Encoder Control and Quantization
**Quantization**

**Quantizer** $Q$: Encoder mapping $\alpha$ and decoder mapping $\beta$

- **Encoder**: Maps input samples $s$ to a quantizer index $q$
  \[ q = \alpha(s) \]

- **Decoder**: Maps quantizer index $q$ to reconstructed samples $s'$
  \[ s' = \beta(q) = \beta(\alpha(s)) = Q(s) \]

**Scalar quantizer**: One quantization index $q$ per input sample $s$

\[ s' = \beta(q) = \beta(\alpha(s)) = Q(s) \]
Scalar Quantization: Input/Output Function

**Scalar quantizer mapping:**

\[ Q : \mathbb{R} \mapsto \{ s'_0, s'_1, s'_2, \ldots \} \]

**Quantization cells/intervals:**

\[ C_k = [u_k, u_{k+1}) \]

**Quantization step sizes:**

\[ \Delta_k = u_{k+1} - u_k \]
Performance of Scalar Quantizers: Distortion (MSE)

Average MSE distortion $D$ is given by

$$D = E\left\{ \left( S - Q(S) \right)^2 \right\} = \int_{-\infty}^{\infty} \left( s - Q(s) \right)^2 f(s) \, ds = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 f(s) \, ds$$

- Determined by encoder mapping $\alpha$ and decoder mapping $\beta$
- Depending on decision thresholds $u_k$ and reconstruction levels $s'_k$
Performance of Scalar Quantizers: Bit Rate

- Discretization of Pdf

\[ s' = Q(s) \]

- Average bit rate \( R \) is given by

\[ R = \mathbb{E}\{ \ell(\alpha(S)) \} = \sum_{\forall k} p_k \ell_k \quad \text{with} \quad p_k = \int_{u_k}^{u_{k+1}} f(s) \, ds \]

- Minimum rate for scalar entropy code: Entropy

\[ R \geq H = -\sum_{\forall k} p_k \log_2 p_k \]

\( \rightarrow \) Determined by encoder mapping (decision thresholds \( u_k \))
Optimal Scalar Quantizer (MSE Distortion)

**Design Goal**

- Choose decision thresholds $u_k$ and reconstruction levels $s'_k$ so that the distortion $D$ is minimized for given maximum rate $R_{\text{target}}$ (or vice versa)

$$\min D \quad \text{subject to} \quad R \leq R_{\text{target}}, \quad \text{or} \quad \min R \quad \text{subject to} \quad D \leq D_{\text{target}}$$

- Approximate rate $R$ with entropy $H$ (decouple from entropy coding)

$$R = H = - \sum_{\forall k} p_k \log_2 p_k \quad \text{or} \quad \ell_k = - \log_2 p_k$$

- Convert into unconstrained optimization problem

$$\min J \quad \text{with} \quad J = D + \lambda R$$

with Lagrange multiplier $\lambda > 0$ corresponding to $R_{\text{target}}$ (or $D_{\text{target}}$)

**Entropy Constrained Scalar Quantizer (ECSQ)**
Lagrangian Optimization

Points on convex hull: Minimize distance $C$ to line $D = -\lambda \cdot R$

Geometrical interpretation: Rotate coordinate system by angle $\alpha$
Lagrangian Optimization

Minimize distance:  \[ C = D \cdot \cos \alpha + R \cdot \sin \alpha \]

Equivalent minimization:  \[ J = D + \lambda \cdot R \]  (note:  \( \lambda = \tan \alpha \))
**Entropy Constrained Scalar Quantizer (MSE Distortion)**

- Minimize Lagrange function $J$ for given $\lambda$ (with respect to $s'_k$ and $u_k$)

$$J = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 f(s) \, ds + \lambda \cdot \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds$$

1. Optimal reconstruction levels $s'_k$ (for given $u_k$)
   - Convex problem: Set partial derivative equal to zero

$$\frac{\partial}{\partial s'_k} J = 2 \int_{u_k}^{u_{k+1}} (s - s'_k) f(s) \, ds = 0$$

→ Centroid condition for MSE distortion

$$s'_k = \mathbb{E}\{S \mid S \in C_k\} = \frac{1}{p_k} \int_{u_k}^{u_{k+1}} s f(s) \, ds = \frac{1}{\int_{u_k}^{u_{k+1}} f(s) \, ds} \int_{u_k}^{u_{k+1}} s f(s) \, ds$$
Review of Scalar Quantization / Optimal Scalar Quantizer

Entropy Constrained Scalar Quantizer (MSE Distortion)

\[ J = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 f(s) \, ds + \lambda \cdot \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds \]

2 Optimize codeword length \( \ell_k \) (for given \( u_k \))

- Remember: \( R \geq H \)

\[ \sum_{\forall k} p_k \cdot \ell_k \geq - \sum_{\forall k} p_k \cdot \log_2 p_k \quad \text{(equality iff } \ell_k = - \log_2 p_k) \]

