Title

Efficient Debanding Filtering for Inverse Tone Mapped High Dynamic Range Videos

Permalink

https://escholarship.org/uc/item/6003s4s5

Journal

IEEE Transactions on Circuits and Systems for Video Technology, 30(8)

ISSN

1051-8215

Authors

Song, Qing
Su, Guan-Ming
Cosman, Pamela C

Publication Date

2020-08-01

DOI

10.1109/tcsvt.2019.2928270

Peer reviewed

Efficient Debanding Filtering for Inverse Tone Mapped High Dynamic Range Videos

Qing Song, *Member, IEEE,* Guan-Ming Su, *Senior Member, IEEE,* Pamela C. Cosman, *Fellow, IEEE,*

***Abstract*—When a low or standard dynamic range video is inverse tone mapped to high dynamic range (HDR), there can be banding artifacts in the output HDR video. We design a highly integrated architecture that can detect and alleviate banding artifacts while preserving true edges and details in an extremely efficient way. This is achieved by a 7-tap edge-aware selective sparse filter. Coding artifacts, such as blocky artifacts, can also be reduced by this filter. The filter includes some parameters that depend on the strengths of the banding artifacts. A parameter selection mechanism is presented which considers smoothness of the banding regions and fidelity of the filtering output. The filter yields significant PSNR gain at the regions of artifacts. Subjective tests demonstrate the great quality improvement achieved by the proposed filter, compared to the quality before filtering. The visual quality provided by the filter is better than or similar to that of algorithms which are far more complex.**

***Index Terms*—Sparse filter, high dynamic range, false contour removal, debanding, visual enhancement.**

I. Introduction

THERE is a growing interest in high dynamic range (HDR) videos as well as HDR displays in recent years. HDR videos are represented with 12+ bit depth, i.e., 12 or more bits per color component [1]. They provide a wider range of brightness and a larger color gamut than the traditional 8-bit 100-nit standard dynamic range (SDR) videos. The peak brightness of HDR displays can be over 1,000 nits ($\mathrm{cd/m^2}$). HDR displays can show darker blacks and brighter whites, leading to more details in the displayed images.

However, the current distribution of videos is mostly at 8-bit depth. Although many cameras nowadays can capture 12-bit or even 16-bit videos, videos are quantized to 8 bits for compression and distribution. Videos are also tuned for 100 nits SDR displays which, for example, use gamma encoding [2] and Rec.709 [3] color space. To watch the videos on a 1,000+ nits HDR display, one needs to apply some inverse tone mapping operator (iTMO) [4], [5] to the SDR videos. This mapping is called inverse tone mapping because usually the mapping from HDR content to SDR content is called tone mapping [5]. The iTMO may not be linear. For example, highlights and light sources in an image may be expanded more than other pixels. The iTMO can also include an electro-optical transfer function (EOTF) [1] conversion and color space conversion, if the HDR display uses different EOTF (e.g., Perceptual Quantizer [1]) and color space (e.g., DCI-P3 [6]). The iTMO has been investigated in [7]–[10]. Because it can be content dependent, many distributors send iTMO as metadata to help the 1,000+ nits HDR displays convert the SDR content to HDR content.

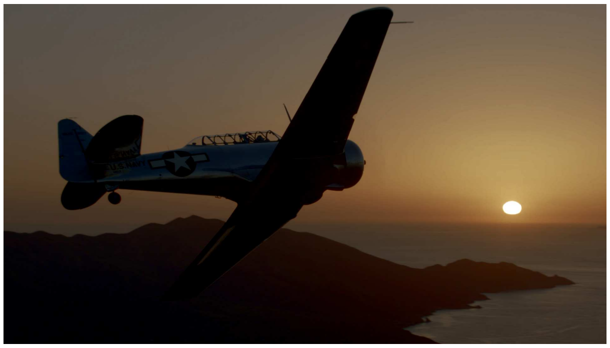
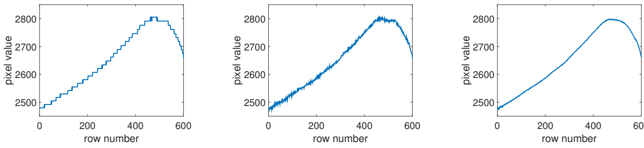


Fig. 1: Banding example.



(a) (b) (c)

Fig. 2: Pixel values of column 1700 of Fig. 1. (a) inverse tone mapped HDR. (b) reference 12-bit HDR. (c) Filtered HDR by the proposed method.

HDR video generated by iTMO sometimes suffers from false contours, also referred to as banding artifacts or ringing artifacts. The artifacts are due to the Mach band effect [11], [12], in which the human visual system enhances step boundaries by undershooting or overshooting at each step boundary. The artifacts occur especially when iTMO is a one-to-one mapping function, because 8-bit SDR video has at most 256 codewords (there are only 220 codewords in Rec. 601 [13] and Rec. 709 [3]), and after mapping, the HDR video also has 256 codewords. According to [1], 12-bit, i.e., 4096 codewords, is necessary to show a banding-free image on a 1,000+ nits display. Lack of codewords in the inverse tone mapped HDR results in the visually annoying banding artifacts.

Fig. 1 shows an inverse tone mapped HDR image with banding artifacts around the sun in the sky. The banding is much more visible if the image is shown on a HDR display. We plot the pixel values of a column in the banding region in Fig. 2a. The pixel values look like staircases, and the stair height is over 10 amplitude levels (codewords) of 12-bit depth in this figure. These large steps appear as false contours.

Q. Song and G.-M. Su are with Dolby Laboratories, Inc., Sunnyvale, CA 94085 (email: qingsong@ieee.org, guanmingsu@ieee.org).

P. C. Cosman is with the University of California, San Diego, La Jolla, CA 92093 (email: pcosman@eng.ucsd.edu).

To remove the banding artifacts, i.e., debanding or decontouring, dithering has been used in the work of Ahn and Kim [14] and the work of Yoo et al. [15], but the output is often not visually pleasant either (e.g, noisy in smooth regions). For stronger banding, the dithering strength has to increase, yielding noisier output. Those works aim to produce the output at the same bit depth as the input image. In our case, however, only a few codewords have been used in the HDR image generated by iTMO, so there is room to generate new codewords to smooth the banding artifacts. In the work of Su et al. [16], this is achieved by linear interpolation in the banding area after the banding width is identified by median filtering. The banding area can be detected by the algorithm in [17]. Some works apply lowpass filters to increase codewords. Daly and Feng [18] proposed to predict and extract false contours by lowpass filtering and quantization. The predicted false contours are subtracted from the image. This method can introduce new false ringing if the predicted false contours are inaccurate, which happens when the banding steps are non-uniform. In the work of Lee et al. [19], false contours are reduced by 1D directional smoothing filters whose directions are orthogonal to the false contours. This method requires high computational complexity to detect false contours. To avoid blurring true edges, the filter size should be variable, which is hardware unfriendly. In the work of Bhagavathy et al. [20], the banding region is first detected by analyzing the neighborhood at multiple scales. New codewords are generated by the expected mean value of the local neighborhood. The performance is good, but the computation is quite intense. Huang et al. [21] detect false contours by checking the eight neighbors. Several conditions are applied to exclude very smooth regions, texture and sharp edges. The contours are then removed by probabilistic dithering followed by lowpass filtering. The method was designed for removing banding in 8-bit SDR images. The conditions used in the banding detection will have to be modified for detecting banding in HDR images, and the conditions may depend on the iTMO. In the work of Niu et al. [22], the bit depth is increased by taking a weighted sum of the pixels in a window, where the weights are adaptive to the content. The window size has to be large enough if this approach is applied for banding removal.

We proposed a selective sparse filter which combines smooth region detection and banding reduction in [23]. The filter removes the banding artifacts, and reduces some coding (such as blocky) artifacts. The properties of the inverse tone mapped HDR are exploited in the filter design. We aim at implementing the filter in hardware, so computational complexity and memory cost are carefully considered. This work is an extension of our preliminary work. The major contributions are: 1) we propose a strategy to select the two content dependent parameters of our filter, and 2) we evaluate the performance of our filter using both objective metrics and subjective testing. Experimental results show excellent performance of the proposed filter.

The overview of the system is shown in Fig. 3. Content tuned for SDR displays is called SDR video, and content tuned for HDR displays is called HDR video. The camera outputs 12-bit SDR video, which is quantized to 8 bits for compression and distribution. A legacy encoder, e.g., AVC [24] or HEVC [25], is used to encode the 8-bit SDR video. We generate the iTMO, which can be content dependent, at the encoder, and send the corresponding iTMO parameters as metadata. How to generate the iTMO is not in the scope of this paper. At the decoder, the SDR bitstream is decoded and can be displayed directly on a SDR display. For HDR displays, the iTMO$^{(8)}$ from metadata is applied to the reconstructed 8-bit SDR video to generate the HDR video. The superscript (8) indicates the input bit depth is 8. Although the inverse tone mapped HDR video is in 12-bit representation, it uses only 256 codewords, because the iTMO is a one-to-one mapping.

To remove banding artifacts, our proposed debanding filter is applied to the inverse tone mapped HDR video at the decoder. There are parameters in the filter which are content dependent. The parameters are solved at the encoder side, where we have access to the 12-bit SDR video and have more computing resources. At the encoder, a reference 12-bit HDR video is generated by applying the iTMO$^{(12)}$ to the 12-bit SDR video. The iTMO$^{(12)}$ is the higher precision version of the iTMO$^{(8)}$. This reference 12-bit HDR video is free of banding because it is represented using the full 4096 codewords. Fig. 2b shows the pixel values of the corresponding reference 12-bit HDR of Fig. 2a. The reconstructed 8-bit SDR video is also available at the encoder. To obtain the same output as the decoder side, we apply iTMO$^{(8)}$ to the reconstructed 8-bit SDR video. Both the reference HDR video and the inverse tone mapped HDR video are used to select the parameters of our proposed debanding filter. The parameters are sent as metadata along with the iTMO parameters.

We notice here if the proposed debanding filter is applied to the reconstructed 8-bit SDR video instead of the inverse tone mapped HDR, the bit depth of the filtering output should be kept at 12-bit, because lower bit depth would quantize out the new created codewords. The iTMO$^{(12)}$ should be used to map the 12-bit SDR filtering output to the 12-bit HDR. However, it is possible that only the iTMO$^{(8)}$ is implemented in hardware of the decoder, i.e., the iTMO has only 256 different entries. In this case, the filter has to be applied to the inverse tone mapped HDR. Besides, the iTMO is probably nonlinear, and can increase the difference between codewords. A banding-free SDR can still yield banding artifacts in the mapped HDR. Therefore, the perceptual quality measurement and the filter parameter selection have to be performed on HDR. Filtering the HDR would be more perceptual-oriented and is thus recommended.

The rest of the paper is organized as follows: In Sec. II we explain our proposed debanding filter, and discuss the parameter selection in Sec. III. Performance evaluation and comparisons of our filter with other debanding algorithms are in Sec. IV, and Sec. V concludes the paper.

