I. Context & Contribution

- MPEG Intra Coding generates dependencies between Coding Units (CU): Closed-Loop prediction and CABAC
- Dual Joint Rate-Distortion Optimization (Dual-JRDO) is an exhaustive search for prediction parameters which minimizes a cost function affecting two neighbor CUs
- Contribution: A low complexity version is proposed in order to estimate opportunities for real-time encoding

II. Dual-JRDO Model

- **Notations**
  - Coding parameters: \( \tilde{p} \)
  - Index of CU: \( i \)
  - Distortion \( D \) and Rate \( R \)
  - R-D Cost Function: \( J(\tilde{p}) \)

- **Dual-JRDO equation**
  - Exhaustive optimization of CUs 2 by 2 (dotted areas)
  - Parameter to optimize \( \tilde{p} \) is the prediction mode
  \[
  \{ p_i^*, p_{i+1}^* \} = \arg \min_{\{ p_i, p_{i+1} \}} \left( J(p_i) + J_{i+1}(p_i, p_{i+1}) \right)
  \]

III. Acceleration Methods

1. Adapting to Spatial Activity

- **Down-sample** 16x16 CUs into 4x4 CUs
  - Pixel real position: \( m, n \)
  - Pixel down-sampled position: \( p, q \)
  - Pixel's luminance: \( l \)
  \[
  l(p, q) = \frac{1}{16} \sum_{m=1}^{4} \sum_{n=1}^{4} l(p/4 + m, q/4 + n)
  \]
- **Compute the spatial activity**
  - Activity of CU \( i \) to \( g_i \)
  \[
  g_i = \frac{1}{16} \sum_{p=1}^{4} \sum_{q=1}^{4} \min \left\{ |l_i(p, q) - l_i(p-1, q)|, |l_i(p, q) - l_i(p, q-1)| \right\}
  \]
  - If \( g_i \) lower than a threshold \( Th \)
    - Do not apply Dual-JRDO
- **Th is dependent of quantizer \( QP \)**
  \( Th(QP) = \alpha + e^{\beta QP} \)

2. Short-listing of the depending CU

- **Observations**
  - Dual-JRDO sequentially optimized two prediction modes \( \{ p_i, p_{i+1} \} \)
  - During \( p_{i+1} \) optimization, \( p_i \) is fixed
  - The second mode analysis is an independent optimization
- Any fast solution which reduce the number of modes can be applied
  - We choose one of the most efficient: Rough Mode Decision (RMD)

- **RMD**
  - Estimate the Most Probable Modes (MPMs)
  - Estimate the modes with lowest SATD score
  - Create a shortlist based on this two sets
  - Apply RDO only on this shortlist

3. Residual Analysis based Clustering

- **Focusing on Distortion dependency**
  - Identical residue leads to identical reconstructed CU
  - If two modes results into the same decoded CU, the next CUs are impacted the same manner

   \[
   J_i(p_i) + J_{i+1}(p_i, p_{i+1})
   \]

   **Solution based on modes clustering**

   I. During RMD process for \( p_i \)
   - Cluster all modes based on their residue
   II. If \( p_i \) is the first analyzed mode of its cluster
   - Consider all possible modes for \( p_{i+1} \)
   III. Otherwise
   - Consider \( p_{i+1}^* \) attached to this cluster
   - Consider the new MPMs if it applies

IV. Performances

- **Test environment**
  - HM16.12 Anchor
  - RDO configuration
  - Common Test Conditions: All-Intra
  - PSNR based Bjontegaard computation
  - 5 QPs (22,27,32,37,42)

- **Configurations of Dual-JRDO**
  - \( C_i \): No acceleration
  - \( C_j \): Adapting to spatial activity
  - \( C_k \): Short-listing of \( p_i \)
  - \( C_l \): Residue clustering of \( p_i \)
  - \( C_m \): All accelerations combined

- **Average BD-BR**

<table>
<thead>
<tr>
<th>Class</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( C_4 )</th>
<th>( C_5 )</th>
<th>( C_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>-0.45%</td>
<td>-0.42%</td>
<td>-0.38%</td>
<td>-0.46%</td>
<td>-0.35%</td>
</tr>
<tr>
<td>Class B</td>
<td>-0.61%</td>
<td>-0.54%</td>
<td>-0.47%</td>
<td>-0.61%</td>
<td>-0.42%</td>
</tr>
<tr>
<td>Class C</td>
<td>-0.63%</td>
<td>-0.59%</td>
<td>-0.46%</td>
<td>-0.64%</td>
<td>-0.44%</td>
</tr>
<tr>
<td>Class D</td>
<td>-0.64%</td>
<td>-0.58%</td>
<td>-0.52%</td>
<td>-0.64%</td>
<td>-0.47%</td>
</tr>
<tr>
<td>Class E</td>
<td>-0.87%</td>
<td>-0.76%</td>
<td>-0.67%</td>
<td>-0.88%</td>
<td>-0.60%</td>
</tr>
<tr>
<td>All</td>
<td>-0.63%</td>
<td>-0.57%</td>
<td>-0.49%</td>
<td>-0.63%</td>
<td>-0.48%</td>
</tr>
</tbody>
</table>

   - **Best Sequence**
     - -1.12%  
     - -1.01%  
     - -0.87%  
     - -1.11%  
     - -0.82%  

   - **Worst Sequence**
     - -0.19%  
     - -0.21%  
     - -0.20%  
     - -0.20%  
     - -0.20%  

- **Complexity (%)**

<table>
<thead>
<tr>
<th>Class</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( C_4 )</th>
<th>( C_5 )</th>
<th>( C_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>113%</td>
<td>411%</td>
<td>199%</td>
<td>699%</td>
<td>133%</td>
</tr>
<tr>
<td>Class B</td>
<td>1025%</td>
<td>540%</td>
<td>192%</td>
<td>702%</td>
<td>156%</td>
</tr>
<tr>
<td>Class C</td>
<td>945%</td>
<td>565%</td>
<td>197%</td>
<td>703%</td>
<td>160%</td>
</tr>
<tr>
<td>Class E</td>
<td>1091%</td>
<td>336%</td>
<td>199%</td>
<td>592%</td>
<td>116%</td>
</tr>
<tr>
<td>Class F</td>
<td>1099%</td>
<td>401%</td>
<td>193%</td>
<td>607%</td>
<td>124%</td>
</tr>
<tr>
<td>All</td>
<td>1062%</td>
<td>454%</td>
<td>198%</td>
<td>666%</td>
<td>138%</td>
</tr>
</tbody>
</table>

   - **Best Sequence**
     - 920%  
     - 198%  
     - 185%  
     - 467%  
     - 74%  

   - **Worst Sequence**
     - 1266%  
     - 662%  
     - 212%  
     - 735%  
     - 178%  

V. Conclusion

- **Observations**
  - Dual-JRDO is most effective on textured areas
  - Methods reducing the number of modes to test (as RMD) are efficient when extended to dependent CUs
  - Many predictions lead to same residual data and create redundant computations in dependent schemes

- **Conclusion**
  - Dual-JRDO can be highly speed up and be 5x faster
  - Even faster implementation can bring constant BD-impvement (-0.45%)

- **Future Work**
  - Use Dual-JRDO to improve other coding parameters: Quantization, Transform, Filters, ...