

Advanced Video Debanding

Gary Baugh
Sigmedia Group
Trinity College Dublin, Ireland
baughg@tcd.ie

Anil Kokaram
Google Inc.
Mountain View, CA, USA
anilkokaram@google.com

François Pitié
Sigmedia Group
Trinity College Dublin, Ireland
fpitie@mee.tcd.ie

ABSTRACT

High efficiency video coding has made it possible to stream video over bandwidth constrained communication networks. Depending on bit rate requirements, a video encoder sacrifices some image details which can then introduce visual artefacts. Due to aggressive encoding a contouring staircase artefact called banding can be observed in image regions with very low texture. This paper presents a solution for removing banding artefacts using image filtering and dithering techniques. A new banding index (BI) metric is also presented for quantitatively measuring the amount of banding in an image. Using this BI metric, we assess how much banding YouTube video encoding introduces in a video test dataset. There is a debanding filter in `ffmpeg` called *gradfun*. We compare the results of our debanding technique with those of *gradfun* on the YouTube test dataset.

Categories and Subject Descriptors

I.4.3 [Image Processing and Computer Vision]: Enhancement

General Terms

debanding, online video, encoding artefacts

Keywords

debanding, ffmpeg, enhancement, filtering, segmentation, dithering

1. INTRODUCTION

Streaming video over a communication network requires that the data rate does not exceed the available bandwidth. On the internet, the bandwidth available to the typical user is usually not sufficient for delivering high bit rate video. Hence, the source material is encoded to produce a lower bit rate version for streaming. YouTube for example, uses H.264 and VP9 codecs for video content encoding. There may be offline constraints such as disk storage capacity which determine that all source content must be encoded at a certain bit rate to minimise disk usage. Video encoding in these scenarios is lossy, so there is a reduction in video quality with respect

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CVMP'14 13-14 November 2014, London, United Kingdom

Copyright 2014 ACM 978-1-4503-3185-2/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2668904.2668912>.

to the source. The bottom left image in Fig. 1 shows a frame from a sequence (*SP111*) that was uploaded to YouTube. The bottom center image is the YouTube encoded version of this source frame. Also, these pictures are shown in the corresponding row above with some contrast enhancement. There are obvious degradations in the YouTube frame. Some of the textured details on the gray wall is lost (see top row). This is a typical sacrifice in quality the encoder makes in order to meet the bit rate constraint. On image surfaces with relatively slow changing gradients such as this wall, the encoder will create a staircase artefact as shown in Fig. 1 top center. Here, the contours formed by pixels with the same intensity level is what is referred to as banding. Details of the staircase effect are shown in Fig. 2 for a 1D slice of the image.

The aim of this work is to remove this banding artefact that arises from video encoding.

Importance of Debanding. Although encoded video may contain banding artefacts, they may not be observed by the content viewer. The size and quality of the viewing display are major factors here. Also the distance to the display and the amount of ambient light in the viewing environment are significant factors as well. However, with the ever improving quality of displays (such as Retina displays by Apple) in portable devices, the chances of observing banding artefacts are increasing. The viewer is usually positioned close to the display which also increases the odds.

For online premium video on demand services such as Netflix, iTunes and Google Play, video quality and bandwidth preservation are important issues. The consumer of these services indirectly correlates video quality with quality of service. Obvious artefacts such as banding would point toward poor video quality. Therefore, giving the viewer the illusion of a better quality video through debanding can maintain brand reputation. The debanding process can take place after decoding on the client side, so there is no extra demand for bandwidth.

Related Work. Some prior debanding solutions exist. They are implemented either as post-processes (`ffmpeg` has a debanding plugin called *gradfun*) or as in-loop[7] processes integrated to the back of the decoder (Yoo et al. [9], Jin et al. [6] and Choy et al. [4]). The proposed treatments usually follow a three step approach. The first step is to detect regions of banding. Since banding is most visible on smooth gradients, they usually look for low-textured regions of the picture. The next step is to undo the quantization and restore the original aspects of the image. This is done by applying a smoothing process to the degraded picture and thus removing the banding discontinuities. Finally, in order to match the screen bit depth, quantization needs to be reapplied. However, this time, a dithering process is used, which adds a noise pattern to create the illusion of a smooth gradient (Bhagavathy et al. [3], Yoo et al. [9]).

Unfortunately the existing approaches are limited, either by the



Figure 1: Frame 100 from the Artbeats SP111 sequence. *Bottom row:* The original (left), YouTube (middle), and debanded (right) versions of this frame. *Top row:* Contrast enhancement of the corresponding frames in the bottom row.

