

# Real-time noise-aware tone mapping

Gabriel Eilertsen  
Linköping University, Sweden

Rafał K. Mantiuk  
Bangor University, UK &  
The Computer Laboratory,  
University of Cambridge, UK

Jonas Unger  
Linköping University, Sweden



**Figure 1:** A comparison between a naïve tone mapping which leads to an amplification of image noise, and our noise-aware approach. The example shows that local contrast, details, and the overall appearance are preserved. Video frame from [Froehlich et al. 2014].

## Abstract

Real-time high quality video tone mapping is needed for many applications, such as digital viewfinders in cameras, display algorithms which adapt to ambient light, in-camera processing, rendering engines for video games and video post-processing. We propose a viable solution for these applications by designing a video tone-mapping operator that controls the visibility of the noise, adapts to display and viewing environment, minimizes contrast distortions, preserves or enhances image details, and can be run in real-time on an incoming sequence without any preprocessing. To our knowledge, no existing solution offers all these features. Our novel contributions are: a fast procedure for computing local display-adaptive tone-curves which minimize contrast distortions, a fast method for detail enhancement free from ringing artifacts, and an integrated video tone-mapping solution combining all the above features.

**CR Categories:** I.3.3 [Computer Graphics]: Picture/Image Generation—Display Algorithms; I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering

**Keywords:** tone mapping, video tone mapping, high dynamic range imaging

## 1 Introduction

High dynamic range (HDR) video will offer unprecedented improvements in viewing experiences for high end cinemas as well

as consumer level products. Driven by the demands for extended visual fidelity and artistic freedom, HDR technology is currently moving forward very rapidly. On the capturing side, we see the development of both professional HDR-camera systems such as the Arri Alexa XT and the Red Epic Dragon with an extended dynamic range of up to 14 - 16.5 *f-stops*, as well as research prototypes [Tocci et al. 2011; Kronander et al. 2013] exhibiting a dynamic range of up to 20 - 24 *f-stops*. On the production side, major studios are meeting this ongoing trend by developing fully HDR-enabled production pipelines, putting a completely new creative toolset in the hands of the artists. Also on the display side, HDR technology is in strong focus. Manufacturers, e.g. Sim2, have moved towards extending the dynamic range using high contrast local dimming techniques and Dolby Vision X-tended Dynamic Range PRO has recently been announced. Although significant efforts are being spent at each step in the HDR-video pipeline, from capture and processing to compression and display, one important challenge still requires substantial improvement: *tone mapping for HDR-video*.

Despite the need for robust video tone mapping, the existing algorithms often fall short of expectations as they tend to reveal or amplify noise, cannot handle large contrast compression, introduce ringing, ghosting or temporal flicker [Eilertsen et al. 2013], do not adapt to the display and viewing conditions, or are slow to compute.

We propose a new tone mapping operator (TMO) for video, which controls the visibility of the noise, adapts to the display and viewing environment, minimizes contrast distortions, preserves or enhances image details, and can be run in real-time on an incoming sequence without any preprocessing. The intent of our tone mapping is either to preserve the contrast in the original image given the constraints of a target display, or to provide creative controls for contrast enhancement, with a real-time full-resolution visual feedback of the final result. We do not attempt to match the appearance or visibility in scene-referred HDR content, for example for driving simulators, although such models can be combined with our approach.

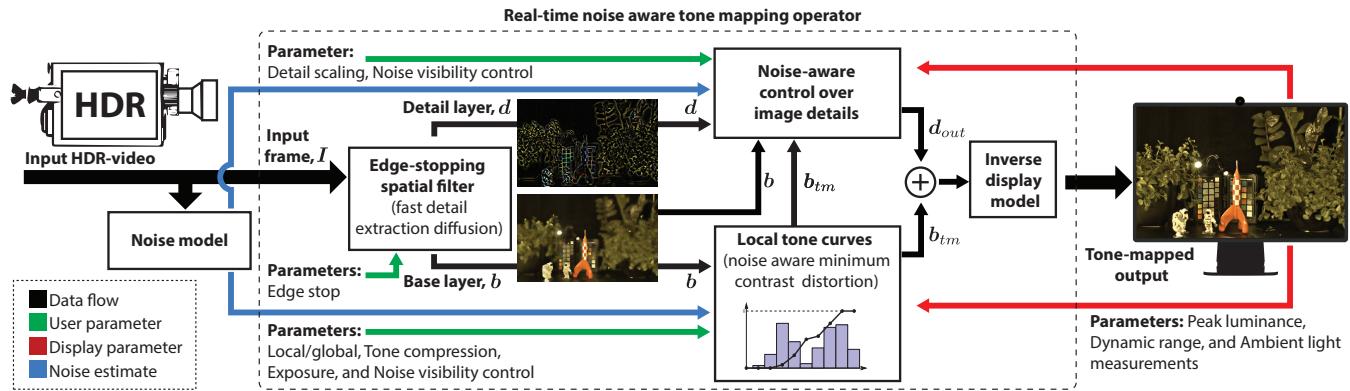
Our main technical contributions are:

- A fast procedure for computing local display-adaptive tone-curves that minimize contrast distortion.
- Noise-aware tone curves adaptive to image noise.

**ACM Reference Format**  
Eilertsen, G., Mantiuk, R., Unger, J. 2015. Real-time noise-aware tone mapping. ACM Trans. Graph. 34, 6, Article 198 (November 2015), 15 pages. DOI = 10.1145/2816795.2818092.  
<http://doi.acm.org/10.1145/2816795.2818092>.

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

SIGGRAPH Asia '15 Technical Paper, November 02 – 05, 2015, Kobe, Japan.  
Copyright 2015 ACM 978-1-4503-3931-5/15/11 ... \$15.00.  
DOI: <http://doi.acm.org/10.1145/2816795.2818092>



**Figure 2:** Diagram of the proposed video tone-mapping. The input is split into a base and a detail layer, where the base layer is compressed using locally adaptive tone curves. The layers are then combined and converted into pixel values (inverse display model) before display.

- Filter design for base – detail layer decomposition for tone mapping, leading to a fast edge-stopping non-linear diffusion approximation for detail enhancement without ringing artifacts.
- An integrated, *real-time* video tone-mapping solution, which combines the novel noise control with display-adaptivity to produce high contrast and detailed video given the display limitations.

To demonstrate the usefulness of our approach, we evaluate our TMO using a range of challenging input HDR-video sequences comparing our algorithm to existing state-of-the-art. We also show how our noise-aware TMO can be used both with unprocessed noisy input data, and to improve the visual quality of input data processed using state-of-the-art denoising algorithms. Finally, we also show the capabilities of the display adaptive features, as well as the artistic freedom provided by the detail extraction technique.

## 2 Related work

Eilertsen *et al.* [2013] evaluated and analyzed 11 video tone-mapping operators. They categorized them into those that simulate the properties and limitations of the visual system — visual system simulators (VSS), those that attempt to preserve the original scene appearance — scene reproduction operators (SRP), and those that produce subjectively preferred images — best subjective quality operators (BSQ). Our solution fulfills the intents of SRP operators when used with default parameters, but also offers control over contrast enhancement to reach the goals of BSQ operators. Eilertsen *et al.* concluded that all tested operators were prone to introduce artifacts such as flickering, ghosting, amplified level of noise, or lack of details, which we address in the proposed solution.

Temporal artifacts, such as flickering, are a significant problem for many video TMOs. For temporal stability, global operators often rely on filtering over time of the tone curve [Mantiuk *et al.* 2008], or the TMO parameters [Pattanaik *et al.* 2000; Kiser *et al.* 2012]. While this allows for efficient implementation, the situation is more complicated for local TMOs, where the tone reproduction can change incoherently over time on a local level. To overcome such problems, and to reduce noise, many local operators employ spatio-temporal filters in the pixel domain, [Ledda *et al.* 2004; Bennett and McMillan 2005; Van Hateren 2006], or along motion paths, [Aydin *et al.* 2014]. However, these filters are usually expensive to compute and do not lend themselves well to real-time processing which is a key goal for our TMO. Another problem is that that they are prone to introduce ghosting artifacts or may not work well where the optical flow fails. In contrast to these meth-

ods we rely on a new spatial filter specifically designed for robust base–detail layer decomposition, so that spatial changes in the tone reproduction are coherent over time.

The proposed spatial filter is comparable to many existing edge-preserving filters [Perona and Malik 1990; Aurich and Weule 1995; Tomasi and Manduchi 1998; Aubry *et al.* 2014]. In contrast to those, however, our filter is designed for the best extraction of details in a tone-mapping scenario, rather than edge-preserving smoothing or denoising. It also yields a simple implementation, which is well suited for parallel hardware architectures.

Throughout the paper, we use HDR-video input from the publicly available database from [Froehlich *et al.* 2014]. This dataset is suitable for evaluation of tone mapping operators, as it contains high quality footage captured in realistic scenarios

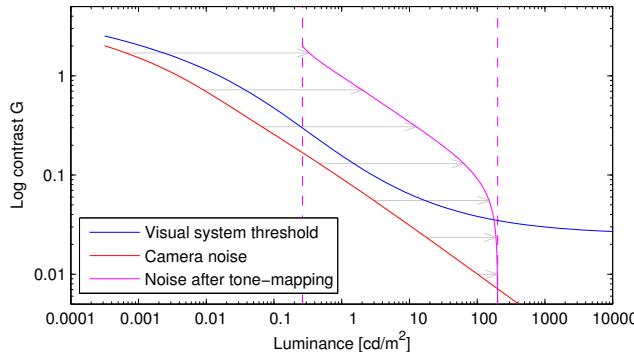
## 3 Algorithm design and overview

Our TMO, outlined in Figure 2, is built around three fundamental requirements: noise awareness, temporal robustness, and display adaptivity. Two basic building blocks are the noise model described in Section 3.1, and the inverse display model described in Section 3.2. As indicated in the figure, our algorithm can be divided into three main parts:

*Edge-stopping spatial filter* - Each input HDR frame,  $I$ , is first transformed to the log domain,  $l = \log(I)$ , and decomposed into a base layer,  $b$ , describing luminance variations over the image, and a detail layer,  $d$ , describing local features. Section 5 presents a discussion on filter design for robust base - detail layer decomposition.

*Local tone curves* - The dynamic range of the base layer  $b$  is then compressed using a set of local tone-curves distributed spatially over the image. Each tone-curve maps from the luminance range of the input into the range of luminance achievable on a display, as explained by the *display model* discussed in Section 3.2. The tone-curves minimize contrast distortions introduced by the mapping while controlling the visibility of noise, and their shapes are filtered over time to avoid temporal flickering. The noise-aware minimum contrast distortion tone mapping is discussed in Section 4.

*Noise-aware control over image details* - Our TMO allows the user to preserve (SRP) or artistically enhance local contrast and details (BSQ) by scaling the detail layer,  $d$ . The visibility of noise in the processed detail layer,  $d_{out}$ , is controlled by taking into account the noise characteristics of the original base layer  $b$  and the tone-mapped base layer  $b_{tm}$ . This is described in detail in Section 6.



