Jerome Jiang
Marco Paniconi

VP9 Features & Optimizations for Real-Time Video

Jerome Jiang

jianj@google.com
Google LLC

Outline

- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

Outline

- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

Introduction

- Why optimize encoder for real-time
  - “good” mode too slow
    - Made for VOD (Video on Demand) use cases

    - Encoder runs on servers which have plenty of power
  - Real-time video can’t use 2 pass

Outline

- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

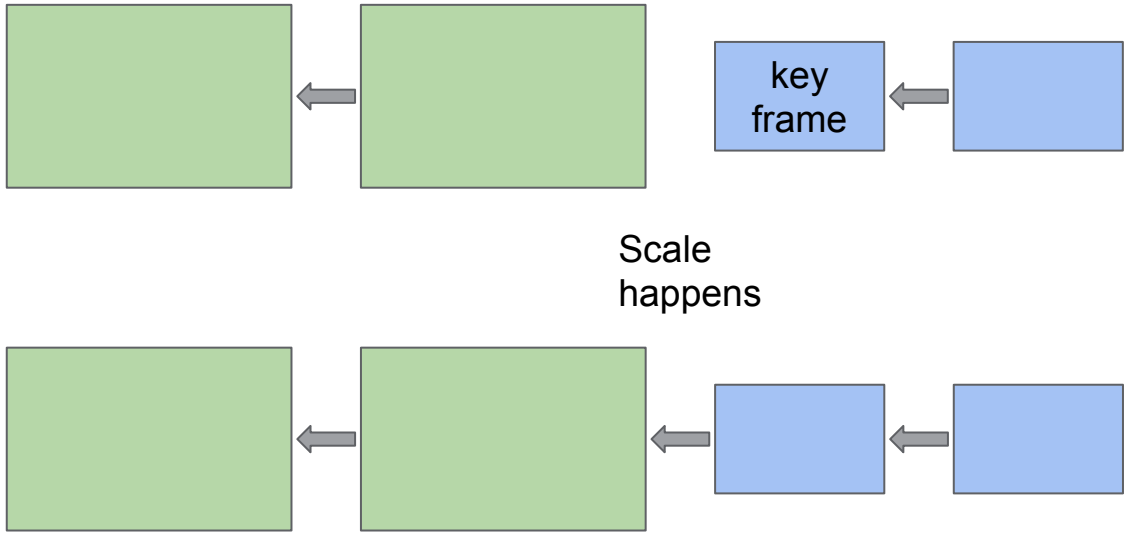
SVC (Scalable Video Coding) in VP9

- Fully integrated in WebRTC
- Actively being experimented/tuned
- New features rolling in actively
  - Frame dropping
  - Dynamic pattern updates
  - Speed ups
  - Quality Improvements
  - Screenshare

SVC in VP9

- Unique feature of reference frame scaling
  - Spatial layers for SVC
  - Dynamic resize (change resolution within stream without key frame)
- Intra-only frame
- Multiple spatial & temporal layers
- Change layer pattern on the fly (flexible SVC mode)
- Cyclic refresh
  - Segment level QP
- Noise estimation & denoising
  - All spatial layers

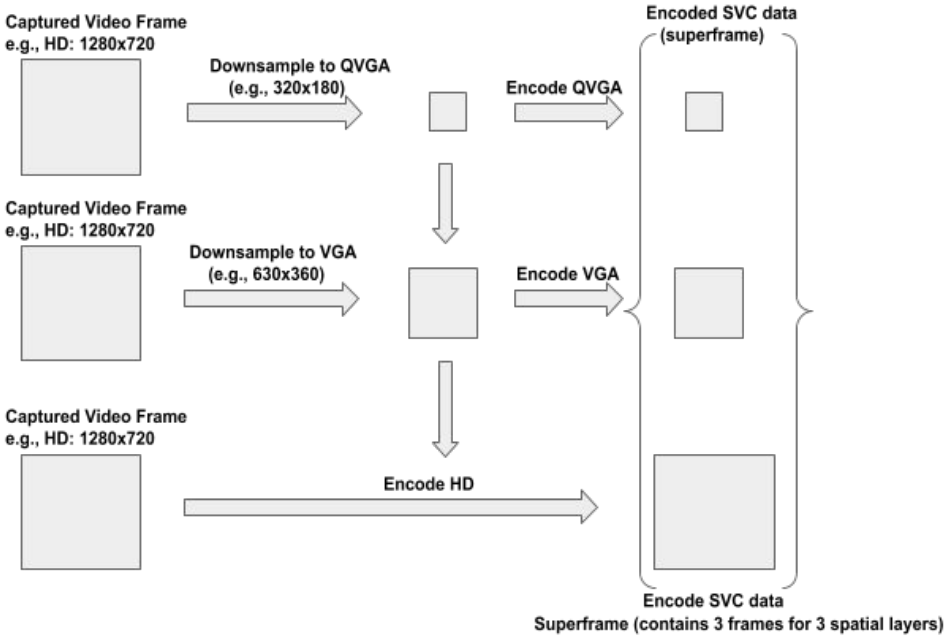
Dynamic Resize in VP9



- The stream must lower resolution to hit bitrate
- No key frame needed for prediction from last frame
  - Smaller frame size
  - Less fluctuation
  - Smoother quality
- Same thing as the resolution scaled up
- Used as default for VP9 in WebRTC

