# WebRTCon

为开发者赋能　　为行业加速

2018年5月19日-20日 • 上海光大会展中心

主办方：LiveVideoStack 音视频技术社区

# AV1 - The New Open Video Codec

***Zoe Liu*, Google Inc**
**(zoeliu@google.com)**

# Outline

- AOMedia and AV1
- Coding Techniques
- Coding Performance
- What’s Next

# Outline

- ***AOMedia and AV1***
- Coding Techniques
- Coding Performance
- What’s Next

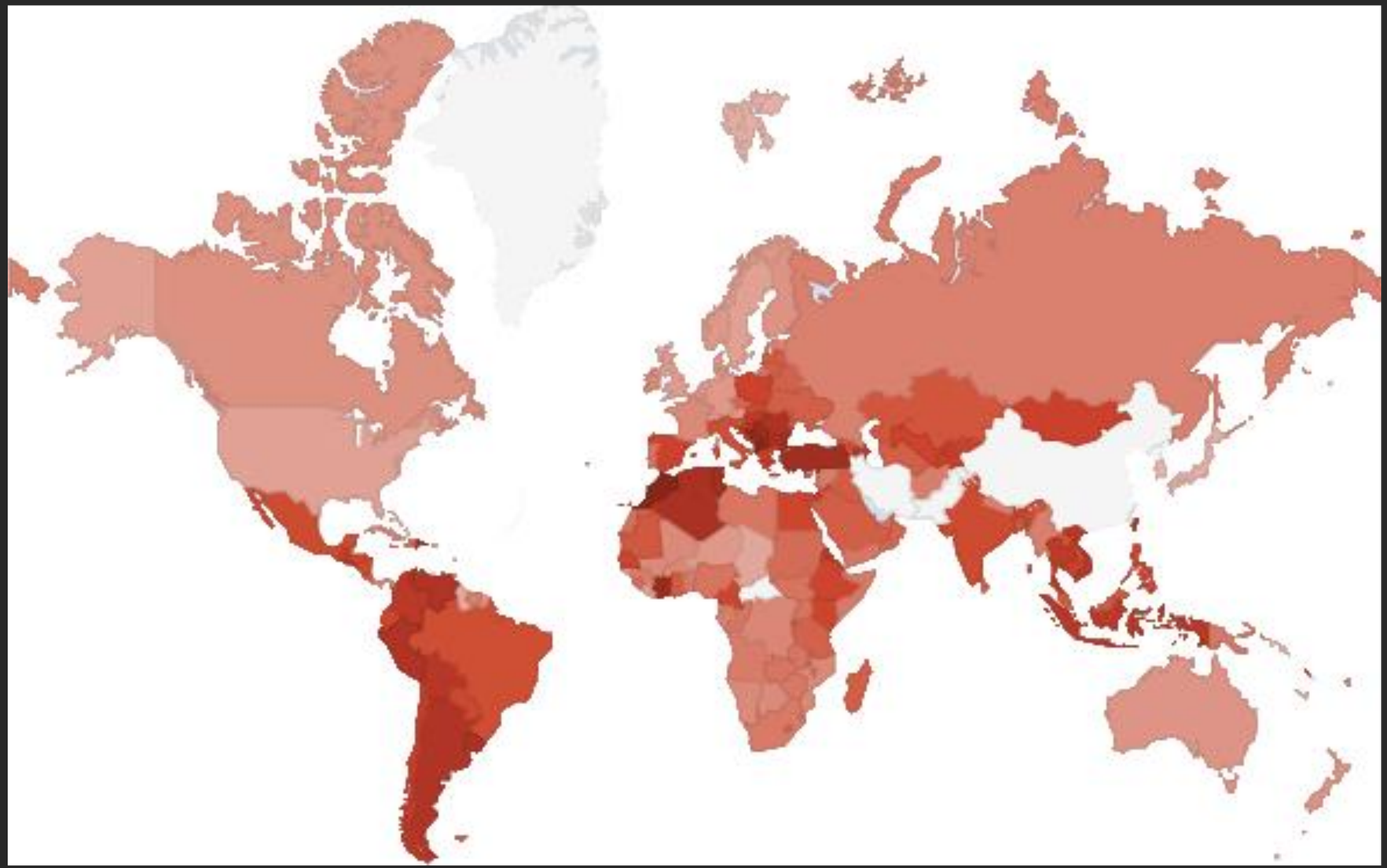
# AOMedia

- **AV1 is the first video codec developed by Alliance for Open Media (AOM)**
- AOM was formed in 2015 Q4 to jointly deliver an open video format because
  - A fast pace of innovation is needed
  - A royalty-free codec is desirable
- 37 members: content providers, web platforms, hardware makers, ...
- Welcome to join!

# AOMedia

- **AV1 is the first video codec developed by Alliance for Open Media (AOM)**
- AOM was formed in 2015 Q4 to jointly deliver an open video format because
  - A fast pace of innovation is needed
  - A royalty-free codec is desirable
- 36 members: content providers, web platforms, hardware makers ...
- Welcome to join!

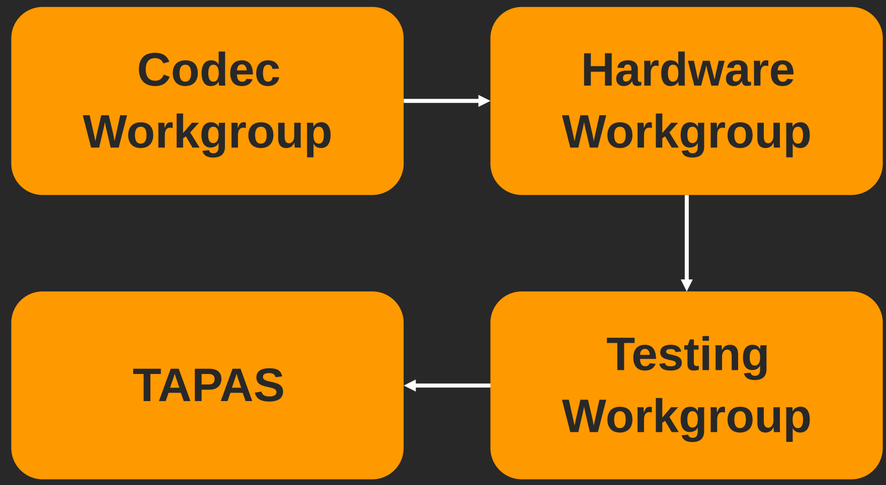
# VP9




YouTube
watch time gains
VP9 vs. H.264


- **The focus of AV1**
  - **Royalty-free**
  - **Open[1] and interoperable**
  - Optimized for **hardware**
  - **Highest quality, real-time video delivery**
  - **Ubiquitous and flexible**

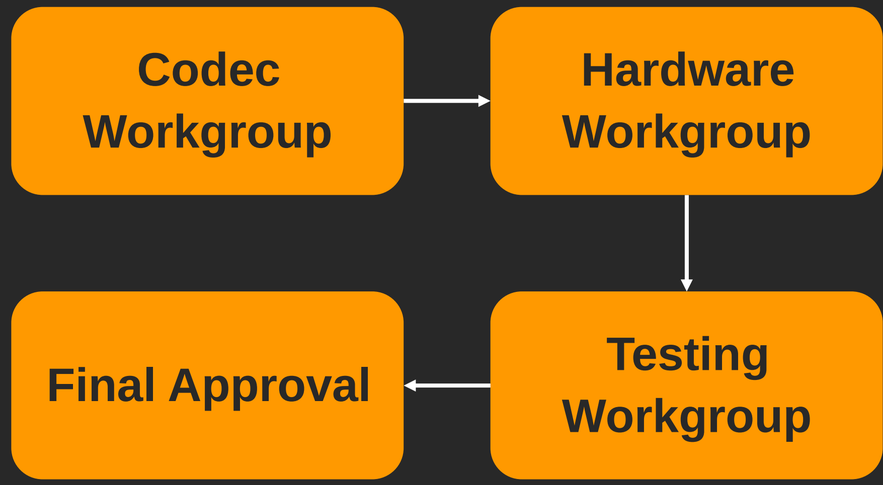
- **AV1 development** starts from an extension of libvpx-VP9 and features in daala and thor
- New tools are proposed and iterated before decisions (adopt/reject) are made




[1] https://aomedia.googlesource.com/


- Each new coding tool has been iterated through workgroups to make sure it is
  - **Beneficial**
  - **Hardware friendly for real-time decoding**
  - **Harmonized with adopted tools**
  - Reviewed to work towards the goal of **royalty-free**

- Now, the bitstream has been frozen, and the spec has been published



WebRTCon
为开发者赋能 为行业加速

# Outline

AOMedia and AV1

***Coding Techniques***

Coding Performance

What’s Next

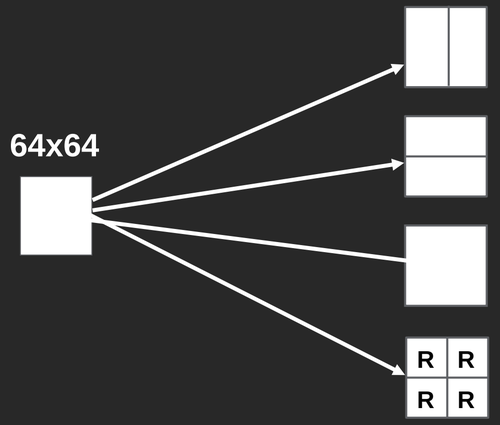
# Video coding at a glance

- Partition
- Prediction
- Transform Coding
- Quantization
- Entropy coding
- Restoration and Post-processing

# Coding Block Partition

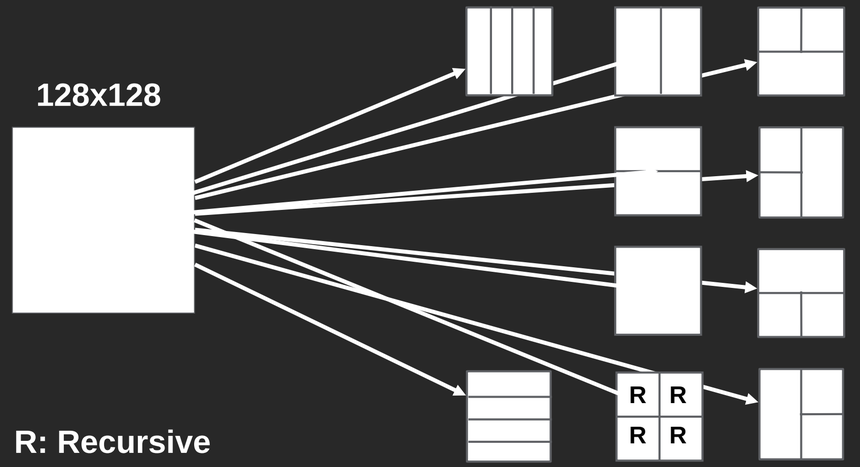
## VP9

Recursive partition
64x64 down to 4x4
4-way partition
Constraints on sub 8x8 blocks




