IEEE Transactions on Multimedia (2014). 16(7), 1863-1873.
DOI: http://dx.doi.org/10.1109/TMM.2014.2347257

Standard-Compliant Low-Pass Temporal Filter to Reduce the Perceived Flicker Artifact

Amaya Jiménez-Moreno*, *Student Member, IEEE*, Eduardo Martínez-Enríquez, *Student Member, IEEE,* Vinay Kumar, and Fernando Díaz-de-María, *Member, IEEE*

***Abstract*—Flicker is a common video compression-related temporal artifact. It occurs when co-located regions of consecutive frames are not encoded in a consistent manner, especially when Intra frames are periodically inserted at low and medium bit rates. In this paper we propose a flicker reduction method which aims to make the luminance changes between pixels in the same area of consecutive frames less noticeable. To this end, a temporal low-pass filtering is proposed that smooths these luminance changes on a block-by-block basis. The proposed method has some advantages compared to another state-of-the-art methods. It has been designed to be compliant with conventional video coding standards, i.e., to generate a bitstream that is decodable by any standard decoder implementation. The filter strength is estimated *on-the-fly* to limit the PSNR loss and thus the appearance of a noticeable blurring effect. The proposed method has been implemented on the H.264/AVC reference software and thoroughly assessed in comparison to a couple of state-of-the-art methods. The flicker reduction achieved by the proposed method (calculated using an objective measurement) is notably higher than that of compared methods:** 18.78% **vs.** 5.32% **and** 31.96% **vs.** 8.34%**, in exchange of some slight losses in terms of coding efficiency. In terms of subjective quality, the proposed method is perceived more than two times better than the compared methods.**

***Index Terms*—Flicker artifact, flicker reduction, H.264/AVC, low-pass temporal filtering, motion-guided temporal filtering, on-the-fly filter strength control, standard compliant.**

I. Introduction

THE transmission and storage of videos without compression is very costly in terms of bandwidth and space requirements, respectively. Though compression solves this problem, at the same time, it gives rise to compression-related artifacts. One of them is a temporal artifact called temporal fluctuation or stationary area fluctuation or simply flicker [1]. Since it is inherent to compression, the flicker artifact is generally perceived when using video coding standards, from Motion JPEG 2000 [2]–[5] to MPEG-2 [6] and H.264/AVC [7]–[19].

Flicker happens as a result of the fact that the video encoder does not consistently treat the co-located blocks of consecutive frames, thus increasing the inter-frame difference with respect to that of the original video. As stated in [10], this is a consequence of using different coding processes for these co-located blocks. Specifically, it becomes a more serious problem when coding a periodic Intra frame (I-frame), since the redundancy-removing methods used by the encoder notably change in the subsequent or previous Inter-coded frame (P- or B-frame). Furthermore, this artifact can be more clearly perceived when static areas are encoded at low or medium bit rates. In these cases, Inter frames tend to copy the pixel values from the previous frames [20], creating time consistency that is abruptly broken when an I-frame comes since the prediction source changes (from temporal to spatial neighbors), and flicker becomes much more apparent.

Fig. 1 depicts two consecutive reconstructed frames of the video sequence *Container* encoded with H.264/AVC. Within the circle we can perceive the difference between the I-frame (frame #25) and the previous P-frame (frame #24). Being a temporal artifact, it is difficult to perceive it when consecutive frames are looked at individually, but it is clearly visible and annoying in a normal playback of the video sequence.

Flicker reduction has been a relevant topic of research for many years [7]–[19]. However, a definite solution has not been found yet. Some of the proposed solutions are not standard compliant (e.g., [11], [19]), what seriously limits their field of application. Others (e.g., [12], [13], [17]) require to perform a filtering process at the decoder side, what prevents their use when any of the available standard decoder implementations must be used. Others (e.g., [9], [10], [15]) depend on fixed thresholds that hinder a proper generalization ability. Moreover, none of them controls the PSNR losses incurred by the algorithm in exchange for flicker reduction.

In this paper we propose a method that aims to overcome the above mentioned drawbacks. In particular, we suggest a fully standard-compliant method that uses a temporal low-pass filter implemented in the encoder. Furthermore, the parameters of the filter are estimated *on-the-fly*, so that the method is able to adapt to the video content. Finally, our approach allows for controlling the PSNR drop to prevent the perceived visual quality from impoverishing due to the blurring effect caused by the temporal filtering.

The proposed method was initially described in [21]. In this paper we present a more in-depth discussion and a much more comprehensive assessment of the method, that includes a pertinent subjective evaluation that, from our point of view, becomes necessary when dealing with this kind of artifacts.

The rest of the paper is organized as follows. Section 2 summarizes the state-of-the-art methods. Section 3 provides

EDICS: Signal Processing for Multimedia Applications ⟶ Compression and Coding (CPRS).

This work has been partially supported by the National Grant TEC2011-26807 of the Spanish Ministry of Science and Innovation. Amaya Jiménez-Moreno, Eduardo Martínez-Enríquez, and Fernando Díaz-de-María are with the Department of Signal Theory and Communications, Carlos III University, Leganés (Madrid), Spain (e-mail: ajimenez@tsc.uc3m.es, emenriquez@tsc.uc3m.es, fdiaz@tsc.uc3m.es). Vinay Kumar is currently with the Department of Electronics and Communication Engineering, Thapar University, Patiala, India.

(a) P-frame #24.

(b) I-frame #25.

Fig. 1. Example of the flicker artifact: (a) reconstructed P-frame; and (b) subsequent reconstructed I-frame of the sequence *Container*.

a detailed explanation of the proposed method. Section 4 describes the experiments conducted and presents and discusses the results obtained. Finally, Section 5 summarizes our conclusions.

II. State-of-the-art

A brief description of a few relevant state-of-the-art proposals is provided in this section.

In [7] a method was proposed that aims to reduce the discontinuity of the coding noise patterns between Inter- and Intra-coded frames. In particular, the quantization of DCT coefficients of co-located blocks may vary from frame to frame causing flicker. To address this problem the authors proposed the so-called "Detented Quantization", which is a quantization scheme that produces an Intra-coded MB more similar to an Inter-coded version of the same MB. To this end, the motion estimation (ME) process is carried out also in I-frames, and the transformed coefficients obtained with the usual Intra coding process are modified to bring them closer to those obtained when the frame is Inter-coded.

In [8] a two-pass I-frame coding method was presented. The first pass involves obtaining an Inter-coded version of the frame that is used as no-flicker reference. In the second pass, the flicker-prone MBs are coded using this Inter-coded version of the frame as reference, while the remaining MBs are coded using the normal Intra coding procedure. Furthermore, [8] described an effective flicker metric that will be used in our proposal and therefore will be explained later.

In [9] and [10] an Intra prediction mode selection method was discussed to reduce the flicker artifact. The method relies on modifying the distortion value of the Rate Distortion Optimization (RDO) process by adding a new flicker-related term, which is used in the flicker-prone MBs. This term is based on the sum of squared differences between pixel intensity values of an MB and those of its co-located.

In [11] a modified Intra prediction scheme was proposed for flicker reduction in all-Intra coding mode. Its goal is to reduce the flicker artifact caused by the lack of consistency in the selection of the Intra prediction mode from frame to frame. The algorithm quantizes the predicted block in the frequency domain before calculating the residue, so that the predicted and quantized block has a less-significant effect on the reconstructed values.

In [12] a post-filtering approach was explained to reduce the typical artifacts in compressed image and video such as flicker, blocking, or ringing. Specifically, a novel spatio-temporal adaptive fuzzy filter was proposed. The filter acts according to the correlation between the current pixel and its spatio-temporal neighbors (after motion-compensation). The fuzzy filter is able to reduce the artifacts by increasing the pixel correlation, while preserving the edges of the image.

Another post-filtering based approach was presented in [13]. In particular, a temporal filtering algorithm, based on a Maximum a Posteriori (MAP) estimation and named as robust statistical temporal filter (RSTF), was proposed to reduce the flicker artifact. The authors claim that RSTF reduces flicker and preserves motion sharpness.

The work described in [17] was also a post-filtering approach to reduce flicker distortion. Specifically, a motion-compensated version of the I-frame is estimated from previous P- or B-frames, and it is used to filter the I-frame employing a weighted average. A flicker metric is proposed to measure the strength of the flicker artifact, so that the number of frames considered by the filtering process can be adapted according to it.