**Figure 3:** Many high-end cameras offer noise levels below the perceivable contrast of the visual system (red line below the blue line). However, after tone-mapping, when the image is made much brighter and details and contrast are enhanced, the noise is often pushed into the visible range (the magenta line). Therefore, controlling the visibility of noise is the critical component of a high quality tone-mapping. The dashed-magenta lines denote the dynamic range of a target display.

The tone-mapped base layer,  $b_{tm}$ , and the processed detail layer,  $d_{out}$ , are then combined, color is restored from input using standard techniques [Mantiuk et al. 2009] and the colorimetric values are transformed into pixels using an inverse display model.

### 3.1 Noise in tone mapping

The visible noise in a video sequence can be greatly reduced using modern denoising algorithms [Maggioni et al. 2012; Aydin et al. 2014]. However, too strong denoising introduces blur and reduces image sharpness. Since the lack of sharpness is a less tolerable artifact than noise, the common video processing practice is to employ conservative noise reduction and then conceal the remaining noise in a *manual* color grading step. Here, we propose how the latter step can be automated by noise-aware tone-mapping. The input to our method is either a (conservatively) denoised or a noisy video sequence. The result is a video, in which visibility of noise is reduced by considering both the noise properties and its visibility on a particular display device.

Figure 3 illustrates the problem. High-end cameras offer large apertures and large sensors, which often yield lower noise levels than those of the visual system. This is shown in Figure 3 as the red line of the camera noise being below the blue line of the HVS thresholds. Yet, when an image is tone-mapped for a display, the absolute luminance levels are increased and fine contrast details are enhanced, resulting in an image with well visible noise (magenta line). The goal of our approach is to keep the noise level below the visibility thresholds. Our method can be used in combination with existing denoising methods, so that a denoising filter removes the most offending, high amplitude noise and the low-amplitude noise is hidden using our noise-aware processing.

We control the noise visibility at two points: when computing the tone-curves (Section 4.1), and when preserving local contrast and details (Section 6). To control the noise, we need to model both the magnitude of the noise and the visibility thresholds. The variance of the noise in a digital camera can be modelled as the function of light intensity  $I$  [Foi et al. 2008]:

$$\sigma_n^2(I) = aI + b, \quad (1)$$

where  $a$  and  $b$  are the parameters responsible for signal dependent (photon noise) and signal independent (read-out noise) components

of the noise. The parameters can be estimated from the input image [Foi et al. 2008], be provided by the camera, or manually adjusted.

To account for the non-linear sensitivity of the visual system to light (Weber-Fechner law), we will operate in the logarithmic domain. The noise magnitude in the logarithmic domain can be approximated with:

$$n(I) = \log_{10} \left( \frac{I + \sigma_n}{I} \right). \quad (2)$$

Such a noise magnitude is plotted as a red line in Figure 3. A conservative estimate of the visibility thresholds, shown as the blue line in Figure 3, is given by the peaks of the contrast sensitivity function (CSF) for a particular display luminance level  $L_d$ . We use the CSF from [Mantiuk et al. 2011] and convert the detection threshold ( $C_t(L_d) = 1/CSF(L_d)$ ) from Michelson contrast to logarithmic contrast to get the smallest detectable difference in log-luminance:

$$V(L_d) = 0.5 \log_{10} \left( \frac{C_t(L_d) + 1}{1 - C_t(L_d)} \right). \quad (3)$$

### 3.2 Display adaptivity

The main constraint of any tone-mapping operator is the available range of luminance that can be shown on a target display device. Such a range depends not only on the particular display technology, such as OLED or LCD, but also on the ambient light levels. A portion of ambient light is reflected from a display screen and thus reduces the available display contrast. Our operator was designed to adjust for that and produce the best image given current ambient light levels, which can be readily measured with a light sensor.

The available range of luminance can be modeled using the standard Gamma-Gain-Offset display model [Berns 1996] with the modification for ambient light [Mantiuk et al. 2008]:

$$L_d(L') = (L')^\gamma \cdot (L_{max} - L_{black}) + L_{black} + L_{refl} \quad (4)$$

where  $L_d$  is displayed luminance or radiance (as measured coming from the display surface),  $L'$  is the pixel value (0–1),  $\gamma$  is a display gamma (usually close to 2.2),  $L_{max}$  is the peak display luminance (about 200 cd/m<sup>2</sup> for office displays).  $L_{black}$  is the display black level, which is the luminance of the black pixel displayed in a perfectly dark room (usually from 0.1 to 0.8 cd/m<sup>2</sup> for LCD displays).  $L_{refl}$  is the ambient light reflected from the display surface. For non-glossy screens this can be approximated as:

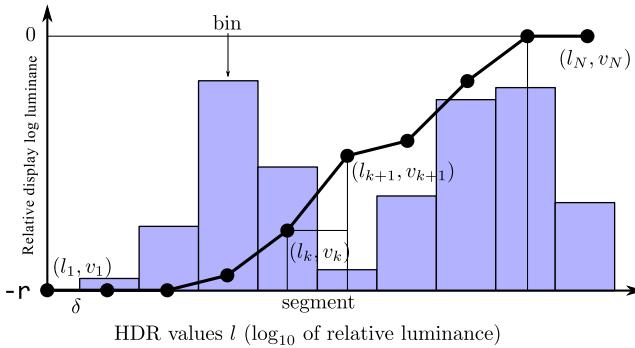
$$L_{refl} = \frac{k}{\pi} E_{amb} \quad (5)$$

where  $E_{amb}$  is ambient illuminance given in lux units and  $k$  is the reflectivity for a display panel (0.5–1% for LCD displays).

## 4 Minimum contrast distortion tone-curve

A tone-curve, which maps pixel values from input luminance to the luminance of the display, is the primary tool for reducing image contrast to the displayable range. The tone-curves used traditionally in photography have an S-shape, which preserves contrast in middle tones at the cost of higher distortions at low and high tones. If a fixed shape of a tone-curve was a necessity for analog film, digital processing allows variations of the tone-curve for each image, and even image region. This section shows a derivation of such a local tone-curve, which minimizes contrast distortions and is fast to compute.

To keep the problem analytically tractable, we parameterize the tone-curve as a piece-wise linear and non-decreasing function with



**Figure 4:** Tone-curve is parameterized as a piecewise linear function from input log-luminance to the displayed log-luminance.

the nodes  $(l_k, v_k)$ , as shown in Figure 4.  $l$  is the logarithmic luminance in the input, and  $v$  is the logarithmic luminance of the displayed values. Each segment  $k$  between two nodes  $(l_k, v_k)$  and  $(l_{k+1}, v_{k+1})$  has a constant width in log-luminance values equal to  $\delta$  (0.2 in our implementation). For simplicity, the maximum log-luminance value that can be shown on a display is fixed at 0 and the minimum varies with the effective display dynamic range  $r$ .  $r$  can be computed for the current ambient light level using the display model from Equation 4 as

$$r = \log_{10} \left( \frac{L_d(1)}{L_d(0)} \right). \quad (6)$$

**Expected contrast distortion** Arguably, the most relevant distortion due to tone-mapping is the change of image contrast, which we want to minimize. Let us denote input image contrast with the symbol  $G$  and the contrast after tone-mapping with the symbol  $\tilde{G}$ . In the simplest case, contrast  $G$  could be a difference between two neighboring pixels (we will show that the derivation generalizes to a range of contrast definitions). With each input log-luminance level  $l$  we can associate a certain distribution of contrast values  $p(G|l)$ . Given that, we can express the expected value of the contrast distortion due to tone-mapping as:

$$E[||G - \tilde{G}||_2^2] = \int p(l) \int (G - \tilde{G})^2 p(G|l) dG dl. \quad (7)$$

The inner integral “sums” the squared distortions,  $(G - \tilde{G})^2$ , over all possible contrast values ( $G$ ) at a certain input log-luminance level  $l$ . The outer integral “sums” that result over all input log-luminance levels ( $l$ ).  $p(l)$  is the probability that a certain contrast  $G$  is shown on a background luminance  $l$ .  $p(G|l)$  is the probability of finding in an image contrast  $G$  given the background luminance  $l$ . For example, if the contrast  $G$  is defined as a gradient, this probability will follow a heavy-tailed distribution. For natural images, the contrast distribution is, in a general case, independent of the local background luminance  $l$ , and therefore we can assume that  $p(G|l) = p(G)$ . If we discretize the above equation using the piece-wise linear tone-curve, we can represent the distorted contrast for segment  $k$  as:

$$\tilde{G} = s_k G, \quad (8)$$

where  $s_k$  is the slope of a tone curve in the segment  $k$ :

$$s_k = \frac{v_{k+1} - v_k}{\delta}. \quad (9)$$

Therefore, the discrete approximation of the expected distortion can be expressed as:

$$\begin{aligned} E[||G - \tilde{G}||_2^2] &\approx \varepsilon(s_1, \dots, s_N) = \sum_k p(l_k) \sum_G (G - s_k G)^2 p(G) \\ &= \sum_k p(l_k) (1 - s_k)^2 \sum_G G^2 p(G). \end{aligned} \quad (10)$$

Note that the term  $(1 - s_k)$  is independent of the contrast  $G$  and thus can be moved outside the sum. The values  $p(l_k)$  can be computed as a histogram of image log-luminance values with the centers of the bins at  $l_k$  for  $k = 1..N$ .

**Minimum contrast distortion** Our goal is to minimize the expected contrast distortion due to a tone-curve from Equation 10:

$$\arg \min_{s_1 \dots s_N} \varepsilon(s_1, \dots, s_N) \quad (11)$$

subject to:

$$\begin{aligned} s_k &\geq 0 \quad \text{for } k = 1..N \\ \sum_{k=1}^N s_k \cdot \delta &\leq r, \quad \text{where } r = v_N - v_1. \end{aligned} \quad (12)$$

The first constraint ensures that the tone-curve is non-decreasing, and the second that the maximum available dynamic range on the display (Equation 6) is not exceeded.

Note that the sum over  $G$  in Equation 10 is independent of the tone-curve slope  $s_k$ . Therefore, when minimizing  $\varepsilon(s_1, \dots, s_N)$  as a function of a tone-curve given by  $s_{1..N}$ , the contrast distribution  $p(G)$  has no impact on the minimum. The problem can thus be simplified to minimizing the functional

$$\varepsilon'(s_k) = \sum_k p(l_k) (1 - s_k)^2, \quad (13)$$

subject to the conditions given in Equation 12. This problem can be solved analytically by calculating the first order Karush-Kuhn-Tucker (KKT) optimality conditions of the corresponding Lagrangian. This gives the solution (refer to Appendix A for the derivation):

$$s_k = 1 + \frac{\frac{r}{\delta} - N}{p(l_k) \sum_{i=1..N} \frac{1}{p(l_i)}} \quad (14)$$

The above solution may result in negative slopes and thus violate the first constraint. We therefore set to 0 the slopes for those segments for which the probability  $p(l_k)$  is less than a certain threshold. From Equation 14 we see that  $s_k \geq 0$  when:

$$p(l_k) \geq \frac{N - \frac{r}{\delta}}{\sum_{i=1..N} \frac{1}{p(l_i)}} \quad \text{for } k = 1..N. \quad (15)$$

Note that the above inequality cannot be solved directly because it contains  $p(l_k)$  both on the left side of the equation and in the sum on the right side. Also, the equation cannot be solved if  $p(l_i)$  is equal to 0 for any  $i$ . Therefore, to find the segments with non-zero slopes, we split the segments into those whose probability  $p(l_k)$  is above a certain threshold  $p_t$ :

$$\Omega_t = \{k = 1..N : p(l_k) > p_t\} \quad (16)$$

and assign slope 0 to the remaining segments, and update the threshold probability iteratively:

$$p_{t+1} = \frac{|\Omega_t| - \frac{r}{\delta}}{\sum_{i \in \Omega_t} \frac{1}{p(l_i)}}. \quad (17)$$

for  $t = 1, 2, \dots$ , where  $|\Omega_t|$  is the cardinality of the set  $\Omega_t$ . When initialized with a small starting probability value, here  $p_0 = 0.0001$ , the recursive formula quickly converges and lets us separate segments into those with zero and positive slopes, and enforce the first constraint of the optimization problem.