II. Proposed edge-aware sparse filter

Banding artifacts usually occur in regions of small gradient, and the artifacts appear as steps. We define a banding step as a group of consecutive pixels which have the same codeword, and the pixel on the left (top) and the pixel on the right (bottom) of this group have different codewords from the group. To

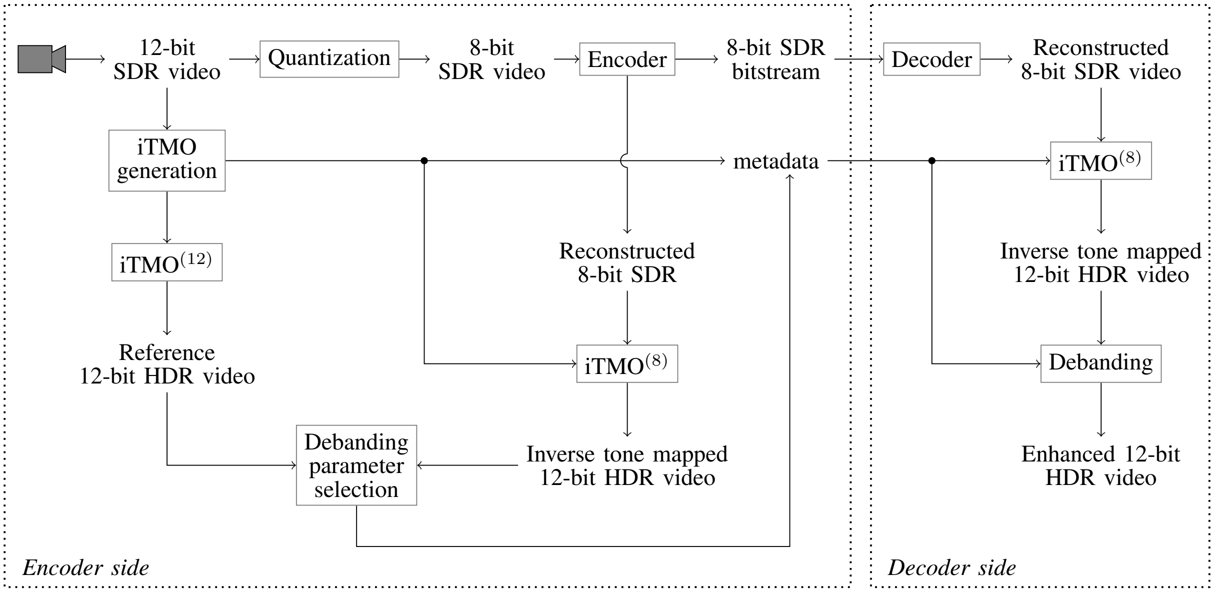
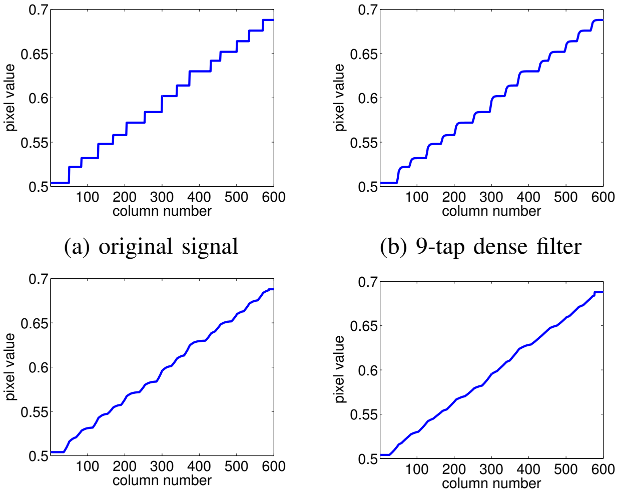


Fig. 3: Overview of system.

remove the artifacts, one can smooth the area by adding more codewords between each banding step. One simple method is to apply a lowpass filter. A traditional dense 2D FIR filter can be represented as: $y[m,n] = \sum_{i=-u}^{u} \sum_{j=-v}^{v} w_{i,j} \cdot x[m+i, n+j]$, where $x[m,n]$ is the input signal at row m and column n, $y[m,n]$ is the corresponding output signal, and $w_{i,j}$ is the filter coefficient. An unweighted lowpass filter has equal coefficients: $w_{i,j} = \frac{1}{(2u+1)(2v+1)}$. This filter averages a total of $(2u+1)(2v+1)$ input pixels centered at $x[m,n]$. The 2D filtering is separable: it is equivalent to sequentially applying a 1D $(2v+1)$-tap horizontal averaging filter and a 1D $(2u+1)$-tap vertical averaging filter, which is much more efficient.

To remove banding, the span of the filter has to be wide enough. Fig. 4a shows a 1D signal with non-uniform steps. The pixel values are normalized to $[0,1]$. Figs. 4b-4d show the outputs of the dense filter with different numbers of taps. The 9-tap filter is unable to remove the false contours; many wide steps are left. The 29-tap filter works better. The banding is almost gone when the number of taps increases to 49.

The dense filter can smooth banding only when pixels on more than one step are involved in the averaging, even if dynamic filter coefficients are allowed. When the span of the filter is not wide enough, many consecutive pixels of the output will have the same codeword, as the pixels taken for averaging are from the same banding step. If the banding step width is uniform and equal to W, the false contours can be completely smoothed out when the span (equivalent to the number of taps) of the dense filter is $2W-1$. The conventional way to implement a filter in hardware is to put each row of pixels into one line buffer (static random-access memory, i.e., static RAM) for vertical filtering. The filtered image would be stored in a frame buffer (dynamic RAM). The actual memory needed for the filter is the line buffers (as working memory) and the frame buffer. The number of line buffers equals the number of filter taps. Owing to the high cost of static RAM, reducing the number of filter taps is highly desired.



(a) original signal

(b) 9-tap dense filter

(c) 29-tap dense filter

(d) 49-tap dense filter

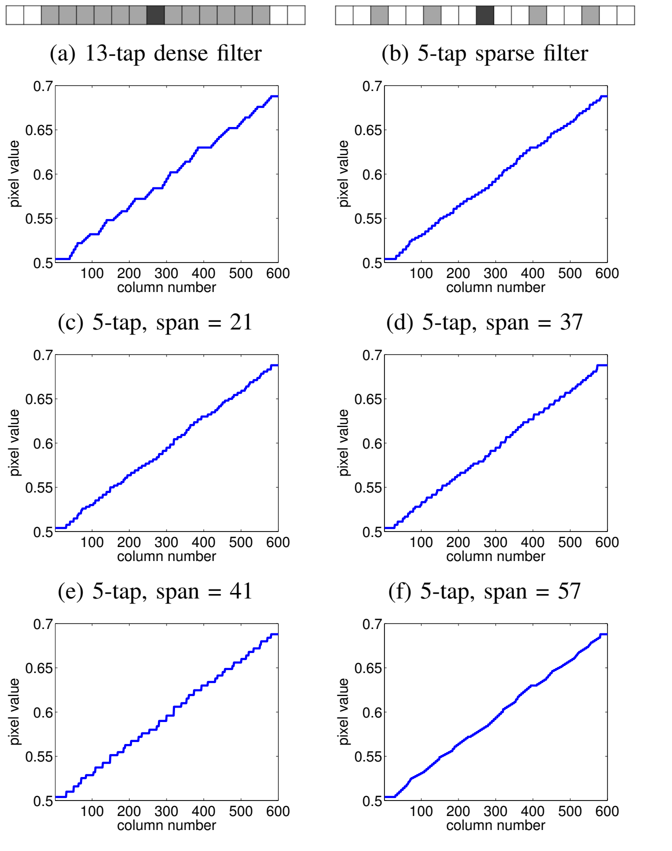
Fig. 4: Performance of dense filter.

A. *Sparse filter*

From the observations above, we learn that to remove banding, the key is to get samples from different steps. A sparse filter does that efficiently. A 1D horizontal sparse FIR filter [26], [27] is defined as:

$$z[m,n] = \sum_{j=-v}^{v} \hat{w}_j x[m, n+s_j], \tag{1}$$

where s_j is the distance from the original pixel to the sampled input signal, and $\hat{w}_j$ is the coefficient of the j-th tap. The number of taps is $2v+1$. Fig. 5b shows a 5-tap horizontal sparse filter with the same span as the 13-tap dense filter in



(a) 13-tap dense filter

(b) 5-tap sparse filter

(c) 5-tap, span = 21

(d) 5-tap, span = 37

(e) 5-tap, span = 41

(f) 5-tap, span = 57

(g) 3-tap, span = 41

(h) 9-tap, span = 41

Fig. 5: Performance of 5-tap sparse filter.

Fig. 5a. The origin is marked by dark gray. The distances between each two neighboring samples are the same. Figs. 5c - 5f show the 5-tap horizontal sparse filtering outputs of Fig. 4a with different spans. The filter coefficients are fixed and equal: $\hat{w}_j = \frac{1}{2v+1}$ for all j. The outputs in Figs. 5d and 5e look relatively smooth. Figs. 5g and 5h show the outputs of 3-tap and 9-tap sparse filters with span 41, respectively. The 3-tap sparse filter creates fewer codewords than the 5-tap filter, so the output looks more jagged than Fig. 5e. The output of the 9-tap sparse filter is smoother than the output of the 5-tap filter, and looks similar to Fig. 4d. The computation is much lighter than the dense filter.

Note that we use a 1D signal as an example here. The 2D sparse filtering can be obtained by applying this filter horizontally and then vertically. The final output is:

$$y[m,n] = \sum_{i=-u}^{u} \tilde{w}_i z[m+t_i, n] = \sum_{i=-u}^{u}\sum_{j=-v}^{v} \bar{w}_{i,j} x[m+t_i, n+s_j], \tag{2}$$

where $\bar{w}_{i,j} = \tilde{w}_i \cdot \hat{w}_j$.

The sparse filter needs fewer taps than the dense filter. A $(2u+1)$-tap vertical sparse filter with any span requires only $2u+1$ line buffers. To filter a 12-bit video with HD resolution (1920×1080), a 7-tap filter requires working memory size of $7 \times 1920 \times 2$ bytes for vertical filtering, which saves 86% of that required by a 49-tap filter. Only one line buffer is needed for horizontal filtering, which is the same as a dense filter. Fewer adders and multipliers are needed for the sparse filtering.

B. *Edge-aware selective filter*

It is clear that sparse FIR filters can help remove false contours, but they can also blur true edges and remove details. To address this issue, we propose to apply the sparse filter selectively, to smooth areas only. Banding is only observed in smooth areas. Also, smoothing a smooth area would not cause much loss of detail even if there were no banding in the area.

The proposed filter includes a horizontal filter and a vertical filter. The two filters will be applied sequentially. The horizontal filter has $2v+1$ taps. For simplicity, the corresponding vertical filter has the same structure as the horizontal filter, with the same number of taps and the same sample locations.

