



# Perceptually-Driven Video Coding with the Daala Video Codec

---

**Timothy B. Terriberry**



# Summary



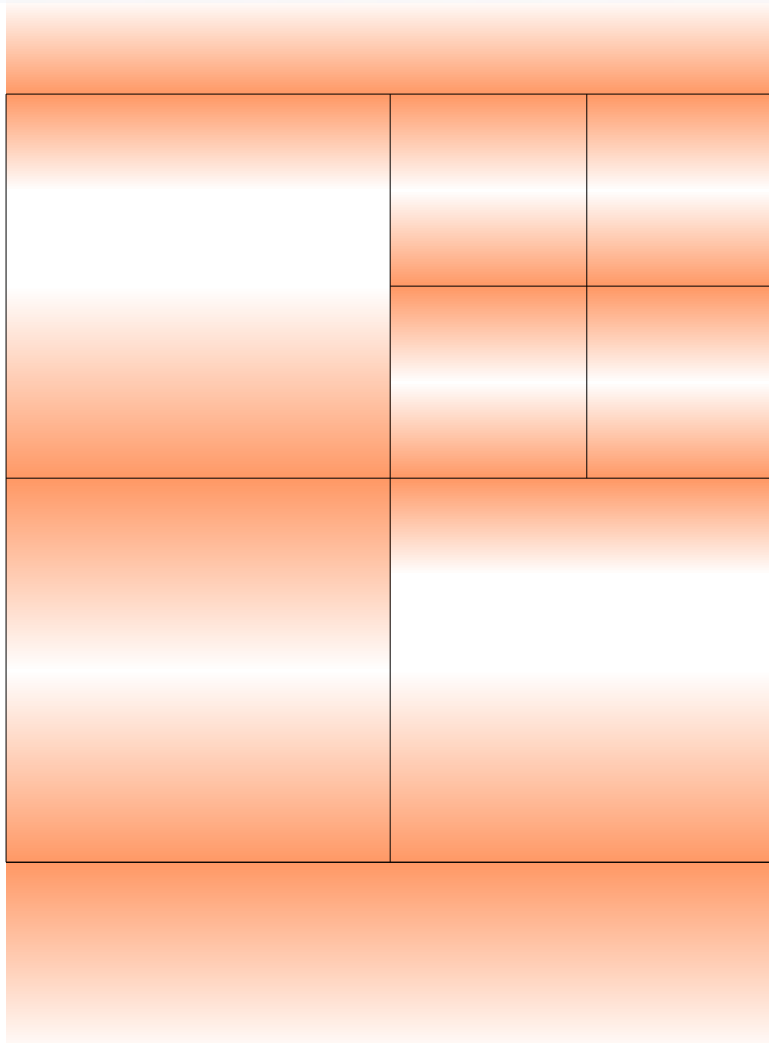
- Daala is an attempt to completely avoid royalty-bearing technologies
- Used many unconventional tools
- Some worked well, others more challenging
  - We think the challenges are more interesting
- Many lessons learned that can inform AV1 development
  - Only a few presented here, see paper for more



