

Article

Model-Based Design of Contrast-Limited Histogram Equalization for Low-Complexity, High-Speed, and Low-Power Tone-Mapping Operation

Wei Dong [†], Maikon Nascimento [‡] and Dileepan Joseph ^{*} 

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada; wd4@ualberta.ca (W.D.); maikon@ualberta.ca (M.N.)

^{*} Correspondence: dil.joseph@ualberta.ca; Tel.: +1-780-492-9164

[†] Current address: Independent Researcher, Shenzhen 518100, China.

[‡] Current address: Independent Researcher, 37073 Göttingen, Germany.

Abstract: Imaging applications involving outdoor scenes and fast motion require sensing and processing of high-dynamic-range images at video rates. In turn, image signal processing pipelines that serve low-dynamic-range displays require tone mapping operators (TMOs). For high-speed and low-power applications with low-cost field-programmable gate arrays (FPGAs), global TMOs that employ contrast-limited histogram equalization prove ideal. To develop such TMOs, this work proposes a MATLAB–Simulink–Vivado design flow. A realized design capable of megapixel video rates using milliwatts of power requires only a fraction of the resources available in the lowest-cost Artix-7 device from Xilinx (now Advanced Micro Devices). Unlike histogram-based TMO approaches for non-linear sensors in the literature, this work exploits Simulink modeling to reduce the total required FPGA memory by orders of magnitude with minimal impact on video output. After refactoring an approach from the literature that incorporates two subsystems (Base Histograms and Tone Mapping) to one incorporating four subsystems (Scene Histogram, Perceived Histogram, Tone Function, and Global Mapping), memory is exponentially reduced by introducing a fifth subsystem (Interpolation). As a crucial stepping stone between MATLAB algorithm abstraction and Vivado circuit realization, the Simulink modeling facilitated a bit-true design flow.

Keywords: image signal processing; high dynamic range; tone mapping operators; Simulink modeling; field-programmable gate array circuits; bit-true design



Academic Editors: Nabil Yassine, Abdallah Tammam and Khaled Hayatleh

Received: 31 March 2025

Revised: 23 May 2025

Accepted: 24 May 2025

Published: 13 June 2025

Citation: Dong, W.; Nascimento, M.; Joseph, D. Model-Based Design of Contrast-Limited Histogram Equalization for Low-Complexity, High-Speed, and Low-Power Tone-Mapping Operation. *Electronics* **2025**, *14*, 2416. <https://doi.org/10.3390/electronics14122416>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automotive, smartphone, and other applications have motivated research into high-dynamic-range (HDR) image capture and processing [1,2]. Takayanagi and Kuroda [1] reviewed technologies to capture HDR images, which represent scene luminance from dim shadows to bright highlights without saturation. Such technologies include nonlinear response HDR, linear response HDR with multiple exposures, linear response HDR with single exposures, frame oversampling and photon counting, and linear response HDR with multiple photodiodes. Hajisharif et al. [3] presented a linear response HDR with single-exposures apparatus, where a complementary metal-oxide-semiconductor (CMOS) active pixel sensor (APS) array uses alternating pixels with low and high gains. Compared to multiple-exposure methods, single exposures reduce motion artifacts in video applications. Brunetti and Choubey [2] elaborated on a recent development with a nonlinear response

apparatus. With their experimental results, they demonstrated a logarithmic (log) APS capable of over 160 decibels (dB) of dynamic range with single exposures at video rates.

Regarding image processing that back-ends image capture, there are many possible stages. This paper focuses on ones called tone-mapping operators (TMOs), especially for nonlinear response HDR sensors. As Khan et al. [4] explain, because low-dynamic-range (LDR) images are more suitable for conventional display devices, a TMO serves to map HDR images to LDR images while preserving salient features.

Khan et al. [4] classified TMOs into only two kinds. Global TMOs apply one function to each pixel of an image, although the function may vary with time. By allowing a spatially varying mapping, local TMOs can enhance local details. Like Khan et al. [4], Völgyes et al. [5] favour a histogram-based TMO. Whereas the former adopt a local method for medical X-ray images, the latter prefer a global method, inspired by classic work on contrast-limited histogram equalization from Larson et al. [6] for visible-light images. Whereas Larson et al.'s method determines contrast limits to mimic a human visual system (HVS) model that is agnostic with respect to sensor details, Li et al. [7] modified this approach to actually prevent sensor noise in nonlinear HDR inputs from becoming visible in LDR outputs. While demonstrated only for a visible-band application, the latter approach is equally applicable to invisible-band modalities. Other authors such as Rana et al. [8] have investigated learning-based approaches for TMOs, where the LDR output is meant for a machine as opposed to a human observer.

In scenarios where only an LDR output is retained, after applying a TMO to an HDR input, inverse-TMO mappings can recover HDR images for subsequent processing by machine observers. Gunawan et al. [9] and Tade et al. [10] elaborate on image quality assessments that are especially suited for such applications. These assessments, which fall into full reference, no reference, and reduced reference categories, quantify the impact of visual artifacts that are more likely to be produced by certain TMOs than others. With test cases, visual artifacts may also or instead be evaluated by human observers. Both Gunawan et al. and Tade et al. have discussed the tension between objective and subjective methods. Each has disadvantages and advantages. Nevertheless, in an image processing pipeline, producing an LDR output to be displayed to a human observer may not prevent retention of an HDR input for subsequent processing by a machine observer.

For real-time tone mapping of video, there is a difference between a method and an apparatus. The latter emphasizes hardware. Ou et al. [11] reviewed 60 methods for real-time tone mapping realized as apparatus. New figures of merit beyond image quality assessments become important. These include apparatus complexity and cost, supported frame rates, and power consumption. Whereas graphics processing units (GPUs) support complex local TMOs at video rates, they have high unit costs and high power consumption. field-programmable gate arrays (FPGAs) have low unit costs, demand less power, and are reconfigurable, making them ideal for global and simpler local TMOs at video rates. More recent works by Kashyap et al. [12] and Muneer et al. [13] make the same points.

In a previous apparatus-focused publication [14], we presented a bit-true design flow to realize the contrast-limited histogram equalization method of Li et al. [7] in Xilinx and Altera FPGAs. Realized designs for megapixel resolutions consuming only tens of milliwatts (mW) at video rates fit within the fifth simplest device of the Spartan-6 family, the lowest-cost family in production by Xilinx at the time of the research. Our model-based design flow involved MATLAB and Xilinx's Integrated Synthesis Environment (ISE).

Xilinx is now owned by Advanced Micro Devices (AMD). Today, AMD offers four families of Xilinx 7-Series FPGA devices [15]. In order of their increasing levels of sophistication and unit cost, these are Spartan-7, Artix-7, Kintex-7, and Virtex-7. Integrated within AMD Zynq-7000 platforms [16], Artix-7 devices are the lowest-cost FPGAs suitable

for such system-on-chip (SoC) platforms. Moreover, ISE has been replaced by Vivado, also from AMD, to design, build, and test circuits expressed in a hardware-description language (HDL), e.g., very-high-speed integrated-circuit HDL (VHDL). For this work, we used Vivado 2022.2, together with MATLAB & Simulink R2023a.

In this paper, we propose and validate a new design flow for a contrast-limited histogram equalization apparatus, following Li et al.'s method [7] for nonlinear response HDR sensors. The flow results in circuit designs, called TMO 2025 systems, that significantly outperform those which can be obtained with our previous design flow [14]. We verify and evaluate designs using the hallway2 HDR video that we produced by combining scene luminance data from the literature [17] with a nonlinear response HDR model of a log digital pixel sensor (DPS) array [18], a CMOS image sensor.

Our new design flow leverages Simulink modeling to reduce the number of random-access memory (RAM) bits required by an exponential factor without compromising the number of logic cells required, the maximum frequency of operation, or the power consumption at megapixel video rates. We target the simplest device of the Artix-7 family and remodel our previous realizations, called TMO 2021 systems, with the new design flow to facilitate explanation and for comparative evaluation. The results show, consistent with structural similarity (SSim) scores [19], negligible impact on video output of a lower-complexity TMO 2025 system with respect to the TMO 2021 reference [14].

Although Hai et al. [20] have advocated using Simulink, which accompanies MATLAB, and its HDL Coder blockset to develop an FPGA circuit, we used neither in our previously published TMO research [14]. After summarizing the productivity gains of their design flow, Hai et al. commented on how Simulink models can facilitate classroom explanations of a system for edge detection and human height estimation.

The rest of this paper is structured as follows. Section 2 starts with a design overview of TMO 2021 and TMO 2025 systems, then continues with design details (featuring Simulink models) of five subsystems, called Scene Histogram, Perceived Histogram, Tone Function, Global Mapping, and Interpolation. Section 3 addresses bit-true verification and timing constraint analysis of VHDL models developed from the Simulink ones using Vivado. The same section evaluates circuit realizations for a variety of video formats, focusing on complexity, speed, and power. Finally, Section 4 summarizes the motivation, scope, and main contributions of this work.

2. Apparatus and Methods

The remodeled TMO 2021 and novel TMO 2025 systems have four main parameters: the number of pixels, n , per frame; the video rate, f , in frames per second (fps); the histogram bin size, s_{bin} , in bits; and the contrast limit, h_{max} , which is a histogram ceiling. Starting with a design overview, this section presents subsystem designs in sequence for 16-bit unsigned integer HDR input and 8-bit unsigned integer LDR output.

2.1. Design Overview

Figure 1 presents Simulink models of the TMO 2021 and TMO 2025 systems. Using a hierarchical design with multiple subsystems, which is an ideal way to organize complex circuits, these are top-level models showing system inputs and outputs plus subsystem interconnects, all of which carry sample-based signals.

In the sample-based approach, an HDR video input, y_{in} , streams pixel-by-pixel in row-major order through the systems to produce an LDR video output, w_{out} , strictly in sync with one FPGA clock signal. The last pixel of one frame, in the bottom-right corner, is followed by the first pixel of the next frame, in the top-left corner. Whereas this paper

presents the sample-based Simulink models, the accompanying explanation employs a frame-based notation for simplicity.

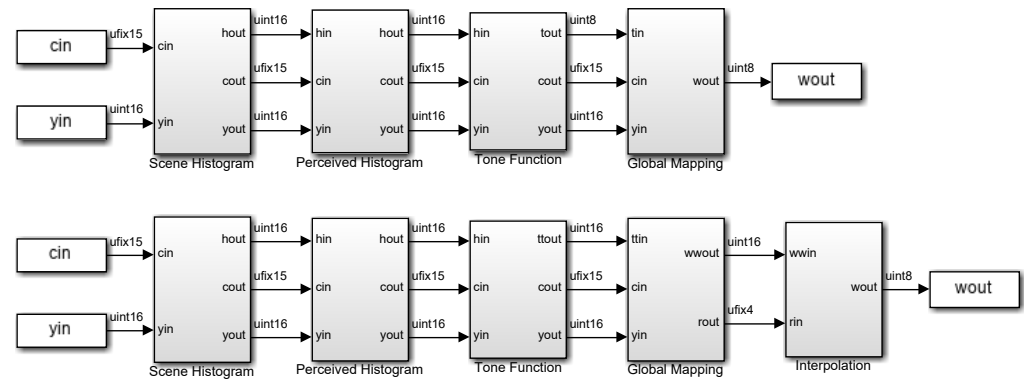


Figure 1. Simulink models of tone-mapping operator (TMO) systems. With Simulink, we refactored our 2021 histogram-based TMO design (top) into four subsystems. With a fifth subsystem, Interpolation, our improved TMO 2025 design, bottom, yields an exponential reduction in field-programmable gate array (FPGA) memory requirements. Both designs require a control input, *cin*, identical for the same parameters, like the number of pixels, n , per frame and the histogram bin size, s_{bin} , in bits. Also, high-dynamic-range (HDR) video streams in as *yin* and low-dynamic-range (LDR) video streams out as *wout*. For illustrative purposes, $\lceil \log_2 n \rceil$ equals 16 and s_{bin} equals 4 in all schematics.