⇒ **Entropy condition** (neglecting inefficiency of actual entropy coding)

\[ \ell_k = - \log_2 p_k = - \log_2 \left( \int_{u_k}^{u_{k+1}} f(s) \, ds \right) \]
Entropy Constrained Scalar Quantizer (MSE Distortion)

\[ J = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 f(s) \, ds + \lambda \cdot \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds \]

3. Optimize decision thresholds \( u_k \) (for given \( s'_k \) and \( \ell_k \))

- Map each \( s \) to interval \( C_k \) that minimizes its contribution \( J(s) \) to \( J \)

\[ J(s) = (s - s'_k)^2 + \lambda \cdot \ell_k \]

- For threshold \( u_k \) between \( C_{k-1} \) and \( C_k \), we require

\[ (u_k - s'_{k-1})^2 + \lambda \cdot \ell_{k-1} = (u_k - s'_k)^2 + \lambda \cdot \ell_k \]

- Modified nearest neighbor condition

\[ u_k = \frac{1}{2} (s'_{k-1} + s'_k) + \frac{\lambda}{2} \left( \frac{\ell_k - \ell_{k-1}}{s'_k - s'_{k-1}} \right) \]
Iterative Design of ECSQs

1. Choose a Lagrange multiplier $\lambda > 0$ (determines R-D trade-off)
2. Choose an initial set of $s'_k$ and associated $\ell_k$ (very large set!)
3. Update decision thresholds $u_k$ (for given $s'_k$ and $\ell_k$)
   \[
   u_k = \frac{1}{2} (s'_{k-1} + s'_k) + \frac{\lambda}{2} \left( \frac{\ell_k - \ell_{k-1}}{s'_k - s'_{k-1}} \right)
   \]
4. Update reconstruction levels $s'_k$ and codeword lengths $\ell_k$ (given $u_k$)
   \[
   s'_k = \frac{1}{p_k} \int_{u_k}^{u_{k+1}} s \, f(s) \, ds \quad \text{with} \quad p_k = \int_{u_k}^{u_{k+1}} f(s) \, ds
   \]
   \[
   \ell_k = -\log_2 p_k
   \]
5. Repeat the previous three steps until convergence

Note: Can also use a large training set $\{s\}$ instead of pdf $f(s)$
Uniform Reconstruction Quantizer (URQ)

- All rec. levels $s'_k$ given by single parameter: **Quantization step size** $\Delta$
  
  $$s'_k = k \cdot \Delta$$

- Very simple decoder mapping
  
  $$s' = q \cdot \Delta$$  \hspace{1cm} (q: transmitted quantizer index)

**Can still freely choose / optimize decision thresholds** $u_k$
Optimization of Uniform Reconstruction Quantizers

Lagrangian Optimization

- Choose Lagrange parameter $\lambda > 0$ (determining R-D trade-off)
- Determine decision thresholds $u_k$ and quantization step size $\Delta$

$\Rightarrow$ Minimization of Lagrangian cost $J$ with $s'_k = k\Delta$

$$J = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - k\Delta)^2 f(s) \, ds + \lambda \cdot \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds$$

$\Rightarrow$ Same conditions for $u_k$ and $\ell_k$ as for ECSQ (with $s'_k = k\Delta$)

$$u_k = \Delta \left( k - \frac{1}{2} \right) + \frac{\lambda}{2\Delta} (\ell_k - \ell_{k-1}) \quad \text{[for given $\Delta$ and $\ell_k$]}$$

$$\ell_k = -\log_2 \int_{u_k}^{u_{k+1}} f(s) \, ds \quad \text{[for given $u_k$]}$$

$\Rightarrow$ How to select quantization step size $\Delta$?
Optimization of Quantization Step Size

\[ J = \sum_\forall k \int_{u_k}^{u_{k+1}} (s - k\Delta)^2 f(s) \, ds + \lambda \cdot \sum_\forall k \ell_k \int_{u_k}^{u_{k+1}} f(s) \, ds \]

→ Convex optimization problem: Optimal \( \Delta \) for given \( u_k \)

\[
\frac{\partial}{\partial \Delta} J = \frac{\partial}{\partial \Delta} \left( \sum_\forall k \int_{u_k}^{u_{k+1}} (s^2 - 2k\Delta + k^2\Delta^2) f(s) \, ds \right)
\]

\[
= 2\Delta \sum_\forall k k^2 \int_{u_k}^{u_{k+1}} f(s) \, ds - 2 \sum_\forall k k \int_{u_k}^{u_{k+1}} s f(s) \, ds = 0
\]

→ Condition for quantization step size \( \Delta \)

\[
\Delta = \frac{\sum_\forall k k \int_{u_k}^{u_{k+1}} s f(s) \, ds}{\sum_\forall k k^2 \int_{u_k}^{u_{k+1}} f(s) \, ds}
\]
Iterative Design of Optimal URQs

1. Choose a Lagrange multiplier $\lambda > 0$ (determines R-D trade-off)
2. Choose an initial quantization step size $\Delta$ and codeword lengths $\ell_k$
3. Update decision thresholds $u_k$ (for given $\Delta$ and $\ell_k$)

$$u_k = \Delta \left( k - \frac{1}{2} \right) + \frac{\lambda}{2\Delta} (\ell_k - \ell_{k-1})$$

