POST-PROCESSING FOR FLICKER REDUCTION IN H.264/AVC

Yair Kuszpet[1], Dimitry Kletsel[1], Yair Moshe[1], Avi Levy[2]

[1]Signal and Image Processing Lab., Department of Electrical Engineering, Technion - IIT
[2]Intel Corporation

ABSTRACT

The H.264/AVC standard mitigates some of the most noticeable artifacts of former video coding standards, such as blocking and ringing. However, it exhibits a new temporal artifact, not prevalent in these standards: a noticeable discontinuity between an intra frame and its preceding inter frame, known as flicker.

This paper proposes a post-processing scheme for dealing with the flicker artifact. A motion compensated version of the intra frame is generated based on its preceding inter frame, and is used to filter the intra frame in order to reduce the discontinuity. In addition, a non-reference flicker measure is proposed, and is used as a basis for an adaptive flicker-reduction technique.

Objective results, as well as subjective impressions, indicate that the proposed method significantly reduces the discontinuity, with an almost negligible drop in PSNR.

Index Terms— H.264/AVC video coding, flicker, post-processing, temporal noise, temporal coding artifacts.

1. INTRODUCTION

The H.264/AVC coding standard [1] offers a significant improvement over former standards in terms of coding efficiency. In addition, it eliminates or reduces some typical coding artifacts prevalent in those standards, such as blocking and ringing. However, H.264/AVC introduces a new coding artifact - a noticeable discontinuity between an intra frame and its preceding frame, known as flicker. At low-to-medium bitrates, H.264-coded videos are known to exhibit flicker, both in all-intra video sequences [2] and in video sequences with periodically-inserted intra frames [3], [4], [5].

Flicker in all-intra coded video sequences has been reported by Fan et al. [2], who identified its cause as differences in intra prediction modes between successive frames. To reduce the flicker, Fan et al. suggested an encoder modification that is not compliant with the H.264/AVC standard, as well as an objective reference flicker measure for all-intra coded video sequences.

In [3] and [4], video sequences with periodically-inserted intra frames were considered, and found to exhibit flicker. This was attributed to coarse quantization, to differences in intra prediction modes in intra frames, and to abundant use of skip mode in inter frames. In [3] it is suggested to alter the intra prediction mode selection by adding a term to the rate-distortion cost function, measuring the differences between the encoded macroblock and the macroblock at the same position in the previous intra frame. Further, [3] proposes to repeatedly encode a macroblock using finer and finer quantization until the resulting flicker is lower than a threshold value. The method proposed in [4] suggests to modify the intra prediction mode selection, by adding to its cost function the flicker measure, proposed in [2], of the intra frame relative to the preceding inter frame. Both [3] and [4] are compliant with the H.264/AVC standard. However, they attribute the flicker artifact mainly to the change in intra prediction modes between different intra frames. This assumption has been found unsatisfactory due to two observations. First, flicker is still present even when encoding videos using a constant DC intra prediction mode. Second, due to inter frame prediction, a block in an intra frame may originate from more than one block in the preceding frame, and may thus be affected by more than a single choice of intra prediction mode.

In [5], the cause of flicker is described as differences in coding noise patterns between inter-coding and intra-coding. The solution proposed in [5] is also at the encoder side, and requires the encoder to calculate a motion compensated image for intra frames, in addition to a spatially predicted image. The quantized values of the motion compensation residuals are used as ‘detent positions’. When the encoder quantizes the spatial prediction residuals of the intra frame, values that fall within the ‘detent range’ are quantized to the ‘detent position’, even if they are closer to another representation level. This causes the decoded intra frame to

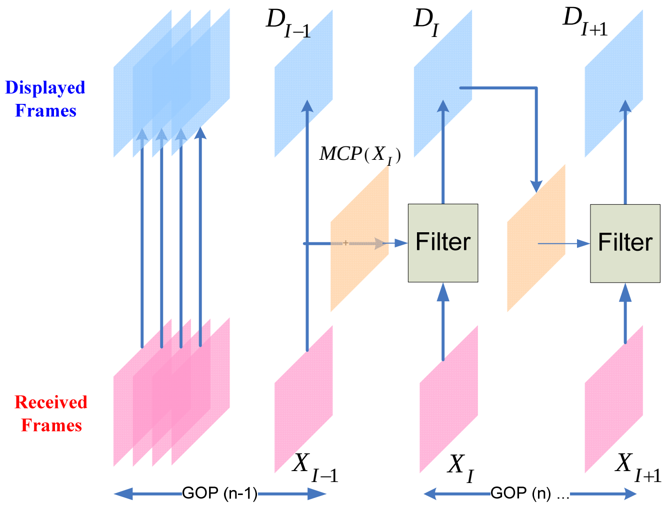


Figure 1: Filter scheme. Received frame and motion-compensated frame are averaged to obtain the displayed frame.