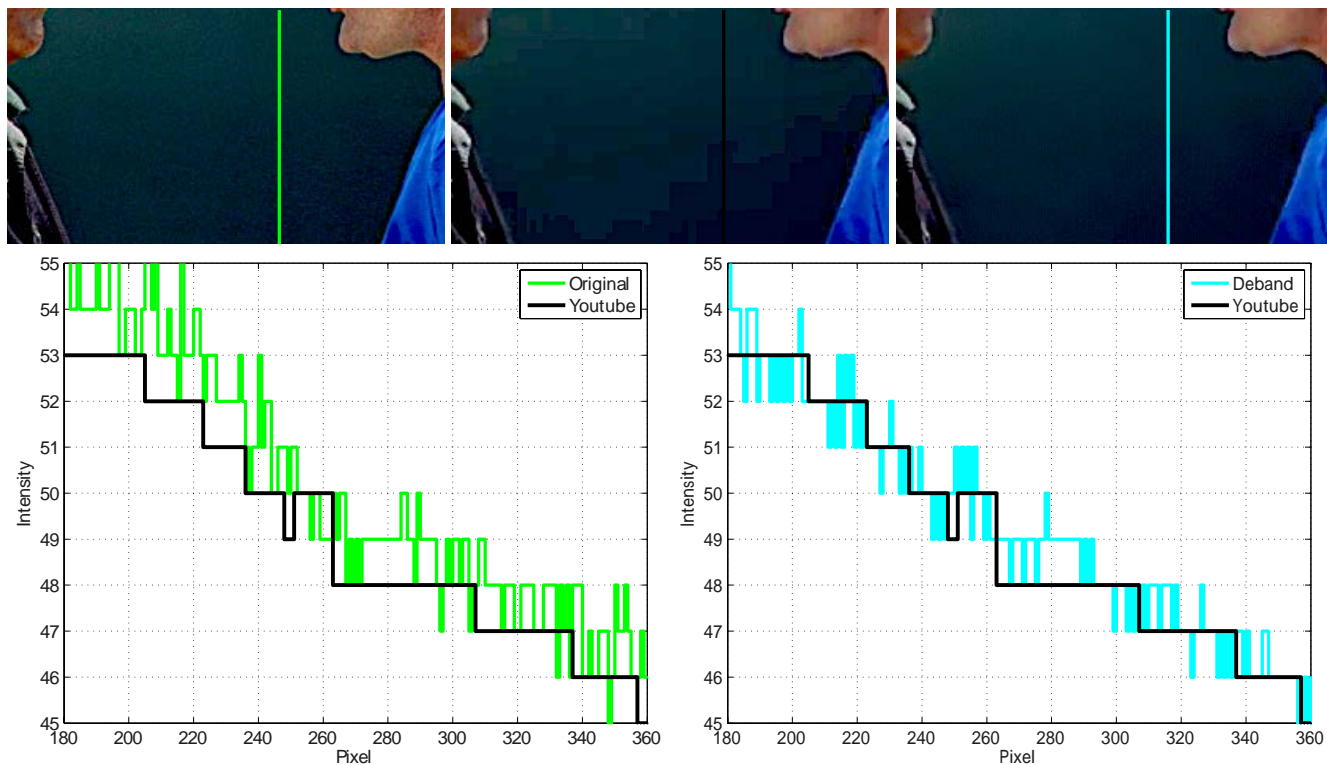


Figure 2: *Top row:* A zoom on the wall in sequence SP111. The pixel intensities along the green (left), black (center) and cyan (right) lines are plotted for the original, YouTube and debanded frames respectively. *Bottom row:* The left graph is the YouTube (black) and original (green) plots of pixel intensities. The right graph is the YouTube (black) and debanded plots of intensities.

level of quantization they can deal with, or by the picture quality they can achieve. For instance, the debanding plugin in the `ffmpeg` software is only suitable for very mild cases of banding as its performance degrades very quickly with the quantization levels. The restoration can blur the original video, which then loses its details and crispness. Other solutions such as Jin et al [6] are fast but incomplete as they leave a significant amount of blocking artefacts. Also this work makes assumptions specific to the decoder (H.264/AVC in this case). It is however common practice to transcode a video with several different codecs along a pipeline. Some of these transcoding steps may be unknown, especially for public user generated content. Therefore a generic debanding solution is required that does not need information about prior encoding steps.

Contributions. The first contribution of this paper is to propose a post-processing debanding method which is 1) able to scale with the quantization level, i.e. which is able to work with both heavily compressed and high quality videos, 2) conservative, i.e. produce very little artefact of its own, such as over smoothing the entire picture and 3) temporally consistent to avoid any flickering artefacts.

The second contribution of this paper is a technique to detect and report whether a sequence contains banding artefacts. For the debanding stage to exist in a video processing pipeline, there must be metrics to indicate when debanding is required. If no banding is detected the sequence passes through unaltered. We present a new banding index (BI) metric in section 2 for quantifying the amount of banding artefacts in an image.

Organisation of the paper. Our banding metric is presented in section 2 and the debanding algorithm in section 3. The performance of the debanding algorithm is quantitatively evaluated and compared against *gradfun* in section 4.

2. BANDING INDEX

Human identification of banding artefacts is a difficult task. Not only is this a subjective process, but there is a strong dependence on display quality and size, and environmental conditions. Prior to this work there has not been any published objective metrics for assessing banding in natural images. The metric presented here is called banding index (BI). The BI metric has a strong correlation to the visual perception of banding. Similar to SSIM[8] used for assessing image quality, the BI metric is between 0 and 1. For an image with no banding artefacts $BI = 1$.

Banding is usually observed in image regions with very smooth or uniform regions. The gray wall in Fig. 2 and the blue sky in Fig. 4 are typical examples. The third row of Fig. 4 shows a segmentation of the colours for the frames in the row above. Here a segment/block is a group of connected pixels with the same RGB colour. A random colour indicates each block. It may be observed that the YouTube segmentation (center) contains relatively large blocks in the sky, compared to the original (left) and debanded (right) segmentations. The block size is defined as the number of pixels in each segment/block. The distribution of these block sizes is shown in the bottom row of fig 4.

