DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263
Reprinted from
Journal of Imaging Science and Technology® 60(2): 020404-1–020404-13, 2016.

Novel Real-Time Tone-Mapping Operator for Noisy Logarithmic CMOS Image Sensors

Jing Li, Orit Skorka, Kamal Ranaweera, and Dileepan Joseph
Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
E-mail: dil.joseph@ualberta.ca

Abstract. *Logarithmic CMOS image sensors are easily able, at video rates, to capture scenes where the dynamic range (DR) is high. However, tone mapping is required to output resulting images or videos to standard low-DR displays. This article proposes a new method, designed especially for logarithmic CMOS image sensors, which can suffer from temporal, and residual fixed pattern, noise. The novel tone mapping, a global operator based on histogram adjustment, uses a model of the camera noise to ensure that the mapping does not amplify the noise above a display threshold. Moreover, to reduce the likelihood of flickering, a temporal adaptation process is incorporated into the histogram calculation. Furthermore, to reduce complexity for real-time processing, a fixed-point implementation is designed for the proposed tone mapping. The novel operator and its fixed-point design are validated through offline and real-time experiments with a logarithmic CMOS image sensor.*

[DOI: 10.2352/J.ImagingSci.Technol.2016.60.2.020404]

INTRODUCTION

With each exposure, a logarithmic complementary metal-oxide-semiconductor (CMOS) image sensor can capture over six decades of luminance.[1] Such an imager easily produces videos where the scene dynamic range (DR) is high, unlike linear CMOS and charge coupled device (CCD) imagers, which can typically capture three decades.[2]

After capture, images and videos are often displayed. The conversion of scene luminance to display intensity is called tone mapping.[3] Unfortunately, the DR of standard displays is significantly lower than that which may occur in real scenes. Although limited high-DR (HDR) displays do exist, standard low-DR (LDR) displays continue to dominate the market.[4] Consequently, tone mapping continues to be important as a software workaround to the hardware limitations.

According to the literature,[5] tone-mapping operators (TMOs) may be divided into global and local operators, also called spatially invariant and spatially variant operators, respectively. Global TMOs apply to each pixel the same mapping, a function that usually depends on all pixels. Local TMOs apply to each pixel a different mapping, a function that usually depends on the pixel and its neighborhood.

Recently, Eilertsen et al.[6] "presented the first systematic survey and evaluation of TMOs for HDR-video content." Their HDR video was either computer generated or obtained by compositing multiple exposures taken with one or more linear image sensors. Their study also did not concern real-time processing, which would constrain TMOs much more than offline processing. Nevertheless, Eilertsen et al. provide insights relevant for the real-time tone mapping of logarithmic CMOS image sensors, the subject of this article.

Regarding "temporal artifacts," such as ghosting or flickering, that were "unacceptable" to human viewers, Eilertsen et al.[6] observe "that 'simpler' TMOs with *global* processing are in general significantly more robust compared to *local* operators." While Aydin et al.[4] have since reported a local TMO without such artifacts, their iterative operator, which employs optical flow, has not been developed for real-time processing. It is computationally very demanding.

Although recent authors have focussed on local TMOs, they concede that "some people have strong aesthetic preferences" for the "visual style" of global TMOs.[4] Indeed, in a pairwise comparison experiment,[6] global TMOs were most preferred, even over a local TMO that did not suffer from temporal artifacts. Among the three most-preferred TMOs, all global, two were based simply on histogram adjustment.

In the literature, TMOs are also differentiated by intent. Eilertsen et al.[6] listed three classes: visual system simulators; scene reproduction operators; and best subjective quality operators. Notably absent were TMOs specifically intended to suppress camera noise. Nonetheless, for logarithmic CMOS image sensors, temporal noise and residual fixed pattern noise (FPN), i.e., the spatial noise that remains after FPN correction, are the limiting factors in images and videos.[2]

In contrast to the literature, the real-time TMO introduced in this article is intended for noisy logarithmic CMOS image sensors. We propose a global TMO that is similar to the well-known histogram-adjustment TMO of Ward (Larson) et al.,[3] who modified the classic histogram equalization operator to simulate aspects of the human visual system (HVS). We show that a similar approach may be used to suppress camera noise instead, the impact of which we also model.

To prove that the proposed TMO works satisfactorily in real time, for a noisy logarithmic CMOS image sensor, we employ a calibrated imaging system that we previously designed, built, and tested.[7] For real-time performance, first we exploit the inherent efficiency of histogram-adjustment TMOs. While a global function is employed at video rate to

Received Nov. 6, 2015; accepted for publication Mar. 1, 2016; published online Feb. 3, 2016. Associate Editor: Marius Pedersen.
1062-3701/2016/60(2)/020404/13/$25.00

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

(a) (b)

Figure 1. Simple tone mapping of a bathroom image. This image from Ward (Larson),[8] where dynamic range (DR) is high, has been processed to demonstrate (a) underexposure and (b) overexposure issues, when simple tone mapping is used. Features in dark and bright parts cannot be displayed simultaneously.

map pixel tones, it is also updated using simple histogram calculations. Second, we incorporate the temporal adaptation process of Krawczyk et al.[5] into the histogram-adjustment framework. This prevents the TMO from changing too quickly, reducing the chance of flickering. Third, we produce a fixed-point design of the TMO, which is even more suited for real-time processing.

The Tone-Mapping Operator section introduces the proposed TMO, including a review of histogram equalization, the formulation of noise ceilings, and the incorporation of temporal adaptation. In the Fixed-Point Design section, a corresponding fixed-point design is provided. The Experimental Results section evaluates the proposed TMO through experiments.

TONE-MAPPING OPERATOR

The details of the proposed TMO are introduced in this section. First, histogram equalization is reviewed briefly. Next, its drawbacks in relation to camera noise are discussed, and a solution, which involves noise ceilings, is introduced. Finally, a first-order low-pass filter (LPF), to approximate the temporal adaption process of the human eye, is incorporated.

Histogram Equalization

We use a LogLuv-encoded TIFF image, taken from Ward (Larson)[8] and processed ourselves, to introduce our TMO. Scene luminance can be calculated from LogLuv-encoded TIFF images.[9] Figure 1 presents the image, a bathroom illuminated by a lamp, after "simple" tone mapping. Because our experimental results concern only a monochromatic image sensor, we restrict our attention to luminance only.

By "simple" tone mapping, we mean the application of white-point saturation, gamma correction, and scaling, as per the sRGB specification (color aside),[10] as follows:

$$w = \begin{cases} \text{round}(255(x/x_{\max})^{1/2.2}) & x < x_{\max}, \\ 255 & \textit{otherwise}. \end{cases} \quad (1)$$

Here, x is the scene luminance, in cd/m^2 for example, and w is the display intensity, assumed to range from 0 to 255 LSB, at each pixel. Simple tone mapping of an HDR scene, for an LDR display, may prove unsatisfactory, as demonstrated in Fig. 1 for two different white-point luminances, $x_{\max}$.

Histogram equalization, a classic image processing operation, may be viewed as a TMO.[9] It maps scene luminances to display intensities so as to achieve a uniform distribution of the latter. Because scene luminances tend not to have a uniform histogram, equalization shrinks

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

sparsely populated luminance regions, which enables DR compression. In densely populated luminance regions, DR expansion may occur.

For histogram equalization, the global TMO is simply:

$$w = 255\, P_X(x), \tag{2}$$

where P_X is the cumulative distribution function (CDF) of the scene luminance, treated as a random variable X, i.e.,

$$P_X(x) = P(X \leq x). \tag{3}$$

According to probability theory, the probability density function (PDF), p_X, is related to the CDF as follows:

$$p_X(x) = P'_X(x). \tag{4}$$

With images, the PDF may be approximated as follows:

$$p_X(x) \approx \frac{h_X(x)}{N\,\Delta x}, \tag{5}$$

where h_X is a histogram of scene luminances, N is the number of pixels, and Δx is the width of histogram bins.

Figure 2 gives histograms of the bathroom image before and after histogram equalization. The equalization process may be divided into two steps. First, a TMO is constructed based on the histogram of scene luminances. Second, the TMO is applied to each pixel, yielding the display intensities.

