Video Encoder Control
Outline

- Introduction
- Encoder Control using Lagrange multipliers
  - Lagrangian optimization
  - Lagrangian bit allocation
- Lagrangian Optimization in Hybrid Video Encoders
  - Mode decision
  - Motion estimation
  - Quantization
  - Selection of Lagrange multiplier
  - Summary of Encoding Algorithm
  - Coding Efficiency
- Summary
**Encoding Problem**

**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
- Actual coding efficiency for 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
Encoding Problem

Encoding problem for given input video $s_v$

- Generate a conforming bitstream $b \in \mathcal{B}$ 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 $R_B$

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

Equivalent problem

- Generate a conforming bitstream $b \in \mathcal{B}$ 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 $D_{\text{max}}$

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

$\implies$ Impossible to find optimal solution (extremely large parameter space)

$\implies$ Split into smaller sub-problems
Lagrangian Optimization

Constrained Optimization Problem

- Consider set of samples $s$ (block, picture, or set of pictures)
- Vector of coding parameters $p \in \mathcal{P}$
- Constrained problem for given rate budget $R_B$, with $D(p) = D(s, s'(p))$

$$\min_{p \in \mathcal{P}} D(p) \text{ subject to } R(p) \leq R_B$$

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

Unconstrained Optimization Problem

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

$$p^*_\lambda = \arg \min_{p \in \mathcal{P}} D(p) + \lambda \cdot R(p)$$

- Cannot find all optimal solutions $\{p_{opt}\}$
- But: Each solution $p^*_\lambda$ is an optimal solution, $\{p^*_\lambda\} \subseteq \{p_{opt}\}$
Optimality of Lagrangian Approach

Each solution of unconstrained problem is also a solution of original problem

- Consider solution \( p_\lambda^* \) for a particular value of \( \lambda \), with \( \lambda \geq 0 \)
- By definition, we have
  \[
  \forall p \in \mathcal{P}, \quad D(p) + \lambda \cdot R(p) \geq D(p_\lambda^*) + \lambda \cdot R(p_\lambda^*) \\
  D(p) - D(p_\lambda^*) \geq \lambda \cdot (R(p_\lambda^*) - R(p))
  \]

- 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

  \[
  p_\lambda^* = \arg \min_{p \in \mathcal{P}} D(p) \quad \text{subject to} \quad R(p) \leq R_B = R(p_\lambda^*)
  \]
Lagrangian Optimization — Illustration

Solutions of unconstrained optimization problem

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

\[ D = -\lambda R \]

\[ D + \lambda R \]

\[ D \mapsto D + \lambda R \]

Convex hull tangent with slope $-\lambda$

Convex hull tangent parallel to R-axis
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

$$\{p_0^*, p_1^*, \ldots\} = \arg \min_{p_0 \in P_0, p_1 \in P_1, \ldots} \sum_k D_k(p_k) + \lambda \sum_k R_k(p_k)$$

- Can be solved by separate minimizations

$$\forall k, \quad p_k^* = \min_{p_k \in P_k} D_k(p_k) + \lambda R_k(p_k)$$

Key Advantage of Lagrangian Optimization
- **Global optimum problem can be solved by separate minimizations**
- Yields optimal bit allocation $\{R_0, R_1, \ldots\}$
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

- Intra-picture prediction
- Motion-compensated prediction
- Motion vector prediction
- Conditional entropy coding

Concept of Lagrangian optimization is still applicable

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

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

\[\Rightarrow\] Past decisions \(\{p_{k-1}, p_{k-2}, \cdots\}\) are taken into account
(by using correct predictors and conditional entropy codes)

\[\Rightarrow\] 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
  - $\rightarrow$ Mode decision
  - $\rightarrow$ Motion estimation
  - $\rightarrow$ 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

Alternative measure for motion estimation (sub-sample vectors):
- $\rightarrow$ SAD after Hadamard transform
Lagrangian Mode Decision

Consider small set $C_k$ of coding modes for block $s_k$

- Example: $C_k = \{\text{Intra, Inter, Skip, Split}\}$
- Associated parameters are determined in advance
  $\Rightarrow$ Motion vectors, intra prediction modes, transform coefficient levels, ...
- Each coding mode $c \in C_k$ is associated with coding parameters $p_k(c)$

$\Rightarrow$ Small subset $P_{C_k} = \{p_k(c) \mid c \in C_k\}$ of the parameter space $P_k$