The distribution of the block sizes (Fig. 4) provides an indication of the presence of banding artefacts. Therefore the banding index (BI) is based on this observation. For a pixel \mathbf{x} we define $b(\mathbf{x})$ as the pixel banding index, which is given as,

$$b(\mathbf{x}) = \frac{1}{1 + e^{-\lambda/\Omega(\mathbf{x})}} \quad (1)$$

Where $\Omega(\mathbf{x})$ is the size of the block that contains pixel \mathbf{x} and λ is a constant. Fig. 3 shows a plot of the pixel banding index $b(\mathbf{x})$ versus the block size $\Omega(\mathbf{x})$, where the constant $\lambda = 61.1$. The constant

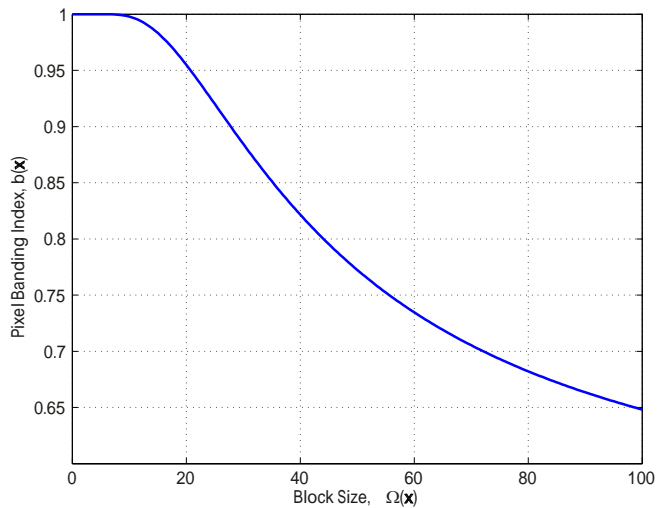


Figure 3: Plot of pixel banding index $b(\mathbf{x})$ versus block size $\Omega(\mathbf{x})$ for the constant $\lambda = 61.1$.

$\lambda = 61.1$ is chosen so the function $b(\mathbf{x})$ starts rolling off towards 0 when $\Omega(\mathbf{x}) = 10$ (10 pixels). From the distribution of block sizes for images with out banding, it is observed that the majority of the block are smaller than 10 pixels.

The banding index (BI) is defined using the pixel index $b(\mathbf{x})$ from Eq. 1 for an image with N pixels,

$$BI = \frac{1}{N} \sum_{\mathbf{x}} b(\mathbf{x}) \quad (2)$$

Based on experimental results, generally $BI < 0.9$ indicates the presence of banding. The first two rows in Fig. 4 show an example sequence (*SP136*) and the corresponding BI metrics. The left column shows the original frame with $BI = 0.999$, which means there is no banding. The YouTube version of this frame is in the center and has $BI = 0.834$, indicating the presence of banding. Here the banding can be seen in the sky. Finally, the debanded version is on the right with $BI = 0.996$. This suggests that a significant amount of the banding artefacts in the sky have been removed.

3. BANDING REMOVAL

Similarly to previous works, our debanding algorithm starts by finding the regions in the frame which suffers from banding artefacts, then applies some filtering correction to these regions to smooth out the bands, and finishes off with a dithering step.

3.1 Detection Mask

The first step is thus to isolate the problematic areas. A binary mask for identifying banding regions can be derived from the colour image segmentation obtained in the banding index evaluation. We mark as non-banding region the blocks that have less than 10 pixels in size. Isolated small segments that are not adjacent to any other small segment are discarded and marked as banding.

The top two rows of Fig. 5 shows examples of two frames with the corresponding banding detection in the bottom row. In the detection masks the blue regions indicate where banding artefacts are located. Just correcting the banding regions means that most of the original image is preserved. The entire image does not have to be processed in some cases requiring minimal computational effort.