With the Simulink models, this work flows naturally from algorithm to circuit design, capturing novel and significant aspects of the latter without requiring details of VHDL representations to be presented. Frame-based explanations of MATLAB classes define “methods” of the TMO 2021 and TMO 2025 systems, whereas sample-based illustrations of Simulink models define “apparatuses” of the same.

The TMO 2021 design has, after remodeling, the four subsystems shown and named in Figure 1. In contrast, the TMO 2025 design, while having the same four subsystems in the same sequence, adds a new subsystem, Interpolation, and a second output to a subsystem, Global Mapping. The presented models indicate the data types of all signals. Compared to the TMO 2021, the TMO 2025 system has *uint16* not *uint8* signals on two interconnects. Name changes for subsystem input/output ports, *ttout* for *tout*, *ttin* for *tin*, and *wwout* for *wout* accompany wordlength doublings. These data type changes in a Simulink model map to double-width buses in an FPGA circuit.

The Scene Histogram subsystem computes the current frame’s histogram while reading out the previous frame’s histogram. The Perceived Histogram subsystem applies a frame-based low-pass filter (LPF) to scene histogram inputs. The Tone Function subsystem outputs a normalized cumulative sum of a modified histogram, computed from a perceived histogram and the contrast limit. The Global Mapping stores a currently computing tone function, while applying a previously computed tone function to the HDR input. Using least-significant bits (LSBs) of the HDR input to refine LDR output from the Global Mapping, the Interpolation produces the LDR output of the TMO 2025 design. Consistent with a global TMO, the Interpolation does not employ neighbouring pixels.

The Scene and Perceived Histogram subsystems require three RAMs, each of a wordlength, $\lceil \log_2 n \rceil$, that depends on the number of pixels. For the TMO 2021 and TMO 2025 designs, the Global Mapping subsystem requires two RAMs that have single-width and double-width wordlengths: 8 and 16, respectively. Considering the number of RAM words, the Simulink modeling predicts the memory in 2^{10} bits (Kb), K_{2021} and K_{2025} , required by TMO 2021 and TMO 2025 realizations as follows:

$$K_{2021} = 2^{16-s_{\text{bin}}}(3\lceil\log_2 n\rceil + 2 \cdot 8)/2^{10}, \quad (1)$$

$$K_{2025} = 2^{16-s_{\text{bin}}}(3\lceil\log_2 n\rceil + 2 \cdot 16)/2^{10}. \quad (2)$$

Before remodeling, the TMO 2021 system had multiple control signals. These are now represented by one multi-bit unsigned integer (ufix15 in Figure 1) control input, *cin*, that operates at the same clock rate as the HDR input and LDR output, *y_{in}* and *w_{out}*. The TMO 2025 system adopts the same approach to control signalling, i.e., the one control input, *cin*, decomposes as needed into $16 - s_{\text{bin}}$ most-significant bits (MSBs) and three LSBs, called *cin₀*, *cin₁*, and *cin₂* below.

2.2. Scene Histogram

Shown in Figure 2, the Scene Histogram subsystem computes and writes, to a RAM, the histogram of the current video frame, called frame *k*, and simultaneously reads out, from another RAM, the histogram of the previous video frame, called frame *k* − 1. Each RAM with additional circuitry defines a RAM system, which is technically a sub-subsystem.

When one control bit, *cin₀*, is on, RAM system A reads out and RAM system B counts. When it is off, RAM system A counts and RAM system B reads out. Each frame is toggled by the control bit. The RAM systems receive complementary versions of this control bit. They require another control bit, *cin₁*, to apply a reset operation with read-out.

A histogram of one video frame, an image, is a count of how many pixels have values that fall within one of multiple non-overlapping bins, where the bins cover all possible pixel values. We can model the sample-based scalar input, *y_{in}*, as a frame-based vector input, *y_{in}*. A vector, at frame *k*, has one uint16 entry for each pixel. Using a binning approach that proves efficient for circuit design, let each histogram bin cover the LSBs of possible pixel values, specifically its *s_{bin}* LSBs. The video input, *y_{in}* or *y_{in}*, is turned into a binned video input, *y_{bin}* or *y_{bin}*, by extracting its $16 - s_{\text{bin}}$ MSBs:

$$\mathbf{y}_{\text{bin}}[k] = \lfloor 2^{-s_{\text{bin}}} \mathbf{y}_{\text{in}}[k] \rfloor. \quad (3)$$

The range of the MSBs, *y_{bin}*, determines the number of histogram bins. This number, $2^{16-s_{\text{bin}}}$, specifies the number of words in each RAM system's RAM. Each RAM word is a counter that may count up to the number of pixels, *n*. We employ a dual-port RAM because counting involves reading how many pixels have already been counted for a particular bin, adding one to that value, and writing the result back to the same bin. At the end of a frame period, a counted histogram stored in a RAM is available for read-out.

The sample-based histogram output, *h_{out}*, at a particular RAM address, *addr*, is the number of pixels from the previous frame whose binned values, corresponding to *y_{bin}*, equal that address. With a nonlinear vector-to-vector mapping function, *H*, we model the Scene Histogram subsystem as follows, where entries of the frame-based histogram output, *h_{out}*, require a wordlength, $\lceil\log_2 n\rceil$, that is weakly dependent on the number of pixels:

$$\mathbf{h}_{\text{out}}[k] = \mathbf{H}(\mathbf{y}_{\text{bin}}[k - 1]). \quad (4)$$

Each RAM system, which counts or reads out a histogram, receives two control bits, *cin₁* and either *cin₀* or not(*cin₀*), that are called by new names, *wr_en1* and *wr_en0*. In the counting mode, when *wr_en0* is on, an increment applies to the RAM data at *addr*, i.e., at the MSBs of the video input, *y_{in}*. The RAM system performs incrementation via a feedback loop, having one-sample latency due to RAM latency and with a one-sample delay at the write port with respect to the read port. For correct operation, a plus-two increment applies in situations where consecutive *addr* samples are identical.

In the read-out mode, when *wr_en0* is off and *wr_en1* is on, a reset constant, 0, passes to the data input, *din*, of the RAM. Each time data at an *addr* is read out, on the next

sample the RAM resets the word at the same addr. Unlike the counting mode, when addr values may present in any order, during the read-out mode addr values present either as a strictly decreasing or a strictly increasing sequence.

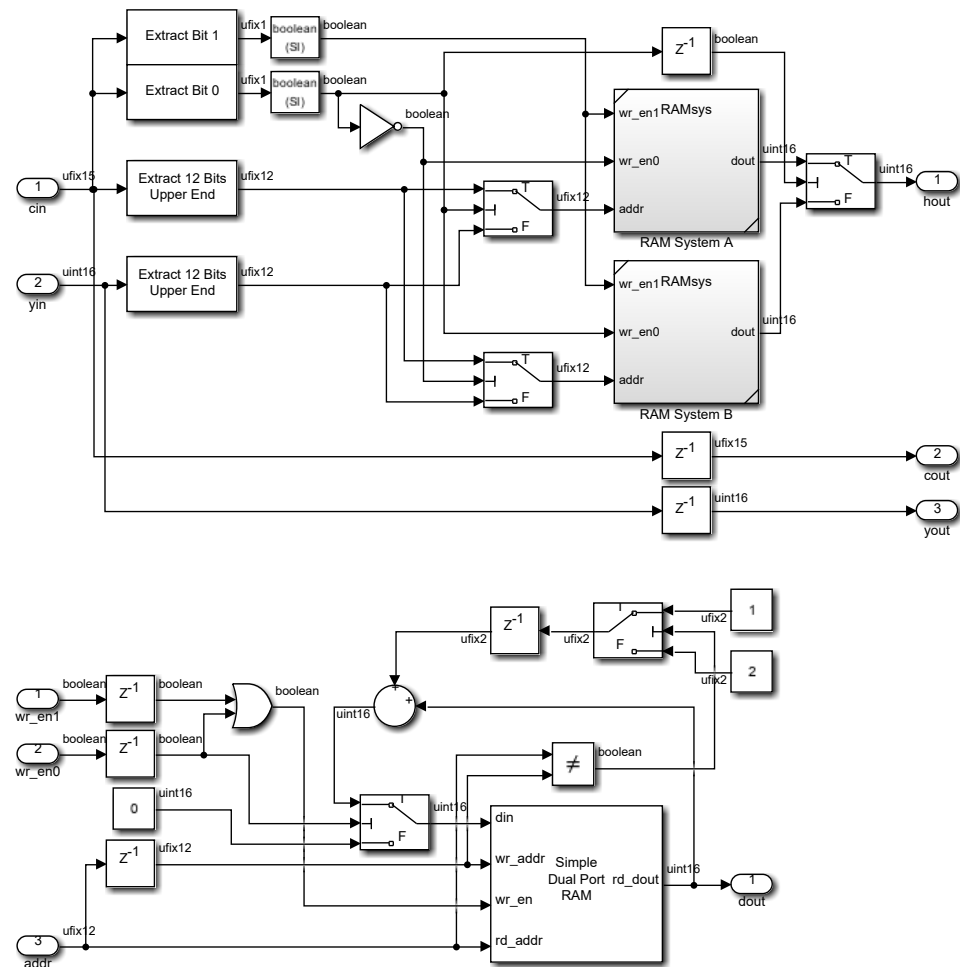


Figure 2. Scene Histogram of TMO 2021 and TMO 2025 systems. A ping-pong configuration allows the subsystem, top, with two instances of a random-access memory (RAM) sub-subsystem, bottom, and three switches to compute the histogram of a current video frame while, in parallel, reading out a computed histogram of the previous frame. When one control bit, cin_0 , is on, sub-subsystems A and B are called ping and pong, respectively. When it is off, roles are reversed. The ping sub-subsystem reads out and resets its RAM when another control bit, cin_1 , is on. The pong sub-subsystem performs RAM operations called counting, with reads and writes to the same addresses.

In Figure 2, the Scene Histogram subsystem's first output, $hout$, is the previous histogram, which is valid when one control bit, cin_1 , is on. Depending on another control bit, cin_0 , one of the two RAM systems reads out. With one-sample latency, the count for an addressed bin presents at a RAM system output, $dout$, and routes to the subsystem output, $hout$. Control and video signals required by the next subsystem pass from inputs to outputs, from cin and yin to $cout$ and $yout$. One-sample delays guarantee synchronization.

2.3. Perceived Histogram

Figure 3 presents the Perceived Histogram subsystem, which computes a perceived histogram, $hout$, from a scene histogram, hin . As the frame period, T_f , may exceed a time constant, 0.4 seconds (s), for HVS adaptation to changing illumination, the TMO 2021 and TMO 2025 systems compute a tone function from the result, i.e., the perceived histogram, of a frame-based LPF applied to the scene histogram.

