AROMA : Augmented Reality Optimized Tone Mapping

ALEXANDRE CHAPIRO, Meta, USA
KRZYSZTOF WOLSKI, Meta, USA
EDGAR VELÁZQUEZ-ARMENDÁRIZ, Meta, USA
JON MCELVAIN, Meta, USA
AJIT NINAN, Meta, USA
RAFAŁ K. MANTIUK, University of Cambridge, United Kingdom



Fig. 1. (L) The target content shown with no additive background. Remaining panels show the same image viewed on an AR display with 6:1 additive contrast. (ML) Content reproduced with no mapping. (MR) A render using only the global tone curve portion of *AROMA* . (R) Shows the full *AROMA* method, including a local detail enhancement step. Note: a conventional display cannot faithfully reproduce optical-see-through AR appearance. Image credit: Chen et al. [2025].

AROMA is a tone mapping operator designed to optimize the visual fidelity of images displayed on optical see-through augmented reality (AR) devices, which are often challenged by limited dynamic range and substantial ambient light interference. The system aims to faithfully preserve the perceived color, brightness, and contrast of source images with moderate dynamic range, even when the display's effective dynamic range is significantly lower than that of the content. Rather than enhancing or altering the original appearance, *AROMA* aims to maintain perceptual equivalence, replicating original intent as closely as possible on the target display. To achieve this, *AROMA* employs a global tone curve minimizing perceived contrast distortion, complemented by adaptive detail enhancement to selectively restore high-frequency details lost due to dynamic range compression, and color saturation correction. *AROMA* 's approach to dynamic range mapping accounts for ambient light by adjusting the tone curve to maximize the display's usable range, ensuring that the final image is both perceptually accurate and visually pleasing, even in challenging lighting environments. A subjective study was conducted, demonstrating that *AROMA* significantly improves perceived visual quality, achieving results equivalent to a 45–59% increase in display luminance. As a result, *AROMA* can be used to enable AR rendering at lower display power, producing substantial energy savings without compromising user experience. Finally, *AROMA* is lightweight and feasible within form-factor-constrained head-mounted AR displays — our implementation runs at 250 FPS on Meta Ray-Ban Display AR glasses.[1].

[1]Code: https://github.com/gfxdisp/aroma

Authors' Contact Information: Alexandre Chapiro, alex@chapiro.net, Meta, USA; Krzysztof Wolski, , Meta, USA; Edgar Velázquez-Armendáriz, evelazquez@acm.org, Meta, USA; Jon McElvain, , Meta, USA; Ajit Ninan, , Meta, USA; Rafał K. Mantiuk, rafal.mantiuk@cl.cam.ac.uk, University of Cambridge, United Kingdom.

© 2026 ACM.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*.

CCS Concepts: • **Computing methodologies** → **Virtual reality**; **Mixed / augmented reality**; **Perception**; **Neural networks**.

Additional Key Words and Phrases: Research

ACM Reference Format:
Alexandre Chapiro, Krzysztof Wolski, Edgar Velázquez-Armendáriz, Jon McElvain, Ajit Ninan, and Rafał K. Mantiuk. 2026. AROMA : Augmented Reality Optimized Tone Mapping. *ACM Trans. Graph.* 1, 1, Article 1 (January 2026), 14 pages.

1 Introduction

Tone mapping is an essential part of high-quality presentation in modern displays. This term could be ambiguous in the literature, and can refer to algorithms that cover a few closely related goals:

(1) Computationally mapping content in one dynamic range (e.g., high-dynamic-range, or HDR) to a smaller range;
(2) Adapting content to display and viewing conditions by optimizing the look of the content for a given scenario;
(3) Mapping content from a scene-referred format to display-referred (e.g., standard container in the ICtCp color space to display output RGB pixels);
(4) A way to artistically customize the look of content, often via manual editing in conjunction with color grading.

The goal of *AROMA* is to generate an automatic pipeline for optical-see-through augmented reality (AR) displays covering points 1-3.

Our task is different from traditional tone mapping algorithms due to qualitatively new challenges posed by AR: exceptionally low contrast due to the presence of high ambient brightness from the natural world overlaid with the virtual image, combined with limitations to the maximum display luminance due to power and thermal constraints. As AR contrast is much lower than any other

display technology available in the market today (Section 2.1), producing visually pleasing and discernible images on an augmented reality device presents a new, previously unexplored challenge for tone mapping. Additionally, head-mounted displays (HMDs) have significant limitations in terms of power usage, so any mapping algorithm must take into account tone mapping's computational cost and complexity, which limit the range of applicable techniques.

AR source content can be incredibly varied: including existing media (HDR and SDR streams), camera outputs (e.g., passthrough), or locally generated content from a variety of different applications (games, productivity, augmented calling, etc.), all of which may carry varying dynamic ranges. Unlike traditional display systems, such as cinema or home theater, the effective dynamic range of AR devices may vary significantly due to the additivity with ambient light. Furthermore, for AR display systems, commonly employed electrochromic/photochromic dimming elements will change effective ambient luminance at the eye, influencing mapping decisions. The deployment target for *AROMA* is broad: any on-screen content shown through an additive optical see-through AR display — text, UI, rendered graphics, video — should be optimally presented across the full ambient-illumination range, aiming to reproduce the source's intended appearance on a conventional emissive display, without manual brightness adjustment by the wearer.

Our novel AR mapping framework's features include:

- *AROMA* **enhances appearance** in real-time based on incoming content and viewing conditions, aiming to maximize fidelity, and **produces a visually consistent appearance** across different devices and environments.
- *AROMA* 's quality improvement is on par with a brighter screen: our user study measured **display engine power savings** of 45–59% over no mapping at equivalent quality.
- *AROMA* is **computationally lightweight**, and relies on data typically available in display pipelines. *AROMA* runs at 250 FPS on the Meta Ray-Ban Display AR glasses.

2 Related Work

2.1 Augmented Reality

High-end computer monitors, such as an LCD MacBook Pro XDR display with local-dimming, can produce a maximum reported contrast of 1000000:1, while a typical low-end desktop LCD monitor may present contrast closer to 1500:1 [Chen et al. 2025]. Optical-see-through AR displays are qualitatively different: display light is mixed with light from the natural world, overlaid with the virtual image. In addition, the luminance of the display is limited by power and heat constraints. In tandem, this means exceptionally low contrast; an AR device's effective contrast range can be in the order of 10:1, and often less than 6:1 [Kim et al. 2025].

Saving display power is key to building lightweight, long-battery-life AR HMDs. Recent work by Chen et al. [2024] introduced several techniques for display power saving in VR. AR displays may not benefit from top-performing foveated techniques due to smaller fields-of-view and the lack of power-intensive eye-tracking. *AROMA* offers significant display power savings in AR settings (Section 9).

Aside from the effective contrast reduction due to the brightness of the ambient light, other characteristics of the environment (e.g., spatial frequency, color) can also affect the displayed content. For example, if content is overlaid on top of a busy background region (e.g., foliage), image details may be masked and have reduced visibility. In a display mapping pipeline, optimizing for variations in local ambient structure can be difficult, since environmental sensing information can often be imperfect, and achieving high-fidelity maps might be expensive to obtain in real time. We follow the recommendation of Chapiro et al. [2024], and simplify treatment by considering only the luminance component of the world image.

2.2 Color Spaces

Display-Referred vs. Scene-Referred Color Spaces. Research distinguishes between display-referred color spaces (e.g., sRGB [Anderson et al. 1996], BT.709 [ITU-R 2002]) and scene-referred HDR spaces (e.g., BT.2100 PQ [ITU-R 2016]). Display-referred spaces are designed for direct reproduction on standard displays, but AR devices typically benefit from tone mapping for all content due to variable luminance and ambient light, which can alter color appearance.

Linear, Display-Encoded, and Perceptually Uniform Color Spaces. Linear color spaces are used for physical modelling and color mixing but are rarely stored due to high bit-depth needs [Poynton 2012]. HMD display pipelines often contain an early linear portion (e.g., used for optical corrections). Display-encoded spaces introduce perceptually-motivated non-linearities (e.g. BT.1886 [ITU-R 2011] or PQ [SMPTE 2014]), making them more practical for storage and reproduction on displays, which are typically bit-depth limited.

