Time Domain Lapped Transforms for Video Coding

draft-egge-netvc-tdlt-00

Nathan Egge

IETF 93 – Prague
2015 July 22
Lapped Transforms

- Originally proposed for video in 1989 by Malvar [1].
- \( n \)-point prefilter applied along block boundaries
  - Removes spatial correlation between blocks
  - Improves coding performance of \( n \)-point DCT
- Decoder applies \( n \)-point postfilter (exact inverse)
  - Quantization error spread over adjacent blocks

Lapped Transforms

- Prefilter makes the image “blocky”

- Postfilter “smoothes” blocking artifacts
Lapped Transforms

• Pros:
  - Larger spatial support means higher compression performance (improved coding gain)
  - Non-adaptive, in-loop postfilter

<table>
<thead>
<tr>
<th>subset-1</th>
<th>4x4</th>
<th>8x8</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLT</td>
<td>12.47 dB</td>
<td>13.62 dB</td>
<td>14.12 dB</td>
</tr>
<tr>
<td>DCT</td>
<td>12.42 dB</td>
<td>13.55 dB</td>
<td>14.05 dB</td>
</tr>
</tbody>
</table>

• Cons:
  - Increased ringing on edges
  - Proven coding techniques no longer work: spatial intra-prediction, intra blocks in inter frames, etc.
Lapped Transforms

- Sizes: 4x4, 8x8, 16x16 and 32x32 (64x64 in progress)
- Lapping
  - Luma blocks larger than 4x4 use 8-point lapping on all edges
  - When splitting an 8x8 down to 4x4:
    - 8-point lapping applied to “exterior” (8x8) edges
    - 4-point lapping applied to “interior” edges
  - 4:2:0 chroma uses 4-point lapping on all edges
- Lapping size does not depend on neighbors’ block size
  - Allows for efficient (exhaustive) block size decision
Filter Order

- Filter top/bottom superbloc edges
Filter Order

- Filter left/right superbloc edges
Filter Order

- Splitting: Filter interior edges
Filter Order

- Splitting: Filter interior edges
Lapped Transform Properties

- **Reversible**
  - $iLT(fLT(x)) == x$ for all $x$

- **Biorthogonal (not orthogonal)**
  - Not all basis functions have the same magnitude

- **Dynamic range expansion**
  - Core DCT is orthonormal (minimum possible)
  - Pre/post-filters add a few more bits

- **Pre-scaling**
  - Lossy input scaled by 16 to reduce impact of rounding
  - 16x16 and above no longer fit in 16 bits
Questions?