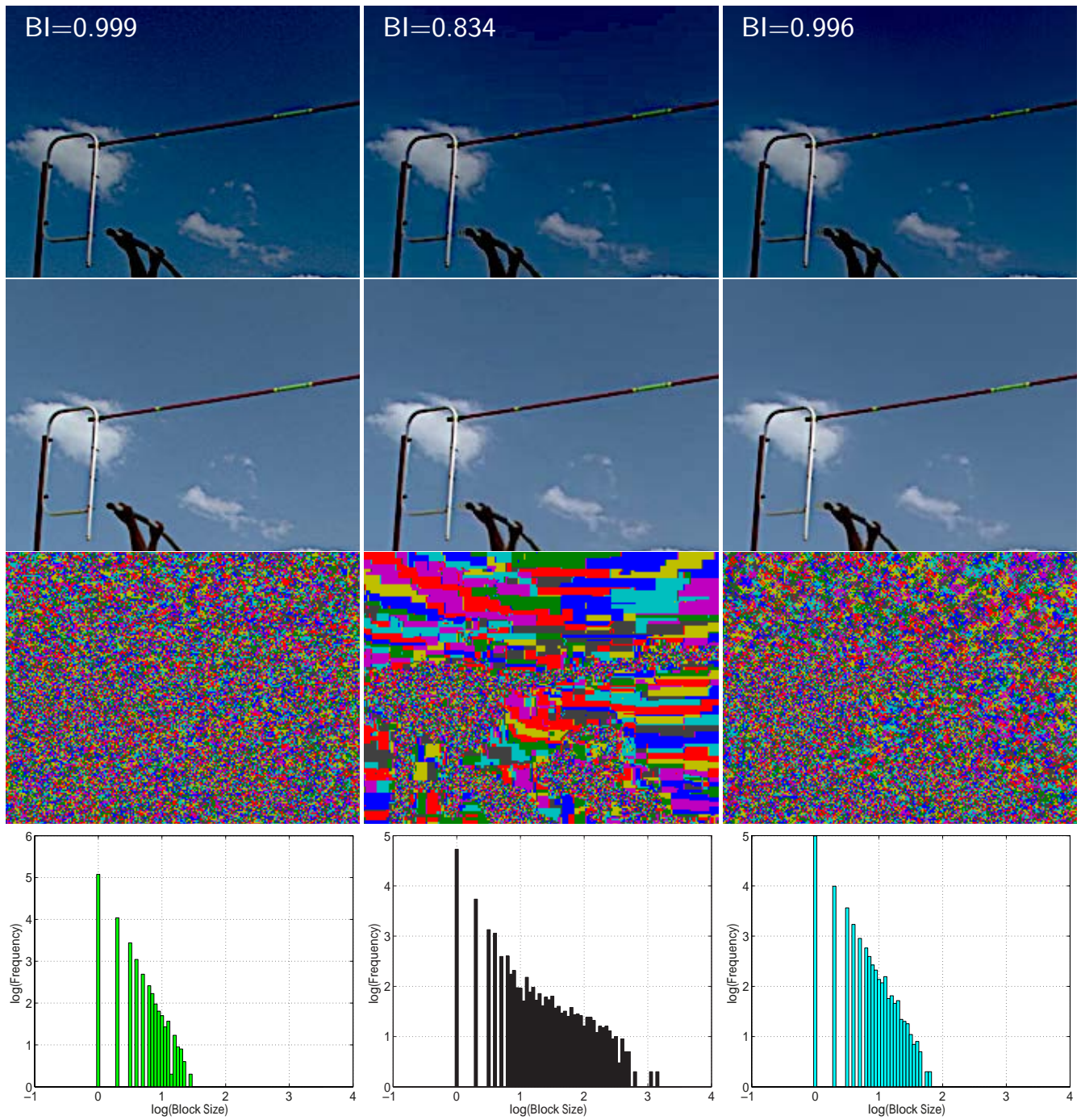


Figure 4: *Top row:* The contrast enhanced versions of the frames in the second row. *Second row:* The original (left), YouTube (center) and debanded (right) images for frame 188 from the SPI36 sequence. *Third row:* The segmentations of the colours for the images above. Each segment/block is indicated with a random colour. *Bottom row:* The distributions of the block sizes for the corresponding segmentations above. There is a relatively wider spread in the distribution for the YouTube block sizes in the center (black), compared to the original (green) and debanded (cyan) distributions.

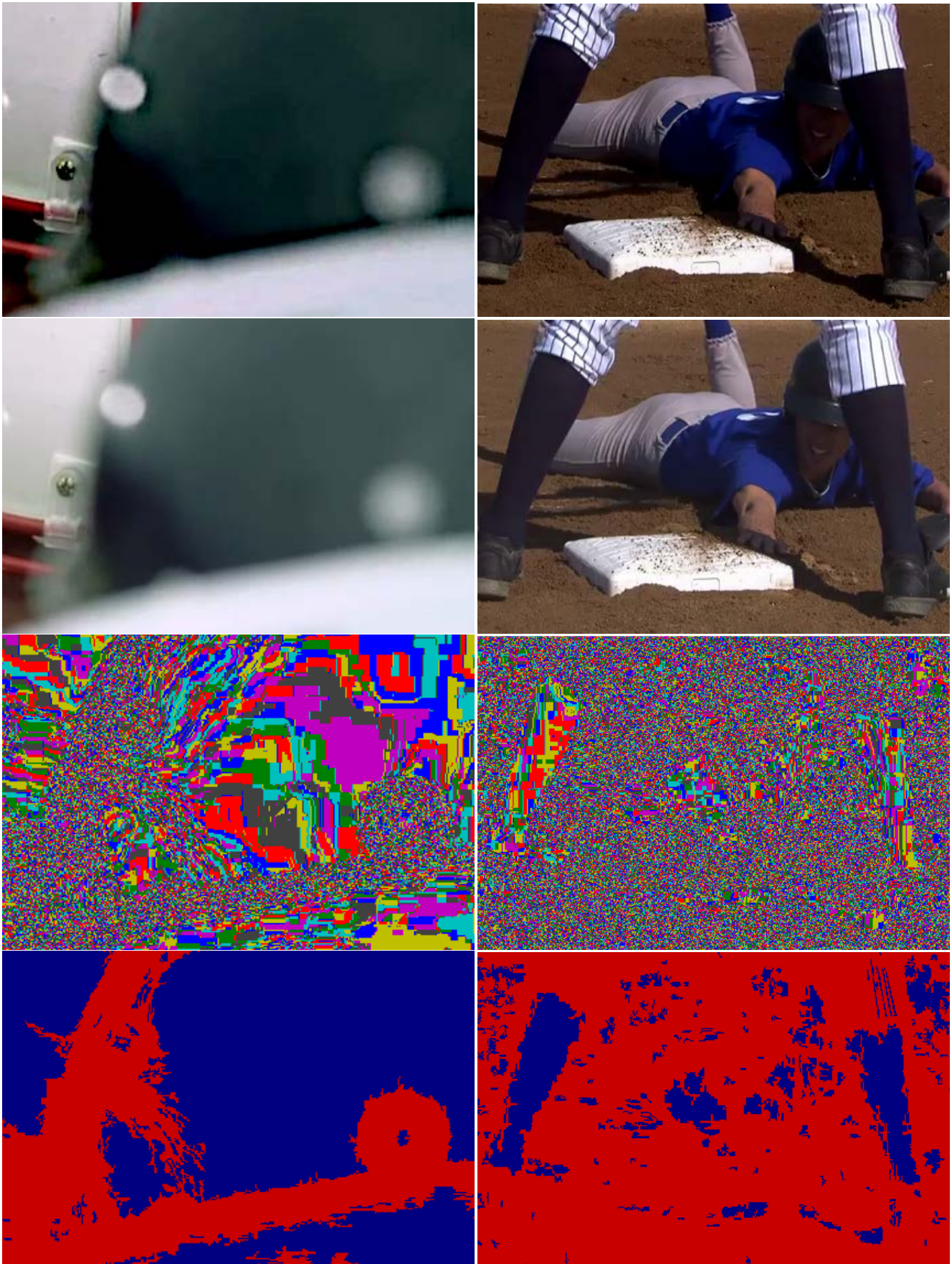


Figure 5: *Top row:* Contrast enhanced versions of the images in the second row. *Second row:* Example frames from sequences SP126 (left) and SP109 (right) with significant and minimal banding respectively. *Third row:* The colour segmentations for the corresponding frames above. *Bottom row:* The banding detection masks obtained from the segmentations in the row above.

