Predicting Chroma from Luma using Frequency Domain Intra Prediction in Codecs Based on Lapped Transforms

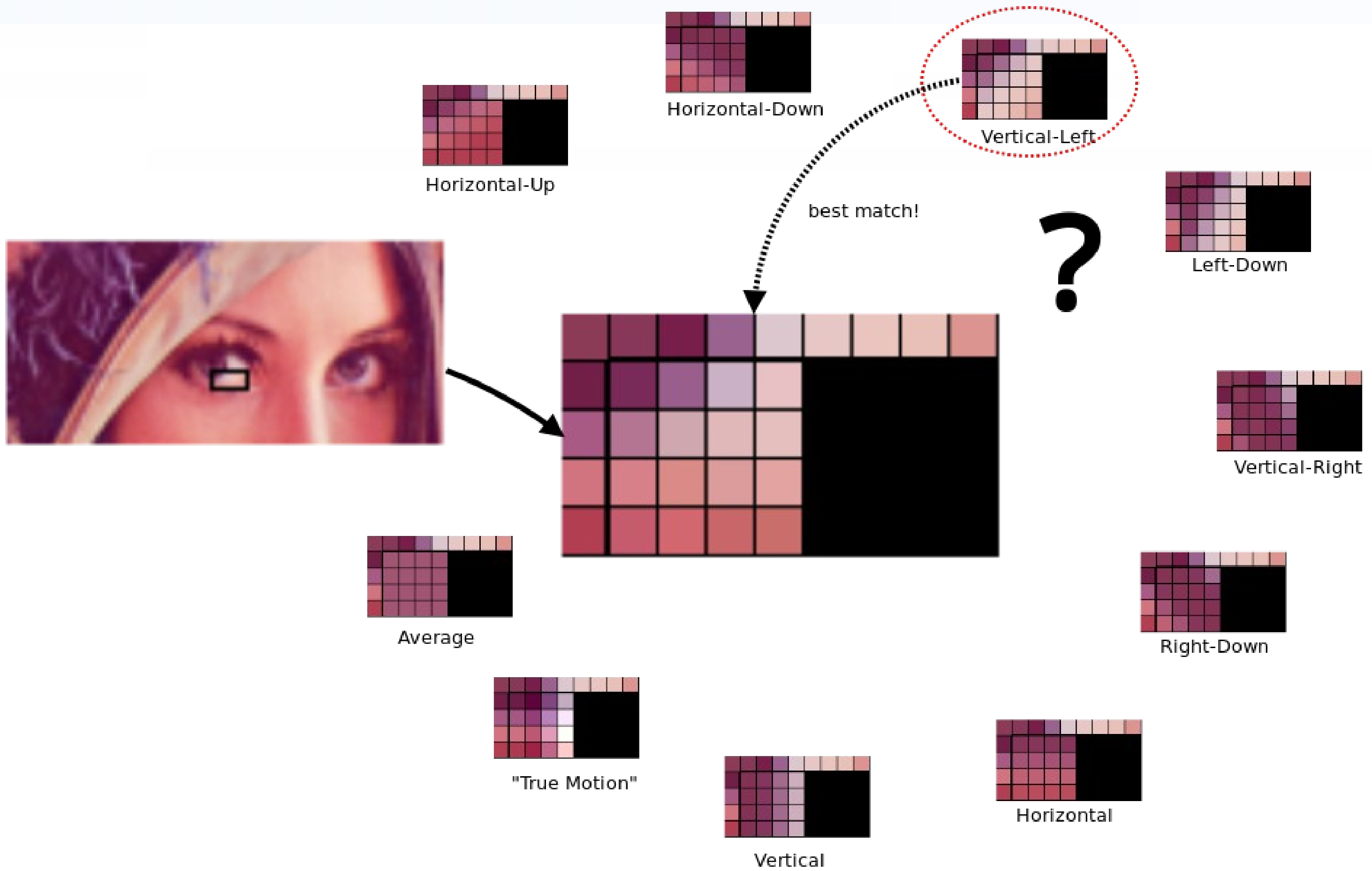
Nathan E. Egge
Jean-Marc Valin

Mozilla & The Xiph.Org Foundation

Intra-Prediction of Chroma

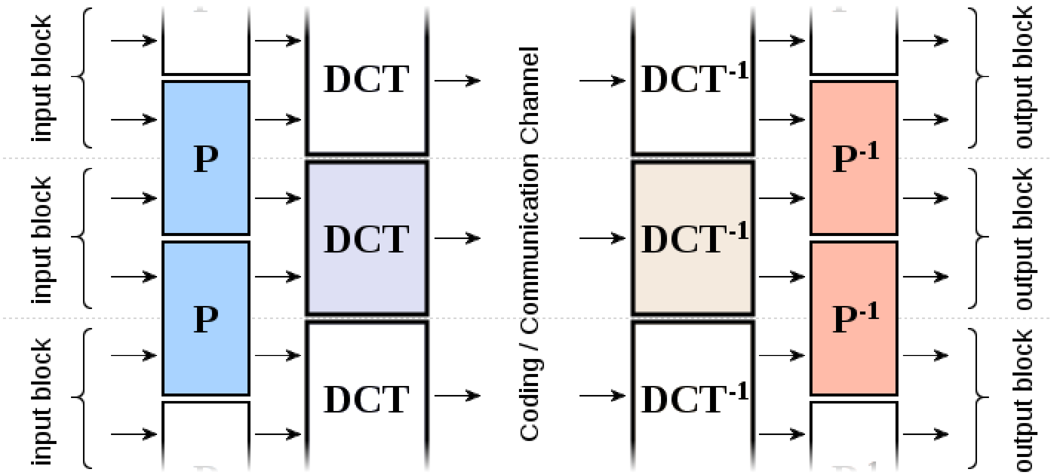
- In 4:2:0 image data, chroma is 50% of luma
- Chroma predicted spatially by signalling a directional mode
  - Reconstructed neighbors must be available to decode a block
  - Limited to predicting from current color plane
- Cross-channel correlation not exploited
- Does not work with codecs using lapped transforms