Fig. 6 shows the flowchart of our proposed horizontal filter with 7 taps as an example ($v = 3$). The input image is the inverse tone mapped HDR image. For each pixel $x[m,n]$ in the input image, we sample itself and v pixels on the left and right sides of it. The positions of the sampled pixels are denoted $n + s_j$ where $j \in \{-v, \cdots, -1, 0, 1, \cdots, v\}$ and $s_0 = 0$. We compute the difference between the central pixel $x[m,n]$ and each of the sampled pixels $x[m, n+s_j]$ where $j \neq 0$. If the absolute value of the difference is below a threshold Δ, we determine that the sampled pixel has a similar value to the central pixel. If the central pixel and all the sampled pixels have similar values, we consider the area to be smooth, and replace the central pixel value with the average of the inputs. The averaging takes only $2v-1$ inputs: $x[m, n+s_{-v+1}], \cdots, x[m, n+s_{-1}], x[m,n], x[m, n+s_1], \cdots, x[m, n+s_{v-1}]$. If the difference between the central pixel and any of the sampled pixels is greater than the threshold, there may be edges or texture in the area. Then the averaging is not applied, and the input pixel value remains unchanged: $z[m,n] = x[m,n]$.

For vertical filtering, v pixels on the top and bottom of the central pixel $z[m,n]$ are sampled from the horizontal filtering output. As before, the averaging is applied only when the differences between the central pixel and all the sampled pixels are within the threshold. Only $2v-1$ inputs are used for averaging. The output of the vertical filtering is denoted $y[m,n]$, which is the enhanced HDR in Fig. 3.

This selective filter combines non-smooth area detection and sparse filtering. The decoder will be able to make the decision whether to apply averaging to each pixel, thus no filtering map needs to be sent from the encoder. The filter only requires one line buffer for the horizontal filtering and $2v+1$ line buffers for the vertical filtering. The selection of the threshold Δ for the selective sparse filter is critical to the debanding performance. Note that the filtering process only takes $2v-1$ pixels, and the extra two pixels are for non-smooth area detection, as will be discussed below. Because banding artifacts appear in smooth regions, and would not occur near object boundaries, we do not want to involve pixels near boundaries. This is a major difference from the epsilon-filter [28]. Besides, the proposed filter is more hardware-friendly due to smaller memory usage and fixed coefficient of each filter tap. In the following, we first describe how to select the threshold, then explain why the extra two pixels are necessary in the decision process.

1) Adaptive threshold: The threshold Δ indicates how much difference we will tolerate in the decision process.

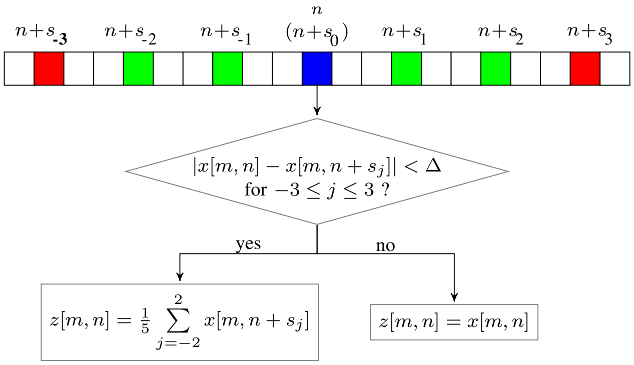


Fig. 6: Flowchart of proposed edge-aware sparse filter.

If Δ is too small, we will only average pixels with small differences, corresponding typically to small areas, so banding artifacts might not be removed. If Δ is too large, the filter could be applied to areas with sharp edges and details, leading to blurred edges and loss of detail.

One important observation of the banding areas is that the codewords of the corresponding pixels in the 8-bit SDR image are very similar to each other. The difference of the 8-bit SDR codewords between neighboring pixels is 1 or 2 most of the time. After inverse tone mapping, the difference between these neighboring pixels shown on a HDR display becomes larger, and that results in the banding artifacts. Since the input of our proposed filter is the inverse tone mapped HDR, Δ can be related to the mapping function.

Assume that the codeword of a pixel in the 8-bit SDR is b where $0\leq b\leq 255$. The corresponding inverse tone mapped HDR codeword is $T(b)$. The difference between two neighboring HDR codewords is denoted $dT(b)=|T(b+1)-T(b)|$. In the inverse tone mapped HDR image, if an area is relatively smooth, we expect the values of nearby pixels to be around $T(b)$. If a pixel is in a textured area, the differences among nearby pixels could be much larger than that. Therefore, we set the threshold Δ to a small number times $dT(b)$.

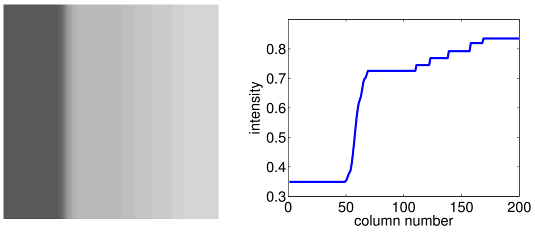
We discuss two commonly used iTMO functions as examples. First, the iTMO is a simple linear mapping: $T(b)=\rho\cdot b+c$, where ρ is positive and constant for the entire image, and c is a constant offset. For example, ρ can be $\frac{2^{12}}{2^8}=16$ for simple bit depth up-conversion. We set the threshold for the entire image to

$$\Delta=\alpha\cdot dT(b)=\alpha\cdot(\rho\cdot(b+1)+c-\rho\cdot b-c)=\alpha\cdot\rho,\quad (3)$$

where α is positive.

Another popular iTMO is the piecewise polynomial [10]. Sometimes the iTMO is non-linear, and is represented by a piecewise polynomial. With $\hat{K}$ segments in total in the iTMO curve, the differential function $dT(b)$ is partitioned into $\hat{K}$ segments. The segment boundary points are denoted p_k, where $1\leq k\leq\hat{K}+1$. The segment slopes can be very different, so different thresholds are needed. When the codeword of the central pixel of the filter inputs ($x[m,n]$) is $T(b)$, the threshold is set to

$$\Delta=f(b)=\alpha\cdot\max_{p_k\leq b<p_{k+1}}\{dT(b)\},\qquad (4)$$



(a) (b)

Fig. 7: Example of banding artifacts.

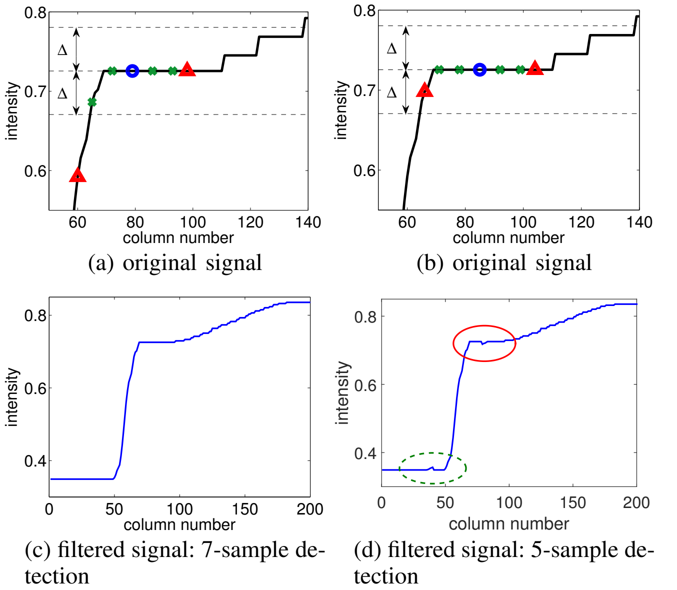
where α is positive. The threshold is set to the maximum differential of the segment multiplied by a factor α. In our tests, for both the linear mapping and the piecewise polynomial, $\alpha=2$ or 3 usually works well.

This method can be extended to other iTMO algorithms. The differential function $dT(\cdot)$ of any one-to-one mapping can be built. The point is that we only allow averaging a few codewords in the filtering process. So the threshold can be set to $\alpha\cdot dT(b)$ when $x[m,n]=T(b)$. Another possible setting for the threshold is α times the maximum codeword differential of the entire image: $\alpha\cdot\max_{0\leq b<255}\{dT(b)\}$.

2) Extra samples for non-smooth area detection: We include $2v+1$ pixels for decision but only take $2v-1$ pixels for filtering. This prevents introducing new false ringing to the output image. We use the patch in Fig. 7a to illustrate; it has an edge between the dark and bright regions, and banding artifacts in the bright region. We plot the intensities of a row of pixels in Fig. 7b. Our goal is to preserve the dark region and the edge, and smooth the banding in the bright part. Using a 7-tap ($v=3$) filter as an example, we set the filter parameters $s_1=-s_{-1}=7, s_2=-s_{-2}=14, s_3=-s_{-3}=17$, and set $\Delta=2H$ where H is the maximum difference between adjacent banding steps. Figs. 8a and 8b show the mid region of Fig. 7b. We want to determine whether to apply the sparse filter to the pixels marked by blue circles. The range of $[x[m,n]-\Delta, x[m,n]+\Delta]$ is marked by dashed lines.

We apply the filter only when all the six samples have values similar to the central pixel. For the pixel marked by the blue circle in Fig. 8a, the difference between it and the leftmost sample exceeds the threshold, so the filtering is not applied. For the pixel marked by a blue circle in Fig. 8b, all the samples marked by green crosses and red triangles are within the threshold. We apply the sparse filtering by averaging only the five central pixels, not all seven pixels. So we exclude the leftmost sample which is an outlier from the averaging. The filtering result is shown in Fig. 8c. The banding artifacts are smoothed, and the edge is well preserved.

If we do not get the two samples marked by red triangles, and determine to apply the filtering as long as the four green cross samples have similar values to the central pixel, a false ringing can be introduced. For the pixel marked by the blue circle in Fig. 8a, the filtering condition would be satisfied. However, the leftmost green cross sample is actually at a transition area. That sample is an outlier, whose value is slightly different from the others though the difference is still within the threshold. The average of the five pixels would be slightly lower than the original value which brings



(a) original signal (b) original signal

(c) filtered signal: 7-sample detection (d) filtered signal: 5-sample detection

Fig. 8: Comparison between 5-sample and 7-sample non-smooth area detection.

an undershoot (marked by the solid red circle in Fig. 8d). Similarly, an overshoot is introduced on the other side of the true edge (marked by the dashed green circle in Fig. 8d). In the image, the overshoot and undershoot appear as faint false ringing.

Therefore, with the extra two samples to probe if there is an edge nearby, we prevent introducing new artifacts into the output image. Also, more details can be preserved with the extra two samples strengthening the condition to apply filtering.