The example frame on the right of Fig. 5 would be a case where minimal correction is required (see detection mask in bottom row).

3.2 Smoothing

With the regions of banding identified in the detection mask, the next step is to smooth out the banded regions to restore the original intensity gradients.

Consider a particular segment/block of the colour segmented image. All the pixels \mathbf{x} in the block have similar colour \mathbf{I} . The first observation is that for each of these pixels, the original, unbanded, colour $\hat{\mathbf{I}}(\mathbf{x})$ cannot differ from the observed pixel colour \mathbf{I} by more than some distance, i.e. $\|\hat{\mathbf{I}}(\mathbf{x}) - \mathbf{I}\| < T$. The threshold T is defined by the quantization process.

The second observation is that within a segment, the colour is probably bounded by the colours of the neighbouring segments. For instance, if we consider the case of a single neighbouring segment (A) with colour \mathbf{I}_a , then it is likely that the original unbanded colours lie in between both colours \mathbf{I} and \mathbf{I}_a :

$$\hat{\mathbf{I}}(\mathbf{x}) = w(\mathbf{x})\mathbf{I} + (1 - w(\mathbf{x}))\mathbf{I}_a \quad , \quad 0 \leq w(\mathbf{x}) \leq 1 \quad (3)$$

This is simply generalized for more than one neighbour as follows,

$$\hat{\mathbf{I}}(\mathbf{x}) = \frac{\sum_k w_k(\mathbf{x})\mathbf{I}_k}{\sum_k w_k(\mathbf{x})} \quad , \quad w_k(\mathbf{x}) > 0 \quad (4)$$

The problem is now to suitably define the weights w_k .

To achieve some smoothness across segments, we propose to design the weights as a function of the distance from the position \mathbf{x} to a particular neighbouring segment k . In other words, we want w_k to get bigger as \mathbf{x} gets closer to segment k . This distance is denoted as $d(\mathbf{x}, k)$ and is simply the distance of the nearest pixel in segment k to pixel \mathbf{x} (see Fig. 6).

We also assume that segments with large size Ω_k should have more influence than smaller segments. Combining all these ideas together, we propose the following definition for the weights,

$$w_k(\mathbf{x}) = \begin{cases} 2 \left[1 - \frac{1}{1 + d(\mathbf{x}, k)^{-\Omega_k/\Omega_0}} \right] & \text{if } \|\mathbf{I} - \mathbf{I}_k\| < T \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where Ω_0 denotes the size of segment containing \mathbf{x} . These weights are thus designed so that $w_k \rightarrow 0$ as $d(\mathbf{x}, k) \rightarrow \infty$. Also the term Ω_k/Ω_0 ensures that small blocks of pixels maintain most of their original colour when they are surrounded by very large blocks. This is a way of preserving the small details from the source image.

For computational considerations, we argue that we do not need to include all neighbouring segments in the computations. It has been observed in real images that each segment usually has a maximum of two neighbouring segments. We have thus limited ourselves to the nearest two segments for which $\|\mathbf{I} - \mathbf{I}_k\| < T$.

In our experiments, we found that $T = 20$ for colours in the 0:255 range was a good default.

3.3 Dithering

The final correction step involves adding local texture to the smoothed image regions using a dithering technique. From the smoothing step, the image $\hat{\mathbf{I}}$ obtained has floating point precision. However, the final debanded images are often only 8-bits per channel. Going from floating point to 8-bit precision can reintroduce banding artefacts. Therefore, dithering[1, 2, 5] is a commonly used technique for compensating for this lost in precision. The final colour for pixel \mathbf{x} is given as $\mathbf{I}_f(\mathbf{x})$ below,

$$\mathbf{I}_f(\mathbf{x}) = \hat{\mathbf{I}}(\mathbf{x}) + \eta(\mathbf{x}) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (6)$$

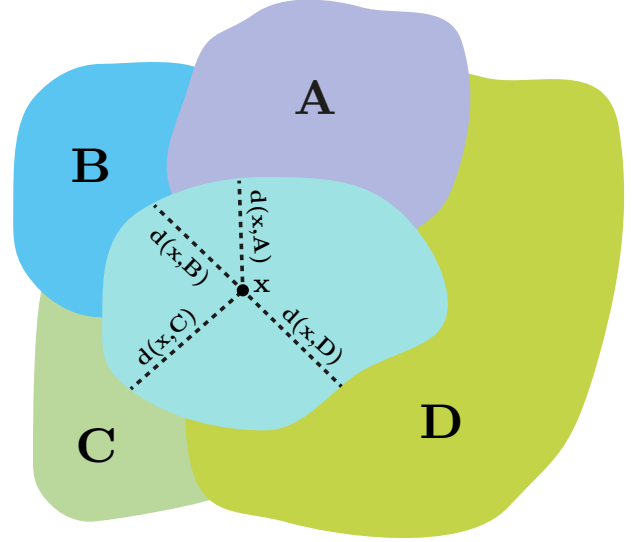


Figure 6: Distances $d(\mathbf{x}, \cdot)$ for pixel \mathbf{x} to neighbouring segments/blocks A, B, C and D.

$$\Upsilon = \frac{1}{16} \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

Where Υ is the dithering kernel, and $\eta(\mathbf{x}) = \Upsilon(x \bmod 4, y \bmod 4)$, for the pixel coordinates $\mathbf{x} = (x, y)$. Each colour channel in $\mathbf{I}_f(\mathbf{x})$ is rounded off to the nearest 8-bit value.

