Zonal Brightness Coherency for Video Tone Mapping

Ronan Boitard[a,b], Rémi Cozot[b], Dominique Thoreau[a], Kadi Bouatouch[b]

[a] *Technicolor, 975 avenue des Champs Blancs, 35576 Cesson-Sevigne, France*
[b] *IRISA, 263 Avenue du Général Leclerc, Rennes, France, ronan.boitard@irisa.fr*

Abstract

Tone Mapping Operators (TMOs) compress High Dynamic Range (HDR) contents to address Low Dynamic Range (LDR) displays. While many solutions have been designed over the last decade, only few of them can cope with video sequences. Indeed, these TMOs tone map each frame of a video sequence separately, which results in temporal incoherency. Two main types of temporal incoherency are usually considered: flickering artifacts and temporal brightness incoherency. While the reduction of flickering artifacts has been well studied, less work has been performed on brightness incoherency. In this article, we propose a method that aims at preserving spatio-temporal brightness coherency when tone mapping video sequences. Our technique computes HDR video zones which are constant throughout a sequence, based on the luminance of each pixel. Our method aims at preserving the brightness coherency between the brightest zone of the video and each other zone. This technique adapts to any TMO and results show that it preserves well spatio-temporal brightness coherency. We validate our method using a subjective evaluation. In addition, unlike local TMOs, our method, when applied to still images, is capable of ensuring spatial brightness coherency. Finally, it also preserves video fade effects commonly used in post-production.

Keywords: Video Tone Mapping, HDR Video, Temporal Coherency, HDR Segmentation, Temporal Adaptation, HDR Video Fade

1. Introduction

Tone mapping is a technique that converts HDR images or HDR video sequences to a lower dynamic range. Common applications map real world luminance to the display capabilities of a media, namely LCD monitors, print-outs, etc. This mapping relies on Tone Mapping Operators (TMOs). Tone mapping static images has been a field of active research over the last decade and several satisfying solutions exist [1, 2, 3]. Thanks to recent developments in the HDR video acquisition field [4, 5], more and more HDR video contents are available. However, applying TMOs naively to each frame of an HDR video sequence leads to temporal artifacts.

Indeed, many issues arise when dealing with video tone mapping, they are usually due to a lack of temporal coherency. Over small time intervals, rapid changes in the scene introduce flickering artifacts. Several solutions exist to deal with this problem [6, 7, 8, 9, 10, 11]. These solutions use temporally close frames to smooth out abrupt changes of luminance. Further details of these solutions are given in [12]. However, longer time ranges introduce temporal brightness incoherency which is not addressed by these solutions. We consider the brightness to be coherent in a tone mapped content when the relative brightness between two zones of the HDR sequence is preserved during the tone mapping (both in the spatial and temporal domain). As a TMO uses for each frame all the available display range, this HDR brightness coherency is not preserved throughout the tone mapping operation. Consequently, areas perceived as the brightest in the HDR sequence are not necessarily the brightest in the LDR one.

One solution to solve this problem is the Brightness Coherency (BC) algorithm [13]. It relies an indication of the overall brightness of each frame of a video sequence to preserve the temporal brightness coherency. By using the brightest frame in the HDR video sequence (say the frame with the highest indication of overall brightness) as an anchor, each other tone mapped frames is scale down relative to this anchor. As each frame is processed relative to the anchor, this solution preserves temporal brightness coherency even between non-adjacent frames. Furthermore, it can be used jointly with the existing flickering artifacts removal solutions. One limitation of this technique is that it deals only with the overall brightness without solving local brightness coherency. Moreover, it trades off spatial contrast to increase the temporal brightness coherency as each pixel of each frame is scaled down without discrimination. In other terms, no pixel, apart from those belonging

to the anchor frame, is mapped to the maximum of the dynamic range. Consequently, the range associated with each frame corresponds to a portion of the available dynamic range (this portion being defined by the scaling).

In this article, we propose to improve the Brightness Coherency (BC) algorithm so as to overcome these limitations. Our new video TMO aims at preserving the local brightness coherency as well as the contrast in each frame. To this end, we divide each frame into segments using a histogram-based segmentation of HDR images. We then define video zones thanks to the resulting frame's segments. Finally, we apply the Brightness Coherency algorithm to each video zone independently. This Zonal Brightness Coherency (ZBC) algorithm better preserves both the temporal brightness coherency and the spatial contrast in each frame. In addition, this solution is also efficient to deal with video fade effects.

In the next section, both the Brightness Coherency algorithm [13] and one of the flickering removal TMO [10] are detailed. The histogram-based segmentation algorithm and its use in the ZBC algorithm are presented in section 3. Section 4 compares the results obtained with the ZBC method to those provided by other TMOs. It also evaluates the efficiency of the ZBC algorithm for video fade sequences. To validate our solution, a subjective evaluation has been conducted, it is presented in section 5. Finally, we conclude and present future works in the last section.

2. Previous Work

A limitation of TMOs is their inability to preserve the brightness coherency throughout a video. As these TMOs provide the best result achievable depending on the available dynamic range, successive frames can be tone mapped quite differently. In other words, abrupt changes of the tone map curve are source of flickering. Temporal filtering is a common technique to smooth out flickering artifacts. In this section, we first present a flickering artifact removal solution designed by Ramsey et al. [10]. This solution is a temporal adaptation of the Photographic Tone Reproduction (PTR) algorithm [14]. In case of gradual changes, flickering removal solutions are less efficient as they do not preserve the temporal brightness coherency. A solution, which performs a post-processing of any TMO's output, has been proposed in [13]. It is presented in subsection 2.2.

2.1. Temporal Adaptation of the Photographic Tone Reproduction

The PTR algorithm uses a system designed by Adams [15] to rescale HDR frames at a defined exposure:

$$\mathbf{L_s} = \frac{\epsilon}{k}\mathbf{L_w}\,, \tag{1}$$

$$k = \exp\left(\frac{1}{n_p}\sum_{x=1}^{n_p}\log(d + L_w(x))\right)\,, \tag{2}$$

where ϵ is the chosen exposure, $\mathbf{L_w}$ the HDR luminance image and $\mathbf{L_s}$ the scaled luminance image. The key value k is an indication of an image overall brightness. It is computed using equation (2), where d is a small value (i.e. $10^{-6}cd/m^2$) to avoid singularity and n_p the number of pixels in the image. The tone mapping curve is a sigmoid function given by equation (3):

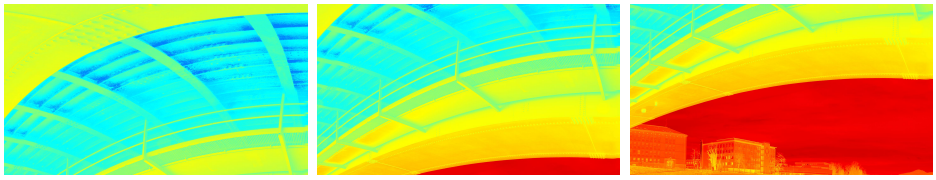
$$\mathbf{L_d} = \frac{\mathbf{L_s}}{1 + \mathbf{L_s}}\left(1 + \frac{\mathbf{L_s}}{\omega^2}\right)\,, \tag{3}$$

where ω is used to burn out areas with high luminance value and $\mathbf{L_d}$ is the tone map LDR luma. Two parameters (ϵ and ω) are then necessary to compute the TMO results. In [14], these parameters are set to $\epsilon = 18\%$ and ω to the maximum luminance value of $\mathbf{L_s}$.

The main issue with this algorithm is that flickering artifacts appear for abrupt changes of the key value. To avoid these artifacts, Kang et al. [7] proposed to filter the current frame's key value k using a set of the n_f previous frames key values. Ramsey et al. [10] further refined this idea by making n_f adaptive. This adaptation discards outliers using a min/max threshold. In the rest of this article, $PTR + R$ stands for Ramsey et al. extension of the Photographic Tone Reproduction. The $PTR+R$ operator removes flickering artifacts efficiently, however the lack of temporal brightness coherency is still an issue. Figure 1 illustrates the temporal brightness incoherency when tone mapping an HDR video sequence (using the $PTR + R$). Notice how the brightness of the downside of the bridge changes in the tone mapped LDR frames. A solution to cope with this problem is presented in the next subsection.

2.2. Brightness Coherency for Video Tone Mapping

To better preserve the temporal brightness coherency, Boitard et al. [13] developed a method which adapts to any TMO through a post-processing



(a) False color luminance, red correspond to the maximum luminance of the video while blue its minimum

(b) Tone mapped result using the $PTR + R$ [10] operator

Figure 1: Example of brightness incoherency after tone mapping. (1a) illustrates false color luminance of frames 50, 100 and 150 of the $UnderBridgeHigh$ sequence. (1b) represents the LDR results of the same frames after applying the $PTR + R$ operator.

operation. This technique uses the frame key value k (computed using equation 2) to preserve the HDR brightness ratio (say the HDR brightness of a frame relative to the anchor) in the tone mapped LDR sequence.