3) Metadata: This selective sparse filter is to be applied at the decoder side and implemented in hardware. The number of line buffers used for filtering needs to be fixed, so the number of taps of the filter has to be fixed. We found that 7 taps in total (i.e., 5 taps for averaging) are usually enough for the sparse filter to remove the banding artifacts. If the banding steps are uniform, a 5-tap 1D sparse filter (no decision process) can create at most 4 new codewords at each banding step (see Appendix). If the banding steps are in 2D and are uniform, we can create at most 16 codewords after applying the sparse filter in both the horizontal and vertical directions. That means the bit depth is increased by 4 (from 8 to 12). If the banding steps are non-uniform, it is possible to create more codewords.

There are several parameters to be determined according to frame content: 1) α in the threshold, and 2) the positions of samples in the sparse filter. In our tests, we found that equidistant samples for averaging works well, i.e., we set $s_j = -s_{-j} = jD$ for $1 \leq j \leq v-1$. For the extra two samples for non-smooth area detection (s_v and s_{-v}), empirically we set $s_v = s_{-v} = \lfloor \frac{2v-1}{2} s_1 \rfloor = \lfloor \frac{2v-1}{2} D \rfloor$, i.e., the distance between $x[m, n+s_v]$ and $x[m, n+s_{v-1}]$ is half of the distance between $x[m, n]$ and $x[m, n+s_1]$. The span of the filter in the averaging process is $2(v-1)D+1$, and the entire span of the filter in the decision process is $2(v-1)D+1+2\lfloor \frac{D}{2} \rfloor$.

In summary, the filter parameters to be determined are

- the threshold factor: α,
- the distance between each two neighboring samples for averaging: D.

The metadata is simple. There is no need to store or transmit a filtering map. A set of parameters is to be determined to smooth all the banding in the image. As can be seen from Fig. 5, the span of the sparse filter is critical. Increasing the span may not always make the signal smoother. In the next section, we will describe how to select the parameters using the reference 12-bit HDR video and the inverse tone mapped HDR video.

III. Parameter Selection

At the encoder, we have more computational resources, and access to the reference 12-bit HDR video. So we can filter with different spans and thresholds, compare with the reference, and select the parameters that yield the best output. One may think that the output which has the minimum distortion from the reference 12-bit HDR video is the best. However, simple metrics, such as mean squared error (MSE) and SSIM [29], are not quite consistent with the visual quality. From our experience and MPEG meetings, the complicated HDR-VDP [30], [31] does not match the perceptual quality very well either. Therefore, we propose a new metric that considers two factors which impact the perceptual quality of the filtered image. We then formulate the problem to select the parameters.

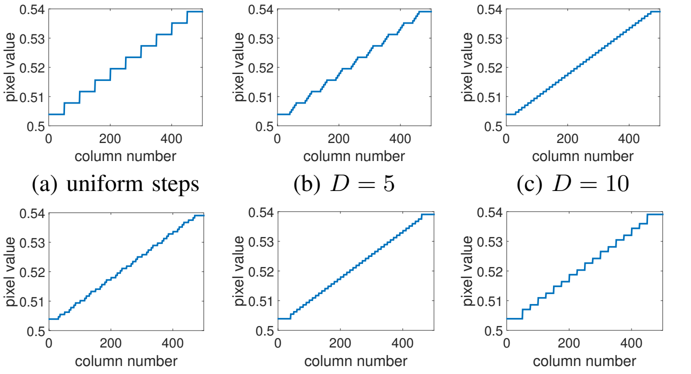
A. *Perceptual distortion*

Generally, we consider two aspects when measuring the quality: (a) how well the banding is smoothed, and (b) how well the true edges and details are preserved.

1) Smoothness after filtering: The pixel value at row m and column n in the inverse tone mapped HDR is denoted $x[m, n]$. We can find banding steps in the inverse tone mapped HDR image in the horizontal and vertical directions individually by two raster scans. We denote a horizontal banding step in row m_0 from column n_1 to n_2 as $\Omega_i = \{m_0, n_1, n_2\}$, where $x[m_0, n]$ is the same for $n_1 \leq n \leq n_2$, and $x[m_0, n_1-1] \neq x[m_0, n_1]$, $x[m_0, n_2+1] \neq x[m_0, n_2]$. The width of Ω_i is denoted L_i^H, where $L_i^H = n_2 - n_1 + 1$. Similarly, we denote a vertical banding step in column n_0 from row m_1 to m_2 as $\Phi_j = \{n_0, m_1, m_2\}$, where $x[m, n_0]$ is the same for $m_1 \leq m \leq m_2$, and $x[m_1-1, n_0] \neq x[m_1, n_0]$, $x[m_2+1, n_0] \neq x[m_2, n_0]$. The width of Φ_j is denoted L_j^V, where $L_j^V = m_2 - m_1 + 1$.

In the averaging process, our proposed debanding filter is a $(2v-1)$-tap sparse filter with fixed equal filter coefficients. The banding steps will be broken into many mini-steps after sparse filtering. We observe that the widest mini-step width after filtering shows how much banding remains. The filtering output is denoted as $y_{D,\alpha}[m, n]$ when D and α are used. The widest width of output mini-steps of Ω_i is denoted $l_i^H(D, \alpha)$, and that of Φ_j is denoted $l_j^V(D, \alpha)$.

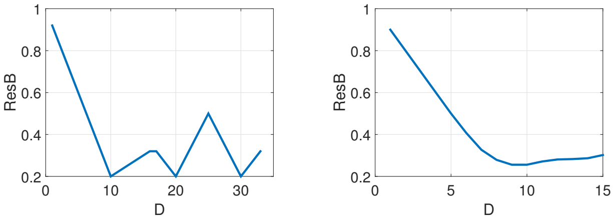
We define a *residual banding ratio* of horizontal banding as $r_i^H(D, \alpha) = \frac{l_i^H(D,\alpha)}{L_i^H}$. Similarly, a residual banding ratio of vertical banding is defined as $r_j^V(D, \alpha) = \frac{l_j^V(D,\alpha)}{L_j^V}$. We pool



(a) uniform steps (b) $D = 5$ (c) $D = 10$

(d) $D = 15$ (e) $D = 20$ (f) $D = 25$

Fig. 9: Uniform steps and filtering outputs with different D.



(a) $ResB$ of Fig. 9a (b) $ResB$ of Fig. 4a.

Fig. 10: Residual banding level of filtering outputs of uniform banding steps in Fig. 9a and non-uniform banding steps in Fig. 4a.

the residual banding ratio of all the banding steps in the image, and compute the *residual banding level* of the whole image as

$$ResB(D, \alpha) = \frac{\sum_i r_i^H(D, \alpha) \cdot L_i^H + \sum_j r_j^V(D, \alpha) \cdot L_j^V}{\sum_i L_i^H + \sum_j L_j^V}. \tag{5}$$

We weight the residual banding ratio of each banding step with the width of the banding because wide banding is usually more visible than narrow banding. The pooling is normalized by the sum of weights: $\sum_i L_i^H + \sum_j L_j^V$. After simplification, we obtain:

$$ResB(D, \alpha) = \frac{\sum_i l_i^H(D, \alpha) + \sum_j l_j^V(D, \alpha)}{\sum_i L_i^H + \sum_j L_j^V}. \tag{6}$$

Note that $0 < ResB(D, \alpha) \leq 1$. Smaller $ResB(D, \alpha)$ means the output is more smoothed.

In the implementation, we exclude banding steps:

- where the 12-bit reference HDR is also flat (pixels in the region have the same codeword);
- the first and the last step of consecutive banding steps. If there are only two consecutive steps in the group, then remove the longer step;
- which are shorter than B pixels. For image resolution 1920×1080, we set B to 7, and for resolution 3840×2160, we set B to 14.

The remaining steps are called *major* banding steps. If there are no major banding steps, i.e., $\sum_i L_i^H + \sum_j L_j^V = 0$, we set $ResB(D, \alpha)$ to 0.

To show the relationship between $ResB$ and the smoothness of the filtered output, we first consider 1D synthesized data with steps of uniform width and uniform codeword difference between adjacent steps. Fig. 9a shows uniform steps with width $W = 50$. We plot the sparse filtering outputs using different filter spans in Fig. 9b-9f. The number of taps for averaging is 5 (i.e., the total number of taps is 7 with the two extra samples for decision). Each banding step in the input data is divided into at most 5 mini-steps. The relationship between D and the widths of mini-steps is in the Appendix.

In Fig. 9, the smoothest output is from $D = 10$ and $D = 20$, where the output mini-step width is uniform. When $D = 5$ (Fig. 9b), the widest mini-step after filtering is 30, and the banding is not smoothed well. When $D = 15$, the widest mini-step width after filtering is 15, which is larger than the widest mini-step width when $D = 10$ (Fig. 9c), and we can observe from Fig. 9d that $D = 15$ yields more jagged output. When D increases to 25, the widest mini-step width after filtering is 25, and the output (Fig. 9f) is coarser than the other outputs. We plot the residual banding level created by different D in Fig. 10a. The residual banding level is consistent with the jaggedness of the filtering output.

If both ends of the signal are ignored, the minima of the residual banding level are obtained when $D = \frac{K'W}{5} + KW$, where W is the width of a banding step, K' is a positive integer and is not a multiple of 5, and K is a positive integer (see Appendix). Averaging different combinations of input codewords may result in the same output, so there are more than one minima of $ResB$. This provides the possibility to achieve the minimum residual banding level for multiple groups of banding steps in an image where the widths of each group are different.

For the example of non-uniform banding steps in Fig. 4a, we plot the residual banding level of the filtering output in Fig. 10b. The minima are at $D = 9$ and $D = 10$, which correspond to span 37 and 41 in Fig. 5. We can see that Figs. 5d and 5e indeed look much better than Fig. 5c, and slightly better than Fig. 5f. We assume Δ is big enough so that all the pixels are filtered.

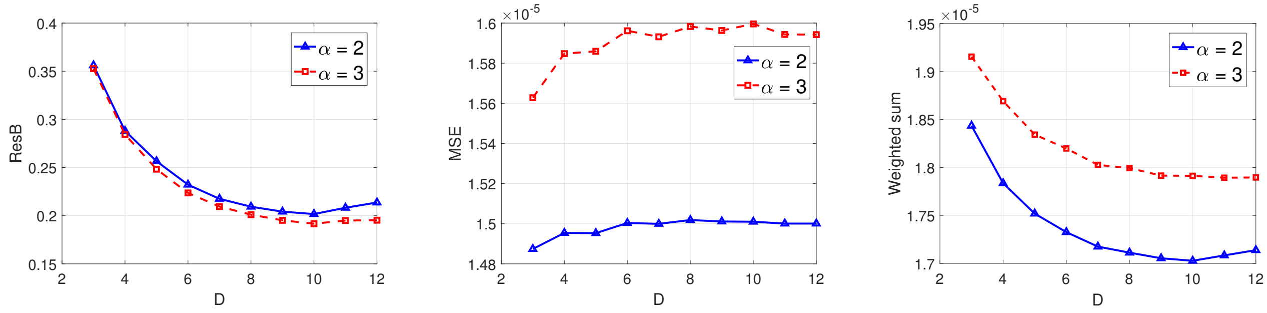
For real data, the banding steps are usually non-uniform. To preserve the decoder hardware efficiency and limit the metadata bit overhead, we do not allow changing the sparse filter span within one image.