Spatial Domain Intra-Prediction



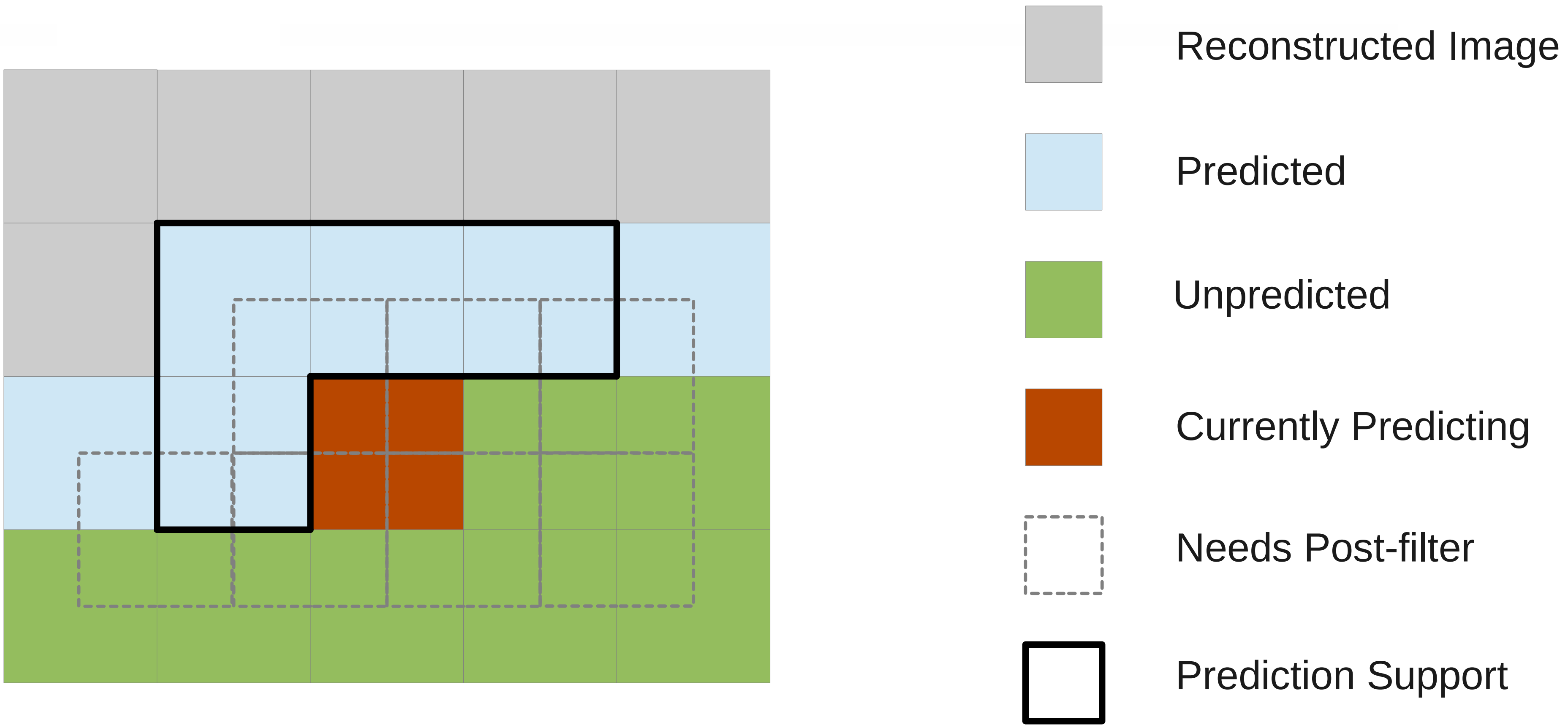
The intra-prediction modes for 4x4 blocks in WebM (VP8).

Lapped Transforms



Decoding an Intra Frame with Lapped Transforms

Neighboring blocks:



Predicting Chroma from Luma

- Key insight: YUV conversion de-correlates luma and chroma globally, but local relationship exists [1]
- Both encoder and decoder compute linear regression:

$$\alpha = \frac{N \cdot \sum_i L_i \cdot C_i - \sum_i L_i \sum_i C_i}{N \cdot \sum_i L_i \cdot L_i - \left(\sum_i C_i\right)^2} \qquad \beta = \frac{\sum_i C_i - \alpha \cdot \sum_i L_i}{N}$$

- Use reconstructed luma coefficients to predict coincident chroma coefficients:

$$C(u,v) = \alpha \cdot L(u,v) + \beta$$

- Not selected for HEVC due to 20-30% increased complexity

[1] S.H. Lee & N.I. Cho: “Intra prediction method based on the linear relationship between the channels for YUV 4:2:0 intra coding” ICIP 2009, pp. 1033-1036

Adapting Chroma from Luma to the Frequency Domain

- Key insight: LT and DCT are both linear transforms so similar relationship exists in frequency domain
- Both encoder and decoder compute linear regression using 4 LF coefficients from Up, Left and Up-Left
- Use reconstructed luma coefficients to predict coincident chroma coefficients:

$$C_{DC} = \alpha_{DC} \cdot L_{DC} + \beta_{DC}$$

$$C_{AC}(u,v) = \alpha_{AC} \cdot L_{AC}(u,v)$$

- Still expensive, but cost constant with block size

Block Size	SD-CfL		FD-CfL	
	Adds	Mults	Adds	Mults
N x N	4*N+2	8*N+3	2*12+5	4*12+5
4 x 4	18	35	29	53
8 x 8	34	67	29	53
16 x 16	66	131	29	53