2.3 Tone Mapping Algorithms

TMOs are classified as global (uniform across the image) or local (varying by image region). SMPTE ST 2094-10 [2016] defines global tone curves for HDR TV, computed from metadata, but does not address elevated black levels. In AR, real-time image statistics are needed to maximize perceived quality under low contrast. CLAHE (Contrast Limiting Adaptive Histogram Equalization) applies histogram equalization locally, capping contrast to avoid noise amplification in flat regions [Larson et al. 1997; Pizer et al. 1987].

Ambient-aware mappers aim to compensate for the effects of environmental light. Display-adaptive tone mapping optimizes a global tone curve to minimize visible contrast distortions, compensating for elevated black levels and ambient light [Mantiuk et al. 2008]. Follow-up work introduces appearance compensation modeling image appearance changes with luminance level, combining global tone curve optimization, spatial processing, and color compensation for dim displays [Wanat and Mantiuk 2014]. Eilertsen et al. [2015] present a TMO preventing noise and encoding artifacts from becoming visible after tone mapping, using fast optimization and filtering. Tariq et al. [2023] model local contrast at the pixel level, optimizing a parametric curve for real-time VR applications. Qian et al. [2024] present a per-display gamma and per-channel offset, optimized against ambient and DICOM JND-uniformity. Notably, these methods do not directly address the extreme challenges of tone mapping for augmented reality, for which *AROMA* introduces several new techniques like partial compensation (Section 4) and contrast-preserving detail enhancement (Section 6).

Finally, Kim et al. [2025] present a model for supra-threshold contrast visibility, aimed at AR applications. The authors leverage their model for auto-brightness in AR. This technique can be used to produce target display brightness values, which can then be optimally used by *AROMA* .

2.4 Color Correction, Gamut Mapping, and Color Appearance in AR

Color correction in tone mapping aims to preserve perceptual color fidelity after dynamic range compression. Mantiuk et al. [2009] explored several formulas, including Schlick [1995] color ratios and a luminance-preserving approach, both tunable by a parameter and applicable in RGB. They found an empirical formula for adjusting color saturation for simple contrast compression (gamma). In this work, we extend their method for use with complex tone curves.

Beyond traditional color-correction recipes for tone mapping, recent work specifically targets how optical-see-through AR displays distort perceived color and appearance [Zhang and Murdoch 2021]. Li et al. [2026] characterize color appearance shifts caused by additive ambient light in see-through AR under aligned reference/display conditions. Herbeck et al. [2024], focusing on face rendering, show that adjusting effective display transparency improves perceived face appearance in AR. These works suggest that AR introduces appearance distortions not captured by pipelines designed for emissive displays, and motivate *AROMA* 's explicit treatment of ambient luminance during tone-curve construction (Section 5) and its post-tone-curve color-correction step (Section 7).

Throughout this paper, "appearance" refers to the perceived color, brightness, and contrast of an image — a *corresponding-color* reproduction intent in the sense of Hunt's classification of color-reproduction objectives [Fairchild 2013; Hunt 2004] — rather than material attributes such as gloss or texture, which are governed by surface properties beyond the scope of tone mapping. *AROMA* aims to make the displayed AR image appear as close to its source as the AR display's effective contrast and gamut allow.

3 AROMA: Augmented Reality Optimized Tone Mapping

AROMA is designed to preserve the perceived contrast of an image presented on an OST-AR that is affected by light from a natural environment. Here, the term *contrast* denotes local spatial pixel differences, measured in Michelson or Weber units, and *perceived contrast* accounts for differences in contrast perception across luminance levels (as explained in Section 5). As shown in the processing diagram in Figure 2, the core of the method operates on tone values extracted from a color image (Section 4.4). First, the tone curve is optimized to minimize contrast and brightness distortions on the AR display (Section 5). Next, it is filtered over time (Section 5.4), and applied to the tone image. The local details are restored in the contrast-preserving detail enhancement step (Section 6). Finally, the color is transferred from the original image to the tone-mapped image in the color correction step (Section 7), and the final image is compensated for the ambient light.

4 Color processing

We first introduce the overall color processing pipeline, which requires the following information as input:

- I^{BT709} — An image with the content to be tone-mapped, represented as absolute linear RGB values, in a standard color space, for example, BT.709.
- Y_{amb} — Estimated amount of ambient light at the eye (after dimming, if present), expressed as mean luminance, that will be combined with the displayed image.
- Y_{dmax} — Peak absolute display luminance (e.g., 200 cd/m^2).
- Y_{dmin} — Minimum absolute display luminance (i.e., black level, e.g., 0.1 cd/m^2).
- Y_{smax} — Peak absolute luminance of the incoming content (e.g. $\approx$ 200 cd/m^2 for SDR).

The output of *AROMA* is the image in an absolute linear color space, to be passed to the display for rendering.

Spatial uniformity of Y_{amb}. AR devices typically rely on ambient-light sensors (ALS) rather than continuously running cameras for power efficiency, so a per-pixel world image is not available to the mapper. *AROMA* follows the recommendation of Chapiro et al. [2024], treating Y_{amb} as a single aggregate luminance value. Saturated world-side highlights (e.g., a lamp) elevate the local ambient value above this mean, but in practice users typically re-orient their head away from such regions.

4.1 Partial compensation

AROMA does not attempt to fully compensate for the entire amount of ambient light. This is done because in many cases the amount of ambient light in AR settings is so high that display contrast becomes too small to effectively reproduce an image (e.g., a contrast of 2:1). Currently, the amount of compensation is computed as the fraction of the total ambient light:

$$Y_{\mathrm{comp}} = k_{\mathrm{comp}}\, Y_{\mathrm{amb}} \quad \text{where} \quad 0 < k_{\mathrm{comp}} \leq 1\,, \tag{1}$$

where $k_{\mathrm{comp}} = 0.25$ in our experiments. The goal of the global tone mapping step is mapping to the range $[\,Y_{\mathrm{dmin}} + Y_{\mathrm{comp}}\,,\, Y_{\mathrm{dmax}} + Y_{\mathrm{comp}}]$, where Y_{dmin} and Y_{dmax} are the black level and the peak luminance of the display. After doing so, we can perform a partial ambient light compensation by subtracting Y_{comp} from the RGB color values. This will improve the contrast in dark areas, otherwise lost due to ambient light (see Figure 1).

Note that, under ambient light, a display has the dynamic range (termed **additive contrast** ratio in AR):

$$\frac{Y_{\mathrm{dmax}} + Y_{\mathrm{amb}}}{Y_{\mathrm{dmin}} + Y_{\mathrm{amb}}}\,. \tag{2}$$

Using partial compensation, we assume a larger dynamic range:

$$\frac{Y_{\mathrm{dmax}} + Y_{\mathrm{comp}}}{Y_{\mathrm{dmin}} + Y_{\mathrm{comp}}} = \frac{Y_{\mathrm{dmax}} + k_{\mathrm{comp}}\, Y_{\mathrm{amb}}}{Y_{\mathrm{dmin}} + k_{\mathrm{comp}}\, Y_{\mathrm{amb}}}\,. \tag{3}$$

4.2 Color space

To reduce the risk of color clipping due to out-of-gamut colors, the operations are performed in the native linear color space of the

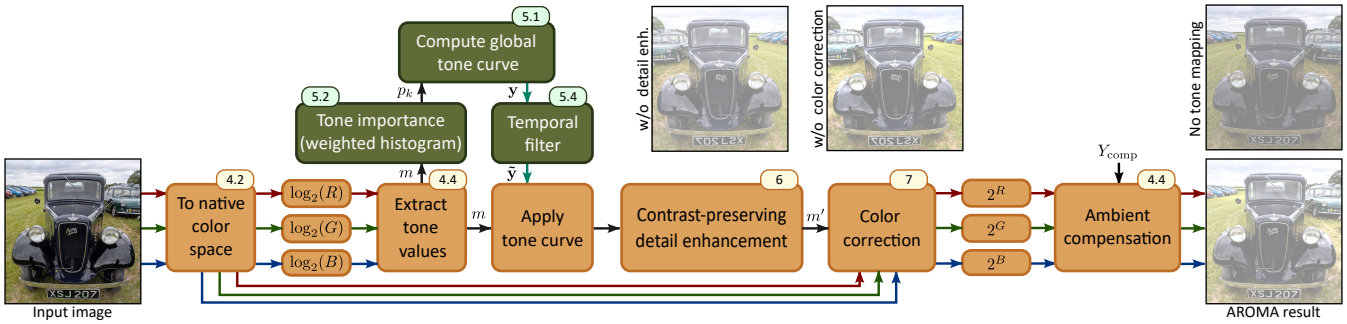


Fig. 2. The *AROMA* processing pipeline. The numbers above the processing blocks point to the section where the processing is explained. The resulting image (bottom-right) and the images at the top row simulate appearance on an AR display that is affected by ambient light.