# Challenge 1: Lapped Transforms with Variable Block Sizes



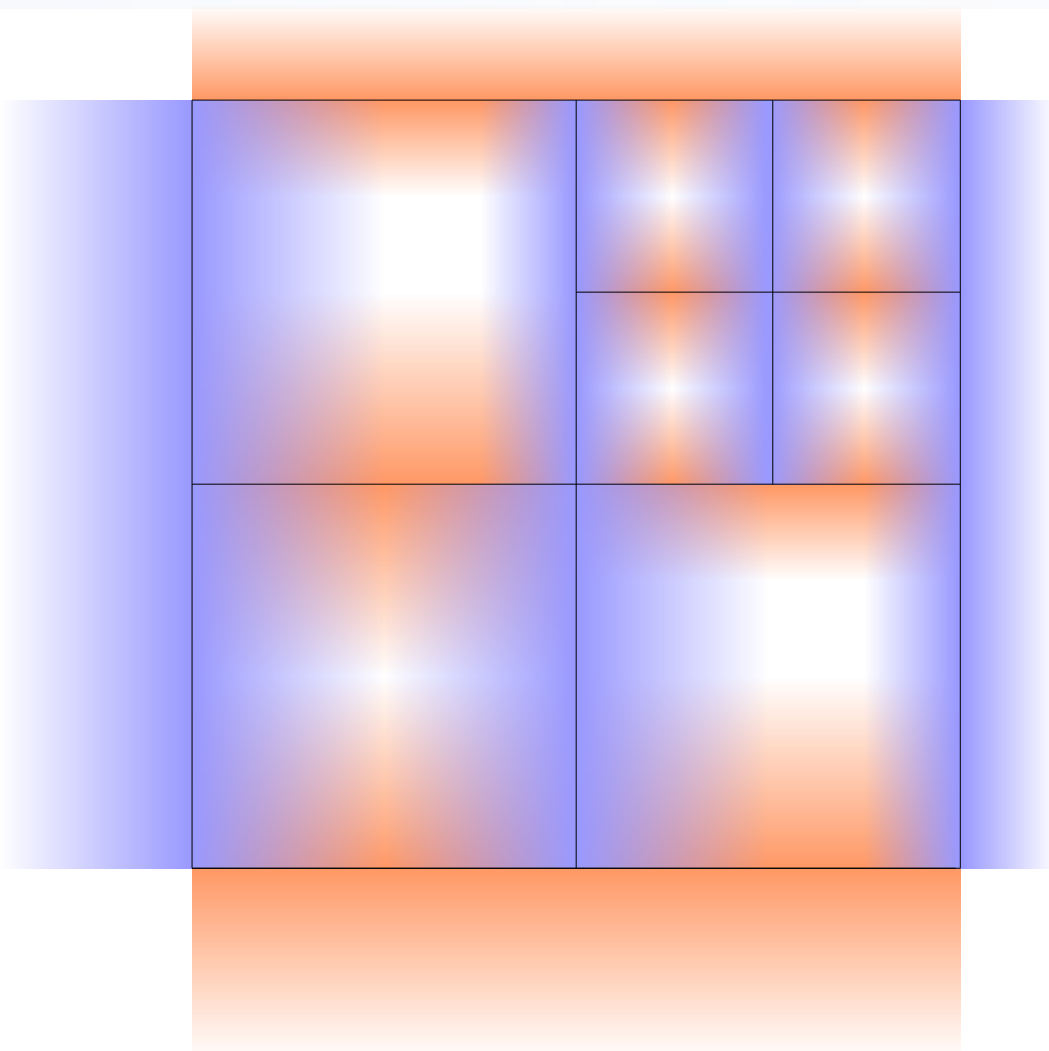
# Original Lapping Strategy



- 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)

