Microsoft

GPGPU Implementation of VP9 in-loop deblocking filter and improvement for AV1 codec

Zhijun (Ryan) Lei[1], ryan.z.lei@intel.com
Victor Cherepanov[2], victor.cherepanov@mail.ru
Srinath Reddy[2], srinath.reddy@Microsoft.com
Zhipin Deng[1], zhipin.deng@intel.com

[1] Intel Corporation, [2] Microsoft Corporation

Overview

- VP9 In-loop Deblocking Filter Algorithm
- GPGPU Implementation of the VP9 Deblocking Filter
- Parallel Deblocking Proposal for AV1 Coding Standard
- Summary and Future Work

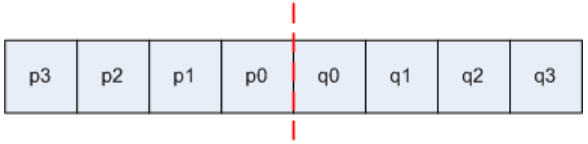
VP9 Deblocking Filter Operation Order

- Operates following raster scan order of 64x64 Super Block within a reconstructed frame (1 Luma plane, 2 Chroma planes)
  - Filter level parameters are encoded in frame header, and can be adjusted based on segmentation setting.
  - Tile boundary is treated the same as super block boundary.
  - Disabled in lossless coding mode.
- For each Super Block, filtering is performed as follows :
  - All vertical edges are filtered first (from left to right) followed by all horizontal edges (from top to bottom).
  - Only transform block boundaries (4x4, 8x8, 16x16 and 32x32) are filtered.
  - Within a prediction block, filtering is applied on a transform block boundary if the transform block has at least one non-zero coefficient or the block is encoded using intra prediction.
    - Except 32x32 and 64x64 block boundaries are always filtered.
  - Picture borders are not filtered.

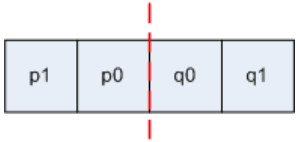
VP9 Deblocking Filter Level Control

- Loop Filter Control
  - Frame Level: loop_filter_level indicates loop filter strength and loop_filter_sharpness indicates the sharpness level. Both are used together to determine whether a block edge should be filtered and by how much filtering can change the sample values.
  - Based on a given default filter level of a frame, final filter level (for 8x8 block) can be further adjusted based on segmentation settings.
  - If final filter level is set to 0, no deblock filtering is performed.
- Deblocking filter decision
  - For every pixel along the edge, decision is made adaptively based on local statistics in the following steps:
    - Block detection
    - High Edge Variance detection
    - Flat region detection
    - Filter length decision

Deblocking Filter Decision



- Block Detection:
  - Detect whether pixels close to the edge vary by less than a threshold or not. Only edges introduced by transform process should be filtered.
  - Read 4 pixels on each side of the edge

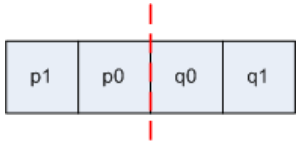


- High edge variance detection.
  - Checks two pixels on each side of the edge, determines how many pixels should be modified.
  - Real edges should be filtered less to preserve edge sharpness.

- Flat region detection.
  - Checks whether pixels from each side of the edge are in flat area or not.
  - Normal flat detection checks 4 pixels on each side of the edge, extended flat detection checks 8 pixels on each side of the edge.
  - Normal flat check is only applied to transform block size greater than 8x8.
  - Extended flat check is only applied to transform block size greater than 16x16.