## AV1

Recursive partition
**128x128** down to 4x4
**10-way** partition
**Flexibility** for sub 8x8 blocks



# Prediction

# Prediction

**Intra Prediction** + **Inter Prediction**

# Intra Prediction

## VP9

8 directional prediction modes
2 smooth modes: DC + TM

## AV1

**56** directional prediction modes
**10** smooth modes
**Chroma from luma prediction**
**Color palette coding**
**Intra block copying**

# Intra Prediction

## VP9

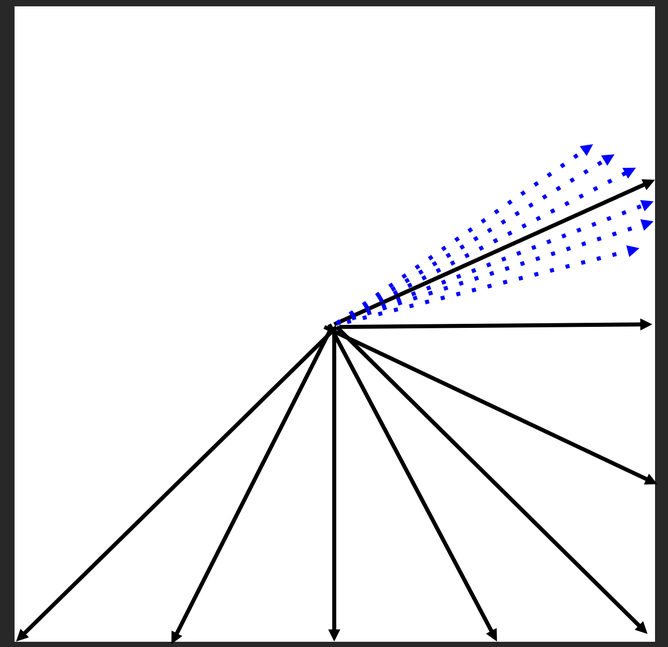
8 directional prediction modes
2 smooth modes: DC + TM

## AV1

**56** directional prediction modes
**10** smooth modes
**Chroma from luma prediction**
**Color palette coding**
**Intra block copying**

# Extended Directional Intra Modes

- On top of VP9’s 8 extrapolation directions, [-3, 3] x 3° angle delta is enabled
- Extended modes are realized by bi-linear interpolation of spatial references



# Intra Prediction

## VP9

8 directional prediction modes
2 smooth modes: DC + TM

## AV1

**56** directional prediction modes
**10** smooth modes
**Chroma from luma prediction**
**Color palette coding**
**Intra block copying**

# Smooth Intra Prediction Modes

## Smooth H/V/AVG

- Approximate bottom and right boundaries

- Quadratic interpolation horizontally, or vertically, or taking the average

## Paeth Predictor

- Copy one from the top, left or topleft edge references, whichever is closest to (top + left - topleft)

- Adopt the reference from the direction with the lower gradient

## Recursive Intra Filter

- Predict in batches of 4x2 blocks by recursively applying 8 7-tap filters to (4+2+1) predicted neighbors

- 5 sets of filters are pre-designed to capture evolutions

# Intra Prediction

## VP9

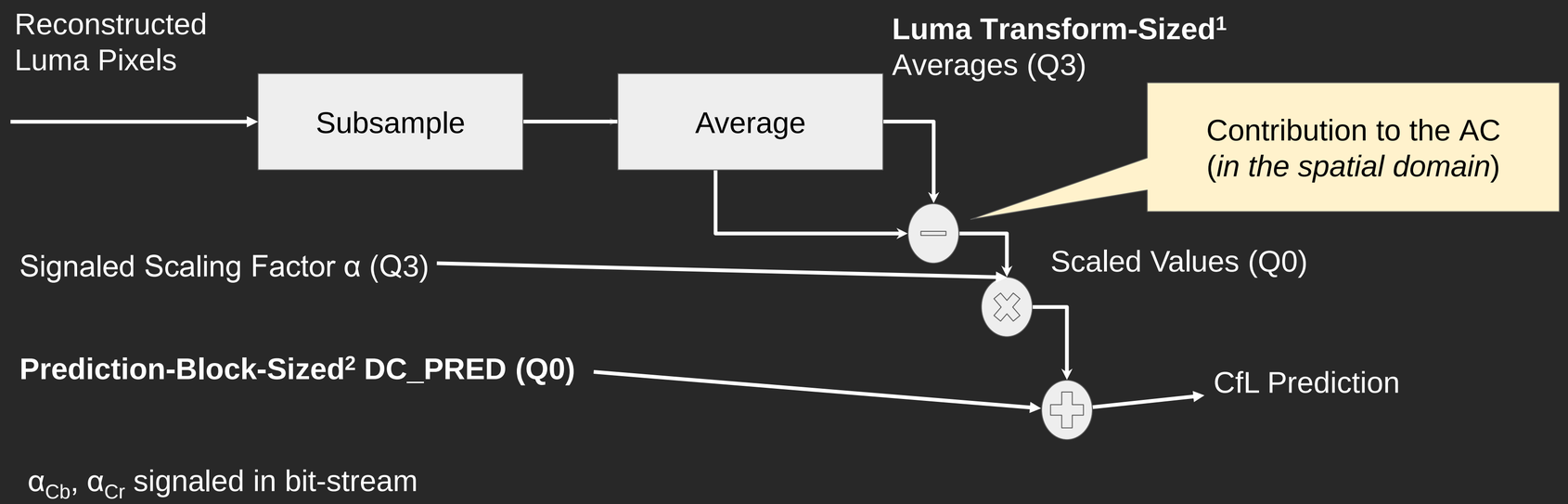
8 directional prediction modes
2 smooth modes: DC + TM

## AV1

**56** directional prediction modes
**10** smooth modes
**Chroma from luma prediction**
**Color palette coding**
**Intra block copying**

# Chroma from Luma Prediction

- Predict Chroma AC from subsampled Luma AC
- Coefficients for linear prediction are conveyed in the bitstream




1. Luma average computed over the luma transform block
2. Chroma DC_PRED computed over prediction block

# Intra Prediction

## VP9

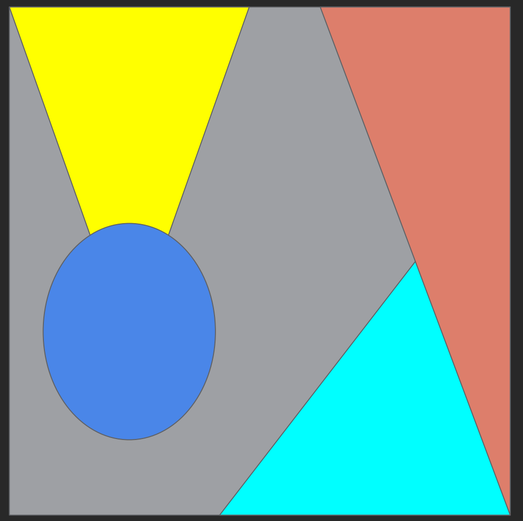
8 directional prediction modes
2 smooth modes: DC + TM

## AV1

**56** directional prediction modes
**10** smooth modes
**Chroma from luma prediction**
**Color palette coding**
**Intra block copying**

# Palette Mode

- Blocks can be approximated by a few unique colors
- A screen-content coding tool as a general code mode
- Palette representation
  - Encode k (2~8) *anchor base* colors, delta coded
  - Encode a k-ary color index map, entropy coded



# Intra Block Copying

- Another screen-content coding tool
- AV1 intra coder can refer to previously reconstructed blocks in the same frame by signaling an **intra frame motion vector**
- Particularly helpful in screen-shots with a lot of texts

# Prediction

**Intra Prediction** + **Inter Prediction**

# Inter Prediction

## VP9

3 reference frames
Fixed MV pred
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp.

# Inter Prediction

## VP9

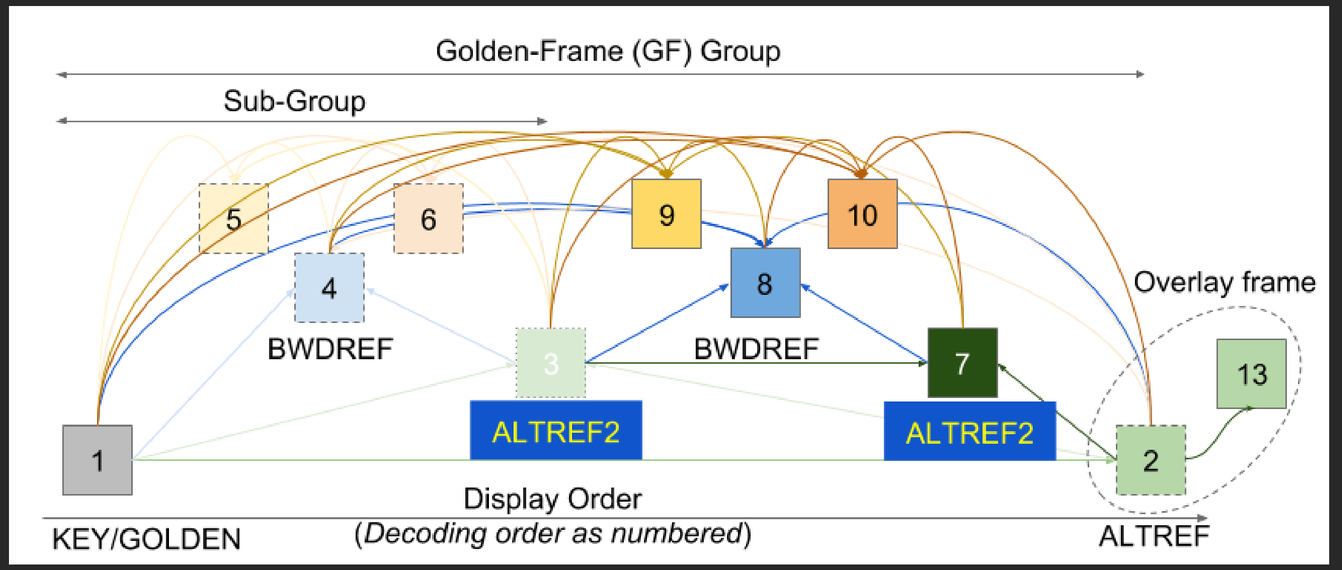
3 reference frames
Fixed MV pred
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp.