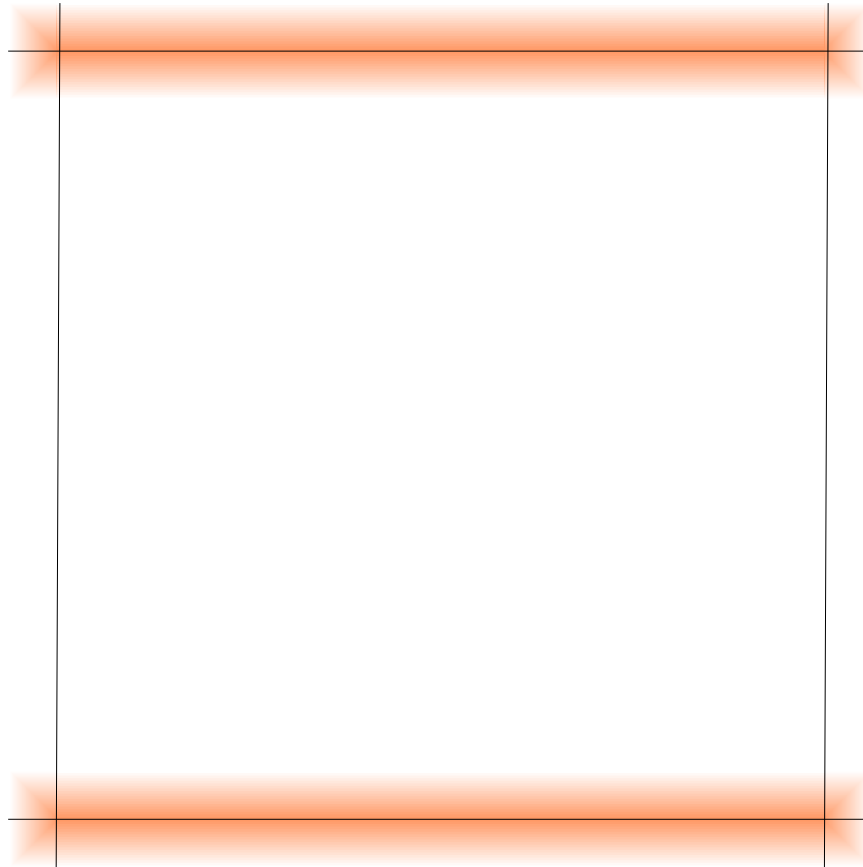




# New Filter Order



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

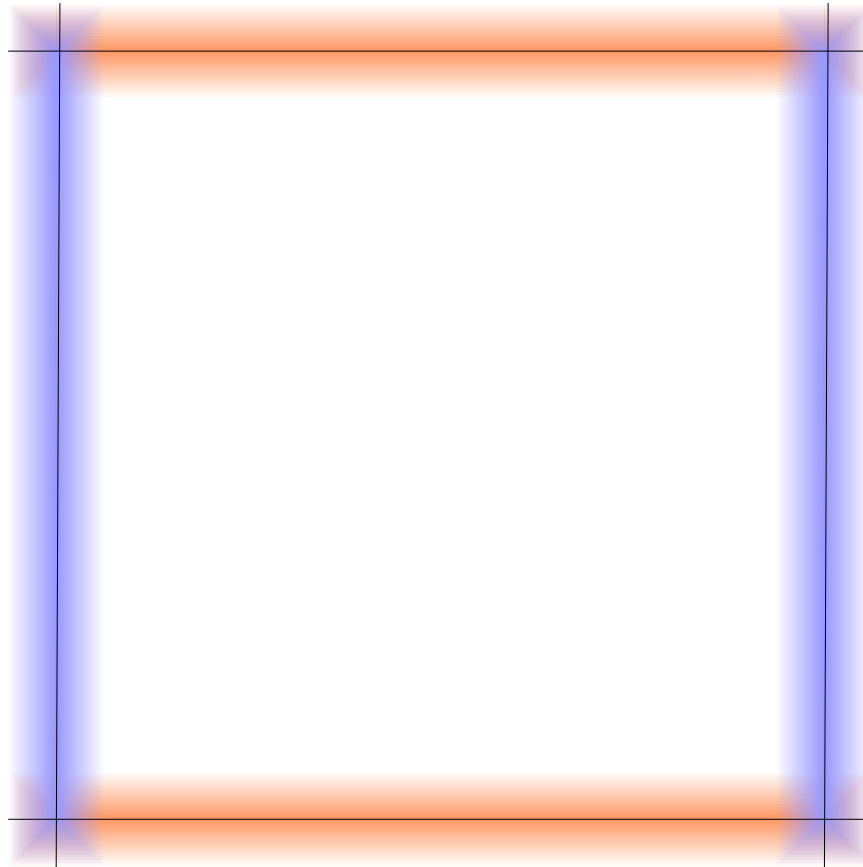




# New Filter Order



- Filter left/right superblock (64×64) edges next

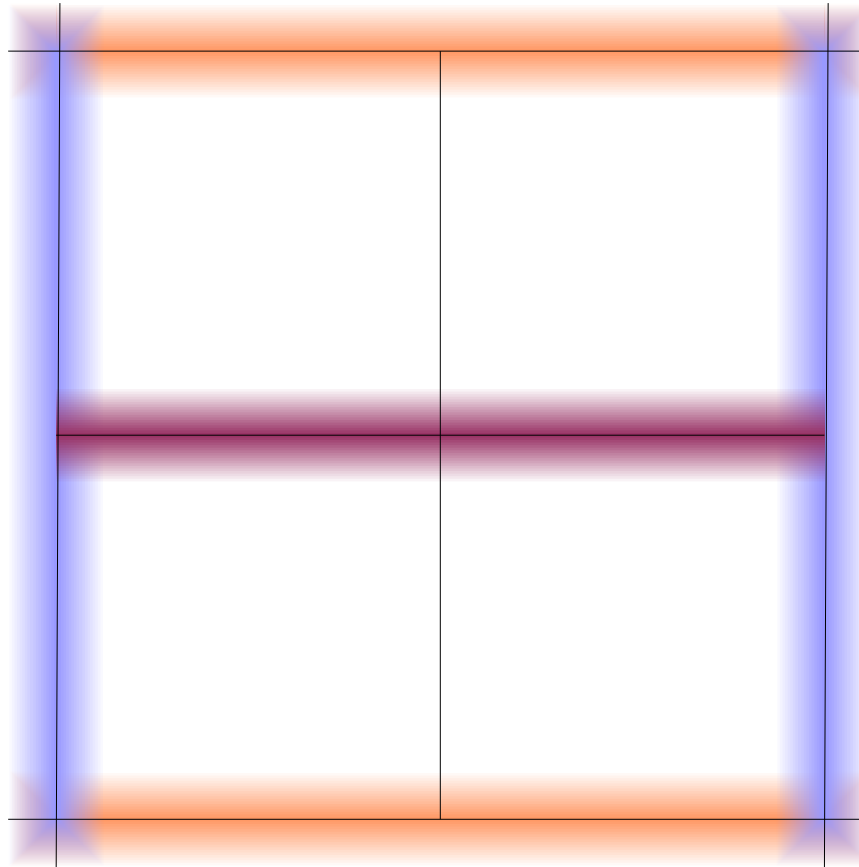




# New Filter Order



- Splitting: Filter interior edges

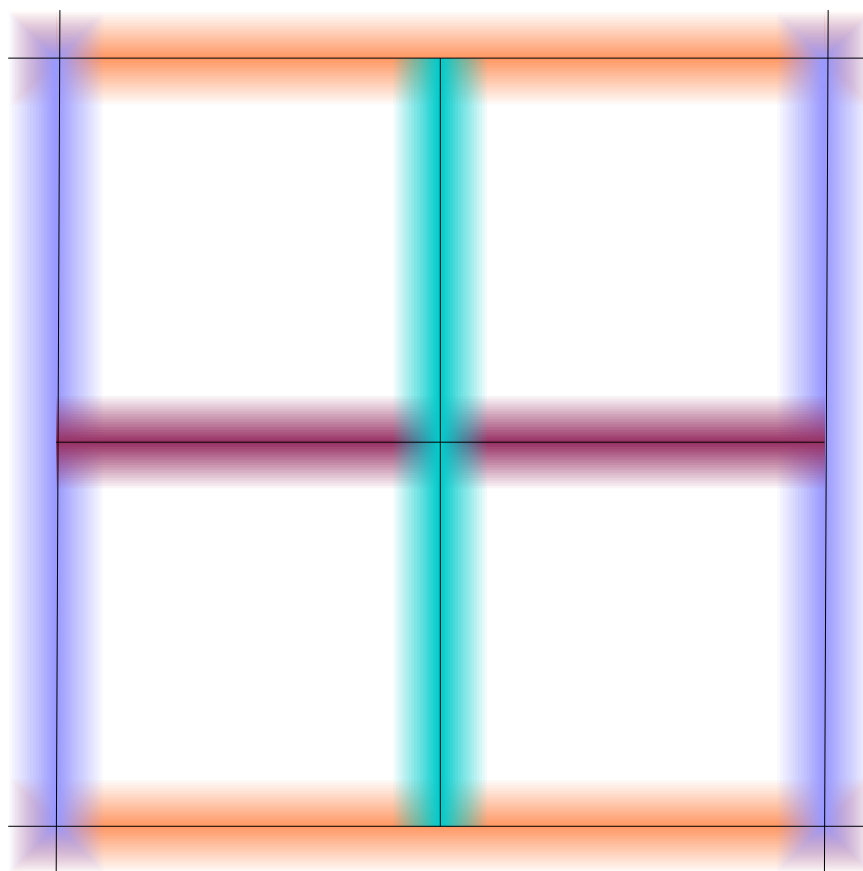




# New Filter Order



- 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

	RATE (%)	DSNR (dB)
PSNR	-10.36612	0.40904
PSNRHVS	-4.48956	0.25806
SSIM	-12.32547	0.38397
FASTSSIM	-5.20467	0.17350

- Smaller lapping means less ringing but more blockiness (especially on gradients)

- Didn't save much on ringing: 4×4 blocks have 12-pixel support instead of 8
- Eventually dropped to 4-point lapping everywhere