Since the sums in the equations are over 20–30 segments, the tone-curve can be found with minimal computational overhead given only an image histogram. The complexity is reduced significantly compared to more complex metrics, *e.g.* the one used in [Mantiuk et al. 2008], where a multi-scale pyramid needs to be built and a series of quadratic programming problems need to be solved. In contrast to [Mantiuk et al. 2008], we measure contrast distortion using a simple  $L_2$  norm instead of a visual model. However, given the limited flexibility of a tone-curve, the benefits of a complex visual model are marginal.

#### 4.1 Noise- and content-aware tone-curve

In the previous sections, we assumed that the probability  $p(l_k)$  corresponds to the frequency of a given intensity value in an image (taken from an image histogram). This, however, is not an ideal estimator of the importance of a particular intensity level in an image. For example, if a substantial portion of an image contains a uniform surface, *e.g.* a white wall, the corresponding  $p(l_k)$  value is going to be high due to the peak produced in the histogram. Since flat surfaces are usually not the most salient part in an image, there is little purpose in assigning them high importance and allocating steep tone-curves for them. Similarly, night scenes often contain large regions with substantial amount of noise and only little details. Allocating dynamic range for such regions will lead to amplification of the noise and produce unattractive results.

To address this issue, we compute  $p(l_k)$  values taking into account both image content and noise levels, assigning higher importance to regions with contrast variations above the noise level. First, we estimate local contrast as a standard deviation computed within a Gaussian window:

$$c(x, y) = \sqrt{(g_\sigma * l^2)(x, y) - (g_\sigma * l)^2(x, y)}, \quad (18)$$

where  $*$  is the convolution operator and  $g_\sigma$  is a Gaussian kernel with the standard deviation  $\sigma$  (3 in our implementation). Then, we compute the probabilities as a histogram that is weighted by the contrast values which are greater than the noise level  $n$  (Equation 2) at the pixel position. Formally, this can be expressed as:

$$p(l_k) = \frac{\sum_{(x,y) \in B_k} c(x, y)}{\sum_{(x,y) \in S} c(x, y)}, \quad (19)$$

where

$$S = \{(x, y) : c(x, y) > n(x, y)\} \quad (20)$$

$$B_k = \{(x, y) \in S : l_k - 0.5\delta \leq l(x, y) < l_k + 0.5\delta\}. \quad (21)$$

$S$  is the set of all pixels whose local contrast is higher than the level of noise, and  $B_k$  is the subset of  $S$  which contains the pixels within a particular histogram bin  $k$ . In practice, this approach shifts the dark regions affected by sensor read-out noise towards dark tones, making them less visible. It also avoids overstretched contrast in large uniform areas.

**Tone priority** The noise-aware image contrast measure proposed above can be considered as a simple measure of image saliency. If this simple saliency measure is not sufficient, our method can easily be extended to more advanced measures, where, for example, higher saliency is assigned to detected faces or skin tones. We also found it useful to include a tone-priority parameter, which balances the importance of high or low-tones by weighting  $p(l_k)$  values depending on their input log-luminance. This gives an additional creative control parameter over the produced images.

#### 4.2 Temporal consistency

The image statistics  $p(l_k)$  can change rapidly between consecutive video frames. This may result in disturbing flickering artifacts, see [Eilertsen et al. 2013]. Flickering artifacts may be introduced either in the base–detail layer decomposition, or if the tone curve changes rapidly between frames.

Since the spatial filter (Section 5) combines a set of low-pass filtered images using a smooth edge-stop function, it will not introduce inconsistencies over time. Thus, the tone-curve is the only component of our operator that can introduce temporal fluctuations, and this is also trivial to smooth out over time to prevent such behavior.

The filtering of the tone curve(s) acts on the visually most relevant data, which are the node positions  $v_k$ . Through discussions with technical artists, we have found that the choice of filter depends on both the intent/artistic goals of the tone mapping as well as the input data. Our implementation is flexible in that it allows the tone curve filter to be interchanged. In most cases we use a 3-tap low-pass IIR filter with a cutoff frequency of 0.5 Hz. This frequency was selected to minimize visible flicker, see [Mantiuk et al. 2008]. IIR filters are trivial to implement and results in a smaller filter size than FIR filters. In some cases, we also use a temporal edge-stop filter which preserves sharp temporal transitions. However, experimenting with these (and other) filters we did not observe visible improvements, except in situations with extreme temporal variations. The choice of temporal filter and its effect on the pixel response is discussed and illustrated in detail in the supplementary material.

It should be noted that the temporal filtering may not preserve object brightness consistency [Boitard et al. 2012] and some objects may gradually change their brightness over time. Preserving brightness, however, would strongly reduce achievable image contrast and requires preprocessing the entire video sequence. Our approach trades off object brightness consistency for better contrast reproduction and real-time on-line processing. It is obviously possible to enforce stronger brightness consistency by using a lower cutoff frequency. Our tone-curve minimizes contrast distortions per-frame while the temporal filter minimizes flickering distortions in the temporal domain. The temporal filter may pull the solution per-frame from the point of optimum and result in a slightly worse solution for that frame, but better solution for the entire video clip. Even though it is possible to find a family of optimal tone-curves for the entire video sequence, this would make the method unsuitable for real-time processing and would bring little improvement in quality.

#### 4.3 Local tone-curves

The visual system is able to perceive a large range of luminances due to its ability to adapt to local scene regions. The exact mechanism and the spatial extent of such spatial local adaptation are still not well understood, but there is ample evidence of pooling information both locally and globally across the visual field, for example when making brightness judgments [Allred et al. 2012]. Inspired by the visual system’s local and global processing, we compute and



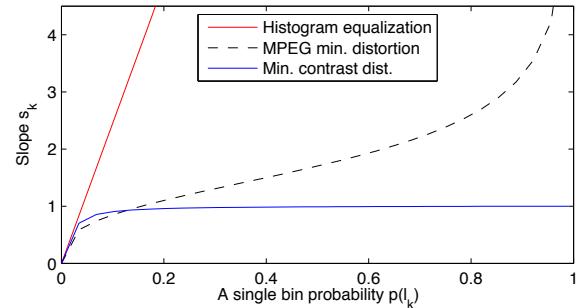
**Figure 5:** Tone-mapping using global (top) or local (bottom) tone-curves. The image was generated assuming high ambient light viewing (low contrast) to better show contrast reduction. Note that other components of the operator, such as local contrast, were disabled for this example. Video frame from [Froehlich et al. 2014].

apply different tone-curves to different image regions. As shown in [Reinhard and Devlin 2005], locally adaptive tone-curves can significantly boost visibility of image details without introducing any noticeable artifacts. Figure 5 shows a frame generated using global and local tone-curves.

We split an image into square tiles of 5 visual degrees each (about 230 pixels for a 15" full HD display seen from 45 cm), which is approximately the diameter of the fovea in the retina. Then, importance values  $p_t(l_k)$  are computed separately for each tile  $t$ . Such local luminance-importance statistics cannot be directly used for computing local tone-curves because it contains 0-values for the luminance levels that are missing in a particular tile but exist in an image. This results in highly locally adapted tone-curves, which are so different across an image that they cause visible discontinuities of tones. For that reason, we linearly blend the per-tile  $p_t(l_k)$  values with the global  $p(l_k)$  for the entire image in the ratio 10% global and 90% local, and then compute local tone-curves. To apply local tone-curves to an image, the tone-curve values are interpolated between the neighboring tiles, so that a 3D look-up is performed instead of a typical 1D look-up in case of a single global tone-curve. Each local tone-curve needs to be filtered independently over time using the IIR filter explained in the previous section. Note that the computational overhead of local tone-curves is minimal since the most expensive operation, which is computing  $p(l_k)$  for each tile, takes in total almost the same time as computing  $p(l_k)$ -values for the entire image.

#### 4.4 Comparison with other methods

Using a histogram to compute a tone-curve may appear similar to histogram equalization, which is a popular technique for image enhancement. We will show, however, that the objectives and the re-



**Figure 6:** The slope of a single segment  $k$  of a tone-curve as a function of probability  $p(l_k)$ . The values are computed for 30 segments and for probabilities uniformly distributed across the remaining segments ( $p(l_i) = (1 - p(l_k))/(N - 1)$  for  $i \neq k$ ). Different curves denote different tone-curve generation algorithms.

sults of the two methods are very different.

As shown in [Mai et al. 2011], histogram equalization is equivalent to allocating tone-curve slopes according to the formula:

$$s_k = \frac{r p(l_k)}{\delta \sum_{i=1}^N p(l_i)}. \quad (22)$$

Given that, we can compare slope allocation of our method (blue line, Equation 14) and histogram equalization (red line) in Figure 6. The figure shows how the slope of a tone-curve is allocated depending on the probability of the corresponding luminance level (or bin),  $p(l_k)$ . While the slope for our minimum contrast distortion approach (blue line) never exceeds the value of one (which corresponds to non-distorted contrast), histogram equalization attempts to assign the highest contrast possible to the segments of high probability  $p(l_k)$ . In Appendix B we show that the tone-curve slopes are exponentially related to the probability values  $p(l_k)$ , assigning very high slopes to the bins with high probabilities. This is one of the reasons why histogram equalization tends to over-enhance regions of uniform pixel values, which cause the probability values  $p(l_k)$  to be high. Larson et al. [Ward Larson et al. 1997] address this issue by clamping the maximum probabilities  $p(l_k)$  to the value determined from the threshold-versus-intensity function. This is, however, an ad-hoc approach, which does not ensure that contrast distortion is minimal.

Our formulation is also similar to the optimum tone-curve derived for the purpose of backward-compatible HDR image and video encoding [Mai et al. 2011], which is shown as a dashed line in Figure 6. This approach is less aggressive with contrast enhancement than histogram equalization, but will still boost contrast instead of preserving it because this results in lower coding errors after inverse-tone-mapping of the LDR layer.

## 5 Filter design for tone mapping

Base and detail layer decomposition ( $b$  and  $d$  in Figure 2) is a commonly used technique for preserving local contrast in tone-mapping. The base layer,  $b$ , obtained using an edge-preserving low-pass filter, is compressed using a tone curve, and the detail layer,  $d$ , is retained, or even enhanced.

