# A debanding algorithm for AV2

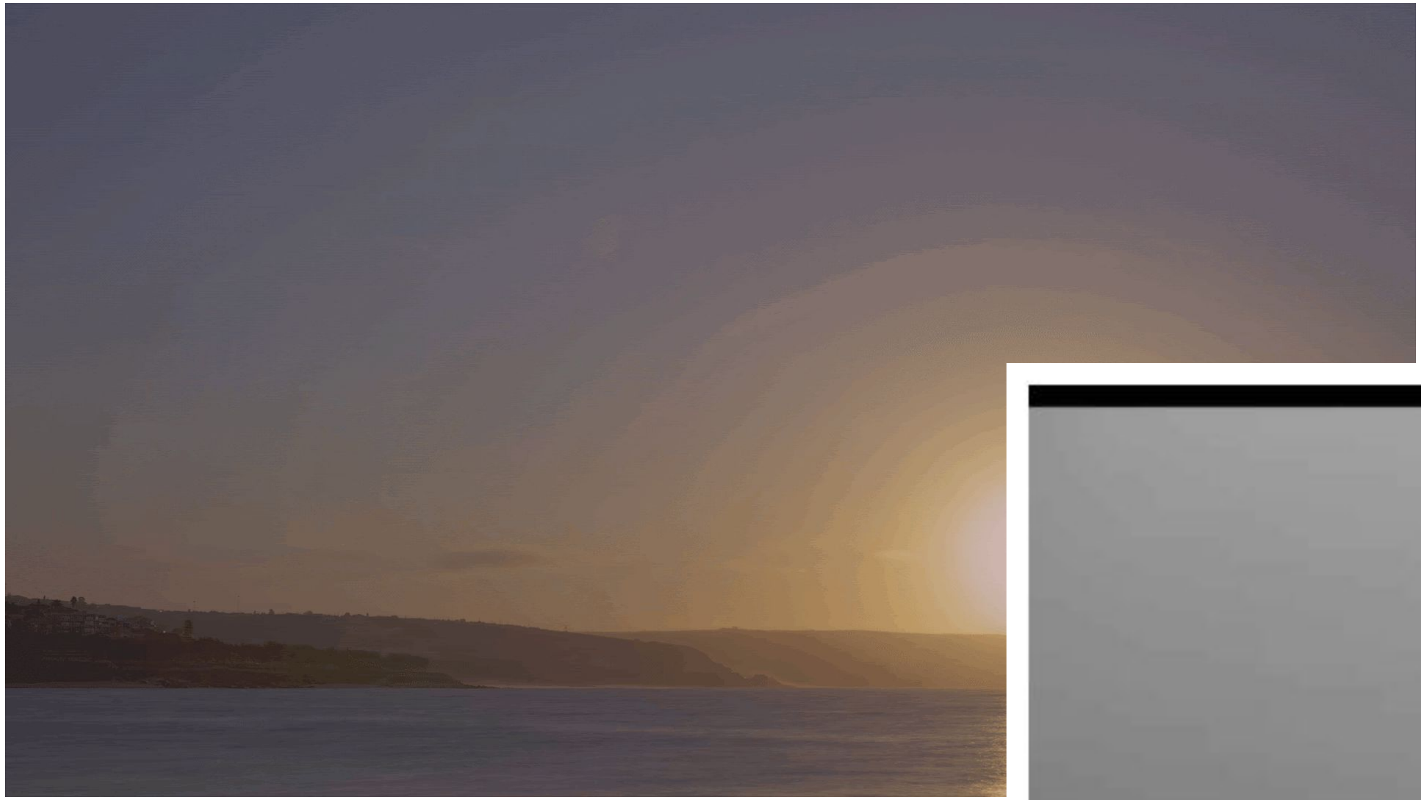
## DCC 2023

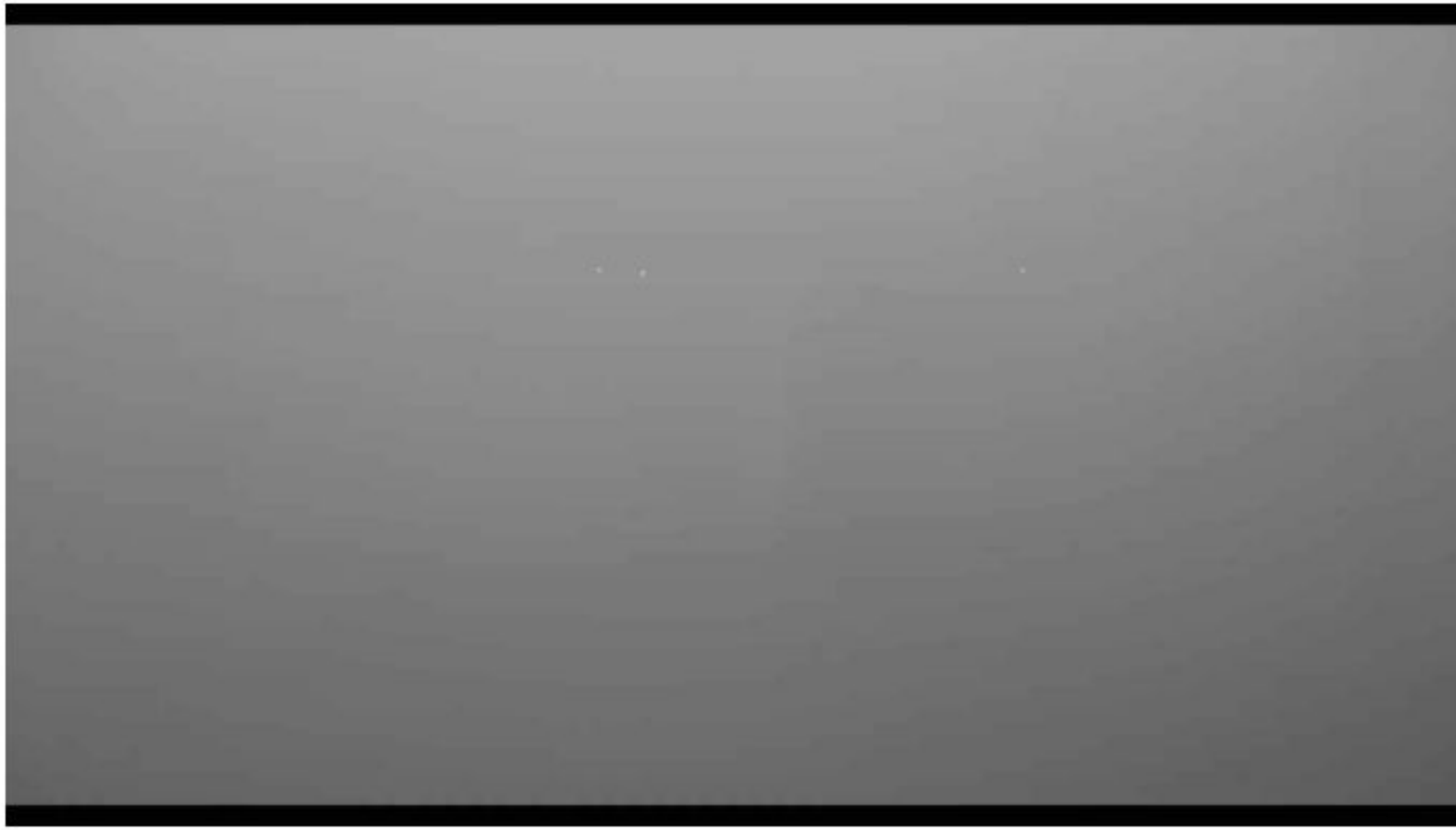
Joel Sole
jsole@netflix.com

Mariana Afonso
mafonso@netflix.com

Video Codecs and Quality
Encoding Technologies

NETFLIX



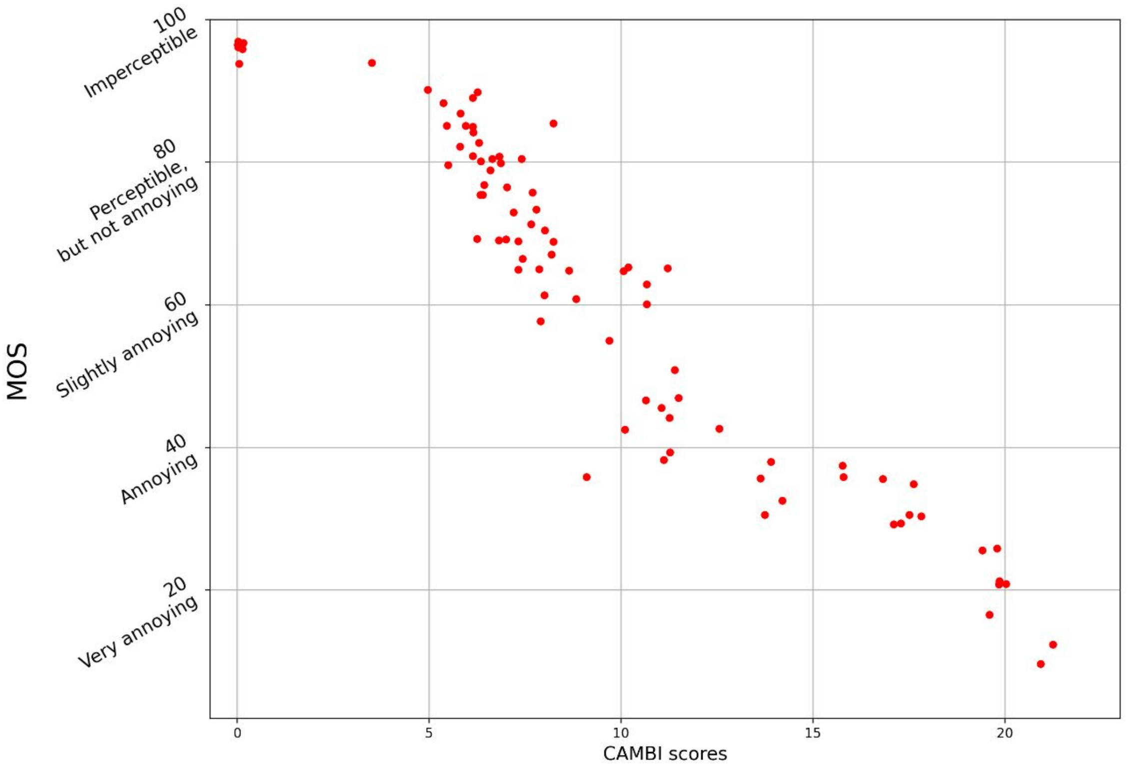


**Banding artifacts are annoying.**

**They are visible even using the latest codecs, even for 10-bit and HDR content.**

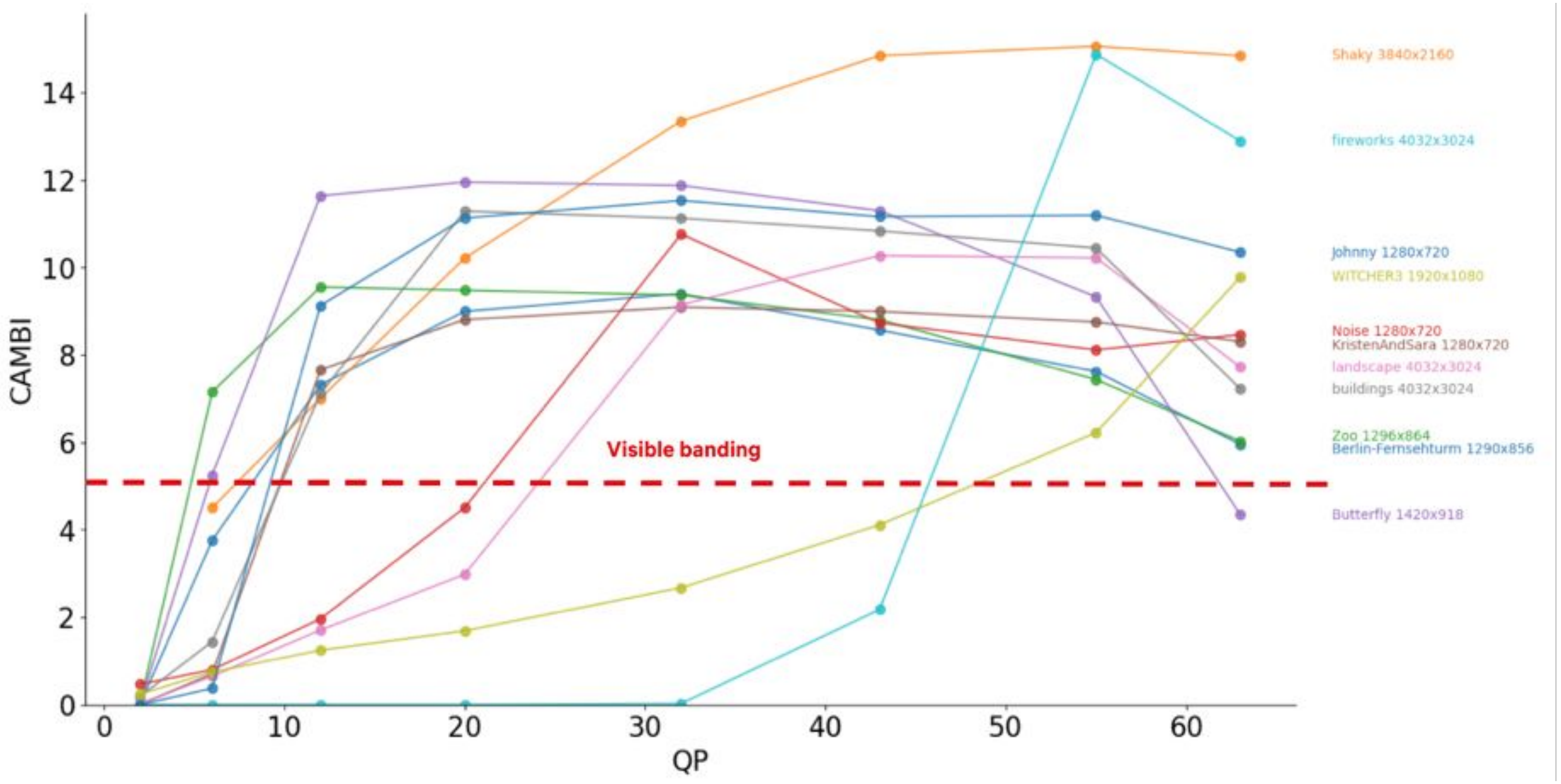
# CAMBI (Contrast-Aware Multiscale Banding Index) is a banding artifact detector highly correlated with MOS on banded content

- Open-sourced in libvmaf
- Part of the AVM Common Test Conditions



# Banding appears in libaom AV1 and AVM encodes

Libaom results for some banding-prone sequences in the AOM Common Test Conditions content