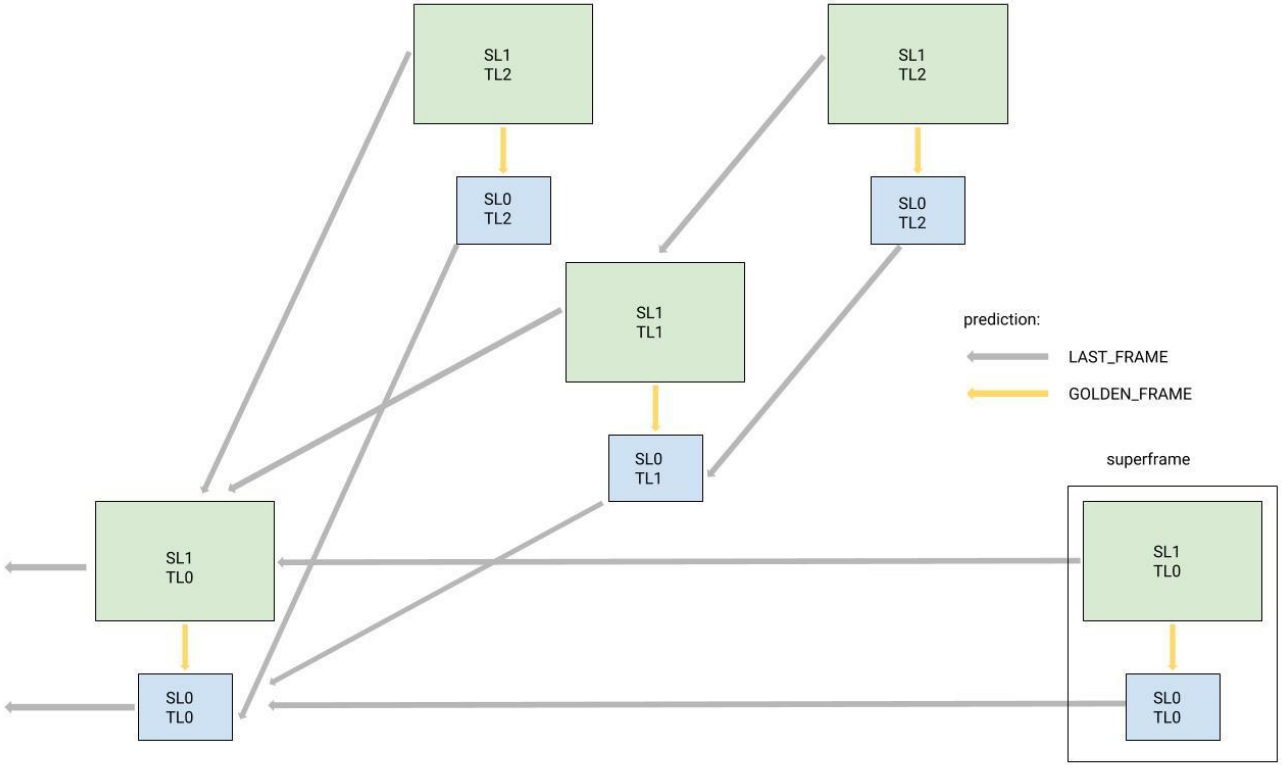
SVC Superframe

- A superframe is a frame packet containing all spatial layers.
- Downsample to lowest resolution first then encode
- Higher resolution frames predict from lower resolution ones



3 Spatial Layers

SVC Patterns - 2 Spatial Layers, 3 Temporal Layers



SVC Reference frame buffer and refresh

- ALTREF reference frame buffer is used in SVC.
- 2SL 3TL example:

	SL0 TL0		SL1 TL0		SL0 TL2		SL1 TL2		SL0 TL1		SL1 TL1		SL0 TL2		SL1 TL2	
	B	R	B	R	B	R	B	R	B	R	B	R	B	R	B	R
0	L	✔	G		L				L							
1			L	✔	G		L		G		L		G			
2					A	✔	G		A	✔	G		L	✔	G	
3							A				A	✔			L	

B = Buffer index.
R = Refresh.
L = LAST_FRAME.
G = GOLDEN_FRAME.
A = ALTREF_FRAME.

SVC Interlayer Prediction

- Users have control about inter-layer prediction (configurable)
- Several modes
  - `INTER_LAYER_PRED_ON`
    - Default mode, interlayer prediction always on.
  - `INTER_LAYER_PRED_OFF`
    - Interlayer prediction always off
  - `INTER_LAYER_PRED_OFF_NONKEY`
    - Interlayer prediction off for non keyframes
  - `INTER_LAYER_PRED_ON_CONSTRAINED`
    - Inter-layer prediction is on on all frames, but constrained such that any layer S (> 0) can only predict from previous spatial layer S-1, from the same superframe.

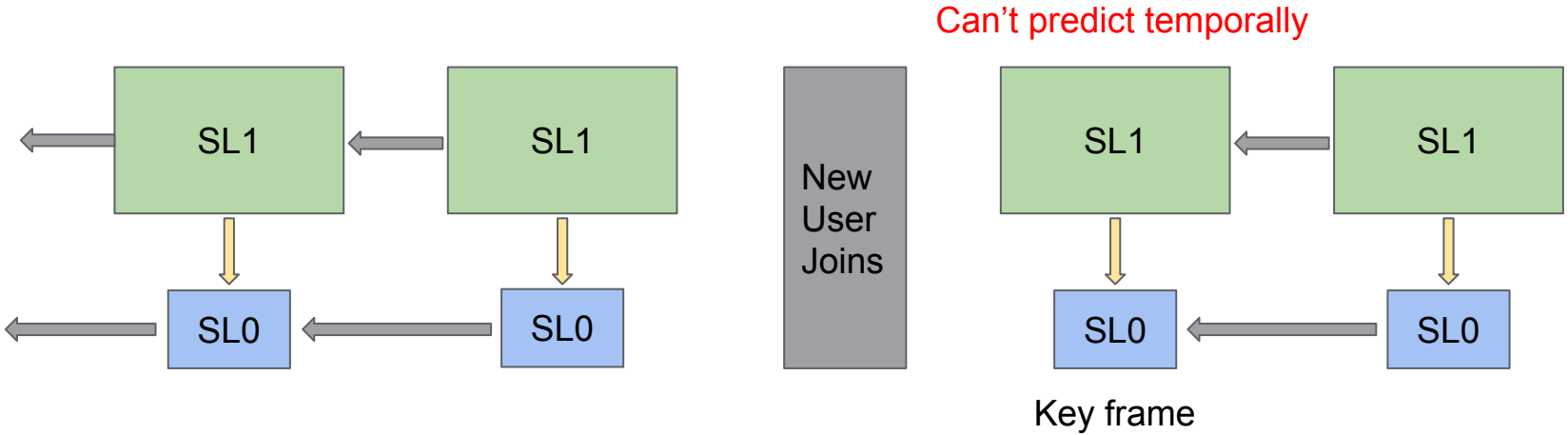
SVC Frame Dropping

Several frame dropping modes:

- `CONSTRAINED_LAYER_DROP`
  - Upper layers are constrained to drop if current layer drops.
- `LAYER_DROP`
  - Any spatial layer can drop.
- `CONSTRAINED_DROPBASE_ENCODESKIP`
  - Base spatial layer can drop, and this forces drop of all spatial layers. Enhancement spatial layer encodes a skip frame instead of dropping.