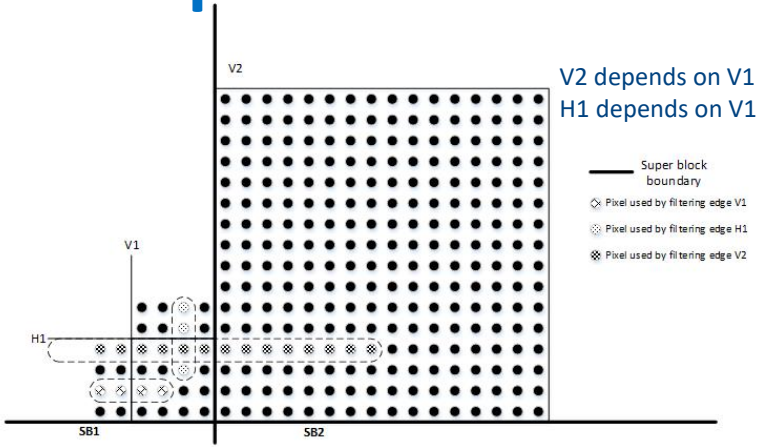
Deblocking Filter Length Decision

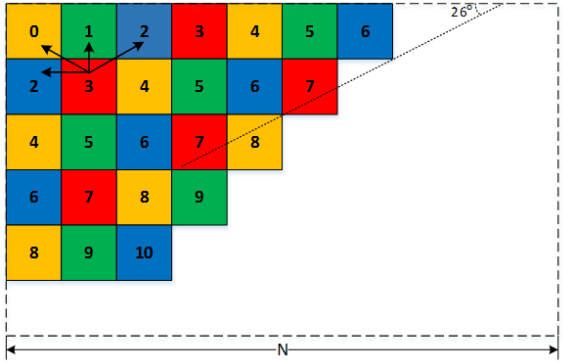
- Filter types are determined by block size and local statistics
  - 4x4 transformed block edges (left and top edges)
    - filter4: read 2 pixels on each side of the edge and modify 1 pixel (p0 and q0) on each side.
    - if it is not high edge, p1 and q1 are also modified.
  - 8x8 transformed block edges (left and top edges)
    - if not in flat area (normal flat), filter4 is used. otherwise, filter8 is used.
    - filter8 is a 7 tap filter. Taps are [1, 1, 1, 2, 1, 1, 1]. Read 8 pixels (4 on each side of the edge), modify 6 pixels.
  - 16x16 and 32x32 transformed block (left and top edges)
    - if not in flat area, filter4 is used.
    - if in normal flat area, but not in extended flat area, filter 8 is used.
    - If in extended flat area, filter16 is used.
    - filter16 is a 15 tap filter. Filter taps are [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1].
    - filter16 read 16 pixels (8 on each side of the edge), modify 14 pixels. Calculation is similar as filter8.



Challenges of VP9 Deblocking Filter Implementation

- Parallel Implementation Challenges:
  - long filter taps and filter processing order cause data dependency between super blocks and between vertical and horizontal edges.
  - In a software or GPGPU implementation, a straightforward threading approach is to use one thread for each Super Block and follow wavefront dependency pattern.
  - wavefront dependency at 64x64 super block granularity cause only 15 kernel thread can be processing in parallel at peak stage for 1920x1080 frame.
- Filter Processing Complexities:
  - filter decision is made for every pixel on the edge.
  - 7 tap and 15 tap filters requires higher memory bandwidth and line buffers.

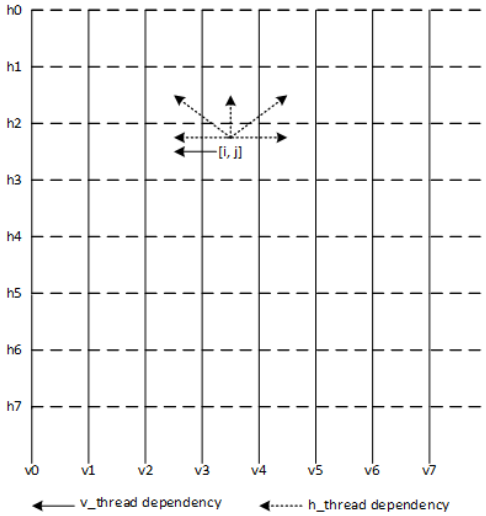




In this diagram, Super Block with the same color and number can be processed in parallel