Although designed in sample-based fashion, the subsystem is easier to explain with equations using frame-based histogram signals, $\mathbf{h}_{\text{out}}$ for hout and $\mathbf{h}_{\text{in}}$ for hin :

$$\mathbf{h}_{\text{out}}[k] = \begin{cases} \lfloor 2^{-s}(\alpha \mathbf{h}_{\text{out}}[k-1] + \beta \mathbf{h}_{\text{in}}[k]) \rfloor, & k \geq 2, \\ \mathbf{h}_{\text{in}}[k], & 0 \leq k < 2, \end{cases} \quad (5)$$

where

$$\alpha = \text{round}(2^s a), \quad (6)$$

$$\beta = 2^s - \alpha, \quad (7)$$

$$a = e^{-T_f/0.4}. \quad (8)$$

In the case of negligible round-off error, i.e., given a sufficiently large scale factor, 2^S , with respect to parameter ratios, the model simplifies as follows:

$$2^s \gg \max\left(\frac{a}{1-a}, \frac{1-a}{a}\right), \quad (9)$$

$$\mathbf{h}_{\text{out}}[k] \approx \begin{cases} a\mathbf{h}_{\text{out}}[k-1] + (1-a)\mathbf{h}_{\text{in}}[k], & k \geq 2, \\ \mathbf{h}_{\text{in}}[k], & 0 \leq k < 2. \end{cases} \quad (10)$$

After the first two frames, representing an internal-state initialization where the perceived histogram, $\mathbf{h}_{\text{out}}$, equals the scene histogram, $\mathbf{h}_{\text{in}}$, the model further simplifies into a frame-based difference equation in forward recursion form as follows:

$$\mathbf{h}_{\text{out}}[k] \approx a\mathbf{h}_{\text{out}}[k-1] + (1-a)\mathbf{h}_{\text{in}}[k]. \quad (11)$$

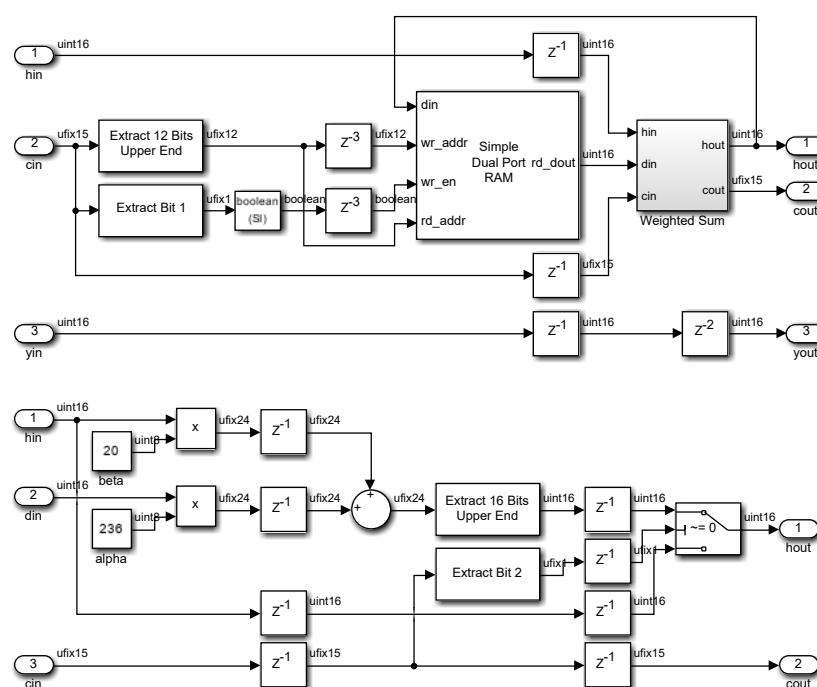


Figure 3. Perceived Histogram of TMO 2021 and TMO 2025 systems. This subsystem, top, employs a dual-port RAM and a weighted-sum sub-subsystem, bottom, to implement a frame-based low-pass filter (LPF) in sample-based fashion. The LPF smooths, consistent with human perception, sudden changes in light-intensity distribution from a scene histogram input, h_{in} , to produce a perceived histogram output, h_{out} . The filter operates when a control bit, cin_2 , is on. Otherwise, it simply passes the input histogram to the output one with a three-sample latency. When it is on during each frame period, another control bit, cin_1 , indicates that the histogram input is valid.

Like the TMO 2021 system, the TMO 2025 system adopts established values, $1/30$ s and 8, for the aforementioned constants, T_f and s . The frame period, T_f , derives from a frame rate, f , of popular video formats. The LPF coefficients, α and β , have uint8 values, 236 and 20, that indicate greater dependence of the current perceived histogram on the previous perceived histogram yet still have significant dependence on the current scene histogram. If the scene histogram is constant, the LPF passes it to the perceived histogram exactly.

The Perceived Histogram subsystem employs one dual-port RAM. The wordlength, $\lceil \log_2 n \rceil$, of each histogram sample on paths from input to output, i.e., from h_{in} to h_{out} , depends on the number of pixels, n . This number for illustrative purposes has a fixed value, 160×240 , in Figure 3 and related figures. The video input, y_{in} , simply passes to a corresponding output, y_{out} , with three-sample latency for synchronization. Likewise, the full control input, c_{in} , passes to a corresponding output, c_{out} .

Scene histogram bin values, h_{in} , enter the subsystem associated with bin addresses. The addresses enter synchronously as MSBs of the control input, c_{in} . When one control bit, c_{in_1} , is on, bin values and addresses are valid. During this part of a frame period, perceived histogram bin values come from corresponding bin addresses of the RAM. Updated through a weighted-sum sub-subsystem, they write back to the RAM at the same addresses. We apply three-sample delays at write address and write enable, wr_addr and wr_en , inputs of the dual-port RAM in the Perceived Histogram. This synchronizes with a three-sample latency of the weighted-sum feedback pathway.

The weighted-sum sub-subsystem has two parallel multiplications with constants followed by an addition. Favours high-speed operation via sequential logic, a unit delay follows each arithmetic operation. One product takes the scene histogram as an input and its filter coefficient, β , as an argument. The other product takes the previous perceived histogram as an input and its filter coefficient, α , as an argument. The previous perceived histogram comes from the read port, rd_dout , of the dual-port RAM. Multiplications of u -bit signals and v -bit constants yield $(u + v)$ -bit signals. These increases in wordlengths precede an equal decrease in wordlength after addition. Extraction of MSBs equates to a left-shift operation, which is practically a zero-cost operation in an FPGA.

When a control bit, c_{in_2} , is off, a switch in the weighted-sum sub-subsystem passes the scene histogram input to the perceived histogram output. Because of the feedback path to the write port, din , of the RAM in the Perceived Histogram subsystem, this serves to initialize the RAM. When the control bit, c_{in_2} , is on, the weighted sum passes to the sub-subsystem output. Delays provide control bit synchronization.

2.4. Tone Function

Figure 4 presents the Tone Function of the TMO 2025 system. The subsystem outputs a uint16 signal, tt_{out} , called double-width because a final stage produces it by concatenating, as low and high bytes, a uint8 signal and its unit delay. The Tone Function of the TMO 2021 system, which is not shown for brevity, is identical except that it does not have the final stage. Therefore, its tone function output, t_{out} , is a uint8 signal called single-width. This signal is also available inside the Tone Function of the TMO 2025 system.

The single-width tone function, t_{out} , is a normalized cumulative sum of a modified histogram. Computed on the fly from a perceived histogram, h_{in} , and a constant, h_{max} , the modified histogram needs no RAM. The histogram computes during a limited part of each frame period, when one control bit, c_{in_1} , is on. A sample-based accumulator involving a resettable delay implements the cumulative sum. The previous output of the accumulator adds to the current input and the result becomes the output of the accumulator

on the next clock cycle. As the accumulator must zero at the start of a cumulative sum, its delay state resets to a constant, 0, upon each rising edge of the control bit, cin_1 .

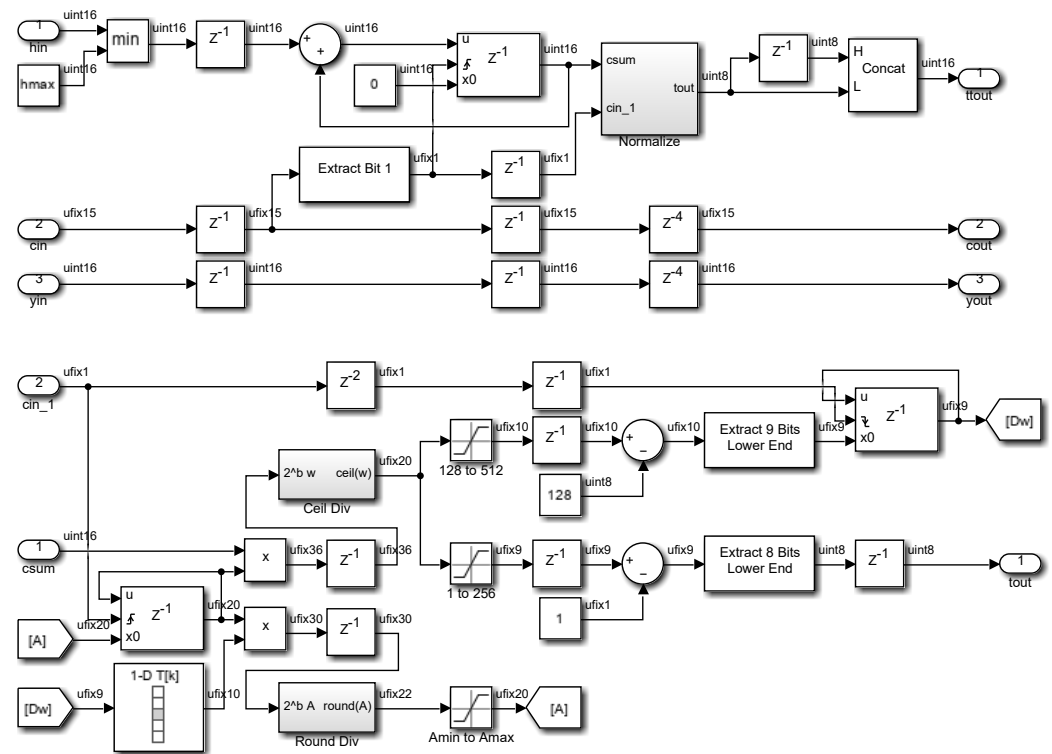


Figure 4. The Tone Function subsystem of the TMO 2025 system. The cumulative sum, top, of a modified histogram, $\min(h_{in}, h_{max})$, yields a double-width tone function, t_{out} , after normalization and concatenation. A control bit, cin_1 , of an edge-triggered delay resets the sum at the start of each frame. A normalize sub-subsystem, bottom, approximates without division a ratio of the cumulative sum, $csum$, to the end value of the previous cumulative sum. It multiplies this nonnegative ratio by 256, rounds up, and subtracts one to produce a single-width tone function, t_{out} . The without-division normalization uses feedback, a small look-up table (LUT), and multiplications.

A normalized cumulative sum of a scene or perceived histogram would yield a tone function for histogram equalization. Because histogram values specify slopes of the tone function, we realize contrast-limited histogram equalization by clipping values to a ceiling, h_{max} , before the sum. Exact normalization requires the end value of the sample-based cumulative sum, $csum$. Unlike with a scene or even a perceived histogram, with a modified histogram, the sum's end value may not be equal to the number of pixels, n , which is a constant.

Ideally, we divide the cumulative sum by its maximum, the end value in a sample-based approach, and round off after multiplying by 255. By normalizing instead with the maximum value of the previous cumulative sum, we gain considerable efficiency and sacrifice little accuracy. The modified histogram follows the perceived histogram, which unlike the scene histogram changes slowly, always from frame to frame.