# Extended Reference Frames

- Candidate pool: **golden, last, *alt*** (VP9) + **last2, last3, *bwd*, *alt2***
  - Past frames: golden + last3, last2, last
  - Future frames: bwd + alt2, alt (temporal filtered)
- A coding block can pick 1 (single pred) or 2 (compound pred) frames
- AV1 compound prediction can be **bi-directional** (VP9) or **uni-directional**



# Inter Prediction

## VP9

3 reference frames
Fixed spatio MV pred.
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp.

# Motion Vector Prediction

- For each reference frame (or pair), a MV candidate pool is constructed

**Spatial MV Pred**

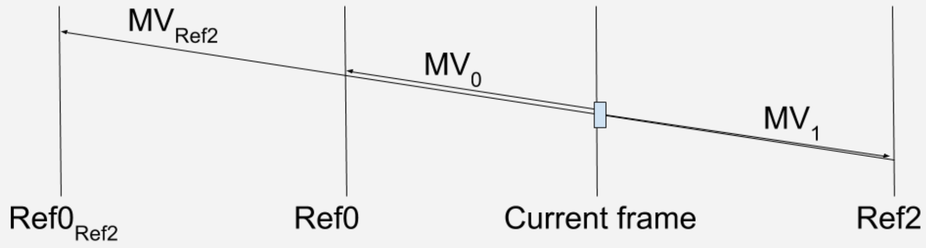
-MVs of neighbors using the same ref frame(s) are added to the pool

-Compared to VP9:
A **deeper** neighborhood is searched.
Building **separate** pools for compound pairs.

**Temporal MV Pred**

-Temporal MV candidates are computed from motion trajectories passing through the current block

-Capable of tracking motion at various velocities



# Dynamic Motion Vector Referencing

- VP9 only considers 2 MV candidates pulled from a fixed searching order
- In AV1, spatial and temporal MV candidates are **prepared**, **scored**, **merged** and **ranked**
- AV1 supports up to 4 candidates
- Index coding is also adaptive to the size of the pool

# Inter Prediction

## VP9

3 reference frames
Fixed MV pred
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp.

# Inter Prediction

## VP9

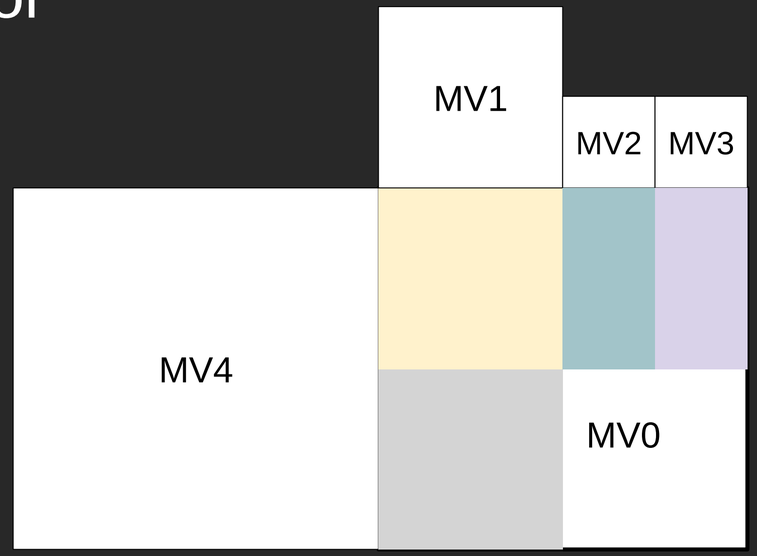
3 reference frames
Fixed MV pred.
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp.

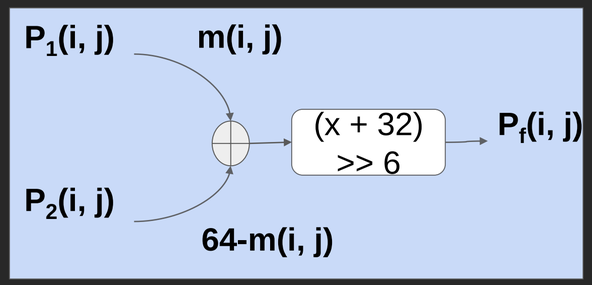
# Overlapped Block Motion Compensation

- Block motion compensation only uses the assigned MV
- OBMC creates secondary predictions from neighbors' MVs, and blend them with BMC to mitigate the effect of discontinued motion field
- AV1 OBMC is a 2-sided causal overlapped predictor
  - Overlapping is operated in the top/left halves
  - Uses predefined 1-D smooth filters
  - Same memory bandwidth as compound pred.