GPGPU Implementation of the VP9 Deblocking Filter

- GPGPU or Shader implementation of the VP9 decoder is a TTM solution before HW (ASIC) solution is available. Both Intel and Microsoft need to provide such solution for CPUs that don't have HW VP9 decoder support.
- Two level threading approach in Intel's GPGPU VP9 decoder implementation.
  - Super Block level following wavefront order.
  - Super Block is divided into 8x8 blocks. For each 8x8, two threads are used to filter the vertical edge (v_thread) and horizontal edge (h_thread).
  - For 8x8 block (I, j) in a super block:
    - v[i, j] depends on v[i, j-1] (left block)
    - h[i, j] depends on v[i, j], h[i-1, j] (top), v[i, j-1] (left), v[i-1, j-1] (top-left), v[i, j+1] (right), v[i-1, j+1] (top-right).
  - Dedicated HW scoreboard in Intel GPU is used to manage thread dependency.
  - Threads that have most threads dependent on are dispatched first.
- VP9 Decoder Kernel Performance
  - Broadwell GT2 @ 900MHz. 24EUs.
  - Can achieve 1080p60 and 4k30 decoding. Thread utilization is 98%.
  - Deblocking is the most computationally intensive stage. Time of deblocking stage is more than 30%



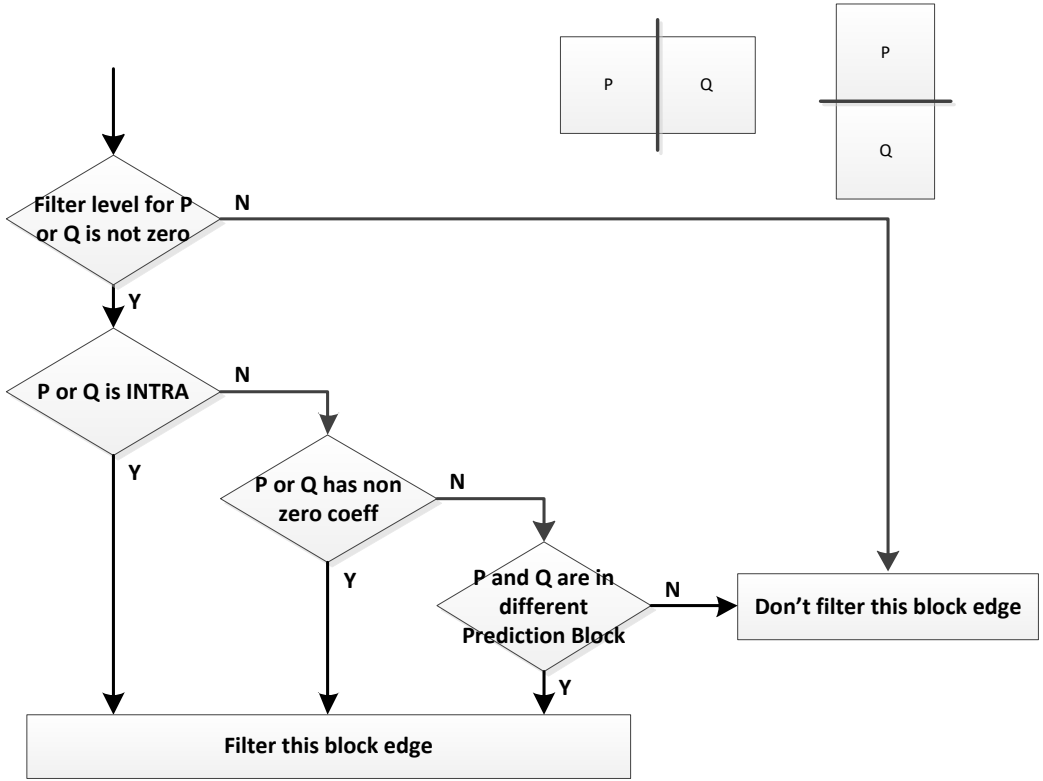
	1920x1080		3840x2160	
	ms	percentage	ms	percentage
IQ/IT	0.847	23.77%	4.634	26.65%
Intra Prediction	1.083	30.39%	4.759	27.36%
Motion Compensation	0.547	15.35%	2.186	12.57%
Deblocking	1.087	30.50%	5.812	33.42%
Total	3.564		17.391	