Consider a frame-based version, $h_{in,f}$, of the sample-based histogram, h_{in} , and a frame-based version, $t_{out,f}$, of the sample-based tone function, t_{out} . A frame-based model of the tone function, called a with-division model, is as follows:

$$\mathbf{h}_{\text{mod}}[k] = \min(\mathbf{h}_{\text{in}}[k], h_{\text{max}}), \quad (12)$$

$$\mathbf{c}_{\text{sum}}[k] = \text{cumsum}(\mathbf{h}_{\text{mod}}[k]), \quad (13)$$

$$c_{\text{max}}[k] = \max(\mathbf{c}_{\text{sum}}[k]), \quad (14)$$

$$\mathbf{t}_{\text{out}}[k] = S(\lceil 2^8 \mathbf{c}_{\text{sum}}[k] / c_{\text{max}}[k-1] \rceil) - 1, \quad (15)$$

where, to help express a required uint8 saturation,

$$S(w) = \begin{cases} 256, & w > 256, \\ 1, & w < 1, \\ w. & \text{otherwise.} \end{cases} \quad (16)$$

In Figure 4, the Tone Function employs a sub-subsystem to realize a normalization that computes without division. Two nonidealities, a previous not current divisor and an approximate not exact division, mean that the normalization requires a 1-to-256 saturation before minus-one blocks. After the minus-one blocks, we obtain the single-width tone function, $\mathbf{t}_{\text{out}}$ or $\mathbf{t}_{\text{out}}$, by simply extracting the eight LSBs, a uint8 signal.

Whether rounding up, down, or off, after scaling by a positive power of two, division of non-constant signals would demand significant FPGA resources. We replace the division in the Tone Function model by multiplication with a non-constant multiplier, A , and with scaling by a negative power of two, $2^{-\lceil \log_2 n \rceil}$, a left-shift operation:

$$A[k] \approx \text{round}(2^{8+\lceil \log_2 n \rceil} / c_{\text{max}}[k-1]), \quad (17)$$

$$\mathbf{t}_{\text{out}}[k] = S(\lceil 2^{-\lceil \log_2 n \rceil} \mathbf{c}_{\text{sum}}[k] A[k] \rceil) - 1. \quad (18)$$

Starting from an initial guess, A_{min} , we compute the multiplier, A , through a feedback process, observing that an expression, w_{max} , should equal 256 at the end of each frame period. If the expression is less than or equal to 128, we double the multiplier. If it is greater than or equal to 512, we halve the multiplier. There are 383 possible values in between these limits that we handle with an LUT. The expression of interest is

$$w_{\text{max}}[k] = \lceil 2^{-\lceil \log_2 n \rceil} c_{\text{max}}[k] A[k] \rceil. \quad (19)$$

To update the multiplier, A , we designed a 385-word LUT to retrieve a positive integer, R , that with a multiplication, a bit-shift, and a rounding-off results in a good prediction of the best multiplier, A , to use during the next frame period:

$$R[k] = \begin{cases} 512, & \Delta w[k] \leq 0, \\ \text{round}(2^8 256 / w_{\text{max}}[k]), & 1 \leq \Delta w[k] \leq 383, \\ 128, & \Delta w[k] \geq 384, \end{cases} \quad (20)$$

$$A[k] = \text{round}(2^{-8} A[k-1] R[k-1]), \quad (21)$$

where, to facilitate a realization of the constant LUT,

$$\Delta w[k] = w_{\text{max}}[k] - 128. \quad (22)$$

Moreover, the multiplier computation entails a saturation, indicated in Figure 4, whereby any result less than a minimum, A_{min} , saturates to the minimum and any result greater than a maximum, A_{max} , saturates to the maximum. These values are

$$A_{\text{min}} = \text{round}(2^{8+\lceil \log_2 n \rceil} / n), \quad (23)$$

$$A_{\text{max}} = \text{round}(2^{8+\lceil \log_2 n \rceil} / h_{\text{max}}). \quad (24)$$

As shown in Figure 4, the Tone Function does not use the video input, y_{in} . Neither does it use all but one bit, cin_1 , of the control input, cin . The control and video inputs pass to outputs, c_{out} and y_{out} , with five-sample latencies (“Ceil Div” and “Round Div” incorporate unit delays) for synchronization with respect to the mapping from the perceived histogram, h_{in} , to the double-width tone function, tt_{out} .

2.5. Global Mapping

The Global Mapping subsystem, presented in Figure 5, employs two single-port RAMs in ping-pong fashion. During each frame period, the ping RAM exclusively performs read operations and the pong RAM exclusively performs write operations or no operations at all. Roles reverse each frame. Because neither RAM must perform read and write operations at the same time, we do not employ dual-port RAMs.

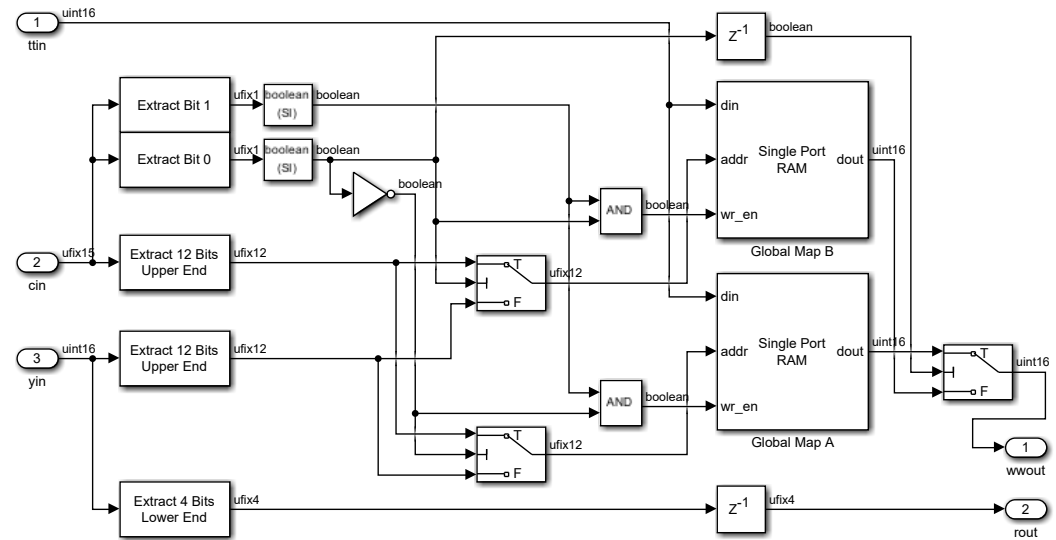


Figure 5. The Global Mapping subsystem of the TMO 2025 system. Using two switches and a control bit, cin_0 , this subsystem operates two RAMs in ping-pong fashion. Addressed by the most-significant bits (MSBs) of the video input, y_{in} , one RAM serves as a read-only LUT. Addressed by the MSBs of the control input, cin , the other RAM writes a tone function, tt_{in} , to memory while another control bit, cin_1 , is on. A third switch toggled by the control bit, cin_0 , ensures the RAM configured for read drives the output, ww_{out} . The subsystem also outputs, with a one-sample latency for synchronization, unused least-significant bits (LSBs) of the video input, y_{in} .

The subsystem takes an HDR input, y_{in} , and produces a double-width LDR output, ww_{out} . These sample-based signals have frame-based representations, y_{in} and ww_{out} . The HDR-to-LDR mapping can then be expressed as follows:

$$y_{bin}[k] = \lfloor 2^{-s_{bin}} y_{in}[k] \rfloor, \quad (25)$$

$$ww_{out}[k] = \mathbf{TT}\{y_{bin}[k], tt_{in}[k-1]\}, \quad (26)$$

where the function, $\mathbf{TT}$, is a nonlinear mapping from two vector signals, y_{bin} and tt_{in} , with the latter delayed by one frame, meaning stored in RAM memory, to one vector signal, ww_{out} . Frame-based signals, y_{bin} and tt_{in} , represent sample-based signals, y_{bin} and tt_{in} , noted in the Simulink model.

We model the vector-to-vector tone function, $\mathbf{TT}$, as the entrywise (global) application of a scalar tone function, TT , to every entry, y , of one vector argument, $\mathbf{y}$, as follows:

$$\mathbf{TT}\{\mathbf{y}, \mathbf{tt}\} = \begin{bmatrix} TT(y_1, \mathbf{tt}) \\ \vdots \\ TT(y_n, \mathbf{tt}) \end{bmatrix}. \quad (27)$$

Each entry, y , is an unsigned integer with well-defined limits, $[0, m)$ where m equals $2^{16-s_{\text{bin}}}$. The scalar function, TT , is a piecewise function defined by its second argument, $\mathbf{tt}$. The form of this function is that of an input-dependent LUT:

$$TT(y, \mathbf{tt}) = \begin{cases} tt_1, & y = 0, \\ \vdots & \vdots \\ tt_m, & y = m - 1. \end{cases} \quad (28)$$

Global mapping from the HDR input, y_{in} , to the double-width LDR output, w_{out} , may be thought of as a uint16-to-uint16 input-dependent LUT which can be realized in an FPGA using two RAMs. Considering the memory available in low-cost FPGAs, we require significantly fewer address bits than 16. With the proposed design, the number of address bits equals the wordlength, $16 - s_{\text{bin}}$, of the binned HDR input, y_{bin} .

Returning to Figure 5, simple bit extraction, which is practically a zero-cost operation in an FPGA circuit, yields the binned video input, y_{bin} , from the video input, y_{in} . The binned video input routes, thanks to a control bit, cin_0 , a NOT gate, and two switches, to the address input, addr , of the ping RAM. Because the ping RAM stores a vector of data corresponding to the previous frame's tone function, $\mathbf{tt}_{\text{in}}$, it produces the required result, w_{out} , at the data output, dout .

While a control bit, cin_1 , is on during each frame period, the pong RAM is the one whose wr_en input is on because of another control bit, cin_0 or $\text{not}(\text{cin}_0)$. The ping RAM is the one whose wr_en input is off for the whole frame period. Two AND gates and one NOT gate implement this logic. A simple two-input switch routes the ping RAM's output, dout , and not the pong RAM's output, also dout , to one subsystem output, w_{out} . After accounting with a unit delay for one-sample latency of RAM reads (i.e., lookups), the ping-pong control bit, cin_0 , controls the switch.

The pong RAM stores a vector of data, using addresses supplied by MSBs of the control input, cin , and entries supplied by the tone function input, $\mathbf{tt}_{\text{in}}$, valid only when one control bit, cin_1 , is on. The stored vector of data specifies a scalar function to be used for global mapping during the next frame period. Two switches route MSBs of the control input to the address input, addr , of the pong RAM. The tone function routes directly to the data input, din , of the pong and ping RAMs. The latter ignores the data input.

The TMO 2025 system takes advantage of the s_{bin} LSBs of the video input, y_{in} or y_{in} . They pass synchronously to a subsequent Interpolation subsystem as a second output, rout , having a frame-based representation, $\mathbf{r}_{\text{out}}$, as follows:

$$\mathbf{r}_{\text{out}}[k] = y_{\text{in}}[k] \bmod 2^{s_{\text{bin}}}. \quad (29)$$

Figure 5 presents the TMO 2025's subsystem only. Three differences specify TMO 2021's corresponding design. First, uint16 input and output signals, $\mathbf{tt}_{\text{in}}$ and w_{out} , become uint8 input and output signals, $\mathbf{tin}$ and wout . Second, wordlengths of RAMs are halved. Third, the TMO 2021's subsystem does not have a second output, rout .