# **CAMDA** is a CAMBI-inspired Debanding Algorithm

**Implemented as an inloop and postloop filter in AVM**

# Debanding in the AVM codec has several benefits

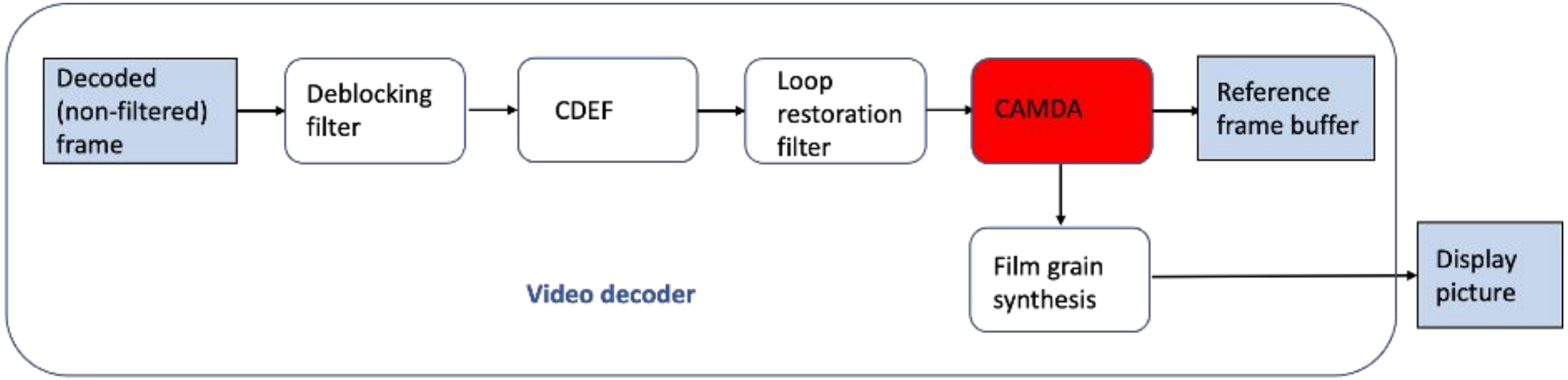
CAMDA allows to provide **subjectively better** encodes in banding-prone content

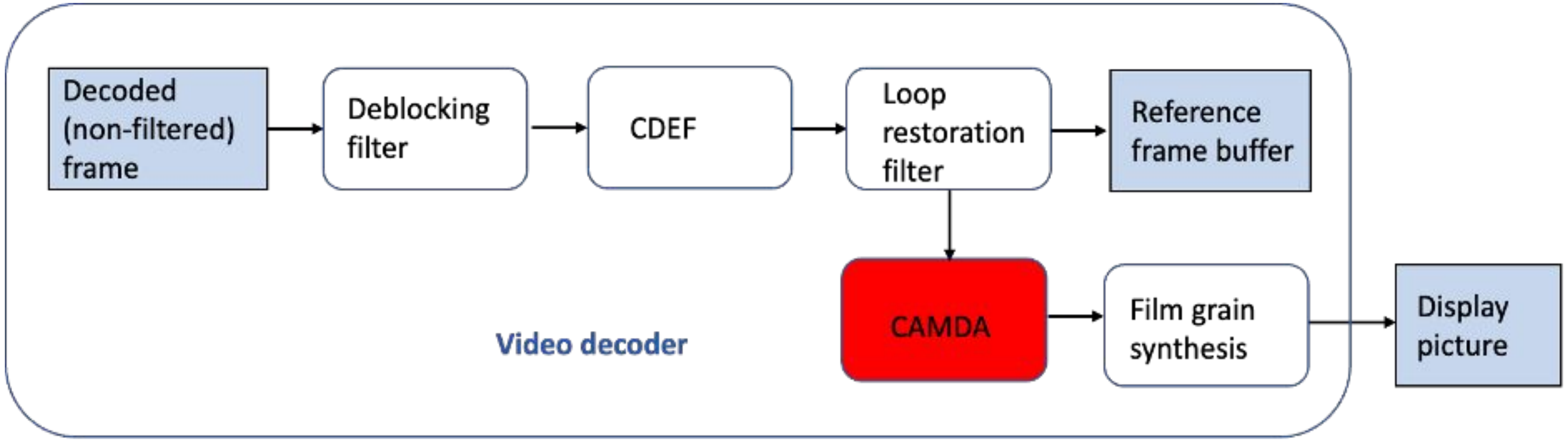
Debanding in the codec **substantially benefits** from having the **source information available**:

- Better control over where debanding is applied
  - Strength can be tuned to content and application
  - Allows respecting the artistic intent where needed
- Film grain synthesis doesn't mask banding, but it makes it difficult to apply debanding
- Consistent and controlled application across products

# **CAMDA** is a CAMBI-based Debanding Algorithm

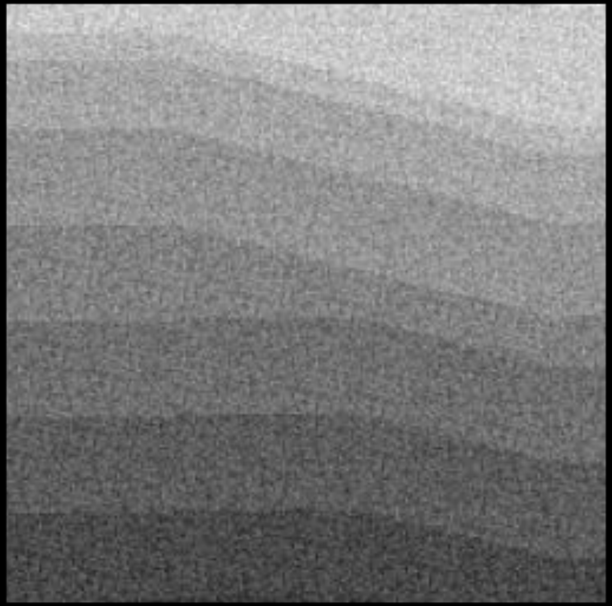
Two implementations: as the last **in-loop** filter & as the first **post-loop** filter in **AVM**