Example

Original uncompressed image

Example

Reconstructed luma with predicted chroma using FD-CfL

Frequency Domain CfL

- Adapted CfL algorithm to the frequency domain
  - No signalling overhead
    - Implicitly defined model parameters (α_{DC}, β_{DC} , α_{AC})
    - Increased decoder complexity
  - Model parameters could be signalled for use cases
  - Works with existing LT based codecs using scalar quantization

Perceptual Vector Quantization

- Separate “gain” (contrast) from “shape” (spectrum)
  - Vector = Magnitude × Unit Vector (point on sphere)
- Given prediction vector $\mathbf{r}$
  - “gain” predicted by magnitude

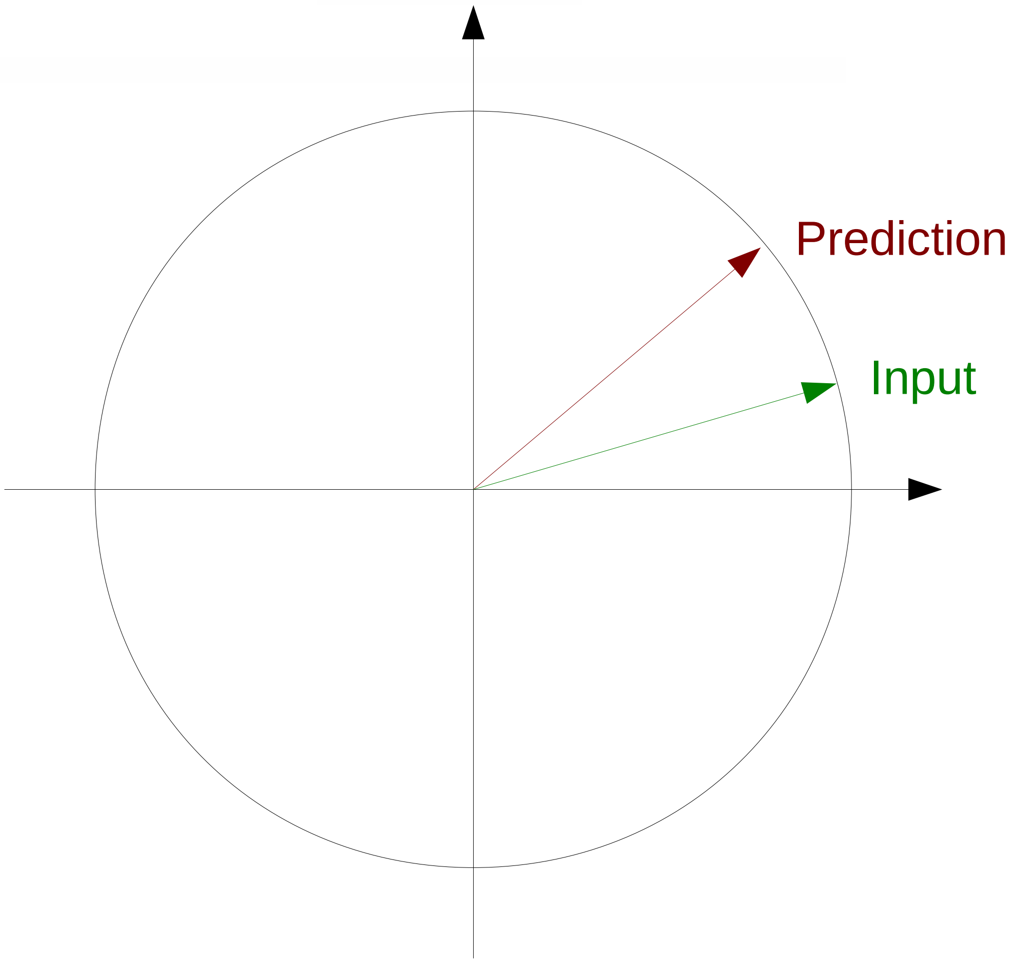
$$\hat{g} = \gamma_g \cdot Q + \|\mathbf{r}\|$$