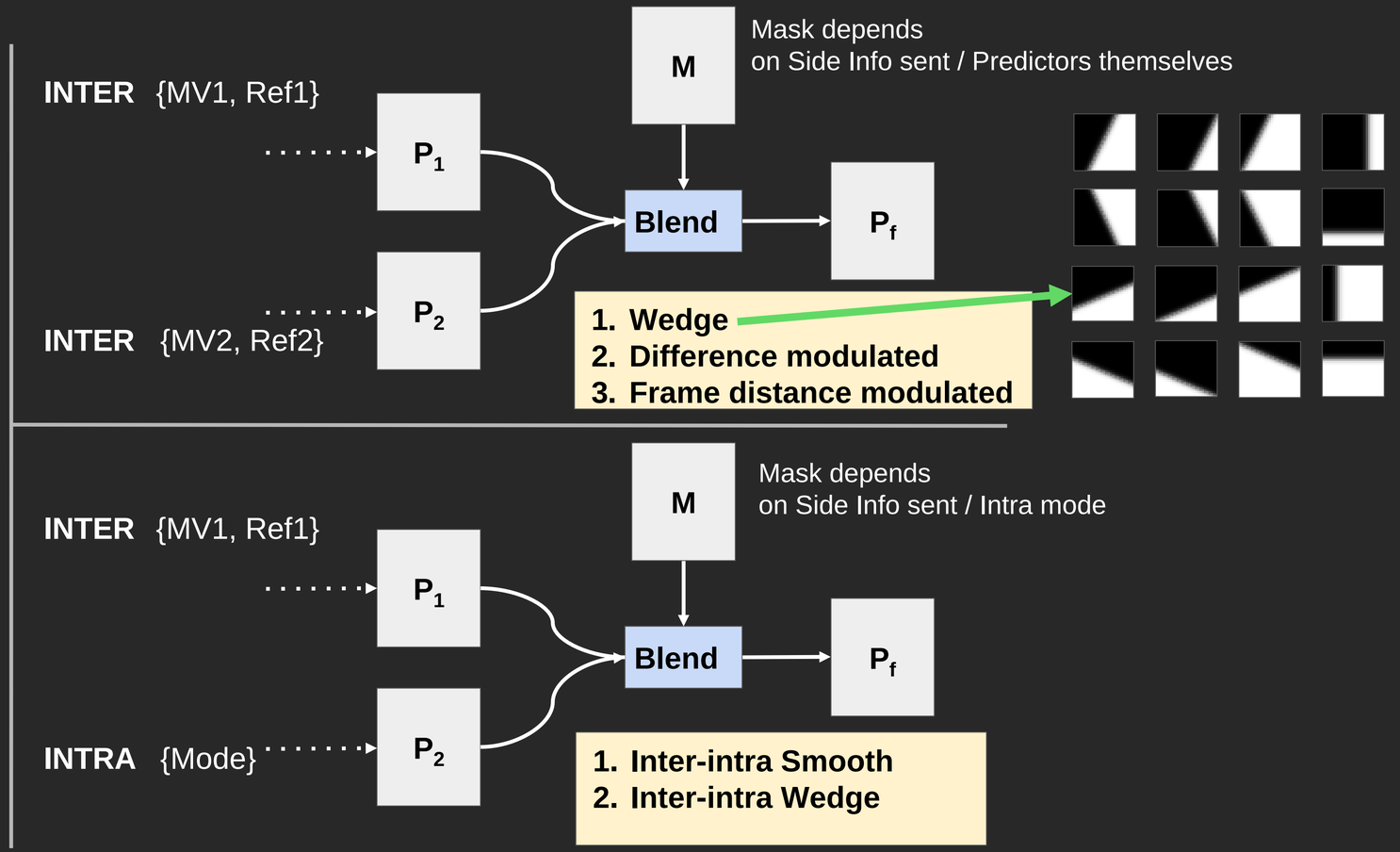


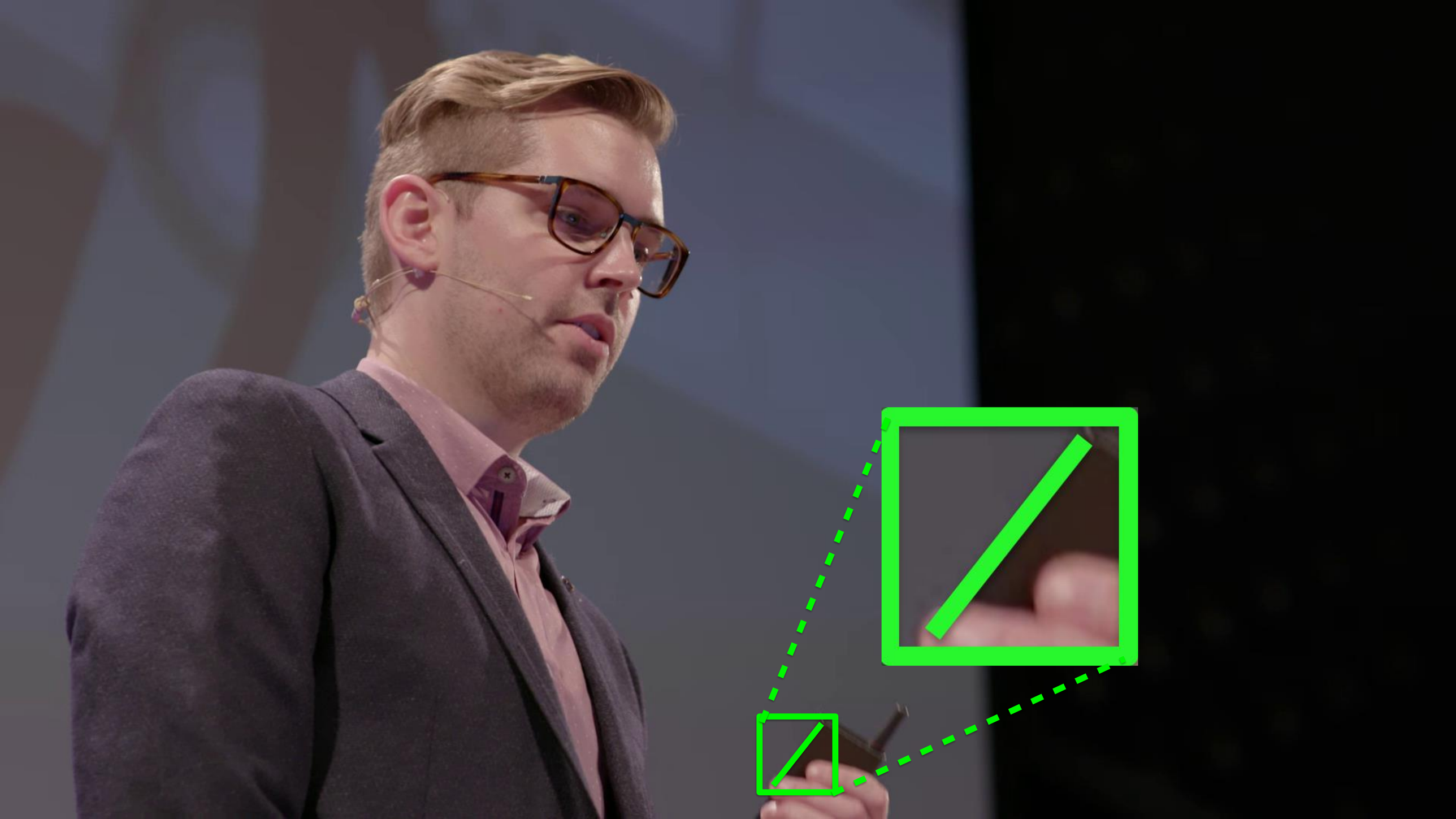
# Masked Compound Prediction

- AV1 masked compound prediction framework enables **per-pixel blending**




Integerized mask m(i, j) ∈ [0, 64]





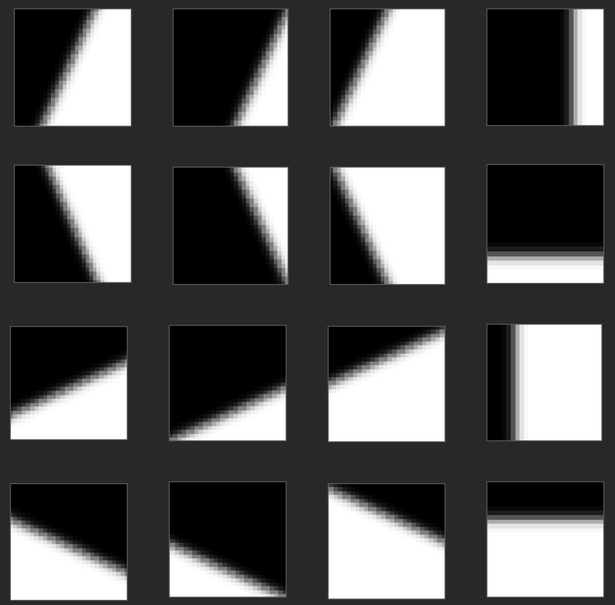
# Wedge Based Inter Prediction

Wedge codebook:

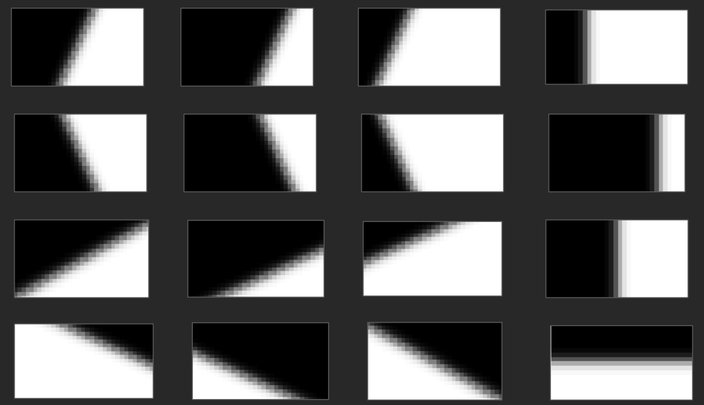
Inter-Inter
4-bit shape
1-bit sign

---

5 bits total



Square codebook



Rectangular codebook

Used for 8x8 up to 32x32 sizes (32x16, 16x8, etc.)

