

Banding prevention for AVM video codec

Andrey Norkin

*Netflix Inc.,
121 Albright Way, CA
95032, USA
anorkin@netflix.com*

Abstract: This paper discusses sources of banding artifacts present in video codecs using an example of the AVM video codec and proposes solutions that help to significantly reduce these artifacts. The proposed approach shows a reduction of banding observed by visual inspection and a decrease in banding according to an objective banding metric. The approach does not add new tools to the video codec. There is a penalty of 1.32% in the PSNR-YUV BD-rate observed on a range of test sequences. The proposed solutions can also be applied to other hybrid video codecs.

1. Introduction

Banding is an artifact that may be present in video compressed by video codecs. Banding (or contouring) is usually present in flat areas of the image with a slowly changing gradient. When higher frequencies are not present in the compressed video, the slowly changing gradient may be quantized to uniform bands of samples with the same value. Even if the difference between the bands is small, one can often see the resulting bands when the video is displayed. This artifact is present in many video codecs, including AV1 [1].

In some scenarios this artifact can be mitigated by increasing the codec bit depth, such as using an internal and output bit depth of a video codec equal to 10 bits for the 8-bit content [2]. However, even when the internal bit depth of 10 is used, banding can still be visible on some HDR content or when using displays with a bit depth of 8.

A few approaches to address banding have been proposed in the literature. For example, in [3], a quantization approach in the encoder has been proposed, which keeps some transform coefficients in banding prone areas. Another group of approaches applies debanding algorithms in the form of a postprocessing filter [4], [5]. In [6], debanding filtering can be applied as either a post- or in-loop filter. Typically, a debanding filter tries to find areas in the picture that are prone to banding. A detector for banding artifacts may be used, which identifies which areas of the reconstructed picture are likely to contain banding. If banding is detected, the algorithm applies dithering to this area, thus mitigating banding. For example, [6] uses a CAMBI metric [7] to decide whether a debanding filter should be applied to a particular frame, which is signaled using one bit in the frame header. The debanding filter approaches, however, often have a certain cost in terms of the implementation complexity since they assume that certain computational blocks are added to a video decoder. A normative approach to reduce contouring artifacts appearing due to intra-prediction was proposed in HEVC [8].

This paper investigates sources of banding in a video codec, with the example of AVM [9], which is the reference software for the Alliance of Open Media investigations on video

compression beyond AV1. The paper also proposes a solution for the banding to be mitigated with simple and low-cost modifications to certain video codec tools.

In the following, Section 2 describes banding caused by intra-prediction, Section 3 addresses banding caused by transforms, Section 4 addresses banding caused by inter-prediction, and Section 5 by in-loop filtering. Section 6 adds notes on the proposed approaches, Section 7 describes the results and visual quality, and Section 8 discusses future work. Section 9 concludes the paper. Even though the proposed approaches have been implemented in AVM, the author believes that these approaches are applicable to other video codecs based on the hybrid architecture.

2. Banding due to intra prediction

One of the most prominent sources of banding in AVM is intra prediction. In particular, directional intra-prediction is prone to banding and contouring [8]. In some cases of intra-prediction, the difference between the bands can be more than 1. These bands may be clearly visible in high QP settings. The reason for these artifacts is that reference samples used in intra-prediction usually have higher distortion at higher QP values and may also contain block artifacts when the reference samples belong to more than one reconstructed block (intra-prediction uses reconstructed samples before the deblocking has been applied). The directional prediction replicates such an artifact along the prediction direction, and the result looks like an inclined or straight line in the intra-predicted block (see Fig. 1). Adding a transform residual does not always mask this kind of artifact since the transformed residuals at high QP / low-bitrate settings often have few or none non-zero coefficients after quantization. Even if such coefficients are present, they are usually the low-frequency ones and cannot “correct” banding.