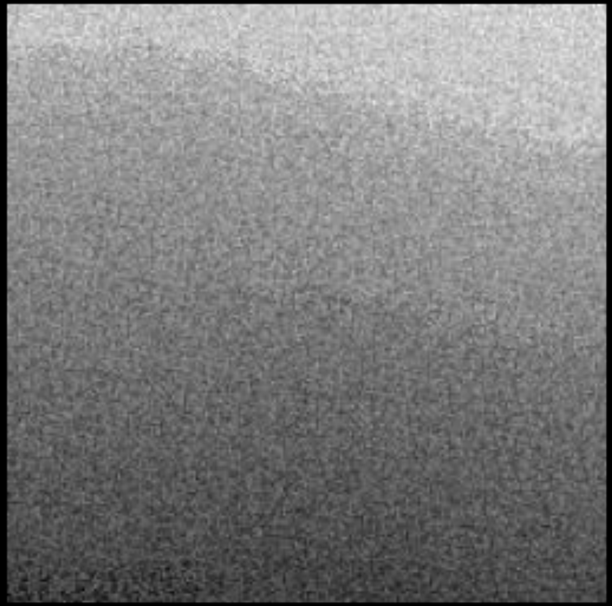



# Interaction with film grain noise

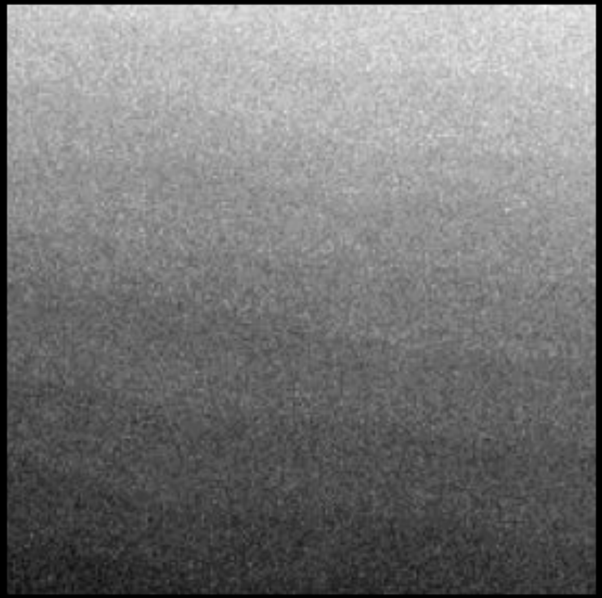
GlassHalf 1080p @ QP185, AVM 3.1



AVM



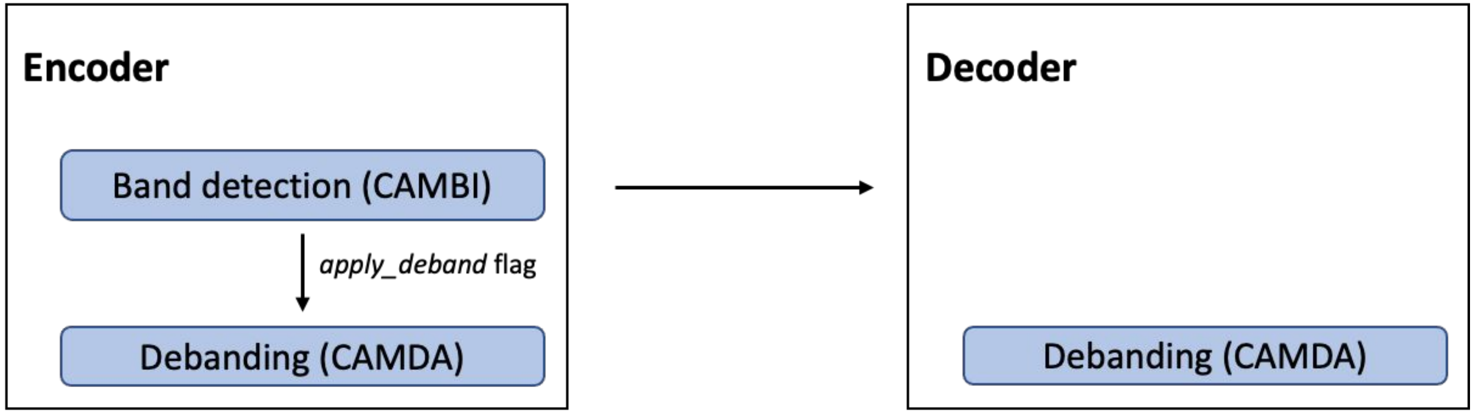
AVM + CAMDA



AVM + post-processing deband

Note: gray-levels stretched for better visibility

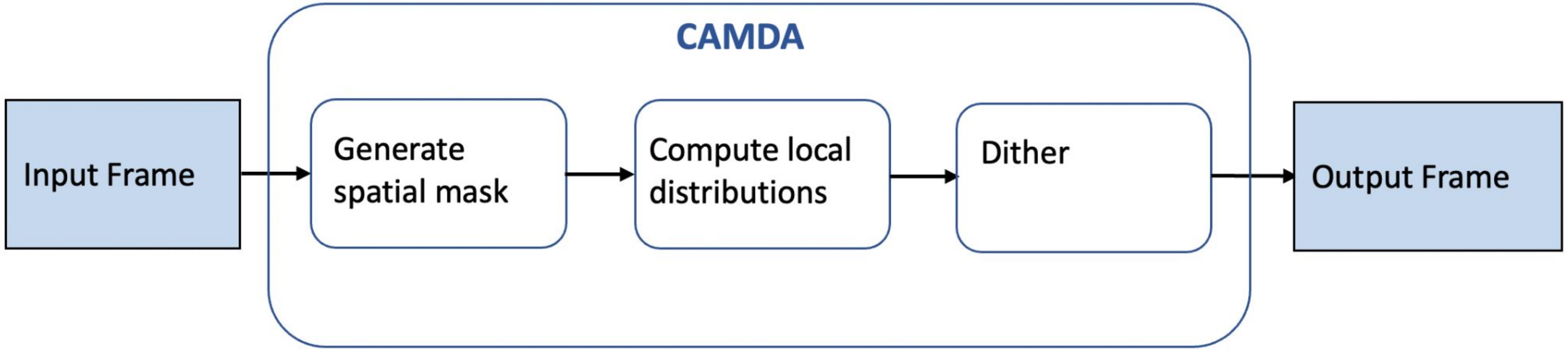
# CAMBI and CAMDA in AVM




- Code available on Gitlab
- Compile & runtime flags: **CONFIG_DEBAND** & **--enable-deband**
- Frame-level condition to apply CAMDA:

  *cambi_enc - cambi_source >= CAMBI_THRESHOLD_BANDING*
  *cambi_source < CAMBI_SOURCE_THRESHOLD*

# CAMDA leverages the CAMBI spatial mask and local distribution steps and adds a dithering step

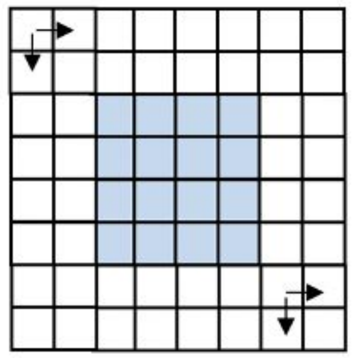



Applied to the **luma component**; **4x4 block-based** processing

# Step 1: spatial mask generation

Pixels likely to contribute to banding are identified by a spatial mask; other pixels are discarded subsequently.

1. Compute zero derivative, i.e., whether the pixel is equal to the right and bottom neighbors
2. **For each 4x4 block**, count number of zero derivative in a 7x7 window around it
3. **Assign the block** to the mask if the count is larger than a threshold



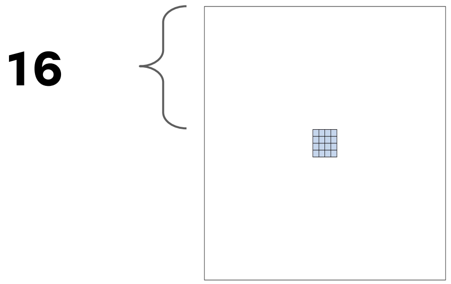
# Step 2: local distribution computation

Gathering local statistics on a **window** around a 4x4 block:

*p(d) = # pixels in the window with value equals to (current pixel_value + d)*

Window size depends on the resolution

Maxim window size:  36x36 around a 4x4 block




This step gets the *p_values* in the window for *d = -max_diff .. max_diff*, where default max_diff=4

Same p_values used for all the pixels in the block

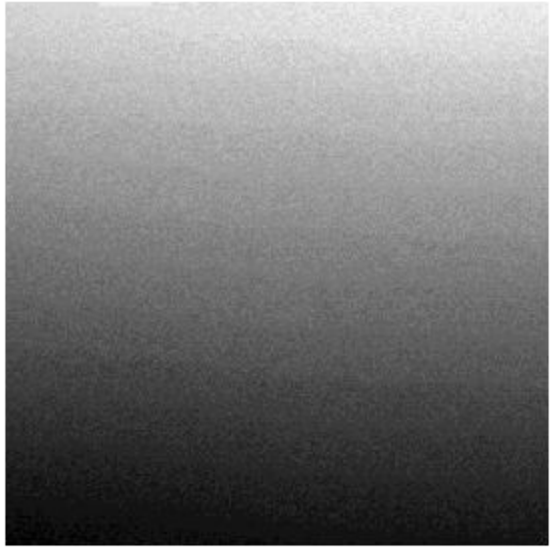
***p(0)*** and  the largest (***p_max***), and  the second largest, (**p_max2**) p(d) with d≠0 are use for dithering

# Step 3: dithering

Dither breaks down the appearance of bands: it is applied to pixels likely to be in large enough bands

$$area\_size\_condition = p(0) > (pixels\_in\_window{>>}3)\ \&\&\ p\_max > (pixels\_in\_window{>>}4)$$