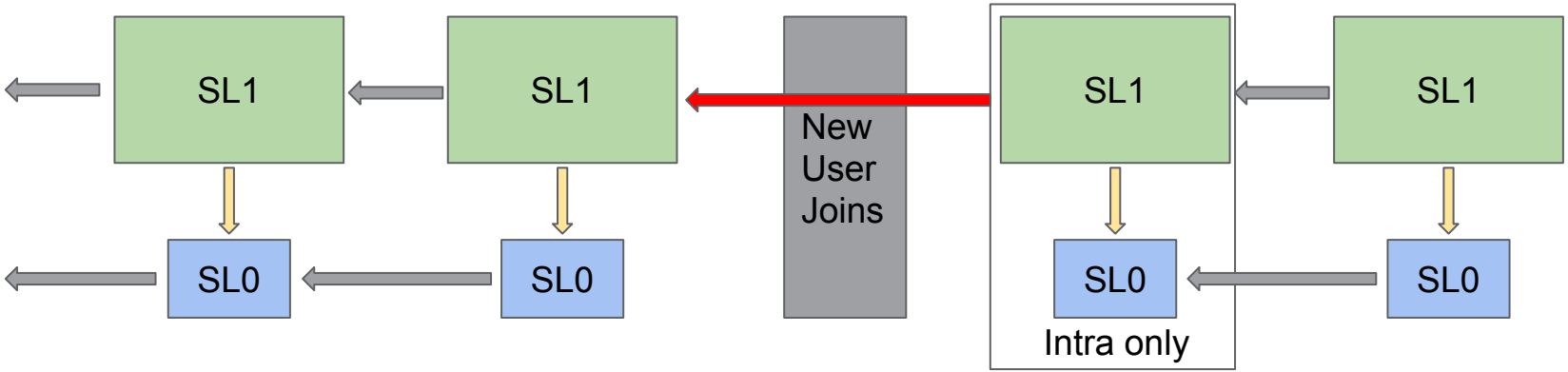
Intra-only Frame

- New user joins the group chat
  - Insert base layer as a key frame
  - All receivers need to restart the videostream



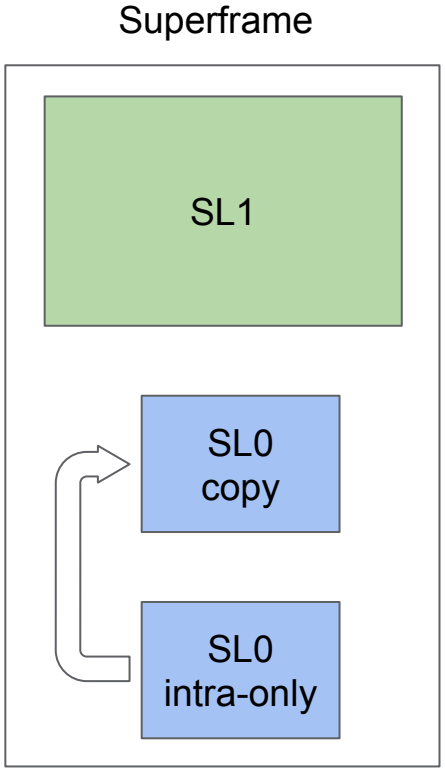
Intra-only Frame

- With intra-only frame
  - Frame encoded with intra only
  - But doesn't refresh all reference buffers
  - Must be a no show frame



Intra-only Frame

- For receivers who decode top layer
  - Can still predict temporally
  - Avoid effects of key frame
- Intra-only frame is still packed into the superframe
  - No show - (not displayed)
  - Can use flag `show_existing_frame` to copy header of intra-only frame in the superframe
- Experimental feature