4. Update codeword lengths $\ell_k$ (given $u_k$)

$$\ell_k = -\log_2 \left( \int_{u_k}^{u_{k+1}} f(s) \, ds \right)$$

5. Update quantization step size $\Delta$ (given $u_k$)

$$\Delta = \frac{\sum_{\forall k} k \int_{u_k}^{u_{k+1}} s f(s) \, ds}{\sum_{\forall k} k^2 \int_{u_k}^{u_{k+1}} f(s) \, ds}$$

6. Repeat the previous three steps until convergence
Performance Comparison: Optimal URQ vs ECSQ

Coding efficiency of Optimal URQs

- Very small loss relative to ECSQ for typical pdfs
- URQ = Reasonable structural constraint
- Important for performance: Optimize decision thresholds!
Uniform Reconstruction Quantizer / Comparison to ECSQ

Quantization in Transform Coding

Bit Allocation among Transform Coefficients (see lecture on JPEG)

- Optimal: ECSQs designed at constant $\lambda$
- Optimal with URQs: Optimal URQs designed at constant $\lambda$
- In practice: **URQs with constant step size $\Delta$**
  - Simple decoder and low side information:
    One quantization step size $\Delta$ for complete transform block
  - Small coding efficiency loss compared to optimal design

Encoding problem: How to select quantization indexes?

- Simple rounding yields noticeable losses in coding efficiency
  - Pdfs for transform coefficients are typically unknown
  - Entropy codes are typically specified in standard
  - Entropy codes exploit dependencies between transform coefficients
  - Difficult to optimize decision thresholds during encoding
  - **Can we use simpler approach?**
General Encoding Problem in Image and Video Coding

Given

- Bitstream syntax (format for transmitting coding parameters)
- Decoding process (algorithm for reconstructing video pictures)

Coding Efficiency

- Maximum achievable coding efficiency is determined by set of syntax features and coding tools supported in bitstream syntax and decoding process

  **Coding efficiency of a bitstream is determined by encoding process**

  - Selection of coding modes
  - Selection of motion parameters
  - Selection of transform coefficient levels (quantization indexes)

Main Encoding Problem

- Select coding parameters such that the coding efficiency is maximized
- Have to consider encoding delay and complexity of the algorithm
General Encoding Problem in Image and Video Coding

Encoding problem for given input video $s_v$

- Generate a conforming bitstream $b \in B$ such that the distortion $D(s_v, s'_v(b))$ between the input video $s_v$ and its reconstruction $s'_v(b)$ is minimized while the bit rate $R(b)$ does not exceed a given bit rate budget $R_B$

$$b^* = \arg \min_{b \in B} D(s_v, s'_v(b)) \quad \text{subject to} \quad R(b) \leq R_B$$

Equivalent problem

- Generate a conforming bitstream $b \in B$ such that the bit rate $R(b)$ is minimized while the distortion $D(s_v, s'_v(b))$ between the input video $s_v$ and its reconstruction $s'_v(b)$ does not exceed a given maximum distortion $D_{\text{max}}$

$$b^* = \arg \min_{b \in B} R(b) \quad \text{subject to} \quad D(s_v, s'_v(b)) \leq D_{\text{max}}$$

- Impossible to find optimal solution (extremely large parameter space)
- **Split into smaller sub-problems**
Lagrangian Optimization for Discrete Sets

Constrained Optimization Problem

- Consider set of samples $s$ (block, picture, or set of pictures)
- Discrete vector of coding parameters $p$
- Constrained problem for given rate budget $R_B$, with $D(p) = D(s, s'(p))$

$$\min_p D(p) \quad \text{subject to} \quad R(p) \leq R_B$$

- Varying $R_B$: Optimal coding parameter vectors $\{p_{opt}\}$

Unconstrained Optimization Problem

- Using Lagrange multipliers $\lambda \geq 0$, we obtain the unconstrained problem

$$p_\lambda^* = \arg \min_p D(p) + \lambda \cdot R(p)$$

- Cannot find all optimal solutions $\{p_{opt}\}$
- But: Each solution $p_\lambda^*$ is an optimal solution, $\{p_\lambda^*\} \subseteq \{p_{opt}\}$
Consider solution $p_*^\lambda$ for a particular value of $\lambda$, with $\lambda \geq 0$

By definition, we have

$$\forall p, \quad D(p) + \lambda \cdot R(p) \geq D(p_*^\lambda) + \lambda \cdot R(p_*^\lambda)$$

$$D(p) - D(p_*^\lambda) \geq \lambda \cdot (R(p_*^\lambda) - R(p))$$

Consider all parameter vectors $p$ with $R(p) \leq R(p_*^\lambda)$

Since $\lambda \geq 0$, the above inequality implies

$$\forall p \in \mathcal{P} : \quad R(p) \leq R(p_*^\lambda), \quad D(p) \geq D(p_*^\lambda)$$