Only the pixels in the banding regions are smoothed and dithered.

4. EVALUATION

The aim of this work is to improve on the quality of the result obtained from the *gradfun* debanding filter in *ffmpeg*. To distinguish between our debanding result and that of *gradfun* we will refer to our technique as *deband*. The *gradfun* filter is an implementation of a box blur spatial filtering technique. *Gradfun* also does dithering of the final image to compensate for the 8-bits per channel precision of *ffmpeg*.

4.1 Visual Comparison

Some of the differences in performance between *gradfun* and *deband* are visually obvious. Fig. 7 shows the debanding results for both techniques on images from the *Memphis* and *House of Cards* sequences. The source images are in the left column, while the results for *gradfun* and *deband* are in the center and right columns respectively. The images in the first and third rows are contrast enhanced versions of the second and fourth rows respectively. These enhancements help to show the banding artefacts more clearly. For the *Memphis* result in the top two rows of Fig.7, *gradfun* (center) removes most of the banding artefact from the source on the left. However, there are still some subtle bands remaining. A very obvious visual artefact in the results produced by *gradfun* is a halo around edges. Observe the edges of the horse and the roof in center image, second row of Fig. 7. Our *deband* (right column) technique does not produce the edge halo artefact. *Deband* removes all the banding artefacts for the *Memphis* sequence while preserving the edge details.



Figure 7: Left column: Frames from the *Memphis* (1280×720) and *House of Cards* (1920×1080) sequences. Center column: The result of *gradfun* technique. Right column: The result of our *deband* technique. Contrast enhanced images are in the odd rows.

The *House of Cards* sequence in the bottom row of Fig. 7 contains more challenging banding artefacts. The banding contours are relatively larger here (left column) compared to the *Memphis* sequence. *Gradfun* fails to remove the bands in the *House of Cards* sequence. See the center image in third row for the contrast enhanced *gradfun* result. It may be observed that the edge details are very blurred compared to our *deband* result on the right. Our *deband* result in the right column shows that all the banding artefacts are completely removed and we again preserve all the edge details. Note that the stars in the blue sky are quite sharp.

4.2 YouTube Experiment

To perform an objective quantitative analysis between the results of *deband* and *gradfun*, we uploaded 36 high quality video sequences to YouTube. We obtained the *YouTube* encoded versions of these videos, which are then debanded using *gradfun* and our *deband* technique. The high quality version of each sequence is referred to as the *original* in subsequent discussions.

The 36 test sequences (PAL resolution, 720×576) are from the *Artbeats* video collection, which is available online at, artbeats.com/collections/362-Sports-1

There is a total of 18827 frames from all 36 sequences, which are called *SP101*, *SP102*, ..., *SP136*. Fig. 8 shows frames from the *SP101*, *SP101*, *SP112*, *SP122* and *SP110* sequences. The first and second columns contain the *original* and *YouTube* images respectively. It may be observed that there is visible banding in the *YouTube* images for the top three sequences. Again contrast enhancements are provided in rows one, three, five and seven for viewing the banding artefacts.

We use our banding index (BI) metric presented in section 2 to measure how much banding is introduced into a sequence by the *YouTube* encoding process. Also using the BI metric, we measure how much banding is removed by *gradfun* and *deband*. All measurements are made with respect to the *YouTube* versions of the test sequences. The *BI* for a sequence is the mean *BI* over all the frames