display. In the first step, we convert RGB color values from the input color space (such as BT.709) to the native space:

$$I^{\text{native}}(x,y) = M_{\text{BT709-to-native}} \cdot I^{\text{BT709}}(x,y)\,, \tag{4}$$

where $M_{\text{BT709-to-native}}$ is a 3x3 color transformation matrix. In practice, this matrix is computed from the display's native primaries and white point, obtained either by colorimeter measurement or from the panel manufacturer's specification. The pixels $I^{\text{native}}(x,y)$ and $I^{\text{BT709}}(x,y)$ are represented as column vectors. Note that both pixel values are represented in absolute units, for example, $I^{\text{BT709}}(x,y) = \begin{bmatrix}100 & 100 & 100\end{bmatrix}^T$ denotes a display white (e.g., D65) that has a luminance of 100 cd/m^2. Note that $I^{\text{BT709}}(x,y)$ is a linear color space (without sRGB or gamma non-linearity). For SDR content, we assume that the whitepoint (the highest pixel value) has a luminance of 200 cd/m^2 [ITU-R 2022].

4.3 Logarithmic encoding

To operate on values that are more perceptually uniform [Mantiuk et al. 2015, Sec. 2.4], we convert the native RGB values into logarithmic values:

$$l_c(x,y) = \log_2\left((1-y_{\text{smin}})\frac{I_c^{\text{native}}(x,y)}{Y_{\text{smax}}} + y_{\text{smin}}\right) \quad c\in\{R,G,B\}\,, \tag{5}$$

where Y_{smax} is the maximum *absolute* value of the source content, which is 200 cd/m^2 for SDR and could be higher for HDR content. y_{smin} is the minimum *relative* value of the source content and is necessary to avoid singularity when computing the logarithm of color values close to 0. Its value depends on the bits allocated to store the logarithmic values. If N bits are available we set it to:

$$y_{\text{smin}} = \frac{1}{2^N - 1}\,. \tag{6}$$

We will use upper case symbols to denote absolute color values, and lower case for relative. Because $l_c(x,y)$ values are relative, they will be in the range from $\log_2(y_{\text{smin}})$ to 0.

4.4 Tone values

Instead of operating on RGB triplets, both global and local tone mapping operate on tone values. There are multiple strategies for computing tone values from the RGB triplet: "the maximum of RGB (which avoids clipping and enhances color contrast), a weighted average, or the mean of RGB. We found that using maximum causes gray tones to appear brighter than pure colors, causing gray levels to "glow" in the resulting images. To avoid this issue and retain some benefits of the maximum-RGB tone, we use a mix of maximum and pseudo-luminance to compute the tone:

$$m(x,y) = 0.5\max\{l_{\text{R}}, l_{\text{G}}, l_{\text{B}}\} + 0.5(w_{\text{R}}\,l_{\text{R}} + w_{\text{G}}\,l_{\text{G}} + w_{\text{B}}\,l_{\text{B}}), \tag{7}$$

where $w_{\text{R,G,B}} = \begin{bmatrix}0.213 & 0.715 & 0.072\end{bmatrix}$.

We found that in some cases, the formula above results in color clipping, as the luminance component can make some values exceed the allowed maximum. A simple solution to this problem is to use maximum for the brightest tones, luminance for the darkest tones, and interpolate in between. Therefore, Eq. (7) becomes:

$$\begin{aligned} m(x,y) &= \alpha\ \max\{l_{\text{R}}, l_{\text{G}}, l_{\text{B}}\} + (1-\alpha)(w_{\text{R}}\,l_{\text{R}} + w_{\text{G}}\,l_{\text{G}} + w_{\text{B}}\,l_{\text{B}}),\\ \text{where } \alpha &= \left[\frac{\max\{l_{\text{R}}, l_{\text{G}}, l_{\text{B}}\}}{N} + 1\right]_0^1 \end{aligned} \tag{8}$$

The denominator in the second line of this equation (N) is the dynamic range of the input signal, expressed in stops (see Eq. (6)). Adding one ensures that the α values are in the range 0 to 1. Additional clipping between 0 and 1 ($[]_0^1$) is a precaution added in case the input signal exceeds the expected dynamic range. The tone extraction from Eq. (8) makes a particular difference for HDR content. It makes the resulting images brighter and more "punchy" in terms of colors.

4.5 Tone mapping

Tone values are transformed by the global tone curve $t()$ and detail enhancement $d()$ to post-mapping tone values:

$$m'(x,y) = d(t(m(x,y))) \tag{9}$$

These functions will be explained in detail in Section 5 and Section 6. The color is restored from the tone-mapped values using the color correction formula from Mantiuk et al. [2009].

$$l'_c(x,y) = (l_c(x,y) - m(x,y))\cdot s(m(x,y)) + m'(x,y)\,, \tag{10}$$

where $s()$ is the color correction function explained in Section 7. This is equivalent to the original color correction formula (see Eq. 2 in Mantiuk et al. [2009]) computed in the logarithmic domain.

Tone-mapped relative color values are restored by inverting Eq. (5):

$$i'_c(x,y) = \frac{2^{l'_c(x,y)} - y_{\text{smin}}}{1 - y_{\text{smin}}}\,. \tag{11}$$

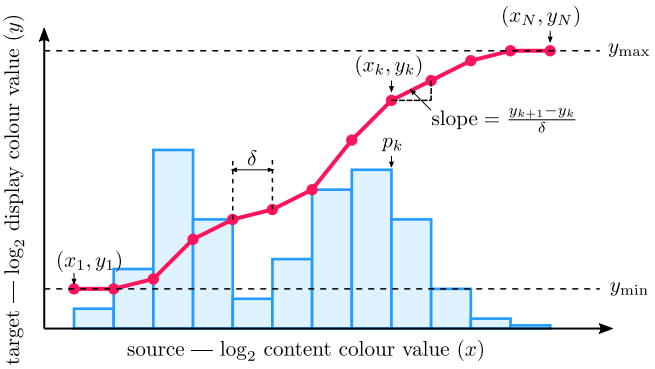


Fig. 3. An illustrative example of a tone curve showing the notation.

4.6 Ambient light compensation

The final step involves mapping the relative values to the absolute values of light emitted from the display and compensating for the ambient light. The linear intensities emitted from the display are:

$$I_c''(x,y) = \left[(Y_{\text{dmax}} + Y_{\text{comp}})i_c'(x,y) - Y_{\text{comp}}\right]_{Y_{\text{dmin}}}^{Y_{\text{dmax}}} + Y_{\text{amb}}\,, \quad (12)$$

where Y_{comp} is the amount of light (luminance) we are compensating for. $]_{Y_{\text{dmin}}}^{Y_{\text{dmax}}}$ is the clamping operator that restricts the values between the maximum intensity value for the display Y_{dmax} and its black level Y_{dmin}. This clamping is necessary as the detail enhancement step can push the values outside this range.

Finally, we compute an image scaled between 0 and Y_{dmax} (instead of Y_{dmin} to Y_{dmax}), without ambient light Y_{amb} to pass to further stages of display processing. We rescale the resulting linear values:

$$I_c'''(x,y) = Y_{\text{dmax}} \frac{I_c''(x,y) - Y_{\text{amb}} - Y_{\text{dmin}}}{Y_{\text{dmax}} - Y_{\text{dmin}}}\,. \quad (13)$$

5 Global tone curve

A key goal of tone mapping is modifying luminance and contrast so that resulting color values fit within the display's peak brightness and dynamic range. The tone curve is responsible for these tasks.

Figure 3 shows an example of a tone curve, which maps input color values x (representing $\log_2$ of red, green or blue) into the output color values on the display, y. The target values must lie between the (relative) display peak value, $y_{\max}$, and its black level, $y_{\min}$. *AROMA* uses a piecewise linear function with uniformly distributed nodes on the x-axis. The slope of the tone curve, $(y_{k+1}-y_k)/\delta$, has an important meaning — a slope equal to 1 (on a log-log scale) means that the contrast of the image is unchanged. Slopes greater than 1 enhance contrast, and slopes less than 1 reduce contrast. Ideally, we would like to keep the slope close to 1, as one of the objectives of our tone mapping is to faithfully preserve the source contrast.