Hence, $p_*^\lambda$ is a solution of the constrained problem

$$p_*^\lambda = \arg\min_p D(p) \quad \text{subject to} \quad R(p) \leq R_B = R(p_*^\lambda)$$

Each solution of the unconstrained optimization problem is also a solution of original constrained optimization problem.
Solutions of unconstrained optimization problem

- Subset of solutions of unconstrained optimization problem
- Minimize distance $d$ to lines $D = -\lambda \cdot R$
- Lie on convex hull of area of all possible rate-distortion points
Lagrangian Bit Allocation

Lagrangian Optimization for Independent Subsets

- Consider partitioning of \( s \) into independent subsets \( s_k \)
- Consider any additive distortion measure, \( D = \sum D_k \)
- Overall optimization problem

\[
\{ p_0^*, p_1^*, \cdots \} = \arg \min_{p_0, p_1, \cdots} \sum_{\forall k} D_k(p_k) + \lambda \sum_{\forall k} R_k(p_k)
\]

Can be solved by separate minimizations

\[
\forall k, \quad p_k^* = \min_{p_k} D_k(p_k) + \lambda R_k(p_k)
\]

Key Advantage of Lagrangian Optimization

- Global optimization problem can be solved by separate minimizations
- Yields optimal bit allocation \( \{ R_0, R_1, \cdots \} \)
Example: 5 subsets (A,B,C,D,E), each with 6 operating points

- For entire set: \(6^5 = 7776\) coding options
- Constrained optimization: Evaluate all 7776 combinations
- Lagrangian approach: Only \(5 \cdot 6 = 30\) comparisons required
Lagrangian Optimization in Video Encoders

Decisions for blocks are **not independent** of each other

- Motion-compensated prediction
- Motion vector prediction
- Intra-picture prediction
- Conditional entropy coding (e.g., run-level coding)

**Concept of Lagrangian optimization is still applicable**

- Partly neglect dependencies between coding decisions
- Approach with same complexity as the method for independent sets

\[
\min_{p_k} D_k(p_k \mid p_{k-1}, p_{k-2}, \cdots) + \lambda \cdot R_k(p_k \mid p_{k-1}, p_{k-2}, \cdots)
\]

- Past decisions \( \{p_{k-1}, p_{k-2}, \cdots\} \) are taken into account (by using correct predictors and conditional entropy codes)
- Impact on decisions for following blocks is ignored
Decisions on Block Level

Still very large parameter space for single block
  - Split decision process for a block into smaller problems
    - Mode decision
    - Motion estimation
    - Quantization
  - Use different amount of simplifications for sub-problems

Distortion measure for encoder decisions
  - Require simple additive distortion measure

\[ D_k = \sum_{s_i \in s_k} |s_i - s_i'|^\beta \]

- \( \beta = 2 \): Sum of squared differences (SSD) \( \Rightarrow \) Mode decision & quantization
- \( \beta = 1 \): Sum of absolute differences (SAD) \( \Rightarrow \) Motion estimation
Lagrangian Encoder Control / Modern Hybrid Video Encoders

Lagrangian Mode Decision

Consider small set \( C_k \) of coding modes for block \( s_k \)
- Example: \( C_k = \{ \text{Intra, Inter} \} \)
- Associated parameters are determined in advance
  - Motion vectors, intra prediction modes, transform coefficient levels, ...
- Each coding mode \( c \in C_k \) is associated with coding parameters \( p_k(c) \)

Lagrangian mode decision
- Subset of parameter space is small enough for testing all included parameter vectors

\[
c_k^* = \arg \min_{c \in C_k} D_k(c \mid p_k(c), p_{k-1}, \cdots) + \lambda \cdot R_k(c \mid p_k(c), p_{k-1}, \cdots)
\]

with
- \( D_k(c \mid \cdot) \) – SSD between original block \( s_k \) and its reconstruction \( s'_k(p_k(c)) \)
- \( R_k(c \mid \cdot) \) – Number of bits for transmitting coding parameters \( p_k(c) \)
Lagrangian Mode Decision in Practice

For all blocks in coding order

1. Check intra coding mode
   - Perform intra prediction (if supported, will be discussed later)
   - Perform quantization and reconstruction
   - Measure distortion $D_{\text{intra}}$ between original and reconstructed block
   - Measure number of bits $R_{\text{intra}}$ (for mode, transform coefficient levels)

2. Check inter coding mode
   - Perform motion estimation and motion-compensated prediction
   - Perform quantization (of prediction error signal) and reconstruction
   - Measure distortion $D_{\text{inter}}$ between original and reconstructed block
   - Measure $R_{\text{inter}}$ (for mode, motion vectors, transform coefficient levels)

3. Check additional coding mode
   - ...

4. Choose coding mode $m \in \{\text{intra, inter, \cdots} \}$ that minimizes

$$J_m = D_m + \lambda \cdot R_m$$
Lagrangian Encoder Control / Modern Hybrid Video Encoders

## Lagrangian Encoder Control in Practice

### Lagrangian Mode Decision
- Decision between intra- and inter picture coding
- Determination of intra picture coding modes
- Decision whether block is subdivided into smaller blocks
- Selection of transform size (subdivision of blocks for transform coding)