Lagrangian mode decision

- Subset $P_{C_k}$ 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 $p_k(c)$

- Note: Requires complete transform coding for each tested coding mode
Lagrangian Mode Decision — Example

Example: H.262 | MPEG-2 Video, IPPP coding structure

- Compared with Test Model 5 (TM5): Reference encoder implementation
- Consider 4 coding modes:

\[ C_k = \{ \text{Intra, Inter, NoCoeff, ZeroMv} \} \]

- Average bit-rate savings relative to TM5 (for test set): 8.5%
Lagrangian Mode Decision in Modern Video Encoders

Lagrangian mode decision is typically used for the following decisions:

- Decision between intra and inter coding modes
- Determination of intra prediction modes
- Decision whether a block is subdivided into smaller blocks
- Selection of transform size or subdivisions for transform coding

Well-suited for determining tree-based partitionings

- For each internal node:
  - Decide whether block is split
  - Evaluate blocks in depth-first order
  - Correct predictors from neighboring blocks
- Example: Two quadtree levels
  1. Decide splitting for A, B, C, D
  2. Decide splitting for entire block
- Can be combined with fast pruning strategies
Motion Estimation — Block Matching Algorithm

The measurement window is compared with different shifted blocks in the reference frame and the best match is determined.

The considered block of samples in the current frame is selected as a measurement window.
Lagrangian Motion Estimation

Lagrangian Cost Measure

- Mode decision concept (with transform coding of residual) is too complex for hundreds or thousands of motion vector candidates

⇒ Assume that reconstructed prediction error signal is equal to zero

⇒ Select motion vector \( m \) in search range \( M \) according to

\[
    m_k^* = \arg \min_{m \in M_k} D_k(m) + \lambda_M R_k(m)
\]

with

- \( D_k(m) \) – Distortion between original block \( s_k \) and prediction signal \( \hat{s}_k \)
- \( R_k(m) \) – Number of bits for transmitting the motion vector \( m \)
- \( \lambda_M \) – Lagrange multiplier (depends on chosen distortion measure)

Distortion measure

- SSD (\( \lambda_M = \lambda \)) or SAD (\( \lambda_M = \sqrt{\lambda} \)) ⇒ SAD is often faster to compute
- SAD after Hadamard transform (better approximation of real RD costs)
Lagrangian Motion Estimation — Cost Measure

Compare exhaustive Lagrangian motion search with TM5 approach

- Full transform coding (similar to mode decision) → very complex
- SSD between original and prediction signal
- SAD between original and prediction signal
- Hadamard SAD between original and prediction signal
Search Strategy

Sub-sample accurate motion vectors
- Sub-sample locations: Interpolation
- Split search into integer-sample search and sub-sample refinement(s)
- Use simpler distortion measure for integer-sample search

Fast integer-sample search
- Reduce number of tested candidates
- Multiple strategies suggested
- Examples:
  - Three-step search
  - Logarithmic search
  - Conjugate directional search
  - Enhanced predictive zonal search
  - ...

• integer-sample positions
• half-sample positions
• quarter-sample positions
Fast Search Strategies: Logarithmic Search

Logarithmic search [Jain, Jain, 1981]

- Iterative comparison of the cost measures at 5 points (corners and center) of a diamond-shaped pattern
- Move pattern so that pattern in centered around best match
  - No more than 3 new candidates
- Logarithmic refinement of search pattern (4 new candidates) if
  - Best match is in center of pattern
  - Or best match is at the border of the search range
- Motion search is terminated if
  - Best match is in center of pattern
  - And smallest pattern size is used
Fast Search Strategies: Diamond Search

Diamond search [Li, Zeng, Liou, 1994] and [Zhu, Ma, 1997]

- Iterative search with 9 points of a diamond pattern
- Similar search strategy as logarithmic search

Start with large diamond pattern at motion vector (0,0) or at a predicted vector

If best match is in the center of a large diamond, proceed with a smaller diamond

If best match does not lie in the center of the diamond pattern, center next diamond pattern at the best match
Fast Search Strategies: Choosing of Start Point

Non-adaptive choices of start point

- Use motion vector (0,0) as start point of motion search
  - Suitable for applications like video conferencing
  - Problematic if large motions occur in video sequence
- Use motion vector predictor as start point for motion search
  - Typically results in faster termination of motion search