2) Fidelity to the reference HDR: Measuring the smoothness after filtering may not be sufficient to represent the overall quality. Detail preservation should be considered. We measure the distortion between the filtering output and the reference 12-bit HDR over the whole image, including both banding and non-banding regions. MSE is used to measure the distortion for simplicity.

We denote the reference 12-bit HDR image as $\hat{x}[m, n]$. The distortion is computed as

$$MSE(D, \alpha) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} (y_{D,\alpha}[m, n] - \hat{x}[m, n])^2, \tag{7}$$

where M and N are the image height and width, respectively. Smaller MSE means higher fidelity to the reference HDR.



(a) ResB

(b) MSE

(c) Weighted sum

Fig. 11: Residual banding level, MSE and weighted sum using $\lambda = 10^{-5}$ of filtering outputs of Fig. 1.

B. Problem formulation

We want to reduce both the residual banding level and the MSE. We define a perceptual metric $J(D, \alpha) = MSE(D, \alpha) + \lambda \cdot ResB(D, \alpha)$, where λ is a weighting factor that controls the trade-off between smoothness and fidelity. We select the filter parameters by

$$\{D^*, \alpha^*\} = \underset{\{D,\alpha\} \in \mathcal{D} \times \mathcal{A} \cup \{0,0\}}{\arg\min} MSE(D, \alpha) + \lambda \cdot ResB(D, \alpha), \tag{8}$$

where $\mathcal{D}$ and $\mathcal{A}$ are pre-defined sets of available candidates of D and α. $\{D, \alpha\} = \{0, 0\}$ means the filter is not applied. MSE is computed between the reference 12-bit HDR ($\hat{x}[m, n]$) and the inverse tone mapped HDR ($x[m, n]$). The residual banding level, $ResB(0, 0)$, is 1 if $\sum_i L_i^H + \sum_j L_j^V > 0$; otherwise, $ResB(0, 0) = 0$.

We find the filter span and the threshold factor by minimizing the weighted sum of the two terms. This formulation is commonly used in denoising and image enhancement algorithms [32]–[36]. This optimization is applied to each frame individually. We found that determining an optimal set of parameters for each scene does not smooth out all the banding, because the banding widths and strengths can be changing dramatically over a scene, especially in fade-in / fade-out scenes. Selecting the parameters of each frame by minimizing the sum of the distortion of a window of frames gives very similar result to the frame-based solution, but the complexity is increased. So we use the frame-based method.

For Fig. 1, the residual banding level, MSE, and weighted sum of the filtering outputs vs. different D are plotted in Fig. 11. The residual banding level decreases as D increases until D reaches 10 for a given α. Larger α yields slightly lower residual banding level, i.e., smoother output, because the threshold in the decision process (Fig. 6) is higher, so more pixels are filtered. However, more details are removed, yielding much higher MSE. Combining the two terms, $D = 10$ and $\alpha = 2$ are selected to minimize the weighted sum.

C. Computational complexity

At the decoder, our proposed filter needs 6 comparisons in the decision process, and one multiplication and 4 additions in the averaging process, for each pixel. A comparison needs two additions. In total, our proposed filter demands one multiplication and 16 additions for each pixel.

At the encoder, each pixel has to be compared with its top and left neighbors to determine the vertical and horizontal banding steps, respectively. For each $D \in \mathcal{D}$ and each $\alpha \in \mathcal{A}$, the proposed filter is applied once. To compute the residual banding level, each pixel is compared with its top neighbor at most once and with its left neighbor at most once. Computing MSE costs one multiplication and two additions for each pixel. For $\{D, \alpha\} = \{0, 0\}$, only MSE is computed. In total, the parameter selection requires $(2|\mathcal{D}||\mathcal{A}| + 1)$ multiplications and $(22|\mathcal{D}||\mathcal{A}| + 6)$ additions for each pixel.

IV. Performance evaluation

We verified our proposed filter using 4 video sequences and 8 images extracted from 5 video clips. All the 8-bit SDR videos are compressed using HEVC at 5.2 Mb/s. The resolution is 1920 × 1080. The EOTF of the inverse tone mapped HDR is Perceptual Quantization (PQ) [1]. The color space is YCbCr. The number of filter taps is set to 7, and $\lambda = 10^{-5}$ in (8) for all the sequences and images. The value of λ is selected such that D^* and α^* obtained by minimizing $J(D, \alpha)$ match the subjectively optimal values for a training dataset of images. Those images are not used in the following experiment. $\mathcal{D} = \{3, 5, 7, 9, 11, 15, 19, 23\}$ and $\mathcal{A} = \{2, 3\}$. Only the luma channel is filtered, but this filter can be applied to a single color component (e.g., luma), or more components (e.g., chroma). It can be also applied to any inverse tone mapping with any EOTF.

The filtering output of Fig. 1 is shown in Fig. 12. The edges of the plane, the sun and the mountains remain sharp. Even the tiny structures on the plane are well preserved. A patch is cropped around the sun where the banding is severe, and is shown in Fig. 13. The banding is smoothed out. Note that banding artifacts in Fig. 13a are much more obvious on a HDR display. Since HDR displays may not be available to all readers, we plot the pixel values of column 1700 of the filtered HDR in Fig. 2c; the signal is much smoother than Fig. 2a.

The proposed filter is also good at removing blocky artifacts. Pixels in regions with blocky artifacts usually have similar values, and our filter smoothes them out. Pixels in those regions can have different gradient directions, so the algorithms using directional features to detect and reduce artifacts may not work well. The proposed filter does not depend on the direction or gradient.

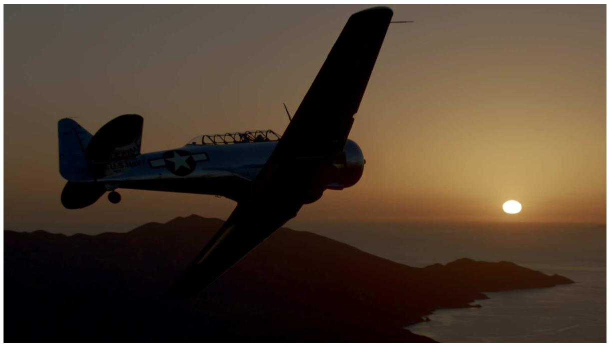
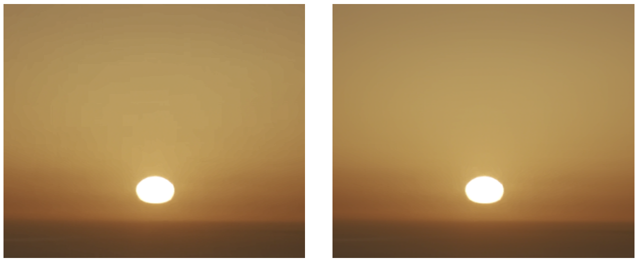


Fig. 12: Filtered (enhanced) 12-bit HDR.



(a) inverse tone mapped HDR

(b) filtered HDR

Fig. 13: Results. Note that banding artifacts in (a) are more noticeable on a HDR screen than on paper.

We implemented the proposed method in C and tested it on a 3.5 GHz CPU with single threading. The average filtering time at the decoder is 40.8 ms (i.e., 24.5 fps). At the encoder, due to the 16 combinations of D and α, and the computation of MSE and ResB, the average processing time is 698.2 ms. The processing time can be greatly reduced if multi-threading is employed. Hardware implementation can also accelerate the proposed filtering.

We compared our proposed filter against three debanding or dithering algorithms:

a) The method of Bhagavathy et al. [20]: This method is designed for 8-bit SDR images. Banding is detected at each pixel by looking for pixels with value $b \pm 1$ in a neighborhood, where $x[m,n] = b$ (un-normalized value), and $x[m,n]$ is the central pixel of the neighborhood. If some criteria are satisfied, $x[m,n]$ is replaced by a weighted sum: $\tilde{y}[m,n] = g_{-1} \cdot (b-1) + g_0 \cdot b + g_1 \cdot (b+1)$, where the weights, g_{-1}, g_0 and g_1, depend on the ratio of pixels with values $b-1$, b, and $b+1$. We modified this method such that it can be used for inverse tone mapped HDR videos. Details can be found in supplementary materials. In total, this method requires 40 multiplications and 72672 additions for each pixel at the decoder. The computation is considerably more complex than our filter. If one wants to move the computation of weights to the encoder, she must transmit the three weights of each pixel to the decoder, equivalent to sending three more images. The overhead would be very high.

b) Bilateral filter [37]: the filtering output is $y[m,n] = \frac{1}{W_p[m,n]} \sum_{i=-u}^{u} \sum_{j=-v}^{v} x[m+i,n+j] w_p[m,n,i,j]$, where $w_p[m,n,i,j] = w_d[i,j] \cdot w_r[m,n,i,j]$, $W_p[m,n] = \sum_{i=-u}^{u} \sum_{j=-v}^{v} w_p[m,n,i,j]$, and where $w_d[i,j] = e^{-\frac{i^2+j^2}{2\sigma_d^2}}$ and $w_r[m,n,i,j] = e^{-\frac{(x[m,n]-x[m+i,n+j])^2}{2\sigma_r^2}}$. $x[m,n]$ is the inverse tone mapped HDR, and the output $y[m,n]$ is the filtered HDR. There are four parameters: the vertical span $2u+1$, the horizontal span $2v+1$, the spatial kernel sigma σ_d and the range kernel sigma σ_r. We set $u = v$, and manually select the span and the two sigma values for each test image, ensuring that banding is removed from the image with the most details preserved by visual inspection. We set v to 14 for 5 out of the 8 test images, and set v to 24 for the other 3 test images. One may determine the parameters at the encoder by solving some optimization problem, but the decoder has to compute the weights. At the decoder, the bilateral filter demands $2(2v+1)^2+1$ multiplications and $3(2v+1)^2$ additions for each pixel (see supplementary materials). For $v = 14$, it costs 1683 multiplications and 2523 additions. If one computes the weights at the encoder and sends the weights of each pixel to the decoder, the overhead would be $(2v+1)^2$ images, which is infeasible. Although there are approximation methods (e.g., Chen et al. [38], Banterle et al. [39]) which can accelerate the implementation of bilateral filtering in GPU, the time and memory usage are still higher than the proposed filter.

c) Gaussian noise injection: we add zero-mean Gaussian noise to the reconstructed HDR. This is a simple method to cover banding and blocky artifacts in images. We select the standard deviation manually for each image so that banding becomes unnoticeable by visual inspection. Note that sometimes it is impossible to cover the banding even with extremely strong noise. The computation is lighter than our method. It costs only one addition for each pixel.

Besides the three filtering / dithering methods, we also compare the inverse tone mapped HDR without debanding. Note the parameters of bilateral filtering and Gaussian noise injection are selected by visual inspection, because this guarantees the best perceptual performance of these methods. The famous debanding algorithm, Daly and Feng [18], is not compared here, because it introduces new false ringing. Also, Bhagavathy et al. [20] claimed better performance than the method of Daly and Feng.