### Lagrangian Motion Estimation
- Use Lagrangian cost in motion search
- Consider number of bits required for coding motion vectors
- Will be discussed later in more detail

### Lagrangian Quantization
- Use Lagrangian cost for selecting transform coefficient levels
- Discuss in more detail
Quantization of Transform Coefficients

Transform Coding in Image and Video Codecs

- Orthogonal transforms
- SSD distortion in sample space = SSD distortion in transform domain

\[ D = \sum_{k} (s_k - s'_k)^2 = \sum_{k} (t_k - t'_k)^2 \]

Modern Video Codecs: Uniform reconstruction quantizers (URQs)

- Inverse quantizer mapping

\[ t'_k = \Delta \cdot q_k \]

- Distortion for vector \( q \) of quantization indexes is given by

\[ D(q) = \sum_{k=0}^{N-1} D_k(q_k) = \sum_{k=0}^{N-1} (t_k - \Delta q_k)^2 \]
**Lagrangian Optimization for Quantization**

**Simple Approach:** Minimize SSD distortion for given quantization step size $\Delta$

- SSD distortion is minimized by simple rounding according to
  
  $$q_k = \text{sgn}(t_k) \left[ \frac{|t_k|}{\Delta} + \frac{1}{2} \right]$$

- Does not consider rate required for transmitting quantization indexes $q_k$

**Lagrangian optimization**

- Improve coding efficiency by taking into account bit rate
  
  $$q^* = \arg \min_{q \in Q^N} D(q) + \lambda \cdot R(q)$$

- Entropy coding exploits dependencies between transform coefficient levels
  
  ➡ Transform coefficient levels cannot be treated separately

- **Evaluation of product space** $Q^N$ is much too complex
Rate-Distortion Optimized Quantization (RDOQ)

\[
\min_{q \in Q^N} D(q) + \lambda \cdot R(q)
\]

Reasonable assumptions

- Reconstruction vector \( t' \) lies inside associated quantization cell
- Levels with absolute value \( |t_k| \) do not require more bits than the less probable levels with an absolute value \( |t_k| + 1 \)
- Consider at most two candidate levels per transform coefficient

\[
q_{k,0} = \text{sgn}(t_k) \left\lfloor \frac{|t_k|}{\Delta_k} \right\rfloor \quad \text{and} \quad q_{k,1} = \text{sgn}(t_k) \left\lfloor \frac{|t_k|}{\Delta_k} + \frac{1}{2} \right\rfloor
\]

Rate-distortion optimized quantization

- Consider a small number of candidate levels (e.g., 1-2 per coefficient)
- Perhaps: Neglect some aspects of the entropy coding technique
- Actual algorithm depends on entropy coding
Entropic Coding Example: Run-Level Coding

**Run-Level Coding** (e.g., H.262 | MPEG-2 Video)

- Map scanned sequence of transform coefficients to (run,level) pairs
  - **run**: Number of transform coefficient levels equal to zero that precede the next non-zero transform coefficient level
  - **level**: Value of the next non-zero transform coefficient level

- Codewords are assigned to (run,level) pairs
- Code includes **end-of-block symbol (eob)**:
  - Signals that all following transform coefficient levels are equal to zero

**Example**:

- Scanned sequence of 20 transform coefficient levels
  
  5  -3  0  0  0  1  0  -1  0  0  -1  0  0  0  0  0  0  0  0

- A conversion into run-level pairs (run,level) yields

  (0,5) (0,-3) (3,1) (1,-1) (2,-1) (eob)
Example: RDOQ for Run-Level Coding

Consider sub-sequences of transform coefficient levels (in coding order)

- Distortion $D(q_k)$ for sub-sequences $q_k = (q_0, q_q, \cdots, q_k)$

$$D(q_k) = \sum_{i=0}^{k} (t_i - \Delta_i \cdot q_i)^2$$

- Number of bits $R(q_k)$ for sub-sequences $q_k = (q_0, q_q, \cdots, q_k)$
  - $q_k \neq 0 \implies$ Add up codeword lengths for (run,level) pairs
  - $q_k = 0 \implies$ Rate $R(q_k)$ depends on following levels

→ Trellis-based approach (no further simplification required)

RDOQ algorithm for run-level coding: Process coefficients in scanning order

- Consider (up to) two candidate levels for each coefficient: $q_{k,0}$ and $q_{k,1}$
- Keep (at most) one sub-sequence $q_k = (q_0, q_q, \cdots, q_k)$ with $q_k \neq 0$
- Keep (at most) $k + 1$ sub-sequences $q_k = (q_0, q_q, \cdots, q_k)$ with $q_k = 0$
  (each with a different number of zeros at the end)
- Final decision at end of block
Simple Example for $\Delta = 10$ and $\lambda = 10$

- Consider quantization of the following six transform coefficients

\[
\begin{array}{cccccc}
36 & -8 & 12 & 7 & -2 & 6 \\
\end{array}
\]