  - “shape” predicted using Householder reflection

$$\mathbf{v} = \frac{\mathbf{r}}{\|\mathbf{r}\|} + s \cdot \mathbf{e}_m \qquad \mathbf{z} = \mathbf{x} - 2\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

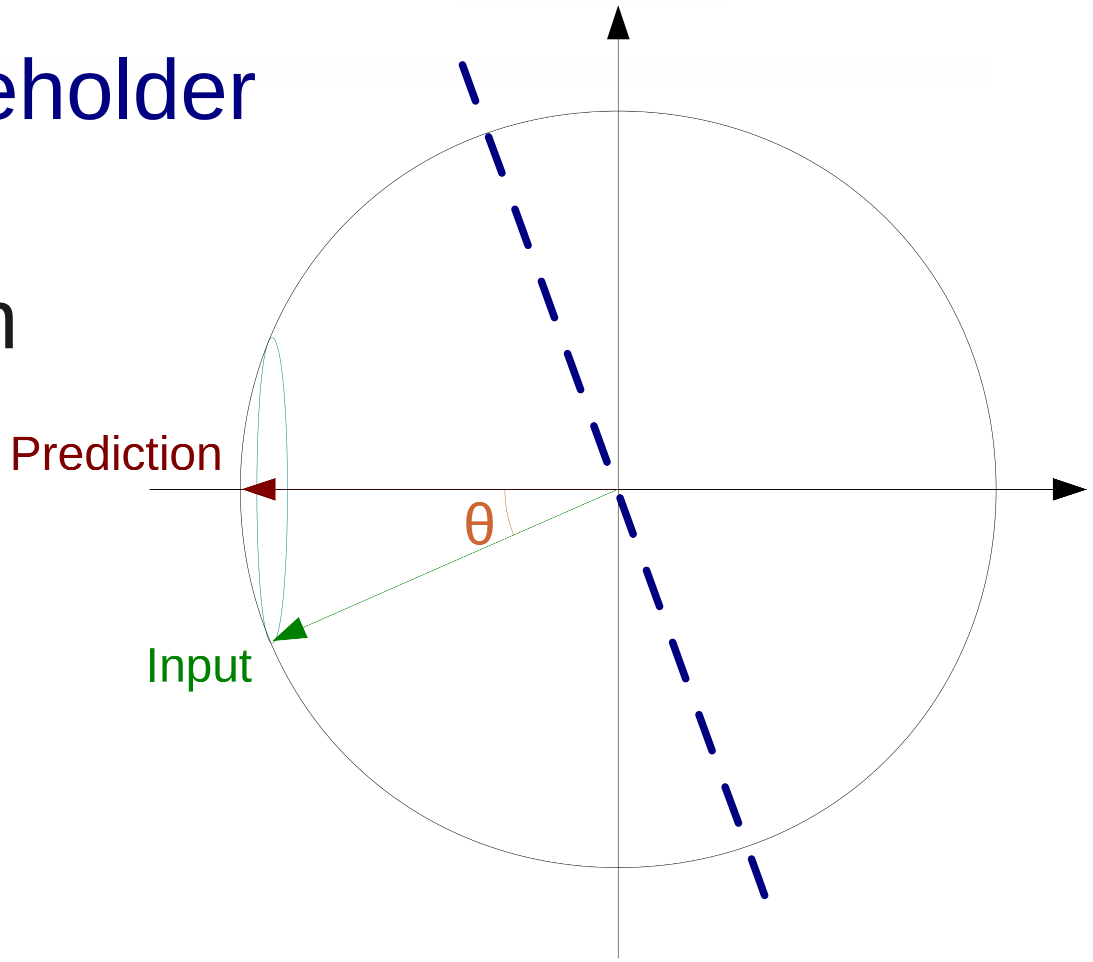
Shape Prediction Example

- Input + Prediction



Shape Prediction Example

- Input + Prediction
- Compute Householder Reflection
- Apply Reflection
- Compute & code angle
- Code other dimensions