The most important aspect in the decomposition is the choice of filtering method. Previous TMOs, see e.g. [Eilertsen et al. 2013] for an overview, have mostly relied on classical image filters designed for e.g. noise reduction. A common choice for base – detail

layer decomposition has been the bilateral filter [Aurich and Weule 1995; Tomasi and Manduchi 1998], mainly due to its simplicity. A problem, however, is that there are fundamental differences between the intent of the classical image filters and our goal here, *detail extraction*. First, image details are found at larger spatial scales than noise, i.e. detail extraction filters require large supports in the spatial and intensity domains without introducing visible artifacts. Secondly, the final result here is the base,  $b$ , and detail layer,  $d$ , and not a filtered image. The detail layer is highly sensitive to filtering artifacts, where the behavior along edges is extremely critical. Even small artifacts (which may be invisible in ordinary image filtering) may become visually disturbing after tone mapping, especially for video.

In Appendix C we show why the bilateral filter fails to correctly reconstruct the underlying signal (base layer) of a smooth edge, and which implications this have on the detail layer separation. We use these observations in the filter construction for our TMO, where we first relate the bilateral filter to anisotropic diffusion [Perona and Malik 1990], and from there derive an efficient filter that is specifically designed for the purpose of base – detail layer decomposition.

## 5.1 Filter construction

A diffusion operation for image filtering is carried out as a conduction process, in which the pixel values,  $I(\mathbf{p})$ , are propagated over time,  $t$ , to their neighbors according to the diffusion PDE:

$$\frac{\partial I(\mathbf{p})}{\partial t} = \operatorname{div}(w_r(\mathbf{p}, t) \cdot \nabla I(\mathbf{p})) \quad (23)$$

If  $w_r(\mathbf{p}, t) = c$  is constant, this reduces to the isotropic heat diffusion equation  $\frac{\partial I}{\partial t} = c\Delta I$ . In non-linear diffusion edges are preserved by constraining the diffusion to uniform image regions using the image gradient magnitudes as a stopping criterion,  $w_r(||\nabla I(\mathbf{p})||)$ . In image filtering, this is approximated by discretizing the time domain, and by iteratively calculating the inflow,  $V$ , at each pixel,  $\mathbf{p}$ , from its closest neighbors in an adaptive smoothing procedure. For iteration  $k + 1$  this can be described as:

$$I^{k+1}(\mathbf{p}) = I^k(\mathbf{p}) + V(I^k, \mathbf{p}) \quad (24)$$

To preserve high contrast structures, the weights are based on distances in both the spatial and intensity domain. In contrast to this, the bilateral filter runs in a single iteration using a larger neighborhood for the weighted average. As described in [Durand and Dorsey 2002], 0-order anisotropic diffusion can be extended with a larger spatial support to allow for a connection to the bilateral filter:

$$V(I, \mathbf{p}) = \alpha \sum_{\mathbf{q} \in \Omega_p} w_s(||\mathbf{q} - \mathbf{p}||) w_r(I(\mathbf{q}) - I(\mathbf{p})) (I(\mathbf{q}) - I(\mathbf{p})) \quad (25)$$

|              | Anisotropic diffusion (AD)   | Bilateral filter (BF)   | Our approach   |
|--------------|--|---|--|
| Distance     | $w_s(x) = \begin{cases} 1 & (x = 0) \\ 1/x, & (x \neq 0) \end{cases}$                  | $w_s(x) = \exp\left(-\frac{x^2}{2\sigma_s^2}\right)$                                    | $w_s(x) = \exp\left(-\frac{x^2}{2\sigma_k^2}\right)$   |
| Edge-stop    | $w_r(x) = 1/\left(1 + \left(\frac{x}{\lambda}\right)^2\right)$                         | $w_r(x) = \exp\left(-\frac{x^2}{2\sigma_r^2}\right)$                                    | $w_r(x) = \begin{cases} \left(1 - \left(\frac{x}{\lambda}\right)^2\right)^2, & (x \leq \lambda) \\ 0, & (x > \lambda) \end{cases}$ |
| Neighborhood | $\Omega_p = \{\mathbf{q} \in \mathbb{Z} :   \mathbf{q} - \mathbf{p}   \leq \sqrt{2}\}$ | $\Omega_p = \{\mathbf{q} \in \mathbb{Z} :   \mathbf{q} - \mathbf{p}   \leq 3\sigma_s\}$ | $\Omega_p = \{\mathbf{q} \in \mathbb{Z} :   \mathbf{q} - \mathbf{p}   \leq 3\sigma_k\}$  |
| Iterations   | $k = 1, \dots, N$  | $k = 1$   | $k = 1, \dots, N$  |

**Table 1:** The different local behavior and properties of commonly used methods, compared to the isotropic diffusion approximation filter. The different edge-stop functions are illustrated at the right, Figure 7.

## Algorithm 1 Outline of our fast detail extraction diffusion

---

```

1: input  $l, N, \sigma, \lambda$  ▷ Input  $l$  is in the log domain
2:  $l_f \leftarrow l$ 
3: for  $k = 1, 2, \dots, N - 1, N$  do
4:    $\sigma_k \leftarrow \sigma_g(k)$  s.t.  $\sqrt{\sum_{i=1}^k \sigma_g^2(i)} = k\sigma$  ▷ Iterative filtering of a signal corresponds to filtering original signal with linearly increasing filter size
5:    $l_n \leftarrow g_{\sigma_k} * l_f$ 
6:    $||\nabla l_f|| \leftarrow$  Equation 27 ▷ Gradient magnitude
7:    $||\nabla l_f|| \leftarrow \max(||\nabla l_f||, k|l_n - l|)$  ▷ Constrain gradient magnitude
8:    $w_r \leftarrow$  Table 1 ▷ Edge-stop function
9:    $l_f \leftarrow (1 - w_r) \cdot l_f + w_r \cdot l_n$  ▷ Iterative filtering refinement
10: end for
11: return  $l_f$ 

```

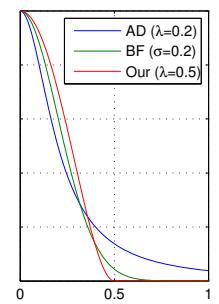
---

Here,  $\alpha$  is the diffusion rate, which for the bilateral filter can be interpreted as the normalization  $\alpha = 1 / \sum w_s w_r$ . Using different neighborhoods  $\Omega$ , spatial weights  $w_s$  and intensity weights  $w_r$ , a range of different filters can be described. For anisotropic diffusion and the bilateral filter these functions are shown in Table 1. In order to maintain low complexity and avoid artifacts, we employ an isotropic approach:

$$V(I, \mathbf{p}) = \alpha w_r(||\nabla I(\mathbf{p})||) \sum_{\mathbf{q} \in \Omega_p} w_s(||\mathbf{q} - \mathbf{p}||) (I(\mathbf{q}) - I(\mathbf{p})) \quad (26)$$

where  $\nabla I(\mathbf{p})$  is the image gradient at a pixel position  $\mathbf{p}$ . This filter can be evaluated very fast by iteratively applying linear filters, which are weighted using the per-pixel edge-stop function,  $w_r$ . Since the isotropic kernel has uniformly distributed samples around  $\mathbf{p}$ , it yields an unbiased result without over-sharpening problems. To preserve edges, the kernel size adapts to image structure, which means that a smaller number of samples are used along steep (high gradient) edges. However, since the aim of the spatial filtering is base – detail separation, it is safer to let details be less pronounced along edges than risking artifacts (see Figure 9). Furthermore, since the edge itself is perceptually dominant we argue that a possible loss of texture detail is significantly less disturbing compared to an incorrect and/or temporally incoherent edge.

The filter behavior along edges is determined by the edge-stop function and the way the gradient estimates are computed. We make the observation that isotropic kernels inherently require a conservative stopping function in order to propagate the flow close to, but not across edges. To accomplish this, we use Tukey's biweight [Black et al. 1998], see Table 1, which conservatively stops the diffusion at gradient magnitudes equal to or greater than  $\lambda$ . Also, we use a robust gradient formulation expressed as a linear ramp over the local neighborhood  $\Omega_p$  around pixel  $\mathbf{p} = (x, y)$ , according to:



**Figure 7:**  $w_r(x)$

$$\begin{aligned} \|\nabla I(x, y)\| &= \sqrt{\nabla_x^2 I(x, y) + \nabla_y^2 I(x, y)}, \\ \nabla_x I(x, y) &= \sum_{\delta=-\lceil 3\sigma_k \rceil}^{\lceil 3\sigma_k \rceil} \delta I(x + \delta, y), \\ \nabla_y I(x, y) &= \sum_{\delta=-\lceil 3\sigma_k \rceil}^{\lceil 3\sigma_k \rceil} \delta I(x, y + \delta) \end{aligned} \quad (27)$$

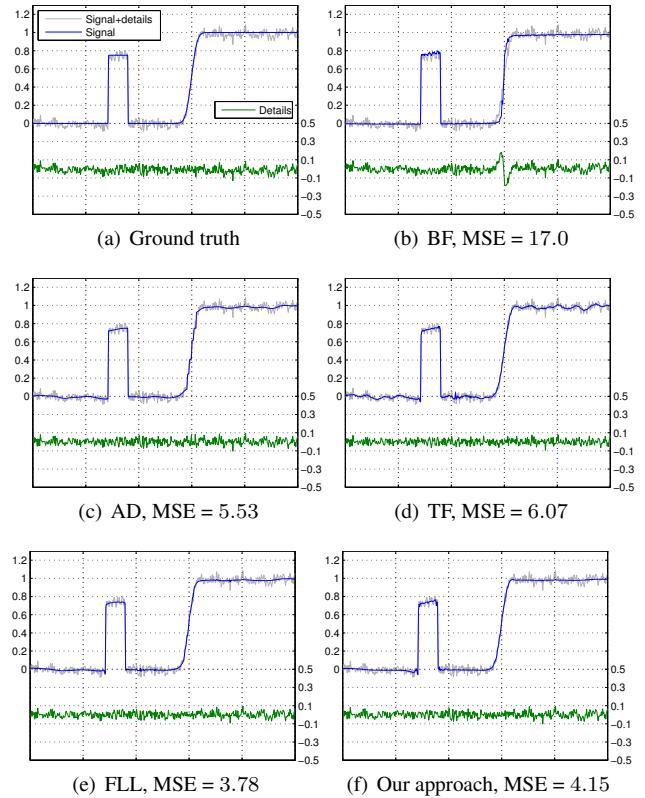
where  $\lceil \cdot \rceil$  denotes the ceiling operation. With this formulation the diffusion stops faster at edges compared to using *e.g.* difference of Gaussians (DoG). When Equation 26 is combined with Equation 27, the flow is completely stopped when a pixel is closer to a high contrast edge than the radius of the neighborhood  $\Omega$ . To enable filtering close to edges while ensuring fast diffusion, we iteratively change the size of  $\Omega$ , starting with a very small kernel and increasing its radius as the diffusion progresses. The size of  $\Omega$  varies so that the net size of the filter after  $N$  iterations is  $N\sigma$ , where  $\sigma$  is the starting size of the kernel. That is,  $\sqrt{\sum_{k=1}^N \sigma_k^2} = N\sigma$ . Finally, we constrain the diffusion using the distance to the original image, to prevent possible extreme values in the reconstruction (see line 7 in Algorithm 1).

