Chroma from Luma Intra Prediction for NETVC

draft-egge-netvc-cfl-01

Nathan E. Egge
Luc N. Trudeau
David Barr
Mozilla and the Xiph.Org Foundation

IETF 100 - Singapore
2017 Nov 15
What is Chroma from Luma?

Intra prediction tool

*No dependencies on other frames*

Only available to chroma planes

*Predicts chroma using coincident-reconstructed luma pixels*
What’s New in -01?

**Based on the chroma from luma proposal for AV1**
Instead of Daala implementation

**No longer relies on PVQ**
Prediction is done in the spatial domain

**Considers only AC contribution of reconstructed luma pixels**
Spatial domain equivalent of shape prediction

**Uses existing chroma DC prediction for DC contribution**
Available in AV1, requires no signaling and is more precise
# What’s different?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prediction domain</strong></td>
<td>Spatial</td>
<td>Spatial</td>
<td>Frequency</td>
<td>Spatial</td>
</tr>
<tr>
<td><strong>Bitstream signaling</strong></td>
<td>No</td>
<td>No</td>
<td>Sign bit PVQ gain</td>
<td>Signs + Index</td>
</tr>
<tr>
<td><strong>Activation mechanism</strong></td>
<td>LM Mode (4x4, 8x8)</td>
<td>Threshold</td>
<td>Signaled</td>
<td>CFL_PRED (UV-only mode)</td>
</tr>
<tr>
<td><strong>Requires PVQ</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Encoder model fitting</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Via PVQ</td>
<td>Search</td>
</tr>
<tr>
<td><strong>Decoder model fitting</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

[1] draft-midtskogen-netvc-chromapred-02
[2] draft-egge-netvc-cfl-00
How Does it Work?

Reconstructed Luma Pixels → Subsample → Average

Chroma Transform-Sized Averages (Q3)

Contribution to the AC (in the spatial domain)

Signaled Scaling Factor $\alpha$ (Q3) → Scaled Values (Q0)

DC_PRED (Q0) → CfL Prediction
Why use Chroma DC_PRED?

\( \beta \) is the average chroma reference pixels for a block

\[
\beta = \frac{\sum_{i} \sum_{j} C_{ij} - \alpha \sum_{i} \sum_{j} L_{ij}}{M \times N}
\]

AC contribution is zero mean (it sums to 0)

DC_PRED predicts the average value of a block
By computing the average of the neighboring pixels adjacent to the above and left borders of the block

No Signaling required
What are Scaling Factors ($\alpha_{Cb}$, $\alpha_{Cr}$)?

Scaling factors set the tone

Scaling factors are in Q3 and range from -2 to 2

Scaling factors are chosen by a rate-constraint search

$$\alpha = \arg\min_{a \in A} (D(CfL(a)) + \lambda R(a))$$

Scaling factors are signaled to the decoder
How are Scaling Factors Signaled?

A sign can either be \([0, -, +]\)

**Signs are jointly coded**

using an 8-value\(^1\) CDF

**Each non-zero scaling factor is coded**

using a 16-value CDF \((0,2]\)

Joint sign used as context

\(^1\): \((0,0)\) is not a valid code as it is equivalent to DC_PRED
UV Mode Selection Example
(https://goo.gl/6tKaB8)

- **CFL_PRED** 17%
- **DC_PRED** 44.36%
- **TM_PRED** 7.98%
- **SMOOTH_PRED** 4.85%

Ohashi0806shield.y4m
QP = 55
# Results (AWCY High Latency)

## Subset 1

<table>
<thead>
<tr>
<th></th>
<th>BD-Rate (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>PSNR-HVS</td>
<td>SSIM</td>
<td>CIEDE2000</td>
<td>PSNR Cb</td>
<td>PSNR Cr</td>
</tr>
<tr>
<td>Average</td>
<td>-0.46</td>
<td>-0.29</td>
<td>-0.33</td>
<td>-4.65</td>
<td>-12.99</td>
<td>-10.84</td>
</tr>
</tbody>
</table>