**AVM**

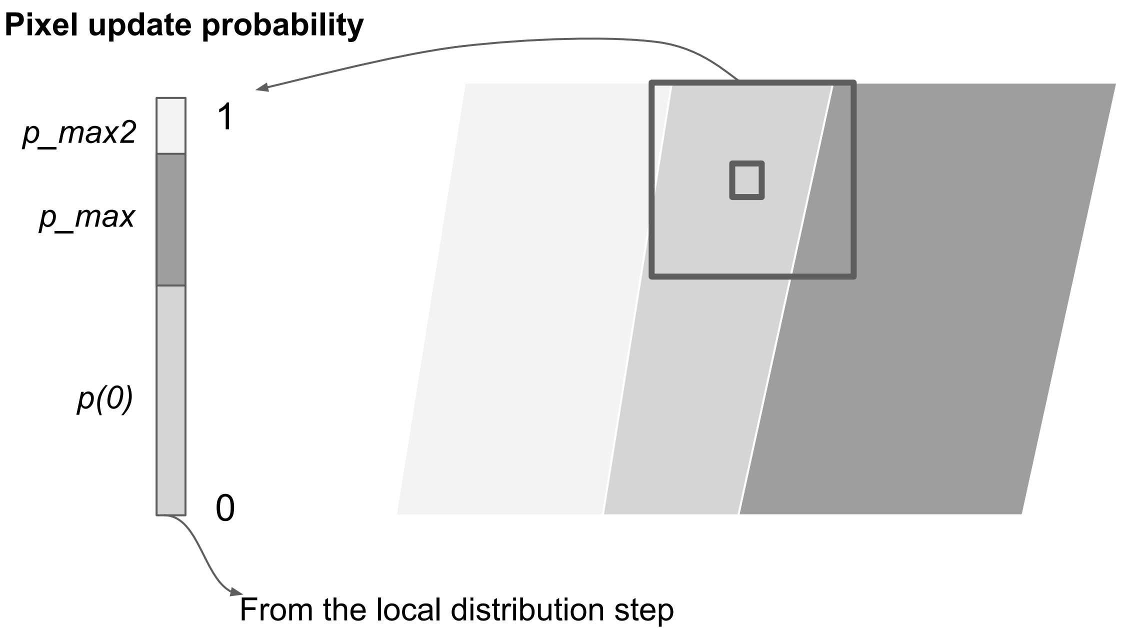


**AVM with CAMDA**

**Note:** **gray-levels stretched for better visibility**

# Dither is added to the 4x4 blocks in the spatial mask

Pixel update probability depends on the computed local distribution.




*pr_range = pseudo_random % (p(0) + p_max + p_max2)*

*if (p(0) + p_max2 <= pr_range)*

*pixel_value = p_max_pixel_value*

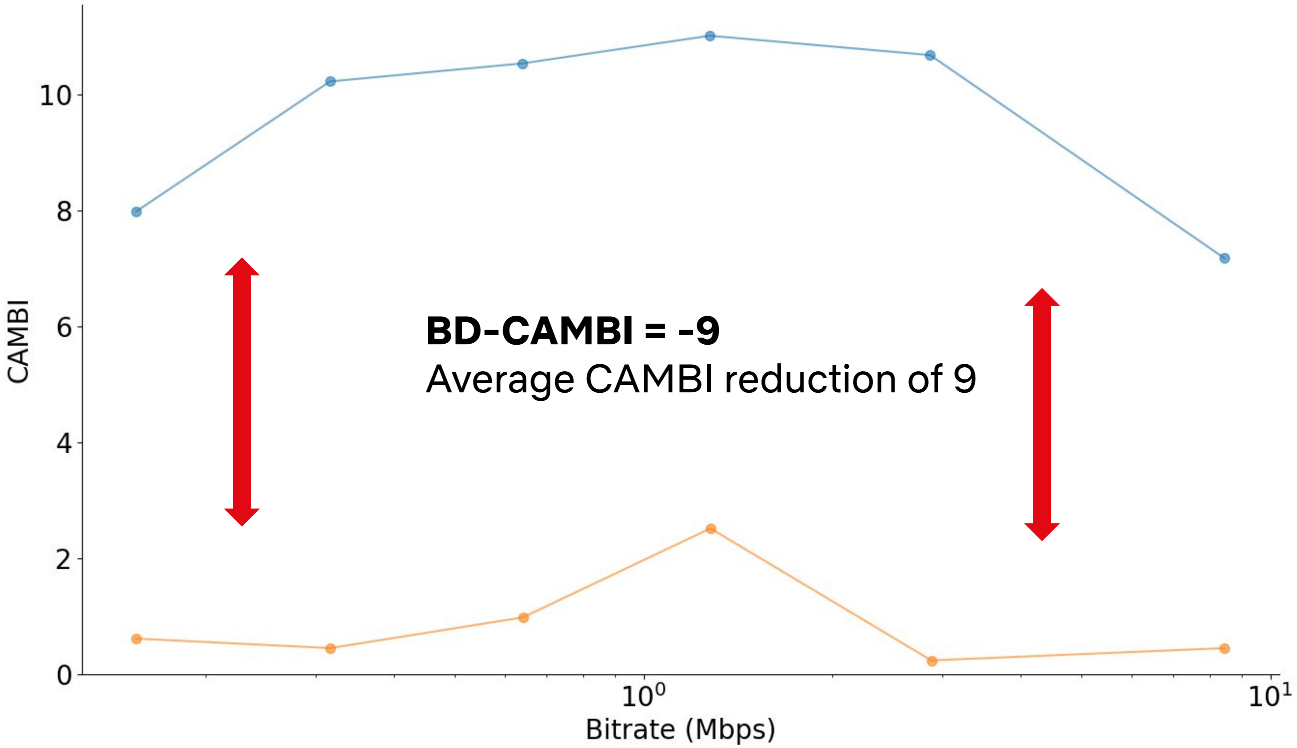
*else if (p(0) <= pr_range)*

*pixel_value = p_max_pixel_value2*

*else pixel_value is not modified*

# Performance

# BD-CAMBI



CAMBI
10
8
6
4
2
0
$10^0$
$10^1$
Bitrate (Mbps)
**BD-CAMBI = -9**
Average CAMBI reduction of 9

# CAMDA only impacts banding-prone sequences

AOM CTC v3, random access configuration, sorted by BD-CAMBI

| Sequence | CAMBI | $VMAF_{BA}$ | $VMAF_{neg}$ | PSNRY | PSNRU | PSNRV | SSIM |
|---|---|---|---|---|---|---|---|
| Johnny_1280x720_60 | **-9.37** | -41.75 | 0.36 | 2.83 | 0.55 | 0.54 | 0.13 |
| Butterfly | **-7.83** | -21.58 | 0.07 | 1 | 0 | 0 | -0.42 |
| Berlin-Fernsehturm | **-7.31** | -25.28 | 0.15 | 2.01 | 0.01 | 0.01 | 1.73 |
| Zoo_de_la_Barben | **-7.19** | -23.66 | 0.04 | 1.93 | 0 | 0 | 1.33 |
| buildings_02 | **-6.96** | -27.2 | 0.02 | 0.24 | 0 | 0 | -7.19 |
| Shaky_Fireworks_3840x2160_2997 | **-6.94** | -41.34 | 0.57 | 8.52 | 2.71 | 0.21 | 4.03 |
| Noise_Animation_1280x720_2398f | **-6.78** | -40.53 | -0.75 | 2.74 | 0.93 | 0.21 | 1.31 |
| landscape_28 | **-6.02** | -23.78 | 0.01 | 0.15 | 0 | 0 | -3.26 |
| KristenAndSara_1280x720_60 | **-5.88** | -26.52 | 1.44 | 3.56 | 0.43 | 0.29 | 4.83 |
| fireworks_01 | **-5.77** | -30.21 | 0.11 | 0.3 | 0 | 0 | -0.04 |
| SnowMountain_640x360_2997 | **-5.55** | -23.16 | 0.2 | 0.2 | 0.08 | -0.06 | 1.94 |
| GlassHalf_1920x1080_24fps_8bit | **-5.42** | -22.95 | -0.95 | 0.58 | 0 | 0.12 | -1.27 |
| Fontaine_Place_Stanislas | **-5.27** | -21.23 | 0.06 | 0.09 | 0.01 | 0.01 | 0.07 |
| Shaky_Baseball_3840x2160_5994f | **-4.71** | -19.63 | -0.46 | 4.88 | 0.27 | -0.15 | 11.43 |
| WITCHER3_1920x1080_60_8bit | **-3.59** | -20.83 | 0.76 | 0.79 | -0.79 | -1.09 | 0.53 |
| *…129 more sequences & still images…* | | | | | | | |
| **Average in-loop** | **-0.82** | **-3.59** | **0.02** | **0.25** | **0.02** | **0.01** | **0.13** |
| | | | | | | | |
| **Average post-loop** | **-0.85** | **-3.63** | **0.02** | **0.28** | **0.02** | **0.01** | **0.13** |