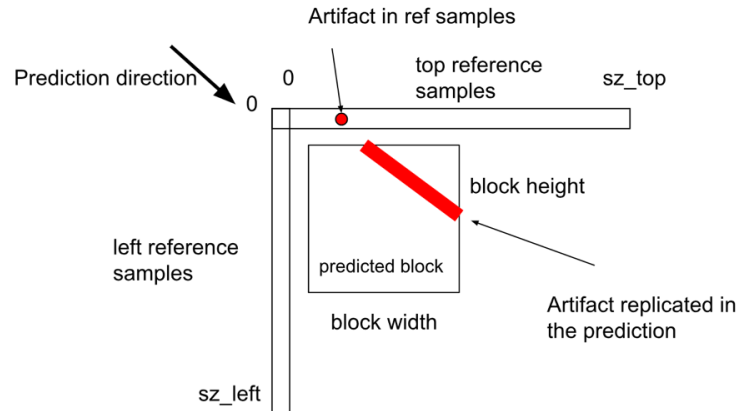


Figure 1: Artifact in the predicted samples may be copied along the prediction direction in the intra-predicted block.

2.1. Finding flat blocks

Some larger banding artifacts in intra-predicted blocks can be mitigated by applying a low-pass filter to reference samples used in intra-prediction. A low-pass filter should be of a sufficient length, so that the artifacts in the reference samples get attenuated. To apply such

filtering only to blocks where the artifacts would be noticeable, the following set of conditions is used to evaluate the block's reference samples for flatness:

$$block_width + block_height \geq T1, \quad (1)$$

$$|above(0) + above(N-1) - 2 * above(N/2 - 1)| < T2, \quad (2)$$

$$|left(0) + left(M-1) - 2 * left(M/2 - 1)| < T3. \quad (3)$$

In the above, $above(i)$ are the above reference samples, $left(i)$ are left reference samples, N is the number of the above reference samples, and M is the number of left reference samples. $block_width$ and $block_height$ are width and height of the block, respectively. $T1$, $T2$, and $T3$ are thresholds that can be set depending on the QP.

2.2. Filtering

When conditions (1)-(3) hold, the filtering applied to the reference samples in AVM is replaced with the following filter. The filter is an adaptive one-dimensional filter that uses restriction on the sample contributions similar to that used in the CDEF filter [10]. An advantage of using such a filter over a regular FIR filter is that it improves details preservation in reference samples and hence can preserve sharper details in the prediction while decreasing the artifacts. When this filter is used, a constraint is applied to filter coefficients, i.e.

$$x'(i) = \sum_k w(k) f(x(i+k) - x(i), S, D), \quad (4)$$

where $x'(i)$ is the sample value after the filtering, $x(i)$ are input sample values, $w(k)$ are filter coefficients, S and D are strength and damping parameters, and $f()$ is the damping function as described in [10]. The effect of the damping function is to reduce the contributions of samples whose values are far from the value of the sample that is being filtered. This helps to reduce blur from the filtering in the cases when a non-flat block is classified as flat. The strength and damping parameters are fixed in the current implementation. The linear filter kernel weights used in the experiment can vary depending on the blocks size and are as follows:

$$w_1(i) = \{1, 1, 2, 2, 2, -14, 2, 2, 2, 1, 1\} / 16, \quad (5)$$

$$w_2(i) = \{1, 1, 1, 1, 1, 1, 1, 1, -16, 1, 1, 1, 1, 1, 1, 1\} / 16. \quad (6)$$

The “linear” filter kernels used in this approach are chosen to not change a signal with a slowly changing gradient. For such signals, one way to avoid changing the sample values at the boundary is to use “antisymmetric padding” for samples at the side of the block boundary. The filtering of top samples uses left adjacent samples for padding. Likewise, filtering of left samples uses top samples for padding. This enables a smooth transition between the left and top reference samples.

2.3. Dithering

The method described above improves both objective compression performance and visual quality. However, this approach cannot eliminate banding that has a difference of one between the bands because steps of size one may still be present in the reference samples.

This remaining banding can be efficiently eliminated by the following approach. In integer arithmetic, when obtaining the sample $x'(i)$ after the filtering, its (integer) value typically needs to be rounded to an integer at the same bit depth as the internal codec bit depth. In integer arithmetic, this can be represented as:

$$x'(i) = (x_{\text{sum}}(i) + ((1 \ll s) \gg 1)) \gg s, \quad (7)$$

where s is the bit shift used to achieve the desired bit depth. Equation (7) adds half of the divisor to the weighted sum obtained as the result of the filtering and performs integer division, represented by the right shift operation $\gg$. In the proposed filter (4), division of the weighted sum by 16 is used, which can be written as:

$$x'(i) = (x_{\text{sum}}(i) + 8) \gg 4 \quad (8)$$

Instead of rounding the samples to the range $[0, 2^N - 1]$ for the N -bit signal, it is proposed to keep the samples after filtering at the higher precision. Then, rounding is applied when the predicted sample value is obtained. However, instead of adding 8 before the shift, a pseudo-random value $\text{rand}(i, j)$ is added. Typically, the pseudo-random value would have an average value of 2^{s-1} and a range of $[0, 2^s - 1]$, which for the s equal to 4 corresponds to the average value of 8 and the range of $[0, 15]$. Adding pseudo-random values with a higher or lower range is also possible depending on whether more or less dithering is desirable. In general, the rounding with the dithering operation is performed as follows:

$$x'(i, j) = (x_{\text{sum}}(i, j) + \text{rand}(i, j)) \gg s. \quad (9)$$

Here i and j are samples' vertical and horizontal coordinates. The value is taken from a predefined array of pseudo-random values according to a sampling function, such as:

$$\text{rand}(i, j) = \text{ra_num}[(j * \text{mcol} + i * \text{mrow}) \& 31] \quad (10)$$

where mrow and mcol are multipliers for the vertical and horizontal positions, respectively. In this example, ra_num array contains 32 values. A fixed array ra_num of size 32 is used. An example of the pseudo-random array ra_num is given below:

$$\text{ra_num}[32] = \{ 12, 0, 6, 4, 15, 12, 6, 11, 11, 9, 15, 12, 9, 1, \\ 14, 4, 7, 5, 12, 13, 13, 14, 1, 11, 8, 5, 9, 6, 3, 4, 2, 2 \}; \quad (11)$$

Application of intra-prediction filtering and subsequent dithering significantly reduces banding at a high QP range. The proposed approach is illustrated in Fig. 2. In the current implementation, the intra filtering and dithering are applied to all three components.

2.4. Restriction of Paeth mode

Turning off the intra-prediction Paeth mode has been found to reduce some remaining banding in intra-prediction.

3. Banding due to transforms

The proposed intra prediction approach helps to fix banding at high QPs / low bitrates, where the residuals in the flat areas are small or zero. However, higher bitrates may still exhibit banding due to transforms. When large transforms have few low-frequency coefficients of low magnitude, the transformed residuals may exhibit banding artifacts too. To attenuate those artifacts, a dithering operation can be applied to the transform output.

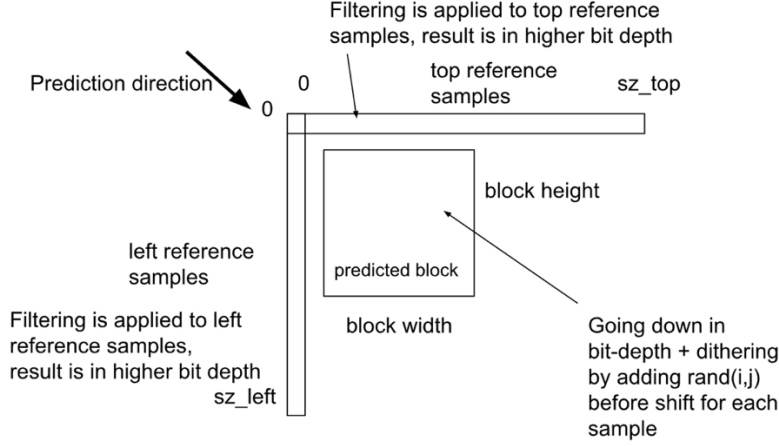


Figure 2: Decreasing the bit depth combined with dithering when the reference samples are copied into the block predictor.

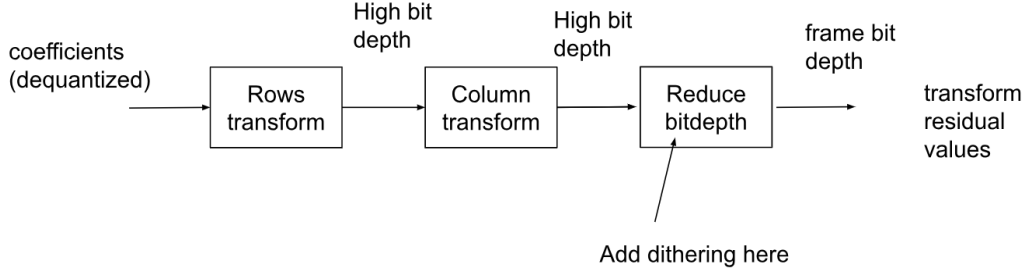


Figure 3: Adding dithering at the transform stage.