The final formulation of the filter design is given in Table 1 and outlined in Algorithm 1. A convergence analysis including motivations for setting the number of iterations  $N$  can be found in the supplementary material. For all examples in the paper, we have used  $N = 12$  iterations.

## 5.2 Comparison with other methods

In order to show the behaviors of different low-pass edge-stop filters and relate them to our fast detail extraction diffusion filter, we first illustrate this using a one-dimensional signal in Figure 8. The input, see 8a, is a signal (blue) with details (green) added using a noise function. We compare our local detail extraction diffusion to the classical bilateral filter (BF) and anisotropic diffusion (AD), tri-lateral filter (TF) [Choudhury and Tumblin 2003], and the current state-of-the-art filter fast local laplacian (FLL) [Aubry et al. 2014]. At the step edge on the left, where a piece-wise constant assumption holds, the 0-order filters show no problems. However, for the smooth edge on the right, BF shows banding and AD is prone to staircase artifacts. TF and FLL, however, are both computationally expensive compared to our method. Another important difference is that our filter is specifically designed for detail extraction for tone mapping without introducing artifacts, while *e.g.* FLL show a tendency to create false edges at smooth intensity transitions.

Finally, in Figure 9, we evaluate our filter design in the context of local tone mapping. The tone curve described in Section 4 has been applied to the filtered images, followed by adding back extracted details, scaled for easier comparison in print. The fast local laplacian filter (FLL) is displayed to demonstrate the difference between our method and current state-of-the-art filtering. In addition, the permeability filter (PF) introduced by [Aydin et al. 2014] for detail extraction, is demonstrated in the figure. This filter works well for minor to moderate detail enhancements, but artifacts are clearly visible for stronger manipulations. From the examples it becomes evident that it is better to limit the filtering around edges instead of risking the artifacts shown by classical 0-order filters (BF and AD). Compared to FLL and PF, our fast detail extraction diffusion may lose a small amount of detail along some of the edges. However, in the case of tone mapping, this is the preferred behavior as



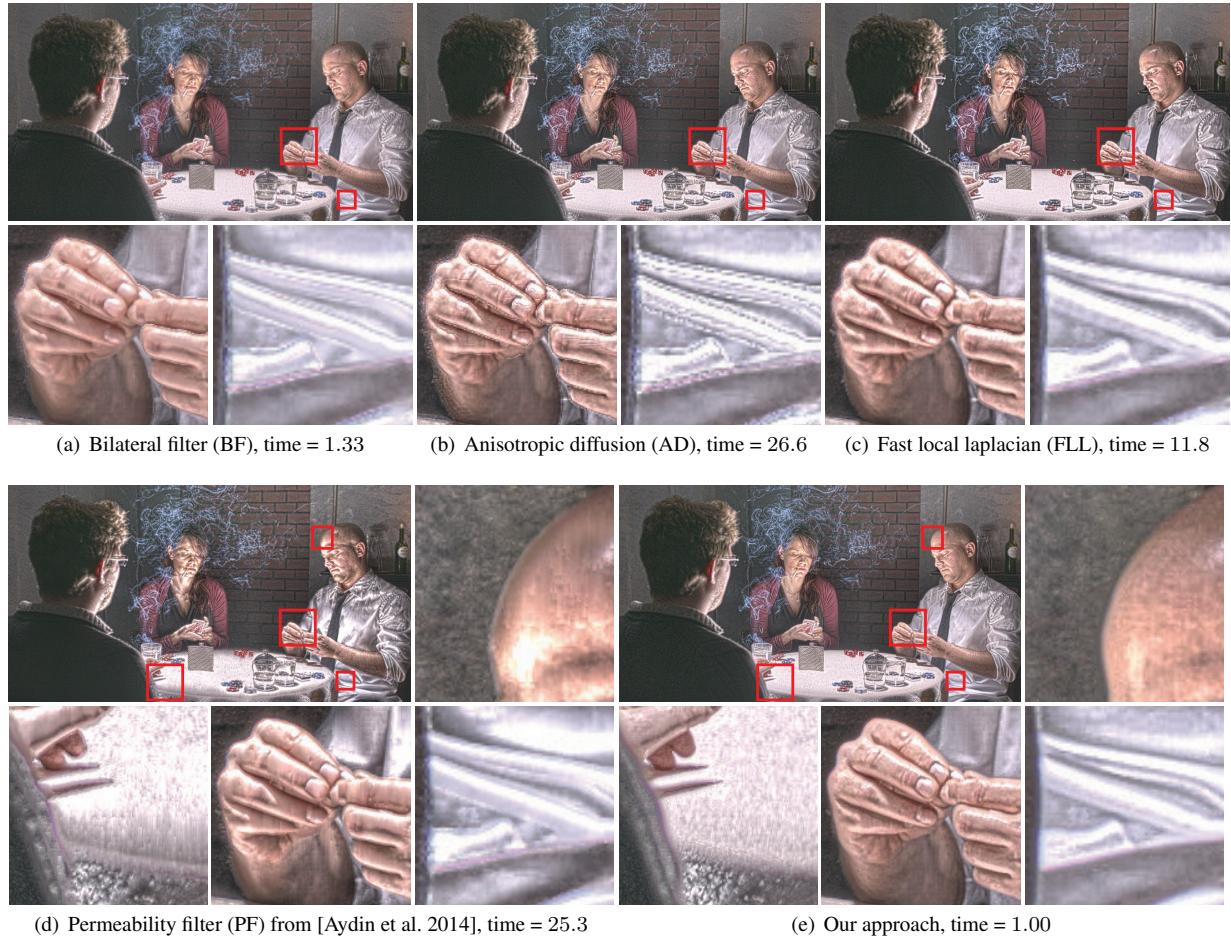
**Figure 8:** Different filters are applied to the input in the top left plot, where the details in green, in this case represented by noise, are added to the original signal. The filters are then used to reconstruct the signal without details. The MSE of the reconstructions are scaled by  $10^4$ , and provided for numerical comparison.

it ensures robust detail enhancement, while the other filters may introduce artificial edges (ringing artifacts) and temporal inconsistencies at smooth image transitions (see *e.g.* the edges on the shirt and the fingers in Figure 9).

To give an indication on the performance of the filters, processing times are also shown in Figure 9. These are relative to our approach, and all report timings for Matlab implementations (the bilateral grid acceleration scheme is used for the BF [Chen et al. 2007]). This serves as a simple indicator of performance of non-parallelized code, in order to reflect the complexity of evaluation. It should be noted, however, that the different methods have different potential for parallelization, where *e.g.* our detail extraction diffusion is specifically designed for this purpose, and shows a performance increase of about 2 orders of magnitude on the GPU.

## 6 Noise-aware local contrast control

Once the base layer is tone-mapped with local tone-curves, it can be recombined with the detail layer. If the detail layer is left unchanged, the tone-mapping will preserve local contrast. If the detail layer values are amplified, the image details are boosted, which can produce attractive looks when combined with a high quality edge-stopping spatial filter. Enhancing image details, however, carries the risk of amplifying noise. The noise-aware tone-curve (Section 4.1) can conceal some noise in darker tones, especially sensor-read-out noise, but it is not effective in hiding noise in brighter image parts. Therefore, our detail layer  $d$  is modulated relative to the



**Figure 9:** The figures use the same tone curve, but different edge-preserving filters for detail extraction. The details are scaled by a factor 5 to highlight filtering differences and artifacts, and enlarged versions show the areas indicated by the red squares in the images. Timings of Matlab implementations of the filters are also given, compared relative to our approach. Video frame from [Froehlich et al. 2014].

visibility of noise in the displayed image:

$$d_{out}(x, y) = e \min \left( 1, \frac{V(b_{tm}(x, y))}{n(b(x, y))} \right) \cdot d(x, y), \quad (28)$$

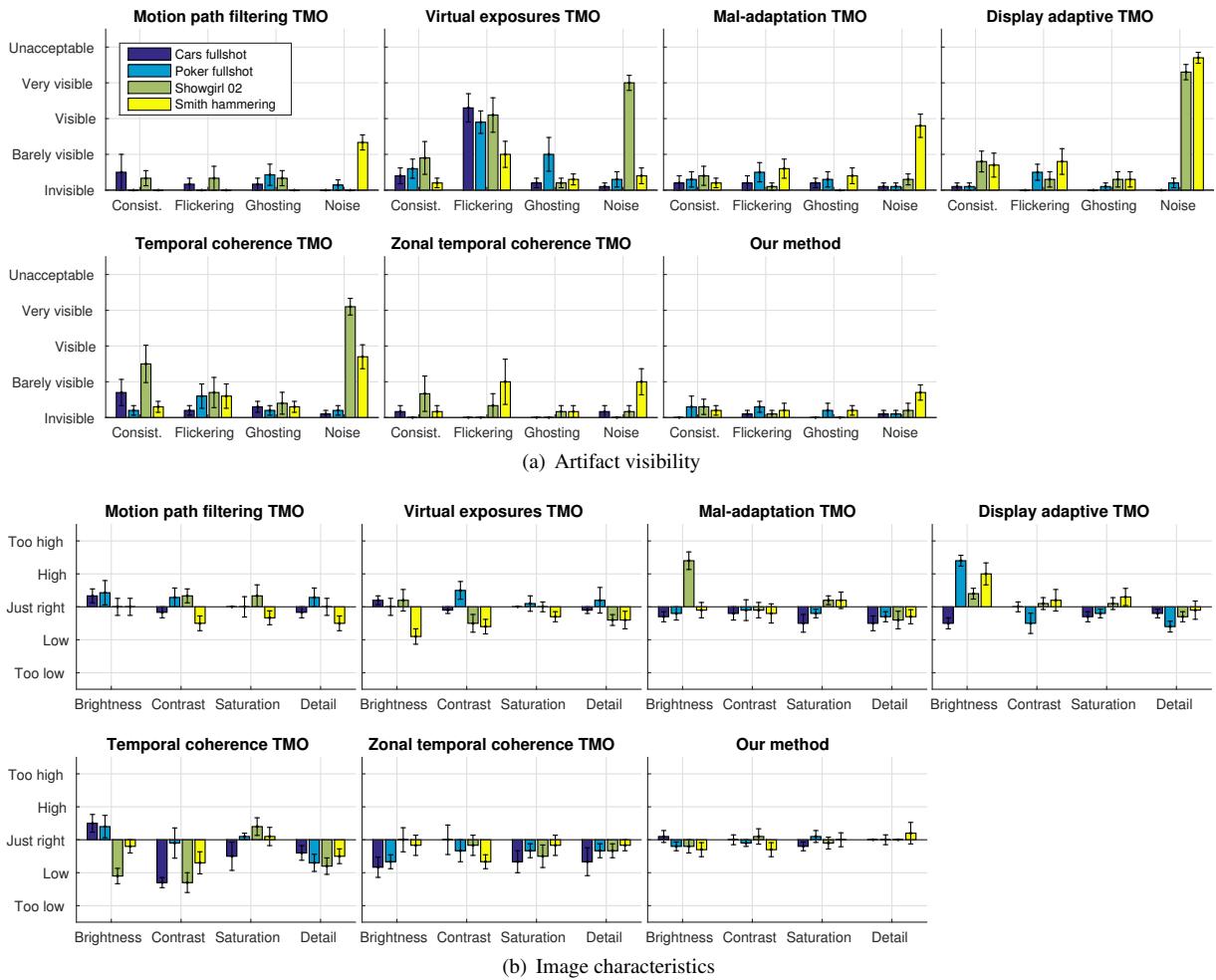
where  $V()$  is the visibility threshold from Equation 3 and  $n()$  is the noise level in the log-luminance domain (Equation 2). Note that the visibility is assessed based on the displayed log-luminance produced by tone-curves  $b_{tm}$ , while the noise depends on the log-luminance of the input image.  $e$  is an optional local-contrast enhancement parameter, which enables boosting details and thus creative control. The “min” term in the equation effectively reduces contrast whenever the amplitude of noise is above the detection threshold, which corresponds to the case when the magenta line of noise amplitude in Figure 3 is above the visibility thresholds, shown as the blue line.