Nominally, the input of a TMO is luminance x. However, with a logarithmic CMOS image sensor, it may be convenient to operate on $\ln x$,[5] the logarithm of luminance. For a different reason, namely to model aspects of the HVS, Ward (Larson) et al.[3] also use $\ln x$, which they call "brightness," in their TMO, also based on histogram equalization.

Histogram equalization yields the same display intensities whether presented with scene luminances or brightnesses. Because e^x is monotonic, Eq. (3) may be rewritten as follows:

$$P_X(x) = P(e^{\ln X} \leq e^{\ln x}), \tag{6}$$

$$= P(\ln X \leq \ln x), \tag{7}$$

$$= P_{\ln X}(\ln x). \tag{8}$$

Because our logarithmic imaging systems measure brightness readily,[11] our proposed TMO focuses on brightness.

Figure 3(a) shows the bathroom image after histogram equalization, and Fig. 3(b) (top part) shows the corresponding TMO. Compared to simple tone mapping, more detail is visible at dark, bright, and intermediate tones. The computational complexity of global TMOs, in general, and histogram equalization, in particular, is low. Histogram-based methods are especially suited for processing video in real time.

Noise Ceilings

Unfortunately, logarithmic imagers may be ten times more noisy than linear imagers.[2] For the purposes of exposition, we simulated the noise of our previously reported logarithmic CMOS active pixel sensor (APS) array,[7] where root mean square (RMS) noise depends on brightness, and then mixed the noise with the bathroom image. (The Experimental Results section gives experimental results using the actual image sensor, a low-resolution prototype.)

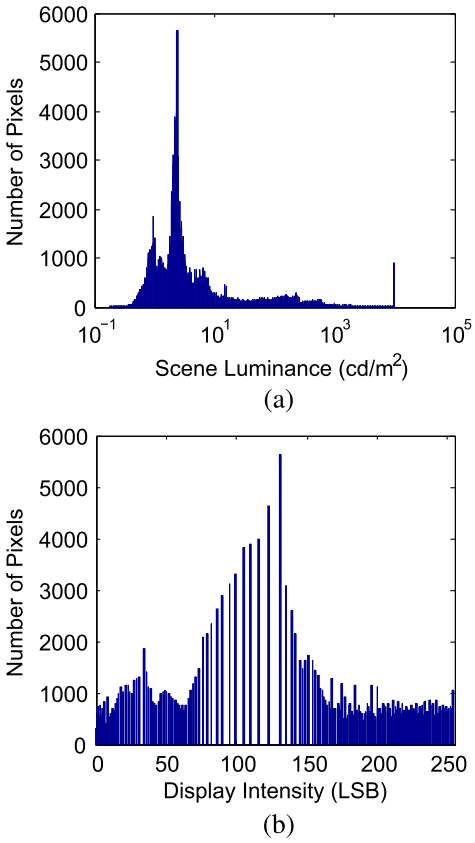
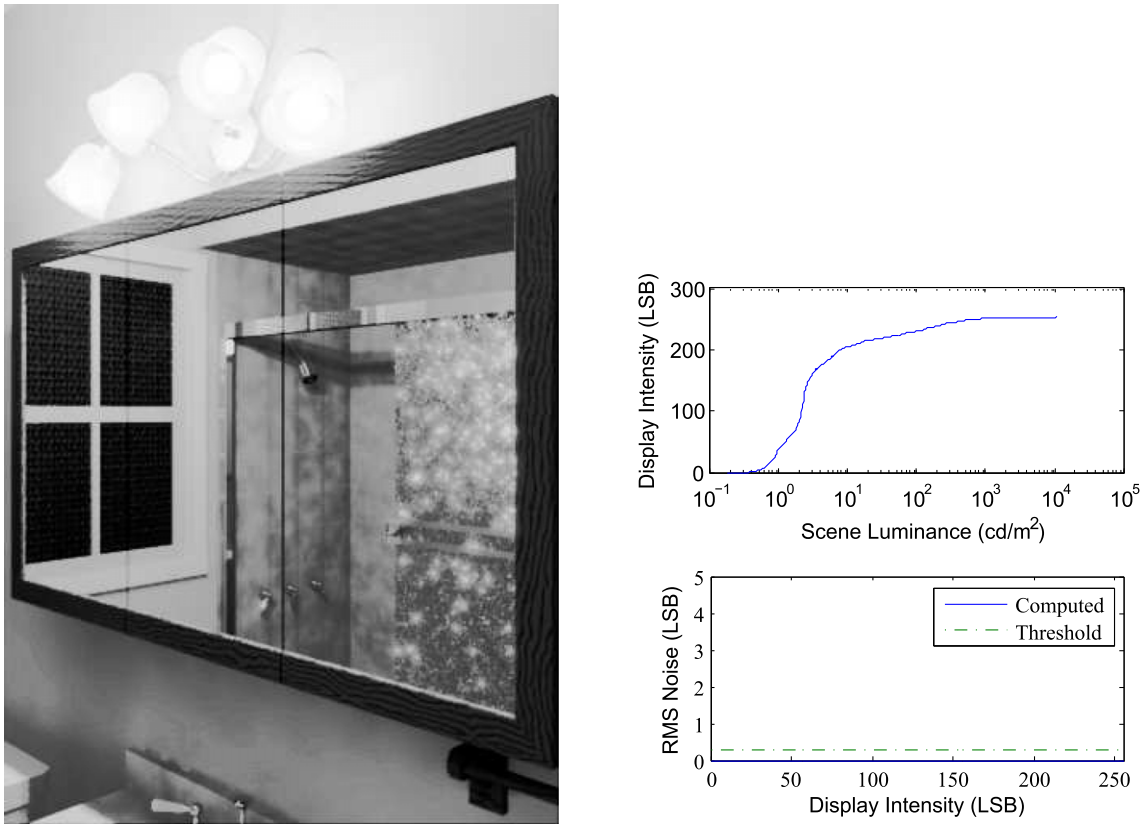


Figure 2. Histograms before/after equalization. (a) Scene luminances are nonuniformly distributed. (b) Display intensities are distributed more uniformly. Some nonuniformity remains because the input data is discrete.

Figure 4(a) presents the noisy bathroom image after histogram equalization. Clearly, the camera noise is visible. Fig. 4(b) (top part) gives the corresponding TMO, which is practically identical to the one, in Fig. 3(b) (top part), for the clear bathroom image. Therefore, with noisy logarithmic CMOS image sensors, a problem with the histogram equalization TMO is evidently its ignorance of camera noise.

Consequently, we derive a model to estimate the camera noise in the output, i.e., the display intensity, of a global TMO, e.g., histogram equalization. Figs. 3(b) and 4(b) (bottom parts) demonstrate use of this model for the clear and noisy bathroom images, respectively. With the model in hand, we then propose how to modify histogram equalization to suppress the camera noise, as shown in Figure 5. Our proposed TMO turns out to resemble the classic histogram-adjustment TMO of Ward (Larson) et al.,[3] which is noteworthy because their intention was to mimic aspects of the HVS. These are novel results, and are the contributions to imaging science of this multi-disciplinary (science and technology) work.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263



(a) (b)

Figure 3. Bathroom image after histogram equalization. (a) Dark to bright features are all visible with this tone mapping. (b) Although the mapping function (top) may amplify contrast, root mean square (RMS) noise (bottom) stays below the display threshold because camera noise is zero in this simulation.

We model a calibrated logarithmic imager as follows:

$$\ell_j = \ln x_j + \epsilon_j^c, \tag{9}$$

where j indexes pixels, ℓ_j is the measured brightness, x_j is the scene luminance, and ϵ_j^c is the camera noise, which we assume follows a zero-mean Gaussian distribution. Although RMS noise may depend on both scene brightness, $\ln x_j$, and pixel index, we assume dependence on the former only. Median RMS noise versus scene brightness, denoted $\sigma_c(\ln x)$, can be computed from calibration data. We use $\sigma_c(\ln x)$ to represent the RMS noise for all pixels in the image sensor.