The HDR brightness ratio is equal to the LDR brightness ratio if

$$\frac{k_f^{i,HDR}}{k_v^{HDR}} = \frac{k_f^{i,LDR}}{k_v^{LDR}}\,, \tag{4}$$

where $k_f^{i,HDR}$ is the i^{th} HDR frame key value and k_v^{HDR} the highest key value of the sequence (corresponding to the brightest frame, i.e. anchor). Similarly, $k_f^{i,LDR}$ and k_v^{LDR} are respectively the i^{th} LDR frame key value and the key value of the tone mapped version of the anchor. To satisfy equation (4), the tone mapped luma $\mathbf{L_d}^i$ of the i^{th} frame is scaled according to equation (5) to get the BC post-processed tone map luma $\mathbf{L^i_{BC}}$:

$$\mathbf{L^i_{BC}} = \left(\zeta + (1-\zeta)\frac{k_f^{i,HDR}k_v^{LDR}}{k_v^{HDR}k_f^{i,LDR}}\right)\mathbf{L^i_d} = s^i\mathbf{L^i_d} \tag{5}$$

where s^i represents the scale ratio of the i^{th} frame and ζ is a user-defined parameter to avoid low scale ratio. In order to determine the anchor, i.e. the

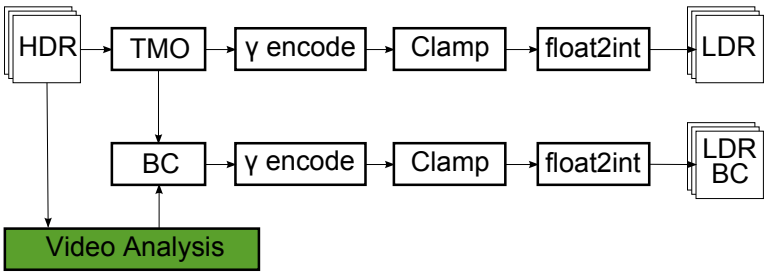
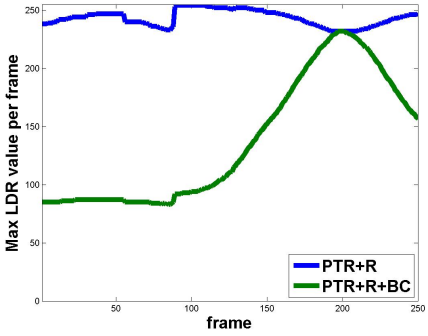


Figure 2: General workflow for tone mapping a video sequence with and without the BC algorithm. The video analysis (green box) is performed as a preprocessing step. "γ encode" distributes the tones roughly evenly across the entire range (usually $\gamma = 1/2.2$). "Clamp" clips the values outside the range [0, 1]. "float2int" converts floating point values into integers.

frame with the maximum HDR frame key value, a video analysis is performed prior to the tone mapping operation. Figure 2 depicts the workflow used to tone map a video with and without preserving the brightness coherency.

Similar to the method proposed in [16], the BC technique performs well when the brightness fluctuations in the scene change in a global way. However, for local fluctuations, this technique scales similarly each pixel of a frame, resulting in a lack of spatial contrast due to the reduced dynamic range. Figure 3a plots the used dynamic range of the tone mapped $UnderBridgeHigh$ sequence with and without using the BC algorithm. This reduced dynamic range entails a loss of contrast as illustrated in figure 3b. In this frame, as only the overall brightness coherency is preserved, the brightness of the sky is also scaled down although it is the brightest element in the video sequence. From this example, we see that preserving only the overall brightness coherency is not sufficient. Indeed, using only one key value per frame is not enough as shown in figure 4. The image (fig. 4a) can be segmented into three spatial areas (two green and one blue) while in the luminance domain, the histogram (fig. 4b) suggests only two segments (leftmost and rightmost of the histogram). A single key value (green line in fig. 4b) represents none of the mentioned zones.

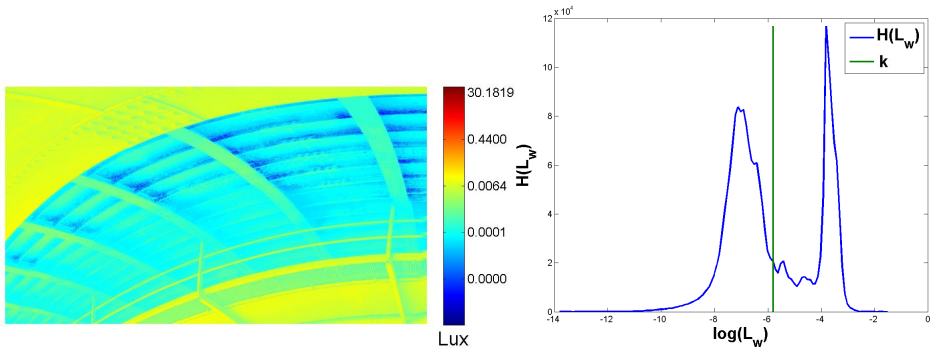
To sum up, preserving the overall temporal brightness coherency results in smooth frames with too low contrast (dim zones). Using one key value per zone instead of one key value per frame preserves the coherency of the local and overall temporal brightness as shown in the next section.



(a) Max LDR luma with and w/o using BC

(b) Loss of contrast using BC

Figure 3: Loss of dynamic range and loss of contrast due to the BC algorithm. (3a) represents the maximum LDR luma value of the $UnderBridgeHigh$ sequence with and without using the BC algorithm. Without brightness coherency, the maximum LDR luma remains close to 255 (maximum value in 8 bits). On the other hand, the BC technique reduces the used dynamic range up to 2/5 of its maximum value. (3b) shows frame 120 of the $UnderBridgeHigh$ sequence with (lower frame) and without (upper frame) the BC algorithm. Notice the loss of contrast between the brightness of the sky and the downside of the bridge.



(a) False color HDR luminance

(b) Histogram and k value

Figure 4: Spatial areas and luminance segments for frame 50 of the $UnderBridgeHigh$ sequence. False color luminance (4a) shows the spatial areas, while the luminance histogram (4b) represents the luminance segments. The vertical green line in 4b indicates the location of the frame's key value.

3. Zonal Brightness Coherency

3.1. Overview

In the previous section, we outlined the issues related to the BC algorithm. They are due to the use of only one key value k_f per frame that represents only an indication of a frame's overall brightness. However, preserving the overall brightness coherency does not preserve local brightness coherency.

In order to improve the preservation of the temporal brightness coherency, we propose to apply the BC algorithm to zones rather than to a whole frame. To this end, a histogram-based segmentation algorithm divides each frame into segments in the luminance domain. As the segment's boundaries change from frame to frame, flickering artifacts may appear as detailed in section 3.2. To prevent flickering, we compute video zones based on the segments' key values. We define each video zone by two luminance values kept constant throughout the video sequence. The frame segmentation as well as the video zones determination are performed during the video analysis step (fig. 2). To apply the BC per zone, we compute a scale ratio per zone instead of per frame. As in the BC algorithm, the ZBC scale ratio is used to process any TMO's output.

The rest of this section is organized as follows. First, we present a technique to segment an HDR frame based on its histogram. Then, we show how to compute video zones based on the segments' key values. The application of the BC algorithm to each zone and its boundaries are then detailed. In subsection 3.5, we explain how to use the ZBC algorithm to preserve video fade effects. We then discuss, in subsection 3.6, the tuning of the parameters of our algorithm. Finally, we summarize each step of the ZBC algorithm.

3.2. Frame Segmentation

Segmenting a frame is usually done either in the spatial or the luminance domain. We chose to segment the HDR frames in the luminance domain for the following reasons. First, we want to preserve the spatial brightness coherency. When using the luminance, we make sure that two spatially distant areas of same luminance have the same scale ratio. Second, spatial segmentation is never perfect. Third, spatial areas cannot be kept coherent throughout the video, especially when objects appear or disappear.

Our segmentation consists of several successive operations:

1. Compute the luminance histogram of each frame (w_b, log domain)
2. Find local maxima in the histogram (higher than threshold t)
3. Remove local maxima that are too close to each other (distance ρ)
4. Find local minima between successive local maxima
5. Define local minima as segment's boundaries
6. Compute the key value for each segment (equation 2)

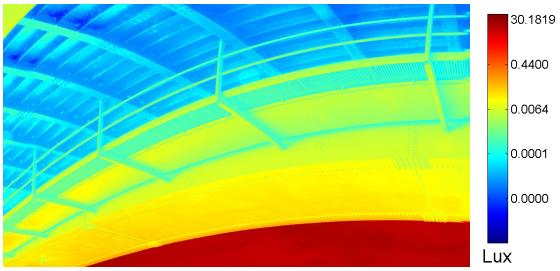
Three parameters, namely w_b, t and ρ, allow to tune the segmentation. w_b corresponds to the width of each bin of the histogram (expressed in the log domain). Each local maximum of the histogram must be higher than the threshold t. Finally, ρ corresponds to the minimum distance between two local maxima (in log scale). The effects of these parameters on the segmentation are detailed in subsection 3.6.

The whole frame segmentation process is summarized in figure 5. In this example, the algorithm efficiently divides the HDR frame into three separate luminance segments. Figure 6 presents, in false color, the corresponding luminance segments computed in figure 5.

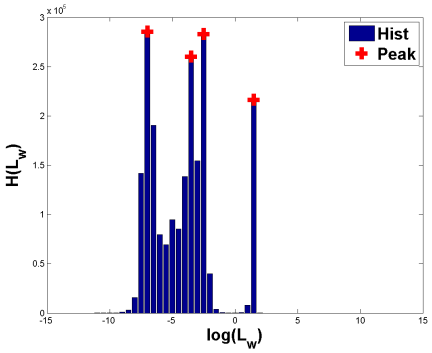
We now have several segments per frame whose boundaries change from frame to frame. Figure 7 plots the video analysis of the $UnderBridgeHigh$ sequence, showing the results obtained for one key value k_f per frame (fig. 7a) and each j^{th} segment's key value k_s^j per frame (fig. 7b). However, the change of segment's boundaries throughout the video is source of flickering artifacts. Large variations of segment boundaries between successive frames result in different key values and hence different scaling factors. Figure 8 illustrates a change of boundaries between frames 127 and 128 of the $UnderBridgeHigh$ sequence. To avoid this problem, we compute video zones that have constant boundaries throughout the video, they are detailed in the next subsection.

3.3. Video Segmentation

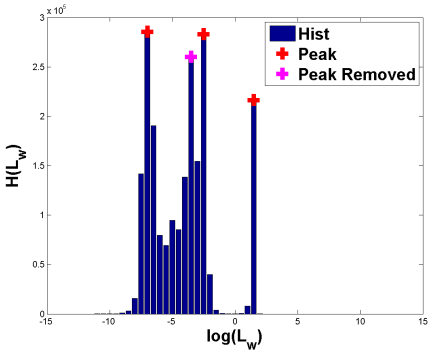
Frame segmentation results in several key values k_s per frame. In this subsection, we explain how to determine the video zones based on the segments' key values. We chose to use the segments' key values because they represent an information on the distribution of the luminance values of each frame. Computing video zones on a histogram containing all the pixels of each frame of the video sequence would not take into account this information.