Adaptive choice of start point

- General idea: Motion of a block is similar to at least one of the neighboring blocks
- First evaluate the motion vectors of the already estimated neighboring blocks
  - Example: Blocks A, B, C and D
  - Candidates can also include a temporally predicted motion vector
- Choose best match among the candidates as start point of the motion search
**Motion Estimation — Search Strategy**

**TM5 approach:**
- Integer search with SAD
- Sub-sample refinement with SAD
- Cost measure: Distortion (without a rate term)

Compare different search strategies with TM5 approach
- Exhaustive search (all sub-sample locations) with Hadamard SAD
- Exh. integer search + sub-sample refinement (both with Hadamard SAD)
- Exh. integer search with SAD + sub-sample refinement with Hadamard SAD
- Exh. integer search with SAD + sub-sample refinement with SAD
- Fast integer search with SAD + sub-sample refinement with Hadamard SAD
Motion Estimation — Summary

Require significant complexity reduction compared to mode decision
- Neglect dependencies between motion vectors and transform coefficient levels
  ⇒ Assume residual signal equal to zero during motion estimation
- Split motion search into integer-sample search and sub-sample refinement
- Apply fast search strategies

Configuration with reasonable trade-off between coding efficiency and complexity
- Fast integer-sample search with Lagrangian cost
  - Distortion: SAD between original and prediction signal
  - Rate: Number of bits required for transmitting motion vectors
  - Fast search strategy: HM approach (combination of different concepts)
- Sub-sample refinement with Lagrangian cost
  - Distortion: Hadamard SAD between original and prediction signal
  - Rate: Number of bits required for transmitting motion vectors
Reference Picture Selection

Typical approach

- Determine motion vector $m_r$ for each considered reference picture $r \in \mathcal{R}$
- Select motion parameters $\{r, m_r\}$ among the pre-determined sets

Criterion for reference picture selection

- Lagrangian decision similar to motion search
- Choose motion parameters $\{r, m_r\}$ according to

$$r^\star = \arg \min_{r \in \mathcal{R}} D_k(r, m_r) + \lambda_M R_k(r, m_r)$$

with

- $D_k(r, m_r)$ – Distortion between original block $s$ and prediction signal $\hat{s}$
- $R_k(r, m_r)$ – Number of bits for reference index $r$ and motion vector $m_r$

⇒ Usually: Same distortion measure as for sub-sample search
Quantization

Video Codecs: Transform coding with orthogonal block transforms

- Inverse transform and forward transform are given by
  \[ s' = B t' \quad \text{and} \quad t = B^{-1} s = B^T s \]
  with \( B \) being the inverse transform matrix

- SSD distortion in sample space = SSD distortion in transform domain
  \[ D = (s - s')^T(s - s') = (t - t')^T(B^T B)(t - t') \]
  \[ = (t - t')^T(t - t') \]

Modern Video Codecs: Uniform reconstruction quantizers (URQs)

- Inverse quantizer mapping
  \[ t'_k = \Delta_k \cdot q_k \]

- Distortion for vector \( q \) of quantization indexes is given by
  \[ D(q) = \sum_{k=0}^{N-1} D_k(q_k) = \sum_{k=0}^{N-1} (t_k - \Delta_k q_k)^2 \]
Quantization — Lagrangian Optimization

Simple: Minimize SSD distortion for given quantization step sizes $\Delta_k$

- Orthogonal transforms: Independent treatment of transform coefficients

$$D(q) = \sum_{k=0}^{N-1} D_k(q_k) = \sum_{k=0}^{N-1} (t_k - \Delta_k q_k)^2$$

- SSD distortion is minimized by simple rounding according to

$$q_k = \text{sgn}(t_k) \left\lfloor \frac{|t_k|}{\Delta_k} + \frac{1}{2} \right\rfloor$$

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

$\Rightarrow$ Transform coefficient levels cannot be treated separately
Rate-Distortion Optimized Quantization (RDOQ)

Problem: Evaluation of product space $Q^N$ is much too complex

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

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
Entropy 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 an additional *end-of-block* symbol (eob)
  
  - Signals that all following transform coefficient levels are equal to zero
- Run-level coding is a practical example of a V2V code

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

- 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 $q_k = (q_0, q_1, \cdots, q_k)$

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

- Number of bits $R(q_k)$ for sub-sequences $q_k = (q_0, q_1, \cdots, q_k)$
  - $q_k \neq 0 \implies$ Add up codeword lengths for (run,level) pairs
  - $q_k = 0 \implies$ Rate $R(q_k)$ depends on following levels