# 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)

Transform size	DCT+VP8 Intra			LT+FDIP (full)			LT+FDIP (sparse)	
	$C_g$	$P_g$	Total	$C_g$	$P_g$	Total	$P_g$	Total
$4 \times 4$	13.8511	2.9154	16.7665	14.9600	2.2103	17.1703	2.0799	17.0399
$8 \times 8$	15.1202	0.86940	15.9894	15.6468	0.72486	16.3716	0.58957	16.2364
$16 \times 16$	15.5870	0.19483	15.7818	15.8721	0.37593	16.2480	0.09541	15.9675



# 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



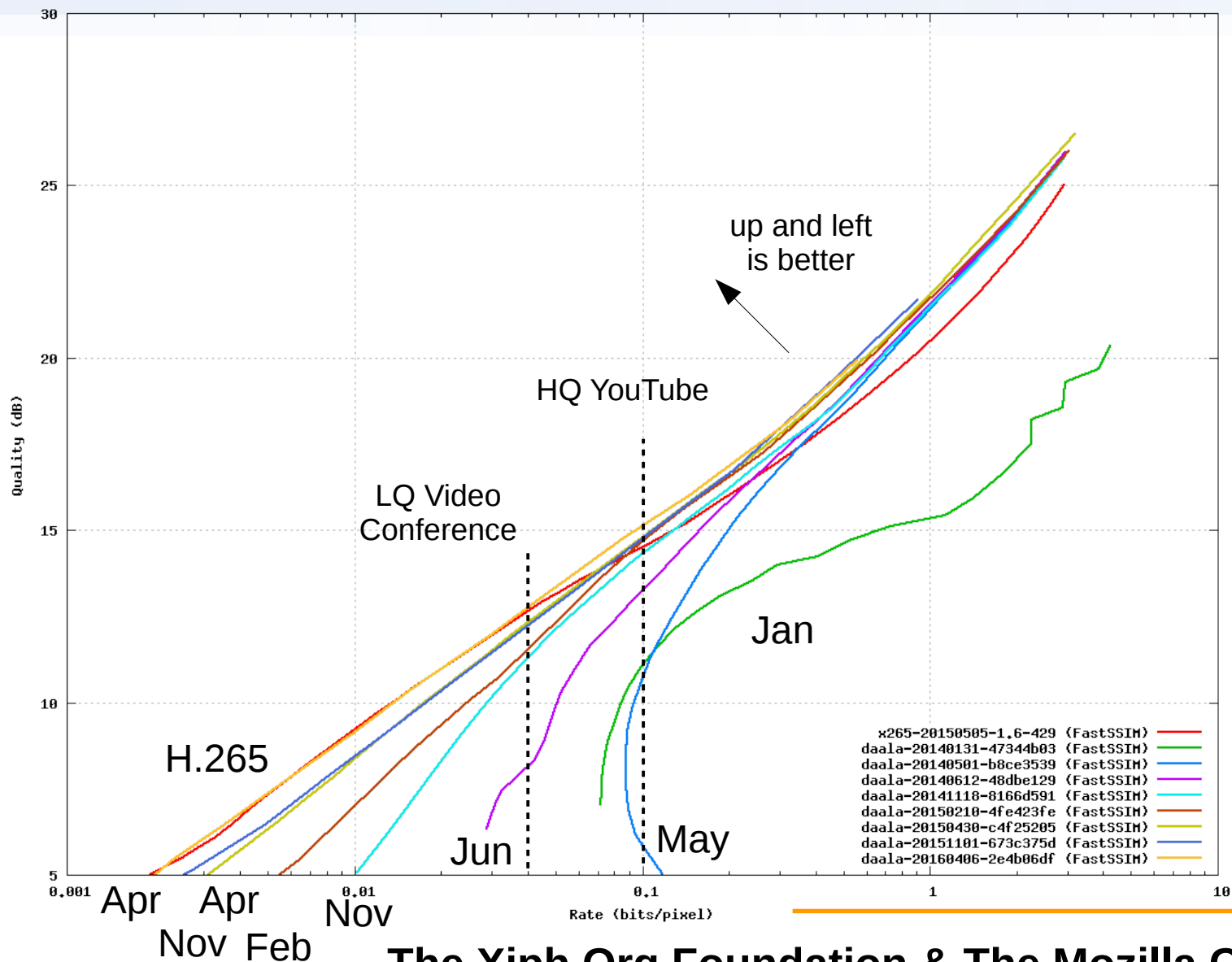
# Directions for AV1



- 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

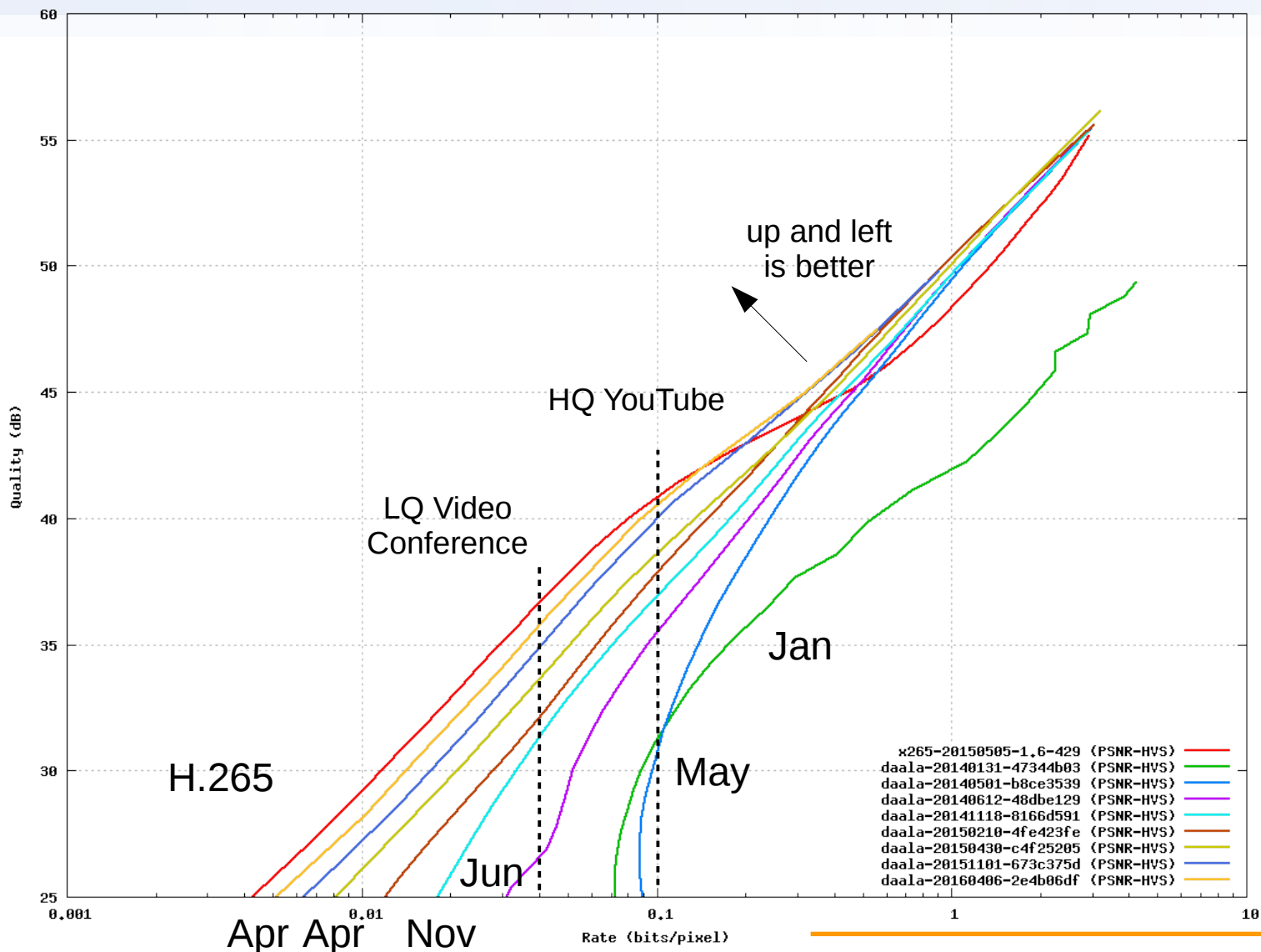


# Daala Progress (Fast MS-SSIM): January 2014 to April 2016





# Daala Progress (PSNR-HVS): January 2014 to April 2016





# Questions?