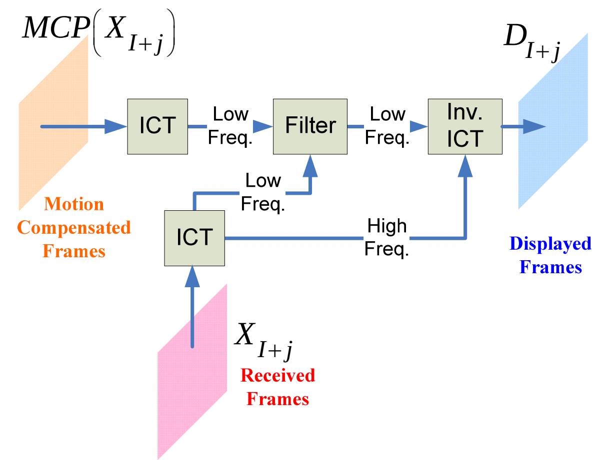


Figure 2: The filter averages only the low-frequencies of the frame. The high-frequencies of the displayed frame are those of the received frame.

resemble its preceding inter frame, thus reducing the flicker.

The approaches suggested in [3], [4] and [5] focus on the encoder side, and while they do reduce flicker to some degree, they do not eliminate it completely. In this paper a post-processing flicker reduction technique is proposed. This technique can be used to complement one of the encoder modifications mentioned above, or implemented independently. Section 2 provides a brief discussion of the flicker artifact, and describes a basic post-processing technique for its reduction. The approach is known in the general field of video interpolation, but obtains encouraging results for H.264/AVC as well. Section 3 describes a non-reference flicker measure and an adaptive flicker reduction based on that measure. Section 4 provides results for the adaptive flicker reduction. Section 5 concludes the paper.

2. FLICKER REDUCTION BY POST-PROCESSING

Following [2], let us denote by O_t an image at time t in the original video, and by P_t the predicted image used by the decoder, either motion compensated or intra predicted. The residual, $O_t - P_t$, is transformed (denoted by T) and quantized (denoted by Q) before being entropy coded. The decoder reverses this process. The reconstructed image R may be described as $R(O_t, P_t) = T^{-1}Q^{-1}QT(O_t - P_t) + P_t$. From this description it is apparent that $O_t \cong O_{t+1}$ does not guarantee $R_t \cong R_{t+1}$ if $P_t \neq P_{t+1}$. This can result in a discontinuity, which is considered flicker.

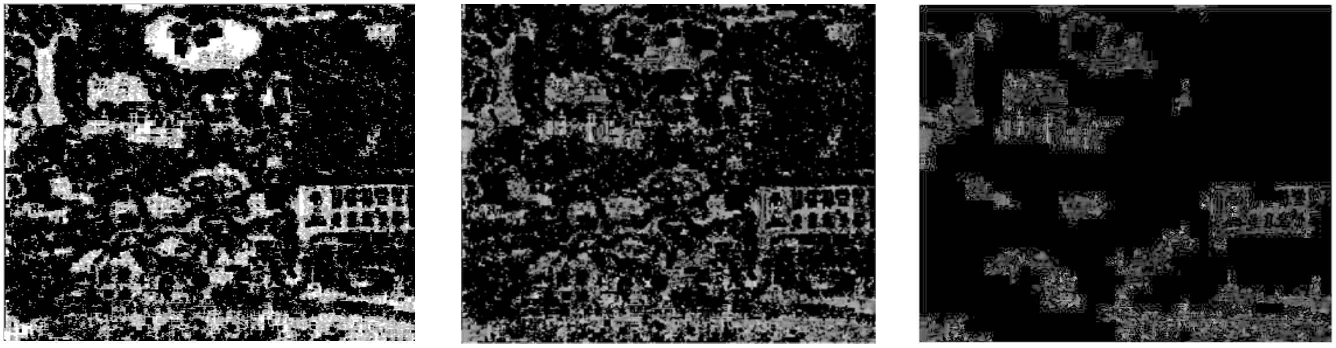
Following [5] we denote the motion-compensated predicted image for a frame X by $MCP(X)$. For intra frames, motion vectors are extracted from the decoded video sequence using block-matching with an MSE criterion. In addition, we denote the received frames of a GOP by $X_I, X_{I+1}, \ldots$ (where X_I is the intra frame), and by $D_I, D_{I+1}, \ldots$ the corresponding filtered frames, to be displayed on the screen. Flicker may be reduced by averaging only the intra frame X_I with $MCP(X_I)$, to obtain D_I. However, when the flicker is significant, it was observed that this caused a noticeable discontinuity between D_I and X_{I+1}. Thus it is suggested to filter k frames of each GOP (i.e. $X_I, \ldots, X_{I+k-1}$ where $k < GOP\ size$). The displayed frame D_{I+m} is used to generate the motion compensated image $MCP(X_{I+m+1})$, which in turn is averaged with X_{I+m+1}, etc., as illustrated in Fig. 1. $MCP(X_I)$ is generated using the last frame of the preceding GOP. Each received frame X_{I+m} $(0 \le m < k)$ is averaged with $MCP(X_{I+m})$ using a weighted average:

$$D_{I+m} = \begin{cases} (1-\alpha)MCP(X_{I+m}) + \alpha X_{I+m}, & m \in \{0, \ldots, k-1\} \\ X_{I+m}, & else \end{cases} \quad (1)$$

where $\alpha = (m+1)/(k+1)$.