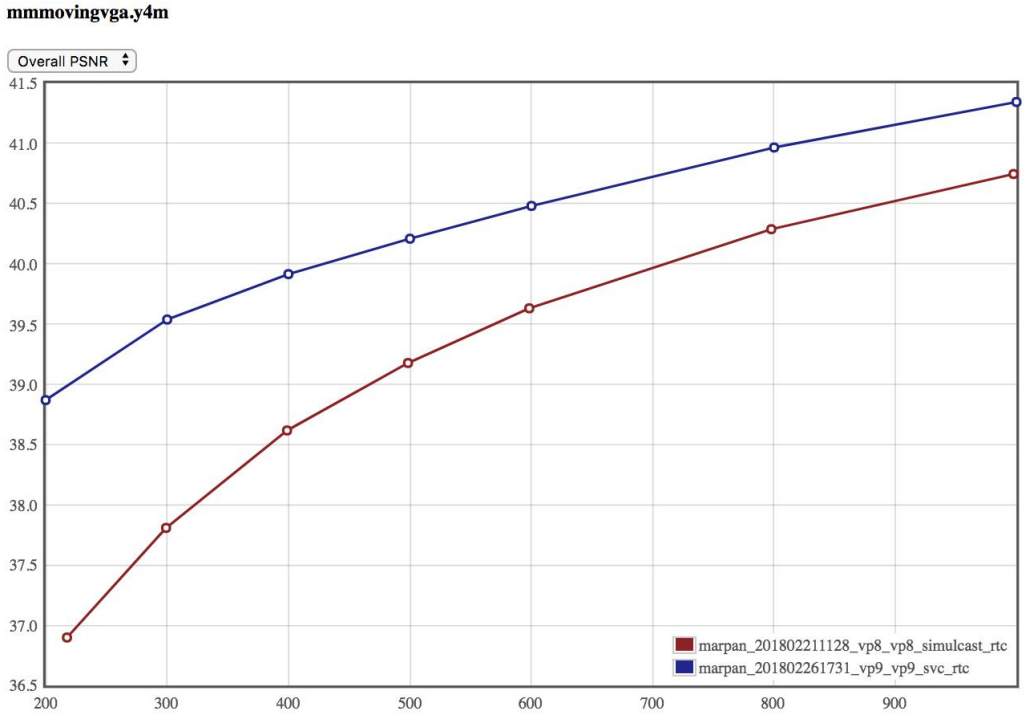


Outline

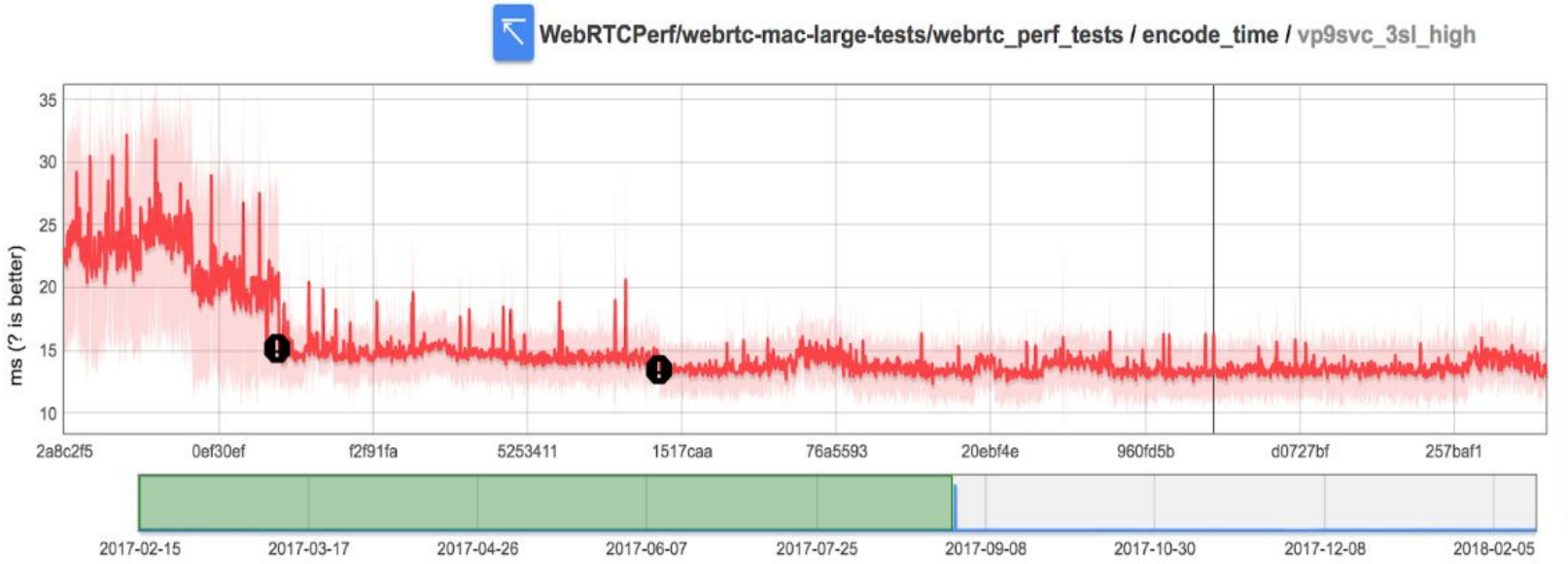
- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

VP9 SVC v.s. VP8 Simulcast

File	Match	Problem	avg_psnr:
dark720p.y4m	✓	8/0	-54.809
desktop2360p.y4m	✓	7/0	-39.724
desktop360p.y4m	✓	7/0	-58.236
fourpeople720p.y4m	✓	8/0	-38.593
gipsrecmotion720p.y4m	✓	8/0	-17.756
gipsrestat720p.y4m	✓	8/0	-34.539
jimredvga_25fps.y4m	✓	7/0	-47.880
kirlandvga.y4m	✓	7/0	-60.486
marcooffice720p.y4m	✓	8/0	-28.842
mj1vc720p.y4m	✓	8/0	-40.792
mj2vc720p.y4m	✓	8/0	-29.558
mj3vc720p.y4m	✓	8/0	-35.089
mj4vc720p.y4m	✓	8/0	-25.326
mmmovingvga.y4m	✓	7/0	-42.653
mmstionaryvga.y4m	✓	7/0	-51.641
niklas720p.y4m	✓	8/0	-27.064
niklasvga.y4m	✓	7/0	-35.198
still_bright_360_640.y4m	✓	7/0	-40.011
tacomanarrowsvga.y4m	✓	7/0	-70.085
tacomascmvvga.y4m	✓	7/0	-39.452
testnoise720p.y4m	✓	8/0	-59.004
thaloundeskmtgvga.y4m	✓	7/0	-45.923
vidyo1_1280x720_60.y4m	✓	8/0	-28.229
vidyo3_1280x720_60.y4m	✓	8/0	-24.326
vidyo4_1280x720_60.y4m	✓	8/0	-38.088
{OVERALL}	✓	None	-40.532



VP9 SVC Speed up



Overall 45% speed up on HD (720p).

Outline

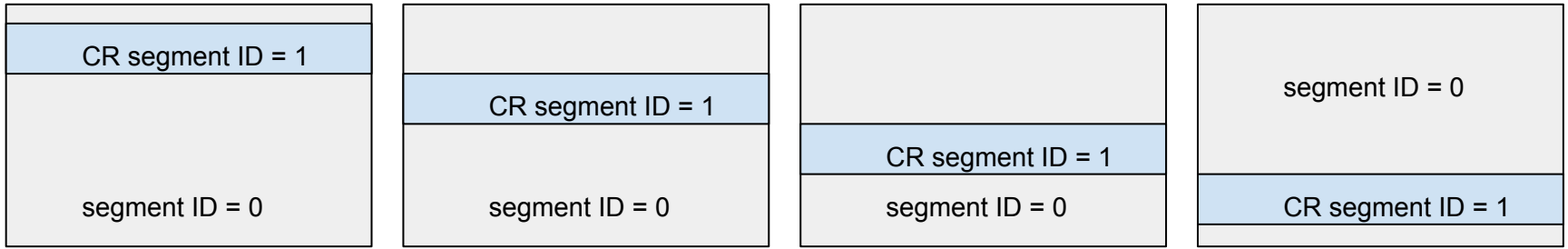
- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

Segmentation

- VP8 and VP9 allow users to specify areas to apply different parameters with the rest of the frames.
  - Quantization Parameters
  - Loop filter strength
  - Static threshold - only in VP8, set threshold for skipping motion search
  - Skip encoding or not - only in VP9, copy the block from previous frame
  - Reference frame - only in VP9
- VP8 - 16x16 block, 4 segments
- VP9 - 8x8 block, 8 segments
- Two features based on segmentation
  - AQ Mode - cyclic refresh
  - ROI - Region of Interest

Cyclic Refresh (AQ_mode = 3)

- VP9 encoder selects percentage of frame to apply lower quantization parameter (QP).
- The area will move from frame to frame, adaptively.
- After a period of time, whole frame will be refreshed.
- Good quality boost for video conference content.
- Integrated into the CBR rate control.
- Used as default for WebRTC.



