Adaptive Debanding Filter
Zhengzhong Tu, Jessie Lin, Yilin Wang, Balu Adsumilli, and Alan C. Bovik, Fellow, IEEE

Abstract—Banding artifacts, which manifest as staircase-like color bands on pictures or video frames, is a common distortion caused by compression of low-textured smooth regions. These false contours can be very noticeable even on high-quality videos, especially when displayed on high-definition screens. Yet, relatively little attention has been applied to this problem. Here we consider banding artifact removal as a visual enhancement problem, and accordingly, we solve it by applying a form of content-adaptive smoothing filtering followed by dithered quantization, as a post-processing module. The proposed debanding filter is able to adaptively smooth banded regions while preserving image edges and details, yielding perceptually enhanced gradient rendering with limited bit-depths. Experimental results show that our proposed debanding filter outperforms state-of-the-art false contour removing algorithms both visually and quantitatively.

Index Terms—Debanding, false contour, compression artifact, dithering, post-processing

I. INTRODUCTION

Recent years have witnessed significant advancements of video compression technologies. The implementation of modern video coding standards, such as H.264/AVC [1], HEVC [2], VP9 [3], and AV1 [4] have greatly benefited high-quality video streaming applications over bandwidth-constrained networks. Despite the improved capabilities of video codecs, debanding artifacts remain a dominant visual impairment of high-quality, high-definition compressed videos. Banding artifacts, which present as sharply defined color bands on otherwise smoothly-varying regions, are often quite noticeable, in part because of the tendency of visual apparatus to enhance sharp gradients, as exemplified by the Mach bands illusion [5]. To further optimize the perceptual quality of compressed user-generated videos [6], [7], developing ways to detect and reduce banding artifacts is a problem of pressing interest.

Prior approaches to banding reduction may be categorized in three ways. If debanding is attempted on the source content before encoding, it is a pre-processing step [8]–[11]. However, these methods have generally been developed for decontouring heavily quantized pictures rather than on compressed/banded video content. A second approach is to apply in-loop processing, whereby the quantization process is adjusted inside the encoder to reduce banding effects [12], [13]. The third approach, post-filtering, has been most widely studied, since it offers maximum freedom for decoder implementations, i.e., the design of post-filters can be relatively unconstrained and flexible. Most banding removal algorithms follow a two-step procedure: first, banding regions are detected and located in the source video frame; then, spatially local filtering is applied to reduce the banding artifacts with dithering sometimes incorporated. For the banding detection stage, some methods [10], [14]–[18] exploit local features, either pixel- or block-wise, such as the image gradient, contrast, or entropy, to measure potential banding statistics. Other methods utilize image segmentation techniques [19], [20]. Either way, banding artifacts are subsequently suppressed by applying low-pass smoothing filters [10], [15], [16], dithering techniques [14], [18], [21], [22], or combinations of these [17], [20].

We deem post-filtering a better approach to handle banding artifacts, since it can be performed outside of the codec loop, e.g., in the display buffer, hence offering maximum freedom of design. Moreover, state-of-the-art adaptive loop filters like those implemented in VP9 or HEVC have not been observed to suppress or have any effect on banding artifacts [17], [19]. Here we propose a new adaptive debanding filter, which we dub the AdaDeband method, as a post-processing solution to reduce perceived banding artifacts in compressed videos. Unlike many prior debanding/decontouring models, we recast the problem differently, as a reconstruction-requantization problem, where we first employ an adaptive interpolation filter along each banded region to estimate the “ideal” plane at a higher bit-depth, then we re-quantize the signal to 8-bit via a dithering technique to suppress quantization error, yielding a visually pleasant enhancement of the banding region. Fig. 1 exemplifies VP9-compressed banding artifacts and the corresponding AdaDeband-filtered output. We demonstrate that our proposed method outperforms other recent debanding methods, both qualitatively and quantitatively.

The rest of the paper is structured as follows. We present the details of the proposed adaptive debanding filter in Section II while evaluative results are given in Section III. Finally, Section IV concludes the paper.
II. ADAPTIVE DEBANDING FILTER

Fig. 2 illustrates a schematic of AdaDeband, which is comprised of three modules. First, a banding detector is deployed to localize the banding edges and band segments with pixel precision. A content-aware size-varying smoothing filter is then applied to reconstruct the gradients within band segments at a higher bit-depth, while preserving image texture details. The final step involves re-quantizing the reconstructed smooth regions to SDR 8-bit resolution using a dithering process to reduce quantization error, yielding a perceptually pleasant debanded image.

A. Banding Region Detection

We utilize the Canny-inspired blind banding detector with same parameters proposed in [23] for banding edge extraction. Only the first two modules, pre-processing and banding edge extraction, are performed to obtain a banding edge map (BEM). For convenience, we restate some definitions from [23]: after self-guided pre-filtering, pixels having gradient magnitudes less than $T_1$ are labelled as flat pixels (FP); pixels with gradient magnitudes exceeding $T_2$ are marked as textured pixels (TP). The remaining pixels are grouped into a candidate banding pixel (CBP) set, based on which the BEM is extracted. Connected-component (CC) labeling is applied on the set of non-textured pixels ($FP \cup CBP$), thereby generating a band map (BM). Band edges (BE) define the boundaries of adjacent bands (B); i.e., the bands (B) are framed by band edges (BE), as shown in Fig. 3. In this way we define and extract two component sets, BEM and BM, which together compose all the banded regions of a given video frame.