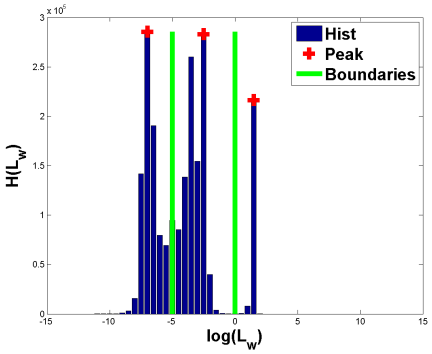
(a) False color luminance



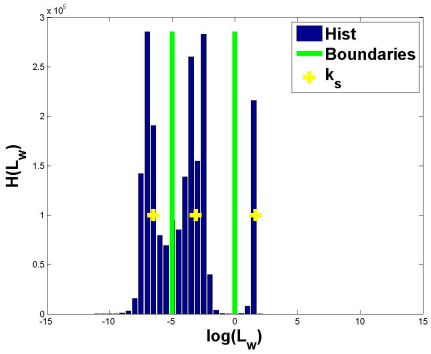
(b) Compute histogram and find local maxima



(c) Remove close local maxima

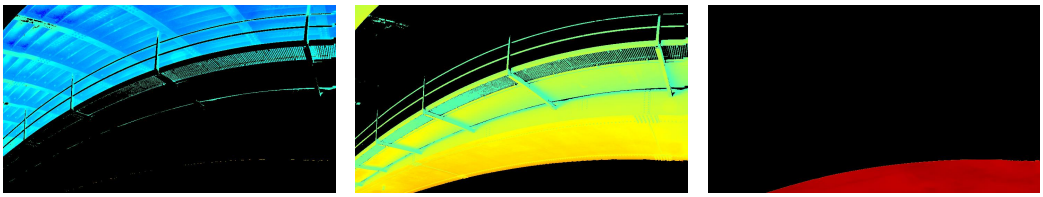


(d) Find local minima and set as segment boundary



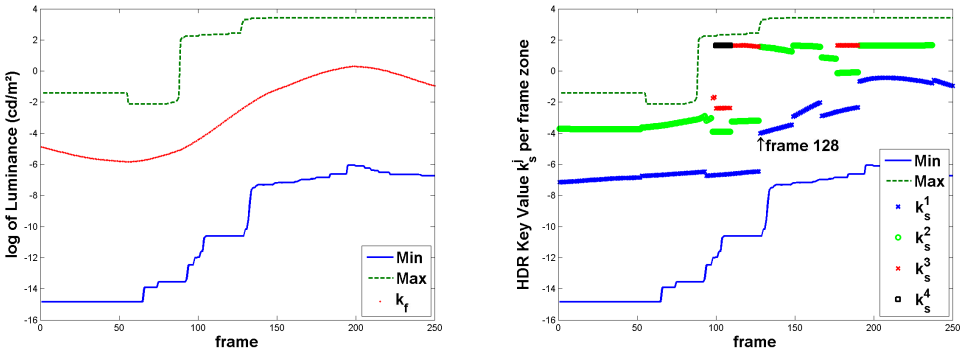
(e) Compute key value per segment

Figure 5: Example of histogram-based segmentation with frame 108 of the *UnderBridgeHigh* sequence. (5a) illustrates this frame in false color luminance while (5b) to (5e) summarize all the segmentation steps.



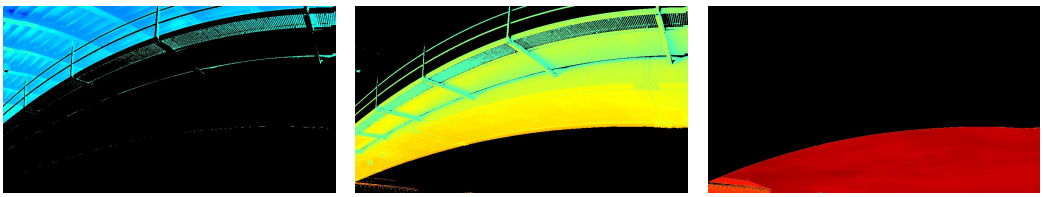
(a) Dim value segment (b) Mid-value segment (c) Bright value segment

Figure 6: Segmentation of frame 108 of the UnderBridgeHigh sequence. Images are displayed in false color luminance. Spatially close pixels may appear in different segments if their luminance is close to a boundary (cyan pixel in both (6a) and (6b)).

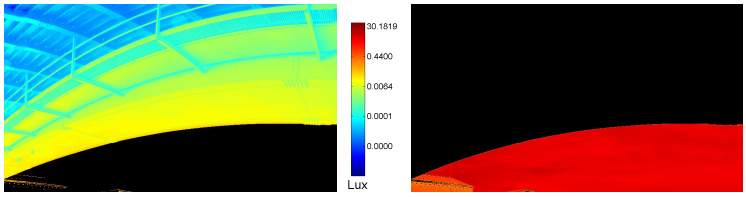


(a) Min, max and key value k_f per frame (b) Min, max and j^{th} frame segment's key value k_s^j per frame

Figure 7: (7a) represents the min, max and key value per frame (*UnderBridgeHigh* sequence). (7b) plots for each frame the key value k_s^j of the j^{th} segment. In this sequence, up to 4 segments are detected (around frame 100). The discontinuity at frame 128 is illustrated in figure 8.



(a) Frame 127, segment 1, 2 and 3



(b) Frame 128, segment 1 and 2

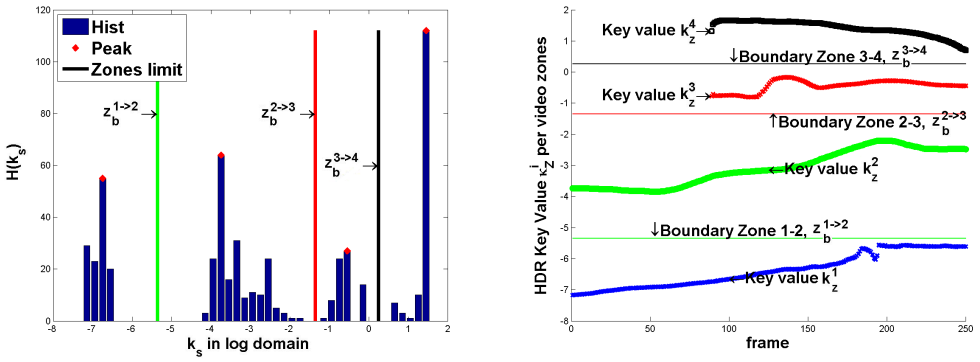
Figure 8: Example of change of segment's boundaries. (8a) represents the three resulting segments obtained using the histogram-based segmentation of frame 127. When applied to frame 128, only two segments remain. This change of boundaries corresponds to the discontinuity of the green and black curves in figure 7b.

We want to compute video zones that best fit the distribution of the segments' key values of all the frames. We chose to use the same histogram-based segmentation as in subsection 3.2, the luminance values being replaced by the segments' key values k_s of each frame. Figure 9 plots the results of the segmentation of the key value histogram for the $UnderBridgeHigh$ sequence. Figure 9a represents the segmented key value histogram, z_b being the zones' boundaries delimiting the video zones to which we apply the BC algorithm. Figure 9b plots again the video zones' boundaries along with the video zones' key values k_z for each frame. We see that the key value within each zone varies continuously, which ensures temporal coherency.

Now that we have video zones, we apply the BC algorithm to each video zone of each frame. We consider two cases: a pixel luminance lies within a video zone or on the boundary.

3.4. Applying BC to video zones

After computing a key value for each video zone of each frame, we apply the BC algorithm to each zone. For each pixel located within a video zone (say, its luminance lies in-between the boundaries of a video zone), we apply



(a) Segmented key value histogram (b) Zone's boundary and key value

Figure 9: Video segmentation for the *UnderBridgeHigh* sequence. (9a) represents the segmented key value histogram where $z_b^{j->j+1}$ represents the video zone boundary between zones j and j+1. (9b) delimits the video zones and plots the j^{th} zone key value k_z^j for each frame.

the scale ratio defined in equation 5. In this case, the key value $k_f^{i,HDR}$ of the i^{th} HDR frame is replaced by the key value of the j^{th} zone of the i^{th} HDR frame $k_z^{i,j,HDR}$. Similarly, the highest video zone's key value of the HDR sequence is denoted k_{vz}^{HDR} while k_{vz}^{LDR} denotes its corresponding LDR key value.

At each video zone's boundary, we define a blending zone (b_z) containing all the pixels whose luminance is close to this boundary up to a distance δ. To each pixel luminance within a b_z, we assign a scale ratio s which is a combination of those of the video zones sharing this boundary. This weighted scale ratio prevents abrupt scale ratio changes that could cause high spatial contrast not present in the HDR frame. δ is a user-defined parameter in the log domain. The weight w_l of the lower scale ratio s_l assigned to a pixel x lying in a blending zone is defined as follows:

$$w_l(x) = \frac{exp\left(-\frac{(\log(L_w(x))-\log(b_l))^2}{2\sigma^2}\right)}{s^h + s^l}, \tag{6}$$

where the luminance b_l represents the lower bound of b_z. s^h and s^l are the scale ratios, used in equation (5), of the two video zones sharing the boundary and are used to normalize each weight. To compute the weight w_h of the higher scale ratio s_h, b_l is replaced by the higher bound b_h in equation

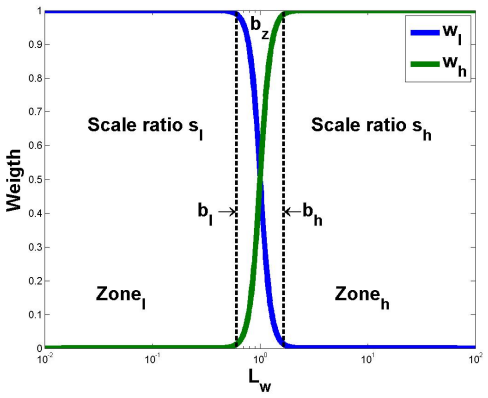


Figure 10: Weight distribution function of the blending zone. w_l corresponds to the weight of $Zone_l$ scale ratio and w_h the weight of $Zone_h$ scale ratio.

