Optimizing the Deblocking Algorithm for H.264 Decoder Implementation

By Ken Kin-Hung Lam
klam@mail.ece.utexas.edu
What is “deblocking” again?

• Remove the discontinuity across block boundaries resulted from block-based DCT and coarse quantization steps in low bit rate encoding.

[List, Joch, Lainema, Bjøntegaard and Karczewicz, 2003]
Adaptive Deblocking Filter (Revisit)
[List, Joch, Lainema, Bjøntegaard and Karczewicz, 2003]

• An edge is associated to an ‘edge strength’ computed with a set of conditions on the data points normal to the edge. On each edges, there are 16 set of normals.

• Filters with different strength are applied to edge of different strength.

• The algorithm is adaptive because the conditions for edge strength computation are based on the quantization parameters included in the relevant blocks.
Where is the Bottleneck?

- JM 9.3 - H.264 reference codec is instrumented to gather statistics on the operations performed by the deblocking filter.

<table>
<thead>
<tr>
<th>No. of operations on edge strength computation</th>
<th>30966524</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of operations on edge filtering</td>
<td>503132</td>
</tr>
</tbody>
</table>

- Edge strength computation weights more than 90% of the total number of operations. Obviously, we should focus on optimizing the edge strength computation algorithm.
Optimizations

• Adjacent *normals* on an edge may result into similar edge strengths.

• Skipping some of the edge strength computations by taking the results from adjacent strength computations.

• Basically a ‘subsampling’ of edge strength computations, and ‘upsample’ the results by speculating the skipped strength computations with the adjacent edge strength computation results.
## Some Results

<table>
<thead>
<tr>
<th></th>
<th>Original Algorithm</th>
<th>Optimized Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of operations on edge strength computation</td>
<td>30966524</td>
<td>15483262</td>
</tr>
<tr>
<td>No. of operations on edge filtering</td>
<td>503132</td>
<td>503388</td>
</tr>
<tr>
<td>Total no. of operations</td>
<td>31469656</td>
<td>15986650</td>
</tr>
</tbody>
</table>
Some Results (SNR)
Some Results (UQI)

![Graph showing some results (UQI) with data points and line segments indicating the UQI (16x16) for different frames and quality factors. The graph includes lines and markers for different conditions: foreman_qcif-8192-nofilt-muqi, foreman_qcif-8192-filt-muqi, and foreman_qcif-8192-filt-opt2. The x-axis represents the frame numbers from 40 to 58, and the y-axis represents the UQI values from 0.7 to 0.84. The graph shows variations in UQI values across different frames and conditions.](image-url)
Conclusions

• Operations on edge strength computation is reduced by half approximately. Operations on edge filtering remains more or less the same as the original algorithm. The overall saving in total number of operations is close to 50%.

• SNR(Y) and UQI achieved by the optimized algorithm closely approximate those achieved by the original algorithm.

• More savings may be possible with even more aggressive approach.