B. Banding Segment Reconstruction

Banding artifacts usually occur on regions of small (but non-zeros) gradients, usually appearing as bands separated by steps of color and/or luminance. As depicted in Fig. 3, pixels lying within a band have very similar colors and luminances. A simple method of restoring an ‘original’ smooth local image is to apply a low-pass filter (LPF) to interpolate across bands. A traditional 2D-LPF may be formulated as:

$$J(i,j) = \sum_{k \leftarrow -K}^{K} \sum_{\ell \leftarrow -L}^{L} w_{k,\ell} I(i-k,j-\ell),$$

where $I(i,j)$ are input pixel (luminance or color) values at spatial locations $(i,j)$, and $w_{k,\ell}$ are the filter coefficients, where $\sum_{k,\ell} w_{k,\ell} = 1$. Here we use the simple moving average filter ($w_{k,\ell} = 1/(2K+1)(2L+1)$) as the smoothing LPF, although one may use any other smoothing filter, such as a Gaussian-shaped filter, or even a nonlinear device such as a median filter.

To effectively smooth band discontinuities, the span of LPF should be adequately wide relative to the band width. In Fig. 3(a), for example, it is only necessary to apply a small LPF on the pink band, whereas a larger LPF would be required for the blue segment. Thus, we apply a dynamic way of determining the filter size for each band, or segment of a band, if the width of the band varies along its axis, based on the extracted BEM and BM. Assume $B_j$ to be an exemplar band, which is framed by detected band edges, $BE_{i-1}$ and $BE_i$, as depicted in Fig. 3(a). For all the pixels $I(m,n)$ located within band $B_j$, the spatial extent of the LPF is defined in terms of the ratios of the band area to the lengths of the adjacent BEs. In the unusual instance where a band is enclosed by a single BE, then the ratio is scaled by four. Both cases are expressed here:

$$l(m,n) = \begin{cases} 4 \times |B_j|/|BE_{k}|, & k \in N_{B_j} \quad \text{if} \quad |N_{B_j}| = 1 \\ \max_k |B_j|/|BE_{k}|, & k \in N_{B_j} \quad \text{if} \quad |N_{B_j}| > 1 \end{cases},$$

where $|S|$ is the cardinality of the pixel set $S$ (area of band, or length of band edge), and $N_{B_j}$ denotes the set of band edges that enclose $B_j$. Finally, define the space-varying radius of the LPF window at $(m,n)$ to be half of $l(m,n)$:

$$h(m,n) = \max\{1, \lfloor (l(m,n) - 1)/2 \rfloor \}.$$  

We have ensured that the span of the LPF adapts to the local geometries of banded areas, as shown in Fig. 3(a). Nevertheless, it must also be recognized that the LPFs may process visually important textures or object boundaries while smoothing large banding regions. This kind of content blurring is not acceptable. Accordingly, we constrain the LPF not to include any TP in the sampling window, which is achieved by recursively halving the filter size (as in Fig. 3(b)):

$$\text{repeat } h(m,n) := \max\{1, [h(m,n)/2]\}$$

until $\forall (i,j) \in H^h(m,n), (i,j) \notin TP$

where $H^h(m,n) = \{(m+x,n+y)\}, x = -h,\ldots,-h, y = -h,\ldots,h$, is the set of indices of the (square) filter window centered at $(m,n)$, with linear dimensions $2h + 1$.

The above filter-size-determining process is performed on every banded pixel in set CBP to generate a window-size map.
for further deployment of size-varying LPFs. We observed that, however, the estimated window-size map tends to be noisy due to the potential unrobustness of banding detector. Thus, a median filter is applied to further “denoise” the estimated window-size map, based on which LPFs are then conducted via Eq. (1).

### C. Requantization with Dithering

Dithering techniques are used in a variety of ways in the design of debanding algorithms. Some methods, for example, only apply filtering without dithering to remove false contours [10], [15], [16]. Other methods apply only a very small amount of dither, either by adding random noise (noise-shaping) [12], [18], [21], [22], or by stochastic shuffling of pixels [14], [17], but without any smoothing filter. Among those that combine filtering with dithering, Baugh [20] proposed to apply dithering after smoothing on banded regions, while the authors of [17] suggested probabilistic dithering prior to average filtering. Here we will show that dithered re-quantization very effectively ameliorates banding artifacts arising from compression.