Since the discontinuity is between X_{I-1} and X_I, this is where the filter should be strongest. Thus the filter in (1) gives more weight to the received frame the farther the frame is from X_I.

The described algorithm significantly reduces the flicker, but introduces a new temporal noise. Some small details might be lost in the filtered frames and then gradually reappear. This happens because small details are likely to be lost in the motion compensated frames. Thus, in order to preserve the finer details in each frame, only low frequencies are filtered. Each 4x4 block in the motion compensated and the received frames is transformed using



(a) (b) (c)

Figure 3: Flicker map generation for intra frame 180 of the 352x288 'mobile' sequence, coded at 350Kbps with GOP size of 15. (a) Smoothness map, after removing values higher than f_{min}=1; (b) Flat-difference map before opening by reconstruction, contrast enhanced; (c) Final flicker map, indicating areas where flicker is exhibited and its strength, contrast enhanced.

the Integer DCT transform. The 4 lowest coefficients are filtered in the described manner, while the other coefficients remain untouched, as illustrated in Fig. 2.

3. FLICKER REDUCTION BY ADAPTIVE POST-PROCESSING

The number of filtered frames in each GOP should depend on the severity of the flicker in that GOP's intra frame. In this section a non-reference measure to determine the required strength of the filter in an intra frame is proposed. This measure is based on the absolute difference between the received intra frame X_I and $MCP(X_I)$, denoted *diff*.

3.1. Smoothness score

It was observed that flicker is more common in smooth areas of a frame. Thus the first step is identifying such smooth areas in the intra frame. For each pixel, the standard deviation of pixel values in a 3x3 block centered on that pixel is calculated and denoted *std*. This results in a smoothness map for the frame. Values higher than f_{min} in the smoothness map correspond to non-smooth areas and are therefore discarded. An example for a smoothness map is given in Fig. 3a.

3.2. Flat-difference score

A flat-difference map is generated by multiplying the smoothness score of each pixel with its value in *diff*. In addition, since it was observed that flicker is usually more visually distracting for clusters of pixels, the next step is locating clusters of non-zero differences in the flat-difference map. This is achieved using morphological opening by reconstruction with a disk of radius 4, which results in the final flat-difference map, denoted *fd*. Fig. 3b is an example of a flat-difference map before opening by reconstruction.

3.3. Flicker score

The flicker score for a pixel *x* is given by:

$$flicker(x) = \begin{cases} 0, & fd(x) = 0 \\ diff(x), & fd(x) \neq 0 \end{cases} \quad (2)$$

In other words, the flicker score of a pixel is 0 if its flat-difference value is 0, or *diff(x)* if its flat-difference value is non-zero. This results in a flicker map of the intra frame, illustrated in Fig. 3c.

3.4. Determination of flicker measure and use in an adaptive filter

The flicker map shows the areas of the intra frame that exhibit flicker, and the strength of flicker in those areas. If more than a fraction $t_{no\_flicker}$ of the map is zero (i.e. exhibits no flicker), no filtering is performed for the processed GOP. Otherwise, let us define $s = \{(m,n)|diff(m,n) > 0\}$ and $s_i = \{(m,n)|0 < diff(m,n) \leq i\}$, where *m* and *n* go over the entire frame. In other words, *s* is the set of non-zero pixels in the flicker map, and s_i is the set of non-zero pixels in the flicker map whose value is lower than or equal to *i*. Then the flicker measure, *f*, is defined as the smallest integer i such that $|s_i| \geq 0.75 \cdot |s|$ (i.e., f is the smallest integer such that at least 75% of the non-zero pixels in the flicker map are lower than or equal to *f*).

The flicker measure *f* then indicates the number of frames to filter in the GOP, *k*, as described in section 2. If *f* exceeds the GOP size, a value of $k = GOP\ size$ is used.

4. EXPERIMENTAL RESULTS

The adaptive filter was used to process video sequences at CIF (352x288) resolution with a GOP size of 15 frames at various bitrates, coded with JM10.2 [6]. The results refer to the luminance channel only. The following parameters were used: $f_{min} = 1$, $t_{no\_flicker} = 0.98$.

