Frame Context Selection

Thomas Daede
tdaede@mozilla.com
Requirements for Temporal Scalability

- It should be possible to determine and control which previously coded frames are dependencies of the current frame
  - E.g., allow skipping every other frame
Requirements for Error Resilience

- It should be possible to determine if the decoder is missing a frame required for decoding
  - This allows retransmission or dropping of the frame
  - Allows a decoder that never shows a broken frame
VP9

- Reference frame dependencies
  - Implicitly or explicitly signaled with picture IDs in RTP mapping
  - Up to three allowed per frame (from pool of 8)

- Frame contexts (probabilities)
  - Stores probabilities that are backwards-adapted based on data from previous frames
  - Decoder maintains four independent sets
  - Each frame signals which one to use
    - Optionally writes back to the slot it read from
  - Choice uncorrelated with references or picture IDs
Problems with Frame Contexts

• If you lose a frame, you don’t know which slot it updated
  – You no longer know if you can decode any frame

• The last frame to update the slot you’re using might not be one of your reference frames
  – Frame contexts introduce potential hidden fourth frame dependency (this is surprising)
  – RTP mapping only signals three picture IDs (could fix)

• You can’t fork probabilities and evolve them independently
  – Every layer pays cost of re-learning probabilities
AV1

- Reference frame dependencies
  - Explicitly signaled and coded with frame IDs in codec payload
  - Up to six allowed per frame (from pool of 8)
- Probabilities
  - Currently same as VP9, but expanded to 8 slots
- Motion vectors for temporal MV prediction
  - Always from last coded frame
  - Fixed up by tempmv_signaling proposal
- Global motion data
  - Coded as deltas relative to last coded frame
Proposal

- Make all dependencies between frames track with reference frame structure
Before

Frame 1

Frame 2

Frame 3

Frame Context Slots (8)
After
Details

- Remove frame context slots
- Remove all syntax elements for saving and restoring frame contexts
- Instead, always save frame context with reference buffers
  - When storing a reference frame, store updated probabilities, temporal MVs, etc., too
- No more syntax to reset frame contexts
  - Implicit on a keyframe
Complexities

• Interaction between reference number and function
  – In current encoder, first reference is always last frame from same layer
  – Need to re-order reference list to use probabilities from long-term reference (golden frame), alt-ref, etc.

• Using a previous frame context for intra-only frames
  – Currently not supported (same as VP9)

• Using probabilities from a non-reference frame
  – No longer supported
  – With up to 6 of 8 references per frame, impact seems low
TODO

- Still some things to move to frame context
  - Global motion
  - Frame size prediction
Chroma from Luma

Luc Trudeau (luc@trud.ca)
David Michael Barr (b@rr-dav.id.au)
New Update to CfL Proposal

- Current evolution of `draft-egge-netvc-cfl`
  - A lot has changed
- Complementary to `draft-midtskogen-netvc-chromapred`
What is Chroma from Luma?

- Exploits *local* correlation between color planes

Original

Reconstructed luma + chroma DC prediction (SSE: 315,364)

Reconstructed luma + CfL linear prediction
\[ \alpha = \langle -0.25, -0.25 \rangle \]
(SSE: 116,884)
Adapting CfL for AV1

- Daala CfL predicted coefficients directly
- Hard to do frequency-domain processing in AV1
  - Up to 16 different transform types
  - Luma transform type may not match chroma type
  - Luma transform size may not match chroma size
    - Could handle this for the DCT
    - Hard to do for all transform type combinations

- Lesson from Daala
  - Implicit model building is not very good
  - Okay when it works, can be very bad when it doesn’t
### Proposed vs. Prior CfLs

<table>
<thead>
<tr>
<th>Prediction Domain</th>
<th>LM Mode¹</th>
<th>Thor CfL²</th>
<th>Daala CfL³</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitstream Signaling</td>
<td>No</td>
<td>No</td>
<td>Sign bit, PVQ gain</td>
<td>Polar index</td>
</tr>
<tr>
<td>Activation Mechanism</td>
<td>LM Mode (4×4 and 8×8)</td>
<td>Threshold</td>
<td>Signaled per band</td>
<td>CFL_PRED (UV-only mode)</td>
</tr>
<tr>
<td>Requires PVQ</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Encoder Model Fitting</td>
<td>Yes</td>
<td>Yes</td>
<td>Via PVQ</td>
<td>Search</td>
</tr>
<tr>
<td>Decoder Model Fitting</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

1. JCTVC-E266 (Chen et al.)
2. draft-midtskogen-netvc-chromapred
3. draft-egge-netvc-cfl
CfL: Encoder Side

- Luma average computed over transform block
  - Minimizes luma buffering in decoder
- Chroma DC computed over whole prediction block
  - Simplifies encoder search

Diagram:

- Reconstructed luma pixels → Average → Subsample → Search
- Original chroma pixels
- Chroma DC prediction

\[ \alpha_{Cb}, \alpha_{Cr} \]
CfL: Decoder Side

- Luma average computed over transform block
  - Minimizes luma buffering in decoder
- Chroma DC computed over whole prediction block
  - Simplifies encoder search
Complications (1)

- Sub-8×8 block sizes for 4:2:0 (and 4:2:2, 4:4:0)
  - Chroma uses one 4×4 transform with mode information from bottom right
    - Chroma is intra if bottom-right luma is intra
  - Must buffer luma from inter blocks above, left
- Chroma DC prediction for non-square blocks
  - Number of above + left pixels not a power of two
  - Average requires division (can implement with LUT)
  - AV1 adding rectangular transforms with rectangular intra prediction, so not a CfL-specific problem
Complications (2)

- Boundary handling
  - Frame size rounded up to nearest multiple of 8
  - Intra transform blocks completely outside this region are *not coded*
  - Chroma transform blocks may cover larger area than luma
Boundary Handling: Example

- Use simple extrapolation to recover missing luma pixels
- Also complicates DC prediction (not CfL-specific)
Outdated Example: AV1

205_ _Vallée_de_Colca_Panorama_ _Juin_2010_ _5_de_6.y4m (subset1)
QP = 55, PSNR Cr = 37.26 dB
Analyzer link: https://goo.gl/69N6LC
Outdated Example: AV1+CfL

205_-_Vallée_de_Colca_Panorama_-_Juin_2010_-_5_de_6.y4m (subset1)
QP = 55, PSNR Cr = 38.58 dB
Analyzer link: https://goo.gl/69N6LC
Latest Results

- AWCY BD-rate results on subset1 (still images)

<table>
<thead>
<tr>
<th>PSNR</th>
<th>PSNR Cb</th>
<th>PSNR Cr</th>
<th>PSNR HVS</th>
<th>SSIM</th>
<th>MS SSIM</th>
<th>CIEDE 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2948</td>
<td>-14.8567</td>
<td>-12.7782</td>
<td>0.5817</td>
<td>0.5682</td>
<td>0.5886</td>
<td>-5.0903</td>
</tr>
</tbody>
</table>

reorder@2017-07-07T15:41:21.181Z →
cfl-mode-uniform-alpha-it1-20@2017-07-13T13:50:456Z

- Working on adjusting encoder’s luma-chroma balance
  - Currently different parts of the encoder use different weights
Questions?