(6). As for σ, it corresponds to the width of the weighting function support:

$$\sigma = \frac{\delta}{2\sqrt{2\log(3)}} \tag{7}$$

Finally the scale ratio $s^{l,h}$ of a pixel is given by:

$$s^{l,h} = s^l w_l + s^h w_h, \tag{8}$$

where δ is the boundary width expressed in the log domain (set to 1 for all our experiments). Figure 10 plots the weight distribution function in a blending zone b_z. Outside the zone delimited by b_l and b_h, the pixels are scaled by only one zone scale ratio. Inside this zone, the scale ratio is computed using equation (8).

Recall that with this technique we preserve the local brightness coherency without loss of spatial contrast. Another goal of our technique is to preserve video fade effects in video sequences as explained in the following subsection.

3.5. Video Fade

A video fade occurs when a shot gradually fades to (or from) a single color, usually black or white. In LDR video processing, a fade to black is usually computed using an alpha-blending:

$$\mathbf{R^i} = \alpha_i \mathbf{O^i} + (1 - \alpha_i)\mathbf{F}, \tag{9}$$

where $\mathbf{O^i}$ is the i^{th} original frame of the sequence, $\mathbf{F}$ a black image and $\mathbf{R^i}$ the i^{th} resulting fade to black frame. α_i ranges from 1 to 0 and controls the speed of the fading. We consider the speed of the fading as the number of intermediate frames n_i needed to satisfy $\mathbf{R^i} = \mathbf{F}$.

Although video fade effects are well known in LDR video processing, HDR video fade has not been addressed yet. This is due to the fact that HDR luminance represents a photometric quantity expressed in cd/m^2, while LDR luma represents a code value relative to a standard. However, a black or white point is not defined in HDR. For this reason, we define the white point as the maximum HDR luminance value of the image and the black point its minimum. To make the video fade effect perceptually linear, we apply equation (9) in the log domain.

To preserve video fade effects, some modifications have to be brought to the ZBC algorithm. First, the parameter ζ (equation 5) must be set to 0. Second, the fading's speed of the HDR sequence has to be known to fit the LDR sequence. Without these modifications, fading still works but at a different pace. To deal with fade to white, the anchor k_{vz}^{HDR} must be defined as the dimmest zone rather than the brightest.

To sum up, the ZBC algorithm deals with fade to black effects without any modification while it needs some changes for fade to white effects. To deal with different effects in the same sequence, we introduce a variable ψ that can be set to 0, 1 or 2, 0 meaning no fading, 1 black fade effect and 2 white fade effect. This variable can either be set by a user (for each frame) or determined automatically during the video analysis stage using a fade detection technique [17]. As other TMOs do not perform a video analysis, they are unable to cope with video fade effects.

3.6. Parameters

Our solution relies on five parameters (ζ, δ, b_w, t and ρ). ζ is the same parameter as the one used in the BC algorithm and should be set between 0 and 1 (default value = 0.1). ζ prevents low scale ratio to trade off brightness coherency for a better preservation of details in dim frames. We found that setting it to 0.1 for every tested sequence provides good results, while it is set to 0 when a fade is detected.

Regarding δ, it is used to prevent abrupt changes of scale ratio on zone's boundaries. Too low a value results in such abrupt changes while a too high one increases the number of pixels that lie on a boundary. As mentioned in

subsection 3.4, we found that setting it to 1 provides good results for all the tested sequences.

The three last parameters (namely b_w, t, ρ) allow to determine the number of video zones. As all three parameters are highly correlated, it is hard to set each of them separately. That is why, we propose a way which facilitates the tuning of these parameters. We express the bin's width b_w so as to achieve the same quantization as for 8-bit LDR imagery (256 gray levels corresponding to 8 f-stops in dynamic range):

$$b_w = \theta \frac{8}{256} \tag{10}$$

where θ is a parameter used to increase or decrease the quantization. A high θ provides a coarse quantization thereby reducing the precision of the histogram. From b_w, we can derive the number of bins used for each segmentation as:

$$n_b = \frac{r}{b_w}, \tag{11}$$

where r is the dynamic range of the HDR video sequence and n_b the number of bins considered in the histogram.

Recall that a local maximum of the histogram must be higher than the threshold t. Consequently, t must be coherent with the number of bins:

$$t = \tau \frac{n_p}{n_b}, \tag{12}$$

where n_p is either the number of pixels in the image for the frame segmentation or the number of key values in the histogram for the video segmentation. If τ is lower than 1, the number of selected peaks increases otherwise it decreases. Experiments showed that a good setting for all the tested sequences is $\theta = 1$ and $\tau = 2$.

Finally, we express the distance between two peaks ρ directly in the log domain. More peaks are discarded with a high ρ, reducing then the number of segment/zones used. The number of video zones directly influences the locality of the ZBC technique. We found that a value ranging from 0.5 to 1.5 is a good compromise to tune the locality of our algorithm. Note that using only one video zone amounts to use the BC algorithm.

To sum up, we recommend to set $\delta = 1$, $\theta = 1$, $\tau = 2$ and tune only ρ for the locality (default value = 0.65) and ζ for the details/brightness coherency trade off (default value = 0.1).

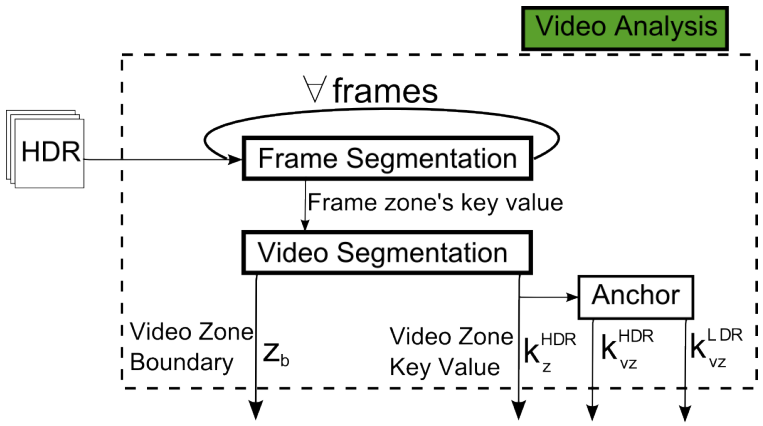


Figure 11: Details on the video analysis. The *Frame Segmentation* function segments each frame of the sequence and computes each segment's key value. The *Video Segmentation* determines the video zone's boundaries and their corresponding key values. The *Anchor* function determines the brightest zone in the HDR sequence k_v^{HDR} and computes its corresponding LDR key values k_v^{LDR}.

3.7. ZBC Summary

Similar to the BC method, the ZBC resorts to a video analysis prior to the tone mapping operation. This analysis needs a histogram-based segmentation as shown in figure 11. Performing the segmentation on a per frame basis results in several segments per frame. The key value of the j^{th} segment of each frame is noted $k_s^{j,HDR}$ and $k_s^{j,LDR}$ respectively for the HDR and LDR frames. Using the segments' key value histogram, a video segmentation is performed to provide zone boundaries. We then compute the key value $k_z^{i,j,HDR}$ of each zone j of each frame i.

To sum up, a key value can be associated with either a frame (k_f), a segment (k_s) or a zone (k_z). k_v^{HDR} and k_{vz}^{HDR} correspond respectively to the highest frame's key value and the highest video zone's key value of the HDR sequence. Note that the equivalent of these two variables in the LDR sequence, k_v^{LDR} and k_{vz}^{LDR}, do not always correspond to the highest key value of the LDR sequence but rather to the key value of the corresponding frame and zone respectively.

Once the video analysis has been performed, a scale ratio is applied to each video zone. More precisely, for each video zone j and each frame i, we compute an LDR zone key value ($k_z^{i,j,LDR}$) and use it to determine the

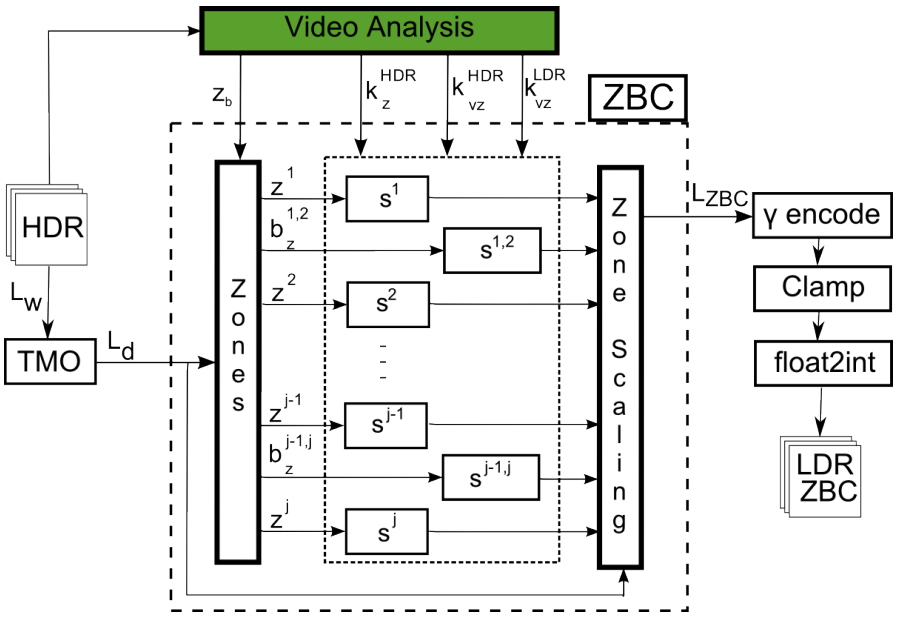


Figure 12: Complete ZBC workflow with details on the scaling phase. The *Zones* function determines, for each tone mapped frame and each pixel, the corresponding video zone z^j as well as the video blending zone $b_z^{j,j+1}$. Their respective scaling ratios s^j and $s^{j,j+1}$ are computed. The *Zone Scaling* function applies the scale ratios to the tone mapped frames.

associated scale ratio s^j (equation 5). We then compute the blending scale ratio $s^{j,j+1}$ (equation 8) for each pixel belonging to a blending zone b_z. The whole workflow of our method is given in figure 12.