The flicker measurement proposed in [2] is a referenced measure for all-intra coded videos, measuring flicker over the entire video sequence. However, in periodically-inserted intra frames, the perceivable flicker is concentrated near the intra frame. Thus this measure would measure flicker for frames that are not filtered and exhibit very little flicker. A modified version of Fan's measure is therefore used. This measure considers the intra frame and the 5 frames following it. For this measure, $\varepsilon = 1000$ (see [2]) was used.

Table 1, Table 2, and Table 3 list PSNR and flicker measurements for the first 250 frames of the 'container', 'mobile', and 'paris' video sequences, respectively. It can be seen that the flicker measure has been reduced by almost 30% in some cases, while the PSNR drop is no greater than roughly 0.5dB. Thus the adaptive filter successfully mitigates the flicker with a very small degradation of image quality. Furthermore, the filter correctly identifies low-flicker videos (e.g., 'container' at 500Kbps) and uses a weaker filter. The technique proved particularly useful at medium bitrates. In all cases, flicker was significantly reduced objectively as well as by subjective impressions.

Resulting videos are available on the website [7].

5. CONCLUSION

In this paper, a novel post-processing technique for flicker reduction in H.264-coded videos was proposed, based on averaging received frames with their motion-compensated counterparts. In addition, a non-reference flicker measure was proposed and used in an adaptive flicker reduction technique. Simulations demonstrate that the proposed technique can reduce the flicker measure by up to 30% while lowering the PSNR by no more than 0.5dB.

Bitrate [Kbps]	Measure	Unfiltered	Filtered	Difference
200	PSNR [dB]	33.90	33.88	0.01
	Flicker	4.42	3.60	18.51%
250	PSNR [dB]	34.83	34.80	0.03
	Flicker	4.08	3.37	17.39%
350	PSNR [dB]	37.36	37.25	0.11
	Flicker	4.09	3.34	18.52%
500	PSNR [dB]	38.84	38.71	0.13
	Flicker	3.80	3.17	16.54%

Table 1: Flicker reduction results for 'container' sequence

Bitrate [Kbps]	Measure	Unfiltered	Filtered	Difference
200	PSNR [dB]	22.66	22.48	0.18
	Flicker	12.55	9.91	21.05%
250	PSNR [dB]	23.84	23.70	0.14
	Flicker	10.27	8.36	18.59%
350	PSNR [dB]	26.87	26.60	0.27
	Flicker	8.34	6.41	23.14%
500	PSNR [dB]	28.51	28.16	0.35
	Flicker	6.69	5.55	17.00%
1000	PSNR [dB]	31.84	31.31	0.53
	Flicker	5.05	4.58	9.28%

Table 2: Flicker reduction results for 'mobile' sequence

Bitrate [Kbps]	Measure	Unfiltered	Filtered	Difference
200	PSNR [dB]	28.01	27.97	0.04
	Flicker	10.82	7.92	26.79%
250	PSNR [dB]	29.32	29.27	0.05
	Flicker	8.26	6.15	25.55%
350	PSNR [dB]	33.00	32.85	0.15
	Flicker	6.28	4.58	26.96%
500	PSNR [dB]	35.09	34.90	0.19
	Flicker	4.92	3.79	22.96%
1000	PSNR [dB]	39.57	39.24	0.33
	Flicker	3.87	3.16	18.14%

Table 3: Flicker reduction results for 'paris' sequence

6. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11 (MPEG), "Coding of audio-visual objects – Part 10: Advanced Video Coding," International Standard 14496-10, ISO/IEC, 2004.

[2] X. Fan, W. Gao, Y. Lu, D. Zhao, "Flicking Reduction in All Intra Frame Coding," JVT-E070, Joint Video Team of ISO/IEC MPEG & ITU-T VCEG Meeting, 2002.

[3] S. Sakaida, K. Iguchi, S. Gohshi, Y. Fujita, "Adaptive Quantization Control for Reducing Flicker of AVC/H.264 Intra Frames," *Proc. of Picture Coding Symposium*, 2004.

[4] S. S. Chun, J-R Kim, S. Sull, "Intra Prediction Mode Selection for Flicker Reduction in H.264/AVC," *IEEE Trans. on Consumer Electronics*, Vol. 52 (4), pp. 1303-1310, 2006.

[5] K. Chono, Y. Senda, Y. Miyamoto, "Detented Quantization to Suppress Flicker Artifacts in Periodically Inserted Intra-Coded Pictures in H.264 Video Coding," *Proc. of IEEE International Conf. on Image Processing (ICIP 2006)*, pp. 1713-1716, 2006.

[6] JVT Reference Software – JM 10.2: http://iphome.hhi.de/suehring/tml/.

[7] H.264 Post Processing for Flicker Reduction website: http://sipl.technion.ac.il/Info/Teaching_Projects_Flicker_e.shtml.