ROI - Region of Interest

- ROI enables users to specify any area to apply different encoding parameters
- API

```
if (vpx_codec_control(&codec, VP8E_SET_ROI_MAP, &roi))
      die_codec(&codec, "Failed to set ROI map");
```

- `roi` is type `vpx_roi_map_t`
  - Users need to specify `roi` first
  - ROI area is marked by 0 and non-zero values - which segment to use
  - Specify different QP, loop filter strength etc.

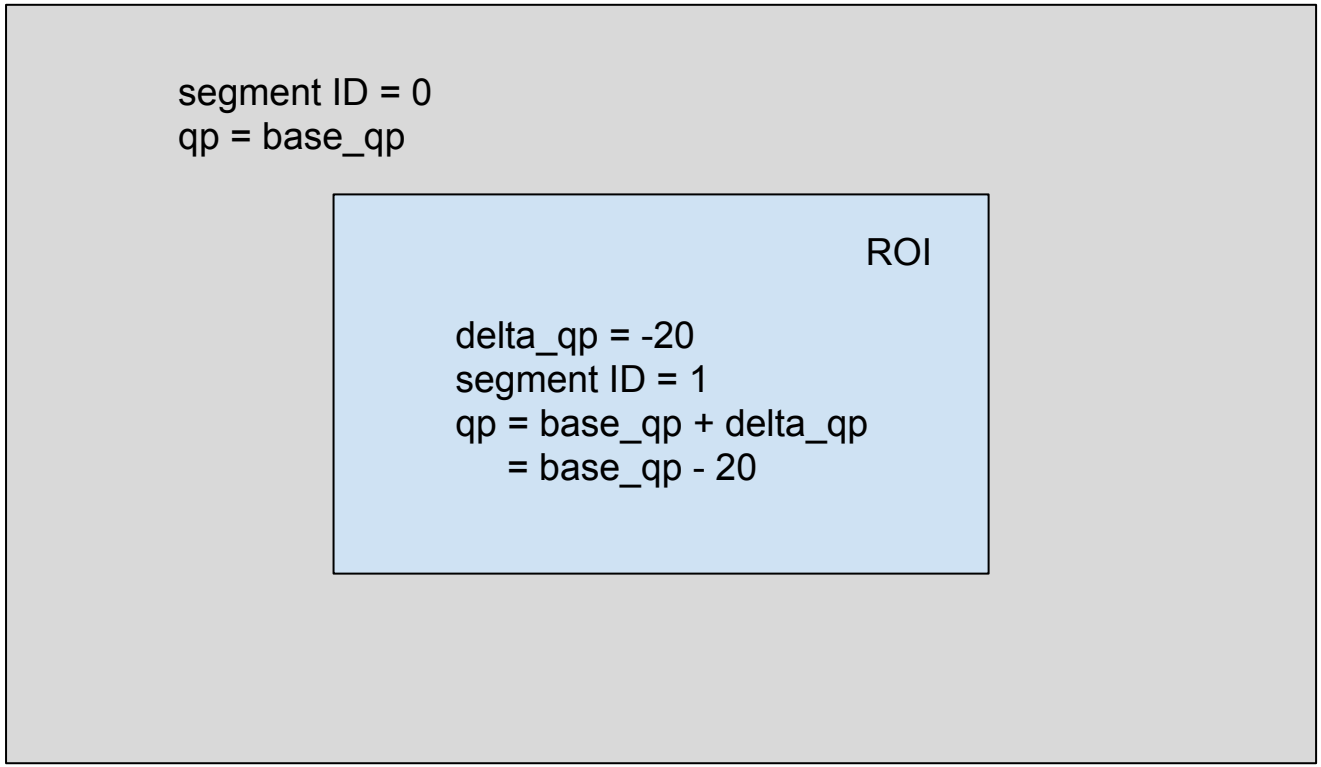
ROI struct key elements

```
typedef struct {
/* ... */
/*! VP8 only uses the first 4 segments. VP9 uses 8 segments. */
  int delta_q[8];  /**< Quantizer deltas. */
  int delta_lf[8]; /**< Loop filter deltas. */
  /*! skip and ref frame segment is only used in VP9. */
  int skip[8];      /**< Skip this block. */
  int ref_frame[8]; /**< Reference frame for this block. */
  /*! Static breakout threshold for each segment. Only used in VP8. */
  unsigned int static_threshold[4];
} vpx_roi_map_t;
```

ROI Parameters - QP & Loopfilter

- For QP and loop filter, the struct specifies “delta”
  - `qp_roi = base_q + delta_q`
  - If `delta_q < 0`, then `qp_roi` is lower than `base_q`, which is QP for the frame. Thus ROI has better quality.
  - If `delta_q > 0`, then `qp_roi` is higher than `base_q`, thus ROI has worse quality.
- `-63 <= delta_q <= 63`
- `-63 <= delta_lf <= 63`