Our ZBC algorithm performs better than the BC algorithm as will be seen in the next section.

4. Results

Let us recall that the BC algorithm suffers from two issues: lack of temporal local brightness coherency and loss of spatial contrast. The ZBC algorithm overcomes these issues, as shown by the obtained results presented in this section. Moreover, we evaluated our ZBC technique regarding the preservation of video fade effects.

As our technique is a post-processing, we can apply it to any TMO. In our experiments, we used 3 different TMOs: a flickering removal operator (Ram-

sey et al. [10], named $PTR + R$), a global operator (Exponential Mapping [18], called $ExpM$) and a local operator (Li et al. [19], termed SUB). For clarity, we add the BC/ZBC suffix to the TMO acronym (e.g. $PTR+R+BC$, $ExpM + ZBC$, $SUB + BC$, etc) when used.

We experimented with different HDR sequences. The first sequence is called $UnderBridgeHigh$ and is composed of 250 1920x1080 High Definition (HD) frames at the frequency of 25 frames per second (fps). We generated these frames using a 10 pixel vertical traveling inside a high resolution HDR image. Consequently, two successive frames are similar except the ten lower rows. Such a sequence is particularly helpful to test the temporal coherency as we have the same HDR values from frame to frame.

The second sequence is a real-world video captured using two cameras fixed on a rig. This sequence, called $Tunnel$, is composed of 2159 HD frames captured at the frequency of 25 fps. We only present results regarding the frames ranging from 300 to 700, showing a car entering a tunnel (noted 1 to 400 hereinafter).

The third sequence, called $GeekFlat$, represents a camera moving in a 3-room flat. Two well illuminated rooms are separated by a dim corridor. This sequence consists of 400 HD computer generated frames at the frequency of 25 fps. It illustrates changes of illumination conditions in the same cut.

Finally, we created two HDR sequences with a video fade effect. $MtStMichelBlack$ is a fade to black sequence while $MtStMichelWhite$ a fade to white. Both sequences consist of 100 HD frames at the frequency of 25 fps. The original image resolution is 3872x2592 and has been downsampled to get an HD resolution.

In section 4.1, we present results concerning the preservation of contrast and spatial brightness coherency. The temporal local brightness coherency is addressed in section 4.2. Finally, the last section shows some results regarding the preservation of video fade effects.

4.1. Spatial Contrast and Brightness Coherency

Let us recall that one of the goals of the ZBC algorithm is to preserve spatial brightness coherency and contrast. This property is valid for still images as well as video sequences. Figure 13 presents an HDR image tone mapped with several local TMOs with and without our algorithm. Local TMOs fail to preserve spatial brightness coherency (leftmost images fig. 13b, 13c and 13d): the room seems as bright as outdoors. On the contrary, the

ZBC preserves the indoor/outdoor contrast (rightmost images fig. 13b, 13c and 13d).

For video sequences, the BC algorithm scales each frame except the brightest one. As the ZBC uses as anchor the brightest zone rather than the brightest frame, all frames are scaled. Figure 14 illustrates such a case where the tone mapped frame is the brightest one of the $UnderBridgeHigh$ sequence (BC scale ratio is equal to 1). Note that no change is noticed when the BC is used (fig. 14c). However, when applying the ZBC, a higher contrast is achieved resulting in a frame with a sharper look.

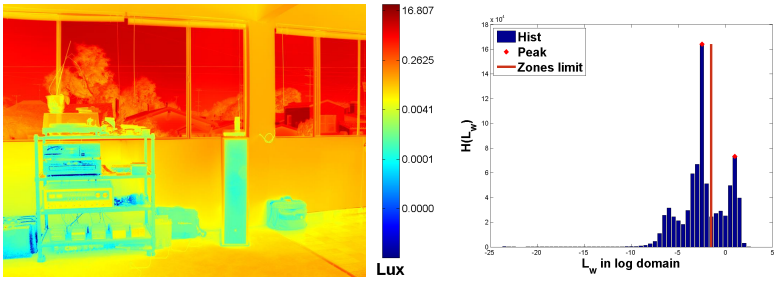
Another upside of the ZBC is that it preserves details that otherwise would be clamped by a TMO. Indeed, many TMOs choose to burn areas in order to associate a higher dynamic range with mid-tones areas. As the ZBC scales differently each zone of each frame, details ignored by a TMO can be recovered (fig. 15). Notice the lack of contrast through the window in figures 15b and 15c while it is preserved with our solution (fig. 15d). Another interesting feature of this image is the specular highlight located on the dragon. This highlight is present in each image, however only the ZBC preserves the original contrast of the HDR image (fig. 15a).

We showed that the ZBC preserves spatial brightness coherency and contrast even when tone mapping a still image. Concerning video sequences, the results presented here focused on each frame of a sequence individually. In the next subsection, we provide results regarding the temporal evolution of local brightness throughout a video.

4.2. Local Brightness Coherency

The second goal of the ZBC technique is to preserve temporal local brightness coherency using video zones. We assess the preservation of local brightness coherency by comparing the HDR and LDR evolutions of the video zones' key values throughout a video sequence.

Figure 16 plots such an evolution for the $UnderBridgeHigh$ sequence. Figure 16a represents the HDR video zones' key value ($k_Z^{j,HDR}$). Figure 16b plots the video zones' key value ($k_Z^{j,LDR}$) after applying the $PTR + R$ operator. Notice how the TMO fails to preserve the brightness ratios (spatial coherency) between each zone. Moreover, the key values of dim zones (blue and green curves) at the beginning of the sequence and the brightest ones at the end of the sequence have approximately the same value, which is not coherent. When using the BC technique (fig. 16c), this incoherency is reduced



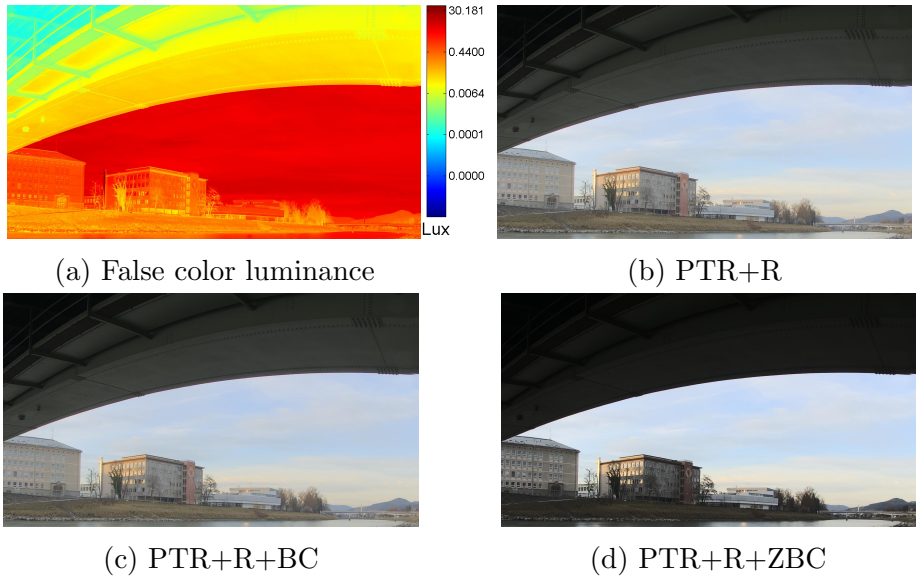
(a) False color luminance and segmented histogram

(b) $ICAM$, [20]

(c) Subband operator (SUB), [19]

(d) Local version of the PTR, [14]

Figure 13: Local TMOs without (left) and with the ZBC technique (right). The ZBC preserves the indoor/outdoor spatial contrast and coherency in the tone mapped results.



(a) False color luminance (b) PTR+R

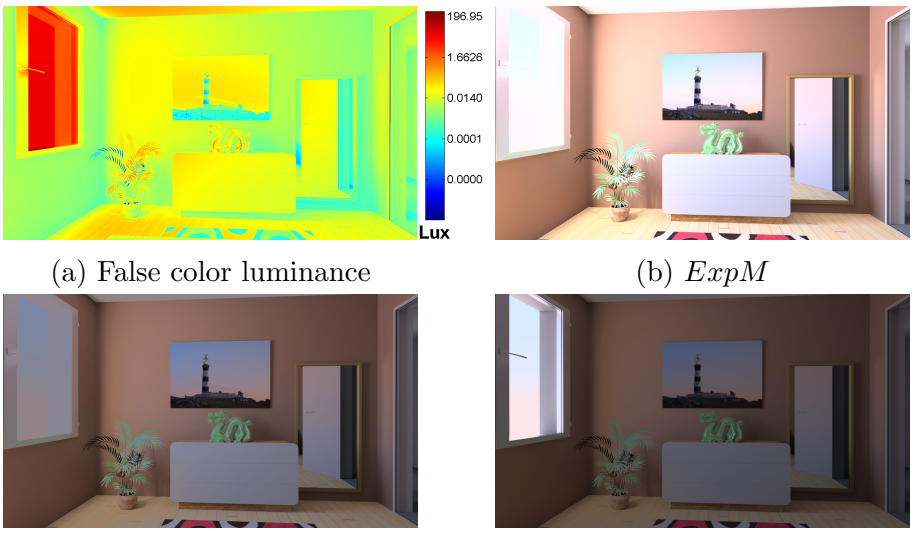
(c) PTR+R+BC (d) PTR+R+ZBC

Figure 14: Tone mapping of *UnderBridgeHigh* sequence, frame 160. Improvement brought by the ZBC when the BC has no effects. Note that to increase the contrast, some details are lost in the downside of the bridge.

thanks to the scaling, but the brightness ratio is still not preserved. With its zonal preservation system, the ZBC algorithm (fig. 16d) solves both issues.