- Simple rounding (with $\Delta = 10$) yields

\[
\begin{array}{cccccc}
4 & -1 & 1 & 1 & 0 & 1 \\
\end{array}
\]

- Run-level pairs: $(0,4) (0,-1) (0,1) (0,1) (1,1) (\text{eob})$
- Codewords: $(00001100) (111) (110) (110) (0110) (10)$
- Distortion & rate: $D = 53, \ R = 23$
- Lagrangian cost: $J = D + \lambda \cdot R = 53 + 10 \cdot 23 = 283$

- RDOQ: Candidate levels

\[
\begin{array}{cccccc}
4 & -1 & 1 & 1 & 0 & 1 \\
3 & 0 & 0 & 0 & 0 \\
\end{array}
\]

- Evaluate costs in coding order
### Simple Example for $\Delta = 10$ and $\lambda = 10$

<table>
<thead>
<tr>
<th>$t_k$</th>
<th>$q_{k,i}$</th>
<th>$(q_0, \ldots, q_k)$</th>
<th>distortion $D$</th>
<th>number of bits $R$</th>
<th>$D + \lambda R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>3</td>
<td>${3}$</td>
<td>$6^2 = 36$</td>
<td>$R(0, 3) = 6$</td>
<td>96 $\implies$ discard</td>
</tr>
<tr>
<td>4</td>
<td>${4}$</td>
<td></td>
<td>$4^2 = 16$</td>
<td>$R(0, 4) = 8$</td>
<td>96</td>
</tr>
<tr>
<td>$-8$</td>
<td>0</td>
<td>${4, 0}$</td>
<td>$16 + 8^2 = 80$</td>
<td>$8 + ? = ?$</td>
<td>? [incomplete]</td>
</tr>
<tr>
<td>$-1$</td>
<td>${4, -1}$</td>
<td></td>
<td>$16 + 2^2 = 20$</td>
<td>$8 + R(0, 1) = 11$</td>
<td>130</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>${4, 0, 1}$</td>
<td>$80 + 2^2 = 84$</td>
<td>$8 + R(1, 1) = 12$</td>
<td>204 $\implies$ discard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${4, -1, 1}$</td>
<td>$20 + 2^2 = 24$</td>
<td>$11 + R(0, 1) = 14$</td>
<td>164</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>${4, -1, 1, 0}$</td>
<td>$24 + 7^2 = 73$</td>
<td>$14 + ? = ?$</td>
<td>? [incomplete]</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>${4, -1, 1, 1}$</td>
<td>$24 + 3^2 = 33$</td>
<td>$14 + R(0, 1) = 17$</td>
<td>203</td>
</tr>
<tr>
<td>$-2$</td>
<td>0</td>
<td>${4, -1, 1, 0, 0}$</td>
<td>$73 + 2^2 = 77$</td>
<td>$14 + ? = ?$</td>
<td>? [incomplete]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${4, -1, 1, 1, 0}$</td>
<td>$33 + 2^2 = 37$</td>
<td>$17 + ? = ?$</td>
<td>? [incomplete]</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>${4, -1, 1, 0, 0, 0}$</td>
<td>$77 + 6^2 = 113$</td>
<td>$14 + R(eob) = 16$</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${4, -1, 1, 1, 0, 0}$</td>
<td>$37 + 6^2 = 73$</td>
<td>$17 + R(eob) = 19$</td>
<td>263 $\implies$ choose</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>${4, -1, 1, 0, 0, 1}$</td>
<td>$77 + 4^2 = 93$</td>
<td>$14 + R(2, 1) + R(eob) = 21$</td>
<td>303</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${4, -1, 1, 1, 0, 1}$</td>
<td>$37 + 4^2 = 53$</td>
<td>$17 + R(1, 1) + R(eob) = 23$</td>
<td>283</td>
</tr>
</tbody>
</table>

---

**excerpt of H.262 | MPEG-2 Video codeword table for transform coefficient levels (s = sign)**

<table>
<thead>
<tr>
<th>run</th>
<th>level</th>
<th>codeword</th>
<th>run</th>
<th>level</th>
<th>codeword</th>
<th>run</th>
<th>level</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>±1</td>
<td>11s</td>
<td>0</td>
<td>±4</td>
<td>0000 110s</td>
<td>2</td>
<td>±1</td>
<td>0101 s</td>
</tr>
<tr>
<td>0</td>
<td>±3</td>
<td>0010 1s</td>
<td>1</td>
<td>±1</td>
<td>011s</td>
<td>eob</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
Simple Example for $\Delta = 10$ and $\lambda = 10$

- Consider quantization of the following six transform coefficients:

  \[
  \begin{array}{cccccc}
  36 & -8 & 12 & 7 & -2 & 6 \\
  \end{array}
  \]

- Simple rounding (with $\Delta = 10$) yields:

  \[
  \begin{array}{cccccc}
  4 & -1 & 1 & 1 & 0 & 1 \\
  \end{array}
  \]