[14] discussed a rate control algorithm that acts on the Quantization Parameter (QP) value to reach a given target bit rate, while taking special care of quality consistency. Specifically, exponential models are proposed that consider both the target bit rate and the buffer occupancy, and keep the distortion as constant as possible, thus reducing the flicker artifact.

In [15] a temporal flicker reduction and denoising (TFRD) in video was explained using sparse directional transforms. The method assumes that the video signals are inherently spatially and temporally sparse and proposes to reduce both flicker and noise by enforcing this sparsity. A set of pixel values called a sub-frame is created for each pixel. This sub-frame extends to pixels from the current frame when spatial sparsity is considered and to pixels from both previous frames and the current frame in the case of temporal sparsity. The sub-frame is then transformed using a 2-Dimensional DCT and a hard threshold is applied such that the transform coefficients below the threshold are set to zero.

The work presented in [16] addressed the flickering artifact due to both the transform coefficient quantization and the use of different prediction modes in co-located MBs. This approach assumes that flicker is noticeable due to the presence of noise in the original sequence and, consequently, presents a Kalman filtering-based pre-processing for flicker reduction. The Kalman filter recursively estimates the noise power in the input signal and acts only on flicker-prone regions, which are determined according to a flickering score.

In [18] a method to reduce the flickering and blocking artifacts was proposed. Flicker is addressed by means of an adaptive multi-scale motion post-filtering. Specifically, for each block of the current frame, the most similar blocks in previous and next frames are found and used in the filtering process, which is selectively applied on smooth areas.

In [19] the flickering artifact was reduced by extending the GOP length. Since a longer GOP may lead to lower performance in terms of random access and error resilience, the authors proposed to periodically insert a new kind of frame called PI-frame, which is encoded and sent both as a P- and as an I-frame. If there is no transmission error or random access, the P version is decoded; otherwise, the I version is used. Obviously, this method is not standard compliant.

III. Proposed Method

In this paper we propose a standard compliant method for reducing the flicker artifact. Since the flicker artifact is a temporal phenomenon, mostly due to changes in the luminance values of co-located MBs of consecutive frames (generally, a P- or B-frame followed by an I-frame), we aim to reduce its effects by making the temporal evolution of the luminance smoother by means of a novel filtering approach. Specifically, a temporal low-pass filter is proposed that makes the reconstructed pixel values of an I-frame more similar to those of the previous Inter-frame. Furthermore, since low-pass filtering unavoidably introduces blurring effect, an algorithm is developed to control this blurring effect in the filtered regions by limiting the PSNR loss.

The method description is organized in three subsections: first, we describe the proposed filtering technique; second, we explain how to make it compliant with typical video coding standards; and third, we describe the blur control algorithm.

A. Selective temporal motion-guided filtering

1) Filter formulation: To reduce the flicker artifact, we propose a filtering process that aims to make those flicker-prone regions of consecutive frames more similar to each other. To this purpose, we use the following temporal low-pass filtering:

$$\hat{f}'^{j}_{n,i} = \alpha \hat{f}^{j}_{n,i} + (1-\alpha)\, \hat{f}^{j}_{n-1,MV}, \tag{1}$$

where $\hat{f}'^{j}_{n,i}$ is the filtered intensity value of the j-th pixel of the i-th MB of the n-th frame; $\hat{f}^{j}_{n,i}$ is the reconstructed intensity value of the same pixel; $\hat{f}^{j}_{n-1,MV}$ is the reconstructed intensity value of the corresponding pixel in the previous frame (the meaning of the subindex MV is explained below); and α is a parameter that balances the weight of the current and the previous frame pixel values. The sub-index MV has been introduced in $\hat{f}^{j}_{n-1,MV}$ to make clear that the co-located pixel is not always the one used by the filter. The reason is that using the co-located pixel ($\hat{f}^{j}_{n-1,i}$) would produce a significant blurring effect when the pixel region undergoes any kind of motion from one frame to another (either camera or object motion). To solve this problem, we propose a motion-guided filter that uses the pixel in the frame $n-1$ that is pointed by a calculated motion vector (MV). For this purpose, an ME process should also be carried out in the I-frames. In our experiments, we have employed 16×16-pixel blocks for this ME process, which is only used for guiding the proposed filtering. Obviously, in static regions the MV will likely point to the co-located block.

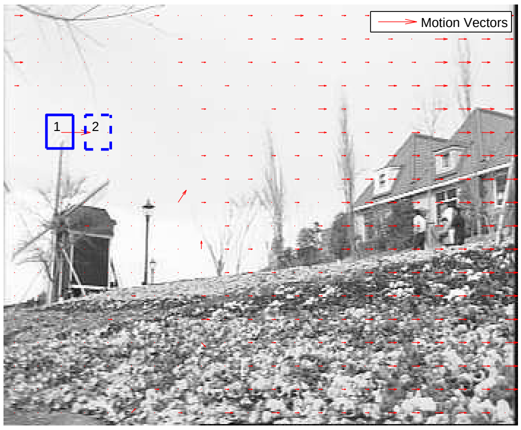


Fig. 2. An I-frame of the sequence *Flower*. Motion vectors are depicted by means of small red arrows. Block labeled as 1 refers to a block of the current frame and block 2 refers to the block in the previous frame that is pointed by the corresponding motion vector.

Fig. 2 shows an illustrative example of an I-frame of the sequence *Flower* where the MVs are overlaid. In order to obtain the filtered version of the pixels belonging to block labeled as 1 ($\hat{f}'^{j}_{n,i}$ in (1)), the proposed filter uses the reconstructed luminance values of the same block in the current frame (pixels denoted as $\hat{f}^{j}_{n,i}$ in (1)) and the reconstructed luminance values of block labeled as 2 in the previous frame (pixels denoted as $\hat{f}^{j}_{n-1,MV}$ in (1)).

Moreover, the proposed filter is selectively used on flicker-prone regions where it actually turns out to be effective according to a proper flicker metric. Next subsection provides a brief description of the metric used for this purpose.

2) Flicker-prone block detection: To detect flicker-prone blocks we need to define a metric that allows us to estimate the flicker-related distortion. An accurate flicker metric was proposed in [8]:

$$D^{j}_{flicker,n,i} = \max\left(0, \left|\hat{f}^{j}_{n,i} - \hat{f}^{j}_{n-1,i}\right| - \left|f^{j}_{n,i} - f^{j}_{n-1,i}\right|\right), \tag{2}$$

where $f^{j}_{n,i}$ is the original intensity value of j-th pixel of the i-th MB of the n-th frame; $\hat{f}^{j}_{n,i}$ is the reconstructed intensity value of the same pixel; and $f^{j}_{n-1,i}$ and $\hat{f}^{j}_{n-1,i}$ are, respectively, the co-located original and reconstructed pixel intensity values in the previous frame.

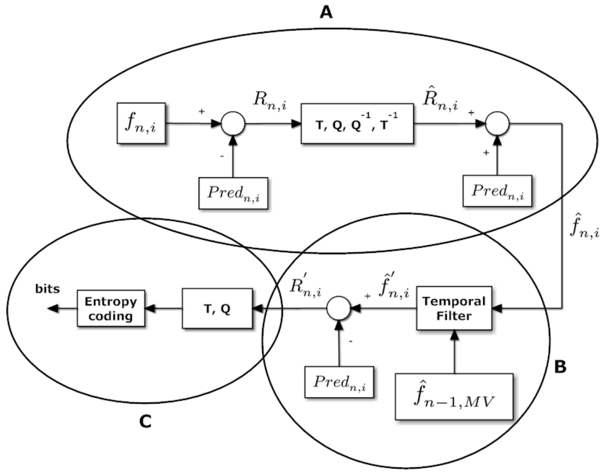


Fig. 3. Block diagram of the standard-compliant temporal filter implementation proposed in this paper for reducing the perceived flicker artifact.

In a few words, this metric assumes, according to the results of some perceptual tests [8], that the flicker distortion is noticeable when the video encoding process actually increases the original difference between co-located pixels of consecutive frames. Or the other way around: it considers that flicker is not perceived when the difference between two co-located pixels keeps equal or smaller after the encoding process.