With no loss of generality, we model a global TMO, where ℓ_j is the measured brightness, a continuous input, and w_j is the display intensity, a discrete output, as follows:

$$w_j = \text{round}(T(\ell_j)). \tag{10}$$

T is a continuous function made discrete by rounding, an operation equivalent to the addition of round-off error:

$$w_j = T(\ell_j) + \epsilon_j^d. \tag{11}$$

We treat the round-off error ϵ_j^d, which we call display noise, as a continuous and independent random variable with a uniform distribution from -0.5 to 0.5 LSB. Thus, the RMS display noise, which is independent of ℓ_j, is as follows:

$$\sigma_d = \frac{1}{\sqrt{12}}. \tag{12}$$

Through Eq. (9), measured brightness ℓ_j includes camera noise. Assuming the function T in Eq. (11) is differentiable, we use a first-order Taylor series to approximate the impact of the camera noise ϵ_j^c on the display intensity w_j as follows:

$$w_j = T(\ln x_j + \epsilon_j^c) + \epsilon_j^d, \tag{13}$$

$$\approx T(\ln x_j) + \epsilon_j^C + \epsilon_j^d, \tag{14}$$

where

$$\epsilon_j^C = T'(\ln x_j)\, \epsilon_j^c \tag{15}$$

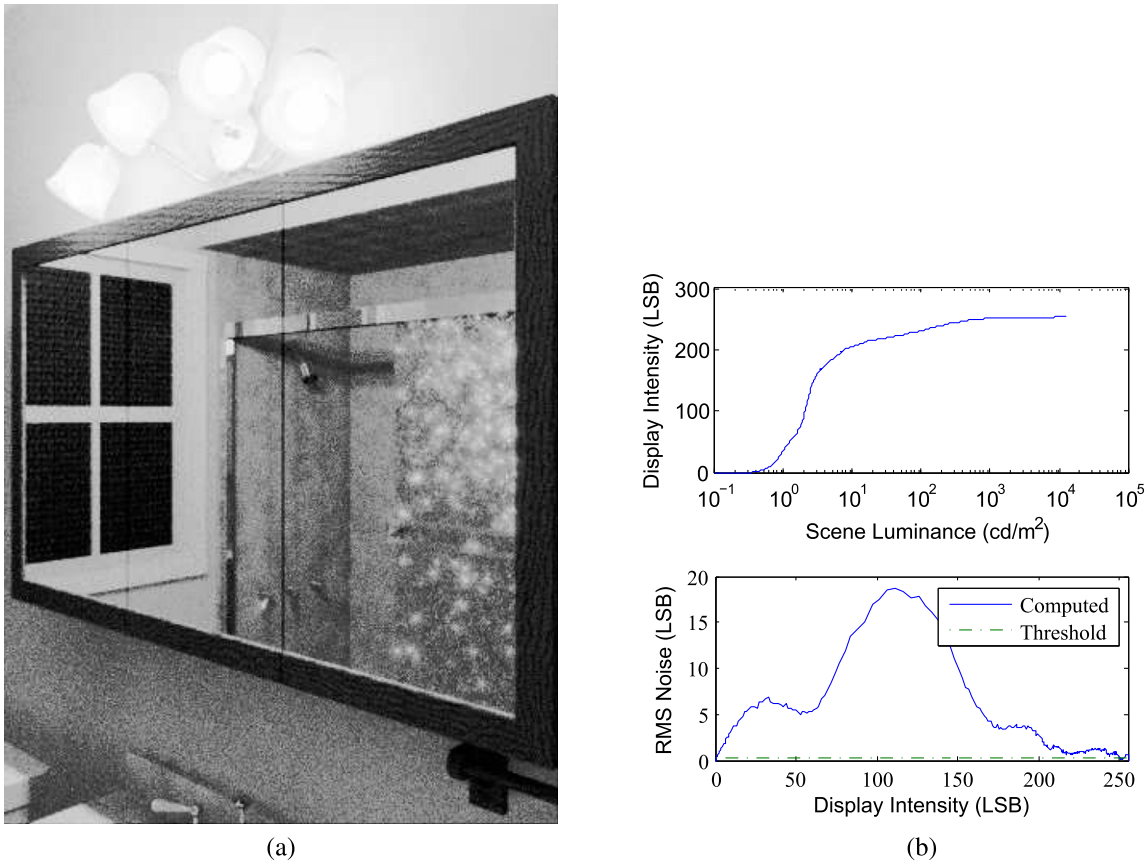
is the camera noise after tone mapping. Its RMS value, which depends on scene brightness $\ln x$, is therefore:

$$\sigma_C(\ln x) = |T'(\ln x)|\, \sigma_c(\ln x), \tag{16}$$

where absolute values are dropped for monotonically nondecreasing functions T (i.e., nonnegative derivatives T').

Thus, by virtue of discreteness, a global TMO always has some display noise, on the order of Eq. (12). If the camera

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263



(a) (b)

Figure 4. Histogram equalization with camera noise. (a) Although superior to simple tone mapping, camera noise is visible especially upon uniform textures. (b) The mapping function (top) is used to compute the RMS noise (bottom), which exceeds the display threshold substantially in this simulation.

noise after tone mapping is at or below this threshold, i.e.,

$$\sigma_C(\ln x) \le \frac{1}{\sqrt{12}}, \tag{17}$$

then it may be considered invisible, i.e., indistinguishable from the display noise background. Even if the camera noise after tone mapping is above this threshold, it may still be invisible because of other factors,[12] outside the scope of this article, that limit the visibility of various display intensities.

For histogram equalization, using Eqs. (2) and (8), the global TMO is proportional to the CDF of the scene brightness:

$$T(\ln x) = 255\, P_{\ln X}(\ln x). \tag{18}$$

Therefore, using Eqs. (4) and (16), the RMS camera noise after tone mapping is proportional to the corresponding PDF:

$$\sigma_C(\ln x) = 255\, P'_{\ln X}(\ln x)\, \sigma_c(\ln x), \tag{19}$$

$$= 255\, p_{\ln X}(\ln x)\, \sigma_c(\ln x). \tag{20}$$

Using Eq. (5), the PDF may be approximated by a histogram:

$$\sigma_C(\ln x) \approx 255 \frac{h(\ln x)}{N \Delta \ln x}\, \sigma_c(\ln x). \tag{21}$$

The RMS camera noise after histogram equalization is computed and plotted in Figs. 3 and 4 for the bathroom image without and with camera noise, respectively. As shown, histogram equalization can make camera noise visible.

Using Eqs. (17) and (21), we arrive at the following constraint on histogram bins for camera noise to be invisible:

$$h(\ln x) \le \frac{N \Delta \ln x}{255\sqrt{12}\, \sigma_c(\ln x)}. \tag{22}$$

Except with simulated results, such as the bathroom image, we do not know the scene brightness $\ln x$. Thus, with real cameras, we use the measured brightness ℓ instead:

$$h(\ell) \le \frac{N \Delta \ell}{255\sqrt{12}\, \sigma_c(\ell)}. \tag{23}$$

In Eq. (23), we specify a ceiling for each histogram bin to keep camera noise from exceeding a display threshold after tone mapping. Ward (Larson) et al.[3] arrived at a similar equation. However, their constraint uses a human contrast sensitivity function, instead of a camera noise function. With logarithmic CMOS image sensors, camera noise may be the limiting factor, which is why we have taken a different approach. Nevertheless, we employ Ward (Larson) et al.'s process to adjust the histogram, as needed, to meet the constraint.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