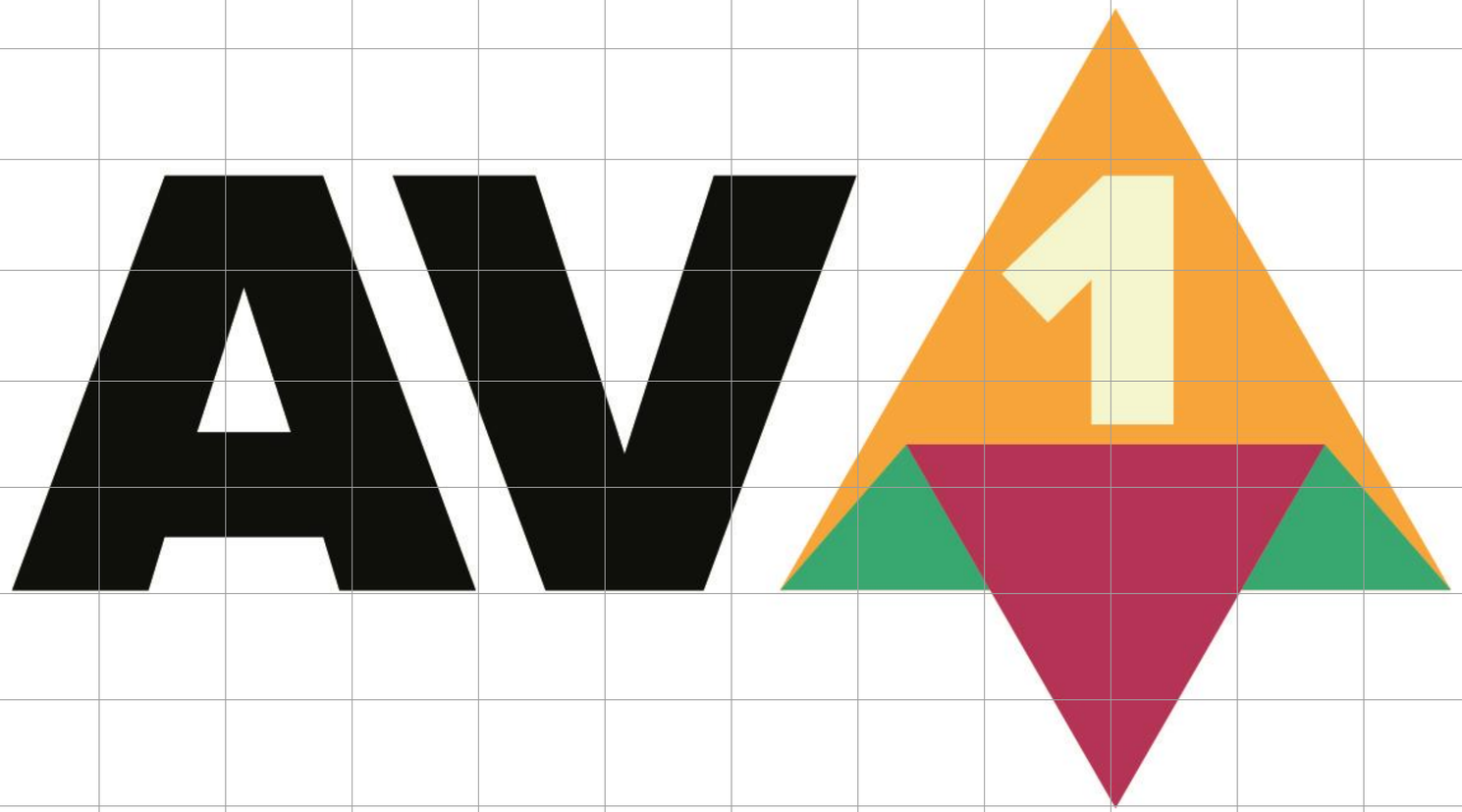
# Inter Prediction

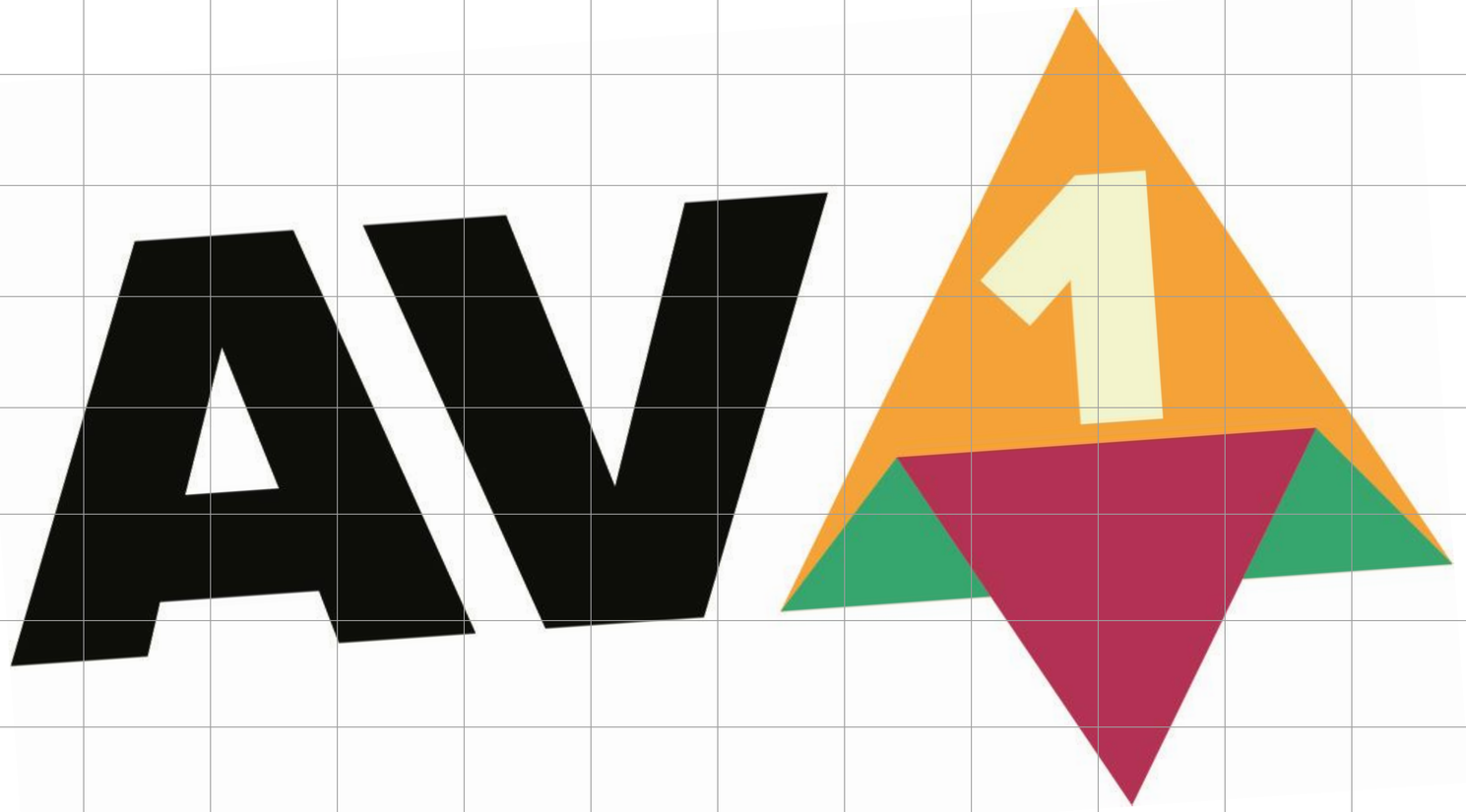
## VP9

3 reference frames
Fixed MV pred.
Same H/V interpolation filter
Block-based prediction
Averaged compound prediction
Only translational motion comp.

## AV1

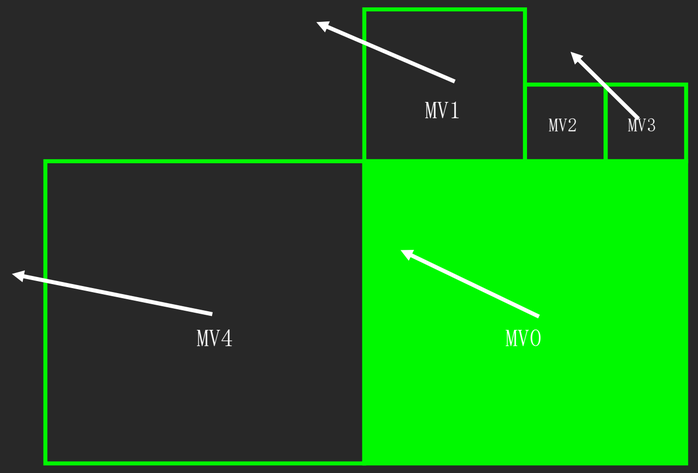
**7** reference frames
**Dynamic spatio+temp.** MV pred.
**Separate** H/V interpolation filters
Support **overlapped** block prediction
Support **masked** compound modes
Support **warped** motion comp





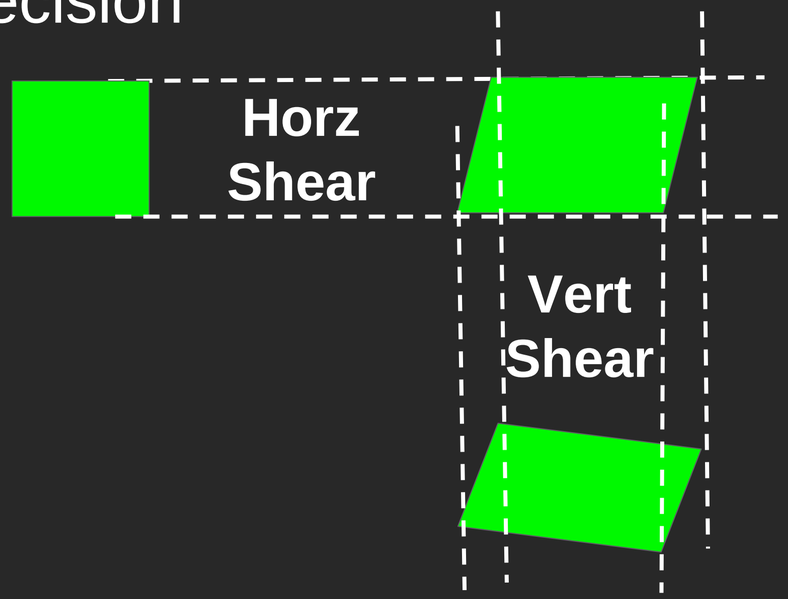
# Warped Motion Compensation

- AV1 warped model: affine or simpler, only small warping allowed
- Advantage of small warping
  - Can be implemented with two shears like separable sub-pel interpolation
  - Processed in 8x8 patches at 1/64 sub-pel precision




Blocks 1, 3, 4 have same reference as current block

AV1 Affine



# Warped Motion Compensation

- AV1 support two warped prediction modes: global warping, or local warping
  - They are picked at block-level
- **Global warping: model is estimated from source and conveyed**
  - Works great with camera motions like zoom, panning, and rotation
- **Local warping: model is estimated from conveyed neighborhood MVs**
  - Low-complexity 4 parameter least square is solved at decoder side

# Transform (TX) Coding

## VP9

DCT only for inter
Hybrid DCT/ADST for intra
32x32 to 4x4 square TX
Fixed-grid TX partitioning

## AV1

**16** hybrid TX kernels for both **inter** and **intra**
**64x64** to 4x4 **square+rect** TX
**Recursive** TX partitioning

# Transform (TX) Coding

## VP9

DCT only for inter
Hybrid DCT/ADST for intra
32x32 to 4x4 square TX
Fixed-grid TX partitioning

## AV1

**16** hybrid TX kernels for both **inter** and **intra**
**64x64** to 4x4 **square+rect** TX
**Recursive** TX partitioning

# Transform Kernels

- **16** separable 2-D kernels: { **DCT**, **ADST**, **fADST**, **IDTX** }$^2$
  - **DCT**: works great in general
  - **ADST** and **flip-ADST**: capture monotonic changes in residual energy
  - **IDTX**: no transform, great for sharp edges
- The kernel sets are gradually reduced as TX sizes increase because some kernels act similarly as sizes get larger

# Transform (TX) Coding

## VP9

DCT only for inter
Hybrid DCT/ADST for intra
32x32 to 4x4 square TX
Fixed-grid TX partitioning

## AV1

**16** hybrid TX kernels for both **inter** and **intra**
**64x64** to 4x4 **square+rect** TX
**Recursive** TX partitioning

# Transform Block Partitioning

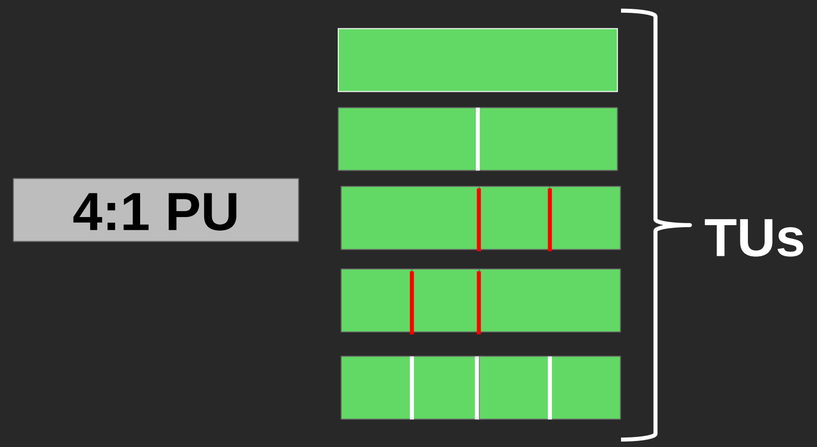
- Support 1:2/2:1 and 1:4/4:1 transform sizes
- Support flexible recursive TX partition in inter PUs

**VP9**

32x32
16x16
8x8
4x4

**AV1**

64x64, 64x32, 32x64, 64x16, 16x64
32x32, 32x16, 16x32, 32x8, 8x32
16x16, 16x8, 8x16, 16x4, 4x16
8x8, 8x4, 4x8
4x4



# Entropy Coding

## VP9

Frame-to-frame adaptive binary arithmetic coder

TX coefficients are coded with uniform context models

## AV1

**Symbol-to-symbol** adaptive **multi-symbol** arithmetic coder

**Level-map** based coefficient coding

# Entropy Coding

## VP9

Frame-to-frame adaptive binary arithmetic coder

TX coefficients are coded with uniform context models

## AV1

**Symbol-to-symbol** adaptive **multi-symbol** arithmetic coder

**Level-map** based coefficient coding

# AV1 Symbol Coding

- Most syntax elements have non-binary long alphabets
- AV1 multi-symbol arithmetic coder facilitates **high throughput** symbol coding and straightforward probability **model adaptation**
  - AV1 arithmetic coding is based on **15-bit CDF** tables
  - CDFs are **tracked** and **updated symbol-to-symbol**