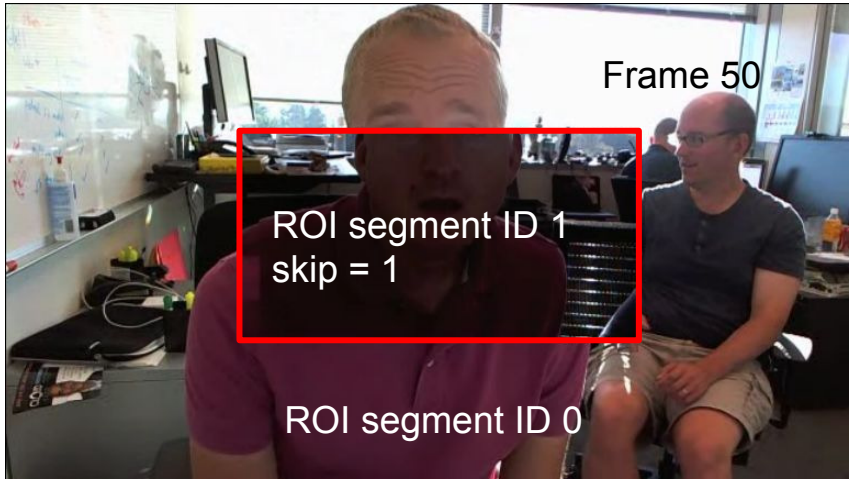
ROI Parameters - QP



ROI Parameters - Reference frame

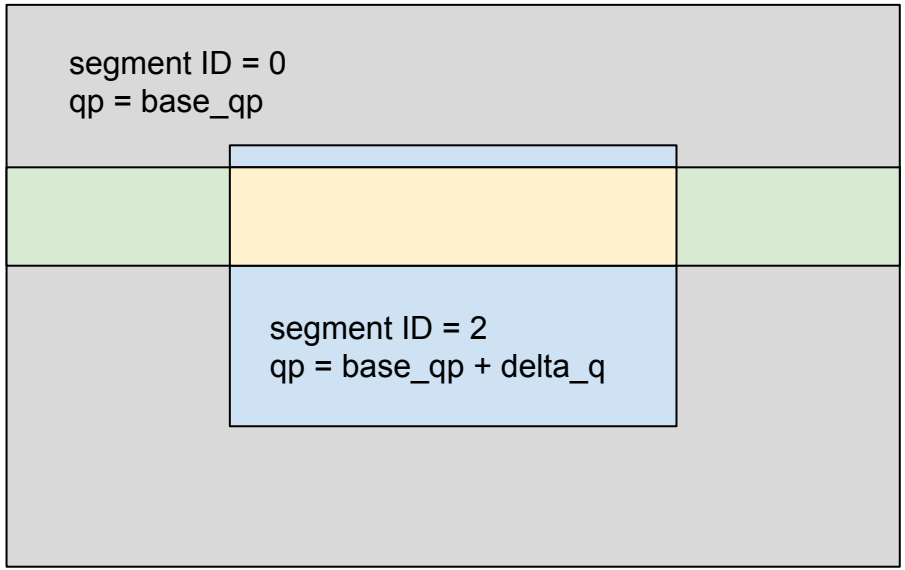
- Force using specified reference frame by users
- Reference frame (`ref_frame`): only used in VP9.
  - -1: Do not apply this feature
  - 0: Force using intra
  - 1: Force using last frame
  - 2: Force using golden frame
- Special cases:
  - `ALTREF_FRAME` is not used in non-rd pickmode for 0 lag. If user forces to use `ALTREF_FRAME`, ignore this feature and don't do anything.
  - When `GOLDEN_FRAME` is not set as one of reference frames and user forces to use `GOLDEN_FRAME`, just ignore this feature.
  - `GOLDEN_FRAME` is updated on last frame, where `GOLDEN_FRAME` and `LAST_FRAME` is the same frame. Map `GOLDEN_FRAME` to `LAST_FRAME`.

ROI Parameters - Skip



Segmentation - Future Work

- Make ROI and AQ-Mode work together
- If users enable both ROI and AQ-Mode, codec needs to put them into two different segments.



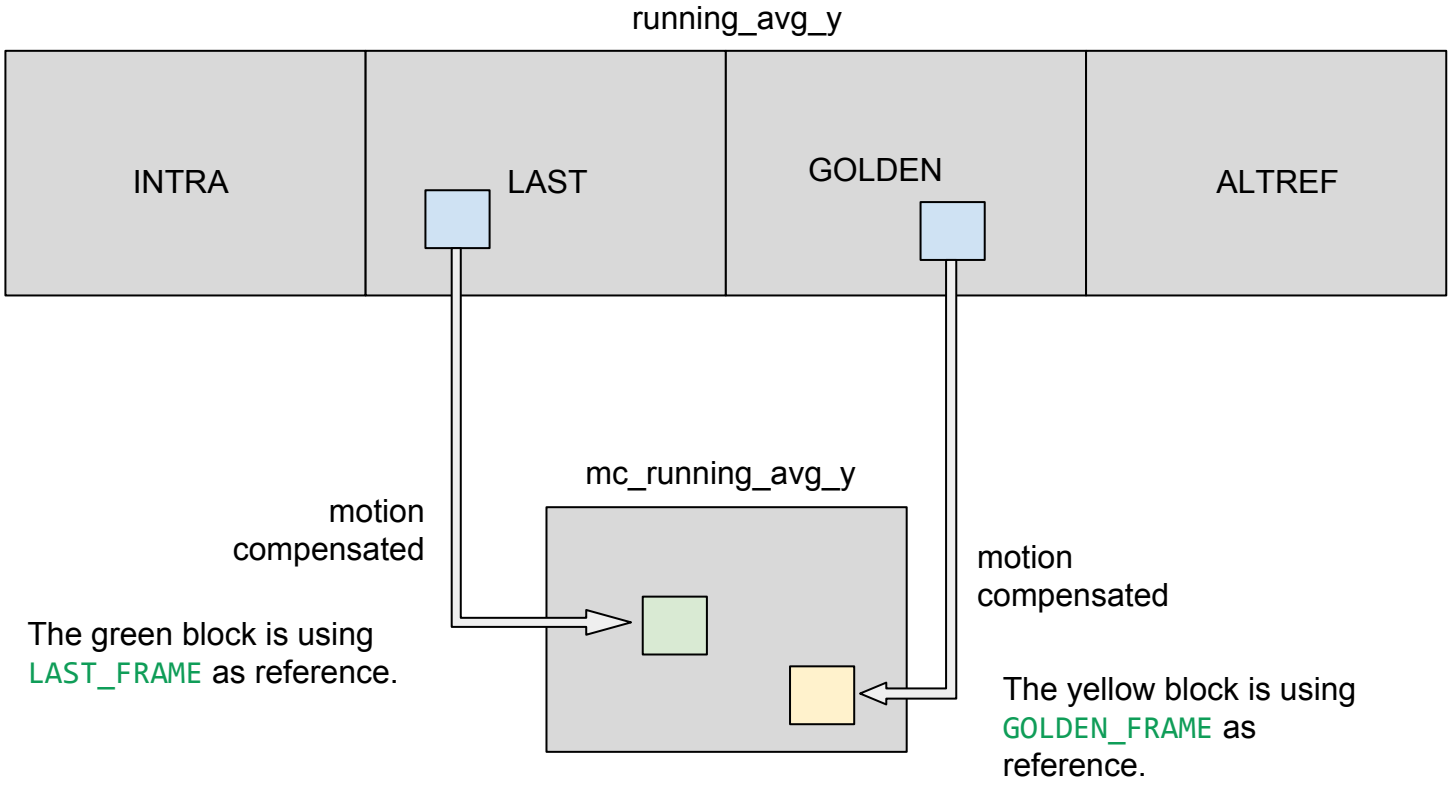
Cyclic Refresh effective
segment ID = 1
qp = base_qp + cr_delta_q

Cyclic Refresh & ROI overlapped
segment ID = 1, 2
qp = base_qp + cr_delta_q
+ delta_q