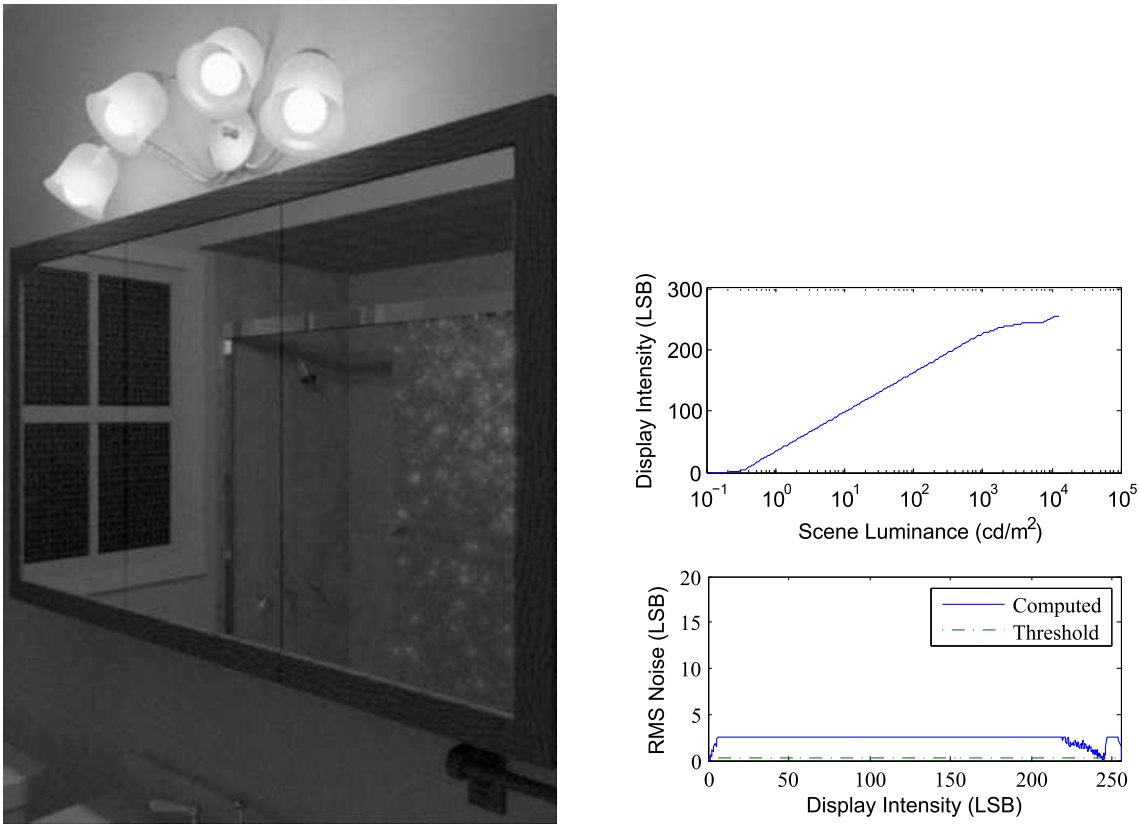


Figure 5. Bathroom image after proposed tone mapping. (a) Results coarsely approximate those of histogram equalization, but camera noise stays invisible. (b) In the mapping function (top), contrast amplification is constrained so that the RMS noise (bottom) stays relatively close to the display threshold.

When the pixel count in a histogram bin exceeds its ceiling, truncation, i.e., resetting the count to the ceiling, proves to be simple and reliable.[3] Although truncation does not guarantee that the constraint is met exactly, meeting it approximately suffices in practice. The truncated histogram, which represents a PDF, is then used to update the global TMO, which represents a CDF, by computing a cumulative sum. Finally, the global TMO is applied to each pixel of the original image. A flowchart of the process is depicted in Figure 6.

Fig. 5(a) shows the bathroom image after the proposed tone mapping. The mapping function and RMS camera noise are given in Fig. 5(b). Compared to Fig. 4(b), slopes of the mapping function are changed because the histogram is changed by truncation. In Fig. 5(b), the RMS camera noise is much lower but still exceeds the display threshold. It is difficult to see this low camera noise in Fig. 5(a).

In truth, the total number of pixels N in the histogram effectively decreases after truncation. According to Eq. (23), this changes the ceilings in turn. Iteration may be employed until some condition is satisfied.[3] For the bathroom image, Figure 7 plots the worst-case RMS camera noise, of all pixels, versus iteration. Performance does not improve much with iteration. However, employing iteration will seriously affect efficiency. Therefore, iteration is omitted in the proposed TMO.

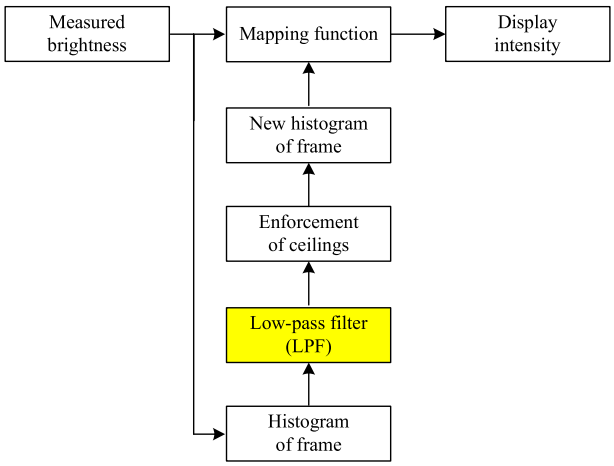


Figure 6. Flowchart of proposed tone mapping. The highlighted block is added for the case with temporal adaptation. Ceilings are calculated once beforehand. The mapping function is built from the new histogram.

Temporal Adaptation

A video is composed of a sequence of images. Therefore, a video TMO may be constructed by applying an image TMO to each frame. However, such an approach would be vulnerable to temporal noise, which may be significant with logarithmic CMOS image sensors.[7] To reduce the chance of

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

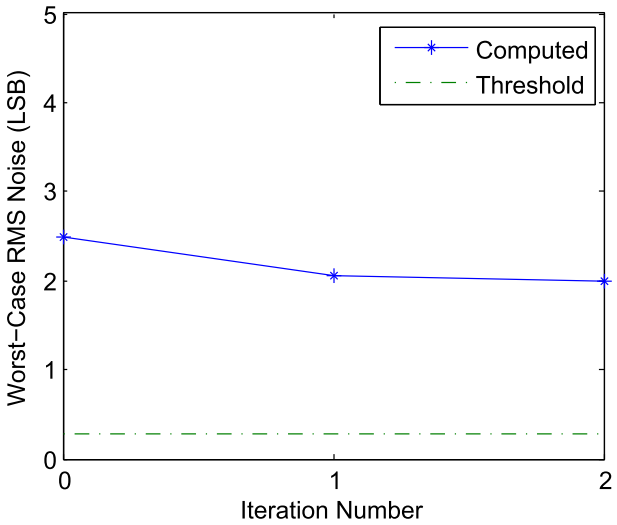


Figure 7. Performance versus number of iterations. Ideally, the RMS noise should be below the display threshold but this does not happen because of approximations. However, it proves low enough, even with zero iterations.

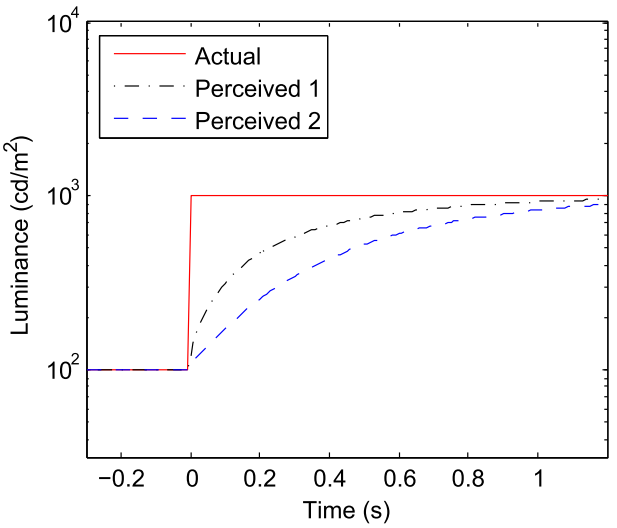


Figure 8. Temporal adaptation to luminance changes. In the "Perceived 1" case, a first-order low-pass filter (LPF) is applied to luminance. In the "Perceived 2" case, the LPF is applied to brightness (log luminance). Both approaches model temporal adaptation but the latter proves more convenient.

flickering, we modify our proposed TMO, incorporating an LPF.

Humans do not see each image over time independently. Our eyes take a period of time to adapt to luminance changes. Krawczyk et al.[5] modeled the temporal adaptation process of the human eye using an exponential decay function:

$$x_p(t) = x_p(t-T) + (x_s(t) - x_p(t-T))(1 - e^{\frac{-T}{\tau}}), \quad (24)$$

where x_s is the scene luminance, at time t, and x_p is the perceived luminance. T is a time interval, between measurements of the perceived luminance, and τ is a time constant.