When decreasing the bit depth of the transform residual, the dithering operation is done by adding pseudo-random offsets before the right shift / integer division, as in (9). Fig. 3 shows when the dithering is applied.

This operation is only applied to transforms that are not an identity transform (a transform skip), and the following condition decides whether to apply dithering to the transform residual

$$block_width + block_height \geq T4 \quad (12)$$

$$num_nz \leq T5, \quad (13)$$

where num_nz is set to the maximum value of the sum of row and column indices of non-zero coefficients in the block, and $block_width$ and $block_height$ are the block's width and height respectively. Threshold $T5$ is selected as follows:

$$T5 = (block_width + block_height) \gg coeff_num_shift. \quad (14)$$

$T4$ and $coeff_num_shift$ depend on the frame QP since smooth areas, which are prone to banding, typically use smaller block sizes at lower QPs than at high QPs. In the implementation, dithering in transforms is applied to luma and not to chroma.

4. Banding due to interpolation filter in inter-prediction

While dithering in intra-prediction and transforms significantly reduces banding in intra-frames, some banding may appear in the subsequent inter-predicted frames. It has been observed that this is caused to a significant extent by using the SMOOTH interpolation filter. To avoid this, the SMOOTH interpolation filter is turned off in the encoder's RDO process.

5. Banding due to in-loop filters

Loop filters can reduce noise added by the dithering and cause mild banding. To reduce banding due to in-loop filters, dithering has been added to deblocking, constrained directional enhancement filter (CDEF), and loop restoration (Wiener and self-guided) filters of AVM. The dithering is added in a similar way as in the transform and intra-prediction stages. When rounding a filtered sample value to the reconstructed picture bit-depth, dithering is introduced with the operation shown in (9). Further studies will check whether dithering in all three in-loop filters is strictly necessary. For example, the deblocking filter may not need dithering since AVM deblocking preserves details [11].

In the presented results, all AVM in-loop filters use dithering in all three color components, so the results represent a nearly “worst case” scenario from the PSNR BD-rate perspective.

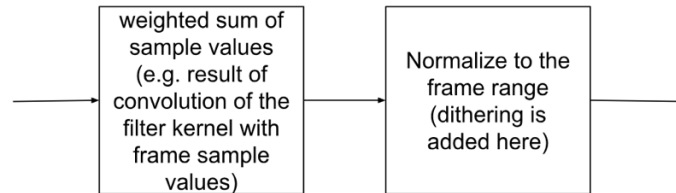


Figure 4: Adding dithering in the filtering operations.

6. Notes on the proposed approaches

There may be interaction between dithering applied in different tools, such as in intra prediction and transforms. Using the same dithering pattern at subsequent stages should be avoided since it may result in noticeable patterns. Instead, different tools may use different sampling functions (10) and / or pseudo-random arrays *ra_num* (11).

Also, different coding tools have different contributions to the overall banding. Perhaps, the strongest decrease in banding at low bitrates is achieved by applying dithering at the intra-prediction stage. Dithering at the transform stage helps to reduce banding at the middle and higher bitrates. These two approaches help to significantly reduce banding. The remaining banding can be reduced by dithering during in-loop filtering.

7. Results

The proposal has been implemented on top of AVM anchor 4.0 [9]. To simplify the implementation, the transform dithering has been done in the following way. If the criteria (12) and (13) are fulfilled, the denoised coefficients are multiplied by 16 for 8-bit content and by 4 for 10-bit content. After the inverse transform is performed at higher bit depth, the rounding is performed with dithering as in the approach described above. In an optimized implementation, the dithering would be applied at the transform rounding stage. A similar approach has been used for the implementation of the intra-prediction module. This is a “proof-of-concept” approach used to avoid implementing rounding for each transform type, transform size, and intra-prediction mode in AVM, including the SIMD.

7.1. Objective performance

The AOMedia CTC version 5 have been used to run the experiments [12]. Random Access (RD) configuration results are reported. The results are obtained based on 55 sequences grouped into classes based on resolution and content type (both natural and synthetic, i.e. animation and gaming, content types are used). Base QP values of [110, 135, 160, 185, 210, and 235] are used in the test. The proposal is run with the parameter `--enable-paeth-intra=0` to disable Paeth mode, whereas the anchor uses `--enable-paeth-intra=1`. The results are shown in Table 1.