AV1 Deblocking Filter Improvement

- Parallel_Deblocking: Joint proposal from Intel and Microsoft.
  - https://jointdevelopment.kavi.com/higherlogic/ws/groups/aomediacodec/download/486/AV1ParallelDeblocking.pdf
  - Main goal is to remove the dependency in VP9 deblocking algorithm and improve parallelism.
- Filter Order Change:
  - Filter all vertical edges first for the entire frame followed by filtering horizontal edges of the entire frame.
  - Dependency between horizontal edge and vertical edge within a super block is removed.
- Removed special handing for 32x32 and 64x64 transform block boundaries, so all transform block boundaries are treated consistently.
- Deblocking can be disabled for tile boundaries when loop_filter_across_tile_enabled flag is set to 0 at frame level.
  - With another proposal to add lool_filter_across_tile_enabled flag in frame header.
  - https://jointdevelopment.kavi.com/higherlogic/ws/groups/aomediacodec/download/487/AV1LoopFilterAcrossTileFlag.pdf

AV1 Deblocking Filter Improvement

- **Filter Decision Change**
  - In VP9, if the current transform block has no non-zero coefficients (skipped), its left edge and top edge are not filtered .
  - In AV1, transform blocks on both sides of the edge are checked. Only when both blocks are skipped INTER blocks belong to the same prediction block, filtering for the edge can be bypassed.



AV1 Deblocking Filter Improvement

- **Filter Length Decision Change:**
  - Main goal is to remove filter overlapping between edges.
  - For 4x4 transform blocks, only 2 pixels on each side of the edge are checked to calculate filter mask.
    - VP9 checks 4 pixels, which cause data overlap for 4x4 blocks.
    - For 8x8 and larger block, still check 4 pixels.
  - The de-blocking filter length is selected based on the transform block size from both sides of the edge. The smaller size is used to determine the filter length.
    - In Figure 1, for V1 edge, filter8 is used. For V2 edge, filter4 is used.
  - If rectangle transform is used, transform direction is also considered when determining filter length.
    - In Figure 2, for V1 edge, filter4 is used. For H1 edge, filter8 is used.