For a digital imaging system, whose interval T between consecutive frames is constant, the above process in discrete time, using n to represent frame number, becomes:

$$x_p[n] = (1-\alpha)\, x_s[n] + \alpha\, x_p[n-1], \quad (25)$$

where

$$\alpha = e^{\frac{-T}{\tau}}. \quad (26)$$

Goodnight et al.[13] used this model on the logarithm of luminance, i.e., brightness, as a scale factor for each frame. They found that this approach gave acceptable results.

Figure 8 depicts the above temporal adaptation process, for a luminance step change, in two ways, i.e., with Eq. (25) applied to luminance and brightness, respectively. Initially, both the scene and perceived luminance are 10^2 cd/m^2. After the scene luminance changes abruptly to 10^3 cd/m^2, the perceived luminance adapts to the change. The time constant τ, in Eq. (26), was set to 0.4 s, the worst-case literature value.[14]

As shown in Fig. 8, temporal adaptation is readily modeled by passing scene luminance or brightness through a first-order LPF. While there are differences when the LPF is applied to luminance or brightness, both approaches are acceptable. With our proposed TMO, applying the LPF to brightness is more efficient because it is what the imaging system provides.

Instead of applying the first-order LPF to the image ℓ_j of scene brightness, we apply it to the histogram $h_s(\ell)$, denoted $h(\ell)$ previously, of scene brightness, as follows:

$$h_p(\ell)[n] = \text{round}(\beta\, h_s(\ell)[n] + \alpha\, h_p(\ell)[n-1]), \quad (27)$$

$$\beta = 1 - \alpha, \quad (28)$$

where $h_p(\ell)$ is the histogram of perceived brightness. In general, the number of histogram bins is fewer than the number of image pixels, which simplifies computation. Also, because an image histogram is meant to represent integer counts of pixels, a rounding operation is introduced in Eq. (27). Otherwise, the LPF outputs could have fractional parts.

Fig. 6 presents the completed process of the proposed tone mapping, which employs both noise ceilings and an LPF. The modified histogram $h_p(\ell)$ is compared to the ceilings. A mapping function for the current frame in the video is derived from the modified histogram after truncation.

FIXED-POINT DESIGN

The computational complexity of the proposed TMO is low because of the inherent simplicity of histogram equalization and (noniterative) enforcement of noise ceilings. These benefits are shared with Ward (Larson) et al.'s[3] TMO, one reason why it served as an inspiration. However, unlike Ward (Larson) et al., we offer a fixed-point design for our TMO, which enables even lower computational complexity.

In the proposed TMO, histogram counting and ceiling enforcement are intrinsically fixed point. What remains is to design fixed-point solutions to compute the mapping

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

function and to apply the temporal adaptation. This section provides the solutions, which are validated in the Experimental Results section.

Mapping Function

Temporal adaptation aside, the proposed TMO includes the following steps. Initially, using Eq. (23), ceilings are computed. They are constant functions of camera and display noise. For each image, a histogram computation counts the number of pixels in predefined bins of brightnesses. In some bins, the pixel count is truncated because it exceeds the corresponding noise ceiling. This leads to a new histogram.

The above steps readily compute in real time with fixed-point operations when measured brightnesses are fixed point, which we assume to be the case. Therefore, the only problem left is how to build a mapping function from the new histogram. The mapping function approximates a CDF and is built by accumulating the new histogram, as follows:

$$w(\ell) = \text{round}\left(\frac{255\sum_{k=\ell_{\min}}^{\ell} h_{\text{new}}(k)}{N_{\text{new}}}\right). \tag{29}$$

Therefore, division is needed in this step. In a fixed-point design, division should be avoided because it is difficult to implement and/or it suffers from low precision. This encourages us to consider an alternative formulation.

The total pixel count N_{new} is determined from the new histogram after truncation. It varies from frame to frame because the new histogram is different each time. However, the range of the total pixel count depends on the noise ceilings, in Eq. (23), which are constant. The maximum value $N_{\max}$ occurs when each bin count equals its noise ceiling. In contrast, the minimum value is difficult to determine. Therefore, we simply take 1 as the minimum value (it will be less than the true minimum). Consequently, the range of N_{new} is:

$$1 \le N_{\text{new}} \le N_{\max}. \tag{30}$$

Revisiting Eq. (29), it may be rewritten as follows:

$$w(\ell) = \text{round}\left(a\sum_{k=\ell_{\min}}^{\ell} h_{\text{new}}(k)\right), \tag{31}$$

$$a = \frac{255}{N_{\text{new}}}. \tag{32}$$

Therefore, parameter a is a function of N_{new}. As described above, the range of N_{new} is constant for a specific image sensor. As a result, we can compute a for all possible N_{new} beforehand. Using Eqs. (30) and (32), the range of a is:

$$\frac{255}{N_{\max}} \le a \le 255. \tag{33}$$

Fixed-point numbers may be stored as integers. Parameter a is scaled by a factor 2^{s_a} and rounded, as follows:

$$A = \text{round}(2^{s_a}a). \tag{34}$$

Therefore, the unscaled round-off error is:

$$\Delta a = \frac{A - 2^{s_a}a}{2^{s_a}}, \tag{35}$$

where the range is:

$$-\frac{1}{2^{s_a+1}} \le \Delta a \le \frac{1}{2^{s_a+1}}. \tag{36}$$

The error range is controlled by the scaling factor, which is chosen based on precision requirements. Once chosen, all possible parameters A are stored in a look-up table (LUT), addressable by N_{new}, using a fixed-point format.

To build a mapping function, the A corresponding to N_{new} is read out from the LUT. Afterward, the cumulative sum of the histogram $\sum_k h_{\text{new}}(k)$ is multiplied with A, bit shifted using s_a, and rounded to get the display intensity w:

$$w(\ell) = \text{round}\left(2^{-s_a}A\sum_{k=\ell_{\min}}^{\ell} h_{\text{new}}(k)\right). \tag{37}$$

Using an LUT in this manner is a good approach for the fixed-point design of the proposed TMO because fixed-point division is avoided while sufficient precision can be achieved.

Finally, because ℓ and w are both fixed point, the mapping function in Eq. (37) is enumerated and stored in another LUT, which is used for real-time tone mapping. The error brought by the round operator in Eq. (37) is the display noise ϵ^d introduced previously. Therefore, it exists with both the floating-point and fixed-point designs, and has already been considered.

Temporal Adaptation

In the proposed TMO, a first-order LPF is used to approximate temporal adaptation of the human eye. This LPF needs to be realized with fixed-point operations. The same scaling method used in the previous section can be used here.

The first-order LPF is defined by Eq. (27). The two coefficients α and β, which are defined in Eqs. (26) and (28), respectively, are calculated based on a time constant. Both coefficients range from 0 to 1. First, they are scaled and rounded:

$$\alpha' = \text{round}(2^{s_h}\alpha), \tag{38}$$

$$\beta' = \text{round}(2^{s_h}\beta). \tag{39}$$

Therefore, the unscaled round-off error for α is:

$$\Delta\alpha = \frac{\alpha' - 2^{s_h}\alpha}{2^{s_h}}. \tag{40}$$

And the range is:

$$-\frac{1}{2^{s_h+1}} \le \Delta\alpha \le \frac{1}{2^{s_h+1}}. \tag{41}$$

The case for β is similar.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

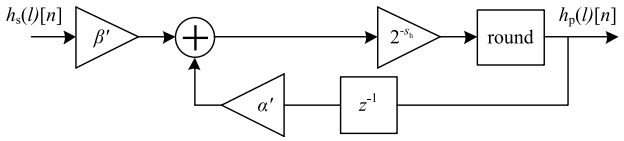


Figure 9. Fixed-point design of temporal adaptation. Coefficients of the LPF, applied to the histogram of scene brightness $h_s(\ell)$ at time n, are scaled and rounded, yielding α' and β'. Bit shifting and further rounding are used, in real time, to compute the histogram of perceived brightness $h_p(\ell)$.

To implement the LPF, a left shift of the binary point is needed to scale the result back to the original magnitude:

$$h_p(\ell)[n] = \text{round}(2^{-s_h}\{\beta' h_s(\ell)[n] + \alpha' h_p(\ell)[n-1]\}). \quad (42)$$

The round operation removes the fractional part. As with Eq. (37), the error caused by the round operation is ignored; it exists with both the floating-point and fixed-point designs. The design of the fixed-point LPF is summarized in Figure 9.

