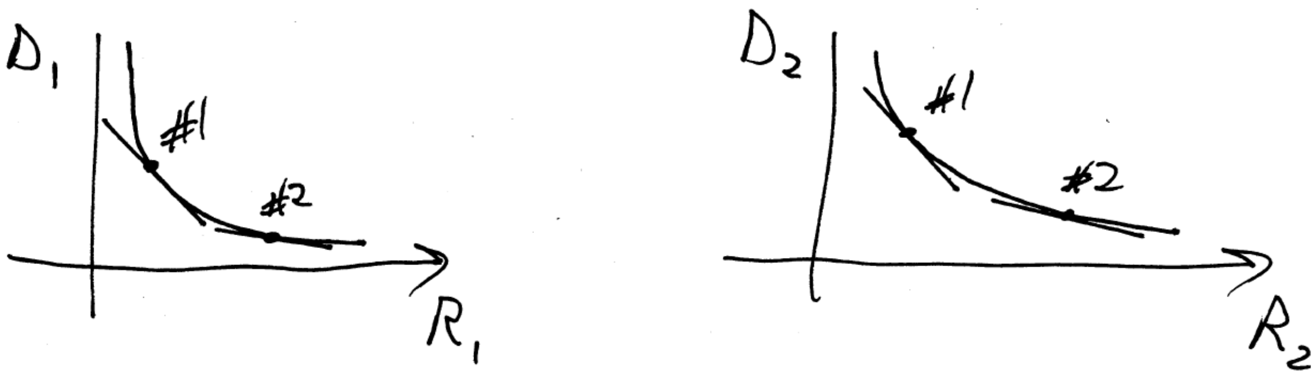
… Yet, we were able to reduce the distortion while meeting bit budget

… So… that non-equal slope operating pt. wasn't optimal after all!!!

… **So, equal slopes must occur at the optimal operating point!!!!**

So We Need Equal Slopes… But Which Slope?

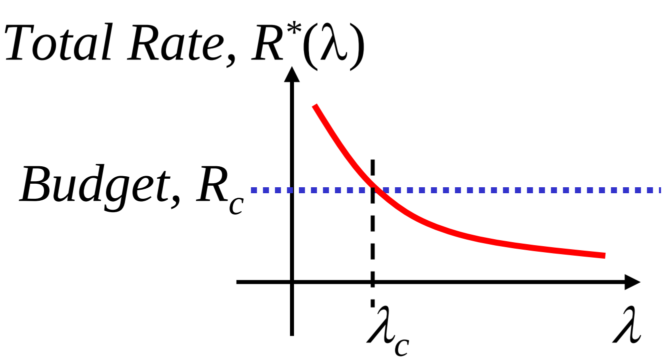
Here are two cases, each with equal slopes… Which should we use?



Note: Slope #1 gives a lower total rate than does Slope #2

➔ Choose Slope that causes Total Rate = Budget Rate

Recall: All slopes = $-\lambda$ ➔ **Choose λ to make Total Rate = Budget Rate**



Recall Theorem

Find $\lambda = \lambda_c$ that gives $R^*(\lambda_c) = R_c$

➔Set λ so that the solution to the unconstrained solution also solves our constrained problem with our constraint