# Entropy Coding

## VP9

Frame-to-frame adaptive binary arithmetic coder

TX coefficients are coded with uniform context models

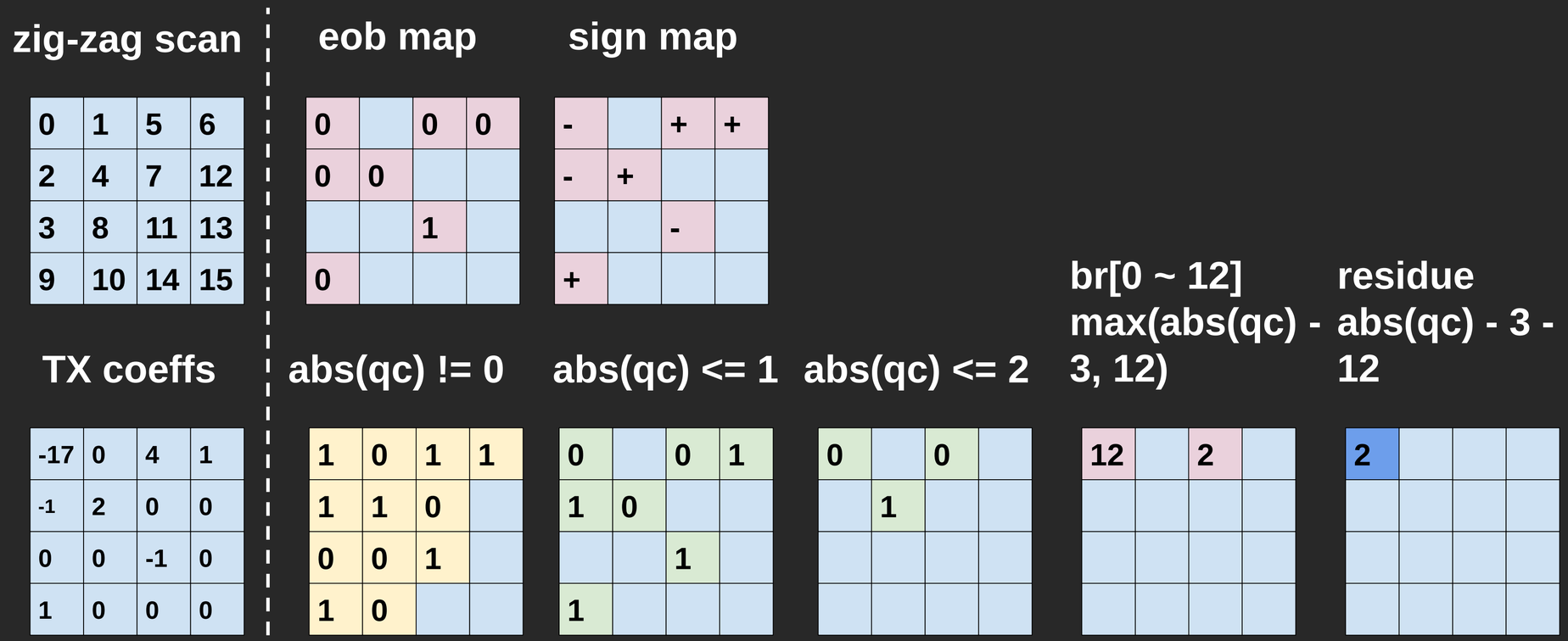
## AV1

**Symbol-to-symbol** adaptive **multi-symbol** arithmetic coder

**Level-map** based coefficient coding

# Level-map TX Coefficient Coding

- TX coefficients are **decomposed into level planes** and coded separately
- Lower levels cost the majority of rates so richer contexts are utilized



# Restoration and Post-processing

## VP9

In loop deblocking filters

## AV1

In loop deblocking filters with
**adaptive strengths**
**Directional enhancement filter**
**Loop restoration filter**
**Frame super resolution**
**Film grain synthesis**

# Restoration and Post-processing

## VP9

In loop deblocking filters

## AV1

In loop deblocking filters with **adaptive strengths**

**Directional enhancement filter**

**Loop restoration filter**

**Frame super resolution**

**Film grain synthesis**

# Restoration and Post-processing

## VP9

In loop deblocking filters

## AV1

In loop deblocking filters with **adaptive strengths**

**Directional enhancement filter**

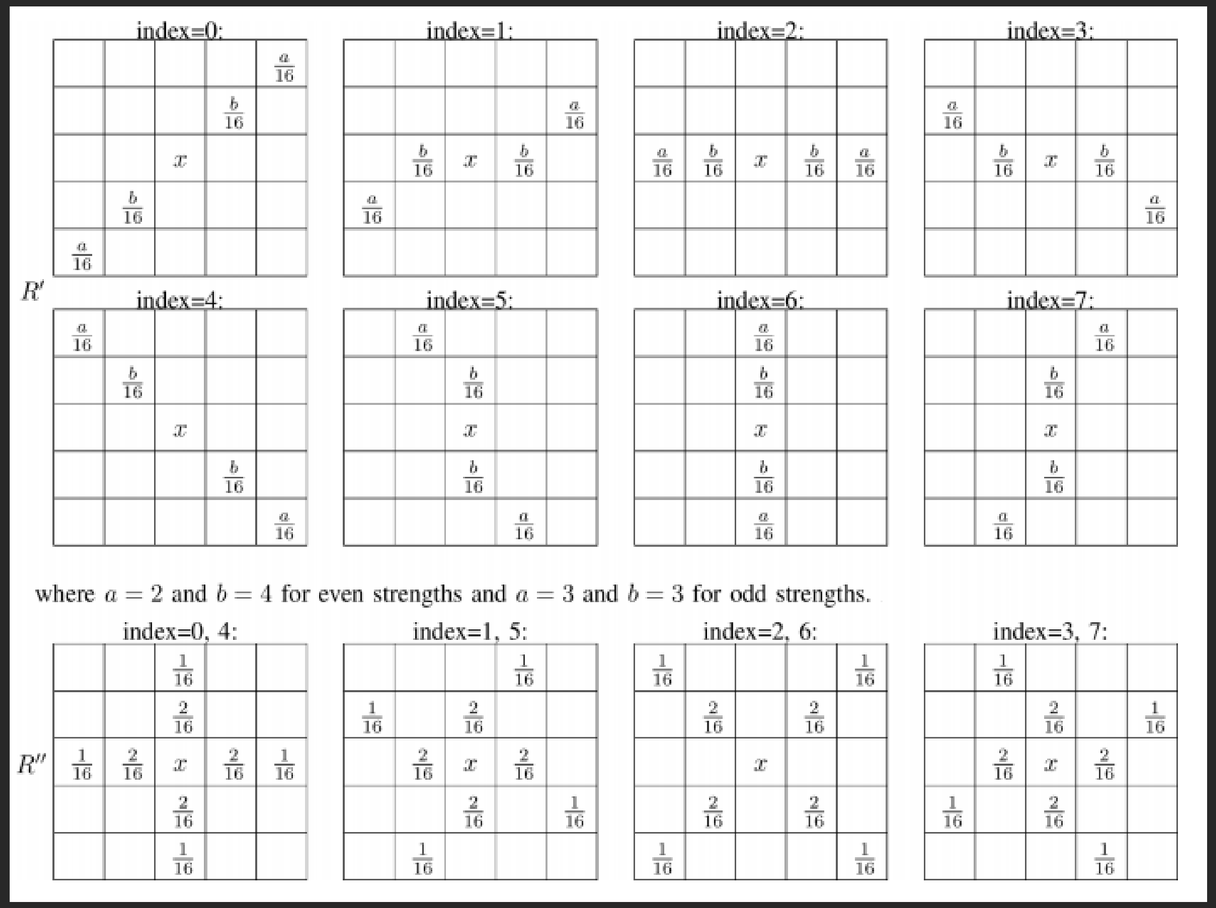
**Loop restoration filter**

**Frame super resolution**

**Film grain synthesis**

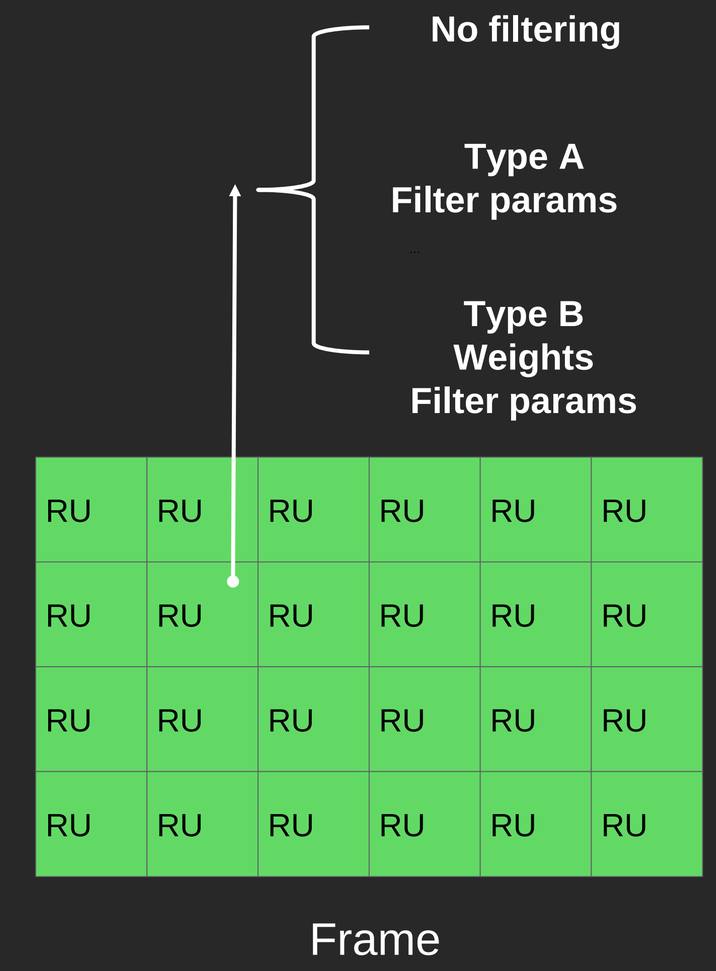
# Constrained Dire. Enhancement Filtering

- Applied after deblocking
- Edge directions are estimated at 8x8 block level
- 5x5 pre-designed detail-preserving deringing filters are applied



# In-loop restoration Filters

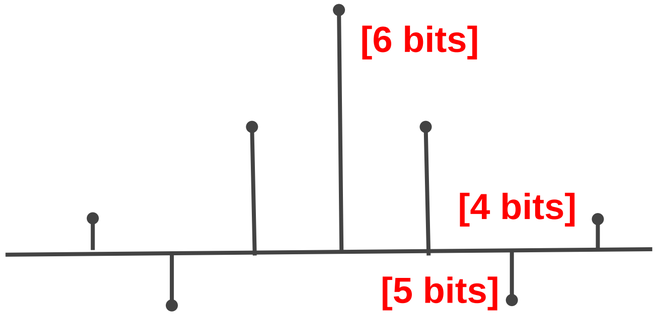
- Applied after CDEF
- By comparing reconstruction with original video, restoration filters are computed and conveyed
  - Switchable filter at RU (128x128) level
  - Type A: Wiener filter
  - Type B: Linear combination of two cheap edge-preserving filters