A. *Objective comparisons*

We compute the PSNR gains of the four debanding / dithering schemes over no debanding. The PSNR gains of the 8 inverse tone mapped HDR images are shown in Table I for banding regions and in Table II for non-banding regions. The average gains of our proposed filter, the method of Bhagavathy et al. and bilateral filter in banding regions are almost the same. Note that the method of Bhagavathy et al. and the bilateral filter are dense filters, and are computationally demanding. The three methods barely change non-banding regions, as PSNR gains are close to zero. The Gaussian noise injection has significant PSNR loss in all regions.

The PSNR gains of the 4 video clips are shown in Table III for the banding regions, and in Table IV for non-banding regions. Each test sequence is 10 - 15 sec at 24 fps. We do not include bilateral filtering because its computation is too intense, and we have to adjust the 3 parameters for each frame manually. The proposed method and the method of Bhagavathy

TABLE I: PSNR gain (dB) over no debanding in banding regions of test images.

Image	Bhagavathy et al.	Bilateral	Noise injection	Proposed
1	5.06	3.16	-23.56	3.76
2	1.70	2.14	-13.68	2.06
3	0.54	0.44	-12.96	0.62
4	3.87	4.03	-19.46	3.84
5	1.95	1.95	-17.32	1.99
6	2.50	2.40	-20.86	2.45
7	2.62	2.52	-20.65	2.38
8	3.84	4.01	-18.49	3.35
Average	2.76	2.58	-18.37	2.56

TABLE II: PSNR gain (dB) over no debanding in non-banding regions of test images.

Image	Bhagavathy et al.	Bilateral	Noise injection	Proposed
1	0.07	-1.59	-15.90	-0.05
2	0.10	-0.28	-7.12	0.16
3	0.03	-0.57	-4.07	0.05
4	0.03	0.01	-4.64	0.01
5	0.33	0.03	11.08	0.29
6	0.15	-0.08	-11.57	0.00
7	0.10	-0.56	-14.48	0.06
8	0.08	0.07	-6.59	0.04
Average	0.11	-0.37	-9.43	0.07

et al. achieve almost the same gain in banding regions, and preserve non-banding regions well. The noise injection has severe PSNR loss.

B. Subjective test

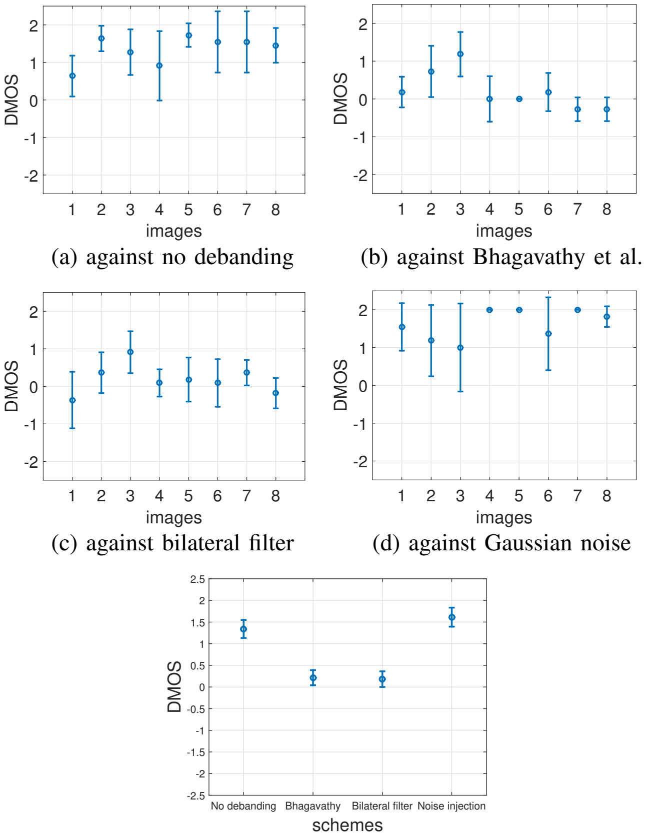
We also evaluate the performance of our debanding filter by a subjective test with 11 observers. The subjective test included two sessions. In the first session, subjects compared images in pairs: one image processed using our proposed debanding filter, and the other is from the method of Bhagavathy et al., the bilateral filter, Gaussian noise injection, or no debanding. The randomized images were labeled A and B. Subjects were given 5 options: “A is much better than B”, “A is slightly better than B”, “A is the same as B”, “A is slightly worse than B”, and “A is much worse than B”. Subjects were also asked to select the reasons why they prefer one to the other one. The possible reasons were: less banding, more details, less noise or other artifacts.

TABLE III: PSNR gain (dB) over no debanding in banding regions of test sequences.

Sequence	Bhagavathy et al.	Noise injection	Proposed
1	3.13	-19.96	3.09
2	1.17	-17.26	1.20
3	0.98	-23.84	0.31
4	0.41	-9.95	0.54
Average	1.42	-17.75	1.29

TABLE IV: PSNR gain (dB) over no debanding in non-banding regions of test sequences.

Sequence	Bhagavathy et al.	Noise injection	Proposed
1	0.14	-12.58	0.06
2	0.04	-6.29	0.02
3	0.22	-18.67	0.18
4	0.11	-6.88	0.07
Average	0.13	-11.11	0.08



(a) against no debanding

(b) against Bhagavathy et al.

(c) against bilateral filter

(d) against Gaussian noise

(e) for all images

Fig. 14: 95% confidence intervals of image comparison DMOS of proposed scheme vs. other schemes.

The second session involved video quality evaluation. Subjects rated the quality of each video sequence individually on a 5-point scale: “5 - excellent”, “4 - good”, “3 - fair”, “2 - poor”, and “1 - bad”. Four schemes were included: no debanding, our proposed method, the method of Bhagavathy et al., and Gaussian noise injection. We again asked subjects to select reasons for their ratings if they rated the quality below good. The possible reasons were: banding, loss of details, too noisy or other spatial artifacts, and temporal flickering.

Before the formal test, we ran a training session to ensure the subjects were familiar with the procedure and the rating system. The whole experiment took about one hour with several breaks. The visual testing was conducted on a Dolby Pulsar 4,000 nits HDR monitor.

1) Image comparison: The difference mean opinion score (DMOS) is computed between our proposed filter and the other schemes. Positive (negative) numbers mean our proposed debanding filter works better (worse) than the other scheme. We plot the DMOS and the 95% confidence intervals (CIs) in Fig. 14. The CIs for the comparison against no debanding are above zero for 7 out of 8 images, which shows our proposed filter improves the quality effectively.

For the comparison against the method of Bhagavathy et al., our proposed method shows advantage for two images. For image 2, eight subjects prefer our proposed filter to the method of Bhagavathy et al. because the output of our filter has less banding. Two subjects favor the method of Bhagavathy et al. due to detail preservation. One subject thinks there is no difference. For image 3, ten out of eleven subjects prefer ours due to less banding. The other one prefers the method of Bhagavathy et al. slightly because it shows more details. We plot the pixel intensities of one row of image 3 in Fig. 15. The output of our filter is smoother than the output of the method of Bhagavathy et al. Some pixels in the output of Bhagavathy et al. (Fig. 15b) are not smoothed, because the reconstructed 8-bit SDR pixel values in the neighborhood are two codewords from the value of the central pixel. Therefore, no neighborhood satisfies the given criteria in [20]. In order to make the method of Bhagavathy et al. work for this case, one has to modify the criteria in [20]. The weighting would become more complicated. For the other six images, we cannot reject the null hypothesis that the two methods achieve the same performance.

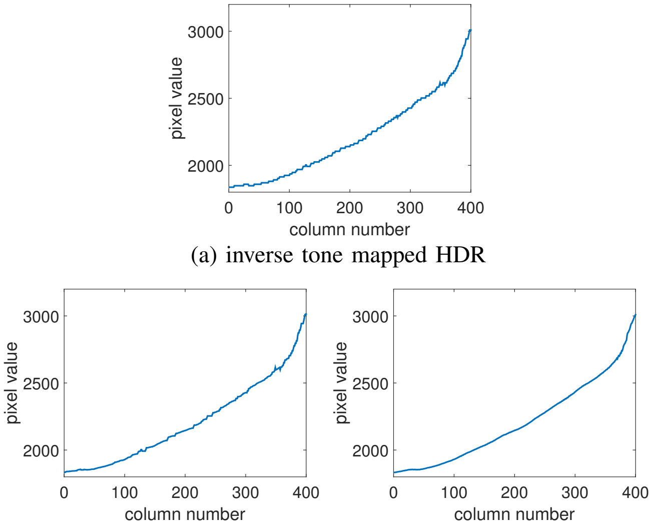
Our proposed filter is slightly better than the bilateral filter for two images. The reason is that our proposed filter preserves more details. That is because the range kernel sigma of the bilateral filter, σ_r, is the same for all pixels, regardless of the iTMO. The range kernel sigma has to be large enough to smooth out banding over the entire image, whereas some areas which need smaller σ_r are blurred. Making σ_r a function of the iTMO is a possible way to improve the bilateral filter, but it requires further study and is not in this paper's scope. The bilateral filter achieves the same performance as our proposed filter for the other 6 images.

Our proposed filter outperforms Gaussian noise injection for 7 out of 8 images. Most subjects think the latter is too noisy. For the other image, subjects have different preferences, so we cannot reject the possibility that the the two schemes achieve the same debanding effect. According to the averaged DMOS, the advantage of our filter over Gaussian noise injection is even larger than the advantage over no debanding.

When we pool the DMOS of all the test images, our proposed filter outperforms all the other schemes. In Fig. 14e, CIs are computed using the scores of all the images. The CIs are all above zero. Our proposed filter has significant advantage over no debanding and Gaussian noise injection, where the average DMOS is 1.34 and 1.61. Our proposed filter also shows a slight advantage over the method of Bhagavathy et al. and bilateral filtering by 0.22 and 0.18.

Contrary to the objective comparison in Table I, the proposed filter outperforms the method of Bhagavathy et al. and the bilateral filter, mainly because the subjects were requested to evaluate the overall quality of the images, while the objective comparison only measures the banding region. Besides, the number of codewords generated by the proposed filter makes the output signal sufficiently smooth. Although the two compared methods (which are dense filters) gives slightly higher PSNR gain by generating more codewords in some images, there is no visual improvement in smoothness compared to our proposed filter.

2) Video quality evaluation: We compute the mean opinion score (MOS) of each video sequence (Table V). Our proposed filter and the method of Bhagavathy et al. yield good quality on the average. No debanding and Gaussian noise injection are poor in quality.



(a) inverse tone mapped HDR

(b) output of Bhagavathy et al.

(c) output of proposed filter

Fig. 15: Pixel values of row 435 of image 3.

TABLE V: MOS of video sequences.