The flicker distortion of a block is computed by accumulating pixel-wise distortions:

$$D_{flicker,n,i} = \sum_j D^j_{flicker,n,i}. \quad (3)$$

Our method relies on this metric to choose the blocks to be filtered, i.e., only those blocks for which the filtering process actually reduces this flicker metric are filtered.

It should be noticed that although the flicker metric is computed on co-located blocks, the proposed motion-guided filtering actually relies on those pixels in the previous frame that result from the motion compensation process, avoiding unpleasant visual effects and substantial PSNR losses when the filter is applied in areas with motion.

B. Standard compliant implementation

One of the main contributions of this work is to develop a flicker-reducing algorithm that can be used in any standard-compliant implementation of a modern video encoder. In other words, the bitstream generated by an encoder incorporating the proposed flicker-reducing technique has to be decodable by any standard decoder implementation.

With this objective in mind, we suggest to transform and quantize a modified version of the Intra-prediction residue calculated taking into account the filtered version of the pixel block obtained with (1), i.e.:

$$R'^j_{n,i} = \hat{f}'^j_{n,i} - Pred^j_{n,i}, \quad (4)$$

where $R'^j_{n,i}$ is the modified residue, $\hat{f}'^j_{n,i}$ is the filtered pixel intensity value, and $Pred^j_{n,i}$ is the corresponding Intra prediction (calculated following the standard encoding process). Fig. 3 represents the complete methodology to obtain a standard compliant implementation, where T, Q, T^{-1}, and Q^{-1} stands for the transformation, quantization, inverse transformation, and inverse quantization processes, respectively. First, the Part A of Fig. 3 shows a typical encoder-decoder loop at the encoder side. As it can be observed, a prediction residue is obtained, transformed and quantized, to generate a reconstructed version of the pixel block ($\hat{f}_{n,i}$), as in the habitual encoding process. Then, Part B of Fig. 3 illustrates how to obtain the filtered version of the pixel block ($\hat{f}'_{n,i}$) following (1), and the modified version of the Intra residue ($R'_{n,i}$) following (4). Finally, this modified version of the Intra-prediction residue is transformed, quantized and entropy coded as illustrated in Part C of Fig. 3.

Since the Intra-prediction $Pred_{n,i}$ is not modified (it is always calculated following the standard encoding process) and the modified residue $R'_{n,i}$ is transformed and quantized the same way the encoder would make with the residue obtained following the standard encoding process, it becomes evident that the decoder can reconstruct a quantized version of the filtered pixel values without any additional information.

C. Controlling the PSNR losses

The strength of the filtering process in (1) can be controlled through the α parameter. On the one hand, the lower the α, the more similar the filtered block becomes to the motion-compensated block, and, consequently, the less noticeable the flicker artifact is. On the other, the lower the α, the higher the PSNR losses, and the more noticeable the blurring effect. Fig. 4 shows an example of this fact. In particular, two versions of the same reconstructed I-frame of the sequence *Akiyo* at QP 40 are shown. The image on the left side was obtained for $\alpha = 0.8$, while $\alpha = 0.5$ was used in the image on the right side. As can be observed, the undesirable low-pass filter effects (missing details and blurring effect) are more noticeable for $\alpha = 0.5$, particularly in the presenter's face and suit. The PSNR loss is also significant: from $38.14\ dB$ ($\alpha = 0.8$) to $37.46\ dB$ ($\alpha = 0.5$). Additionally, the magnitudes of the corresponding filter frequency responses are plotted in Fig. 5, which clearly reveals the more marked low-pass character of the temporal filter obtained for $\alpha = 0.5$.

With this trade off between flicker distortion and PSNR loss in mind, the selection of the optimum value of the α parameter can be formulated as follows. For each block i, we aim to minimize the flicker distortion $D_{flicker,i}$ (where we have removed the dependence with the frame number n for convenience), while maintaining PSNR loss of the block i ($PSNR_{loss,i}$) below a certain target value provided by the user, $PSNR_{loss,tar}$, i.e.:

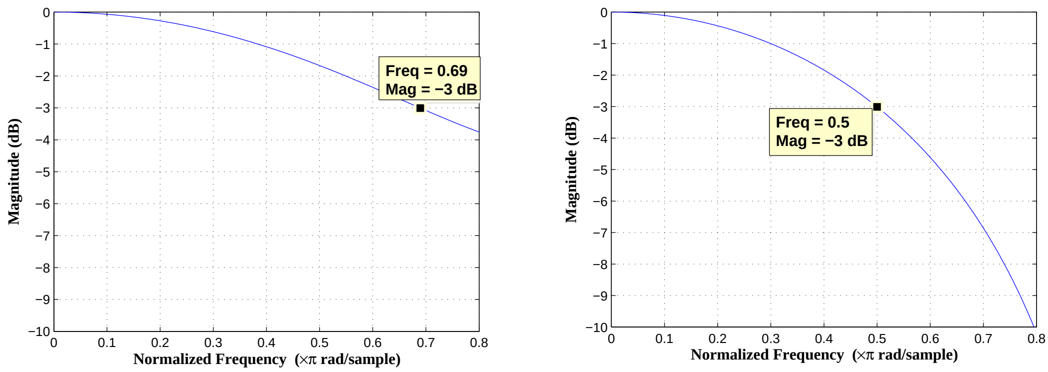
$$\min_{\alpha_i} \{D_{flicker,i}(\alpha_i)\} \text{ subject to}$$
$$PSNR_{loss,i}(\alpha_i) \leq PSNR_{loss,tar}. \quad (5)$$

To solve this problem we have first carefully examined the relation between α_i and $PSNR_{loss,i}$, and the relation between α_i and $D_{flicker,i}$. To this purpose, we have implemented the proposed temporal filter in the reference software of

(a) $\alpha = 0.8$, $PSNR = 38.14dB$.

(b) $\alpha = 0.5$, $PSNR = 37.46dB$.

Fig. 4. I-frame of the sequence $Akiyo$ encoded at QP 40. Illustration of the increasing undesirable effects of the low-pass filtering as α decreases.



(a) $\alpha = 0.8$.

(b) $\alpha = 0.5$.

Fig. 5. Example of the filter frequency response for the same two values of α of Fig. 4. The cutoff frequency ($-3dB$) is approximately $0.5\pi rad/sample$ for $\alpha = 0.5$, and $0.7\pi rad/sample$ for $\alpha = 0.8$.

the H.264/AVC video coding standard [22] and gathered a comprehensive set of data from several video sequences.

By way of example, some results are shown in Fig. 6. Part (a) of the figure shows the results obtained for three low resolution sequences (*Container* at QP 36, *Coastguard* at QP 32, and *Akiyo* at QP 40); and Part (b) shows the results for three high resolution sequences (*Ice Age* at QP 36, *Pedestrian* at QP 32, and *Mobisode* at QP 40). On the left hand side of each figure, the average PSNR loss with respect to the reference software ($PSNR_{loss}$) is plotted as a function of α, where the average PSNR loss computation is calculated using only the filtered blocks. On the right hand side of each figure, the accumulated flicker distortion over the filtered blocks, $D_{flicker} = \sum_i D_{flicker,i}$, is plotted as a function of α.

As can be observed, $D_{flicker}$ is a monotonic increasing function of α and $PSNR_{loss}$ is a monotonic decreasing function of α. Thus, the minimum value of α_i that is able to meet the PSNR loss constraint ($PSNR_{loss,i} \leq PSNR_{loss,tar}$) solves the problem stated in (5). Therefore, if we were able to model $PSNR_{loss,i}$ as a function of α_i for each block i, we could obtain the optimum value of α_i.

Regarding the modeling of $PSNR_{loss,i}$ as a function of α_i, two observations are in order. First, the data gathered to plot the curves of Fig. 6 exhibit a high degree of variability at several levels. In particular, significant differences have been found between sequences of different resolutions, also between different sequences of the same resolution, and even between different MBs of the same sequence. Therefore, we suggest to perform *on-the-fly* adaptation of the model. Second, a simple model is required so that the process of finding a proper value of α_i for each MB i does not result in a computational burden. Consequently, a linear model has been chosen as an acceptable solution for our purposes:

$$PSNR_{loss,i}(\alpha_i) = a + b\alpha_i, \quad (6)$$

where a and b are estimated *on-the-fly* as a function of the video content. To estimate these parameters, we employ two pairs of values $(\alpha_i, PSNR_{loss,i})$. More complex estimations have also been tested, such as using linear regression from four $(\alpha_i, PSNR_{loss,i})$ pairs. However, the performance improvement (around 3 % better in terms of flicker reduction) does not compensate for complexity increase (derived from two additional reconstruction processes per block). Therefore, using two pairs of values was considered the best trade-off to perform the estimation. Fig. 7 illustrates the estimation procedure with data gathered from an MB of the sequence *Pedestrian*. The blue dashed line shows the actual $PSNR_{loss,i}$

achieved for seven different values of α_i (from 0.2 to 0.8), and the red solid line shows the estimated linear model. In our experiments, the values of α_i used to estimate the linear model were 0.4 and 0.6.