$\implies$ 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 $q_k = (q_0, q_1, \cdots, q_k)$ with $q_k \neq 0$
- Keep (at most) $k + 1$ sub-sequences $q_k = (q_0, q_1, \cdots, q_k)$ with $q_k = 0$
  (each with a different number of zeros at the end)
- Final decision at end of block
**Simple Example for $\Delta_k = 10$ and $\lambda = 10$**

<table>
<thead>
<tr>
<th>$t_k$</th>
<th>$q_{k,i}$</th>
<th>$(q_0, \cdots, q_k)$</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>$4^2 = 16$</td>
<td></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>0</td>
<td>${4,-1}$</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>${4,-1,1}$</td>
<td>$20 + 2^2 = 24$</td>
<td></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>${4,-1,1,1,1}$</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>$14 + R(\text{eob}) = 16$</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>$17 + R(\text{eob}) = 19$</td>
<td>263 $\implies$ choose</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>${4,-1,1,0,0,1}$</td>
<td>$77 + 4^2 = 93$</td>
<td>$14 + R(2,1) + R(\text{eob}) = 21$</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>$17 + R(1,1) + R(\text{eob}) = 23$</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>0101s</td>
</tr>
<tr>
<td>0</td>
<td>±3</td>
<td>0010 1s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>011s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>eob 10</td>
</tr>
</tbody>
</table>
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
\[
q_{k,0} = \text{sgn}(t_k) \cdot \left\lfloor t_k / \Delta_k \right\rfloor \quad \text{(rounding towards zero)}
\]

and
\[
q_{k,1} = q_{k,0} + \text{sgn}(t_k) \quad \text{(rounding away from zero)}
\]
Low-Complexity Quantization

Without loss of generality: Consider $t_k \geq 0$

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

\[(t_k - \Delta_k \cdot q_{k,0})^2 + \lambda \cdot R(q_{k,0}) \leq (t_k - \Delta_k \cdot (q_{k,0} + 1))^2 + \lambda \cdot R(q_{k,0} + 1)\]

which yields

\[q_k = q_{k,0} \iff \frac{t_k}{\Delta_k} \leq d_k(q_{k,0})\]

with decision threshold

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

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

- Example: Simple rate model \( R(q) = a + b \cdot |q| \)
  \[ d_k(q_{k,0}) = q_{k,0} + \frac{1}{2} + \frac{\lambda \cdot b}{2\Delta_k^2} \]

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

⇒ Rounding with constant offset \( f_k \)
⇒ Rounding offset \( f_k \) can be determined experimentally
Example: H.265 | MPEG-H HEVC

- Simple rounding ($f_k = 0.5$)
- Experimentally optimized rounding offset $\implies f_k = 0.2$
- Rate-distortion optimized quantization
  $\implies$ Changing of quantization offset $f_k$ yields large coding gain (ca. 16%)
  $\implies$ 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 $\lambda$
- Lagrange multiplier $\lambda_M$ is not considered as additional degree of freedom
  \[ \implies \text{We always choose } \lambda_M = \lambda \text{ (for SSD) or } \lambda_M = \sqrt{\lambda} \text{ (for SAD)} \]
- 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

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

\[ \implies \text{Minimization over product space } C \times Q \text{ substantially increases complexity} \]
\[ \implies \text{Desirable: Deterministic relationship between } \lambda \text{ and } QP \]
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: $D(R) = a \cdot e^{-bR}$

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

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

$$\implies \quad \lambda = \text{const} \cdot \Delta^2$$

Relationship between $\lambda$ and QP

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

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

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 $(\lambda, QP)$ points into diagram (for multiple test sequences)
- Regression yields approximate relationship

$$\lambda = 0.05 \cdot 2^{QP/3} \quad \Rightarrow \quad \text{confirms } \lambda \propto \Delta^2 \quad \text{(note: } \Delta \propto 2^{QP/6})$$
## 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></td>
<td>intra pictures</td>
<td>inter pictures</td>
</tr>
<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 consisting of
  - Fast integer-sample search using SAD as distortion measure
  - Sub-sample refinement using Hadamard SAD as distortion measure
- Rate-distortion optimized quantization
- Lagrangian decision between coding modes

$\implies$ This algorithm will be used in all following experiments
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
Summary

Encoding problem

- Minimize distortion while not exceeding given bit rate
- Minimize bit rate while not exceeding given distortion

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)
  - Quantization (considers actual entropy coding)