# In-loop restoration Filters

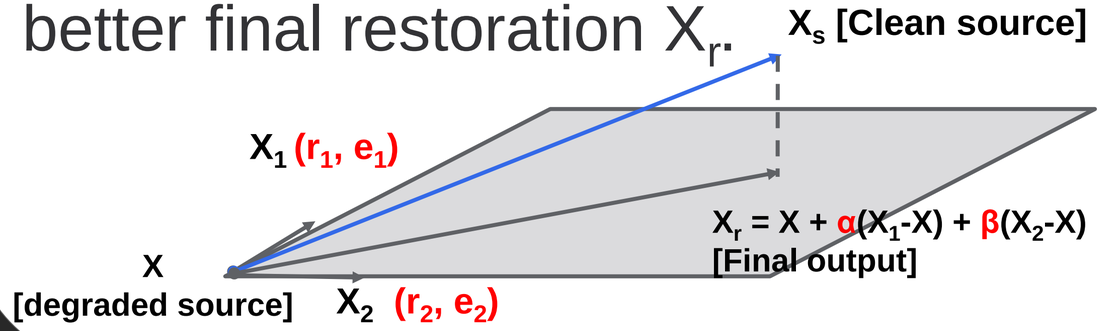
## Type A: Wiener filter

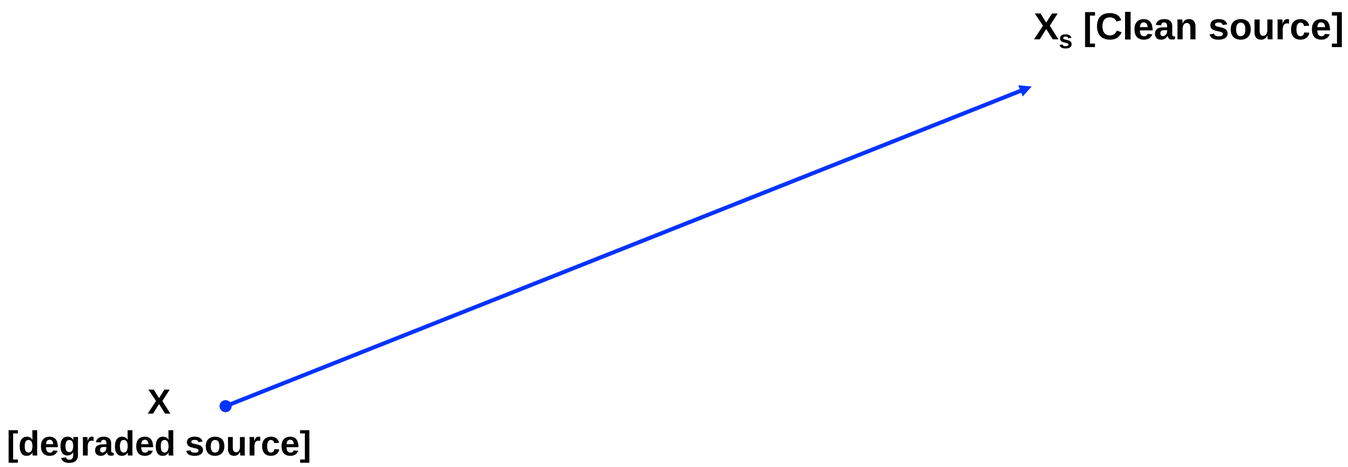
Separable (horz + vert filter)
7-tap, symmetric, normalized

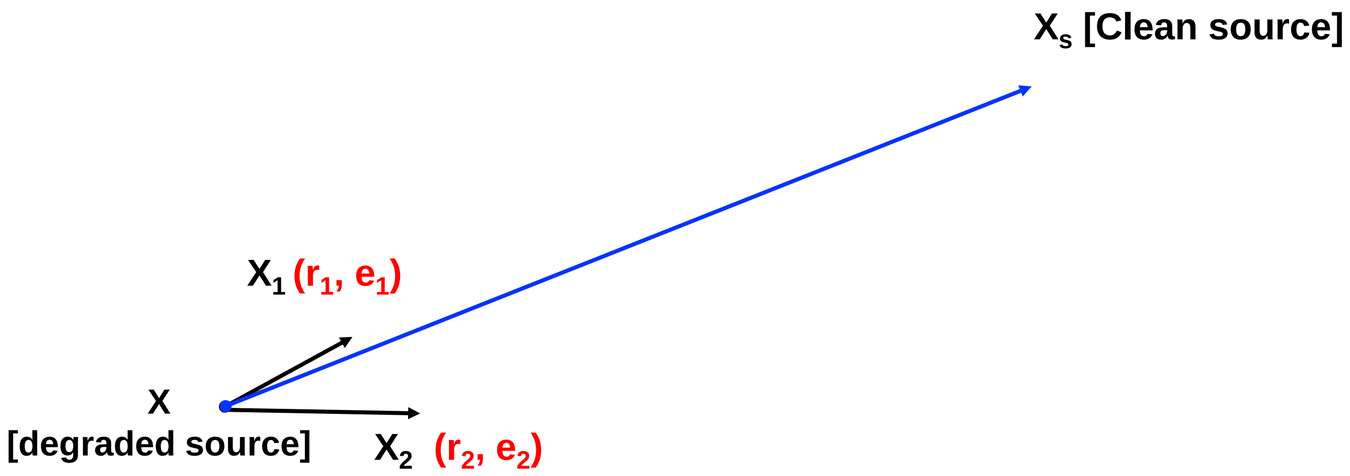


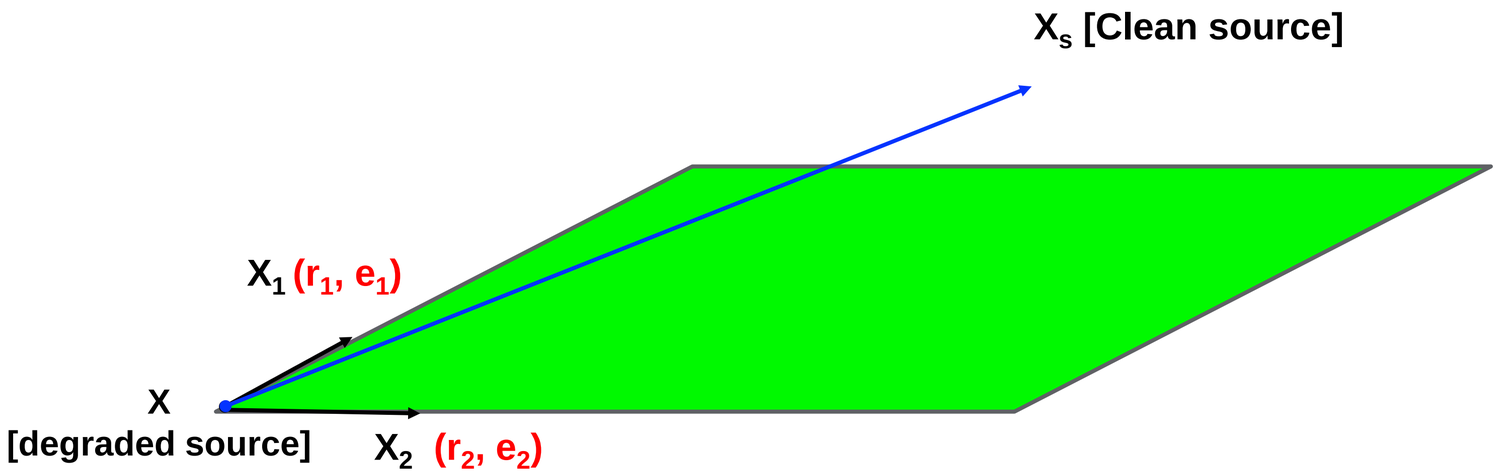

## Type B: Self-guided projected filters

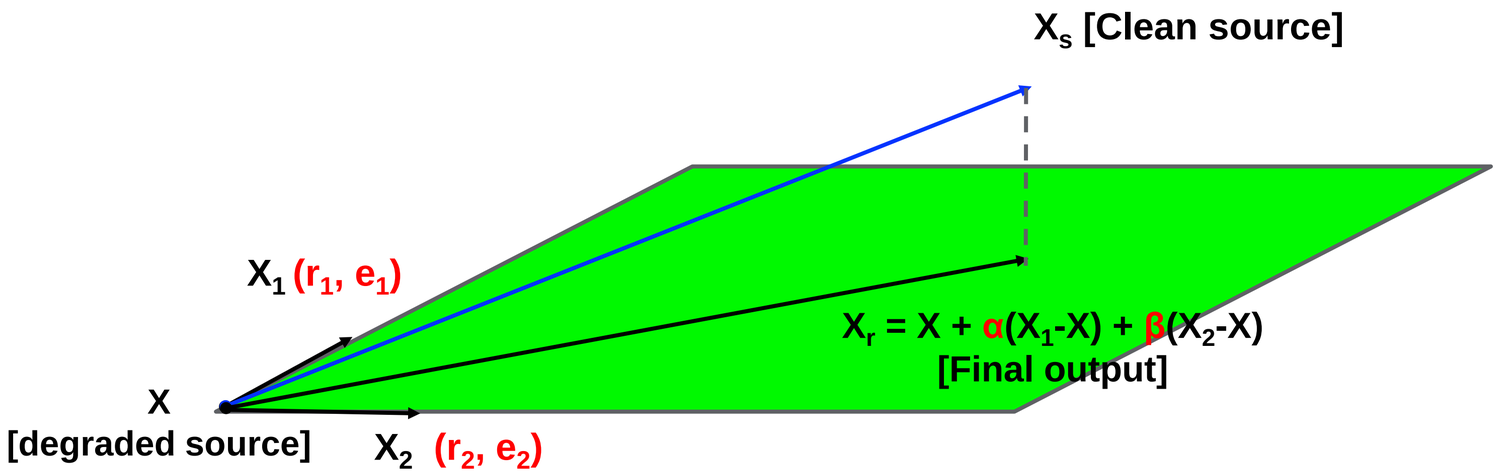
$X_1$ and $X_2$ are cheap restored versions, Subspace projection can yield a much better final restoration $X_r$.