Once the model parameters have been obtained, the model is used to estimate the value of α_i that likely leads to $PSNR_{loss,tar}$ for each MB i. We denote this value as $\alpha_{tar,i}$. Fig. 8 provides a visual example of the selected $\alpha_{tar,i}$ for an I-frame of the sequences *Akiyo* (Part (a)) and *Ice Age* (Part (b)). Specifically, the mean luminance of every block has been set according to the estimated $\alpha_{tar,i}$ (the higher $\alpha_{tar,i}$, the brighter the block). As can be seen, the areas with more detail (the presenter's face in *Akiyo*, or the mountains and the characters in *Ice Age*) are filtered with higher $\alpha_{tar,i}$ values, as expected. In this way, PSNR losses are controlled in these detailed areas, in exchange for lower flicker reductions.

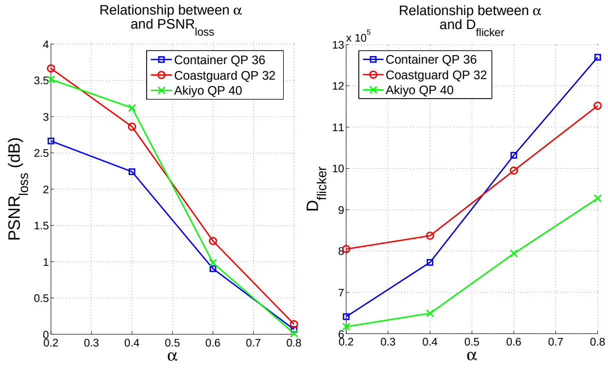
The complete method is summarized in Algorithm 1. It is worth noticing that, when the obtained $\alpha_{tar,i}$ produces a notably higher $PSNR_{loss,i}$ than the $PSNR_{loss,tar}$, the non-filtered version is selected by the encoder.

Algorithm 1 Proposed flicker-reducing encoding process.

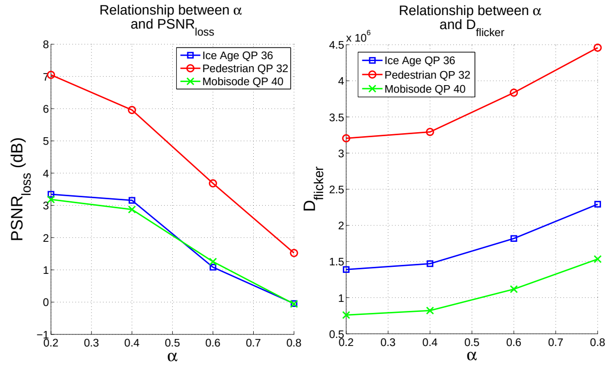
Require: $PSNR_{loss,tar}$: target PSNR loss
Require: $L = 2$: number of α_i values necessary to estimate the linear model $PSNR_{loss,i}(\alpha_i)$
Require: I: number of blocks

```
for i = 1 ... I do
    Calculate the non-filtered version of the reconstructed block and its flicker distortion D_{flicker,i,non-fil}
    for l = 1 ... L do
        Calculate the filtered version of the reconstructed block
        Compute and store the PSNR loss, PSNR_{loss,i,l}
    end for
    Estimate the parameters of the linear model
    Calculate α_{tar,i} to meet the PSNR target loss using in the previous model
    Compute the filtered version of the reconstructed block for α_{tar,i} and its flicker distortion D_{flicker,i,fil}
    if D_{flicker,i,fil} < D_{flicker,i,non-fil} then
        Select the filtered version of the block
    else
        Select the non-filtered version of the block
    end if
end for
```

The proposed method requires to calculate four different reconstructed versions of each block: the conventional non-filtered version, those using the two α values necessary to estimate the linear $PSNR_{loss,i}(\alpha_i)$ model, and the one using the estimated $\alpha_{tar,i}$. Additionally, the method also requires a simplified ME process that is only performed for the 16x16 partition size. Obviously, the corresponding increment of computational cost only concerns I-frames, while the encoding process of P- or B-frames remains the same as that of the reference software. Therefore, in usual applications where the Inter-frame full ME and mode decision (MD) processes become the most complex parts of the encoder implementation, the complexity increment of the proposed method concerning just the Intra-frames is clearly acceptable.



(a) Low resolution sequences.



(b) High resolution sequences.

Fig. 6. Experimental models of the relations between α and $PSNR_{loss}$, and α and $D_{flicker}$ for several sequences.

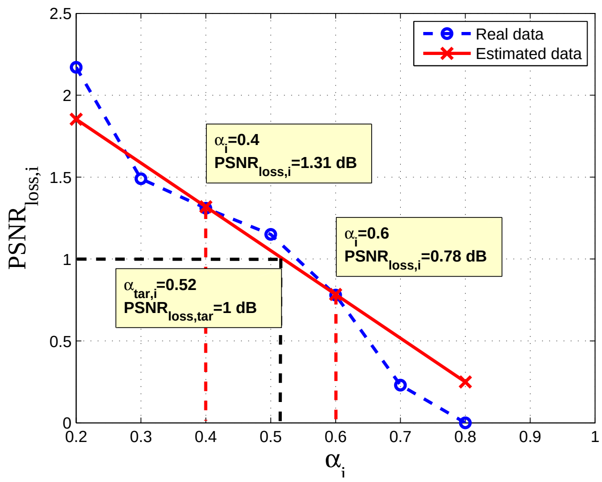
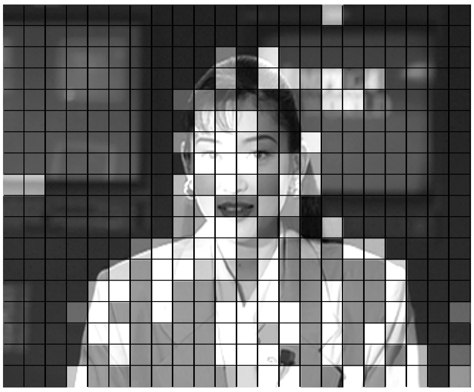


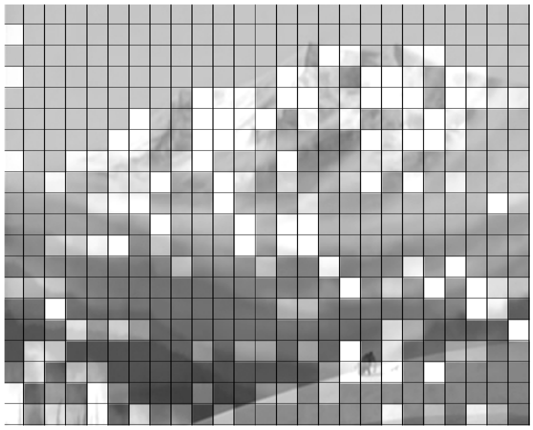
Fig. 7. An illustration of the estimation of the parameters that define the linear model. The data are gathered from an MB of the sequence *Pedestrian* at QP 32.

IV. Experiments and Results

To assess the performance of the proposed method, it was implemented on the H.264/AVC JM15.1 reference software



(a) Akiyo at QP 40 with $PSNR_{loss,tar} = 1dB$.



(b) Ice Age at QP 36 with $PSNR_{loss,tar} = 2dB$.

Fig. 8. Visual example of the selected $\alpha_{tar,i}$ for an Intra frame. The higher $\alpha_{tar,i}$, the brighter the block

[23]. In particular, we start proving the improvements obtained when the α_i parameter (and thus the temporal filter strength) is estimated *on-the-fly* against using a fixed α_i. Then, we compare the proposed approach to two state-of-the-art methods. Finally, we carry out a subjective quality evaluation, which, in our opinion, turns out to be necessary when assessing a perceptual artifact such as the flicker artifact.