## 7 Results and applications

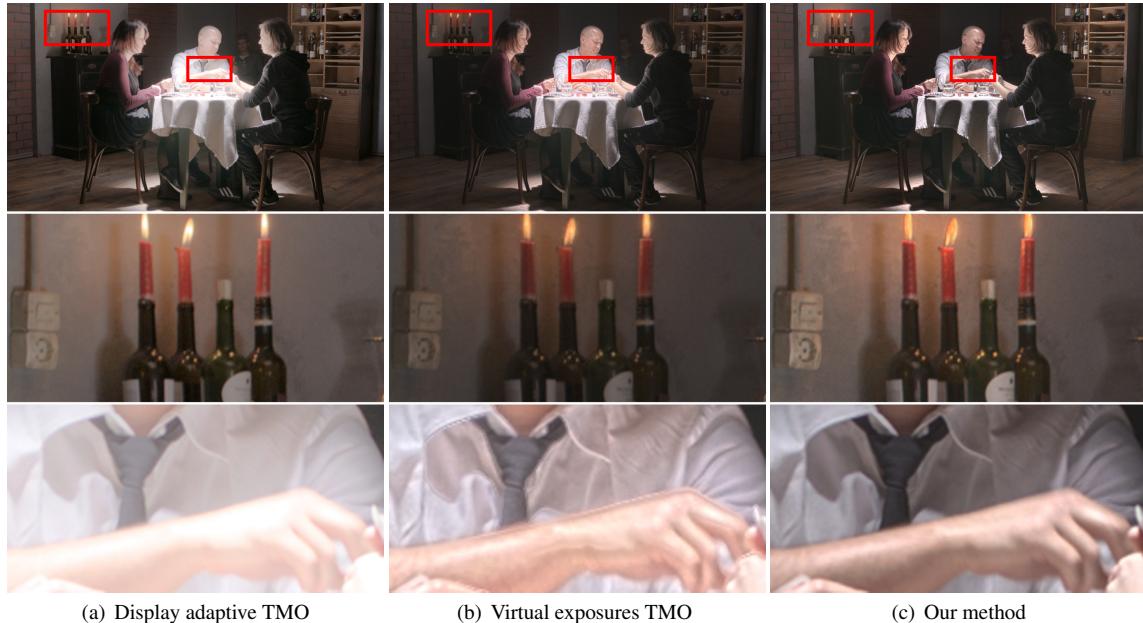
In this section we present an overview of our tone mapping operator in terms of visual quality, performance and features. We also discuss the specific features of our algorithm, including noise awareness, adaptation to display and viewing conditions, and detail enhancement, and demonstrate how our TMO can be applied in the context of a set of common imaging applications. More results are also included in the supplementary video and document.

### 7.1 Results and evaluation

To evaluate the performance of our method in terms of visual quality, we performed a subjective evaluation, as a qualitative analysis experiment. The experiment compared our TMO to six state-of-the-art video tone mapping methods; two global operators: *Mal-adaptation TMO* [Irawan et al. 2005] and *Display-adaptive TMO* [Mantiuk et al. 2008], and four local operators: *Virtual exposures TMO* [Bennett and McMillan 2005], *Temporal coherence TMO* [Boitard et al. 2012], *Zonal temporal coherence TMO* [Boitard et al. 2014] and *Motion path filtering TMO* [Aydin et al. 2014]. The evaluation was carried out as a rating experiment where 10 users experienced in image processing viewed, in random order, a set of video clips. These were taken from [Froehlich et al. 2014], and were each tone-mapped with the seven operators. The users were asked to provide ratings for each clip according to the following attributes: overall *brightness*, overall *contrast*, overall color *saturation*, temporal color *consistency*, temporal *flickering*, *ghosting*, excessive *noise*, as well as *detail* reproduction to assess the contrast at a local level. The final result of the ratings, averaged over the observers, is illustrated in Figure 10 for the different sequences. Overall it is clear that the proposed noise aware TMO consistently produces results that show image characteristics at about just the right level without visible artifacts. The full experiment and analysis of the data is described in detail in the supplementary material.



**Figure 10:** Qualitative analysis result, showing the average ratings from the conducted experiment, with error bars for the standard errors.



**Figure 11:** Comparison to a selection of current state-of-the-art TMOs: a) [Mantiuk et al. 2008], and b) [Bennett and McMillan 2005]. Video frame from [Froehlich et al. 2014].



**Figure 12:** The filtering approach used in [Aydin et al. 2014], with the tone curves used by the authors, compared to ours. a) scaling in the log domain, b) log mapping from [Drago et al. 2003], and c) our local tone curve. Video frame from [Froehlich et al. 2014].

Figure 11 shows a representative example of a visual comparison between our algorithm and two state-of-the-art TMOs taken from the user evaluation. The figure compares our method to the best performing TMO from the evaluation in [Eilertsen et al. 2013], *display adaptive TMO* [Mantiuk et al. 2008], and the TMO which is conceptually most similar to our approach, *virtual exposures TMO* [Bennett and McMillan 2005]. The magnifications show examples of the tone mappings in low and high luminance areas, respectively. The display adaptive TMO, which relies on global processing, shows problems in compressing the large dynamic range in the scene which leads to bright results and loss of local details and contrast. The virtual exposures TMO relies on bilateral filtering which in some situations leads to ringing artifacts along edges. There are also problems in adapting the tone curve over time, resulting in disturbing temporal flickering (see Figure 10). Compared to these TMOs, our method handles the dynamic range in the scene very well and preserves local detail and contrast without any artifacts.

Figure 12 shows the result of the method from [Aydin et al. 2014] using different tone curves. In 12a and 12b the tone curves are the same as used in the original article (*i.e.* log scaling and log mapping from [Drago et al. 2003]), and in 12c we use our local tone curves as described in Section 4. Although the overall filtering employed by Aydin et al. shows good performance in terms of visible artifacts (see Figure 10), the actual tone reproduction and temporal processing of the tone curve are not considered. From the figure it is clear that we achieve a better compression of the dynamic range while preserving the local contrasts. Also, the optical flow filtering used may not work well in complicated image transitions (see *e.g.* the noise level in the *smith hammering* sequence in Figure 10). Finally, due to the filtering process, the method is computationally expensive and not suited for real-time applications.

## 7.2 Performance

All steps in our algorithm can be computed in parallel and are suitable for GPU implementation. The spatial filtering is constructed using separable low-pass filters, together with horizontal and vertical gradient filters. This means that we only need to run four 1D

|                        | 1280×720px | 1920×1080px |
|------------------------|------------|-------------|
| Detail extraction (ms) | 7.19       | 17.1        |
| Tone curve (ms)        | 1.62       | 3.31        |
| Total time (ms)        | 9.18       | 21.5        |
| Framerate (fps)        | 109        | 46.5        |

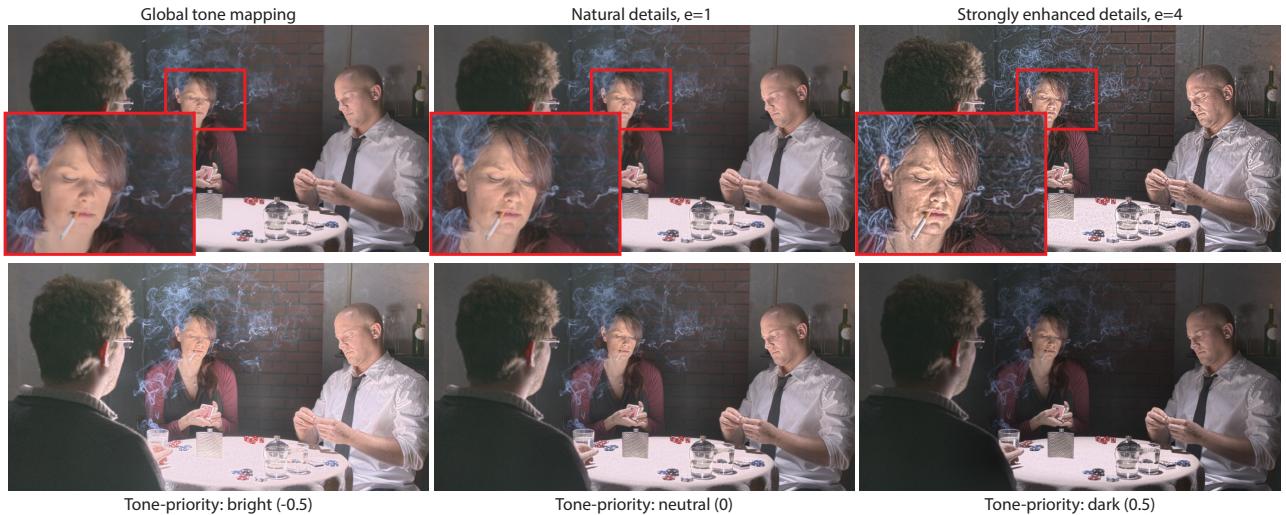
**Table 2:** Timings of the complete TMO-chain

filter kernels at each iteration of the diffusion process. The local histogram calculation and the temporal filtering of the tone curves are trivially parallel. All parts of the TMO were implemented using CUDA 6.5. With a modern graphics card, the complete TMO pipeline runs in real-time on full high definition material. Table 2 shows the performance of our algorithm using 720p and 1080p HD input running on a GeForce GTX 980.