PVQ Prediction with CfL

- Consider prediction of 15 AC coefficients of 4x4 Cb
- The 15-dimensional predictor $\mathbf{r}$ is scalar multiple of coincident reconstructed luma coefficients $\hat{\mathbf{x}}_L$

$$C_{AC}(u,v) = \alpha_{AC} \cdot L_{AC}(u,v) \implies \mathbf{r} = \alpha_{AC} \cdot \hat{\mathbf{x}}_L$$

- Thus “shape” predictor is almost exactly $\hat{\mathbf{x}}_L$

$$\frac{\mathbf{r}}{\|\mathbf{r}\|} = \frac{\alpha_{AC} \cdot \hat{\mathbf{x}}_L}{\|\alpha_{AC} \cdot \hat{\mathbf{x}}_L\|} = \operatorname{sgn}(\alpha_{AC}) \frac{\hat{\mathbf{x}}_L}{\|\hat{\mathbf{x}}_L\|}$$

- Only difference is *direction* of correlation!

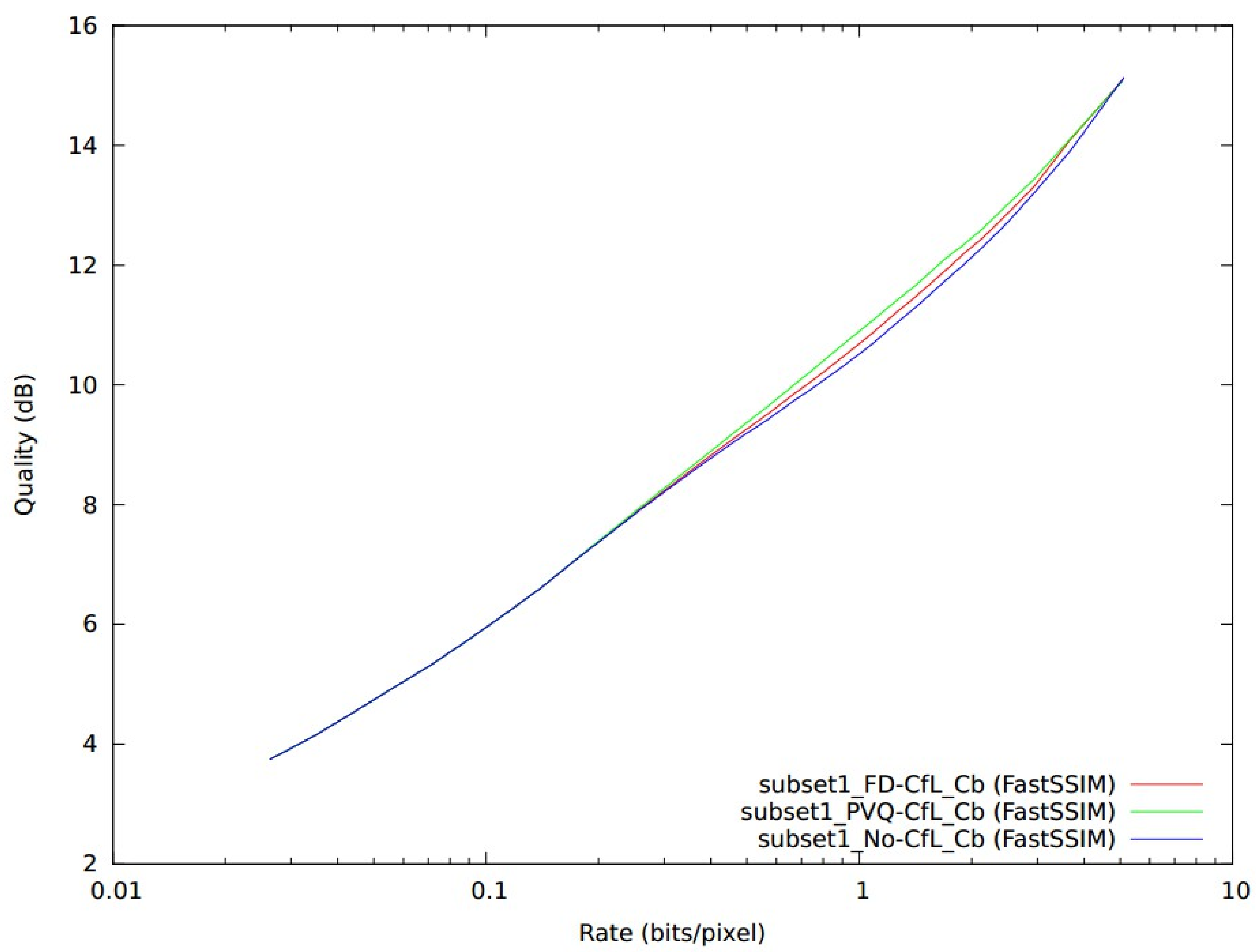
PVQ Chroma from Luma

1: Let $\mathbf{r} = \hat{\mathbf{x}}_L$, compute θ
2: If $\theta = 0$ prediction is exact, code θ
3: Else
4: Code a *flip* flag, $f = \theta > 90°$
5: If f, let $\mathbf{r} = -\hat{\mathbf{x}}_L$
6: Code $\mathbf{x}_C$ with PVQ using predictor $\mathbf{r}$
7: End

Still Image Experiment

- Sample of 50 high resolution still images taken from Wikipedia down-sampled to 1 megapixel
- Comparison of No-CfL, FD-CfL and PVQ-CfL
  - Encode with 28 different quantization levels
  - Compute rate/distortion on Cb and Cr planes using four metrics: PSNR, PSNR-HVS, SSIM, FastSSIM
  - Hold all other techniques constant