Not every color value in the input image has the same importance when reproducing an image on a display. Color values that are poorly represented or absent in an image can be mapped to a much lower contrast than those that are abundant. Such an "importance function" can be computed using a histogram of pseudo-grayscale values, but better results can be obtained by considering more advanced measures of "importance" (see Section 5.2).

5.1 Tone curve as a constrained optimization problem

To maintain curve slopes close to 1 (preserve contrast) and the mapped color values y close to the original values x (preserve brightness), we can express our optimization problem as:

$$\begin{aligned} \boldsymbol{y}^* = \underset{\boldsymbol{y}}{\text{argmin}} & \sum_{k=1}^{N-1} \underbrace{p_k \left(1 - \tfrac{y_{k+1}-y_k}{\delta}\, c_{\text{R}}(Y)\right)^2}_{\text{preserve contrast}} \\ & + \lambda \sum_{k=1}^{N} \underbrace{p_k (y_k - x_k)^2}_{\text{preserve brightness}} \\ \text{s.t.} \quad & y_{k+1} \geq y_k, \quad y_1 = y_{\min}, \quad y_N = y_{\max} \end{aligned} \quad (14)$$

The constraints ensure that the resulting tone curve is monotonically increasing and the values are within the allowed range between the display black level ($y_{\min}$) and its peak ($y_{\max}$). Because we are only partially compensating for the ambient light (see Section 4.1), darker tones will be affected by the non-compensated portion of the ambient. We account for that with the term c_{R}:

$$c_{\text{R}}(Y) = \frac{Y}{Y + Y_{\text{amb}} - Y_{\text{comp}}}\,, \quad (15)$$

which is computed for the absolute luminance of the tone curve node Y. The derivation can be found in [Kim et al. 2025, Sec.2].

The p_k values represent tone importance, which can be a histogram of tone values or a more complex measure. The coefficient λ controls the trade-off between preserving contrast and brightness and is typically set to give higher priority to preserving contrast (set to $\lambda = 0.01$ in our experiments).

The optimization problem in Eq. (14) can be efficiently solved (without the brightness component) following [Eilertsen et al. 2015]. However, this problem formulation only preserves physical contrast rather than perceived contrast.

Perceived contrast. It has been shown that contrast at lower luminance levels is perceived as smaller than that at higher luminance levels [Georgeson and Sullivan 1975; Kulikowski 1976; Peli et al. 1991], i.e., physical contrast at low luminance must be increased to match contrast at higher luminance (see Peli et al. [1991], Fig.3a). Recent work by Kim et al. [2025] identified historic models used to model perceived supra-threshold contrast, and demonstrated their applicability to modeling visibility in AR. We employ the model by Peli et al. [1991], which assumes matching ratios of physical contrast and detection thresholds:

$$\frac{c(Y_1)}{c_{\text{T}}(Y_1)} = \frac{c(Y_2)}{c_{\text{T}}(Y_2)}\,, \quad (16)$$

where $c(Y)$ is a physical contrast presented at luminance Y and $c_{\text{T}}(Y)$ is the contrast detection threshold, also at luminance Y.

To obtain a tone curve minimizing perceived, rather than physical contrast distortion, we can incorporate Eq. (16) into the minimum contrast and brightness optimization problem in Eq. (14):

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmin}} \underbrace{\sum_{k=1}^{N-1} p_k \left(\frac{\pi(1, Y_{\mathrm{src},k}) - \pi\left(\frac{y_{k+1}-y_k}{\delta}\, c_{\mathrm{R}}(Y_{\mathrm{disp},k}), Y_{\mathrm{disp},k}\right)}{w_k} \right)^2}_{\text{preserve contrast}} + \lambda \underbrace{\sum_{k=1}^{N} p_k (y_k - x_k)^2}_{\text{preserve brightness}}$$
$$\text{s.t.} \quad y_{k+1} \geq y_k, \quad y_1 = y_{\min}, \quad y_N = y_{\max} \tag{17}$$

where $\pi(\cdot, Y)$ is the perceived contrast at luminance Y. Assuming Peli's supra-threshold contrast perception model, we have:

$$\pi(c, Y) = \frac{c}{c_{\mathrm{T}}(Y)} \tag{18}$$

where c is the physical contrast, and $c_{\mathrm{T}}(Y)$ is the threshold contrast at luminance Y. $Y_{\mathrm{src},k}$ in Eq. (17) is the luminance of the tone k in the source, and $Y_{\mathrm{disp},k}$ is the luminance of the tone on the display:

$$Y_{\mathrm{src},k} = Y_{\mathrm{smax}}\, 2^{x_k}, \quad Y_{\mathrm{disp},k} = Y_{\mathrm{dmax}}\, 2^{y_k}, \tag{19}$$

To find the threshold contrast $c_{\mathrm{T}}(Y)$ in Eq. (18), we find the peak of achromatic contrast sensitivity at a given luminance from castleCSF [Ashraf et al. 2024] and invert the value to obtain the detection threshold. The peak sensitivity is a proxy for the detection of sharp edges [Ashraf et al. 2023]. The weights w_k are introduced in Eq. (17) as the normalization term that ensures that the supra-threshold detection model does not change the importance of different tones. The weight is set to the geometric mean:

$$w_k = \sqrt{\frac{1}{c_{\mathrm{T}}(Y_{\mathrm{src},k})\, c_{\mathrm{T}}(Y_{\mathrm{disp},k})\, c_{\mathrm{R}}(Y_{\mathrm{disp},k})}} \tag{20}$$

Incorporating Eq. (18) into Eq. (17) gives us:

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} \sum_{k=1}^{N-1} p_k \left(\frac{1}{w_k\, c_{\mathrm{T}}(Y_{\mathrm{src},k})} - \frac{(y_{k+1} - y_k)\, c_{\mathrm{R}}(Y_{\mathrm{disp},k})}{w_k\, \delta\, c_{\mathrm{T}}(Y_{\mathrm{disp},k})} \right)^2 + \lambda \sum_{k=1}^{N} p_k (y_k - x_k)^2, \quad \text{s.t. } y_{k+1} \geq y_k,\ y_1 = y_{\min},\ y_N = y_{\max}. \tag{21}$$

Figure 4 shows examples of tone curves optimized for ambient light levels varying from 0.1 cd/m^2 to 20 cd/m^2 and a display with peak luminance $Y_{\mathrm{dmax}} = 200$ cd/m^2. As ambient light increases, tone curves become steeper at lower tones to compensate for the added light. This will also brighten images, as brighter tones better reproduce contrast when the display is affected by ambient light.

5.2 Tone importance

The tone importance function is responsible for assigning weights to segments of the piecewise linear tone curve, adapting its shape to image content. In practice, the tone importance function is used to compute the coefficients p_k used in Eq. (14), Eq. (17) and Eq. (21).

A basic tone importance function consists of an image histogram, which works well in practice in most cases. However, if an image contains a large, flat area, the image histogram assigns very high importance to that area. Flat areas are rarely key scene elements, and it is undesirable to reproduce them at higher contrast at the cost

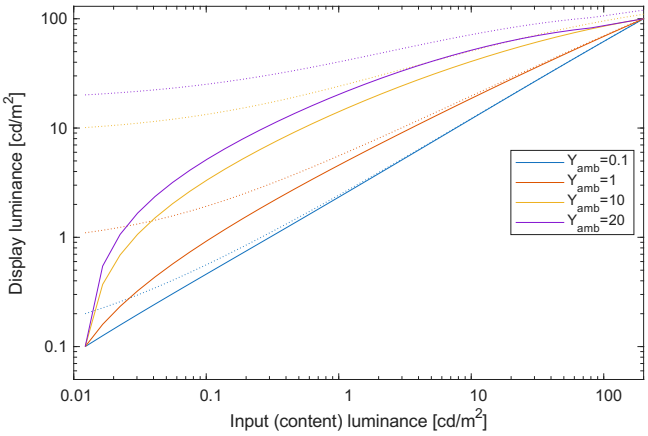


Fig. 4. Global tone curves optimized for different ambient light levels for $Y_{\mathrm{amb}} \in \{0.1, 1, 10, 20\}$ cd/m^2. The continuous lines are the tone curves with ambient light compensation (with Y_{comp} subtracted, see Eq. (12)), and the dotted lines represent the effective luminance of tones on the display with ambient light added ($+Y_{\mathrm{amb}}$). The curves were plotted assuming a flat importance function (p_k = const).

of other image regions. A simple extension of an image histogram that addresses this problem is a histogram weighted by the local contrast of the frame [Eilertsen et al. 2015]. Local contrast can be efficiently calculated as the local (windowed) standard deviation:

$$c(x, y) = \sqrt{(g_\sigma * m^2)(x, y) - ((g_\sigma * m)(x, y))^2}, \tag{22}$$

where g_σ is a window kernel, such as Gaussian function, m is the image (with tone values) and (x, y) are the pixel coordinates.