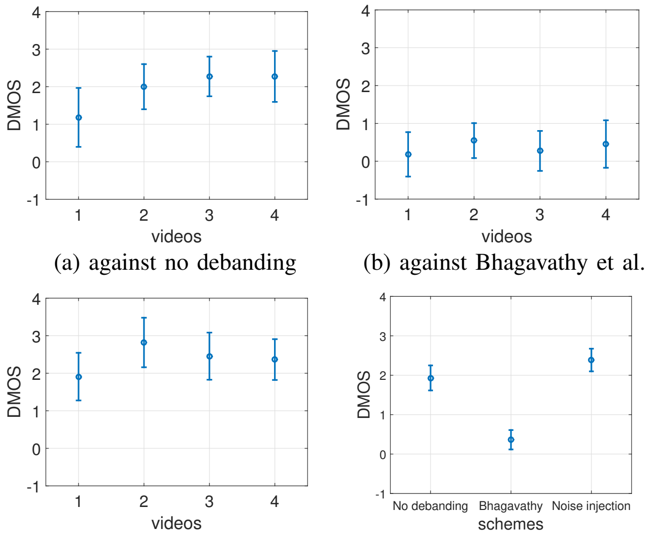
Sequence	No deband	Bhagavathy	Noise injection	Proposed
1	2.91	3.91	2.18	4.09
2	2.36	3.82	1.55	4.36
3	2	4	1.82	4.27
4	1.91	3.73	1.82	4.18
Average	2.30	3.86	1.84	4.23

We compute the DMOS of our proposed filter versus other schemes and plot the 95% CIs in Fig. 16. It is clear that our proposed filter performs much better than no debanding and Gaussian noise injection for all the test sequences. We cannot reject the null hypothesis that our proposed method performs the same as the method of Bhagavathy et al. for 3 out of 4 sequences. For Sequence 2, our proposed method has a small advantage over the method of Bhagavathy et al.

When we pool the DMOS of all the test sequences, our proposed filter wins over all the other schemes. All the CIs of Fig. 16d are above zero. We even have an advantage over the method of Bhagavathy et al.

During the rating of no debanding, banding artifacts were reported 95% of the time (Table VI). This is reduced to 18% when our proposed filter is rated, while banding is reported in the output of Bhagavathy et al. 43% of the time. That means our proposed filter is more effective at removing banding. The detail preservation of the two methods is quite similar, as detail loss was reported 7% and 9% of the time for the proposed method and the method of Bhagavathy et al., respectively.

When rating Gaussian noise injection, subjects disliked the noise, and reported banding 75% of the time. In the image comparison between our proposed filter and Gaussian noise injection, banding is reported in the noise injection scheme only 14% of the time. This is because banding is not masked well in moving pictures, especially when banding is moving. The high frequency noise does not make banding in low temporal frequency invisible.



(a) against no debanding (b) against Bhagavathy et al.

(c) against Gaussian noise (d) for all sequences

Fig. 16: 95% confidence intervals of video DMOS of proposed scheme vs. other schemes.

TABLE VI: Percentage of time when artifacts are reported.

Schemes	No deband	Bhagavathy	Noise injection	Proposed
Banding	95%	43%	75%	18%
Detail loss	2%	9%	0	7%
Noise	0	0	95%	0
Flickering	5%	9%	14%	5%

Flickering was reported only 5% of the time for the proposed filter, which is the same as that of no debanding (Table VI). The proposed filter does not increase temporal incoherence.

V. Conclusion

We proposed an edge-aware selective sparse filter to remove banding artifacts and reduce coding artifacts in inverse tone mapped HDR videos. The filter combines non-smooth area detection and filtering, and is able to preserve edges and details. No banding map or filtering map is required to store at, or transmit to, the decoder. We proposed to select the parameters of the filter by minimizing a perceptual distortion metric at the encoder. The parameters can be sent to the decoder as metadata. This filter can be implemented and executed in hardware efficiently at the decoder, while providing visual quality that is better than or equal to that of algorithms which are far more complex. Subjective tests demonstrate significant advantage over the simple Gaussian noise injection.

Appendix

For uniform banding steps, a 5-tap unweighted sparse filter (no decision process) with equidistant samples can create at most four new codewords at each banding step in one direction. The proof is as follows.

The input uniform banding signal is represented as:

$$x[m,n] = p + \lfloor \frac{n-n_0}{W} \rfloor \cdot H, \tag{9}$$

where $W \geq 2$ is the width of each banding step, and $H > 0$ is the difference of codewords between adjacent steps (Fig. 17). Now we derive the output of horizontal filtering at $n_0 \leq n \leq n_0 + W - 1$ where $n_0 \geq 2$. The four samples entering the filter in addition to the sample at $x[m,n]$ are: $x[m,n-2D] = p - aH$, $x[m,n-D] = p - bH$, $x[m,n+D] = p + cH$, and $x[m,n+2D] = p + dH$, where $a,b,c,d \in \mathbb{N}^0$. We will prove two properties: 1) $a-1 \leq d \leq a+1$, and 2) $\lfloor \frac{a}{2} \rfloor \leq b \leq \lceil \frac{a}{2} \rceil$.

1) $a-1 \leq d \leq a+1$: from (9), we obtain:

$$x[m,n-2D] = p + \lfloor \frac{n-2D-n_0}{W} \rfloor \cdot H. \tag{10}$$

Since $x[m,n-2D] = p - aH$, we obtain:

$$\begin{aligned} p - aH = p + \lfloor \frac{n-2D-n_0}{W} \rfloor \cdot H, && (11)\\ \Rightarrow -aW \leq n - 2D - n_0 \leq -aW + W - 1, && (12)\\ \Rightarrow n - n_0 + aW - W + 1 \leq 2D \leq n - n_0 + aW. && (13) \end{aligned}$$

Add n to the inequality (13):

$$2n - n_0 + (a-1)W + 1 \leq n + 2D \leq 2n - n_0 + aW. \tag{14}$$

Since n is in the range: $n_0 \leq n \leq n_0 + W - 1$, we obtain:

$$\begin{cases} n + 2D \geq 2n - n_0 + (a-1)W + 1 \geq n_0 + (a-1)W + 1 \\ n + 2D \leq 2n - n_0 + aW \leq n_0 + 2W - 2 + aW \end{cases} \tag{15}$$

Combining both these inequalities with (9) and the fact that $x[m,n]$ is non-decreasing, we obtain:

$$p + \lfloor a - 1 + \frac{1}{W} \rfloor \cdot H \leq x[m,n+2D] \leq p + \lfloor 2 - \frac{2}{W} + a \rfloor \cdot H. \tag{16}$$

Since $x[m,n+2D] = p + dH$,

$$p + \lfloor a - 1 + \frac{1}{W} \rfloor \cdot H \leq p + dH \leq p + \lfloor 2 - \frac{2}{W} + a \rfloor \cdot H. \tag{17}$$

As $W \geq 2$, we obtain

$$a - 1 \leq d \leq a + 1. \tag{18}$$

2) $\lfloor \frac{a}{2} \rfloor \leq b \leq \lceil \frac{a}{2} \rceil$: from (13), we obtain:

$$\frac{-n + n_0 - aW}{2} \leq -D \leq \frac{-n + n_0 - aW + W - 1}{2}. \tag{19}$$

Add n to the inequality (19):

$$\frac{n + n_0 - aW}{2} \leq n - D \leq \frac{n + n_0 - aW + W - 1}{2}. \tag{20}$$

Since $n_0 \leq n \leq n_0 + W - 1$:

$$n_0 - \frac{aW}{2} \leq n - D \leq n_0 - \frac{aW}{2} + W - 1. \tag{21}$$

Combining (21) with (9) and the fact that $x[m,n]$ is non-decreasing, we obtain:

$$p + \lfloor \frac{-a}{2} \rfloor \cdot H \leq x[m,n-D] \leq p + \lfloor -\frac{a}{2} + 1 - \frac{1}{W} \rfloor \cdot H. \tag{22}$$

Since $x[m,n-D] = p - bH$,

$$\begin{aligned} & p + \lfloor \frac{-a}{2} \rfloor \cdot H \leq p - bH \leq p + \lfloor -\frac{a}{2} + 1 - \frac{1}{W} \rfloor \cdot H \\ & \Rightarrow -\lfloor -\frac{a}{2} + 1 - \frac{1}{W} \rfloor \leq b \leq -\lfloor -\frac{a}{2} \rfloor. \end{aligned} \tag{23}$$

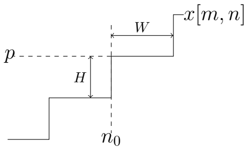


Fig. 17: Input uniform signal.

TABLE VII: All the possible combinations of b, c and d when $a = 2K+1$ where $K \in \mathbb{N}^0$.

b	c	d	Output	Conditions of D
K	K	$2K$	$p - \frac{1}{5}H$	$0 < D - KW < \frac{W}{3}$
$K+1$	K	$2K$	$p - \frac{2}{5}H$	$0 < D - KW < \frac{W}{2}$
K	K	$2K+1$	p	$\frac{W}{4} < D - KW < \frac{W}{2}$
K	$K+1$	$2K+1$	$p + \frac{1}{5}H$	$\frac{W}{3} < D - KW < \frac{2W}{3}$
$K+1$	K	$2K+1$	$p - \frac{1}{5}H$	$\frac{W}{3} < D - KW < \frac{2W}{3}$
$K+1$	$K+1$	$2K+1$	p	$\frac{W}{2} < D - KW < \frac{3W}{4}$
K	$K+1$	$2K+2$	$p + \frac{2}{5}H$	$\frac{W}{2} < D - KW < W$
$K+1$	$K+1$	$2K+2$	$p + \frac{1}{5}H$	$\frac{2W}{3} < D - KW < W$

Since $W \geq 2$, we obtain:

$$\lfloor \frac{a}{2} \rfloor \leq b \leq \lceil \frac{a}{2} \rceil. \tag{24}$$

Similarly, we can prove that

$$\lfloor \frac{d}{2} \rfloor \leq c \leq \lceil \frac{d}{2} \rceil. \tag{25}$$

When a is odd, there are 8 possible combinations of b, c and d that satisfy (18), (24) and (25). The combinations are shown in Table VII where a is represented as $2K+1$ with $K \in \mathbb{N}^0$. When a is even, there are only 5 possible combinations of b, c and d that satisfy the properties. The combinations are shown in Table VIII where $a = 2K$ for $K \in \mathbb{N}^0$. "Output" in the tables means the filtering output, $\frac{1}{5}\sum_{-2}^{2} x[m, n+jD]$. The output has 5 possible values: p, $p - \frac{2}{5}H$, $p - \frac{1}{5}H$, $p + \frac{1}{5}H$ and $p + \frac{2}{5}H$. Therefore, a 5-tap sparse filter with fixed equidistant sample spacing can generate at most four new codewords when used on equi-width banding steps.

