Perceptually-Driven Video Coding with the Daala Video Codec

Timothy B. Terriberry
Summary

- Daala is an attempt to completely avoid royalty-bearing technologies
- Used many unconventional tools
- Some worked well, others more challenging
  - We think the challenges are more interesting
- Many lessons learned that can inform AV1 development
  - Only a few presented here, see paper for more
Challenge 1: Lapped Transforms with Variable Block Sizes
Original Lapping Strategy

- Filter size chosen based on size of smallest block on an edge (to prevent overlap)
- Filter order chosen to mimic a loop filter’s
  - Horizontal edges first
Original Lapping Strategy

- Filter size chosen based on size of smallest block on an edge (to prevent overlap)
- Filter order chosen to mimic a loop filter’s
  - Then vertical
  - Maximal parallelism, minimum buffering
Problem #1: Basis Weirdness
Problem #2: Block size decision

• Have to know neighbors’ block sizes to compute lapping size

• Used a heuristic based on the estimated visibility of ringing to pick block sizes up front
  – Worked “okay” for still images (at least not obviously broken)
  – Was not making good decisions for inter frames

• Wanted to try explicit block size RDO (like other encoders)...
  – But lapping dependency makes this infeasible
“Fixed Lapping”: Remove the Dependency

- Always use 8-point lapping (4 pixels on either side of an edge)
  - Except on 4×4 blocks (details in a few slides)
  - Always use 4-point lapping for chroma (because of subsampling)
New Filter Order

- Filter top/bottom superblock (64×64) edges first
New Filter Order

- Filter left/right superbloc (64×64) edges next
New Filter Order

• Splitting: Filter interior edges
New Filter Order

- Splitting: Filter interior edges

- 4×4 blocks:
  - Exterior edges use 8-point filter (from previous levels)
  - Interior edges use 4-point filter (overlaps 8-point filter)
Results

- Big boost in metrics
  - Almost all from decision
  - Used fixed lapping decision with old lapping scheme and got *almost* all of the gains

- Smaller lapping means less ringing but more blockiness (especially on gradients)
  - Didn’t save much on ringing: $4 \times 4$ blocks have 12-pixel support instead of 8
  - Eventually dropped to 4-point lapping everywhere

<table>
<thead>
<tr>
<th>Rate (%)</th>
<th>DSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>-10.36612</td>
</tr>
<tr>
<td>PSNRHVS</td>
<td>-4.48956</td>
</tr>
<tr>
<td>SSIM</td>
<td>-12.32547</td>
</tr>
<tr>
<td>FASTSSIM</td>
<td>-5.20467</td>
</tr>
</tbody>
</table>
Challenge 2: Frequency Domain Intra Prediction
Frequency Domain Intra Prediction

- Perform prediction in transform domain
  - Shorter pipeline dependency for hardware
- Multiple (linear) prediction matrices trained from large dataset (approx. equiv. to spatial directions)
- Computational complexity controlled by enforcing “sparsity” (4 muls per output coefficient)

<table>
<thead>
<tr>
<th>Transform size</th>
<th>DCT+VP8 Intra</th>
<th>LT+FDIP (full)</th>
<th>LT+FDIP (sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C_g$</td>
<td>$P_g$</td>
<td>Total</td>
</tr>
<tr>
<td>$4 \times 4$</td>
<td>13.8511</td>
<td>2.9154</td>
<td>16.7665</td>
</tr>
<tr>
<td>$8 \times 8$</td>
<td>15.1202</td>
<td>0.86940</td>
<td>15.9894</td>
</tr>
<tr>
<td>$16 \times 16$</td>
<td>15.5870</td>
<td>0.19483</td>
<td>15.7818</td>
</tr>
</tbody>
</table>
Frequency Domain Intra Prediction

- Variable block sizes make this worse
  - Best results: convert all neighbors to 4×4 with “TF”
- Most multiplies spent on predicting DC
- A simpler approach:
  - Haar DC: combine DCs from smaller blocks with Haar transform (down to one DC per 64x64 block)
    - Hugely effective, no multiplies
  - Use first row/column of neighbors’ coefficients as sole AC predictor (only when block sizes match)
    - Works just as well as orig. FDIP (not very), much simpler
Things We Did Not Try

- Spatial prediction from outside lapping region
  - Very complicated with original lapping scheme
  - Feasible with fixed lapping scheme

- Correcting for biorthogonal basis function scales
  - Intractable with original lapping

- “Smart” factorization of prediction matrices
  - Only improves up to the limit of non-sparse predictors
Directions for AV1

- Directional Deringing
  - Fully SIMDable, good perceptual improvements

- Non-binary Arithmetic Coding
  - Small effective parallelism in entropy coding

- Perceptual Vector Quantization
  - Already showing small gains vs. scalar on PSNR
  - Potential for large perceptual improvements
  - Enables freq. Domain Chroma-from-Luma, others

- Rate control improvements
Daala Progress (Fast MS-SSIM): January 2014 to April 2016

The Xiph.Org Foundation & The Mozilla Corporation

Graph showing the progress of Daala, with timelines from January 2014 to April 2016. The graph compares HQ YouTube, LQ Video Conference, and H.265 with different markers indicating improvements and progress.
Daala Progress (PSNR-HVS): January 2014 to April 2016

The graph shows the progress of Daala from January 2014 to April 2016 in terms of PSNR-HVS. The x-axis represents the rate (bits/pixel) while the y-axis represents the PSNR (dB). The graph indicates that higher values on the y-axis correspond to better compression quality. The key points highlighted are:

- **HQ YouTube**
- **LQ Video Conference**
- **H.265**
- **Jan**, **May**, **Jun**, **Nov**, **Apr**

The graph also highlights that generally, a movement up and to the right signifies better performance in terms of PSNR-HVS.
Questions?