Computing a Gaussian-filtered image is too expensive for AR HMD tone mapping. Instead, following [Wolski et al. 2022], we compute local contrast via a MIP-map, which stores $m(x, y)$ in one channel, denoted as $M_1^{(n)}$, and $m^2(x, y)$ in another channel, denoted as $M_2^{(n)}$ (for the n-th level of the MIP-map). We approximate local contrast using the n-th MIP-map level ($n = 5$ in our implementation):

$$c(x, y) \approx \sqrt{M_2^{(n)}(x, y) - \left(M_1^{(n)}(x, y)\right)^2}. \tag{23}$$

Then, each pixel contributes $c(x, y)$ to the bin containing $m(x, y)$ (instead of 1), the histogram is normalized to sum 1, and used as tone importance p_k.

5.3 Histogram shaping

The (weighted) histogram computed from the image may result in large differences between bins, introducing significant mismatches in tone-curve slopes between segments. This can be reduced by computing fractional exponents of bin importance values. *AROMA* uses the following manually tuned exponents:

$$p'_k = \begin{cases} p_k^{\frac{1}{2}} & \text{for} \quad k = 1, \ldots, K-5 \\ p_k^{\frac{1}{4}} & \text{for} \quad k = K-4, \ldots, K \end{cases}. \tag{24}$$

The 4th root used for the top 5 bins ensures these bins have higher importance. This reduces the chance of clipping when details are enhanced (see Section 6).

5.4 Temporal filter

When a tone curve is computed per frame, it can change its shape rapidly, causing disturbing flicker artifacts. A simple way to address this problem is to filter the shape of the tone curve over time.

The filtering function F is applied to the final tone curve:

$$\tilde{y}_k = F(y_k^{(f)}, y_k^{(f-1)}, \ldots), \quad k = 1, \ldots, K \tag{25}$$

where $y_k^{(f)}$ are the tone-mapped tone values from Eq. (21) and the superscript (f) denotes the frame index, with f being the current frame. We use only current and past frames: *AROMA* runs at the display level, downstream of graphics processing, so future frames are unavailable even for pre-rendered content, and ambient conditions can change unpredictably between frames.

We aim to design a temporal filter that minimizes the noticeable changes in the tone curve over time. The sensitivity of the visual system peaks between 4 and 11 Hz, depending on the luminance, and is relatively low below 0.5 Hz (see Cai et al. [2024], Figure 8(a)). To make tone-curve changes as imperceptible as possible, we design a filter that reduces frequency components above 0.5 Hz. Because sensitivity depends on the signal frequency in Hz, the filter needs to be derived for each video frame rate.

In Figure 5, we compare four filters that were designed to filter frequencies above 0.5 Hz. The top two rows show inexpensive filtering options: the box and leaky integrator filters. Both options allow some of the signal above 0.5 Hz to pass. However, as shown in the right column of Figure 5, the box filter achieves the desired signal level faster. In the bottom two rows, more computationally expensive alternatives, FIR and IIR, are shown. These filters are better at attenuating higher frequencies, but also introduce a larger phase delay (i.e., reach the original signal level later) than the box filter. Additionally, the FIR filter requires a large support (60 taps at 30 fps), and the IIR filter introduces overshoots in the filtered signal.

Given the simplicity of the box filter, the possibility of efficient hardware implementation, and good performance during testing, we employ it to filter tone curves in *AROMA* . To reach the first crossing at 0.5 Hz, as shown in Figure 5, the width of the box filter must be equal to the number of frames per second.

6 Contrast-preserving detail enhancement

Detail enhancement is meant to preserve high-frequency details, which could be lost due to the intrusion of ambient light or contrast compression in global tone mapping. The enhancement is performed on the tone values after global tone mapping (see Eq. (9)). The enhancement is meant to preserve the original image contrast while considering both the ambient light and how global tone mapping affected that contrast.

Tone mapping techniques with such detail enhancement rely on edge-stopping filters [Durand and Dorsey 2002] or anisotropic diffusion [Eilertsen et al. 2015] to separate the base (low frequency) signal from the detail (high frequency). This separation allows us to enhance details (high frequencies) while preserving the base (low frequencies). This operation is equivalent to unsharp masking, but uses more advanced filters. However, both of these advanced filters are computationally expensive and may not be easily implemented in hardware. Instead, we reuse the downsampled versions of the input image stored in a MIP-map, also needed to compute the tone importance map (Eq. (23)). In our implementation 32× downsampling was employed (corresponding to the 5th level of the MIP-map). This image is then enlarged back to the original resolution using bilinear interpolation. Using the notation from Section 5.2, we introduce into the MIP-map a 3rd channel with tone-mapped values $t(m(x,y))$ and denote it as $M_{\mathrm{t}}^{(n)}$. Then, we can compute the base and detail layers as:

$$m_{\mathrm{base}}(x,y) = M_{\mathrm{t}}^{(n)}(x,y) \tag{26}$$

$$m_{\mathrm{detail}}(x,y) = t(m(x,y)) - m_{\mathrm{base}}(x,y) \tag{27}$$

where $n = 5$, $t()$ is the global tone mapping operator and m is the map with tone values (see Eq. (9)). To obtain the enhanced image, we combine the base and detail with an adaptive gain factor $g()$:

$$m'(x,y) = d(t(m(x,y))) = m_{\mathrm{base}}(x,y) + g(m(x,y))\, m_{\mathrm{detail}}(x,y)\,. \tag{28}$$

We need to find the value of gain $g()$ that preserves contrast. To simplify notation, we will omit pixel coordinates (x,y). We express the local contrast in the original image as:

$$C = \frac{\Delta Y}{Y} \tag{29}$$

where Y is luminance and ΔY is a luminance increment. If the image is affected by the ambient light Y_{a}, its local contrast becomes

$$C_{\mathrm{a}} = \frac{\Delta Y}{Y + Y_{\mathrm{a}}} \tag{30}$$

Because ΔY is the difference between two luminance levels, Y_{a} is canceled out in the numerator. This contrast is further affected by the global tone mapping. Because our tone curve is defined in the logarithmic space, its slope (derivative) corresponds to the amount of contrast compression. We can express the contrast affected by the combined effect of ambient light and tone mapping as

$$C_{\mathrm{gtmo}} = \frac{\partial t}{\partial m}(m)\, \frac{\Delta Y}{Y + Y_{\mathrm{a}}} \tag{31}$$

where $\frac{\partial t}{\partial m}(m)$ is the derivative of the tone curve at the tone value m. Finally, to compensate for the loss of contrast due to ambient light and global tone mapping, we need to multiply the details by the ratio of the original contrast (Eq. (29)) and the contrast affected by tone mapping and ambient light (Eq. (31))

$$g_{\mathrm{u}}(m) = \frac{C}{C_{\mathrm{gtmo}}} = \frac{Y + Y_{\mathrm{a}}}{Y\, \frac{\partial t}{\partial m}(m)}\,, \tag{32}$$

where the absolute luminance, Y, is computed from the tone value of the tone-mapped image

$$Y = Y_{\mathrm{dmax}}\, 2^{t(m)}\,. \tag{33}$$

Note that ambient light Y_{a} accounts for only the portion of the ambient light that has not been compensated by the tone curve: $Y_{\mathrm{a}} = Y_{\mathrm{amb}} - Y_{\mathrm{comp}}$ (see Section 4.1). The contrast gain from Eq. (32) can have very high values, especially in dark image regions. Such