The range of D has to satisfy the conditions listed in Tables VII and VIII so that there are $n \in [n_0, n_0 + W - 1]$ that can achieve the combination. The ranges of D overlap, so some values of D can generate as many as four new codewords. Table IX shows the output codewords and the corresponding widths of the mini-steps for different ranges of D. For simplicity, D' is used to represent $D - KW$. The widths of mini-steps are computed by determining the range of n for each combination of the input codewords. Zero width means the codeword cannot be generated by this range of D. In most of the circumstances, four new codewords can be generated by the filter. Fewer than 4 new codewords will be created only when $D = KW + \frac{K'}{q}W$ where $q = 4$ or 3 and $K' \in \mathbb{Z}$. Note that D is an integer, so these values of D can be achieved only when W or K' is a multiple of q. The table indicates that D and $D + KW$ yield exactly the same filtering output. It also indicates that the width of the output mini-steps is greater than or equal to $\frac{W}{5}$, where the minima occur at $D = \frac{K'W}{5} + KW$, where $K' \in \mathbb{Z}$ and K' is not a multiple of 5.

TABLE VIII: All the possible combinations of b, c and d when $a = 2K$ where $K \in \mathbb{N}^0$.

b	c	d	Output	Conditions of D
K	$K-1$	$2K-1$	$p - \frac{2}{5}H$	$-\frac{W}{2} < D - KW < 0\ (K > 0)$
K	K	$2K-1$	$p - \frac{1}{5}H$	$-\frac{W}{3} < D - KW < 0\ (K > 0)$
K	K	$2K$	p	$-\frac{W}{4} < D - KW < \frac{W}{4}$
K	K	$2K+1$	$p + \frac{1}{5}H$	$0 < D - KW < \frac{W}{3}$
K	$K+1$	$2K+1$	$p + \frac{2}{5}H$	$0 < D - KW < \frac{W}{2}$

References

[1] S. Miller, M. Nezamabadi, and S. Daly, "Perceptual signal coding for more efficient usage of bit codes," in *Annual Technical Conference Exhibition, SMPTE 2012*, Oct 2012, pp. 1–9.

[2] "ITU-R Recommendation BT.1886: Reference electro-optical transfer function for flat panel displays used in HDTV studio production," March 2011.

[3] "ITU-R Recommendation BT.709-6: Parameter values for the HDTV standards for production and international programme exchange," June 2015.

[4] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*, 2nd ed. Morgan Kaufmann, 2010.

[5] F. Dufaux, P. L. Callet, R. Mantiuk, and M. Mrak, *High Dynamic Range Video: From Acquisition, to Display and Applications*. Academic Press, 2016.

[6] "SMPTE recommended practice: D-Cinema quality - reference projector and environment," *SMPTE RP 431-2:2011*, pp. 1–14, April 2011.

[7] A. G. Rempel, M. Trentacoste, H. Seetzen, H. D. Young, W. Heidrich, L. Whitehead, and G. Ward, "Ldr2Hdr: on-the-fly reverse tone mapping of legacy video and photographs," in *ACM SIGGRAPH*, 2007.

[8] F. Banterle, P. Ledda, K. Debattista, and A. Chalmers, "Inverse tone mapping," in *Proc. the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, New York, NY, USA, 2006, pp. 349–356.

[9] P.-H. Kuo, C.-S. Tang, and S.-Y. Chien, "Content-adaptive inverse tone mapping," in *Proc. IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Nov 2012, pp. 1–6.

[10] Q. Chen, G.-M. Su, and P. Yin, "Near constant-time optimal piecewise LDR to HDR inverse tone mapping," in *Proc. SPIE*, vol. 9404, 2015, pp. 94 040O–11.

[11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson, 2007.

[12] H. Foley and M. Matlin, *Sensation and Perception*, 5th ed. Psychology Press, 2009.

[13] "ITU-R Recommendation BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," March 2011.

[14] W. Ahn and J.-S. Kim, "Flat-region detection and false contour removal in the digital TV display," in *Proc. IEEE International Conference on Multimedia and Expo*, July 2005, pp. 1338–1341.

[15] K. Yoo, H. Song, and K. Sohn, "In-loop selective processing for contour artefact reduction in video coding," *Electronics Letters*, vol. 45, no. 20, pp. 1020–1022, September 2009.

[16] G.-M. Su, T. Chen, P. Yin, and S. Qu, "Guided post-prediction filtering in layered VDR coding," U.S. Patent 8,897,581 B2, Nov. 25, 2014.

[17] G.-M. Su, S. Qu, and S. Daly, "Adaptive false contouring prevention in layered coding of images with extended dynamic range," U.S. Patent 8,873,877 B2, Oct. 28, 2014.

[18] S. J. Daly and X. Feng, "Decontouring: prevention and removal of false contour artifacts," in *Proc. SPIE*, vol. 5292, 2004, pp. 130–149.

[19] J. W. Lee, B. R. Lim, R.-H. Park, J.-S. Kim, and W. Ahn, "Two-stage false contour detection using directional contrast and its application to adaptive false contour reduction," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 179–188, Feb 2006.

[20] S. Bhagavathy, J. Llach, and J. Zhai, "Multi-scale probabilistic dithering for suppressing banding artifacts in digital images," in *Proc. IEEE International Conference on Image Processing*, vol. 4, Sept 2007, pp. IV–397 – IV–400.

[21] Q. Huang, H. Y. Kim, W. J. Tsai, S. Y. Jeong, J. S. Choi, and C. C. J. Kuo, "Understanding and removal of false contour in HEVC compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2016.

TABLE IX: Widths of output mini-steps after filtering for different ranges of $D' = D - KW$ where $K \in \mathbb{N}^0$.

Range of D'	Widths of mini-steps at each codeword				
	$p - \frac{2}{5}H$	$p - \frac{1}{5}H$	p	$p + \frac{1}{5}H$	$p + \frac{2}{5}H$
$0 < D' < \frac{W}{4}$	D'	D'	$W - 4D'$	D'	D'
$D' = \frac{W}{4}$	D'	D'	0	D'	D'
$\frac{W}{4} < D' < \frac{W}{3}$	D'	$W - 3D'$	$4D' - W$	$W - 3D'$	D'
$D' = \frac{W}{3}$	D'	0	$4D' - W$	0	D'
$\frac{W}{3} < D' < \frac{W}{2}$	$W - 2D'$	$3D' - W$	$W - 2D'$	$3D' - W$	$W - 2D'$
$D' = \frac{W}{2}$	0	$3D' - W$	0	$3D' - W$	0
$\frac{W}{2} < D' < \frac{2W}{3}$	$2D' - W$	$2W - 3D'$	$2D' - W$	$2W - 3D'$	$2D' - W$
$D' = \frac{2W}{3}$	$2D' - W$	0	$2D' - W$	0	$2D' - W$
$\frac{2W}{3} < D' < \frac{3W}{4}$	$W - D'$	$3D' - 2W$	$3W - 4D'$	$3D' - 2W$	$W - D'$
$D' = \frac{3W}{4}$	$W - D'$	$3D' - 2W$	0	$3D' - 2W$	$W - D'$
$\frac{3W}{4} < D' < W$	$W - D'$	$W - D'$	$4D' - 3W$	$W - D'$	$W - D'$
$D' = W$	0	0	W	0	0

[22] Y. Niu, X. Wu, and G. Shi, "Image enhancement by entropy maximization and quantization resolution upconversion," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4815–4828, Oct 2016.

[23] Q. Song, G. M. Su, and P. C. Cosman, "Hardware-efficient debanding and visual enhancement filter for inverse tone mapped high dynamic range images and videos," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 3299–3303.

[24] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[25] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[26] Y. Neuvo, C.-Y. Dong, and S. Mitra, "Interpolated finite impulse response filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 563–570, Jun 1984.

[27] Y. C. Lim, H. K. Kwan, and W.-C. Siu, *Trends in Digital Signal Processing: A Festschrift in Honour of A.G. Constantinides*. Pan Stanford, 2015.

[28] M. Matsumoto and S. Hashimoto, "Noise reduction and edge enhancement based on band-pass epsilon-filter," in *2009 International Conference on Information Management and Engineering*, April 2009, pp. 111–115.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[30] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 40:1–40:14.

[31] M. Narwaria, R. Mantiuk, M. P. D. Silva, and P. Le Callet, "HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images," *Journal of Electronic Imaging*, vol. 24, pp. 24 – 24 – 3, 2015.

[32] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, Feb 1997.

[33] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, "Variable bandwidth image denoising using image-based noise models," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–7.

[34] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM SIGGRAPH 2008 Papers*, 2008, pp. 67:1–67:10.

[35] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," in *ACM SIGGRAPH 2006 Papers*, 2006, pp. 646–653.

[36] M. Nikolova, "Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers," *SIAM Journal on Numerical Analysis*, vol. 40, no. 3, pp. 965–994, 2002.

[37] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, Jan 1998, pp. 839–846.

[38] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," in *ACM SIGGRAPH 2007 Papers*, ser. SIGGRAPH '07. New York, NY, USA: ACM, 2007.

[39] F. Banterle, M. Corsini, P. Cignoni, and R. Scopigno, "A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain," *Computer Graphics Forum*, vol. 31, no. 1, pp. 19–32, February 2012.

Qing Song (S'15-M'18) received the B.Eng. degree in Automation from Tongji University, Shanghai, China, in 2011, and the M.S. and Ph.D. degrees in Electrical Engineering from University of California, San Diego, CA, in 2013 and 2017, respectively. She joined Dolby Labs, Sunnyvale, CA, in 2017. Her research interests include image / video processing, compression and transmission.

Guan-Ming Su is the senior manager of Dolby Labs, Sunnyvale, CA. Prior to this he has been with the R&D Department, Qualcomm, Inc., San Diego, CA; ESS Technology, Fremont, CA; and Marvell Semiconductor, Inc., Santa Clara, CA. He is the inventor of 70+ U.S. patents and pending applications. He is the co-author of 3D Visual Communications (John Wiley & Sons, 2013). He served as an associate editor of Journal of Communications; associate editor in APSIPA Transactions on Signal and Information Processing, Chair of APSIPA Industrial Publication Committee, and Director of review board and R-Letter in IEEE Multimedia Communications Technical Committee. He also served numerical chair positions in multiple IEEE and ACM conferences. He is a Senior member of IEEE. He obtained his Ph.D. degree from University of Maryland, College Park.

Pamela C. Cosman (S'88-M'93-SM'00-F'08) obtained her B.S. with Honor in Electrical Engineering from Caltech in 1987, and her Ph.D. from Stanford University in 1993. Since 1995 she is with the department of Electrical and Computer Engineering at the University of California, San Diego, where she is currently a Professor. Her past positions include Director of the Center for Wireless Communications (2006-2008), ECE Department Vice Chair (2011-2014), and Associate Dean for Students (2013-2016). Her research interests are in image and video compression and processing, and wireless communications. Dr. Cosman's recent awards are the 2016 UC San Diego Affirmative Action and Diversity Award, and 2017 Athena Pinnacle Award. She was the Editor-in-Chief (2006-2009) of the IEEE Journal on Selected Areas in Communications.