To evaluate the performance of the compared methods we have computed two different measures. The first one is a flicker reduction (FR) measure relative to the flicker distortion produced by the H.264/AVC reference software, i.e.:

$$FR(\%) = \frac{D_{flicker}(JM15.1) - D_{flicker}(Method)}{D_{flicker}(JM15.1)} \times 100, \tag{7}$$

where $D_{flicker}$ is the flicker distortion defined in Section III-C, computed only over the filtered blocks.

The second one was reported in [19] and is based on the normalized cross-correlation (NCC). Specifically, we compute the NCC between consecutive error frames (obtained by subtracting the original frame from the reconstructed one) and calculate the NCC gain (ΔNCC) that the assessed method achieves with respect to the reference software, computed only over filtered blocks:

$$\Delta NCC(\%) = \frac{NCC(Method) - NCC(JM15.1)}{NCC(JM15.1)} \times 100. \tag{8}$$

As we are actually measuring correlation, the higher ΔNCC, the better.

The loss of quality due to the filtering process has also been evaluated in terms of $PSNR_{loss}$, as defined in Section III-C (i.e., measured with respect to the PSNR achieved by reference software and considering only the filtered blocks). Moreover, the losses in coding efficiency are measured as the bit rate (BR) increments relative to the reference software:

$$\Delta BR(\%) = \frac{BR(Method) - BR(JM15.1)}{BR(JM15.1)} \times 100. \tag{9}$$

TABLE I
TEST CONDITIONS

Encoder configuration	
Profile	Main
Frame rate	25
RD Optimization	Enabled / Disabled
GOP pattern	IPPP
QP values	32, 36, and 40
Intra period	25
Symbol Mode	CABAC
Number of Reference Frames	1
Frames to be encoded	100

Finally, we have also computed the percentage of filtered blocks ($\%Blocks$).

A. Objective quality evaluation

1) Adaptive vs. non-adaptive approach: To assess the proposed adaptive version of the algorithm described in Section III-C, we have compared it to a non-adaptive version of the same algorithm. In the non-adaptive version, a fixed value of the α parameter was used; while in the adaptive version, the α parameter was derived *on-the-fly* on a block-by-block basis to meet a maximum PSNR loss constraint. The test conditions are summarized in Table I. The experiments were conducted using a set of 8 sequences of CIF resolution, 5 of SD, and 2 of HD, all of them listed in Table II. For the non-adaptive approach, $\alpha = 0.7$ was experimentally selected to obtain a proper balance between FR and $PSNR_{loss}$; while for the adaptive approach, we used two target PSNR loss constraints $PSNR_{loss,tar} = 1dB$ and $2dB$. Table II shows the results obtained in terms of FR, ΔNCC, $PSNR_{loss}$, ΔBR, and $\%Blocks$.

The following conclusions can be drawn from these results.

First, the FR achieved by our both proposals are high in absolute terms. Specifically, 38.70% is obtained with the non-adaptive method, and 34.82% with the adaptive version

TABLE II
COMPARATIVE PERFORMANCE EVALUATION OF ADAPTIVE AND NON-ADAPTIVE VERSIONS OF THE PROPOSED METHOD.

		Non-adaptive α ($\alpha = 0.7$)					Adaptive α ($PSNR_{loss,tar} = 2dB$)					Adaptive α ($PSNR_{loss,tar} = 1dB$)				
Sequence		FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ loss	ΔBR	$\%Blocks$
CIF	**Container**	31.53	11.23	2.65	0.37	13.49	41.10	33.43	1.52	5.60	57.90	27.41	25.03	0.86	4.89	52.10
	Coastguard	26.70	31.01	2.59	−0.40	41.80	31.92	92.82	1.72	1.14	85.79	20.07	68.63	0.94	0.73	72.63
	Bridge-far	50.73	19.17	1.18	−0.44	3.66	52.68	26.14	1.11	2.24	33.47	40.88	20.49	0.76	2.76	28.42
	Bridge-close	24.77	11.42	1.76	−0.37	18.54	44.07	26.57	1.38	−0.02	66.91	31.16	20.91	0.84	0.68	60.18
	Flower	34.30	78.68	3.26	0.30	35.88	20.03	66.75	1.58	1.59	48.00	12.74	41.44	0.82	0.88	42.67
	Nature	31.29	4.09	1.07	1.04	9.59	28.66	24.73	1.71	3.20	39.84	23.03	20.50	1.02	1.94	28.45
	Akiyo	34.91	13.52	2.29	0.30	6.22	26.38	34.22	1.69	6.78	39.81	17.16	26.24	1.00	6.20	34.84
	Football	31.11	6.79	4.2	0.21	48.81	33.68	26.47	1.4	0.48	58.97	23.73	17.70	0.80	0.45	49.40
SD	**Corvette**	31.31	6.36	2.6	−0.20	13.12	33.17	17.77	1.6	0.88	64.78	23.46	13.38	1.00	1.06	50.78
	Ice Age	36.18	1.85	1.35	−0.24	2.67	33.65	10.03	1.44	1.50	33.38	25.86	9.28	0.92	1.50	26.68
	Last Samurai	42.38	2.34	1.79	−0.27	3.39	43.17	14.76	1.43	−0.29	27.56	32.29	8.94	0.93	−0.29	21.71
	Shields	33.12	6.82	3.02	0.32	19.49	27.04	32.90	1.78	7.62	73.97	18.20	21.18	0.96	7.62	65.09
	Mobisode	55.41	−1.89	2.25	0.92	32.16	43.37	12.18	1.41	2.65	26.87	35.96	12.17	0.85	2.65	45.32
HD	**Pedestrian**	59.61	0.97	7.53	0.61	22.59	30.64	27.63	1.55	1.30	36.00	23.53	22.19	0.94	1.30	28.84
	Rush Hour	57.18	6.08	8.84	0.22	14.57	32.72	34.95	1.56	1.53	30.90	25.79	41.32	0.95	1.53	24.31
Average		**38.70**	**13.22**	**3.08**	**0.15**	**19.06**	**34.82**	**32.09**	**1.52**	**2.41**	**48.27**	**25.41**	**24.62**	**0.90**	**2.26**	**42.09**

for $PSNR_{loss,tar} = 2dB$. The ΔNCC results suggest that the adaptive version works better than the non-adaptive one: $\Delta NCC = 32.09\%$ vs. 13.22%. Second, the adaptive method incurs in a $PSNR_{loss}$ lower than or equal to the $PSNR_{loss,tar}$, what proves that the adaptation mechanism is working properly. Likewise, both FR and ΔNCC behave as expected: the lower $PSNR_{loss,tar}$, the lower FR and ΔNCC, and vice-versa. Third, when comparing the adaptive and non-adaptive methods, we observe that the FR achieved by the non-adaptive approach is higher but comparable to that of the adaptive approach with $PSNR_{loss,tar} = 2dB$. However, the PSNR losses incurred in each case are significantly different; in particular, $PSNR_{loss} = 3.08dB$ for the non-adaptive case, while $PSNR_{loss} = 1.52dB$ for the adaptive one. Additionally, the standard deviation of $PSNR_{loss}$ is much lower in the adaptive case ($0.17dB$ vs. $2.24dB$). The reason for these substantial differences is that the adaptive version is able to vary the filter strength according to the video content. Specifically, in some sequences such as *Flower*, *Football*, or *Pedestrian*, in which some regions have a lot of movement, the fixed-α low-pass filtering produces significant PSNR losses while the adaptive version of the algorithm is able to adapt the filter strength block-by-block in order to meet the $PSNR_{loss,tar}$. Fourth, with respect to ΔBR, the increments in the non-adaptive version are negligible while in the adaptive versions are quite moderate, with average values below 2.5% in both cases and just a couple of sequences above 3%. Therefore, the proposed method turns out to be very effective in terms of flicker reduction without paying a significant penalty in terms of coding efficiency.

The results concerning HD sequences deserve a few words. As can be seen, the non-adaptive approach incurs in a very high $PSNR_{loss}$. These results could mean that the fixed value of α is not appropriate for all the different video resolutions; while the proposed adaptive scheme is able to manage this issue quite effectively, keeping $PSNR_{loss}$ under the specified target and achieving, at the same time, very significant results in terms of both FR and ΔNCC.