Fig. 2 shows an example of the effects of processing banded areas with and without dithering, respectively. It may be observed that dithered quantization on the reconstructed banded regions is able to reduce re-quantization error, yielding a pixel distribution similar to the original, while direct quantization without dithering still retains bands. Formally, randomized (dithered) quantization [24] may be expressed as:

$$P(i, j) = Q[Q^{-1}(i, j) + \mathcal{N}(i, j)],$$  \hspace{1cm} (5)

where $\mathcal{J}(i, j)$ is the filtered image calculated via Eq. (1), $\mathcal{N}(i, j)$ is a 2D noise image, and $Q^{-1}$ is a N-to-8-bit quantizer.

It should be noted that the pattern of the noise image $\mathcal{N}(i, j)$ used for dithering will shape the textures of the resulting debanded regions, hence the type of noise must be carefully selected. We demonstrate several well-known noise patterns and their corresponding dithered outcomes, both visually and quantitatively, in Fig. 3. Among commonly used methods [27]–[29], we chose to employ Gaussian blurred uniform white noise $U(-2, +2)$, as shown in Fig. 3(d); other recommended and effective options include uniform and $1/f^{1/2}$ noise (Fig. 3(c)(e)). It may also be observed from Fig. 3 that AdaDeband outperforms FCDR and FFmpeg-deband when smoothing the banded staircase, yielding a more pleasant visual enhancement.

![C. Requantization with Dithering](image)

Fig. 3. The size of the low-pass filter window that is applied on different bands depends on both (a) the band widths and (b) contextual edges/textures.

### III. Experiments

We compared our proposed adaptive debanding filter (dubbed AdaDeband) against two recent debanding/decontouring methods proposed for compressed videos: FCDR [17] and the FFmpeg-deband filter [25], on ten selected videos from the YouTube UGC dataset [30]. All the test sequences were scaled to 720p for computational convenience, and we compressed the videos using VP9 constrained quality (CQ) mode with $-\text{crf} 39$ to generate noticeable banding artifacts. In our implementations, only the luma channel was filtered since much less banding was observed on the Cb/Cr channels, but the proposed AdaDeband could also be applied to each color or chroma component.

Visual comparisons of the debanding methods are shown in Fig. 4. We may see that AdaDeband effectively smoothed the banding, leaving edges/textures well preserved. FFmpeg-deband, in contrast, tended to over-smooth weak textures, and under-smooth relatively large banded regions. Another advantage of AdaDeband is its adaptiveness, which can remove bands of any scale/shape, whereas FFmpeg-deband and FCDR require specification of a set of filter parameters, which may affect performance on scenarios it has not been exposed to.

To further verify the adaptiveness of AdaDeband against different quantization parameters, we generated different levels of banding effects using the VP9 CQ mode, with $-\text{crf}$ ranging from 1 to 51 on the exemplary video in Fig. 1. We may see in Fig. 5(a) that our proposed debanding filter produced robust BBAND scores against different levels of quantization without any further tuning of filter parameters.

In addition to the visual results, we also quantitatively compared the debanded videos using several common video quality models. However, it has been shown that traditional video quality metrics like PSNR and even SSIM family [31]–[33] do not align very well with human perception of banding [19]. Moreover, if the original video already contains banding artifacts, which is often the case for high-resolution videos, it is less reliable to rely on full-reference quality models, since...

* Filenames: Gaming_1080P-71a5, NewsClip_1080P-2eb0, Sports_1080P-19d8, Vlog_720P-66f8, LyricVideo_1080P-3b60, LyricVideo_1080P-5a1f, MusicVideo_720P-44c1, MusicVideo_720P-3698, Sports_720P-05f6, Vlog_720P-32b2
Fig. 5. Visualizations (1st row) and corresponding pixel value distributions (2nd row) of compared debanding outputs on the exemplary patch in Fig. 2. The compared methods are (a) FCDR [17], (b) FFmpeg-deband [25], (c–f) AdaDeband with (c) uniform noise (UN), (d) Gaussian-blurred uniform noise (GUN), (e) $1/f^{1/2}$ noise, and (f) pink noise ($1/f$). The top row of figures has been contrast-stretched for better visualization.

Fig. 6. Qualitative and quantitative comparisons of debanded frames produced by different debanding algorithms. Frame crops (contrast-stretched for better visualization) from left to right: original, compressed, FCDR, FFmpeg-deband, and AdaDeband. We follow the style in [26].

Table I

| Method    | PSNR     | SSIM     | BBAND  |
|-----------|----------|----------|--------|
| FCDR      | 39.08    | 0.9709   | 0.5790 |
| FFmpeg    | 38.84    | 0.9677   | 0.2264 |
| AdaDeband | 38.97    | 0.9699   | 0.2206 |

IV. Conclusion

We proposed an adaptive post-debanding filter to remove banding artifacts resulting from coarse video compression. The efficacy of the algorithm can be attributed to the accurate detection of banding regions, content-aware band reconstruction, and a dithered re-quantization. Both visual and quantitative results demonstrate significant performance improvement over prior debanding methods. Actually, it can be regarded as a visual enhancement algorithm, whose effects could be accounted for when performing rate-distortion optimized video encoding. Since it is a post-processing model, it may be optimized for efficient, low-power implementations in real-time applications.