Still Image Experiment



Still Image Experiment Cont.

Computation of the Bjontegaard distance (improvement) between two rate-distortion curves

Metric	Cb (plane 1)		Cr (plane 2)	
	Δ Rate (%)	Δ SNR (dB)	Δ Rate (%)	Δ SNR (dB)
PSNR	-1.87644	0.07678	-0.90748	0.04650
PSNR-HVS	-2.57971	0.13205	-1.08077	0.06460
SSIM	-3.09834	0.08842	-1.81715	0.06315
FastSSIM	-3.01455	0.06602	-1.81869	0.04385

Improvement moving from No-CfL to FD-CfL

Metric	Cb (plane 1)		Cr (plane 2)	
	Δ Rate (%)	Δ SNR (dB)	Δ Rate (%)	Δ SNR (dB)
PSNR	-3.13262	0.12853	-1.47899	0.07590
PSNR-HVS	-5.19186	0.26913	-2.31499	0.13921
SSIM	-5.54403	0.15962	-3.45484	0.12093
FastSSIM	-6.10963	0.13577	-4.59056	0.11116

Improvement moving from No-CfL to PVQ-CfL

Conclusions & Future Work

- Introduced 2 algorithms for Chroma-from-Luma intra prediction in codecs using LT
  - FD-CfL suitable for use with scalar quantization
  - PVQ-CfL extends gain-shape quantization
    - No additional per block complexity
    - Improved performance (both rate and quality)
- Can we use both reconstructed Luma and Cb with PVQ to predict Cr?

Resources

- Daala codec website: https://xiph.org/daala/
- Daala Technology Demos: https://people.xiph.org/~xiphmont/demo/daala/
- Git repository: https://git.xiph.org/
- IRC: #daala channel on irc.freenode.net
- Mailing list: daala@xiph.org

Questions?