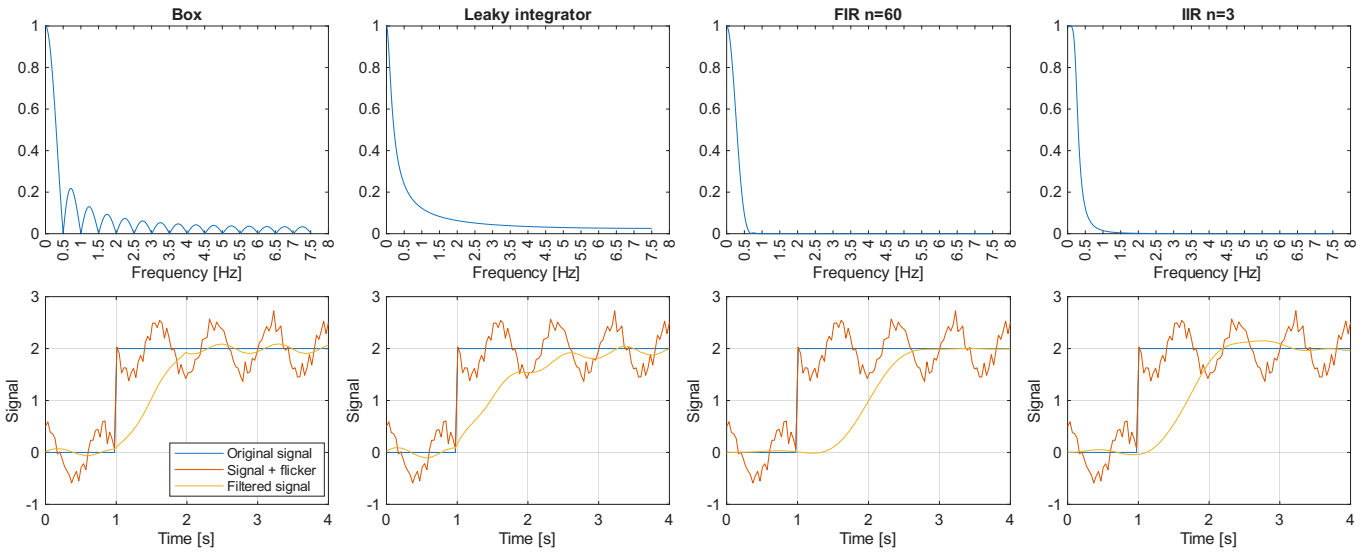


Fig. 5. Comparison of four temporal filters. Top: the frequency response. Bottom: illustration of how each filter removes both low-frequency (1.25 Hz) and high-frequency (10 Hz) flicker. The original signal jumps in value from 0 to 2 at t=1 s (blue line), and is affected by a flicker (red line). The filtered signal is shown in orange. The filters were designed to be low-pass, with the cut-off frequency of 0.5 Hz.

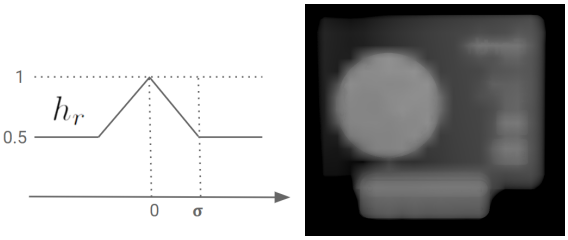


Fig. 6. Left: a typical shape of the range kernel used for the edge-stopping upsampling filter. Right: the result of edge-stopping upsampling.

large values could result in over-enhancement and artifacts. To prevent that, we set the maximum amount of gain to be $g_{\max}$

$$g(m) = \min(g_{\mathrm{u}}(m), g_{\max}) \,. \tag{34}$$

$g_{\max}$ typically varies between 1 (no enhancement) and 1.6 (60% enhancement). In our experiments, we set $g_{\max} = 1.3$ to restore compressed details without visibly amplifying noise or halos.

Note that the contrast equations above assume Weber contrast units, while the detail layer (containing local contrast) is represented in log-contrast units. However, the formulation is equivalent as both units of contrast are approximately linearly related for small and medium contrast, which dominates image content.

6.1 Edge-stopping upsampling

Ideally, we would like to replace the base layer computed from the MIP-map in Eq. (26) with an edge-stopping filter, which could improve detail enhancement. A bilateral filter [Durand and Dorsey 2002; Tomasi and Manduchi 1998] is too expensive to compute and its approximations [Chen et al. 2007; Gastal and Oliveira 2011; Paris and Durand 2009] are too complex for real-time processing on wearable devices. Instead, we approximate the bilateral filter using the data structures that are already present in our processing pipeline. When upsampling tone-mapped values from the coarse pyramid level, $M_{\mathrm{t}}^{(n)}$, $n = 5$, to the full-resolution low-pass image, m_{base}, we modulate the interpolation weight by the range kernel h_r:

$$\begin{aligned} w_{uv}(x,y) &= h_s(x-u, y-v)\, h_r\Big(t(m(x,y)) - M_{\mathrm{t}}^{(n)}(u,v)\Big), \\ m_{\mathrm{base}}(x,y) &= \frac{\sum_{\{u,v\}\in\Omega} w_{uv}(x,y)\, M_{\mathrm{t}}^{(n)}(u,v)}{\sum_{\{u,v\}\in\Omega} w_{uv}(x,y)}\,, \end{aligned} \tag{35}$$

where $w_{uv}(x,y)$ is the bilateral weight, h_s is a bilinear spatial filter (a triangular function), and h_r is the range filter, plotted on the left of Figure 6. The range kernel assigns higher values to the MIP-map texels that are close in value to the currently processed pixel $m(x,y)$. Note that the filter function does not reach 0 as that would result in overly strong edge-stopping behavior and visible artifacts. Ω are the four texels from the coarse pyramid levels that are used for bilinear interpolation. The right part of Figure 6 shows the filtered m_{base} for a frame containing UI elements with sharp edges.

7 Color correction

Color correction reduces saturation in tone-mapped images (see Figure 7). We use the method by Mantiuk et al. [2009],

$$I'_c(x,y) = \left(\frac{I_c(x,y)}{Y_{\mathrm{in}}(x,y)}\right)^s Y_{\mathrm{out}}(x,y)\,, \tag{36}$$

where s is the parameter reducing saturation, Y_{in} is the luminance before and Y_{out} is the luminance after tone mapping. They measured the empirical values of s that resulted in the best color reproduction, and approximated them as a function of the contrast compression

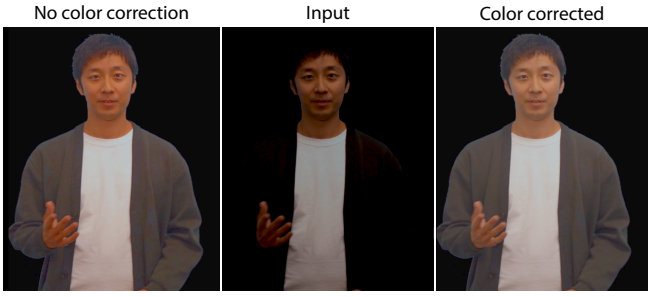


Fig. 7. Color correction is necessary to produce adequate results. The original image of the man (M) is mapped, reducing contrast by 30%. When only naive mapping is applied (L), oversaturated tones produce an unrealistic reddish skin tone, which is improved using color correction, as described in Section 7 (R). Image credit: Chapiro et al. [2024].

factor γ

$$s(\gamma) = \frac{(1+k_1)\,\gamma^{k_2}}{1+k_1\,\gamma^{k_2}} \tag{37}$$

where $k_1 = 1.6774$ and $k2 = 0.9925$.

Here, we adapt their formula to work with arbitrary tone curves. First, we assume that the contrast compression factor corresponds to the slope of the given segment of a tone curve. Second, to avoid abrupt changes in color saturation between segments, we smooth the slopes (contrast values) across segments using a 21-tap triangle filter. Finally, Eq. (36) is computed on the logarithmic pixel values and becomes (expanded from Eq. (10)):

$$l'_c(x,y) = (l_c(x,y) - m(x,y))\cdot s(d_t(m(x,y))) + m'(x,y)\,, \tag{38}$$

where d_t is a filtered tone curve derivative at tone value $m(x,y)$.

Transparent backgrounds. *AROMA* gives special treatment to black pixels, which are effectively transparent in AR when overlaid with the real-world background. AR content may often be sparse, for example containing a picture or UI element surrounded by black (transparent) pixels, as in the example in Figure 7. When zero values are close to bright objects, they create very high contrast, resulting in excessive and unnecessary sharpening of object boundaries. To avoid this, we discard zero-valued pixels from analysis by *AROMA* , maintaining their value at zero throughout. Including zero-valued pixels in the global histogram heavily skews the optimized tone curve, as the extreme dark-vs-bright transitions dominate the contrast budget. In these cases this solution significantly improved rendering, although it may lead to reduced object edge contrast.