Note that albeit the coherency is preserved between zones, it is not necessarily true inside the dimmest zones. This is due to the ζ parameter that prevents low scale ratio. For example, in the *UnderBridgeHigh* sequence, a low bound ζ is assigned to the scale ratio to prevent all dark frames under the bridge. This bounding explains the difference in brightness of the downside of the bridge between the beginning and the end of the sequence. In other words, the ζ parameter trades off brightness coherency to preserves detail in dimmer parts of a video (fig. 20).

Figures 17, 18, 19 and 20 show frames 80, 100, 120 and 140 of the *UnderBridgeHigh* sequence, tone mapped with different algorithms. In figure 18, the lower side of the bridge in frame 80 (that is a dim zone) is as bright as the sky in frame 140. With the BC technique (fig. 19), the bridge brightness appears more stable, however the sky's brightness is significantly reduced. Only the ZBC (fig. 20) preserves both aspects, providing frames with higher contrast and brightness coherency. However, differences are still noticeable in the brightness of the downside of the bridge, due to the use of



(a) False color luminance (b) $ExpM$

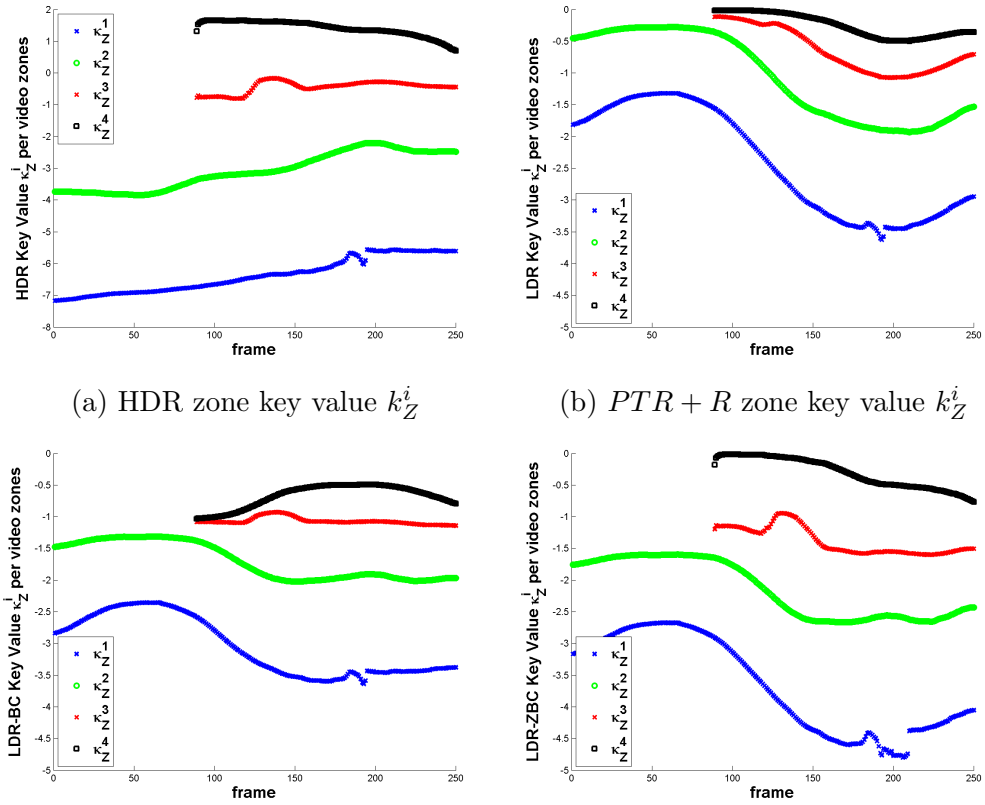
(c) $ExpM + BC$ (d) $ExpM + ZBC$

Figure 15: Tone mapping of the *GeekFlat* sequence, frame 26. When combining the Exponential TMO [18] and the ZBC algorithm, details lost in (15b) reappear in (15d).

ζ that prevents a too low scale ratio.

Results regarding the $Tunnel$ sequence are presented in figures 21 to 24. Notice how the $PTR + R$ algorithm (fig. 22) fails to reflect the variations of the overall brightness of the HDR sequence. In addition, the appearance of the outside of the tunnel is almost burnt out. By applying the BC technique (fig. 23), more overall brightness coherency is achieved, however the outside of the tunnel still lacks details. The ZBC technique (fig. 24) however, preserves both the coherency and the appearance of the outside of the tunnel.

The next subsection shows results on the preservation of video fade effects when applying a TMO to an HDR video sequence.



(a) HDR zone key value k_Z^i

(b) $PTR + R$ zone key value k_Z^i

(c) $PTR + R + BC$ zone key value k_Z^i

(d) $PTR + R + ZBC$ zone key value k_Z^i

Figure 16: Video zone key value (k_Z^j) for the HDR (16a), PTR+R (16b), PTR+R+BC (16c) and PTR+R+ZBC (16d). The HDR brightness ratio between the two higher curves (red and black) is only preserved when using the ZBC (16d). Concerning the two lower curves (blue and green), both the ZBC and the BC solve the temporal brightness coherency problem. The lack of coherency of each curve along the sequence is due to the ζ parameter that prevents low scale ratio.

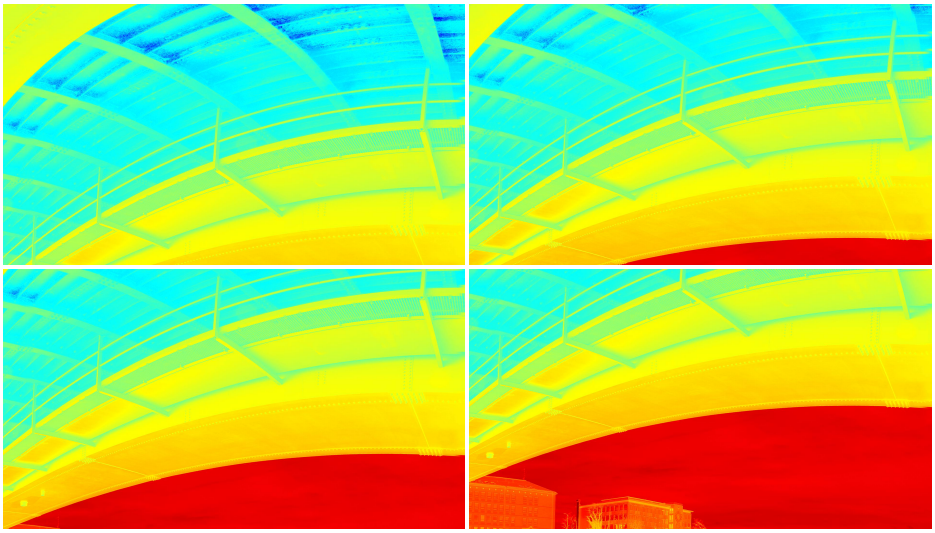


Figure 17: *UnderBridgeHigh* video sequence represented in false color luminance. Extracted frames 80 (Top-Left, TL), 100 (Top-Right, TR), 120 (Bottom-Left, BL) and 140 (Bottom-Right, BR).

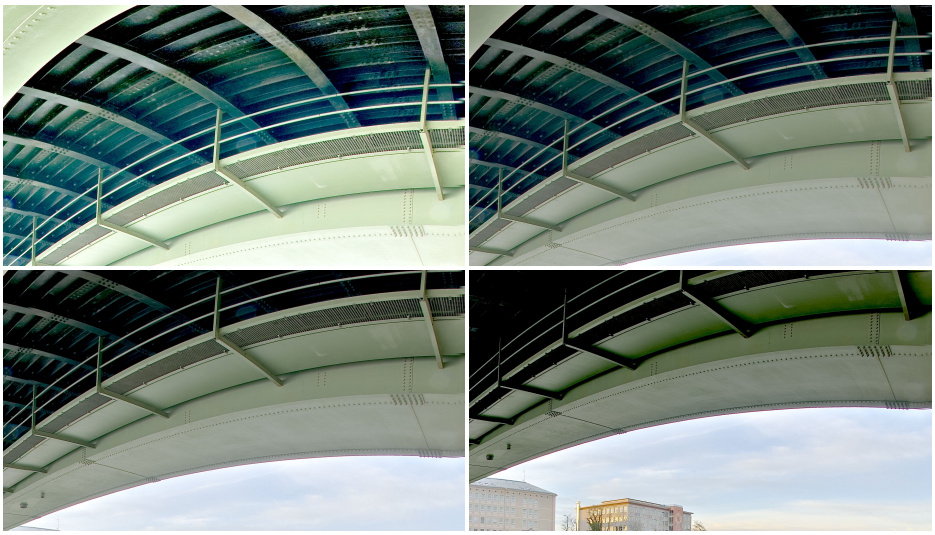


Figure 18: *UnderBridgeHigh* video sequence tone mapped with the *SUB* technique. Extracted frames 80 (TL), 100 (TR), 120 (BL) and 140 (BR).