VP9 Temporal Denoiser

- Noise estimation
- Denoiser will decide according to noise estimation
  - If noise level is low - don’t denoise even if denoising enabled by user
  - If noise level is high
    - Perform motion compensation - return two values
      - `COPY_BLOCK` - Copy block from source without denoising
      - `FILTER_BLOCK` - Denoise the source

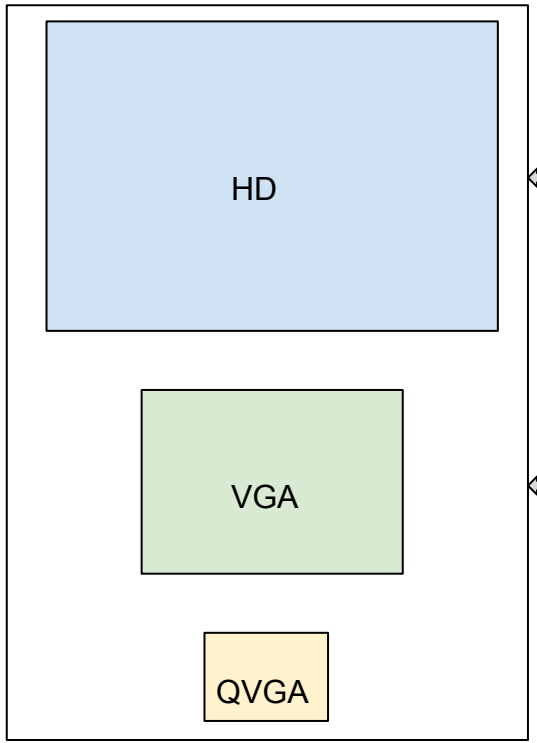
Denoiser Frame Buffer



Denoiser in SVC

More frame buffers: `N_{reference frames} x N_{denoise spatial layers}`

Denoiser in SVC



Superframe

⇐ first_layer_denoise

```
vpx_codec_control(&codec,
VP9E_SET_NOISE_SENSITIVITY, 1)
```

Only denoise top layer,
`N_{denoise spatial layers} = 1`

⇐ first_layer_denoise

```
vpx_codec_control(&codec,
VP9E_SET_NOISE_SENSITIVITY, 2)
```

Denoise top 2 layers,
`N_{denoise spatial layers} = 2`

Outline

- Introduction
- SVC (Scalable Video Coding) in VP9
- SVC Metrics Comparison v.s. VP8 Simulcast
- Segmentation (AQ-Mode, ROI)
- Temporal Denoiser
- VP9 Optimizations for ARM

VP9 Optimizations for ARM

- Different speed settings for VP9
  - Real-time uses speed >= 5
  - Speed 5 and 6 - used by YouTube Live
  - Speed 7 - desktop
  - Speed 8 - ARM (Mobile devices)
- We’re adding speed 9
  - Motion search is very expensive
  - Adaptively prune subpel search according to different content
  - Speedup partitioning, reduce Golden mode search
  - Used on ARM

VP9 Optimization for ARM - Multi-threading

- Tile based multi-threading
- VP9 multi-threading was based on tiles before
  - Every tile is **at least** 256 pixels wide
    - Single thread on low-res
  - Every thread works on one tile
    - Number of threads can’t be greater than number of tiles
    - 4 threads most on HD (720p)
- Example - two tiles
  - It often happens one thread is faster than the other
  - Because of content
  - Faster thread needs to wait for the slower one

Row-based Multi-threading

- Add row-based multi-threading on top of tile-based
- Superblock (64x64) row based
- Allow number of threads to be greater than number of tiles:
  - Allow multi-threading on low-res
  - Example: 4 threads on 2 tiles
    - 2 threads on one tile, 2 threads on the other
    - After the faster tile finishes, the thread(s) will help the slower one
- Remove dependencies between superblock rows
- API

```
vpx_codec_control(&codec, VP9E_SET_ROW_MT, 1);
```

Bonus - Fast Partitioning

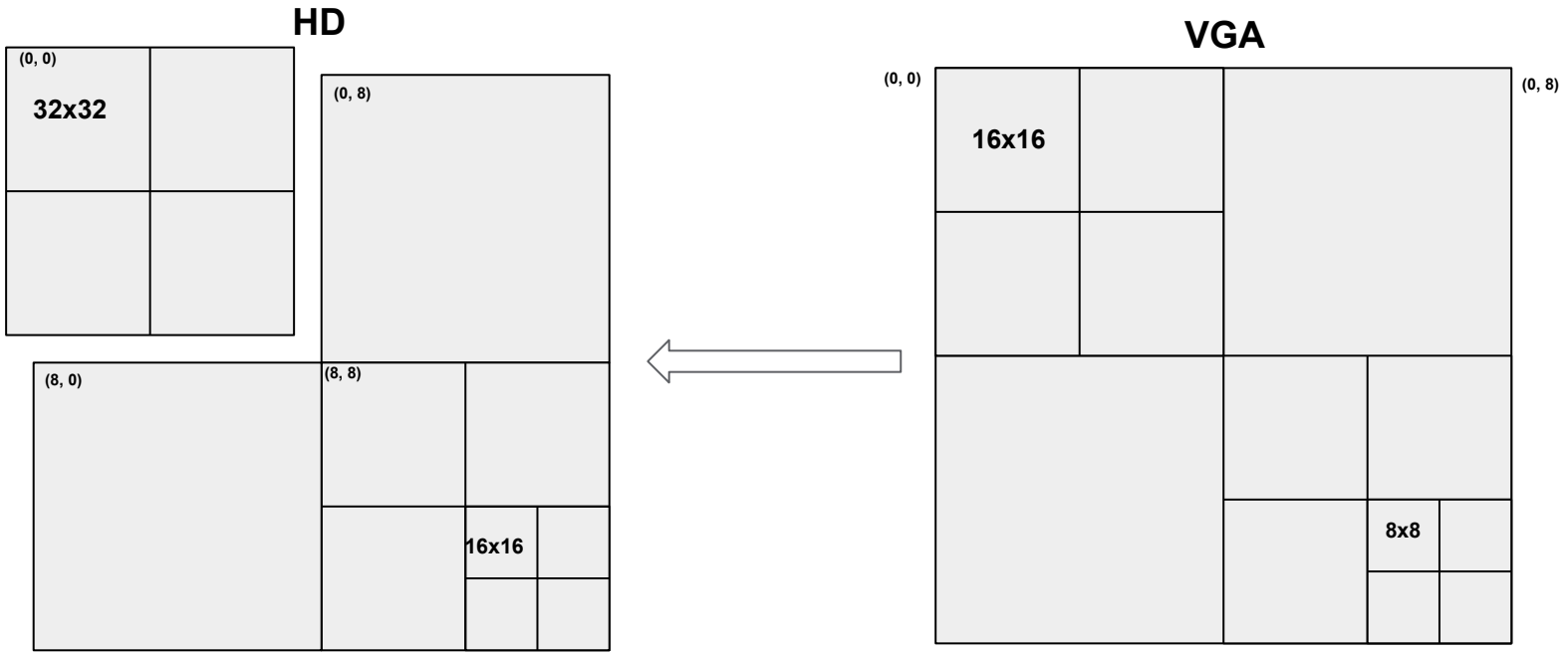
- Superblock Partitioning takes a lot of time
  - Variable Block sizes in VP9 (64X64, 64x32, 32x64,32x32,…,4x4).
  - Recursive structure
  - Compute at each level
- For real-time, we use Variance-based fast partitioning
  - If the SAD between **current superblock** and **motion compensated superblock** from last frame is small
    - Copy partitioning from last frame
  - For SVC:
    - Add scaling superblock from lower resolution to higher