One can see that there is a penalty on certain objective quality metrics, such as the 1.32% PSNR BD-rate. The banding metric CAMBI [7] shows a decrease in values, which indicates that the proposed solution decreases banding. Note that the CAMBI metric results are not BD-rates but the average decrease in the banding metric score. For most sequences in the CTC, CAMBI is quite low, and the reduction in average CAMBI is driven by the sequences that with banding in the AVM anchor. The results for four sequences with higher banding in the AVM anchor are shown in Table 2.

The decoding and encoding times show some increase relative to the anchor, e.g. the average of 4% increase in the decoding time on average in Table 1. This increase is likely due to the implementation aspects, discussed in the previous sub-section. One can also notice a somewhat higher BD-rate penalty on chroma components. It is to be investigated whether adding dithering to chroma components is necessary in terms of the visual quality.

Table 1: Results on the AOMedia CTC, Random Access configuration.

Classes	PSNR-Y	PSNR-U	PSNR-V	PSNR-YUV	VMAF	CAMBI	Enc-time	Dec-time
Class A1_4K	1.04%	1.35%	1.56%	1.10%	0.74%	-0.46	110%	104%
Class A2_2K	1.05%	2.32%	3.24%	1.21%	1.05%	-0.43	109%	105%
Class A3_720p	1.85%	5.47%	4.29%	2.15%	1.80%	-2.45	110%	102%
Class A4_360p	1.02%	2.57%	4.77%	1.22%	0.61%	-1.80	108%	101%
Class A5_270p	0.60%	4.97%	2.76%	0.89%	0.25%	-0.79	108%	101%
Class B1_SYN	1.12%	3.26%	3.01%	1.29%	1.23%	-2.10	109%	104%
Overall	1.14%	3.03%	3.24%	1.32%	1.04%	-1.21	109%	104%

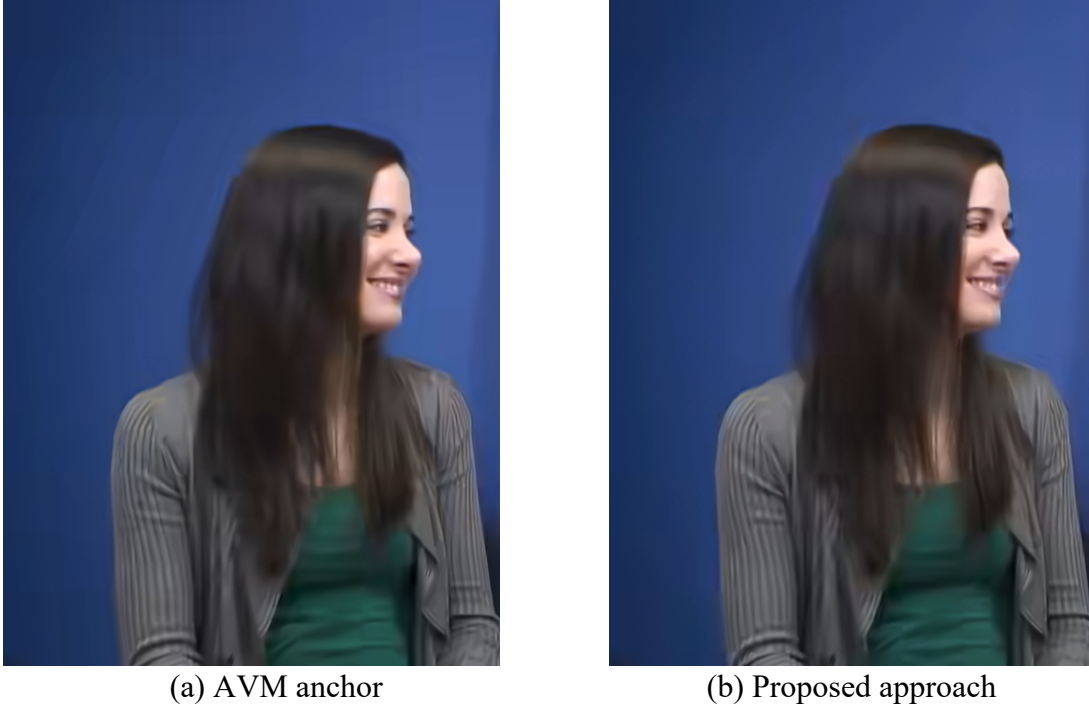


Figure 5: Kristen and Sara QP 210, RA.