2.6. Interpolation

Figure 6 presents the Interpolation subsystem of the TMO 2025 system. Interpolation provides a contribution to the final LDR output, w_{out} , using all bits of the original HDR input, y_{in} , as opposed to a select number, $16 - s_{\text{bin}}$, of MSBs. Though the TMO 2021 system

lacks this subsystem, a frame-based model of its global mapping, which depends on a single-width tone function, is a design stepping stone.

In the figure, “Extract Bits [4 11]” means to extract 8 bits from bit 4 up to and including bit 11. These numbers illustrate a specific subsystem where the bin size in bits is 4. The generic design, represented by the Simulink model and associated MATLAB code specifies an extraction of 8 bits from bit s_{bin} up to and including bit $s_{\text{bin}} + 7$.

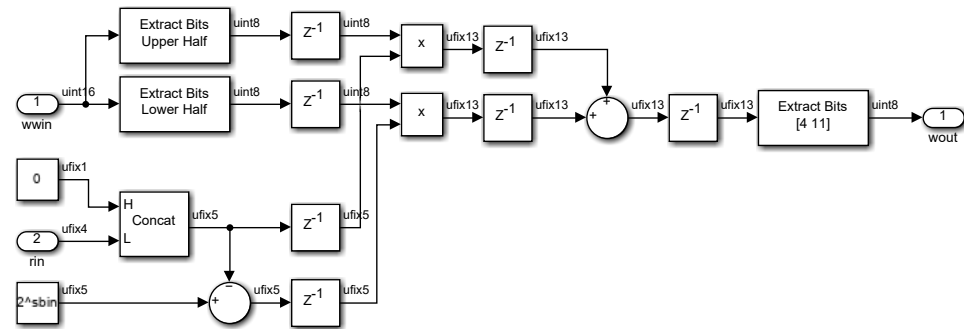


Figure 6. Interpolation subsystem, in the TMO 2025 system only. This subsystem takes a double-width LDR input, $\mathbf{wwin}$, and produces a single-width LDR output, $\mathbf{wout}$, with a three-sample latency to account for three combinational logic stages: addition, multiplication, and addition. A remainder HDR input, $\mathbf{rin}$, and a constant, $2^{s_{\text{bin}}}$, provide two weights employed for linear interpolation using the upper and lower bytes of the LDR input, $\mathbf{wwin}$. For illustrative purposes, s_{bin} equals 4.

Consider a scalar function, T , that defines an entrywise vector-to-vector mapping, $\mathbf{T}$, of the HDR input, $\mathbf{y}_{\text{in}}$, to the TMO 2021’s LDR output, denoted $\mathbf{w}_{\text{out}}^{\text{L}}$ below, where a single-width tone function, $\mathbf{t}_{\text{in}}$, defines the scalar function:

$$\mathbf{T}(\mathbf{y}, \mathbf{t}) = \begin{bmatrix} T(y_1, \mathbf{t}) \\ \vdots \\ T(y_n, \mathbf{t}) \end{bmatrix}, \quad (30)$$

$$T(y, \mathbf{t}) = \begin{cases} t_1, & y = 0, \\ \vdots & \vdots \\ t_m, & y = m - 1, \end{cases} \quad (31)$$

$$\mathbf{y}_{\text{bin}}[k] = \lfloor 2^{-s_{\text{bin}}} \mathbf{y}_{\text{in}}[k] \rfloor, \quad (32)$$

$$\mathbf{w}_{\text{out}}^{\text{L}}[k] = \mathbf{T}(\mathbf{y}_{\text{bin}}[k], \mathbf{t}_{\text{in}}[k - 1]). \quad (33)$$

Because the low byte of the double-width tone function, $\mathbf{t}_{\text{in}}$, exactly equals the single-width tone function, $\mathbf{t}_{\text{in}}$, the TMO 2021’s single-width LDR output, $\mathbf{w}_{\text{out}}^{\text{L}}$, exactly equals the low byte of TMO 2025’s double-width LDR output, $\mathbf{ww}_{\text{out}}$:

$$\mathbf{w}_{\text{out}}^{\text{L}}[k] = \mathbf{ww}_{\text{out}}[k] \bmod 2^8. \quad (34)$$

What the TMO 2025 system does, by virtue of the Interpolation subsystem, is to take advantage of the high-byte, $\mathbf{w}_{\text{out}}^{\text{H}}$, of the double-width LDR output, $\mathbf{ww}_{\text{out}}$, with the help of another input, $\mathbf{r}_{\text{in}}$, the LSB remainder of the original HDR input, $\mathbf{y}_{\text{in}}$:

$$\mathbf{w}_{\text{out}}^{\text{H}}[k] = \lfloor 2^{-8} \mathbf{ww}_{\text{out}}[k] \rfloor, \quad (35)$$

$$\mathbf{r}_{\text{in}}[k] = \mathbf{y}_{\text{in}}[k] \bmod 2^{s_{\text{bin}}}. \quad (36)$$

The LSBs, $\mathbf{r}_{\text{in}}$, are a nonnegative fractional part of the binned HDR input, $\mathbf{y}_{\text{bin}}$. Entries of the fractional part fall in an open interval, $[0, 1)$. If every entry was exactly 1, we

could apply the vector-to-vector mapping, $\mathbf{T}$, to the binned HDR input plus one, $\mathbf{y}_{\text{bin}} + 1$. The result is exactly equal to the high byte, $\mathbf{w}_{\text{out}}^{\text{H}}$, of the double-width LDR output:

$$\mathbf{T}(\mathbf{y}_{\text{bin}}[k] + 1, \mathbf{t}_{\text{in}}[k - 1]) = \mathbf{w}_{\text{out}}^{\text{H}}[k]. \quad (37)$$

Our model requires a small correction, a saturation, to the scalar function, T . With this correction, as follows, the scalar function can represent the incremental output, $T(y + 1)$, for a special case input, y , corresponding to a saturation value, $m - 1$:

$$T(y, \mathbf{t}) = \begin{cases} \vdots & \vdots \\ t_m, & y = m. \end{cases} \quad (38)$$

The TMO 2025 system takes advantage of the LSBs, $\mathbf{r}_{\text{in}}$, by entrywise linear interpolation between the low-byte output, $\mathbf{w}_{\text{out}}^{\text{L}}$, and the high-byte output, $\mathbf{w}_{\text{out}}^{\text{H}}$, as follows:

$$\mathbf{w}_{\text{out}}[k] = \left\lfloor \frac{(2^{s_{\text{bin}}} - \mathbf{r}_{\text{in}}[k])\mathbf{w}_{\text{out}}^{\text{L}}[k] + \mathbf{r}_{\text{in}}[k]\mathbf{w}_{\text{out}}^{\text{H}}[k]}{2^{s_{\text{bin}}}} \right\rfloor. \quad (39)$$

This formula guarantees the Interpolation subsystem output, $\mathbf{w}_{\text{out}}$, has the same range, $[0, 2^8)$, as the low-byte result, $\mathbf{w}_{\text{out}}^{\text{L}}$. Where remainder bits, $\mathbf{r}_{\text{in}}$, approach zero, the output approaches the low-byte result. Where they approach their maximum, $2^{s_{\text{bin}}} - 1$, the output approaches the high-byte result, $\mathbf{w}_{\text{out}}^{\text{H}}$. In between, a linearly weighted sum is obtained.

3. Results and Discussion

This section begins by presenting the general approach and example results by which we verified Simulink models and corresponding FPGA realizations in bit-true fashion. Then, it compares the visual quality achieved by three designs for one video format. The evaluation focuses on circuit specifications achieved by three designs for five video formats.

3.1. Verification

Following the Simulink models, we developed the TMO 2021 and TMO 2025 systems as VHDL designs suitable for FPGA implementation. Using Vivado, we synthesized circuit realizations for Artix-7 target devices. Given test cases of video and control inputs, $\mathbf{y}_{\text{in}}$ and $\mathbf{c}_{\text{in}}$, the video output, $\mathbf{w}_{\text{out}}$, could be compared bit-for-bit between the Simulink and Vivado realizations. Team members could work separately and in parallel on different aspects of the project's design, which included a variety of debugging tasks. In this fashion, Vivado realizations of easier-to-explain Simulink models were productively completed with bit-true verification.

After the functional verification, synthesized TMO 2021 and TMO 2025 designs underwent translation and mapping and place and route, using Vivado, for the three simplest FPGA devices of the Artix-7 family of FPGA devices from Xilinx, now AMD. The simplest device, XC7A12T, has up to 12,800 logic cells and up to 912,384 memory bits available for design synthesis. To fit this device, therefore, the memory required by the five RAMs must not exceed 891 Kb for a given design.

We synthesize a specific FPGA circuit from a generic VHDL design by choosing specific parameters, like the number of pixels, n , the frame rate, f , the histogram bin size, s_{bin} , and the contrast limit, h_{max} . Given a synthesized circuit, Vivado tools report data that can be used to summarize the circuit's complexity, in particular the number of logic cells and memory bits utilized. The smaller these are, relative to the available resources, the more feasible it becomes to include them on the same FPGA other circuits, beside the TMO, to complete a multi-stage image processing pipeline.

After translation and mapping and place and route, we used static timing analysis (STA) to predict the maximum frequency of operation in a target device. A synthesized circuit, either for the TMO 2021 or TMO 2025 system, requires one clock frequency, i.e., the sample rate (not to be confused with the frame rate). During STA, we varied the clock frequency from low to high—actually, we varied the clock period from high to low in 0.1 nanosecond (ns) decrements—until Vivado predicted a timing violation.

Examples of timing violations include failures to meet the setup-and-hold requirements of a logic cell on a critical circuit path. Violations could occur due to propagation delays of combinational logic subcircuits, considering place-and-route details. We designed the TMO 2021 and TMO 2025 systems, using Simulink, to intersperse combinational logic with delays, e.g., z^{-1} blocks. Using Vivado, they map to FPGA resources configured for sequential logic, i.e., pipelined elements that facilitate high-frequency operation.

The maximum frequency that ensures functional correctness, as predicted by Vivado after the place and route, may be compared to the sample rate associated with the number of pixels and the required frame rate. If the maximum frequency is less than the sample rate then the circuit will not work in the target device at the required frame rate. Targeting a more complex device in the same Artix-7 family could solve the problem. Alternately, another device family may be required. This work limits its scope to the simplest devices of the lowest-cost AMD family, i.e., Artix-7, suitable for SoC platforms.

The TMO 2021 and TMO 2025 systems require input and control signals to function. For testing purposes, whether in Simulink or Vivado, we produced signals in MATLAB from a video file called `hallway2` and a sensor file called `CISmodels`. The video is an eye-level view of a person walking toward a window overlooking a city riverfront in daylight and then, after turning left, walking along a hallway with the window at the right. In its native high-definition (HD) format, the 11 s clip has 720×1280 pixels at 30 fps. Using nearest-neighbour interpolation, via the `imresize` function in MATLAB, we upsampled or downsampled frames to vary the video format for testing purposes.

Table 1 lists the video formats of interest. Two of them, the full HD (FHD) and 4K ultra HD (4KUHD) ones, are megapixel formats. Two of them, the video-graphics-array (VGA) and half-quarter VGA (HQVGA) ones, are sub-megapixel formats. The VGA format may also be called the standard-definition (SD) format in the literature.