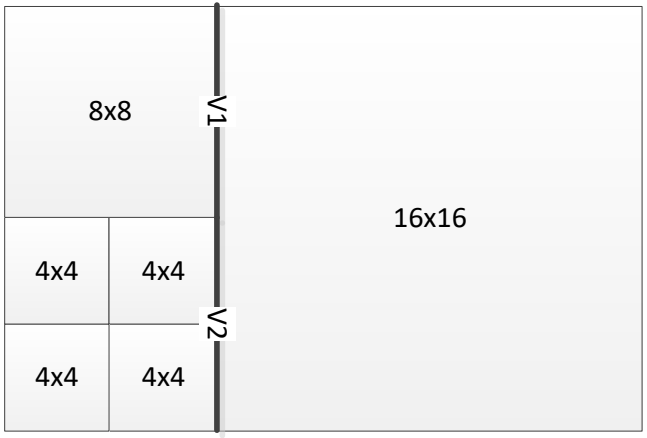


Figure 1

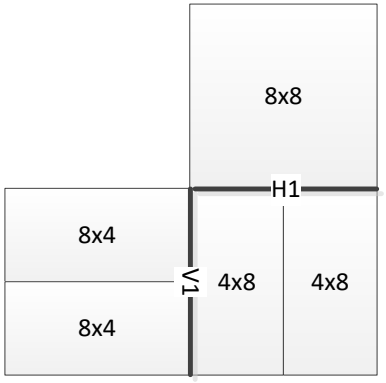


Figure 2

AV1 Deblocking Filter Improvement

- **Remove 15 Tap Filters**
  - 15 tap filter cause higher memory bandwidth requirement and high computation complexity.
  - In HW implementation, 15 tap filter requires line buffer with 8 lines, 24 for Y, U, V planes.
  - Combining with other in-loop filters such as CDEF and Loop Restoration, line buffer required is even higher.
  - Initial proposal was to remove 15 tap filtering completely.
  - However, after extensive visual quality checking, noticeable quality difference can be observed for certain test content when 15 tap filter is removed.
  - Current proposal only removes 15 tap filter for the chroma plane, but still keeps it for the luma plane.

AV1 Deblocking Filter Improvement

- **Status of the proposal:**
  - All implementation has been done with parallel_deblocking experiment name.
  - Proposal has been adopted and enabled by default in AV1 reference software.
- **Objective Quality Test Result with AWCY.**
  - Objective quality impact is negligible.

AWCY Objective-1-fast High Latency

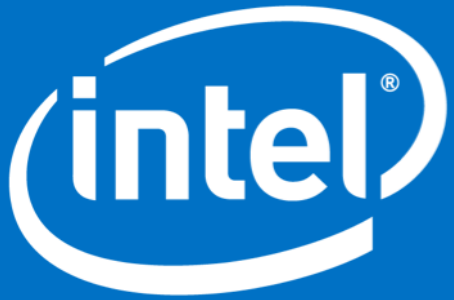
Video	PSNR	PSNR HVS	SSIM	CIEDE 2000	PSNR Cb	PSNR Cr	APSNR	APSNR Cb	APSNR Cr	MS SSIM	VMAF
Average	-0.01	-0.02	-0.02	-0.23	-0.40	-1.05	-0.00	-0.39	-1.03	-0.01	-0.16
1080p	-0.01	-0.03	-0.01	-0.29	-0.68	-1.55	-0.01	-0.68	-1.52	-0.02	-0.11
1080p-screen	-0.01	0.01	0.04	0.09	0.36	0.16	0.00	0.36	0.15	0.10	-0.18
360p	0.00	-0.01	-0.02	-0.23	-0.31	-0.67	0.01	-0.30	-0.64	-0.01	-0.34
720p	-0.01	-0.02	-0.07	-0.29	-0.43	-1.28	-0.01	-0.41	-1.25	-0.07	-0.05

AWCY Objective-1-fast Low Latency

Video	PSNR	PSNR HVS	SSIM	CIEDE 2000	PSNR Cb	PSNR Cr	APSNR	APSNR Cb	APSNR Cr	MS SSIM	VMAF
Average	0.02	0.02	0.01	0.07	0.21	-0.61	0.02	0.19		0.04	
1080p	-0.00	-0.00	0.00	-0.01	-0.12	-1.02	-0.00	-0.14	-0.99	0.01	
1080p-screen	0.11	0.07	0.12	0.35	0.85	0.10	0.11	0.81	0.09	0.09	-0.18
360p	-0.05	-0.00	-0.03	0.06	0.63	-0.29	-0.04	0.61		0.08	0.16
720p	0.06	0.03	0.01	0.04	-0.01	-0.63	0.06	-0.04	-0.63	0.04	-0.29

Summary and Future Work

- Complexity of initial VP9 Deblocking algorithm is very high and not friendly for parallel implementation.
- Parallel_deblocking proposal addresses the data dependency issues in the deblocking algorithm and also reduce its complexity.
- Work is still in progress to integrate with other coding tools proposed for AV1 and decide whether to still keep the 15 tap filter.



Microsoft