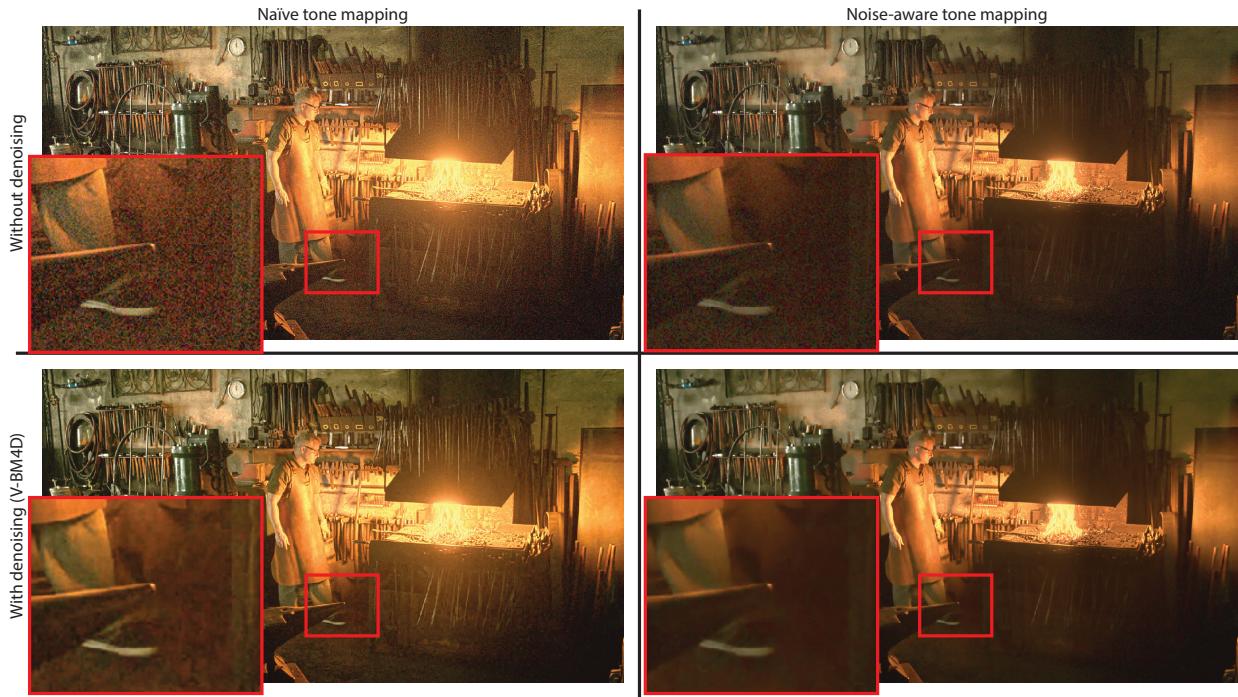
## 7.3 Applications and features

In this section we demonstrate some of the features of our algorithm and review a set of application scenarios for our video tone-mapping solution and demonstrate sample results.

**Video postprocessing** When video requires color-grading, our method offers both high quality automatic adjustment, and a range of creative contrast controls for stylization. Especially attractive is our detail enhancement, which can maintain or strongly increase detail visibility without noticeable artifacts (Figure 13 top). This is shown in Figure 13 top-left and top-center, where the results of global tone-mapping are compared with the image in which the detail layer was preserved at the original contrast. Tone-priority adjustment allows to shift focus between darker and brighter tones while minimizing distortions, such as clipping, without the complication of controlling the exact shape of a tone-curve (Figure 13 bottom). Because the operation can be run at real-time frame-rates, visual feedback is provided at full resolution as a particular parameter is adjusted. As we focus on contrast and do not address color issues, our method can be combined with existing solutions to address all aspects of color-grading.



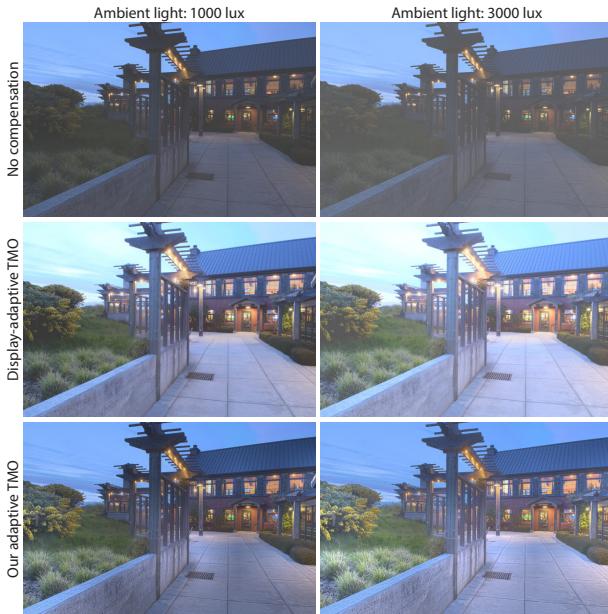
**Figure 13:** Top row: Compared to using a global tone curve (leftmost image), detail enhancement results in improved image sharpness without halo or ringing artifacts, at a smaller computational costs than many edge-preserving filters. Even very strong enhancement, which is often desirable for stylization, does not introduce disturbing artifacts (rightmost image). Bottom row: the tone-priority setting can shift focus between dark and bright tones while minimizing contrast distortions, such as clipping. Video frame from [Froehlich et al. 2014].



**Figure 14:** Comparison of tone-mapping disregarding noise (top left) with our noise-aware approach (top right), where excessive noise is hidden in darker tones. The bottom row shows the result of complementing the tone mapping with a denoising step (V-BM4D). Enlarge to compare the visibility of noise. Video frame from [Froehlich et al. 2014].

**In-camera processing stacks** Images captured by sensors need to go through a chain of operations, such as demosaicing, denoising and sharpening, before they can be displayed on a digital viewfinder or stored as (JPEG) images or (MPEG) movies. A key operation in this chain is tone-mapping, in which the larger dynamic range of the sensor is mapped into a smaller dynamic range supported by a display or an output image file format. Our solution is well suitable for that purpose as it offers automatic tone-mapping and detail enhancement, which adapts to camera noise levels. Figures 1 and 14 show the result of our tone-mapping with and without noise-aware processing. To reduce noise visibility, enhancement is reduced in

noisy image regions and most of the sensor read-out noise is hidden in the darker image tones. However, if the preview of noise levels is desirable, for example in digital viewfinder, noise-aware processing can be disabled, as shown in the top left in Figures 1 and 14. To underline that the noise-aware tone reproduction does not actually compete with denoising, Figure 14 also show the result of applying a state-of-the-art denoising method (V-BM4D, [Maggioni et al. 2012]). The denoising step is performed in the log domain, before the tone mapping. Without the noise-aware capabilities, artifacts from the denoising are clearly visible, while when complementing the both methods these are hidden in the darker tones of the image.



**Figure 15:** The effect of ambient light on the results of tone-mapping. Top-row: our method disregarding ambient light; middle row: display-adaptive TMO [Mantiuk et al. 2008]; bottom row: our method adapting to ambient light. The image contrast (dynamic range) is the same in each column. Simulated display: black level:  $1 \text{ cd/m}^2$ ; reflectivity: 1%; peak lum.:  $100 \text{ cd/m}^2$ . Notice the difference in detail visibility.

**Display algorithms for ambient light compensation** The effective contrast of a display strongly depends on ambient light levels. When a mobile device is used in sunlight, an emissive display (LCD, OLED) is hardly legible since light reflected from the screen reduces its contrast. The usual remedy is to increase screen brightness, but this strongly increases power consumption. Another approach is to tone-map the content shown on the display to adapt to the effective image contrast in particular viewing conditions. This is shown in Figure 15, in which the top row shows non-adapted images and bottom row adapted images. The content is clearly more visible and exhibit an overall better quality in the latter case. When compared to the display adaptive tone-mapping proposed in [Mantiuk et al. 2008] (middle row), our method results in better contrast and detail visibility thanks to its local processing.

## 8 Conclusion

This paper described a new video tone-mapping solution, incorporating a set of novel methods and features not offered by existing methods. One of our main contributions is a novel technique for computing tone-curves, which minimizes contrast distortions. We show that a fast formula for computing such tone-curves can be derived from an optimization problem without the need to numerically solve quadratic programming. In contrast to existing ad-hoc methods, such as histogram equalization, the formula solves a well-defined minimization problem. The local tone-curves dynamically adapt to image content, noise levels and display capabilities, including the compensation for ambient light. The second main contribution is a novel edge-stop filter, which is explicitly designed for preserving and enhancing details in tone mapping. It avoids overshooting and ringing artifacts on soft edges, which is a common problem with most filters used in tone mapping, in particular when detail enhancement is required. Since the filter can be implemented as iterative Gaussian blurring, it leads to an efficient hardware im-

plementation. Finally, we combine both technical contributions in a comprehensive video tone-mapping operator, which controls the visibility of noise and adapts to a display and ambient light.

Our method is limited to two particular intents of tone mapping: scene reproduction and best subjective quality. The method does not attempt to mimic the visual system, for example by adding glare effects or simulating night vision. We also do not address any complex color issues arising from tone-mapping and rely on existing solutions. Higher quality tone-mapping could potentially be achieved with advanced denoising methods or by analyzing the entire length of the movie sequence. This, however, would eliminate one of the main advantages of our method, which is real-time processing.

Although tone mapping is usually seen as a high dynamic range problem, in practice it has a much broader field of applications. When a display is seen at high ambient light levels and its effective contrast is reduced, even low dynamic range content may benefit from tone mapping. Some mobile devices already incorporate such compensation algorithms. Many camera sensors capture higher physical contrast than a display (or a digital viewfinder) can reproduce. Therefore, tone mapping is an essential component in any in-camera processing stack, even if a single exposure (LDR) is captured. Finally, it is important to recognize that in many applications the images need to be reproduced based on the desire of an artist. Tone mapping operators used in such applications need to offer creative control parameters, which are easy to manipulate and can be explored with real-time visual feedback. We believe the proposed solution is the first to meet all those requirements.

## Acknowledgments

We would like to thank the volunteers who participated in the experiment. We also thank Jan Fröhlich *et al.* for the HDR video sequences used throughout the paper (<https://hdr-2014.hdm-stuttgart.de>), and Ronan Boitard for implementations of the temporal coherence TMOs used in the evaluation.

This project was funded by the Swedish Foundation for Strategic Research (SSF) through grant IIS11-0081, Linköping University Center for Industrial Information Technology (CENIIT), the Swedish Research Council through the Linnaeus Environment CADICS, and through COST Action IC1005.

## Appendix A: Derivation of tone-curve slopes

This appendix shows how the optimum slopes (Equation 14) can be found analitically from Equation 13 using the KTT method. Let us consider for now only the equality condition of the second constraint (Equation 12). Minimizing  $\varepsilon'(s_k)$  with this constraint is equivalent to minimizing the functional:

$$\sum_k p(l_k)(1 - s_k)^2 + \lambda \left( \sum_{k=1}^N s_k - \frac{r}{\delta} \right), \quad (29)$$

where  $\lambda$  is the Lagrange multiplier. This functional is minimized by solving the system of equations:

$$\begin{cases} -2(1 - s_1)p_1 + \lambda = 0 \\ -2(1 - s_2)p_2 + \lambda = 0 \\ \vdots \\ -2(1 - s_N)p_N + \lambda = 0 \\ \sum_{i=1}^N s_i - \frac{r}{\delta} = 0 \end{cases}. \quad (30)$$

We can eliminate the  $\lambda$  variable by combining any two equations except the last to get:

$$(1 - s_k) p_k = (1 - s_i) p_i \Rightarrow s_i = 1 - \frac{(1 - s_k)p_k}{p_i}. \quad (31)$$

After introducing the above equation to the last line of Equation 30, we get the solution given in Equation 14.