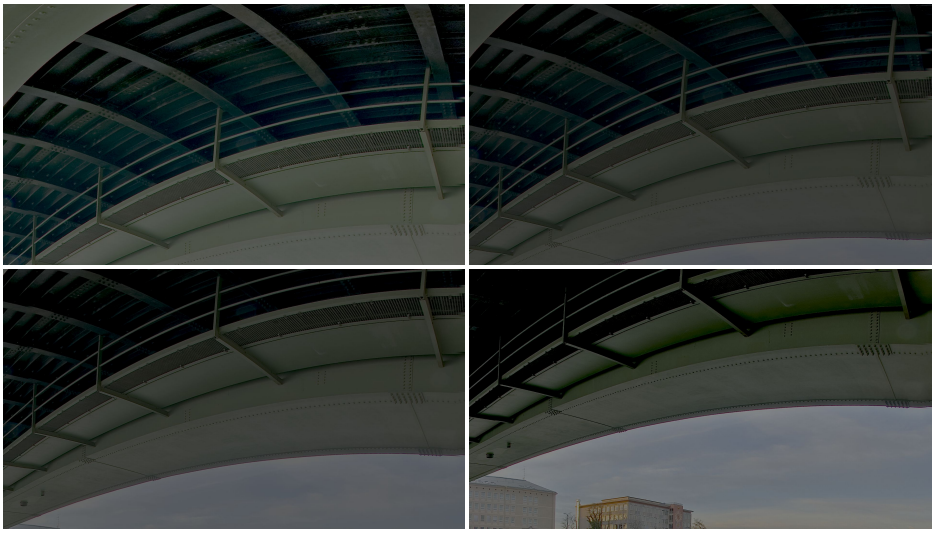


Figure 19: $UnderBridgeHigh$ video sequence tone mapped with the $SUB+BC$ technique. Extracted frames 80 (TL), 100 (TR), 120 (BL) and 140 (BR).

Figure 20: $UnderBridgeHigh$ video sequence tone mapped with the $SUB+ZBC$ technique. Extracted frames 80 (TL), 100 (TR), 120 (BL) and 140 (BR).

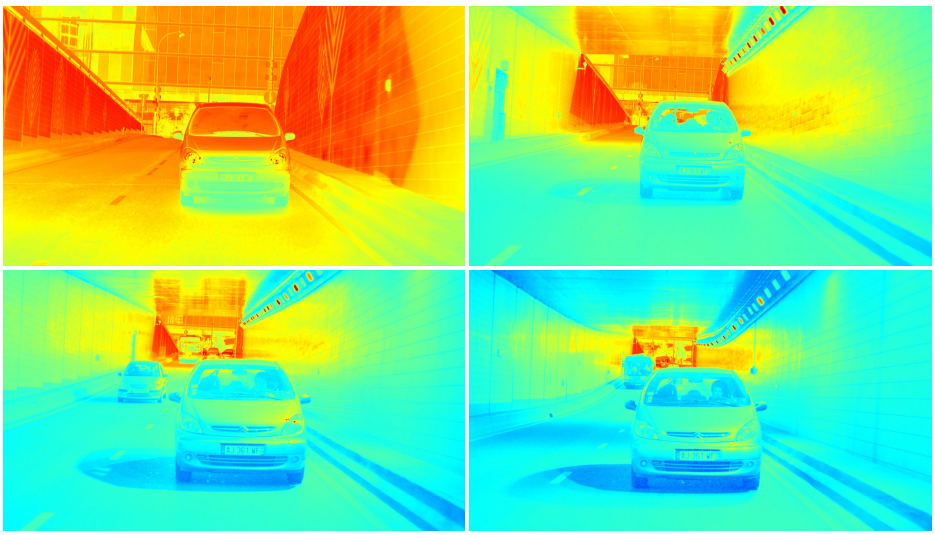


Figure 21: *Tunnel* video clip represented in false color luminance. Frames 1 (TL), 100 (TR), 230 (BL) and 397 (BR) extracted from the sequence.

Figure 22: *Tunnel* video clip after applying the $PTR + R$ TMO. Frames 1 (TL), 100 (TR), 200 (BL) and 300 (BR) extracted from the sequence.

Figure 23: *Tunnel* video clip after applying the $PTR{+}R{+}BC$ TMO. Frames 1 (TL), 100 (TR), 200 (BL) and 300 (BR) extracted from the sequence.

Figure 24: *Tunnel* video sequence after applying the $PTR + R + ZBC$ TMO. Frames 1 (TL), 100 (TR), 200 (BL) and 300 (BR) extracted from the sequence.

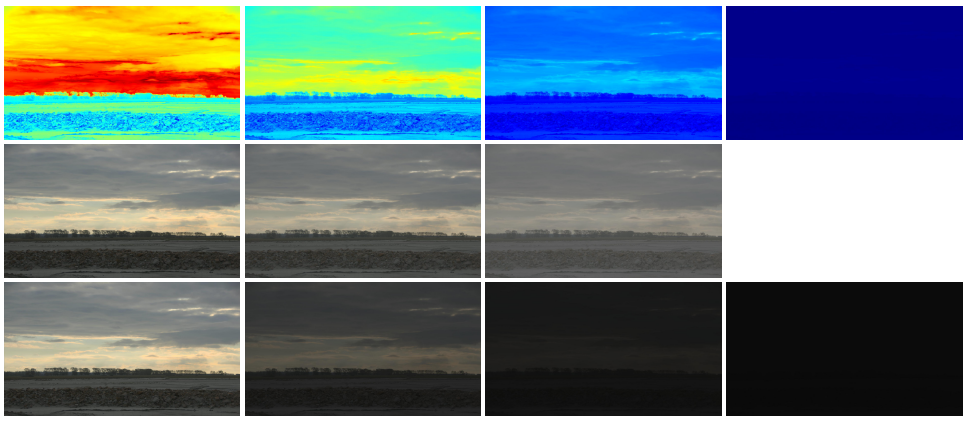


Figure 25: MtStMichelBlack video sequence represented in false color luminance (top-row) and after applying the $PTR+R$ operator (middle-row) and the $PTR+R+ZBC$ operator (bottom-row). Fade effect illustrated through frames 1, 33, 66 and 99 (from left to right).

4.3. Fading Effects

In this section, we show some results obtained when applying the ZBC algorithm on video fade sequences. As shown hereafter, TMOs do not preserve video fade effects as they do not analyze the video before performing the tone mapping.

Figure 25 illustrates the $MtStMichelBlack$ sequence in false color as well as after applying the $PTR+R$ TMO with and without the ZBC algorithm. Unlike the ZBC, the TMO alone fails to reproduce the fade to black effect.

To preserve a fade to white effect, the ZBC algorithm is modified as explained in section 3.5. Figure 26 shows the results of tone mapping a fade to white sequence. Similarly to the fade to black, only the ZBC preserves the video fade effect.

To validate our ZBC technique, we performed a subjective evaluation presented in the next section.

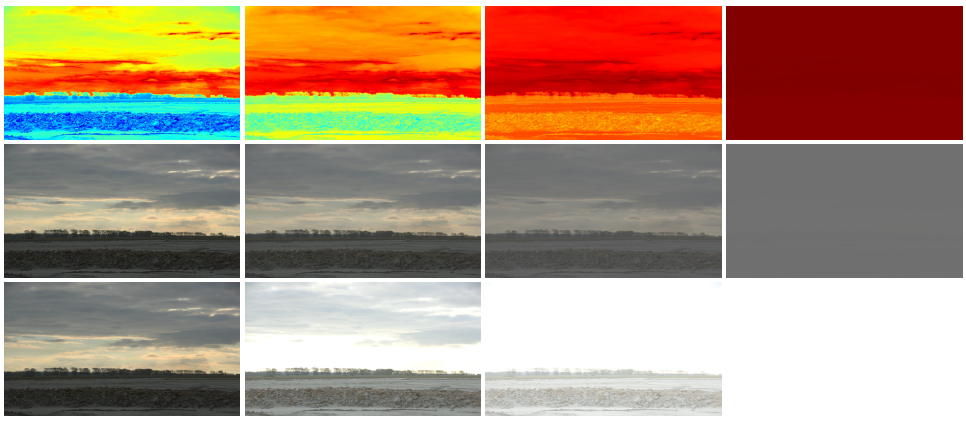


Figure 26: MtStMichelWhite video sequence represented in false color luminance (top-row) and after applying the $PTR+R$ operator (middle-row) and the $PTR+R+ZBC$ operator (bottom-row). Fade effect illustrated through frames 1, 33, 66 and 99 (from left to right).

5. Subjective Evaluation

Despite the large number of contributions addressing the evaluations of TMOs, no standard methodology yet exists. However, we distinguish three kinds of method to evaluate TMOs: the fidelity with reality, the fidelity with HDR reproduction and non-reference methods [21]. In the fidelity with reality method, an observer compares several tone mapped images with a real scene [22]. However, such a study is a tedious task when applied to images and unfeasible for videos.

For the fidelity with HDR reproduction, the reference is shown on an HDR display and an observer chooses the TMO that provides the closest reproduction to the HDR image [23, 24]. This evaluation is easier to set up and allows to test TMOs that preserve an artist intent (color grading, contrast enhancement, etc.)

Finally the non-reference method compares several tone mapped images that are ranked by an observer [25, 26]. This method is the easiest to set up and amounts to choosing the preferred result without knowing the HDR reference.

To validate the fact that preserving spatio-temporal brightness coherency improves the subjective quality, we conducted a subjective experiment using the non-reference method. For the comparison, we chose a forced-choice pairwise comparison as it provides the most accurate results as detailed in

[27]. The experiments were run in a dark room containing a 4K display device (TVlogic 56" Lum560W). The monitor was divided into 4 parts, the two HD videos to be compared were displayed on the upper-left and upper-right parts of the screen.

We experimented with three TMOs ($PTR + R$, $ExpM$ and SUB) and three HDR sequences ($UnderBridgeHigh$, $Tunnel$ and $GeekFlat$). We performed the test for three conditions: the TMO alone (TMO), the TMO with the Brightness Coherency (BC) and the TMO with the Zonal Brightness Coherency (ZBC). Each combination of two conditions was tested twice (inverting the left-right order of the showed videos). A choice is considered as coherent if the same video is chosen for both left-right orders. The experiment consisted of 54 comparisons (3 TMOs $\cdot$ 3 Sequence $\cdot$ 3 Conditions $\cdot$ 2 Combinations). 18 people (aged from 23 to 59) participated in the experiment, among them 14 are computer scientists without knowledge regarding HDR and tone mapping while the last four had prior experience with TMOs.

The results were fitted in a Bradley-Terry model using Branderson's equation [28]. The confidence intervals were computed using a non-coherent choice as a tie, while coherent choices were reported as win/lose results. Figure 27 plots the quality scores of each technique as well as their confidence intervals for each tested video. These results show that for a given TMO, applying the BC post-processing does not improve the subjective quality. On the contrary, when comparing the ZBC with either the TMO+BC or the TMO alone, the resulting video is preferred for the $UnderBridgeHigh$ and $Tunnel$ sequence while no conclusion can be reached for the $GeekFlat$ sequence. This is mostly due to the fact that only two video zones were determined using the default parameters in the case of the $GeekFlat$ sequence.