Ref: https://arewecompressedyet.com/?job=master%402017-07-26T10%3A40%3A11.180Z&job=cfl-baseline%402017-07-29T00%3A04%3A47.130Z

## Objective-1 fast

<table>
<thead>
<tr>
<th></th>
<th>BD-Rate (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>PSNR-HVS</td>
<td>SSIM</td>
<td>CIEDE2000</td>
<td>PSNR Cb</td>
<td>PSNR Cr</td>
</tr>
<tr>
<td>Average</td>
<td>-0.43</td>
<td>-0.42</td>
<td>-0.38</td>
<td>-2.41</td>
<td>-5.85</td>
<td>-5.51</td>
</tr>
<tr>
<td>1080p</td>
<td>-0.32</td>
<td>-0.37</td>
<td>-0.28</td>
<td>-2.52</td>
<td>-6.80</td>
<td>-5.31</td>
</tr>
<tr>
<td>1080p Screen</td>
<td>-1.82</td>
<td>-1.72</td>
<td>-1.71</td>
<td>-8.22</td>
<td>-17.76</td>
<td>-12.00</td>
</tr>
<tr>
<td>360p</td>
<td>-0.15</td>
<td>-0.05</td>
<td>-0.10</td>
<td>-0.80</td>
<td>-2.17</td>
<td>-6.45</td>
</tr>
<tr>
<td>720p</td>
<td>-0.12</td>
<td>-0.11</td>
<td>-0.07</td>
<td>-0.52</td>
<td>-1.08</td>
<td>-1.23</td>
</tr>
</tbody>
</table>

Ref: https://arewecompressedyet.com/?job=master%402017-09-13&job=cfl-inter%402017-09-13T14%3A13%3A13.918Z

1. **CIEDE2000** is the only metric that combines luma and chroma plane (*The distance measured is more perceptually uniform*)
# Awesome for Gaming (Twitch dataset)

<table>
<thead>
<tr>
<th></th>
<th>BD-Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
</tr>
<tr>
<td>Average</td>
<td>-1.01</td>
</tr>
</tbody>
</table>

Ref: [https://arewecompressedyet.com/?job=no-cfl-twitch-cpu2-60frames%402017-09-18T15%3A39%3A17.543Z&job=cfl-inter-twitch-cpu2-60frames%402017-09-18T15%3A40%3A24.181Z](https://arewecompressedyet.com/?job=no-cfl-twitch-cpu2-60frames%402017-09-18T15%3A39%3A17.543Z&job=cfl-inter-twitch-cpu2-60frames%402017-09-18T15%3A40%3A24.181Z)

## Notable Mentions

<table>
<thead>
<tr>
<th>Game</th>
<th>MD</th>
<th>PSNR</th>
<th>PSNR-HVS</th>
<th>SSIM</th>
<th>CIEDE2000¹</th>
<th>PSNR Cb</th>
<th>PSNR Cr</th>
<th>MS SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minecraft</td>
<td>MINECRAFT_10_120f.y4m</td>
<td>-3.76</td>
<td>-3.13</td>
<td>-3.68</td>
<td>-20.69</td>
<td>-31.44</td>
<td>-25.54</td>
<td>-3.28</td>
</tr>
<tr>
<td>GTA V</td>
<td>GTAV_0_120f.y4m</td>
<td>-1.11</td>
<td>-1.11</td>
<td>-1.01</td>
<td>-5.88</td>
<td>-15.39</td>
<td>-5.57</td>
<td>-1.04</td>
</tr>
<tr>
<td>Starcraft</td>
<td>STARCRAFT_10_120f.y4m</td>
<td>-1.41</td>
<td>-1.43</td>
<td>-1.38</td>
<td>-4.15</td>
<td>-6.18</td>
<td>-6.21</td>
<td>-1.43</td>
</tr>
</tbody>
</table>