8 Validation against a recent OST-AR tone mapper

Recent work by Chen et al. [2026] evaluated the outcomes of visual difference metrics for tone mapping, and proposed an extension of ColorVideoVDP [2024] which improved its performance on multiple datasets by admitting *separate reference and test display photometries*. Because this metric operates on photometric quantities, it can also be applied to images with simulated AR ambient conditions. We simulated AROMA's predicted performance against a recently published optical-see-through tone mapper [Qian et al. 2024].

Setup. The same test image set described in the subjective study in Section 9 was used. The 9 images were rendered at peak luminance of $L_P = 200$ cd/m^2 under six ambient levels at the eye, $y_{\text{amb}} \in \{5, 10, 20, 30, 40, 50\}$ cd/m^2, and $\text{Y}_{\text{black}} = 0.1$ cd/m^2. For each cell $(scene, y_{\text{amb}})$, we compared four mapping options: (i) *none* (no mapping); (ii) Qian et al.'s TMO with Γ optimized per cell from L_P, L_R and CR disabled (matching their Fig. 5b headline configuration); (iii) AROMA's *global* tone curve only; and (iv) AROMA's *full* pipeline (global tone curve + local enhancement). All four conditions were scored against the same reference: an unmapped source image shown on a matched 200-nit, no-ambient display (representing traditional high-quality presentation).

Result. Table 1 summarizes the results, and Figure 8 shows a visual example. *AROMA full* achieves the highest mean JOD at every ambient level, *AROMA global* is consistently second, with Qian et al. [2024] and *None* scoring similarly. *AROMA* provides a significant improvement over the method of Qian et al. [2024] (mean per-pair ΔJOD between +0.63 at $y_{\text{amb}} = 5$ and +0.73 at $y_{\text{amb}} = 50$ for the full method), likely due to the additional flexibility offered by our content- and ambient-aware mapping process. Evaluating higher ambient luminances (100-400 cd/m^2) yielded the same trends, but these conditions produce very low additive contrasts. In practice, AR displays raise display luminance or attenuate ambient (world dimming) to avoid extreme contrast loss.

Table 1. Per-condition CVVDP-TM JOD scores (± 1 SD, $n = 9$ images). Higher is better; 10.0 is perfect fidelity. **Best** and second-best per row marked.

y_{amb} (cd/m^2)	None	Qian et al. [2024]	AROMA *global*	AROMA *full*
5	7.83 ± 0.28	7.72 ± 0.30	8.14 ± 0.43	**8.35 ± 0.55**
10	7.25 ± 0.26	7.35 ± 0.25	7.73 ± 0.38	**8.02 ± 0.57**
20	6.60 ± 0.41	6.80 ± 0.29	7.17 ± 0.39	**7.50 ± 0.51**
30	6.18 ± 0.53	6.38 ± 0.39	6.79 ± 0.47	**7.10 ± 0.48**
40	5.87 ± 0.62	6.06 ± 0.50	6.50 ± 0.55	**6.79 ± 0.48**
50	5.64 ± 0.70	5.80 ± 0.59	6.26 ± 0.61	**6.53 ± 0.50**

9 Subjective validation

To test the efficacy of *AROMA* on a physical AR setup, we conducted a subjective study. The goal was to measure the quality of the images shown on an AR display affected by ambient light when presented unprocessed, with only the global portion of the method, and with the full *AROMA* algorithm. As we measured quality at different display luminance levels and ambient light levels, the results allow us to evaluate a quality-equivalent increase in display luminance when employing *AROMA* . In total, the study collected pairwise-comparison data from 50 observers (10 pilot, 40 main) across 9 source images, 2 background textures, 3 background luminances, 5 additive contrasts, and 3 mapping methods, with each condition rated by approximately 13 observers.

Experiment Setup. We employed a custom AR test bed (pictured in Figure 9). The virtual image is produced by two 31.1" Eizo CG3146 displays viewed through beam splitters. Placed at 86 cm away, they produced a field-of-view of $46^\circ \times 26^\circ$ and resolution of 45 ppd (pixels



Fig. 8. Visual comparison at $y_{amb} = 10$ and 30 cd/m^2. Left to right: reference (no mapping, no ambient), *No mapping* (with ambient), Qian et al. [2024], *AROMA global*, *AROMA full*. Insets: aggregated CVVDP-TM JOD score against the reference (higher is better; 10.0 is perfect fidelity; error bars show standard deviation).

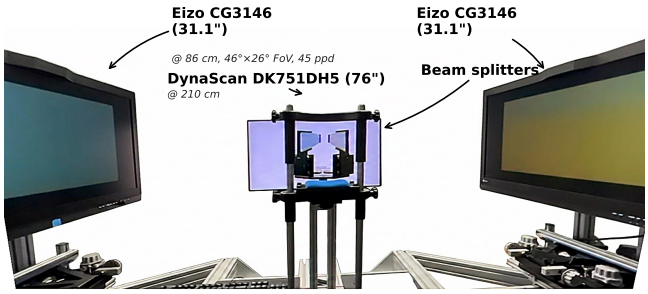


Fig. 9. The custom AR haploscope test bed used for our subjective validation. Two side-mounted 31.1" Eizo CG3146 displays produce the virtual image at 86 cm via beam splitters, optically combined with a 76" DynaScan DK751DH5 background TV at 210 cm.

per degree). The background plane was created by a 76" DynaScan DK751DH5 TV at 210 cm. All three displays were calibrated from the viewer's position using a CS2000-A spectroradiometer, and optical paths were aligned to ensure sub-pixel geometric positioning accuracy. This custom testbed allows us to maintain finely controlled conditions for the stimuli, and avoid confounding artifacts due to product-specific constraints, such as waveguide non-uniformity.

Implementation. Software was implemented in Matlab, using the Psychtoolbox package [Kleiner et al. 2007], which was used to display stimuli, handle user interaction, and dispatch OpenGL shaders that processed input frames.

Participants. Following a pilot study ($N = 10$), 40 participants took part in the main study. Participants signed informed consent, the experiment was approved by a third party IRB. Demographics were balanced in terms of age and gender, with information subsequently discarded. Experimental sessions lasted an average of 46 minutes. Participants were screened for normal or corrected-to-normal vision, and passed an Ishihara color test.

Stimuli and Conditions. The stimuli consisted of one of 9 images on the virtual plane (simulating an AR display, shown in Figure 10), optically combined with one of 6 static patterns shown on the background display (see Figure 11). AR display brightness is commonly discussed in the context of *additive contrast* (see Eq. (2)), as display backlight luminance is modulated to produce a given contrast relative to the background luminance measured by an ambient light sensor. The values used in our study are shown in Table 2. We compared three separate mapping techniques: (i) **do nothing** did not employ any tone mapping; (ii) **global only** used the tone curve

Table 2. Display maximum luminance for each studied condition

Bkg. luminance \ Additive contrast	3:1	4:1	6:1	8:1	10:1
3 nits	6 nits	9 nits	15 nits	21 nits	27 nits
10 nits	20 nits	30 nits	50 nits	70 nits	90 nits
60 nits	120 nits	180 nits	300 nits	420 nits	540 nits

derived in Section 5; (iii) **global+local** used the tone curve and the local detail enhancement described in Section 6. Both (ii) and (iii) use the color correction method described in Section 7.

The chosen background-luminance levels (3, 10, and 60 nits) span the 25th, 50th, and 75th percentiles of an egocentric indoor-luminance dataset [Zhang et al. 2023]. Outdoor scenes may present higher luminances, but commercially available AR headsets (e.g. Microsoft HoloLens, Meta Ray-Ban Display, Magic Leap 2) incorporate world-dimming elements typically active in outdoor conditions to increase the visibility of the virtual image. For background texture patterns, our study follows prior art (Chapiro et al. [2024]) using statistically representative cases, maximizing possible coverage within a feasible study design. The *flat* pattern is used as a proxy for simple backgrounds, and a $1/f$ noise pattern serves as a statistically representative sample of frequencies present in natural scenes.