  - Distortion & rate: $D = 53, \quad R = 23$
  - Lagrangian cost: $J = 283$

- Rate-distortion optimized quantization:

  \[
  \begin{array}{cccccc}
  4 & -1 & 1 & 1 & 0 & 0 \\
  \end{array}
  \]

  - Distortion & rate: $D = 73, \quad R = 19$
  - Lagrangian cost: $J = 263 \quad (< 283)$
RDOQ for HEVC Entropy Coding

Approach implemented in reference software:

1. Determine one candidate level $q_k$ for each scan index $k$ (under assumption that coefficient is actually transmitted)

$$\min \ (t_k + \Delta \cdot q_k)^2 + \lambda \cdot R_k(q_k)$$

2. Determine coded subblock flags by comparing Lagrangian costs of
   - The levels $q_k$ determined above are transmitted
   - All levels inside subblock are set equal to 0

3. Determine position of last non-zero coefficient by comparing Lagrangian costs of choosing any non-zero level as last non-zero level (in which case all following levels are set equal to 0)

4. Determine coded block flag by comparing Lagrangian costs of
   - The levels $q_k$ determined above are transmitted
   - All levels inside transform block are set equal to 0
Low-Complexity Quantization

Complexity of RDOQ
- Rather large due to consideration of dependencies in entropy coding
- Complexity reduction: Neglect dependencies

Low-Complexity Quantization: General Idea
- Neglect dependencies between transform coefficient levels
- Use simple rate models for transform coefficient levels
  ➔ Example:
  
  \[ R(q) = a + b \cdot |q| \]

- Assume that reconstructed coefficient lies inside quantization cell
  ➔ Two candidate levels

  \[ q_{k,0} = \text{sgn}(t_k) \cdot \left\lfloor \frac{|t_k|}{\Delta_k} \right\rfloor \quad \text{(rounding towards zero)} \]

  \[ q_{k,1} = q_{k,0} + \text{sgn}(t_k) \quad \text{(rounding away from zero)} \]
Without loss of generality: Consider $t_k \geq 0$

- Choose $q_k = q_{k,0}$ if and only if

$$ (t_k - \Delta \cdot q_{k,0})^2 + \lambda \cdot R(q_{k,0}) \leq (t_k - \Delta \cdot (q_{k,0} + 1))^2 + \lambda \cdot R(q_{k,0} + 1) $$

- Choose $q_k = q_{k,0}$ if and only if

$$ \frac{t_k}{\Delta} \leq d_k(q_{k,0}) = q_{k,0} + \frac{1}{2} + \frac{\lambda}{2 \Delta^2} (R(q_{k,0} + 1) - R(q_{k,0})) $$
Low-Complexity Quantization

- **Decision threshold**

\[
d_k(q_{k,0}) = q_{k,0} + \frac{1}{2} + \frac{\lambda}{2\Delta^2} (R(q_{k,0} + 1) - R(q_{k,0}))
\]

- **Example: Simple rate model** \(R(q) = a + b \cdot |q|\)

\[
d_k(q_{k,0}) = q_{k,0} + \frac{1}{2} + \frac{\lambda \cdot b}{2\Delta^2}
\]

- **Lagrange parameter** is often set according to \(\lambda = c \cdot \Delta^2\)

\[
d_k(q_{k,0}) = q_{k,0} + \frac{1}{2} + \frac{b \cdot c}{2}
\]

- **Extending considerations to negative values** yields

\[
q_k = \text{sgn}(t_k) \left\lfloor \frac{|t_k|}{\Delta_k} + f_k \right\rfloor \quad \text{with} \quad f_k = \max \left(0, \frac{1}{2} - \frac{b \cdot c}{2}\right)
\]

- **Rounding with constant offset** \(f_k\)
- **Rounding offset** \(f_k\) can be determined experimentally
Comparison of Quantization Methods

Example: H.265 | MPEG-H HEVC

- Simple rounding ($f_k = 0.5$)
- Experimentally optimized rounding offset $\Rightarrow f_k = 0.2$
- Rate-distortion optimized quantization
  - Changing of quantization offset $f_k$ yields large coding gain (ca. 16%)
  - Consideration of actual entropy coding (RDOQ) gives additional 6% gain
Selection of Lagrange Multiplier

Discussed approaches of Lagrangian optimization

- Encoder operation point is determined by
  - Quantization parameter $QP$
  - Lagrange multiplier $\lambda$
- Typically, $QP$ can be modified on a block basis
- For each $\lambda$, there is an “optimal” choice of $QP$ values

**Consequent optimization**: Choose $QP$ values as part of the encoding process

- Could be incorporated into mode decision

$$\{c_k, QP_k\}^* = \arg \min_{c \in C, \, QP \in Q} D_k(c, QP) + \lambda \cdot R_k(c, QP)$$

- Minimization over product space $C \times Q$ substantially increases complexity
- Desirable: Deterministic relationship between $\lambda$ and $QP$
Approximate Relationship between $\lambda$ and QP

High-rate approximation

- Assume: Strictly convex operational distortion rate function $D(R)$

\[
\frac{d}{dR}(D(R) + \lambda R) = 0 \quad \implies \quad \lambda = -\frac{d}{dR}D(R)
\]