Table 1. Video formats investigated with Simulink and Vivado. The number of pixels and frame rate, in frames per second (fps), determine the sample rate, in megahertz (MHz). They are parameters that influence the logic, memory, and power required by specific field-programmable gate array (FPGA) realizations of generic TMO 2021 and TMO 2025 system designs. Scene and perceived histogram RAMs have wordlengths that depend weakly, as shown, on the number of pixels.

Video Format Acronym (Acronym Definition)	Number of Pixels, n	Frame Rate, f	Sample Rate, fn	Width of Histogram RAMs, $\lceil \log_2 n \rceil$
HQVGA (half-quarter VGA)	160×240	30 fps	1 MHz	16
VGA (video-graphics-array)	480×640	30 fps	9 MHz	19
HD (high-definition)	720×1280	30 fps	28 MHz	20
FHD (full HD)	1080×1920	30 fps	62 MHz	21
4KUHD (4K ultra HD)	2160×3840	30 fps	249 MHz	23

The `hallway2` video file encodes pixel intensities as single-precision floating-point numbers calibrated to real-world luminance, in candelas per metre squared (cd/m^2), on a linear scale. Using the `CISmodels` sensor file, a MATLAB script produces the test video input, $\mathbf{y}_{\text{in}}$, from upsampled or downsampled “raw” video input, $\mathbf{x}_{\text{in}}$, as follows:

$$\mathbf{y}_{\text{in}}[k] = \text{round}(F(\mathbf{x}_{\text{in}}[k]) + \epsilon_{\text{in}}[k]). \quad (40)$$

In this frame-based model, where k is the frame number, F is an entrywise monotonic function that converts a frame of scene luminances, $\mathbf{x}_{in}$, to a frame of sensor responses, $\mathbf{y}_{in}$, that define the HDR input. The model incorporates zero-mean Gaussian noise, ϵ_{in} , that is independently and identically distributed from sample to sample.

Figure 7 plots the sensor function, F , as well as the noise standard deviation, σ_{ϵ} . The sensor file specifies the function and the noise standard deviation for a log APS and a log DPS image sensor based on experimentally obtained data. This work uses only the latter. Although we mathematically model the video input in frame-based fashion, during simulation, it streams through Simulink models and Vivado realizations of TMO 2021 and TMO 2025 systems in a sample-based fashion, the same as the control input.

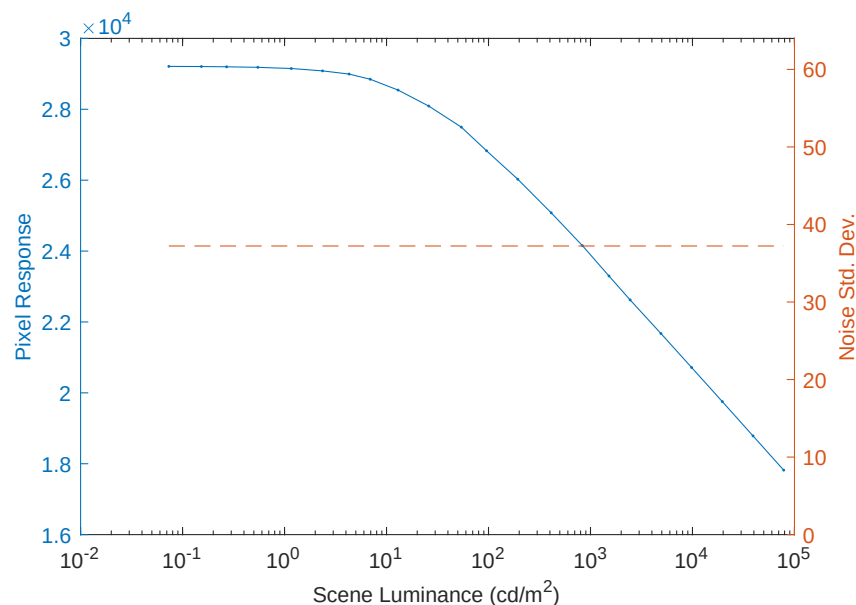


Figure 7. Simulated logarithmic (log) digital pixel sensor (DPS) array. A log DPS image sensor model, adapted from Nascimento et al. [14], maps scene luminances in candelas per metre squared (cd/m^2) to pixel responses. Applied to an 11 second (s) video clip called *hallway2*, shared in cd/m^2 by Kronander et al. [17], this monotonic nonlinear function serves to produce the HDR input, $\mathbf{y}_{in}$, used to test TMO 2021 and TMO 2025 systems. A dashed line on a second y-axis presents the standard deviation of the simulated sensor noise, too small to be shown with the pixel response.

Figure 8 illustrates the control signalling of the TMO 2025 system juxtaposed with example video input and output signals. The remodeled TMO 2021 design adopts exactly the same control strategy and takes the same video input. We partially automated the bit-true verification. Invoking Simulink, MATLAB scripts verified frame-based models automatically. Additionally, the Simulink model and Vivado circuit designers manually shared and compared sample-based files of input, output, and selected intermediate signals. The results were identical, whether produced using Simulink or Vivado.

The control input, $\mathbf{cin}$, has MSB and LSB parts. The MSB part is a periodic sawtooth waveform. During each period, the ramp part specifies a strictly monotonic sequence of addresses used to read/write histogram bin values out of/into RAMs. The duration of the ramp in samples equals the number, $2^{16-s_{bin}}$, of RAM words. Always satisfied here, this number must be strictly less than the number of pixels, n . As for the LSB part, one control bit, $\mathbf{cin}_2$, activates a LPF in the Perceived Histogram subsystem. The bit is zero for the first two frames worth of samples and remains one thereafter. For the remaining two LSBs, $\mathbf{cin}_1$ and $\mathbf{cin}_0$, assume all combinations each frame and repeat periodically. One bit, $\mathbf{cin}_1$, is on during the ramp part of the sawtooth waveform. It is associated, therefore, with read/write operations of bin values.

System designs employ RAMs to store all histograms and tone functions. The designs configure pairs of RAMs, some dual-port, in ping-pong fashion. One RAM, the ping RAM, performs reads alone while the other, the pong RAM, performs writes only or reads and writes. In each pair, ping and pong RAMs operate in parallel without interference. Each frame period, the RAMs of each pair swap roles. What was the ping RAM becomes the pong RAM and vice versa. One control bit, cin_0 , toggles from zero-to-one or one-to-zero at the start of each frame and remains constant thereafter. The bit controls switches that implement the ping-pong behaviour.

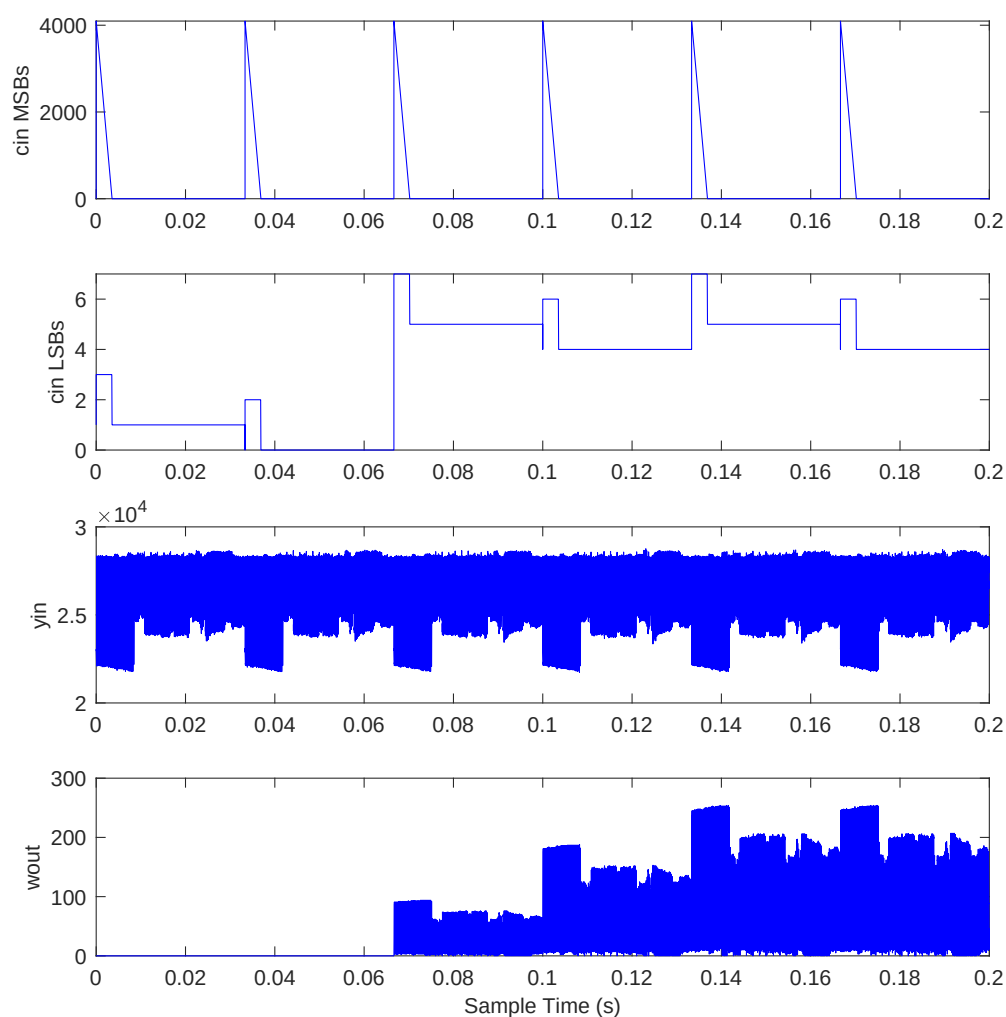


Figure 8. Sample-based input and output of the TMO 2025 system. Three LSBs of the control input, cin , toggle ping-pong RAMs in the Scene Histogram and Global Mapping, enable writes of bin values or tone functions, and enable the LPF of the Perceived Histogram after two frames. The remaining MSBs address all histogram bins in sequence. The HDR input, yin , represents the first six frames of an HQVGA clip produced from *hallway2* and streamed pixel by pixel through the TMO 2025 system to yield an LDR output, $wout$. In this example, the bin size in bits, s_{bin} , equals 4.

3.2. Evaluation

Figure 9 presents the LDR output of two TMO 2021 systems and one TMO 2025 system for the same HDR input, *hallway2*, in the same video format, HQVGA. The figure also shows, using a frame-based MATLAB algorithm, the LDR output of histogram equalization. For two methods, one TMO 2021 system and the histogram equalization, the bin size in bits, s_{bin} , equals 2. Otherwise, it equals 8.

The figure primarily demonstrates a qualitative improvement in tone mapping, from the TMO 2021 to the TMO 2025 system, given an equal bin size in bits that cor-

responds to reduced memory requirements. Especially visible as blotchy textures on upper window regions, mapping artifacts appear in the TMO 2021 results, s_{bin} equals 8. They are absent in the TMO 2025 results, s_{bin} equals 8. The same figure illustrates, whether by the TMO 2021 or the TMO 2025 system, the usefulness of a contrast limit to histogram equalization. Without it, window regions exaggerate sensor noise and too little contrast remains to show bench textures clearly. These subjective observations agree with objective assessments, i.e., SSim scores computed via MATLAB's `ssim` function that are overlaid on filmstrip frames, as shown in the figure.