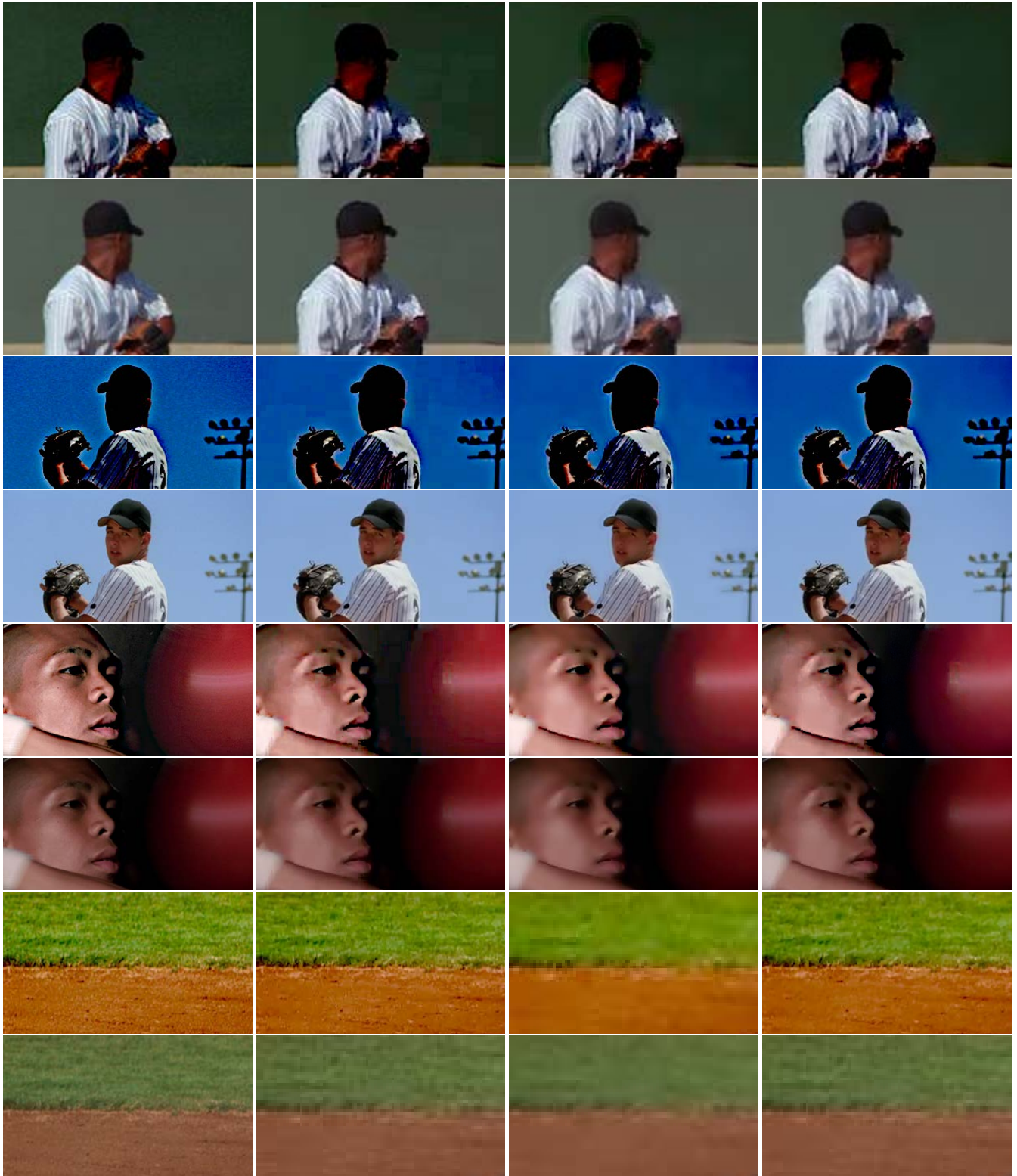


Figure 8: Four sequences from the *Artbeats* video collection. From the top, the sequences are *SP101*, *SP112*, *SP122* and *SP110* respectively. The first, second, third and fourth columns shows images for the *original*, *YouTube*, *gradfun* and *deband* sequences respectively. The contrast enhanced images are shown in the odd rows.

Table 1: Comparison of *original*, *deband* and *gradfun* sequences with respect to the corresponding *YouTube* sequences.

SP*	Frames	Original		Deband		Gradfun		SP*	Frames	Original		Deband		Gradfun	
		$SSIM_o$	ΔBI_o	$SSIM_d$	ΔBI_d	$SSIM_g$	ΔBI_g			$SSIM_o$	ΔBI_o	$SSIM_d$	ΔBI_d	$SSIM_g$	ΔBI_g
101	615	0.94	6.6	0.98	6.4	0.96	3.7	119	379	0.95	0.8	0.98	0.9	0.97	0.5
102	300	0.95	5.5	0.98	5.4	0.96	3.5	120	365	0.96	11.0	0.98	15.0	0.97	9.0
103	324	0.93	1.0	0.97	1.0	0.95	-0.6	121	496	0.97	13.0	0.99	18.0	0.99	9.0
104	149	0.94	0.9	0.97	0.8	0.94	-0.4	122	560	0.96	4.9	0.98	11.0	0.98	5.0
105	719	0.92	1.9	0.98	1.9	0.95	0.7	123	425	0.92	2.3	0.98	2.3	0.95	1.2
106	900	0.92	1.8	0.98	1.8	0.95	0.5	124	665	0.97	1.0	0.99	1.0	0.97	0.9
107	459	0.96	1.9	0.99	1.9	0.97	1.6	125	850	0.97	4.1	0.98	4.7	0.97	3.9
108	288	0.95	5.4	0.98	5.9	0.97	4.0	126	600	0.98	8.3	0.99	8.4	0.98	5.4
109	200	0.90	1.3	0.97	1.2	0.94	0.3	127	345	0.95	3.0	0.98	3.2	0.97	2.3
110	351	0.91	0.1	0.97	0.1	0.93	-0.2	128	651	0.96	0.2	0.99	0.5	0.95	0.5
111	1011	0.96	12.0	0.98	11.0	0.97	5.5	129	696	0.95	0.1	0.99	0.4	0.95	0.4
112	611	0.96	30.0	0.99	29.0	0.98	15.0	130	224	0.94	3.6	0.98	3.6	0.95	2.4
113	282	0.94	8.7	0.98	8.7	0.97	5.4	131	240	0.92	2.6	0.98	3.1	0.95	2.3
114	398	0.97	29.0	0.99	29.0	0.98	12.0	132	1211	0.92	0.7	0.99	0.7	0.96	0.6
115	267	0.94	1.4	0.98	2.2	0.96	1.5	133	574	0.92	6.6	0.98	7.2	0.97	4.2
116	641	0.95	3.4	0.98	3.5	0.97	2.6	134	475	0.90	1.0	0.99	1.1	0.96	0.8
117	406	0.95	3.0	0.98	3.2	0.97	2.4	135	1000	0.98	14.0	0.99	14.0	0.97	11.0
118	450	0.94	0.9	0.98	0.9	0.96	0.5	136	700	0.97	17.0	0.99	17.0	0.98	12.0
								All	18827	0.95	5.8	0.98	6.3	0.96	3.6

in that sequence. The percentage change in the banding index ΔBI is defined as,

$$\Delta BI_v = \left(\frac{BI_v - BI_y}{BI_y} \right) \times 100\% \quad (7)$$

Where BI_y is the banding index of the a *YouTube* sequence, and $v \in \{o, d, g\}$ for the corresponding *original*, *deband* and *gradfun* versions of this sequence.