This subjective evaluation confirmed the improvement brought by our ZBC method. A compressed version of the videos used for the subjective evaluation are available at `http://people.irisa.fr/Ronan.Boitard/shared/ZBC_Videos.zip`.

6. Conclusion

This paper presented a method to adapt Tone Mapping Operators to video sequences. The main goal of this method was to preserve both spatial and temporal local brightness coherencies. The algorithm relies on two main operations: video analysis and scaling.

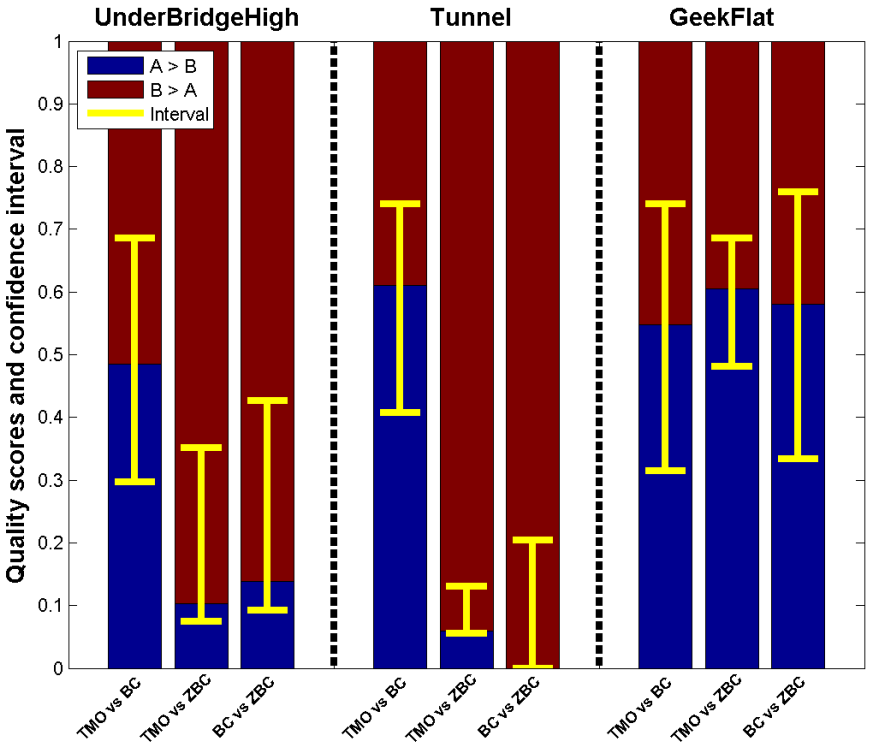


Figure 27: Results of the subjective evaluation per video. The quality score and confidence interval were computed using [28]. When the confidence interval (yellow segment) includes the quality score 0.5, no conclusions can be drawn.

The analysis of a video sequence consists of two operations which are performed successively. First, we proposed a histogram-based segmentation algorithm for HDR images. It subdivides an image into different segments. When applied to a video sequence, the number of segments and their corresponding boundaries change from frame to frame. That is why a second operation determines video zones from the key value of each frame's segment.

A scaling function computes a scale ratio for each video zone. To avoid discontinuities at the zone's boundaries, the scale ratios, associated with pixels on a boundary, are blended.

Our ZBC method preserves the spatial brightness coherency in images as well as in video sequences. Moreover, it preserves the temporal local brightness coherency. The resulting video sequences look sharper and perceptually closer to the HDR original sequences. We validated our technique with a subjective evaluation for three HDR video sequences. These results show that the ZBC's quality is preferred for two out of the three videos while no real conclusions could be made for the third one. Finally, video fading effects are also preserved.

Albeit the ZBC is efficient to solve the temporal brightness coherency, it needs a preprocessing that consists in performing a video analysis. For broadcast applications, since the whole video is not available, this analysis cannot be performed. One solution would be to perform the analysis within the latency time range of the broadcast. However, this solution would not be able to preserve long time range brightness coherency.

Furthermore, our solution relies on the assumption that the brightness coherency should be preserved linearly during the tone mapping. Future work should explore non-linear brightness coherency using psychophysical studies.

Another limitation of the ZBC is that it does not take into account the human visual system. For example, temporal adaptation could be used to adjust the ZBC scaling function. Finally, another subjective evaluation should be conducted using the fidelity with HDR reproduction method.

7. Acknowledgment

The *Tunnel* sequence was produced by Binocle and Technicolor within the framework of the french collaborative project NEVEx. We also thank the anonymous reviewers for their helpful comments.

References

[1] E. Reinhard, T. Kunkel, Y. Marion, J. Brouillat, R. Cozot, K. Bouatouch, Image display algorithms for high- and low-dynamic-range display devices, Journal of the Society for Information Display 15 (12) (2007) 997.

[2] F. Banterle, A. Artusi, K. Debattista, A. Chalmers, Advanced High Dynamic Range Imaging: Theory and Practice, AK Peters (CRC Press), Natick, MA, USA, 2011.

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

[4] M. D. Tocci, C. Kiser, N. Tocci, P. Sen, A versatile hdr video production system, ACM Trans. Graph. 30 (4) (2011) 41:1–41:10.

[5] J. Kronander, S. Gustavson, G. Bonnet, J. Unger, Unified HDR reconstruction from raw CFA data, IEEE International Conference on Computational Photography (ICCP) (2013) 1–9.

[6] B. Guthier, S. Kopf, M. Eble, W. Effelsberg, Flicker reduction in tone mapped high dynamic range video, in: Proc. of IS&T/SPIE Electronic Imaging (EI) on Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications, 2011, pp. 78660C–78660C–15.

[7] S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High dynamic range video, ACM Trans. Graph. 22 (3) (2003) 319–325.

[8] C. Lee, C.-S. Kim, Gradient domain tone mapping of high dynamic range videos, in: Image Processing, 2007. ICIP 2007. IEEE International Conference on, Vol. 3, 2007, pp. III – 461–III – 464.

[9] R. Mantiuk, S. Daly, L. Kerofsky, Display adaptive tone mapping, ACM Transactions on Graphics 27 (3) (2008) 1.

[10] S. Ramsey, J. J. III, C. Hansen, Adaptive temporal tone mapping, Computer Graphics and Imaging - 2004 (3) (2004) 3–7.

[11] C. Kiser, E. Reinhard, M. Tocci, N. Tocci, Real-time Automated Tone Mapping System for HDR Video, IEEE International Conference on Image Processing (2012) 2749–2752.

[12] R. Boitard, D. Thoreau, K. Bouatouch, R. Cozot, Temporal Coherency in Video Tone Mapping , a Survey, in: HDRi2013 - First International Conference and SME Workshop on HDR imaging (2013),, no. 1, 2013, pp. 1–6.

[13] R. Boitard, K. Bouatouch, R. Cozot, D. Thoreau, A. Gruson, Temporal coherency for video tone mapping, in: SPIE Conference Series, Vol. 8499 of SPIE Conference Series, 2012.

[14] E. Reinhard, M. Stark, P. Shirley, J. Ferwerda, Photographic tone reproduction for digital images, ACM Transactions on Graphics 21 (3).

[15] A. Adams, The Print: The Ansel Adams Photography Series 3, Little, Brown and Compagny, 1981.

[16] Z. Farbman, D. Lischinski, Tonal stabilization of video, ACM Trans. Graph. 30 (4) (2011) 89:1–89:10.

[17] A. Alattar, Detecting fade regions in uncompressed video sequences, in: Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, Vol. 4, 1997, pp. 3025–3028 vol.4.

[18] F. Banterle, Exponential TMO available in the HDR toolbox.
URL `http://www.banterle.com/hdrbook/downloads.php`

[19] Y. Li, L. Sharan, E. H. Adelson, Compressing and companding high dynamic range images with subband architectures, ACM Trans. Graph. 24 (3) (2005) 836–844.

[20] J. Kuang, G. M. Johnson, M. D. Fairchild, iCAM06: A refined image appearance model for HDR image rendering, Journal of Visual Communication and Image Representation 18 (5) (2007) 406–414.

[21] G. Eilertsen, R. Wanat, R. K. Mantiuk, J. Unger, Evaluation of Tone Mapping Operators for HDR-Video, Computer Graphics Forum (Proc. of Pacific Graphics) 32 (7).

[22] A. Yoshida, Perceptual evaluation of tone mapping operators with real-world scenes, in: Proceedings of SPIE, Vol. 5666, SPIE, 2005, pp. 192–203.

[23] J. Kuang, R. Heckaman, M. Fairchild, Evaluation of HDR tone-mapping algorithms using a high-dynamic-range display to emulate real scenes, Journal of the Society for Information Display 18 (2010) 461.

[24] P. Ledda, A. Chalmers, T. Troscianko, H. Seetzen, Evaluation of tone mapping operators using a high dynamic range display, ACM Trans. Graph. 24 (3) (2005) 640–648.

[25] J. Kuang, H. Yamaguchi, C. Liu, G. M. Johnson, M. D. Fairchild, Evaluating HDR rendering algorithms, ACM Transactions on Applied Perception 4 (2) (2007) 9–es.

[26] M. Čadík, M. Wimmer, L. Neumann, A. Artusi, Evaluation of HDR tone mapping methods using essential perceptual attributes, Computers & Graphics 32 (3) (2008) 330–349.

[27] R. K. Mantiuk, A. Tomaszewska, R. Mantiuk, Comparison of Four Subjective Methods for Image Quality Assessment, Computer Graphics Forum 31 (8) (2012) 2478–2491.

[28] J.-S. Lee, F. De Simone, T. Ebrahimi, Subjective quality evaluation via paired comparison: Application to scalable video coding, IEEE Transactions on Multimedia 13 (5) (2011) 882–893.