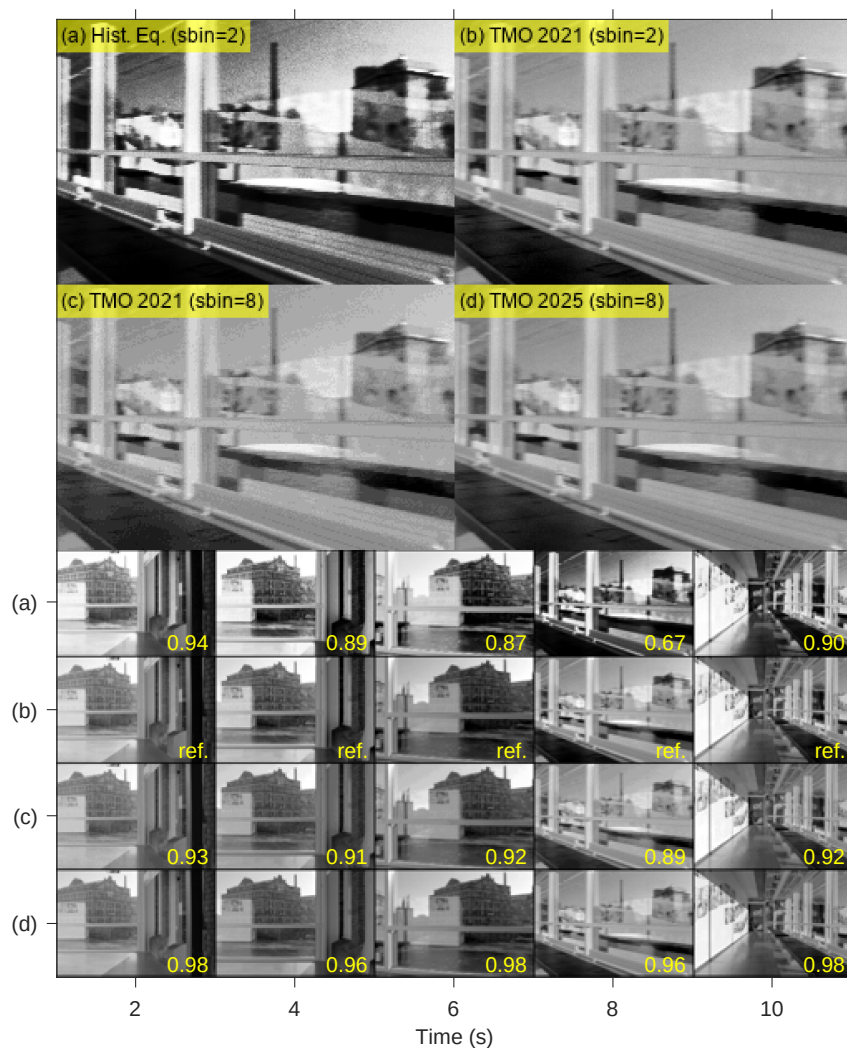


Figure 9. Frame-based output of TMO 2021 and TMO 2025 systems. Enlarged, top, about 8 s after the simulation start for an HDR input, `hallway2`, in the HQVGA format, the LDR outputs of the TMO 2021 system, s_{bin} equals 2, and the TMO 2025 system, s_{bin} equals 8, show the least sensor noise or mapping artifacts. The compared methods include histogram equalization, s_{bin} equals 2, and the TMO 2021 system, s_{bin} equals 8. Overlays, bottom, indicate structural similarity (SSim) scores.

Figure 10 illustrates key intermediate signals, namely histograms and tone functions, corresponding to a processing of the `hallway2` clip. Although the figure shows one frame's worth of data, when consecutive frames are compared, the perceived histogram follows scene histogram changes slowly, smoothing changes out over time. Tone functions likewise change slowly over time.

Whereas the scene and perceived histograms stream through an FPGA implementation one sample at a time in real time, Figure 10 plots bin values, i.e., counts of particular ranges of pixel intensities, on a y-axis versus pixel intensity on the x-axis. With a second y-axis,

the figure also shows the effective tone functions of the TMO 2021 and TMO 2025 systems applied to the video input for the given frame.

Tone functions equate to a normalized cumulative sum of a modified histogram, derived from the perceived histogram and a constant, $h_{\max}$. The systems compute the normalized cumulative sum in real time using histogram bin values that stream from high bin addresses to low bin addresses, i.e., from right to left in Figure 10. As a result, the mapping functions are inverting, an inversion that compensates for the inverting response, shown in Figure 7, of the chosen log DPS sensor.

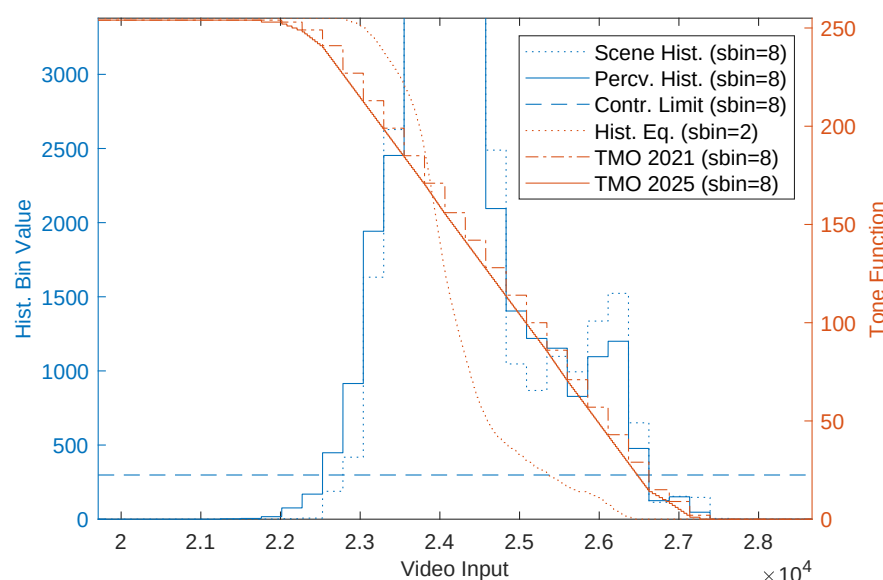


Figure 10. Frame-based histogram and tone function examples. The left y-axis plots scene and perceived histograms for the 240th frame (8 s mark) of the *hallway2* input (HQVGA format). The right y-axis plots the effective tone functions of the TMO 2021 and TMO 2025 systems. With the latter, remainder bits are treated as fractional bits in determining coordinates along the x-axis. Like inputs and outputs, intermediate signals were checked bit-for-bit from MATLAB mathematical models, the methods, to Simulink models and Vivado realizations of FPGA circuits, the apparatuses.

Mapping functions are contrast-limited; they have a maximum possible (negative) slope. We achieve this by utilizing a modified histogram that equals the perceived histogram only when it is lower than a “contrast limit” threshold, illustrated in Figure 10. Otherwise, the modified histogram equals the threshold, a ceiling calculated as follows:

$$h_{\max} = \left\lceil \frac{2^{s_{\text{bin}}} n}{256\sqrt{12}\sigma_{\epsilon}} \right\rceil. \quad (41)$$

Here, the number of pixels, n , the histogram bin size, s_{bin} , and the noise standard deviation, σ_{ϵ} , are parameters. With this ceiling, tone functions determine LDR outputs while limiting the visibility after tone mapping of the sensor noise present in HDR inputs.

Figure 10 shows that the mapping function of the TMO 2021 system has a staircase shape, whereas that of the TMO 2025 system does not. By virtue of its final Interpolation subsystem, the latter brings back discarded bits as remainder bits in a real-time computation that smooths the staircase. One way to reduce the staircase effect of the TMO 2021 subsystem is to reduce the number of discarded bits, equal to the histogram bin size in bits, s_{bin} . However, doing so leads to an exponential-related increase in the RAM memory required to store the histograms and mapping functions.

Table 2 presents results obtained with Vivado. These include circuit complexity and maximum frequency for five video formats, HQVGA to 4KUHD. For each format, three

designs underwent circuit synthesis, translation and mapping, place and route, and STA. As with the examples in Figure 9, contrast limits correspond to the log DPS sensor.

In Table 2, the first two designs realize the TMO 2021 system for two values, 2 and 8, of the histogram bin size in bits, s_{bin} . Although the lower bin size yielded the reference results, shown in Figure 9, it required a significantly more complex circuit in terms of memory utilization for all video formats. With Artix-7 devices, Vivado can allocate block RAM memory in 18 Kb increments. Apart from five such blocks, one for each histogram and mapping RAM, Vivado allocated some non-block or distributed RAM memory to support, for example, the LUT of the Tone Function subsystem.

Table 2. Specifications of FPGA circuits, obtained with Vivado. Percentages are with respect to logic and memory available in the simplest Artix-7 device. At s_{bin} equals 8, although the TMO 2021 is a little less complex, has a higher max frequency, and needs a little less power than the TMO 2025, both designs qualify as low-complexity, high-speed, and low-power. Where timing constraints are satisfied, we report power consumption at 30 fps. At s_{bin} equals 2, the TMO 2021 does not fit the simplest Artix-7 device. Its frequency and power results are for the third simplest device.

Video Format	System Design (Bin Size, Bits)	Logic Cells (Utilization)	Memory Bits (Utilization)	Maximum Frequency	Static Power	Dynamic Power
HQVGA	TMO 2021 (2)	376 (3%)	1154 K (130%)	182 MHz	61 mW	1 mW
	TMO 2021 (8)	306 (2%)	91 K (10%)	164 MHz	60 mW	1 mW
	TMO 2025 (8)	442 (3%)	92 K (10%)	120 MHz	58 mW	1 mW
VGA	TMO 2021 (2)	456 (4%)	1316 K (148%)	172 MHz	61 mW	12 mW
	TMO 2021 (8)	395 (3%)	92 K (10%)	164 MHz	60 mW	2 mW
	TMO 2025 (8)	530 (4%)	92 K (10%)	119 MHz	58 mW	2 mW
HD	TMO 2021 (2)	539 (4%)	1424 K (160%)	169 MHz	61 mW	40 mW
	TMO 2021 (8)	467 (4%)	92 K (10%)	164 MHz	60 mW	7 mW
	TMO 2025 (8)	603 (5%)	92 K (10%)	119 MHz	58 mW	7 mW
FHD	TMO 2021 (2)	525 (4%)	1478 K (166%)	169 MHz	61 mW	94 mW
	TMO 2021 (8)	501 (4%)	92 K (10%)	161 MHz	60 mW	15 mW
	TMO 2025 (8)	637 (5%)	92 K (10%)	119 MHz	58 mW	17 mW
4KUHD	TMO 2021 (2)	637 (5%)	1586 K (178%)	167 MHz	Max freq. $< fn$	
	TMO 2021 (8)	565 (4%)	92 K (10%)	161 MHz	Max freq. $< fn$	
	TMO 2025 (8)	741 (6%)	92 K (10%)	119 MHz	Max freq. $< fn$	

For equal parameters, the TMO 2021 and the TMO 2025 systems have equivalent complexity, as shown in Table 2. With the latter design, we consider only one value, 8, of the bin size in bits, s_{bin} . As shown in Figure 9, with this bin size for the HQVGA format, the TMO 2025 system produces video output comparable to that of the TMO 2021 system with a smaller bin size, 2. However, Table 2 shows that the TMO 2025 design of similar visual quality enjoys an order of magnitude reduction in required memory.

Compared to the TMO 2021, the TMO 2025 design requires a bit more logic, mainly because it involves an extra subsystem: Interpolation. It also doubles the width of some buses and RAMs. Required memory grows with the logarithm of the number of pixels. Because read/write logic synthesizes in proportion to RAM sizes, we find a weak dependence of logic on the number of pixels. Relative to resources available in the simplest Artix-7 device, logic requirements are low and nearly independent of video format.