We also measure the change in picture quality using the popular SSIM metric[8]. The quantities $SSIM_o$, $SSIM_d$ and $SSIM_g$ are defined as the similarity of the *original*, *deband* and *gradfun* sequences with respect to the *YouTube* version.

4.2.1 Results

Table 1 summaries the results for the 36 test sequences. Sequence *SP112* had the most banding artefacts due to YouTube encoding. Frames from this sequence are shown in rows three and four (from top) of Fig. 8. The enhanced *YouTube* image in the third row, second column shows that there is significant banding in the blue sky behind the baseball pitcher. The majority of the frame is occupied by the background sky, which is a prime candidate for having banding artefacts. Hence a banding index change $\Delta BI_o = 30\%$ is a justified result. Our *deband* technique removed the majority of the banding for this sequence. This is supported by the corresponding visual result in the fourth column of Fig. 8, as well as a $\Delta BI_d = 29\%$ value. The corresponding banding change for *gradfun* is $\Delta BI_g = 25\%$. The visual result for *gradfun* (third column) has the edge halo artefact discussed earlier for the *Memphis* sequence. This artefact also exist in the results for the *SP101* sequence in the top two rows.

Sequence *SP110* with a banding change $\Delta BI_o = 0.1\%$ in table 1 has the least banding artefact due to YouTube encoding. The bottom two rows of Fig. 8 shows images from this sequence. The contrast enhanced images are in the second row from the bottom. The *original* frame in the first column shows that this sequence is highly textured. The majority of the frame is occupied by grass and soil. Hence, there is enough local variations in the intensity here to

prevent banding artefacts. The *gradfun* version of this sequence in the third column is very blurred. This is not the case for our *deband* result in the fourth column. The blurring of *SP110* by *gradfun* introduces some slight banding artefacts, which is supported by the banding change $\Delta BI_g = -0.2\%$ value. We again see blurred results by *gradfun* for the *SP122* sequence in rows five and six in Fig. 8.

According to the last row of table 1, our *deband* technique made an overall improvement in the banding index of 6.3% compared to 3.6% by *gradfun*. Also *deband* made less changes to the *YouTube* images as suggested by a mean $SSIM_d = 0.98$ compared to *gradfun* with a value $SSIM_g = 0.96$. This means that we preserve more of the source image. The amount of banding introduced by YouTube encoding was overall $\Delta BI_o = 5.8\%$. This was roughly the amount removed by *deband* $\Delta BI_d = 6.3\%$. Some of the *original* sequences contained some banding artefacts which accounts for the difference in the amount introduced versus removed. Eventhough, the *original* videos were encoded at a high bit rate, the encoding process unavoidably introduced some banding artefact.

5. CONCLUSION

We presented a technique for removing banding artefacts from video sequences. This quality of the result produced by our *deband* technique is a significant improvement over the current debanding solution (*gradfun*) in *ffmpeg*. Unlike *gradfun*, our *deband* technique does not create any visible visual artefacts in the final restoration. We also introduced a new banding index (BI) metric for assessing the amount of banding in an image. Using this BI metric, we were able to quantify the amount of banding the Youtube encoding process introduced into a test video dataset. The performance of our proposed *deband* technique outperformed *gradfun* in terms of the amount of banding it was able to remove from the YouTube video dataset.

The current implementation of our *deband* technique on average processes 1.5 frames per second for HD (1920×1080) sequences. This is written as an unoptimised *ffmpeg* filter running on a single thread with all image processing operations being performed with

floating point precision. For future work we can improve the performance of our filter by doing a multithreaded implementation and reducing the number of floating point operations.

6. REFERENCES

- [1] L. Akarun, Y. Yardunci, and A. Cetin. Adaptive methods for dithering color images. *Image Processing, IEEE Transactions on*, 6(7):950–955, Jul 1997.
- [2] B. Bayer. An optimum method for two-level rendition of continuous-tone pictures. *Image Processing, IEEE Transactions on*, 6(7):11–15, Jun 1973.
- [3] S. Bhagavathy, J. Llach, and J. Zhai. Multiscale probabilistic dithering for suppressing contour artifacts in digital images. *Image Processing, IEEE Transactions on*, 18(9):1936–1945, Sept 2009.
- [4] S. Choy, Y. Chan, and W. Siu. Reduction of block-transform image-coding artifacts by using local statistics of transform coefficients. 4(1):5–7, January 1997.
- [5] R. S. Gentile, E. Walowit, and J. P. Allebach. Quantization and multilevel halftoning of color images for near-original image quality. *J. Opt. Soc. Am. A*, 7(6):1019–1026, Jun 1990.
- [6] X. Jin, S. Goto, and K. N. Ngan. Composite model-based dc dithering for suppressing contour artifacts in decompressed video. *Image Processing, IEEE Transactions on*, 20(8):2110–2121, Aug 2011.
- [7] Y. Lee and H. Park. Loop-filtering and post-filtering for low bit-rates moving picture coding. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 1, pages 94–98 vol.1, 1999.
- [8] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004.
- [9] K. Yoo, H. Song, and K. Sohn. In-loop selective processing for contour artefact reduction in video coding. *Electronics Letters*, 45(20):1020–1022, September 2009.