# CAMDA has almost no encoding but some decoding time impact (when there is banding)

Average **encoding** time increase **~0.5%**

Average **decoding** time increase: **~0.5%** (in-loop) and **~1.5%** (post-loop)

| Random access | **% debanded frames** | **Dec Time (SIMD)** | **Dec Time (No SIMD)** |
|---|---:|---:|---:|
| **In-loop** | 2.6% | 4.3% | **0.5%** |
| **Post-loop** | 8.6% | 15.2% | **1.5%** |

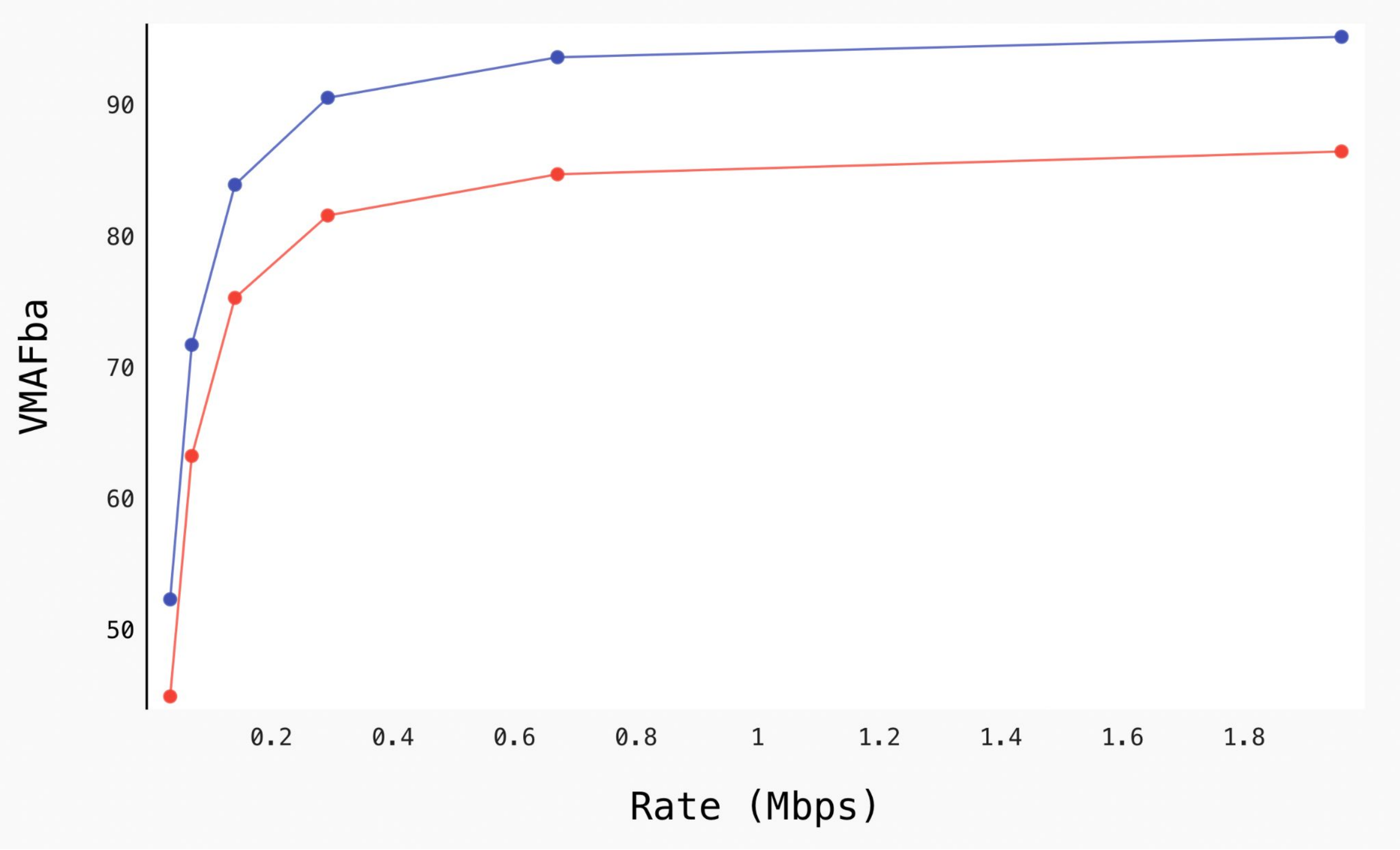
**Worst-case** in-loop decoding time increases (**AVM** with and without SIMD):

| **Sequence** | **BD-CAMBI** | **SIMD Dec Time** | **No SIMD Dec Time** |
|---|---:|---:|---:|
| Johnny_1280x720_60 | -9.37 | 71.7% | 33.6% |
| KristenAndSara_1280x720_60 | -5.88 | 58.5% | 20.5% |
| GlassHalf_1920x1080_24 | -5.42 | 35.9% | 17.6% |
| SnowMountain_640x360_2997 | -5.55 | 24.5% | 17.6% |

# Up to 42% VMAFba BD-rate gain for the CTC mandatory seqs

**VMAFba: Banding-aware VMAF**

"Banding vs. Quality: Perceptual Impact and Objective Assessment", ICIP2022, L. Krasula et al.