Table 2 shows that simple Artix-7 devices support all video formats except 4KUHD for all designs. A device supports a video format when the maximum frequency, as determined by STA, exceeds the required sample rate for the chosen frame rate. For all supported cases, the table reports the static and dynamic power of the circuit at 30 fps, as determined by Vivado. Static power, on the order of 100 mW, is approximately constant. Maximum

frequency tends to decrease as the number of pixels increase, most likely because timing constraints are harder to satisfy when larger RAMs are placed and routed.

Dynamic power depends on the circuit, its sample rate, and the device. Given equal parameters, power consumption hardly increases from the TMO 2021 to the TMO 2025 design, despite the extra circuit complexity. Given a fixed frame rate, the dynamic power increases with the number of pixels. In all reported cases, the dynamic power is less than 100 mW, the order of the static power. By this measure, circuit designs are power-efficient. However, due to an exponential increase in RAM memory, the TMO 2021 design requires more dynamic power than the TMO 2025 design for artifact-free tone mapping.

4. Conclusions

Automotive, smartphone, and other applications have motivated research into TMOs, which are useful stages in image processing pipelines for HDR video. Given an HDR input, a TMO produces an LDR representation. Compared to local TMOs, global TMOs are especially suitable for FPGA realizations. This work does not investigate local TMOs and takes a model-based, not a learning-based, approach to global TMOs called contrast-limited histogram equalizations, as tailored to nonlinear HDR sensors. The work presents an in-depth comparison to one previously published design, the TMO 2021.

To realize an exponential improvement, the TMO 2025 design, this work developed a MATLAB–Simulink–Vivado design flow, applying it first to remodel the TMO 2021 design. With MATLAB, frame-based algorithms abstract key parts of the TMO. With Simulink, sample-based models incorporate blocks that capture essential features of circuits and that prove convenient for debugging. With the design flow, we productively realized functionally verified systems that meet timing constraints for targeted FPGAs. Vivado simulations of developed VHDL models matched bit for bit with Simulink model results and transitively with MATLAB algorithm results.

After presenting a design overview, this paper detailed the TMO 2021 and TMO 2025 designs, one subsystem at a time. Due to remodeling of the TMO 2021 design with Simulink, small changes to third and fourth subsystems, called Tone Function and Global Mapping, with the addition of a fifth subsystem, called Interpolation, yielded the TMO 2025 design. Both designs have identical first and second subsystems, called Scene Histogram and Perceived Histogram. By specifying the number of pixels, the video rate, the histogram bin size, and a contrast limit, generic designs yield specific systems.

This paper accompanies sample-based Simulink models, illustrated in figures, with frame-based equations and text for explanatory purposes. In this fashion, we elaborated on the novel TMO 2025 design while compactly articulating exactly how it differs from the reference TMO 2021 design. With the Interpolation subsystem, the tone function instead of having a staircase mapping, from HDR input to LDR output, has a linearly interpolated refinement of the staircase mapping.

We realized the TMO 2021 and TMO 2025 designs as systems for five video formats and tested them with an 11 s video. Panning from a shadowed interior to a sunlit hallway, the clip incorporates the response and noise of a nonlinear sensor. Following the Simulink models, we developed FPGA circuits as VHDL models that, with parameter values, underwent synthesis, translation and mapping, and place and route using Vivado, for the simplest Artix-7 devices from Xilinx. We presented circuit results for fifteen realizations—test cases that included megapixel video at 30 fps.

The paper features sample-based and frame-based results from HQVGA realizations of the TMO 2021 design with two values and the TMO 2025 design with one value of a parameter that exponentially affects required memory. The presented results explain a simple control signalling scheme, elaborate on bit-true verification, and illustrate key

internal signals, namely scene and perceived histograms as well as tone functions without and with interpolation. Frame-based results also provide a comparison of histogram equalization to contrast-limited histogram equalization, indicating the capability of the latter to limit the visibility of sensor noise in video output.

Whereas the TMO 2021 design enables high-speed, low-power systems in FPGAs, when configured for artifact-free video output, the realizations require too much RAM memory to fit the simplest Artix-7 device, the lowest-cost FPGA from Xilinx, now AMD, suitable for SoC platforms. Thanks to an exponential reduction in required memory, with negligible impact on required logic, maximum frequency, and power consumption, the novel TMO 2025 design fits easily.

Author Contributions: Conceptualization, M.N. and D.J.; methodology, W.D., M.N. and D.J.; software, W.D., M.N. and D.J.; validation, W.D., M.N. and D.J.; formal analysis, W.D., M.N. and D.J.; investigation, W.D., M.N. and D.J.; resources, M.N. and D.J.; data curation, M.N. and D.J.; writing—original draft preparation, W.D. and D.J.; writing—review and editing, M.N. and D.J.; visualization, M.N. and D.J.; supervision, D.J.; project administration, D.J.; and funding acquisition, D.J., for publication. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, apart for the article processing charge.

Data Availability Statement: MATLAB code and Simulink models developed for this work, using MATLAB & Simulink R2023a, are archived in a public GitHub repository, <https://github.com/KoderKong/GlobalTMO>, accessible and last updated on 12 June 2025. They are shared with test data using a Simplified (2-Clause) BSD License, a permissive free license.

Acknowledgments: Maikon Nascimento and Dileepan Joseph would like to thank Rui (Rachel) Sun for her valued contributions, via a Mitacs Globalink Research Internship, to an earlier concept and method for tone mapping not used in this paper nor developed as an electronic apparatus.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

4KUHD	4K ultra HD
AMD	Advanced Micro Devices
APS	active pixel sensor
cd/m ²	candela per metre squared
CMOS	complementary metal-oxide-semiconductor
dB	decibel
DPS	digital pixel sensor
FHD	full HD
FPGA	field-programmable gate array
fps	frame per second
GPU	graphics processing unit
HD	high-definition
HDL	hardware-description language
HDR	high-dynamic-range
HQVGA	half-quarter VGA
HVS	human visual system
ISE	Integrated Synthesis Environment
Kb	2 ¹⁰ bit
LDR	low-dynamic-range
log	logarithmic
LPF	low-pass filter
LSB	least-significant bit

LUT	look-up table
MHz	megahertz
MSB	most-significant bit
mW	milliwatt
ns	nanosecond
RAM	random-access memory
s	second
SD	standard-definition
SoC	system-on-chip
SSim	structural similarity
STA	static timing analysis
TMO	tone-mapping operator
VGA	video-graphics-array
VHDL	very-high-speed integrated-circuit HDL

References

1. Takayanagi, I.; Kuroda, R. HDR CMOS Image Sensors for Automotive Applications. *IEEE Trans. Electron Devices* **2022**, *69*, 2815–2823. [\[CrossRef\]](#)
2. Brunetti, A.M.; Choubey, B. A Low Dark Current 160 dB Logarithmic Pixel with Low Voltage Photodiode Biasing. *Electronics* **2021**, *10*, 1096. [\[CrossRef\]](#)
3. Hajisharif, S.; Kronander, J.; Unger, J. Adaptive dualISO HDR reconstruction. *EURASIP J. Image Video Process.* **2015**, *2015*, 41. [\[CrossRef\]](#)
4. Khan, I.R.; Rahardja, S.; Khan, M.M.; Movania, M.M.; Abed, F. A Tone-Mapping Technique Based on Histogram Using a Sensitivity Model of the Human Visual System. *IEEE Trans. Ind. Electron.* **2018**, *65*, 3469–3479. [\[CrossRef\]](#)
5. Völgyes, D.; Martinsen, A.C.T.; Stray-Pedersen, A.; Waaler, D.; Pedersen, M. A Weighted Histogram-Based Tone Mapping Algorithm for CT Images. *Algorithms* **2018**, *11*, 111. [\[CrossRef\]](#)
6. Larson, G.W.; Rushmeier, H.; Piatko, C. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Trans. Vis. Comput. Graph.* **1997**, *3*, 291–306. [\[CrossRef\]](#)
7. Li, J.; Skorka, O.; Ranaweera, K.; Joseph, D. Novel Real-Time Tone-Mapping Operator for Noisy Logarithmic CMOS Image Sensors. *J. Imaging Sci. Technol.* **2016**, *60*, 020404. [\[CrossRef\]](#)
8. Rana, A.; Valenzise, G.; Dufaux, F. Learning-Based Tone Mapping Operator for Efficient Image Matching. *IEEE Trans. Multimed.* **2019**, *21*, 256–268. [\[CrossRef\]](#)
9. Gunawan, I.P.; Cloramidina, O.; Syafa'ah, S.B.; Febriani, R.H.; Kuntarto, G.P.; Santoso, B.I. A review on high dynamic range (HDR) image quality assessment. *Int. J. Smart Sens. Intell. Syst.* **2021**, *14*, 1–17. [\[CrossRef\]](#)
10. Tade, S.L.; Vyas, V. Tone Mapped High Dynamic Range Image Quality Assessment Techniques: Survey and Analysis. *Arch. Comput. Methods Eng.* **2021**, *28*, 1561–1574. [\[CrossRef\]](#)
11. Ou, Y.; Ambalathankandy, P.; Takamaeda, S.; Motomura, M.; Asai, T.; Ikebe, M. Real-Time Tone Mapping: A Survey and Cross-Implementation Hardware Benchmark. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2666–2686. [\[CrossRef\]](#)
12. Kashyap, S.; Giri, P.; Bhandari, A.K. Logarithmically Optimized Real-Time HDR Tone Mapping With Hardware Implementation. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 1426–1430. [\[CrossRef\]](#)
13. Muneer, M.H.; Pasha, M.A.; Khan, I.R. Hardware-friendly tone-mapping operator design and implementation for real-time embedded vision applications. *Comput. Electr. Eng.* **2023**, *110*, 108892. [\[CrossRef\]](#)
14. Nascimento, M.; Li, J.; Joseph, D. Efficient Pipelined Circuits for Histogram-based Tone Mapping of Nonlinear CMOS Image Sensors. *J. Imaging Sci. Technol.* **2021**, *65*, 040503. [\[CrossRef\]](#)
15. Xilinx. 7 Series: Product Selection Guide. Tech. Rep., Advanced Micro Devices, 2021. Available online: <https://docs.amd.com/v/u/en-US/7-series-product-selection-guide> (accessed on 19 March 2025).
16. Xilinx. Zynq-7000 SoC: Product Selection Guide. Tech. Rep., Advanced Micro Devices, 2019. Available online: <https://docs.amd.com/v/u/en-US/zynq-7000-product-selection-guide> (accessed on 26 March 2025).
17. Kronander, J.; Gustavson, S.; Bonnet, G.; Unger, J. Unified HDR reconstruction from raw CFA data. In Proceedings of the IEEE International Conference on Computational Photography, Cambridge, MA, USA, 19–21 April 2013; pp. 1–9. [\[CrossRef\]](#)
18. Mahmoodi, A.; Li, J.; Joseph, D. Digital Pixel Sensor Array with Logarithmic Delta-Sigma Architecture. *Sensors* **2013**, *13*, 10765–10782. [\[CrossRef\]](#) [\[PubMed\]](#)

19. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
20. Hai, J.C.T.; Pun, O.C.; Haw, T.W. Accelerating Video and Image Processing Design for FPGA using HDL Coder and Simulink. In Proceedings of the IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, Selangor, Malaysia, 15–17 October 2015; pp. 28–32. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.