# Restoration and Post-processing

## VP9

In loop deblocking filters

## AV1

In loop deblocking filters with **adaptive strengths**

**Directional enhancement filter**

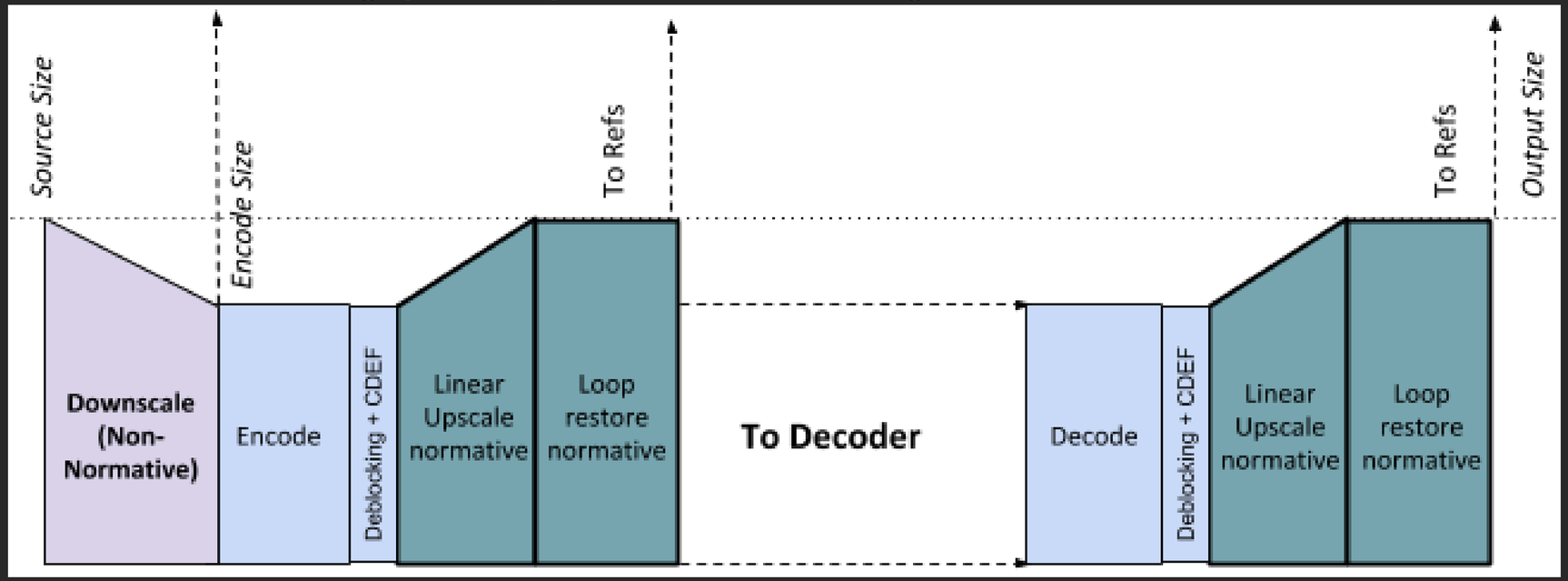
**Loop restoration filter**

**Frame super resolution**

**Film grain synthesis**

# In-loop Frame Super Resolution

- AV1 supports a frame super-resolution coding mode: coding a frame at lower horizontal resolution and then restored it normatively to full resolution
- Super-res = linear upscaling + guided in-loop restoration tool at higher res
- This mode could bring perceptual advantages at low bit-rate



# Restoration and Post-processing

## VP9

In loop deblocking filters

## AV1

In loop deblocking filters with **adaptive strengths**

**Directional enhancement filter**

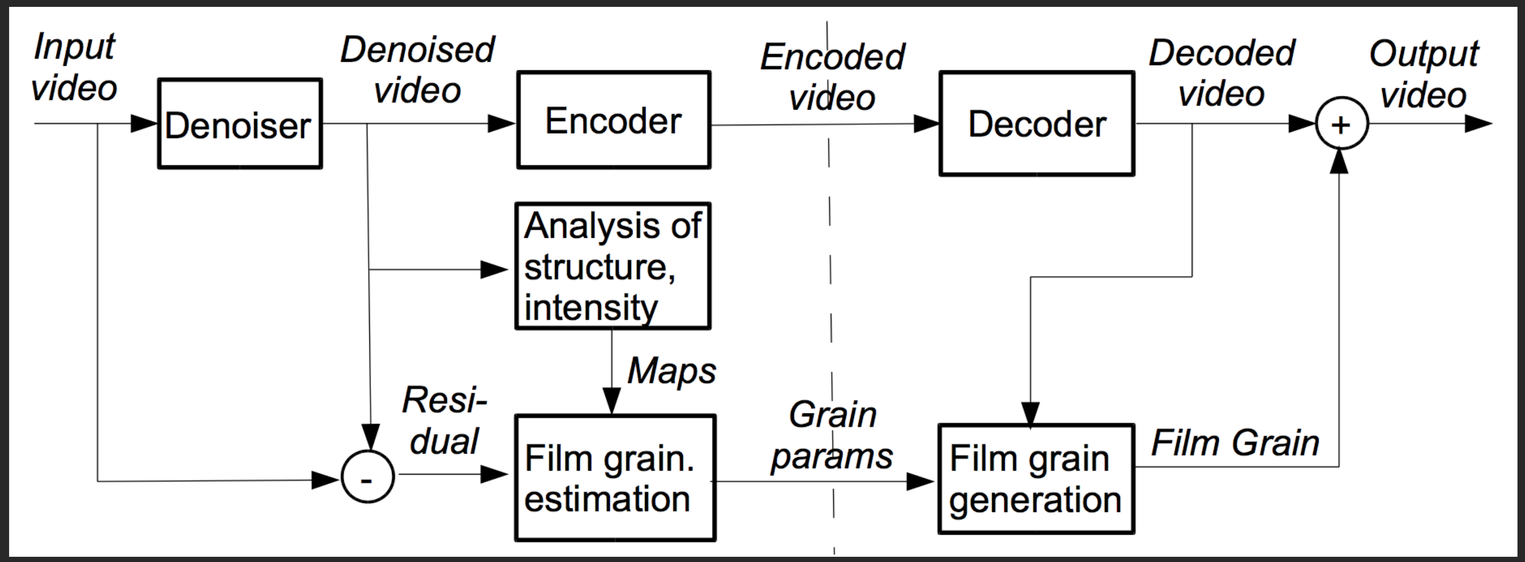
**Guided loop restoration filter**

**Frame super resolution**

**Film grain synthesis**

# Film Grain Synthesis

- Film grain is present in much of the commercial content
- It is difficult to compress but needs to be preserved as part of creative intent
- AV1 supports film grain synthesis via a normative post-processing applied outside of the encoding/decoding loop



WebRTCon
为开发者赋能 为行业加速

# Outline

AOMedia and AV1

Coding Techniques

***Coding Performance***

What’s Next

# Compression Efficiency

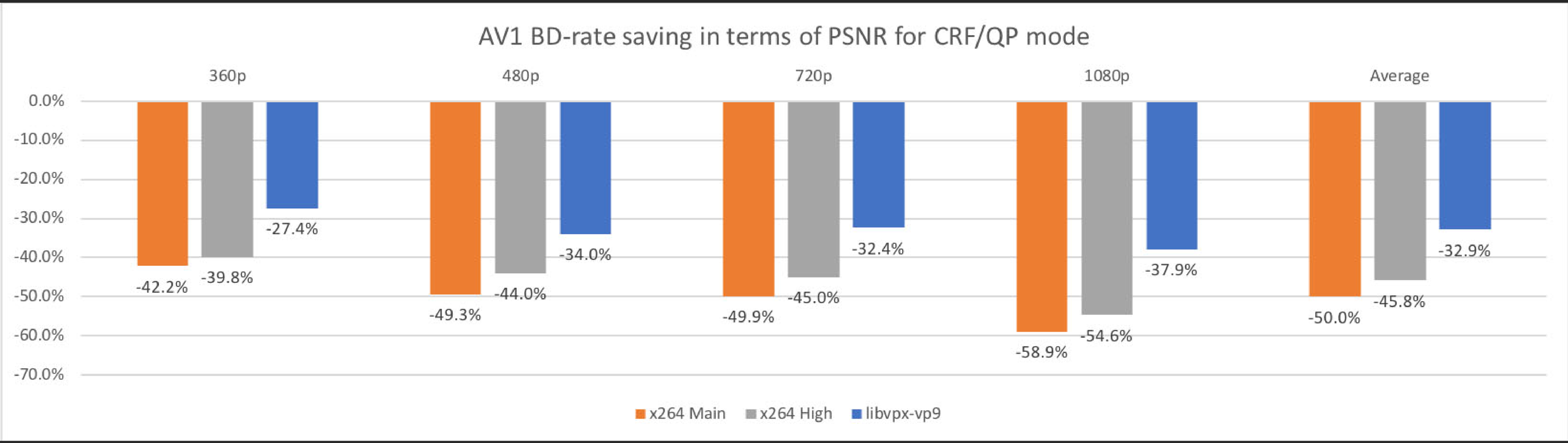
- Test condition: AWCY[1] objective1-fast[2], 30 x 1080p~360p clips, 60 frames
- AV1 CQ mode, libvpx-VP9 CQ mode, x265 CRF mode
- BDRate (%)

| **Codecs \ Metric** | PSNR-Y | PSNR-Cb | PSNR-Cr | CIEDE-2000 |
|---|---|---|---|---|
| **AV1 speed 0** vs. **libvpx speed 0** | **-28.46** | **-30.78** | **-32.99** | **-29.86** |
| **AV1 speed 1** vs. **libvpx speed 0** | **-26.91** | **-30.45** | **-32.00** | **-28.88** |
| **AV1 speed 0** vs. **x265 placebo** | **-23.97** | **-40.60** | **-41.78** | **-34.62** |
| **AV1 speed 1** vs. **x265 placebo** | **-22.43** | **-40.34** | **-41.33** | **-33.74** |

[1] arewecompressedyet.com [2] https://people.xiph.org/~tdaede/sets/objective-1-fast/

# Compression Efficiency

- Results from Facebook Tests[1]




[1] https://code.facebook.com/posts/253852078523394/av1-beats-x264-and-libvpx-vp9-in-practical-use-case/

# Coding Complexity

- Coding complexity of AV1 speed 1
- Compared against libvpx-vp9 speed 0
- More speed-up is on the way

| **Resolution** | ENC time/frame | ENC time vs libvpx | DEC frame/s | DEC time vs libvpx |
|---|---|---|---|---|
| **832x480, 8 bit** | 33s | ***59x*** | 231 | ***4.8x*** |
| **480x360, 10 bit** | 72s | ***41x*** | 183 | ***4.6x*** |

WebRTCon
为开发者赋能 为行业加速

# Outline

- AOMedia and AV1
- Coding Techniques
- Coding Performance
- ***What’s Next***

# What’s next?

- **Speed up** the codec
  - More SIMD coverage, two-pass partition decision making, fast mode determination, including those ML based approaches ...
  - Set up and tune lower complexity speed modes (speed 1 - 8)
- Continue improving **compression performance**
  - Rate control, adaptive quantization, frame super resolution, …
  - Different eng usage modes will be explored, e.g. perceptual quality mode

# Thank You !

*Zoe Liu* (zoeliu@google.com)