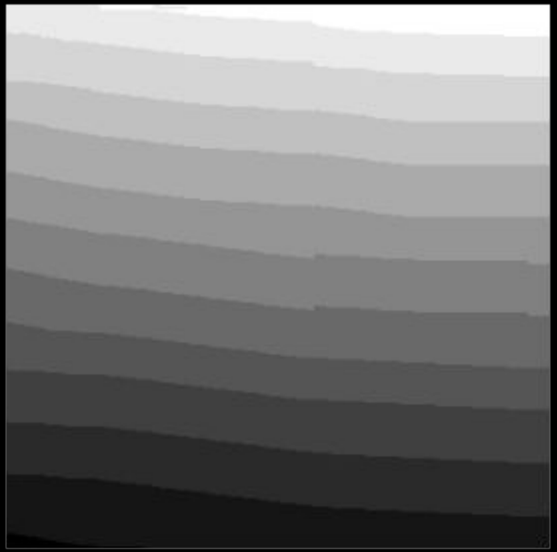

CAMDA

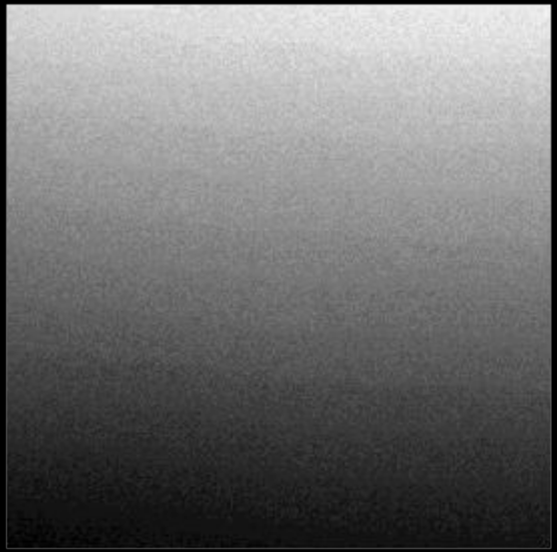
no CAMDA

Johnny 1280x720

Johnny @ QP160

GlassHalf @ QP110



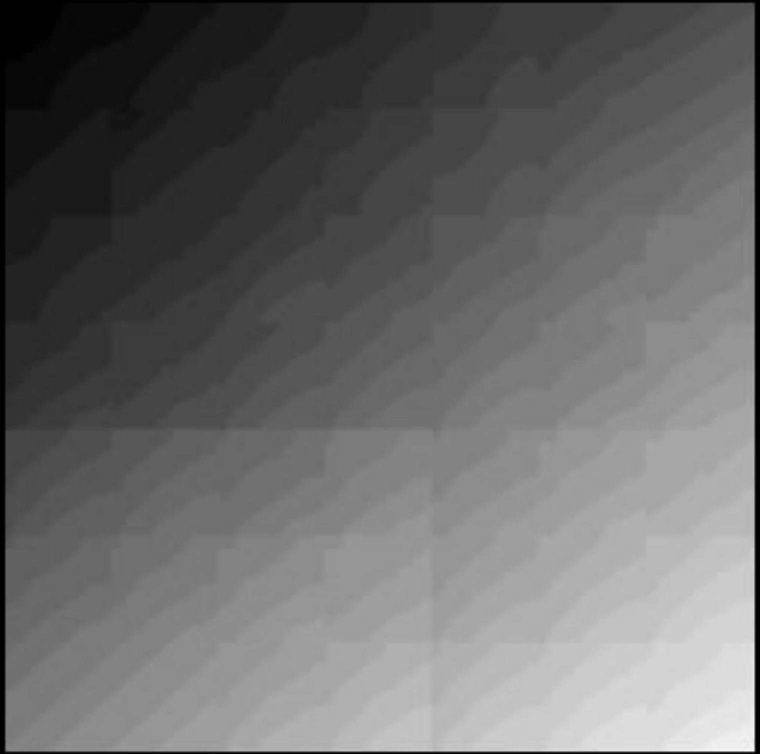


**Note: gray-levels stretched for better visibility**

AVM v3.1

**AVM v3.1 with CAMDA**

## HDR10 content @ QP135

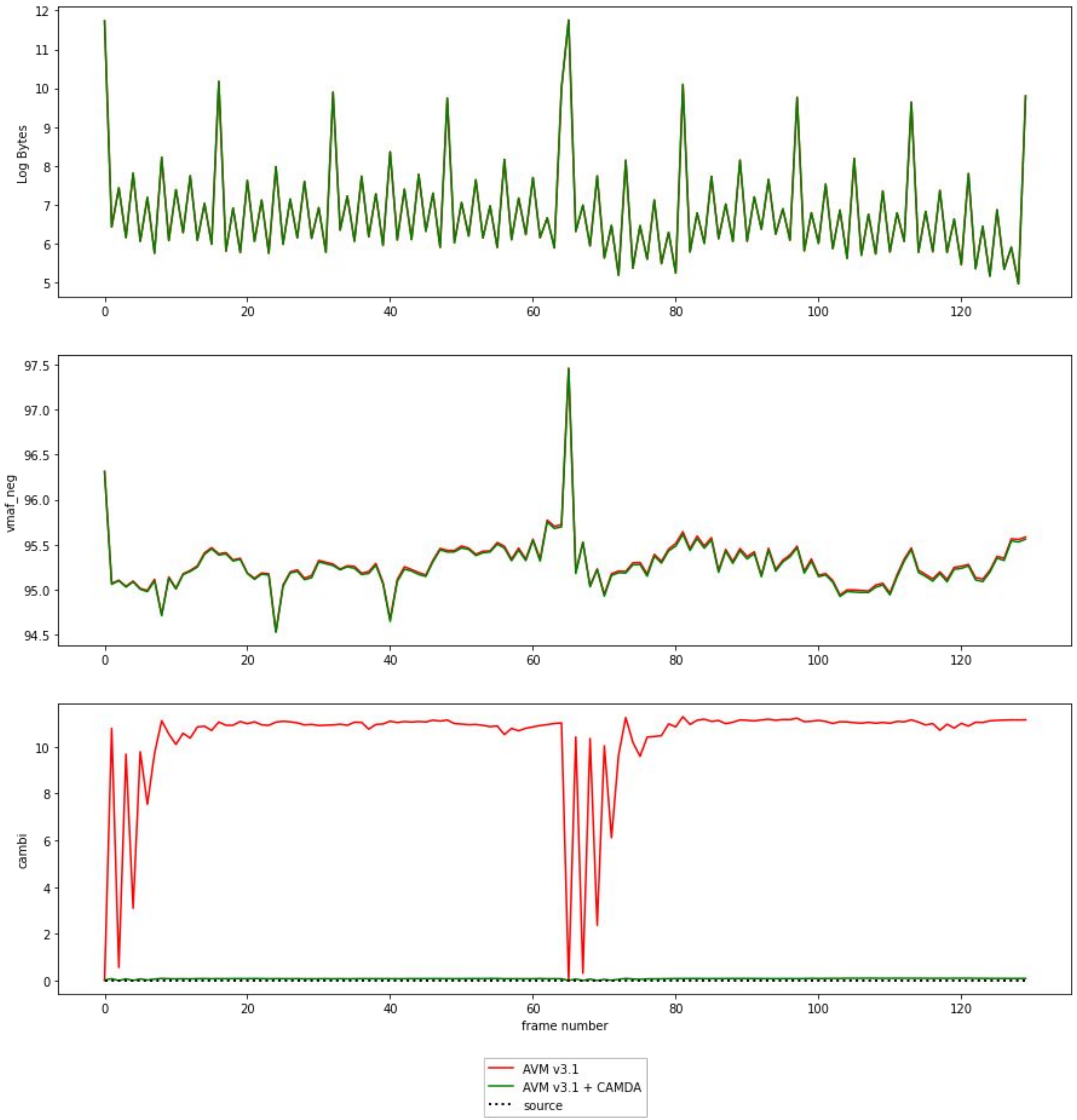


AVM v3.1

**AVM v3.1 with CAMDA**

**Note: gray-levels stretched for better visibility**

# A proper CAMDA encoder can improve the banding effects temporal consistency



# CAMDA addresses an ongoing and commonly found artifact

- CAMDA shows substantial banding reduction in CAMBI and (anecdotal) subjective improvements in both in-loop and post-loop cases

- Encoder and decoder-side complexity are very reasonable.

- CAMDA provides AVM with the functionality to remove bands and the encoder flexibility to apply it on a use-case basis

- On average, the post-loop filter is computationally more complex while having slightly better CAMBI scores and, possibly, being subjectively better.