## Appendix B: Slopes of histogram equalization

Here we show that by reversing the steps we performed to derive the slope allocation formula in Section 4, we can find the hypothetical objective function for histogram equalization, which is:

$$\varepsilon_{HE}(s_k) = \sum_{k=1}^N p(l_k) \log(s_k), \quad (32)$$

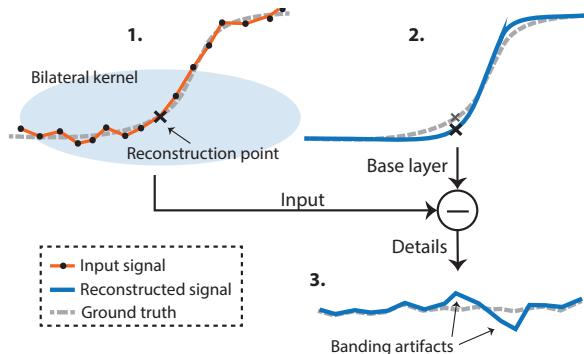
subject to the same constraints as in Equation 12, with the difference that the functional is maximized and not minimized. The solution of the above equation results in the slopes given by Equation 22. The formulation is not ideal as there is a singularity at  $s_k = 0$ , which needs to be handled as a special condition, which we omit here for clarity. The objective function shows that the histogram equalization procedure distributes the logarithms of slopes according to the probability of each bin  $k$ . This means that the tone-curve slopes are exponentially related to the probability values  $p(l_k)$ . Such relation often results in assigning very high slopes to the bins with high probabilities, which is undesirable in most tone-mapping applications.

## Appendix C: Analysis of detail extraction artifacts

Many bilateral filter extensions and acceleration schemes, such as [Durand and Dorsey 2002; Chen et al. 2007; Adams et al. 2009; Adams et al. 2010; Baek and Jacobs 2010; Yoshizawa et al. 2010; Banterle et al. 2012; Yang 2012], make real-time processing fully possible. However, the filters that assume a piece-wise constant underlying signal, such as the bilateral and anisotropic diffusion [Perona and Malik 1990], fail to correctly reconstruct complex spatial intensity transitions in natural images [Takeda et al. 2007b]. This effect is illustrated in Figure 16 where the filter kernel is biased towards one side of a smooth edge (edges in natural images are band limited due to the area sampling at sensor pixels). In the reconstruction, this bias and the assumption of a piece-wise constant signal leads to over-sharpening. The resulting reconstruction artifacts can in many applications be accepted as visually insignificant. For the task of detail extraction, however, this over-sharpening causes non-negligible banding or ringing effects (see for example Figures 8b and 9a), especially if the details are artistically enhanced.

One way to alleviate these problems is to reconstruct the underlying base layer using higher order approximations, e.g. the trilateral filter, kernel regression, or local polynomial approximations [Takeda et al. 2007a; Katkovnik et al. 2006; Milanfar 2013]. However, this comes at the cost of significantly increased complexity which makes real-time evaluation on high resolution footage difficult and in most cases even impossible. These higher order filters also tend to be sensitive to the parameter settings.

Another common option for edge-preserving filtering are the diffusion based algorithms. Anisotropic non-linear diffusion was introduced by Perona and Malik [1990], and bears many resemblances with bilateral filtering. Specific unified formulations of anisotropic diffusion and bilateral filtering have also been shown in e.g. [Barash



**Figure 16:** When filtering a point near the edge with the bilateral filter, the kernel puts higher weights on samples on the same side of the edge, resulting in a bias, which leads to overshooting in the filtered output.

2002] and [Durand and Dorsey 2002]. Since this filter also relies on a piece-wise constant assumption, the output is prone to show inconsistent behavior along edges similarly to the bilateral filter (see Figure 9b).

## References

- ADAMS, A., GELFAND, N., DOLSON, J., AND LEVOY, M. 2009. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graphics* 28, 3, 21:1–21:12.
- ADAMS, A., BAEK, J., AND DAVIS, M. 2010. Fast High-Dimensional Filtering Using the Permutohedral Lattice. *Computer Graphics Forum* 29, 2, 753–762.
- ALLRED, S., RADONJIĆ, A., AND GILCHRIST, A. 2012. Lightness perception in high dynamic range images: Local and remote luminance effects. *Journal of Vision* 12, 2, 1–16.
- AUBRY, M., PARIS, S., HASINOFF, S. W., KAUTZ, J., AND DURAND, F. 2014. Fast local laplacian filters: Theory and applications. *ACM Trans. Graphics* 33, 5, 167:1–167:14.
- AURICH, V., AND WEULE, J. 1995. Non-linear gaussian filters performing edge preserving diffusion. In *DAGM-Symposium*, Springer, G. Sagerer, S. Posch, and F. Kummert, Eds., Informatik Aktuell, 538–545.
- AYDIN, T. O., STEFANOSKI, N., CROCI, S., GROSS, M., AND SMOLIC, A. 2014. Temporally coherent local tone mapping of HDR video. *ACM Trans. Graphics* 33, 6, 1–13.
- BAEK, J., AND JACOBS, D. E. 2010. Accelerating spatially varying gaussian filters. *ACM Trans. Graphics* 29, 6, 169:1–169:10.
- BANTERLE, F., CORSINI, M., CIGNONI, P., AND SCOPIGNO, R. 2012. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. *Computer Graphics Forum* 31, 1, 19–32.
- BARASH, D. 2002. Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 24, 6, 844–847.
- BENNETT, E. P., AND McMILLAN, L. 2005. Video enhancement using per-pixel virtual exposures. *ACM Trans. Graphics* 24, 3, 845–852.
- BERNS, R. S. 1996. Methods for characterizing CRT displays. *Displays* 16, 4, 173–182.

- BLACK, M. J., SAPIRO, G., MARIMONT, D. H., AND HEEGER, D. 1998. Robust anisotropic diffusion. *Trans. Image Processing* 7, 3, 421–432.
- BOITARD, R., BOUATOUCH, K., COZOT, R., THOREAU, D., AND GRUSON, A. 2012. Temporal coherency for video tone mapping. In *Proc. SPIE 8499, Applications of Digital Image Processing XXXV*, 84990D–84990D–10.
- BOITARD, R., COZOT, R., THOREAU, D., AND BOUATOUCH, K. 2014. Zonal brightness coherency for video tone mapping. *Signal Processing: Image Communication* 29, 2, 229–246.
- CHEN, J., PARIS, S., AND DURAND, F. 2007. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graphics* 26, 3, 103:1–103:9.
- CHOUDHURY, P., AND TUMBLIN, J. 2003. The trilateral filter for high contrast images and meshes. In *Proc. Eurographics workshop on Rendering 14*, 186–196.
- DRAGO, F., MYSZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum* 22, 419–426.
- DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graphics* 21, 3, 257–266.
- EILERTSEN, G., WANAT, R., MANTIUK, R. K., AND UNGER, J. 2013. Evaluation of Tone Mapping Operators for HDR-Video. *Computer Graphics Forum* 32, 7, 275–284.
- FOI, A., TRIMECHE, M., KATKOVNIK, V., AND EGIAZARIAN, K. 2008. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Processing* 17, 10, 1737–1754.
- FROEHLICH, J., GRANDINETTI, S., EBERHARDT, B., WALTER, S., SCHILLING, A., AND BRENDL, H. 2014. Creating Cinematic Wide Gamut HDR-Video for the Evaluation of Tone Mapping Operators and HDR-Displays. In *Proc. SPIE 9023, Digital Photography X*, 90230X–90230X–10.
- IRAWAN, P., FERWERDA, J. A., AND MARSCHNER, S. R. 2005. Perceptually based tone mapping of high dynamic range image streams. In *Proc. Eurographics Conference on Rendering Techniques 16*, 231–242.
- KATKOVNIK, V., KATKOVNIK, V., EGIAZARIAN, K., AND ASTOLA, J. 2006. *Local Approximation Techniques in Signal and Image Processing*. Press Monographs. SPIE Press.
- KISER, C., REINHARD, E., TOCCI, M., AND TOCCI, N. 2012. Real time automated tone mapping system for HDR video. In *Proc. IEEE International Conference on Image Processing 19*, 2749–2752.
- KRONANDER, J., GUSTAVSON, S., BONNET, G., AND UNGER, J. 2013. Unified HDR reconstruction from raw CFA data. In *Proc. IEEE International Conference on Computational Photography 5*, 1–9.
- LEDDA, P., SANTOS, L. P., AND CHALMERS, A. 2004. A local model of eye adaptation for high dynamic range images. In *Proc. International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa 3*, 151–160.
- MAGGIONI, M., BORACCHI, G., FOI, A., AND EGIAZARIAN, K. 2012. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Trans. Image Processing* 21, 9, 3952–3966.
- MAI, Z., MANSOUR, H., MANTIUK, R., NASIOPOULOS, P., WARD, R., AND HEIDRICH, W. 2011. Optimizing a tone curve for backward-compatible high dynamic range image and video compression. *IEEE Trans. Image Processing* 20, 6, 1558 – 1571.
- MANTIUK, R., DALY, S., AND KEROFSKY, L. 2008. Display adaptive tone mapping. *ACM Trans. Graphics* 27, 3, 68:1–68:10.
- MANTIUK, R., MANTIUK, R. K., TOMASZEWSKA, A., AND HEIDRICH, W. 2009. Color correction for tone mapping. *Computer Graphics Forum* 28, 2, 193–202.
- MANTIUK, R., KIM, K. J., REMPEL, A. G., AND HEIDRICH, W. 2011. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graphics* 30, 4, 40:1–40:14.
- MILANFAR, P. 2013. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine* 30, 1, 106–128.
- PATTANAIK, S. N., TUMBLIN, J., YEE, H., AND GREENBERG, D. P. 2000. Time-dependent visual adaptation for fast realistic image display. In *Proc. SIGGRAPH 00*, Annual Conference Series, 47–54.
- PERONA, P., AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis Machine Intelligence* 12, 7, 629–639.
- REINHARD, E., AND DEVLIN, K. 2005. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Trans. Visualization and Computer Graphics* 11, 1, 13–24.
- TAKEDA, H., FARSIU, S., AND MILANFAR, P. 2007. Kernel regression for image processing and reconstruction. *IEEE Trans. Image Processing* 16, 2, 349–366.
- TAKEDA, H., FARSIU, S., AND MILANFAR, P. 2007. Higher order bilateral filters and their properties. *Proc. SPIE 6498, Computational Imaging V*, 64980S–64980S–9.
- TOCCI, M. D., KISER, C., TOCCI, N., AND SEN, P. 2011. A versatile hdr video production system. *ACM Trans. Graphics* 30, 4, 41:1–41:10.
- TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proc. International Conference on Computer Vision 6*, 839–846.
- VAN HATEREN, J. H. 2006. Encoding of high dynamic range video with a model of human cones. *ACM Trans. Graphics* 25, 1380–1399.
- WARD LARSON, G., RUSHMEIER, H., AND PIATKO, C. 1997. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Trans. Visualization and Computer Graphics* 3, 4, 291–306.
- YANG, Q. 2012. Recursive bilateral filtering. In *Proc. European Conference on Computer Vision 12*, 399–413.
- YOSHIZAWA, S., BELYAEV, A. G., AND YOKOTA, H. 2010. Fast gauss bilateral filtering. *Computer Graphics Forum* 29, 1, 60–74.