Table 2: Results for selected sequences prone to banding from AOMedia CTC.

Classes	PSNR-Y	PSNR-U	PSNR-V	PSNR-YUV	VMAF	CAMBI	Enc-time	Dec-time
Johnny	2.61%	5.59%	8.52%	3.11%	2.81%	-7.8	108%	95%
Kristen and Sara	2.77%	4.00%	4.44%	2.93%	2.87%	-5.74	110%	96%
GlassHalf	1.43%	1.67%	1.90%	1.47%	1.19%	-6.47	110%	103%
SnowMountain	0.34%	3.34%	3.03%	0.52%	-0.28%	-6.35	109%	103%
Average	1.79%	3.65%	4.47%	2.01%	1.65%	-6.59	109%	99%

An advantage of the proposed approach is that no additional big coding tools are added to the video codec. Moreover, the proposed approach would inherently be an integral part of the video codec providing uniform quality across the implementations.

7.2. Visual quality

Figs. 5 and 6 show examples of a reduction in banding artifacts. In particular, Figs. 5 (a) and 6 (a) show the AVM anchor, whereas Figs. 5 (b) and 6 (b) show AVM with the proposed approach. One can notice that banding has been reduced compared to the AVM anchor. It has also been observed that sequences that have bands in the source keep these bands in the encodes with low QP values. This is an advantage over post-filtering approaches since it allows one to preserve the original creative intent. In general, reduction in banding has been consistently observed across the frames of the video sequence due to introducing banding in the coding loop. Hence, the debanded frames are used for predicting subsequent frames which helps avoid bands fluctuation.

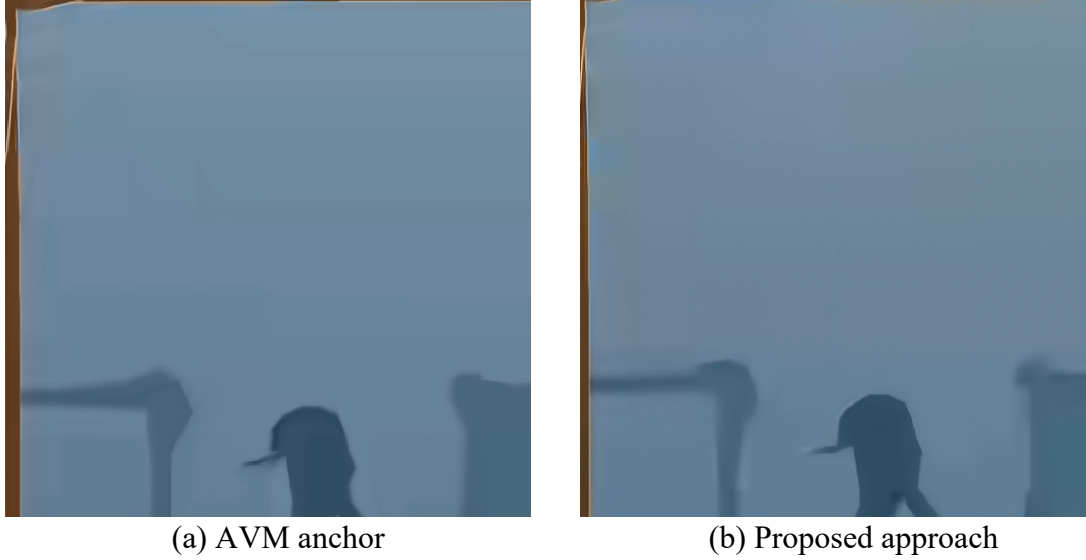


Figure 6: Glass Half QP 235, RA.

8. Discussion and future work

The following improvements may be further investigated. A banding metric, such as CAMBI [7] can be added to the encoder to avoid adding dithering in the cases where there is no visible banding. In this case, dithering would only be applied to sequences where a pre-encode shows banding. Some work can be done to check if the dithering can be turned off in some in-loop filters and in chroma components. Also, the screen content coding needs to be further tuned. Currently, PSNR BD-rate penalty about 3% is observed on SCC, likely due to abundance of flat blocks and low bitrates.