Fast Partitioning

To balance between speed and quality:

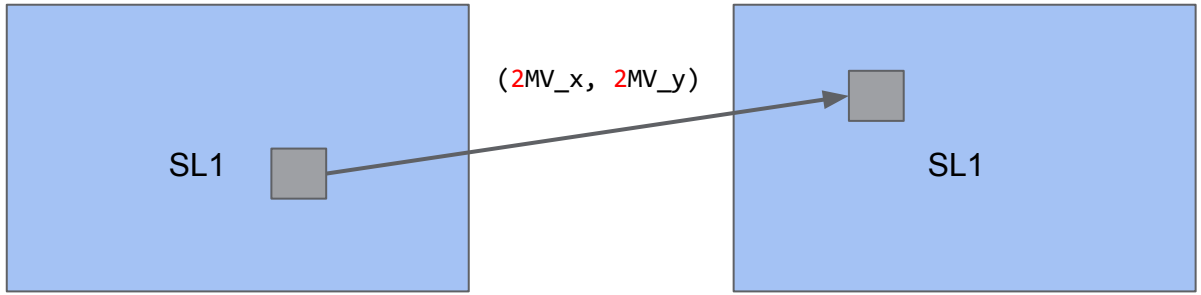
- Add a counter for each superblock
  - How many times this superblock has been copied consecutively
- If the counter is greater than threshold - 5 currently
  - Stop reusing for this block in this frame
  - Do normal partitioning to reduce quality loss
- Only enabled for speed 7 and 8
  - Speed 8 is only used on ARM real-time

Scale partitioning in SVC

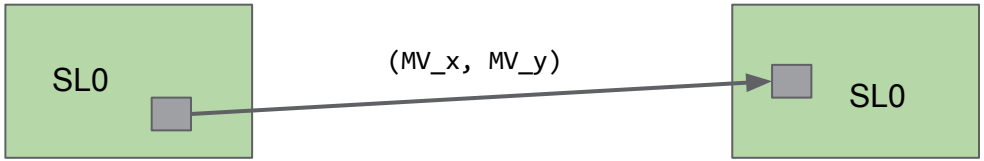


Motion Vector Reuse

Reuse motion vector from base layer for faster NEWMV search on LAST.



Smaller search area
around (2MV_x, 2MV_y)

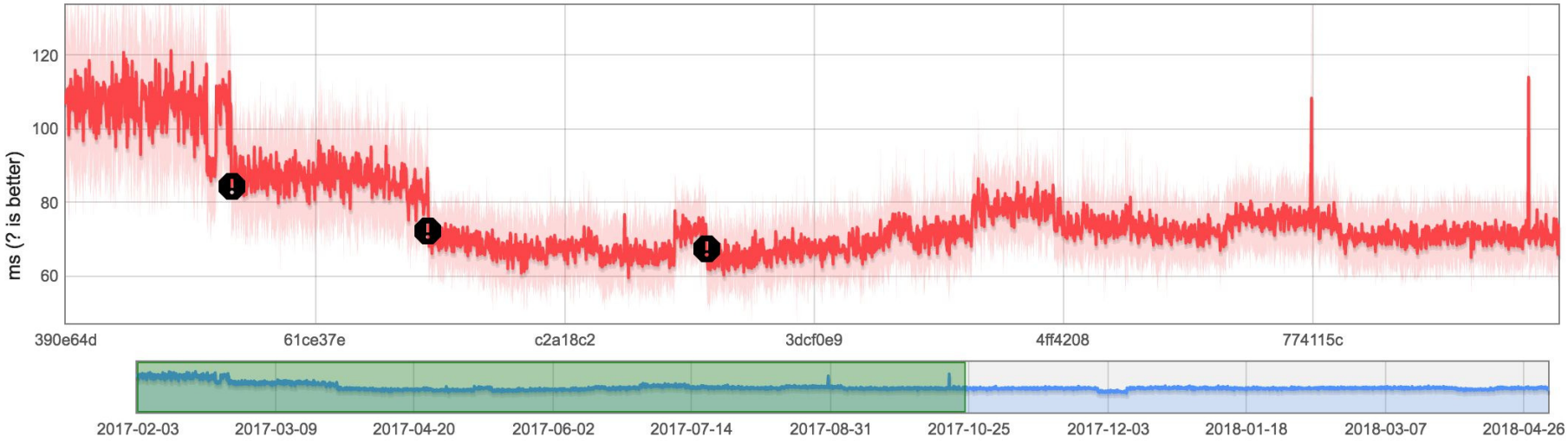


VP9 Improvements on ARM

Tested 2 threads on Google Pixel XL - Qualcomm Snapdragon 820
Over the past year

- VGA
  - 93% speed up
  - 120fps ~ 180fps depending on content
- HD
  - 95% speed up
  - 40fps ~ 64fps

VP9 Improvement on ARM



WebRTC test on HD (720p) on Nexus 6, very old phone.
Encoding time from 110ms to 70ms.

Wrap Up

- Huge progress on VP9 SVC
  - Integrated in WebRTC for Hangouts
- VP9 SVC maintains quality advantage over VP8 simulcast
- Segmentation adds more flexibility to developers
  - Face Recognition - Use lower QP on face
  - Image segmentation - Use higher QP on background
- Denoiser improves quality under noisy situations
- Huge speed up of VP9 on ARM
  - More than 90% for 2 threads on high-end phones

Thank you!