- High-rate approximation of r-d function: $D(R) = a \cdot e^{-bR}$

\[
\implies \lambda = -\frac{d}{dR}D(R) = a \cdot b \cdot e^{-bR} = b \cdot D(R)
\]

- High-rate approximation of distortion: $D(\Delta) = \Delta^2/12$

\[
\implies \lambda = \text{const} \cdot \Delta^2
\]

Relationship between $\lambda$ and QP

- High-rate approximations are not completely realistic for a video codec
- Nonetheless, indicate strong dependency between $\lambda$ and QP

$$\lambda \propto \Delta^2$$ (note: QP specifies quantization step size $\Delta$)
**Experimental Investigation of \( \lambda \)-QP Relationship**

**Diagram:**
- **Kimono (1920×1080, 24 Hz)**
  - Minima are marked by circles.
  - Quantization parameter QP.
  - \( \lambda = 10, 32, 100, 320, 1000 \)
- **Entertainment-quality test sequences**
  - BasketballDrive
  - BQTerrace
  - Cactus
  - Kimono
  - ParkScene
  - \( \lambda = 0.05 \cdot 2^{\text{QP}/3} \)

**Graph:**
- **D+\( \lambda \cdot R \)** (per luma sample)
- **quantization parameter QP**
- **Lagrange multiplier \( \lambda \)**

**Experiment for IPPP coding with H.265 | MPEG-H HEVC**
- Fix \( \lambda \) and run encodings with all supported QP values.
- Choose QP that minimizes \( D + \lambda \cdot R \) for given \( \lambda \).
- Plot obtained (\( \lambda, \text{QP} \)) points into diagram (for multiple test sequences).
- Regression yields approximate relationship:
  
  \[
  \lambda = 0.05 \cdot 2^{\text{QP}/3} \quad \implies \quad \text{confirms } \lambda \propto \Delta^2 \quad \text{(note: } \Delta \propto 2^{\text{QP}/6})
  \]
Experimental Investigation of $\lambda$-QP Relationship

<table>
<thead>
<tr>
<th></th>
<th>quantization step size $\Delta$</th>
<th>Lagrange multiplier $\lambda = f(QP)$ for ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.262</td>
<td>MPEG-2 Video</td>
<td>$\Delta \propto QP$</td>
</tr>
<tr>
<td>MPEG-4 Visual</td>
<td>$\Delta \propto QP$</td>
<td>$\lambda = 0.5 \cdot QP^2$</td>
</tr>
<tr>
<td>H.263</td>
<td>$\Delta \propto QP$</td>
<td>$\lambda = 0.5 \cdot QP^2$</td>
</tr>
<tr>
<td>H.264</td>
<td>MPEG-4 AVC</td>
<td>$\Delta \propto 2^{QP/6}$</td>
</tr>
<tr>
<td>H.265</td>
<td>MPEG-H HEVC</td>
<td>$\Delta \propto 2^{QP/6}$</td>
</tr>
</tbody>
</table>

Results for different video coding standards

- Similar $\lambda$-QP relationships for other video coding standards
- Note: For H.264 | MPEG-4 AVC and H.265 | MPEG-H HEVC, a value of

$$a = 2^{QP/6-2}$$

represents approximately the same quantization step size $\Delta$ as $a = QP$ for the other considered standards
Lagrangian Encoder Control: Summary

Trade-off between coding efficiency and complexity

- Impossible to consider all dependencies between coding parameters
- Neglect impact of certain decisions on selection of other coding parameters
- Chosen degree of simplification determines trade-off between complexity and coding efficiency

Feasible encoding algorithm

- Select operation point using the quantization parameter QP
- Set Lagrange multiplier $\lambda$ according to determined relationship $\lambda = f(QP)$
- Lagrangian motion estimation [will be discussed later]
- Rate-distortion optimized quantization
- Lagrangian decision between coding modes

This algorithm will be used in all following experiments
Final Remarks

Lagrangian Encoder Control: Coding Efficiency

Experimental results for IPPP coding with H.262 | MPEG-2 Video

- Started with Test Model 5 (TM5) and successively enabled Lagrangian approaches for mode decision, motion estimation, and quantization
- Additional test “Exhaustive optimization”
  - Transform coding with RDOQ for all possible motion vectors
  - Increases encoder run time by more than a factor of 1000
Part Summary

General Encoding Problem
- Minimize distortion while not exceeding given bit rate (or vice versa)

Lagrangian Optimization
- Formulate constrained problem as unconstrained problem \((D + \lambda \cdot R)\)
- Solutions of unconstrained problem are also solutions of original problem
- For independent sets and additive distortion measures:
  Global optimum is found by separate minimizations

Lagrangian Encoder Control
- Partly neglect dependencies between blocks (ignore impact on future)
- Partly neglect dependencies between coding parameters for a block
- Split decision for a block into
  - Mode decision (includes transform coding)
  - Motion estimation (assumes zero residual signal) [discussed later]
  - Quantization (considers actual entropy coding)