Another advantage of the adaptive approach has to do with $\%Blocks$. While in some sequences such as *Bridge-far* or *Ice Age* this value is quite low for the non-adaptive approach (making difficult to generalize from the obtained results), it is much higher for the adaptive approach, leading to more significant results. Let us consider one example where this type of behavior happens. In very static regions, the non-adaptive temporal filter produces a filtered region that remains too similar to the current one (because the α value is relatively high). Then, once the prediction residue is quantized, the result becomes similar or identical to the non-filtered version, and, consequently, the flicker distortion is not reduced and the filtered version is rejected. However, when the adaptive version is used, a lower value of $\alpha_{tar,i}$ is selected and the filtered block resembles much more the co-located region in the previous frame. Therefore, the flicker distortion is reduced and the filtered version is selected.

2) Comparison to state-of-the-art approaches: Our second set of experiments was devoted to compare our proposal to two state-of-the-art approaches. Specifically, we have selected two methods, those described in [10] and [7], to serve as references for comparison. We should mention that the method in [7] was designed to work with RDO disabled; thus, we have adapted our proposal to operate in such mode. The test conditions are the same as those of previous experiment (Table I).

Let us first discuss the comparative results regarding [10] (Table III). Since the method by Chun et al. incurred in a negligible PSNR loss, we have configured our proposal to meet a very low $PSNR_{loss,tar}$. In particular, we have used $PSNR_{loss,tar} = 0.2dB$ and $0.5dB$. As Chun's method relies on a modified MD process that takes into account the flicker distortion in the cost function of the RDO process, it does not provide enough degrees of freedom to significantly reduce the flicker distortion. This observation can be readily inferred from the results of Table III (columns 2, 3, 4, and 5), where it can be seen that both $PSNR_{loss}$ and ΔBR are negligible, but FR and ΔNCC are quite moderate. In terms of FR and ΔNCC, our proposal substantially outperforms the Chun's method in almost all the sequences. Regarding the ΔBR results, as it can be expected, our proposal produces a moderate ΔBR that is higher than that of Chun's method (which is negligible). Furthermore, the PSNR losses for $PSNR_{loss,tar} = 0.2dB$,

TABLE III
COMPARATIVE PERFORMANCE EVALUATION VS. [10] (RDO ENABLED).

		[10]					**Proposed Method ($PSNR_{loss,tar} = 0.2dB$)**					**Proposed Method ($PSNR_{loss,tar} = 0.5dB$)**				
	Sequence	FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ loss	ΔBR	$\%Blocks$
CIF	**Container**	7.45	2.55	0	−0.10	31.56	8.95	12.41	0.23	6.85	11.22	17.02	17.38	0.48	5.30	40.26
	Coastguard	0	0	0	0	0	9.57	33.42	0.30	1.16	20.39	13.50	46.06	0.52	0.81	53.47
	Bridge-far	0	0	0	0	1.68	31.23	10.59	−0.03	7.69	2.63	27.84	16.21	0.43	5.39	20.05
	Bridge-close	0.49	0.21	0.02	−0.01	15.15	11.67	10.36	0.21	3.07	7.88	21.85	15.34	0.49	2.01	46.09
	Flower	9.17	2.70	−0.09	0	26.51	5.06	68.26	0.27	0.67	17.84	8.34	53.68	0.49	0.73	36.36
	Nature	6.96	5.92	0.05	0.29	97.72	25.85	11.78	0.27	0.80	4.37	21.26	18.06	0.63	1.26	17.28
	Akiyo	7.40	4.12	0	0.29	83.08	9.84	18.36	0.20	7.58	6.45	11.62	21.63	0.52	6.79	26.76
	Football	0	0	0	0	16.66	8.37	6.76	0.34	0.76	13.55	14.98	10.86	0.55	0.62	36.33
SD	**Corvette**	7.04	2.01	0.03	0.25	57.72	16.03	7.43	0.32	2.18	8.41	17.10	10.36	0.59	1.49	32.84
	Ice Age	13.82	4.46	0	0.07	99.81	25.16	8.06	−0.02	3.42	2.68	19.40	10.22	0.55	2.33	18.64
	Last Samurai	7.87	2.43	0.01	0.28	92.13	25.89	7.84	0.06	3.69	2.59	23.24	11.85	0.54	2.52	14.21
	Shields	0	0	0	0	0	10.78	6.59	0.24	4.75	19.01	13.36	13.79	0.51	4.67	49.97
	Mobisode	2.26	0.40	0.01	0.23	25.12	34.77	4.17	0.08	2.30	7.35	31.18	9.34	0.51	2.56	16.75
HD	**Pedestrian**	7.88	1.02	−0.03	0.15	19.00	22.50	12.04	0.42	1.42	5.93	18.81	18.12	0.61	1.16	20.63
	Rush Hour	9.49	0.36	0	−0.02	4.21	25.85	24.15	0.17	1.43	4.91	22.32	31.82	0.59	1.36	17.23
	Average	**5.32**	**1.74**	**0**	**0.09**	**38.02**	**18.10**	**16.14**	**0.20**	**3.18**	**9.01**	**18.78**	**20.31**	**0.53**	**2.60**	**29.79**

although slightly higher than those of the Chun's method, are kept very low and close to the specified target. Nevertheless, in some sequences $\%Blocks$ becomes very low.

To complement these last results where the percentages of filtered blocks are low, we have also tested our method for $PSNR_{loss,tar} = 0.5dB$, so that $\%Blocks$ turns out to be significant in most of the sequences. In this case, FR and ΔNCC continue to be much higher than that of [10], but now it is in exchange for an also higher $PSNR_{loss}$. However, there are no significant differences in ΔBR terms between the obtained results with both $PSNR_{loss,tar}$ values.

In any case, it is worth recalling that our algorithm only filters the blocks where the flicker distortion is actually reduced. Thus, when a very low $PSNR_{loss,tar}$ is selected, $\alpha_{tar,i}$ tends to be very high, giving a high weight to the current block in the temporal filter equation in (1). In so doing, the filter is actually applied to a few blocks. In contrast, Chun's method selects the blocks to filter relying on a fixed threshold applied on a measure of the difference between original blocks of consecutive frames. As a result, Chun's method is not capable of dealing with the variable content of the video sequences.

As previously mentioned, we have also tested our proposal with respect to the method by Chono et al. [7]. The results obtained are shown in Table IV. The configuration is similar to that of the previous comparison, (i.e., since the method proposed in [7] incurred in low PSNR losses, we have set $PSNR_{loss,tar} = 0.2dB$ and $0.5dB$). The results are also quite similar: in terms of FR and ΔNCC, the proposed method achieves a much higher performance, in exchange for a slightly higher PSNR loss.

In this case, the results in terms of $\%Blocks$ deserve a comment. On the one hand, the results obtained by our method were as expected. The lower $PSNR_{loss,tar}$, the lower the number of filtered blocks. On the other, this measure is not so clear in Chono's method since almost every block is processed in this case. To provide a reasonable comparison, since the authors propose in their work that the flicker measure should be computed only on the blocks belonging to static regions, we have reported the percentage of blocks belonging to static regions as $\%Blocks$. In terms of ΔBR it should be highlighted that the results shown in Table IV are better than those of previous experiments. This is because in this case the RDO process was disabled and, consequently, the coding efficiency is actually lower and the impact of the proposed filtering method on this coding efficiency turns out to be negligible.

Regarding the computational cost, it should be said that Chun's method only requires to compute the flicker distortion to include it in the RDO cost function (barely increasing the computational cost), while both Chono's and the proposed methods require to carry out an ME process in the Intra-frames. Furthermore, as previously mentioned, the proposed method also requires to reconstruct four versions of each MB in the Intra-frames. Therefore, the excellent results achieved are in exchange for an increment of the computational cost associated with the encoding of the Intra-frames.

B. Subjective quality evaluation

In this section we have compared the proposed method to [10] and [7] in terms of subjective quality. As mentioned before, in our opinion, this type of assessment becomes critical when the goal is to assess the effectiveness of a method that aims to reduce a perceptual artifact, such as the flicker artifact. For these experiments, we have also followed the test conditions summarized in Table I.