EXPERIMENTAL RESULTS

The proposed TMO was programmed in MATLAB and C/C++. Experiments were done with a prototype imaging system, employing a logarithmic CMOS image sensor.

Notwithstanding temporal adaptation, it suffices to test the proposed TMO using still images. On the other hand, the proposed TMO with temporal adaptation needs to be tested using videos. MATLAB was first used to process captured images offline. Afterward, the proposed TMO was programmed in C and embedded in a Visual C++ framework. The C/C++ tests focussed on real-time performance evaluation.

The Imaging System section summarizes the imaging system used for experiments. The Offline Experiments section discusses the MATLAB experiments. C/C++ experiments are discussed in the Real-Time Experiments section.

Imaging System

The logarithmic CMOS APS array used in the experiments was fabricated in a 0.35 µm CMOS process. This image sensor, reported previously,[7] has 90 × 120 pixels with a 10 µm pitch. The schematic of a single pixel is shown in Figure 10. The figure also shows the path a signal travels, from pixel to column to array level, until it arrives at an analog-to-digital converter (ADC) line, connected to an off-chip ADC.

The photodiode current, I_{PD}, is composed of the dark current, I_{dk}, and the photocurrent, I_{ph}, which is proportional to scene luminance. Transistors N_1 and N_2 are "diode-connected" transistors. Therefore, they can operate either in sub-threshold or in saturation. Photocurrents of at least 1 µA are needed for these transistors to operate in saturation, a level that is not likely to be reached with natural scenes.

Logarithmic sensors are nonintegrating ones. Therefore, in general, they do not require a reset signal. Nonetheless, a reset transistor, P_1, is included in the pixel in order to shorten transient response times. Logarithmic pixels may suffer from long transition times when the scene brightness drops abruptly because a small photocurrent needs to charge the photodiode capacitance, C_{PD}. To overcome this scenario, at the end of every frame, P_1 is activated through a reset signal, which quickly charges C_{PD} to a voltage close to VDD.

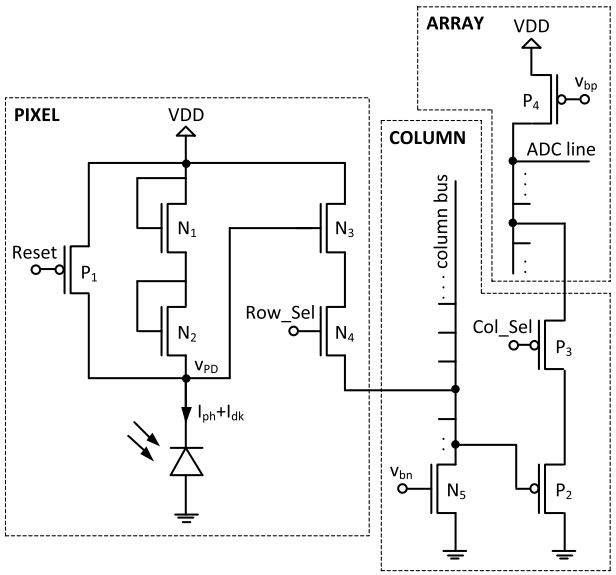


Figure 10. Logarithmic active pixel sensor (APS). Each pixel in the APS array includes a logarithmic sensor, switches, and a source follower (SF). When a pixel is selected, the sensor output V_{PD} is copied to a board-level (off-chip) analog-to-digital converter (ADC) through pixel and column-level SFs.

Transistor N_3 is a source follower (SF) and transistor N_4 is a switch that is controlled by a row-select signal. Transistor N_5 is placed at column level and determines the current of the SF stage. Its current is controlled by the bias voltage V_{bn}. Similarly, transistor P_2 is an SF and transistor P_3 is a switch activated by a column-select signal, while P_4 is a voltage-controlled current source, biased by V_{bp}. The drain of P_4 is the output line; it is connected to the input of an off-chip ADC.

The imaging system includes a custom-made camera body, a lens (Fujinon CF25HA-1), a custom-designed board that accommodates the image sensor, and a QuickUSB board with an Altera Cyclone II field-programmable gate array (FPGA). The two boards are powered through two USB ports of a PC (Intel Pentium D with a 2.80 GHz CPU and 2 GB of DDR2 memory), where the port connected to the QuickUSB board is also used for data transfer. An ADC (Texas Instruments ADS8411) is included on the same board with the image sensor. Control signals are sent from the FPGA to activate the image sensor and the ADC. After conversion, the FPGA reads the data generated by the ADC and sends it to the PC for data processing and display.

Offline Experiments

Before the experiments, the image sensor was calibrated, meaning that Eq. (9) can be used to model captured images. Moreover, the RMS noise of each pixel in the image sensor

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

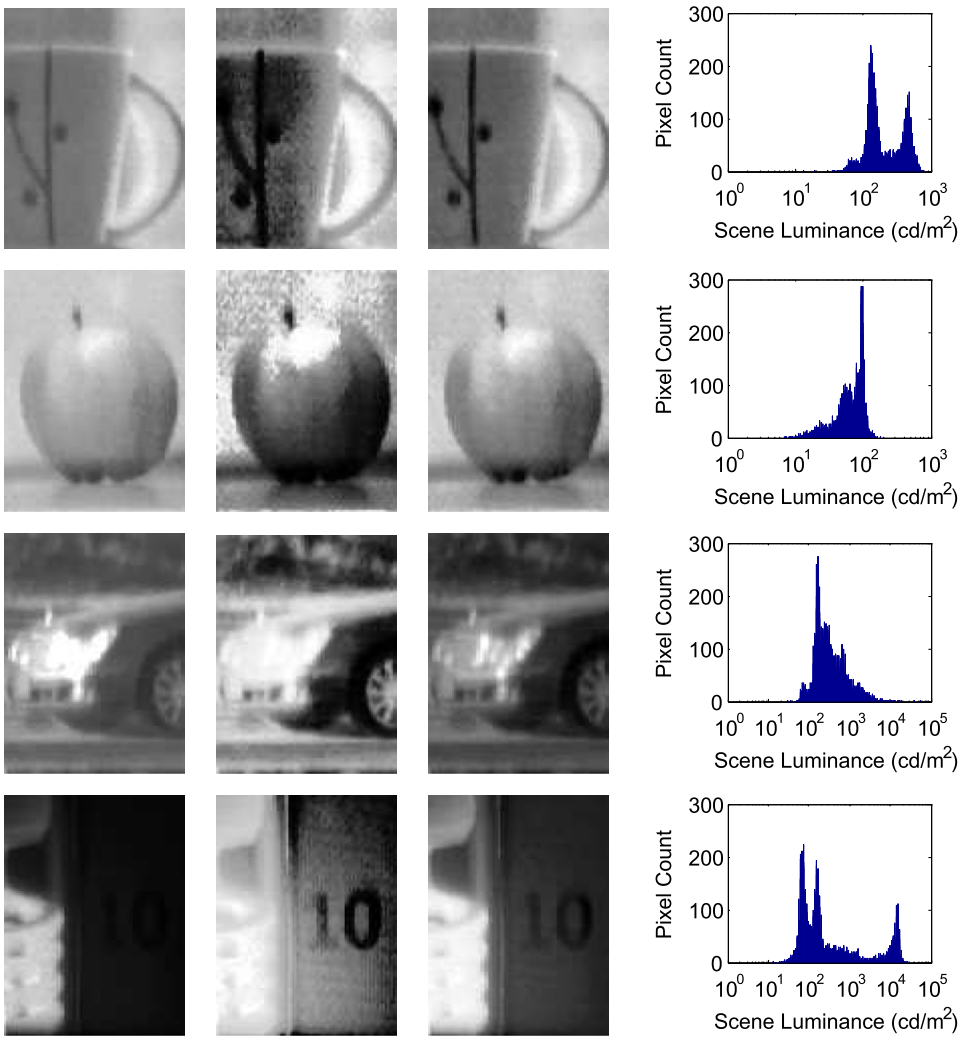


Figure 11. Images after various tone mappings. Low (top two) and high (bottom two) DR examples after (a) simple tone mapping, (b) histogram equalization (no ceilings), and (c) the proposed tone mapping (floating point), respectively. (d) Scene histograms, computed from the original images, are also shown.

was computed. The median RMS noise, a characteristic of the image sensor, was used for the proposed tone mapping.