Motion compensated temporal filtering of the alternative reference (alt-ref) frames can increase banding. Typically, the source filtering would only make a difference at higher bitrate, and this can be addressed by reducing the source filtering at higher bitrates.

In addition to that, the following two approaches can alternatively be used. One alternative is to keep the higher bit depth after prediction, e.g. intra prediction. In this case, intra-prediction for the block is done at the higher bit depth and neither rounding nor dithering is done after the prediction is formed. The transformed residual is also applied at the higher bit depth. After the residual is added, dithering is applied as in (9). In-loop filters may also apply dithering as in the previously discussed approach. Such an approach would maintain higher precision and reduce the number of times the dithering is applied. However, higher bit depth would need to be supported during the block reconstruction process, thus requiring more memory in hardware implementations for the reconstructed block.

Yet another alternative is to perform inter and intra prediction, transforms and in-loop filters at a higher bit depth, such as bit depth of 12. The bit depth is decreased and dithering is applied only when the picture is output or saved for reference. This approach is likely to provide even higher performance and quality of the reconstructed picture since most intermediate results are stored at higher precision. In this approach, in-loop filters would

be applied immediately after the block reconstruction, which is typically done in hardware video decoder implementations. However, the line buffer would still need to be increased accordingly. In software implementations that apply the loop filters in a separate pass, the current frame storage would need to be increased to accommodate higher bandwidth in the case of 8-bit video.

9. Summary

In this paper, the sources of banding in the AVM video codec have been investigated, and normative solutions to mitigate banding have been presented. The proposed approach shows a reduction of banding on visual inspection and a decrease of banding according to the debanding CAMBI metric. There is a penalty of PSNR-YUV BD-rate of 1.32% relative to the anchor on a range of sequences when the proposed modifications to the codec are used. The advantage of the approach is that it does not need additional tools in the codec, such as a separate post-filter, and it only requires limited modifications of some video codec blocks. The approach has been implemented in AVM but can be applied to other video codecs based on the hybrid architecture.

References

- [1] “AV1 Bitstream & Decoding Process Specification”,
<https://aomediacodec.github.io/av1-spec/av1-spec.pdf>
- [2] D. Hoang, V. L. Pham, H. Egilmez, L. Guo, Y. Zheng, G. Li, A. Tourapis, “Coding 8-bit Video using 10-bit”, AOMedia document CWG-D088, July 31, 2023.
- [3] N. Casali, M. Naccari, M. Mrak, and R. Leonardi, “Adaptive quantisation in HEVC for contouring artefacts removal in UHD content”, in Proc ICIP 2015, Sept. 2015.
- [4] Z. Tu, J. Lin, Y. Wang, B. Adsumilli, and A. C. Bovik, “Adaptive Debanding Filter”, IEEE Signal Processing Letters, vo.27, pp. 1715-1719, 2020.
- [5] S. Bhagavathy; J. Llach; J. Zhai, “Multi-Scale Probabilistic Dithering for Suppressing Banding Artifacts in Digital Images”, in Proc ICIP 2007, San Antonio, TX, Oct. 2007
- [6] J. Sole, M. Afonso, “A debanding algorithm for AV2”, in Proc. DCC 2023, Snowbird, UT, USA, March 2023.
- [7] J. Sole, M. Afonso, L. Krasula, Z. Li, and P. Tandon, “CAMBI, a banding artifact detector”, Tech Blog <https://netflixtechblog.com/cambi-a-banding-artifact-detector-96777ae12fe2>
- [8] T. K. Tan and Y. Suzuki, “Contouring artefact and solution,” JCTVC-K0139, JCT-VC, Shanghai, CN, Oct. 2012.
- [9] AOMedia Video Model (AVM): <https://gitlab.com/AOMediaCodec/avm/>
- [10] S. Midtskogen, J.-M. Valin, “The AV1 Constrained Directional Enhancement Filter (CDEF)”, In Proc. ICASSP, 2018.
- [11] A. Norkin, “Generalized deblocking filter for AVM”, in Proc. Picture Coding Symposium (PCS), San Jose, CA, Dec. 2022.
- [12] X. Zhao, Z. (R.) Lei, A. Norkin, T. Daede, A. Tourapis, "AOM Common Test Conditions v5.0", AOMedia Document CWG-D103o, https://aomedia.org/docs/CWG-D103o_AV2_CTC_v5.pdf