Let us start by providing a detailed explanation of the experimental setup. A representative subset of coded sequences was randomly selected among all the sequences and QP values used in the previous assessments. A total of twenty four sequences (twelve with RDO enabled and twelve with RDO disabled) were picked. Four coded versions of each of these sequences were generated using four different coding methods: the reference software H.264/AVC JM15.1, our method with $PSNR_{loss,tar} = 0.5dB$ ($Prop0.5$), our method with $PSNR_{loss,tar} = 2dB$ ($Prop2$), and one of the state-of-the-art methods, [7] when RDO is disabled or [10] when RDO is enabled (hereafter, [7]/ [10]).

A total of 10 subjects participated in the subjective tests. In each trial, each subject was presented with three pairs of

TABLE IV
COMPARATIVE PERFORMANCE EVALUATION VS. [7] (RDO DISABLED).

		[7]					Proposed Method ($PSNR_{loss,tar} = 0.2dB$)					Proposed Method ($PSNR_{loss,tar} = 0.5dB$)				
	Sequence	FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ $loss$	ΔBR	$\%Blocks$	FR	ΔNCC	$PSNR$ loss	ΔBR	$\%Blocks$
CIF	**Container**	38.87	10.68	0.47	2.96	46.63	30.59	8.37	0.24	−0.21	17.73	41.79	11.24	0.62	−0.23	32.85
	Coastguard	0	0	0	0	0	24.61	23.72	0.08	−0.10	29.15	23.97	25.51	0.63	−0.34	49.04
	Bridge-far	−0.94	−0.14	0.19	0.78	1.85	33.23	7.90	0.21	−0.22	6.70	45.03	12.83	0.57	−1.38	18.23
	Bridge-close	−8.48	4.86	0.32	1.05	15.31	22.11	6.14	0.31	0.04	19.83	31.77	9.57	0.61	−0.51	40.99
	Flower	−0.62	4.89	0.16	2.17	27.52	22.12	45.06	0.48	0.14	28.05	27.56	58.79	0.77	0.12	37.59
	Nature	9.93	6.97	0.30	1.21	98.48	31.00	24.45	0.67	1.04	11.72	25.77	23.33	0.72	0.95	13.15
	Akiyo	14.09	0.65	0.56	1.70	85.60	26.38	11.74	−0.07	−0.15	7.51	32.35	9.83	0.61	−0.70	12.47
	Football	15.53	15.11	1.12	0.34	16.66	20.46	8.70	0.19	−0.01	26.79	25.14	9.68	0.63	−0.11	42.56
SD	**Corvette**	17.31	6.68	0.80	0.28	62.85	21.65	6.84	0.28	−0.12	13.45	27.29	7.51	0.64	−0.11	23.24
	Ice Age	11.94	2.02	0.37	−0.32	99.83	21.80	2.13	0.42	−0.02	5.84	27.72	4.35	0.63	0.14	11.79
	Last Samurai	16.85	4.46	0.29	−0.47	95.90	30.64	6.85	0.47	−0.04	5.88	37.68	9.10	0.62	−0.63	11.09
	Shields	0	0	0	0	0	26.62	5.46	0.05	0.48	20.50	28.50	8.26	0.64	0.56	30.02
	Mobisode	4.00	3.05	0.35	0.45	27.13	52.37	1.39	1.37	0.49	14.40	44.10	2.97	0.61	0.64	15.83
HD	**Pedestrian**	6.67	8.53	0.79	0.02	27.82	35.44	2.13	0.77	0.03	12.35	29.83	10.29	0.59	0	15.02
	Rush Hour	−0.03	3.93	0.23	0.32	10.73	37.13	10.71	0.39	0.13	9.37	30.93	17.33	0.58	0.13	10.83
	Average	**8.34**	**4.77**	**0.39**	**0.69**	**41.08**	**29.07**	**11.43**	**0.39**	**0.09**	**15.28**	**31.96**	**14.70**	**0.63**	**-0.09**	**24.31**

TABLE V
SUBJECTIVE QUALITY EVALUATION. OUR PROPOSAL VS. [10] AND [7].

	RDO enabled			RDO disabled		
Viewer	**[10]**	$PSNR_{loss,tar} = 0.5dB$	$PSNR_{loss,tar} = 2dB$	**[7]**	$PSNR_{loss,tar} = 0.5dB$	$PSNR_{loss,tar} = 2dB$
#1	4	4	5	3	5	3
#2	0	3	4	1	2	7
#3	1	6	5	3	4	7
#4	3	5	7	7	4	10
#5	2	8	10	4	5	9
#6	4	9	7	3	6	8
#7	4	3	8	5	6	7
#8	4	2	6	0	0	4
#9	1	4	7	2	1	5
#10	5	4	10	5	5	7
Total (%)	**23.3**	**40.0**	**57.5**	**27.5**	**31.6**	**55.8**

encoded versions of the same sequence, where the first sequence of each pair was always encoded with the H.264/AVC reference software (Ref), and the second sequence was encoded using one of the compared method ($Prop0.5$, $Prop2$ or [7]/ [10], selected in random order). For each of the three pairs, the subjects were asked to tell if they actually perceived a subjective quality improvement or not when comparing the second sequence with respect to the first one, thus evaluating the improvement with respect to the reference software. Therefore, the number of times that the subjects expressed a real preference for the second sequence was our figure of merit. The experimental protocol is summarized in Fig. 9 and the complete subjective quality test is available at [24].

Table V shows the percentage of times that each proposal was chosen considering all the answers and all the viewers. As can be seen, the proposed method, for both $PSNR_{loss,tar}$ values, provided a subjective quality improvement for a higher percentage of cases than the methods reported in [10] and [7]. Specifically, our method with $PSNR_{loss,tar} = 2dB$ was chosen as providing a perceptual improvement more than twice the times than either [10] (RDO enabled) or [7] (RDO disabled). When $PSNR_{loss,tar} = 0.5dB$ was used, the results were not so good but they were still quite superior to those of the compared methods.

We would like to add another consideration regarding the results obtained by the two realizations of our method, for $PSNR_{loss,tar} = 0.5dB$ and $PSNR_{loss,tar} = 2dB$. It seems clear that when the $PSNR_{loss,tar}$ is higher, the method is acting more intensively against the flicker artifact, but in exchange for increasing the blurring effect. Nevertheless, as it can be observed from the previous results, when we set a higher $PSNR_{loss,tar}$, the perceived quality is better. This result highlights that our method, which adaptively control the PSNR losses, achieves very positive effects, allowing high $PSNR_{loss,tar}$ (and thus high flicker reductions) without impoverish the subject's quality perception.

V. CONCLUSIONS AND FURTHER WORK

In this work, we have proposed a standard-compliant flicker reduction method based on a temporal motion-guided low-pass filtering. Specifically, the proposed filtering process reduces the difference between reconstructed pixels of the same region in consecutive frames, reducing the flicker artifact and improving the visual quality. Moreover, to mitigate the appearance of side-effects owing to the low-pass filtering, such as blurring, we have proposed an *on-the-fly* method to control the filter strength so that the algorithm is capable to meet a target PSNR loss.

We have shown experimentally that the proposed method achieves significant flicker reductions for a wide range of sequences exhibiting different type of contents and resolutions, achieving in some cases flicker reductions above 34% with respect to the H.264/AVC reference software. We have also compared our method to two state-of-the art methods. Although these methods incur in very low PSNR losses, they offer a limited capacity of flicker reduction. In contrast, the proposed method is able to reduce the flicker distortion substantially at the same time that allows for keeping the PSNR

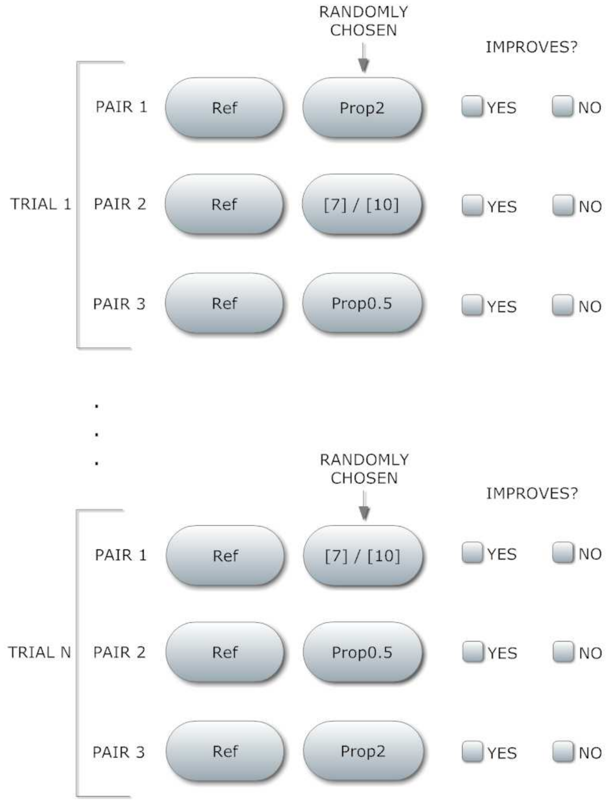


Fig. 9. Experimental protocol for the subjective quality evaluation.

loss under a target that is user-defined, all in exchange for very limited losses in terms of coding efficiency. Furthermore, our experiments have shown the effectiveness of the method to meet the target PSNR loss in a variety of test conditions.