Images captured by the prototype imaging system were processed in MATLAB using the proposed TMO. Because these tests focus on offline performance for still images, temporal adaptation is ignored. Figure 11 compares four images captured by our logarithmic image sensor after simple tone mapping, histogram equalization, and the proposed tone mapping, respectively. The top two examples involve LDR scenes while the bottom two examples involve HDR scenes.

Compared to simple tone mapping, histogram equalization provides a wider subjective DR. However, magnified noise degrades the quality of images while the bright parts may be overexposed. The proposed tone mapping performs better for both LDR and HDR images. For the LDR images (mug and apple), contrast is enhanced but not too much. For the HDR image of a car, a headlight that was overexposed with simple tone mapping becomes visible. In the last example, a separator stands in the middle. At left, a bulb shines while a "10" printed on cardboard is on the right. The cardboard is underexposed with simple tone mapping. It becomes visible with the proposed tone mapping while the ring of the bulb can still be seen. Contrast in the dark part is kept low because camera noise is relatively worse in dim lighting.

Testing of the proposed fixed-point tone mapping was also done. Four captured images are mapped, which include two LDR images and two HDR images. The performance is evaluated through a comparison between the fixed-point and floating-point designs. For the proposed tone mapping without temporal adaptation, this comparison only depends on the wordlength of parameter A in the CDF computation.

Multiple images were mapped by the fixed-point design adopting different wordlengths. RMS differences of images mapped by the floating-point and fixed-point designs were computed. These differences, versus image and wordlength, are shown in Figure 12. As shown, RMS differences are lower

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

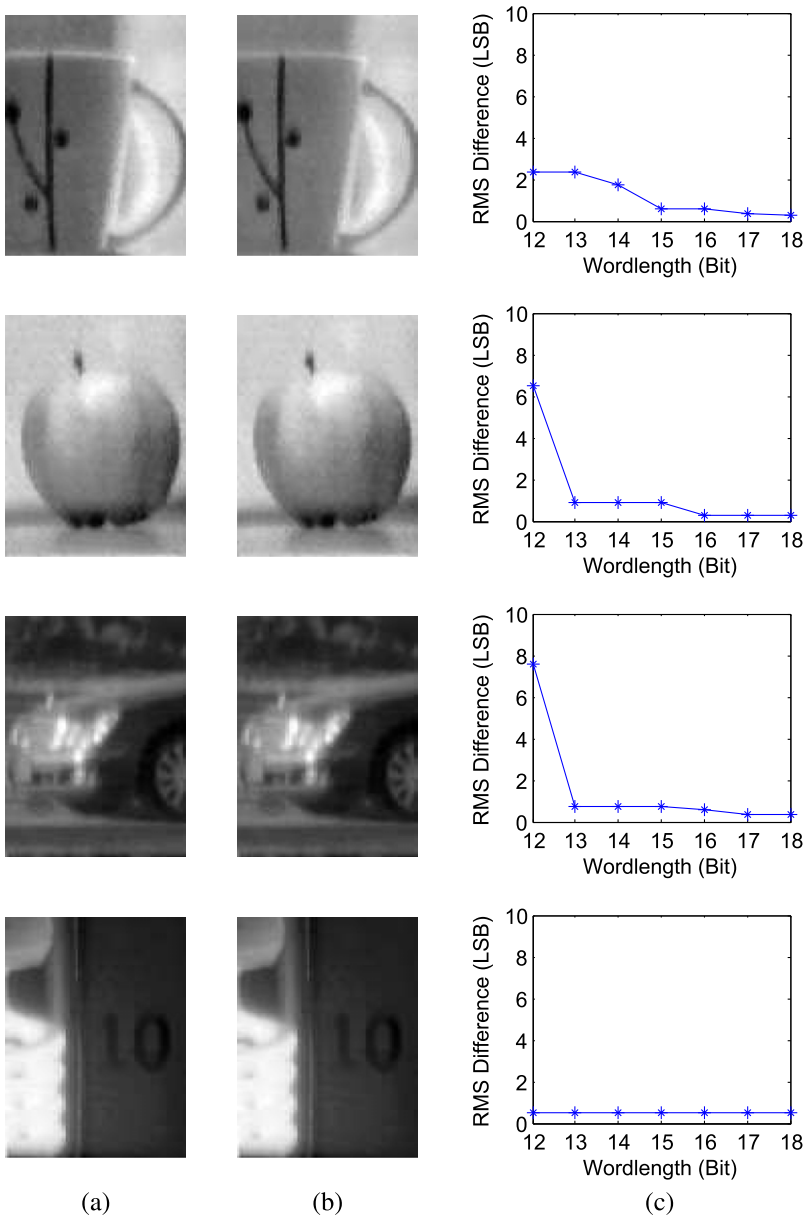


Figure 12. Tone-mapping implementations compared. Output images for (a) floating-point and (b) fixed-point versions of the proposed tone mapping are virtually identical, given sufficient wordlength of a key parameter (*A*). (c) RMS differences of images, versus wordlength, show that 16 bits suffices.

than or close to 1 LSB with a 15-bit wordlength. Considering storage efficiency and C/C++ data types, a 16-bit wordlength is a good choice and offers a little extra precision.

For temporal adaptation, an LPF is also employed. Several 200-frame videos were recorded and processed using floating and fixed-point implementations of the proposed TMO. Examples are shown in Figure 13. Using offline processing, implementations were compared. RMS differences of all video pixels, versus wordlength of LPF coefficients, are shown in Figure 14 for a video that involved an abrupt DR change.

As shown in Fig. 14, RMS differences are lower than 1 LSB with a 7-bit wordlength. Since there are only two coefficients in the LPF, storage efficiency is not as important a factor here as with the 16-bit LUT adopted for parameter *A*. This means wordlength in terms of an integer number of bytes is not critical for the LPF parameters. Nevertheless, considering readily available data types in C/C++, an 8-bit wordlength is convenient, and is adopted for the fixed-point LPF.

MATLAB tests helped us to determine wordlengths of the fixed-point design for our prototype image sensor. The final designs were programmed in C and implemented in a Visual C++ framework to test performance in real time.

Real-time Experiments

Fig. 13 depicts selected frames of the same video processed separately using floating-point and fixed-point implementations. The fixed-point design (C/C++) was computed