- Subjective tests to be conducted to fully assess CAMDA benefits
    - They'll provide more evidence on the best position of CAMDA in the AVM filter chain

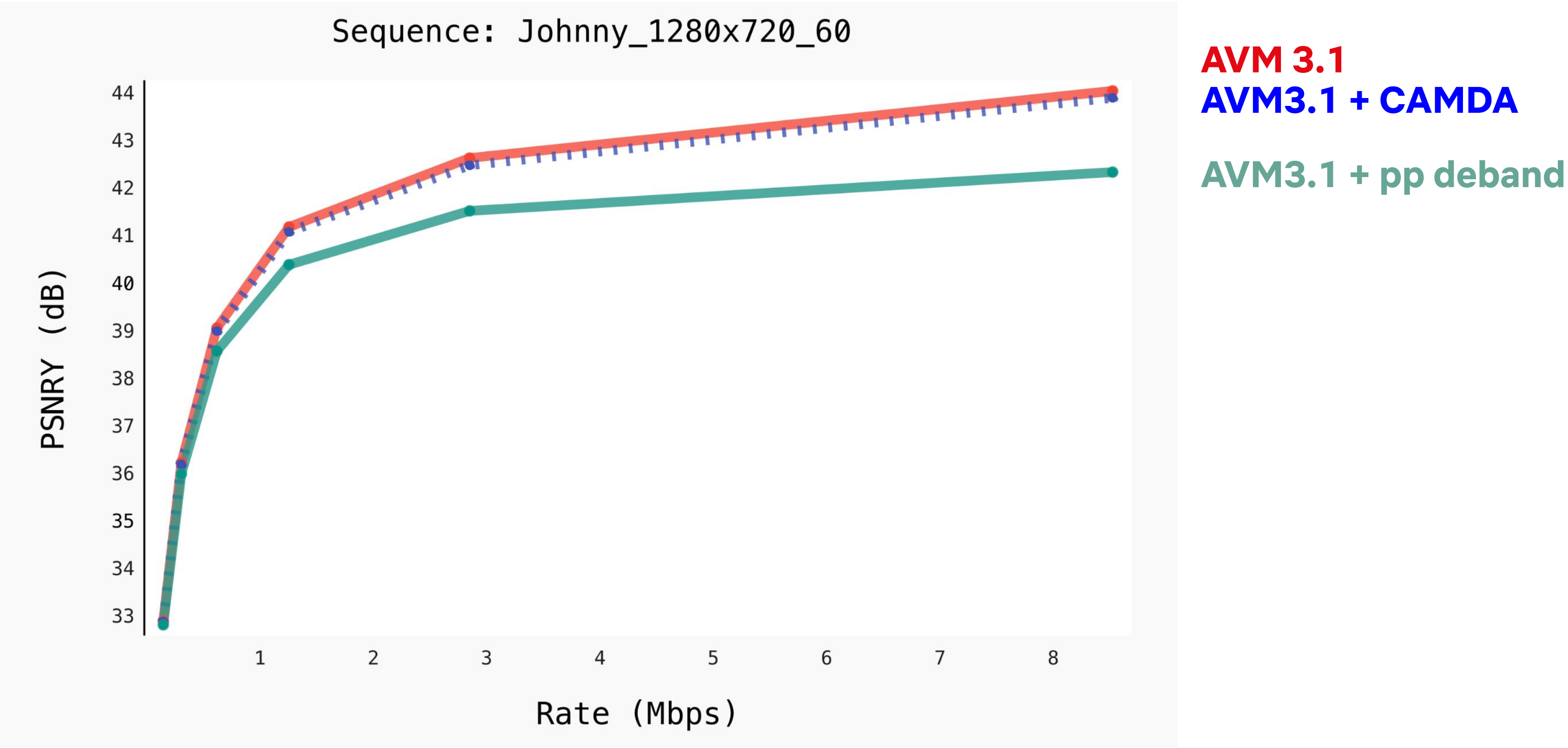
# Questions?

# Comparisons with post-processing filters

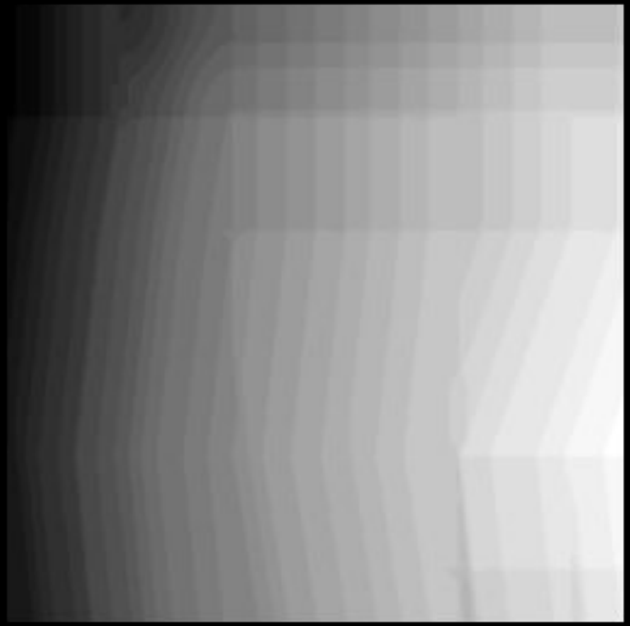
# CAMDA shows large gains vs post-processing debanding

RA config, 8b seqs; gains of CAMDA vs **ffmpeg deband as anchor**:

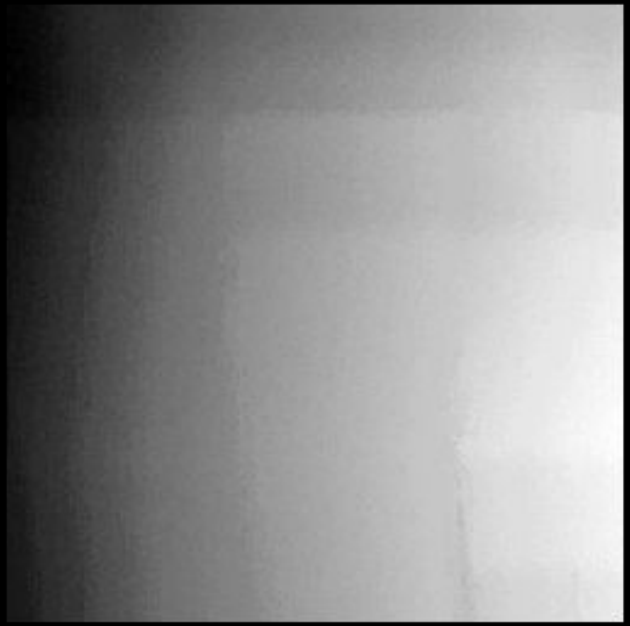
| PSNRY | PSNRU | PSNRV | VMAF | VMAFneg | VMAFba | SSIM | MSSSIM | CAMBI |
|---|---|---|---|---|---|---|---|---|
| -8.91 | -24.11 | -25.34 | -5.94 | -5.06 | -2.3 | -10.32 | -10.64 | 0.52 |



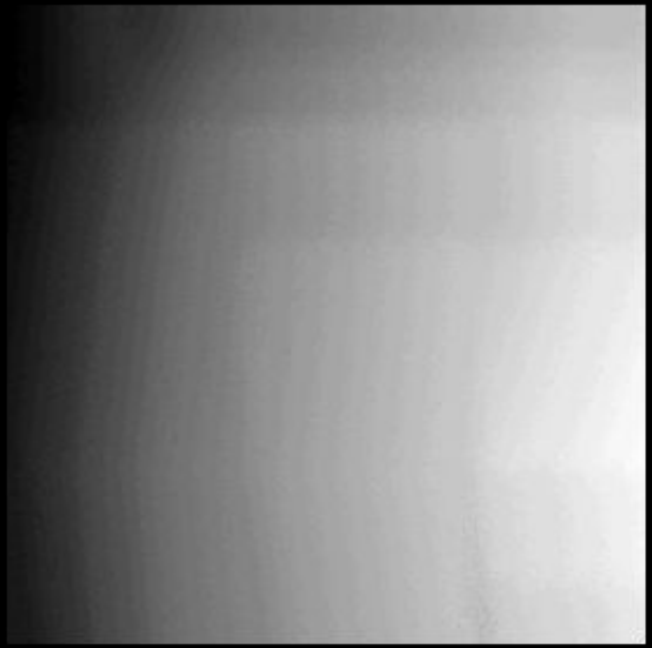
## Johnny @ QP160



AVM v3.1



AVM with CAMDA



AVM v3.1 +
post-processing deband

**Note: gray-levels stretched for better visibility**