We have also conducted a subjective evaluation that involved 10 subjects to compare the proposed method to the same state-of-the-art references with excellent results. The best realization of our algorithm achieves more than two times better results than the reference methods.

As suggested by one of the anonymous reviewers, a subtle modification of the filtering scheme should be explored as future work. Specifically, the input to the proposed low-pass temporal filtering could be the current original block ($f_{n,i}$) instead of the current reconstructed one ($\hat{f}_{n,i}$). See Fig. 3.

Another interesting direction for future work would focus on considering a frame-level PSNR loss constraint in (5). In this manner, the problem should be stated as that of choosing the α value for every block to minimize the flicker distortion of the whole frame while keeping the PSNR loss of this frame under a certain target. The solution to this problem can be addressed by means of Lagrange optimization [25].

An additional future line of research would focus on reducing the computational cost associated with the *on-the-fly* selection of the filter parameter, and on adapting this proposal to the new video coding standard HEVC [26].

REFERENCES

[1] H. Wu and K. Rao, *Digital video image quality and perceptual coding*. CRC, 2005, vol. 28.

[2] A. Becker, W. Chan, and D. Poulouin, “Flicker reduction in intraframe codecs,” in *Data Compression Conference, 2004. Proceedings. DCC 2004*, Mar. 2004, pp. 252 – 261.

[3] T. Kuge, “Wavelet picture coding and its several problems of the application to the interlace HDTV and the ultra-high definition images,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, Jun. 2002, pp. III–217 – III–220 vol.3.

[4] Y. Itani and H. Watanabe, “A study on flicker reduction based on distinction between static and moving region in Motion JPEG 2000,” in *IEICE Tech. Rep.*, vol. 105, no. 164, 2005, pp. 77–82.

[5] A. Leontaris, Y. Tonomura, and T. Nakachi, “Rate control for flicker artifact suppression in Motion JPEG2000,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, 2006, pp. II–II.

[6] S.-H. Hong, S.-J. Yoo, S.-W. Lee, H.-S. Kang, and S. Y. Hong, “Rate control of MPEG video for consistent picture quality,” *Broadcasting, IEEE Transactions on*, vol. 49, no. 1, pp. 1–13, 2003.

[7] K. Chono, Y. Senda, and Y. Miyamoto, “Detented quantization to suppress flicker artifacts in periodically inserted intra-coded pictures in H.264 video coding,” in *Image Processing, 2006 IEEE International Conference on*, Oct. 2006, pp. 1713 –1716.

[8] H. Yang, J. Boyce, and A. Stein, “Effective flicker removal from periodic intra frames and accurate flicker measurement,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct. 2008, pp. 2868 –2871.

[9] S. S. Chun, H.-J. Ryu, and S. Sull, “Flicker reduction in intra coded frames of H.264/AVC,” in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 2025–2028.

[10] S. S. Chun, J.-R. Kim, and S. Sull, “Intra prediction mode selection for flicker reduction in H.264/AVC,” *Consumer Electronics, IEEE Transactions on*, vol. 52, no. 4, pp. 1303 –1310, Nov. 2006.

[11] X.Fan, W.Gao, Y.Lu, and D.Zhao, “Flicking Reduction in All Intra Frame Coding,” JVT-E070, ISO/IEC MPEG and ITU-T VCEG Joint Video Team, Geneva, Oct. 2002.

[12] D. Vo, T. Nguyen, S. Yea, and A. Vetro, “Adaptive fuzzy filtering for artifact reduction in compressed images and videos,” *Image Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1166 –1178, Jun. 2009.

[13] J. X. Yang and H. R. Wu, “Robust filtering technique for reduction of temporal fluctuation in H.264 video sequences,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 3, pp. 458 –462, Mar. 2010.

[14] A. Matsumura, S. Naito, R. Kawada, and A. Koike, “Effective rate control method for minimizing temporal fluctuations in picture quality applicable for MPEG-4 AVC/H.264 encoding,” in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1, Sept. 2005, pp. I – 569–72.

[15] S. Kanumuri, O. G. Guleryuz, M. R. Civanlar, A. Fujibayashi, and C. S. Boon, “Temporal flicker reduction and denoising in video using sparse directional transforms,” *Proc. SPIE 7073*, 2008.

[16] J. Yang, J. B. Park, and B. Jeon, “Flickering effect reduction for H.264/AVC intra frames,” in *Multimedia Systems and Applications IX*, 2006.

[17] Y. Kuszpet, D. Kletsel, Y. Moshe, and A. Levy, “Postprocessing for flicker reduction in H.264/AVC,” in *Proc. 16th Picture Coding Symposium (PCS), Lisbon, Portugal, 2007*, Nov. 2007.

[18] J. Ren, J. Liu, M. Li, and Z. Guo, “Postprocessing of block-coded videos for deflicker and deblocking,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 1631–1635.

[19] S. Qiao, Y. Zhang, and H. Wang, “Pi-frames for flickering reduction in h.264/avc video coding,” in *Computer Science Service System (CSSS), 2012 International Conference on*, 2012, pp. 1551–1554.

[20] E. Martinez-Enriquez, A. Jimenez-Moreno, and F. Diaz-de Maria, “An adaptive algorithm for fast inter mode decision in the H.264/AVC video coding standard,” *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 2, pp. 826–834, 2010.

[21] A. Jimenez-Moreno, E. Martinez-Enriquez, and F. Diaz-de Maria, “Standard compliant flicker reduction method with PSNR loss control,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 1729–1733.

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

[23] “JVT H.264/AVC reference software v15.1.” [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_tml/

[24] Subjective quality evaluation [online], “http://www.tsc.uc3m.es/∼ajimenez/flicker-transactions/”.
[25] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 23 –50, Nov. 1998.
[26] High Efficiency Video Coding [online], “http://hevc.hhi.fraunhofer.de/”.

Amaya Jiménez-Moreno received the Telecommunications Engineering degree from Universidad Carlos III de Madrid, Madrid, Spain in 2012. She is currently working toward the Ph.D. degree in the same university. Her research interests include video coding optimization, high definition video coding, and medical image processing.

Eduardo Martínez-Enríquez (SM’07) received the Telecommunications Engineering degree from Universidad Politcnica de Madrid, Madrid, Spain, in 2006, and Ph.D. degree from the Universidad Carlos III de Madrid, Spain, in 2013. His research interests include lifting transforms on graphs, wavelet-based video coding and video coding optimization. He received the Best Paper Award of ICIP in 2011 for his paper on video coding based on lifting transform on graphs, co-authored with Fernando Daz-de-Mara and Antonio Ortega.

Vinay Kumar completed his Ph.D. from Jaypee University of Info Tech, India. He is associated with Multimedia Processing Group, University Carlos III at Madrid, Spain. Currently he is working at Thapar University, Patiala, India. His interests include image and video processing.

Fernando Díaz-de-María (M’97) received the Telecommunication Engineering degree and the Ph.D. degree from the Universidad Politecnica de Madrid, Madrid, Spain, in 1991 and 1996, respectively. Since October 1996, he has been an Associate Professor in the Department of Signal Processing and Communications, Universidad Carlos III de Madrid, Madrid, Spain. His primary research interests include robust speech recognition, video coding, and video analysis. He has led numerous projects and contracts in the fields mentioned. He is co-author of several papers in peer-reviewed international journals, two book chapters, and has presented a number of papers in national and international conferences.