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

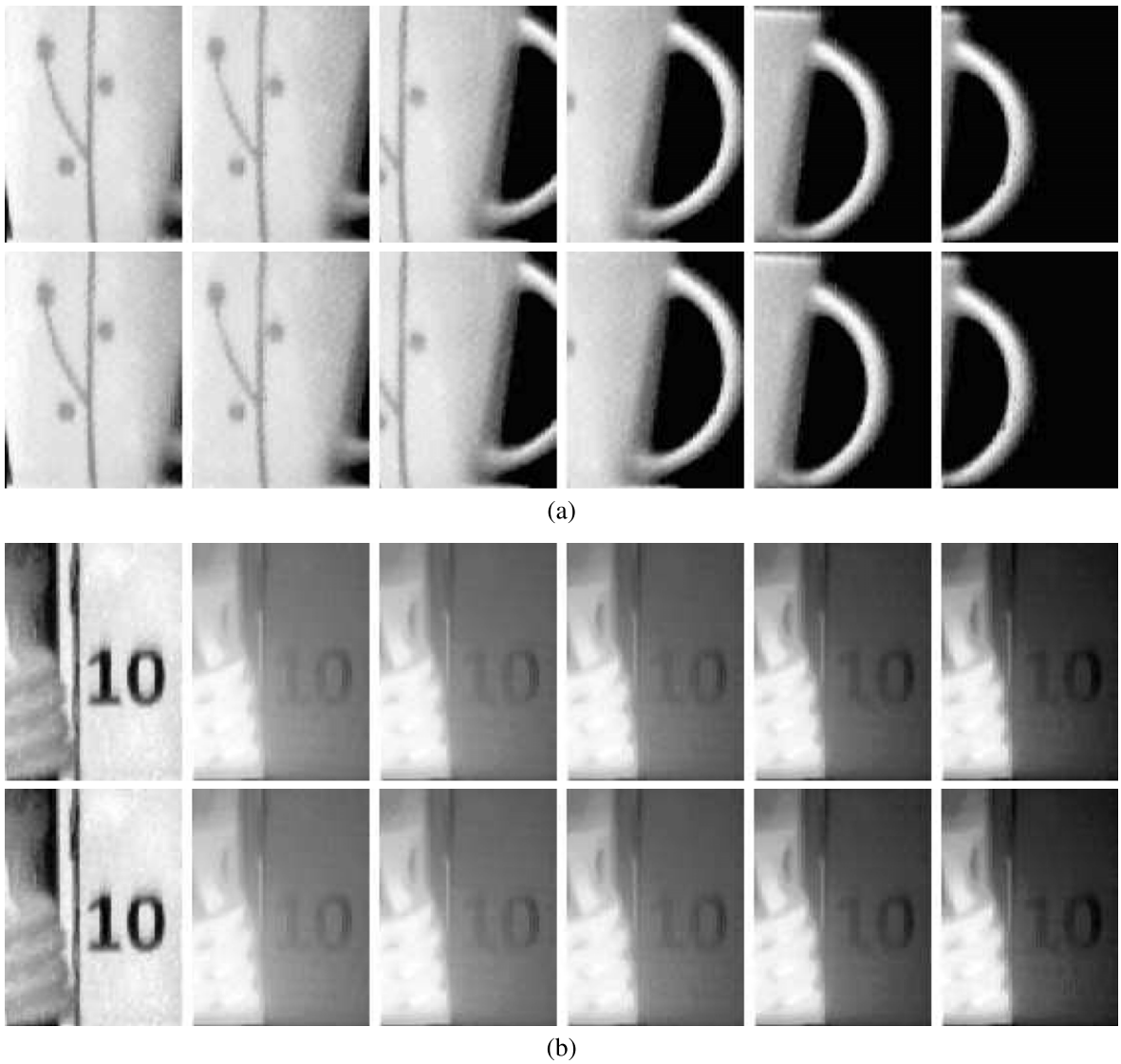


Figure 13. Proposed tone mapping on video frames. Selected frames from processed videos exhibiting (a) motion and (b) a DR step change. While there were 200 frames over a 4 s period, every 10th frame is shown over a 1 s period. For each example, top and bottom frames use floating-point and fixed-point implementations, respectively, computed offline using MATLAB and in real time using C/C++, respectively. Results are virtually indistinguishable.

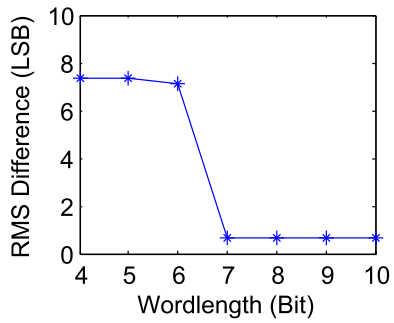


Figure 14. Fixed-point LPF parameter wordlength. The RMS difference between all pixels of a 200-frame video, processed using floating-point and fixed-point implementations of the proposed tone mapping, is small when the wordlength of the LPF coefficients (α' and β') is sufficiently large.

in real time, while the floating-point design (MATLAB) was computed offline using logged data. By comparing the corresponding frames in Fig. 13, any differences that exist are essentially imperceptible. The result demonstrates that the fixed-point design works in real time with good performance.

After implementing the fixed-point design, the prototype imaging system was able to operate 11% faster, i.e., at 50 fps instead of 45 fps. Although the floating-point performance of the desktop PC prevented greater improvement in frame rate, the value of the fixed-point design cannot be underrated. Its low computational complexity makes it feasible for an FPGA implementation, and even on-chip integration with the image sensor. In addition, a fixed-point design enables real-time tone mapping with lower power

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-263

consumption, which is especially important for portable (battery-powered) devices.

CONCLUSION

Logarithmic CMOS image sensors capture HDR images readily at video rates, but can suffer from significant noise. Although HDR displays exist, and are a direct approach for viewing HDR images and videos, they are unlikely to displace standard LDR displays in the near future. Therefore, TMOs are required to compress the DR of source images.

Literature TMOs mainly divide into two categories: global and local. For global operators, each pixel in an image is mapped through a common function. The mapping functions for local operators can vary with the pixel neighborhood. Although local operators can have better visual performance, global operators tend to be computationally more efficient, while also having good performance. Literature TMOs, in addition, mostly focus on still images and are not designed specifically to suppress camera noise. Finally, although video TMOs exist, real-time considerations are often ignored.

Although applying an image TMO to each frame results in a video TMO, this approach is simplistic because video is not perceived as an independent sequence of images. To minimize the chance of flickering due to time-varying camera noise, a TMO should adapt smoothly to luminance changes over time. Such temporal adaptation would also smooth a DR step change. For real-time applications, it is important for the computational complexity of the adaptation to be low.

This article introduced a new global TMO, especially suited for real-time processing and noisy logarithmic CMOS image sensors. The TMO enforces ceilings on contrast enhancement, based on histogram equalization, using simple models of camera and display noise. For video applications, a first-order LPF is applied to brightness histograms, resulting in smooth changes to the TMO over time. Although intrinsically efficient, the proposed TMO is made even more efficient via a fixed-point design. Finally, the proposed TMO was validated through experiments with a prototype imaging system.

ACKNOWLEDGMENTS

The authors thank the Natural Sciences and Engineering Research Council, Canada, and Alberta Ingenuity (now Alberta Innovates—Technology Futures) for their financial support. They also thank CMC Microsystems and IMRIS, especially Dr. Mark Alexiuk, for in-kind contributions.

REFERENCES

[1] B. Hoefflinger and V. Schneider, *High-Dynamic-Range (HDR) Vision*, ser. Advanced Microelectronics (Springer, Berlin Heidelberg, 2007), Vol. 26, pp. 13–56, ch. 2.
[2] O. Skorka and D. Joseph, "Toward a digital camera to rival the human eye," J. Electron. Imaging **20**, 1–18 (2011).
[3] G. Ward Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," IEEE Trans. Vis. Comput. Graphics **3**, 291–306 (1997).
[4] T. O. Aydin, N. Stefanoski, S. Croci, M. Gross, and A. Smolic, "Temporally coherent local tone mapping of HDR video," ACM Trans. Graph. **33**, 1–196 (2014).
[5] G. Krawczyk, K. Myszkowski, and D. Brosch, *High-Dynamic-Range (HDR) Vision*, ser. Advanced Microelectronics (Springer, Berlin Heidelberg, 2007), Vol. 26, pp. 148–178, ch. 11.
[6] G. Eilertsen, R. Wanat, R. K. Mantiuk, and J. Unger, "Evaluation of Tone Mapping Operators for HDR-Video," *Computer Graphics Forum* (Wiley Online Library, 2013), Vol. 32, pp. 275–284.
[7] O. Skorka, A. Mahmoodi, J. Li, and D. Joseph, "Bio-inspired design: nonlinear digital pixels for multiple-tier processes," Proc. SPIE **8691**, 1–12 (2013).
[8] G. Ward, 1997 Computer Graphics Renderings, Bathroom: Nighttime (mir) after simulation. [Online]. Available: www.anyhere.com/gward/pixformat/tiffluvrend.html.
[9] G. Ward Larson, "LogLuv encoding for full gamut, high-dynamic range images," J. Graph. Tools **3**, 15–31 (1998).
[10] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet—sRGB," World Wide Web Consortium, Technical Report, Nov. 1996. [Online]. Available: www.w3.org/Graphics/Color/sRGB.html.
[11] J. Li, A. Mahmoodi, and D. Joseph, "Using polynomials to simplify fixed pattern noise and photometric correction of logarithmic CMOS image sensors," Sensors **15**, 26,331–26,352 (2015).
[12] R. Mantiuk, S. Daly, and L. Kerofsky, "Display adaptive tone mapping," ACM Trans. Graph. **27**, 1–68 (2008).
[13] N. Goodnight, R. Wang, C. Woolley, and G. Humphreys, "Interactive time-dependent tone mapping using programmable graphics hardware," *Eurographics Symposium on Rendering*, 14th Eurographics Workshop on Rendering (2003), pp. 26–37.
[14] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," ACM Trans. Graph. **21**, 257–266 (2002).