Each participant completed one session of the experiment, which consisted of 3 out of 9 base stimuli, totaling: 3 (stimuli) × 2 (background textures) × 3 (background luminance) × 5 (additive contrast) × 3 (mapping methods) = 270 conditions. Each individual condition was rated at least $40/3 \approx 13$ times.

Task. A pairwise 2-interval-forced-choice (2IFC) comparison protocol was used: subjects were instructed to "select the test image which most resembles the reference in terms of brightness and contrast", following a comprehensive training session in which they were instructed to examine the entire scene, including bright and dark areas, and were familiarized with potential artifacts that can arise from tone mapping or its absence. Subjects always saw the reference image first, and could not make a selection before viewing both test images, presented in a random order. ASAP active sampling [Mikhailiuk et al. 2021] was used to obtain accurate quality-score estimates with roughly N trials per subject for N conditions, rather than the $\binom{N}{2}$ pairs of a full pairwise design. The *reference* image was the original source content, scaled to a peak luminance of 500 cd/m^2 and shown on the haploscope's virtual-image display with the background DynaScan blanked (so no ambient light reached the viewer) and no tone mapping applied; this represents the intended, non-AR appearance of the content. Each *test* image used the same source content, processed by one of the three tone mapping conditions (*do*



Fig. 10. 6:1 additive-contrast preview using true $Y_{amb} = 10$ nits, $Y_{peak} = 50$ nits render outputs. Face images exemplify augmented calling and digital-human applications (credit: Chen et al. [2025]); real-world content exemplifies professional or user-generated media (credit: Song et al. [2016]). Three additional productivity stimuli were included in the study but are not shown. The display-relative outputs are previewed with white at 60 nits, matching the study-style 6:1 condition; each mapped cell is split diagonally between flat and $1/f$ noise backgrounds. T

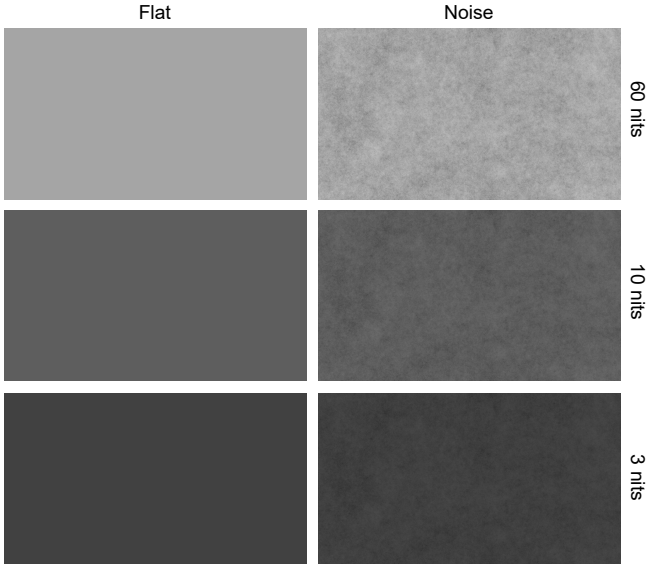


Fig. 11. The backgrounds used in our subjective validation experiment. Two textures were used: a flat image, and a $1/f$ noise pattern, following Chapiro et al. [2024]. Each was presented at one of three luminance values: 3, 10, and 60 nits, roughly corresponding to the 25th, 50th, and 75th percentiles of indoor scene luminance [Zhang et al. 2023].

nothing, *global only*, or *global+local*), shown on the haploscope under one of the (background-luminance, additive-contrast) operating points in Table 2, with the corresponding background pattern (flat or $1/f$ noise; Figure 11) simultaneously displayed on the DynaScan at the studied luminance to produce the additive AR viewing condition. This task framing is analogous to recent perceptual studies of HDR rendering and tone mapping [Chen et al. 2025; Tariq et al. 2023].

Analysis. Four participants were removed from the dataset as outliers using the method proposed by Perez-Ortiz and Mantiuk [2017]. Statistical analysis using N-way ANOVA via bootstrapping [Perez-Ortiz et al. 2019] determined all three variables (additive contrast, background luminance, tone mapping type) to be significant ($p \ll 0.01$). No interactions were found to be significant ($p \gg 0.05$).

Results. Figure 12 shows the results of our experiment, averaged over base image type. The red line shows the *no mapping* condition, with perceived quality rising with increasing additive contrast. The yellow and green lines depict the *global only* and *global+local* conditions, respectively. Both methods significantly improve perceived quality compared to *no mapping*.

Tracing a horizontal line, we can see that the visual quality achieved by enabling *AROMA* at a given additive contrast is equivalent to *no mapping* at a larger additive contrast. The inset depicts this relationship, showcasing the relative gain in percentages. The *global+local* condition achieves the highest gains, with values ranging from 31% to 37% decreases in luminance to match quality, which is equivalent to a 45–59% increase from the *no mapping* condition.

10 On-device implementation

AROMA is designed to be efficient for the limited capabilities of headmounted AR displays. To test this, we ported *AROMA* from the MATLAB / Psychtoolbox prototype to a pure-Vulkan implementation and measured it on the *Meta Ray-Ban Display* — a monocular optical see-through AR headset with a Snapdragon AR1 Gen 1 SoC

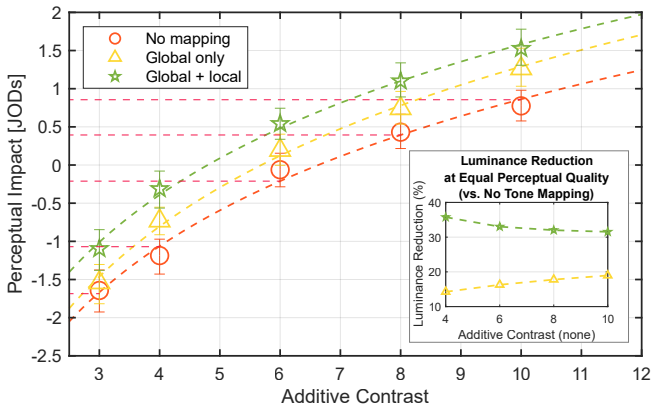


Fig. 12. On the x-axis, additive contrast (AC) is shown. On the y-axis, just-objectionable-difference (JOD) perceptual values are shown, approximately scaled so that 0 corresponds to the midpoint AC of 6:1. Dashed lines show a simple log-linear model fitted to each condition separately. Inset: percentage reduction in display luminance vs. no mapping achievable for each method while preserving the same JOD score (equal quality).

and a 640×640 panel at 30 Hz — and on the *Meta Quest 3* (Snapdragon XR2 Gen 2, Adreno 740). as a portability comparator. The per-frame *compute* timings for full *AROMA* are 3.99 ms per frame on the Ray-Ban Display and 0.72 ms on the Quest 3 (5.5× faster), well within the Ray-Ban Display's 30 Hz (33.3 ms) frame budget (see Figure 13). Quest 3 is tested as a portability reference representing the greater compute headroom expected from future AR hardware. Its result indicates that AROMA's already-small processing cost should decrease further as mobile GPU performance improves. The pipeline introduces a deliberate one-frame algorithmic latency: the histogram from frame $f-1$ is read back, fed to the CPU active-set solver, and the resulting LUT is consumed by the GPU at frame f. The temporal box filter on the tone curve (Section 5.4) acts over ≈0.5 s of frame history, which is over an order of magnitude longer than the one-frame delay, so the async overlap is invisible in the time-evolution of the rendered output.

Per frame, the GPU runs a fused tone-extraction-and-unsharp pass (with the edge-stopping bilinear upsample of Figure 6), a weighted histogram (Section 5.2), and a per-pixel apply-tone-curve pass. In parallel, the CPU runs an active-set solver (Algorithm 16.3 of [Nocedal and Wright 2006]) for the 33-node tone-curve QP of Section 5: $\min_t \frac{1}{2} t^\top Q\, t + c^\top t$ subject to $A_{\text{ineq}}\, t \le b_{\text{ineq}}$, with Q tridiagonal-plus-λI incorporating the supra-threshold weighting of Eq. (21), c the per-tile-weighted brightness target, and A_{ineq} encoding the 32 adjacent-node monotonicity constraints plus two endpoint clamps. With async overlap, the solver hides almost entirely behind the GPU work; running them sequentially would cost 7.75 ms instead (48% saved on the Ray-Ban Display, 43% on the Quest 3; Figure 13). The two-slot scaffold double-buffers the LUT and the histogram-readback target. A binary semaphore signals histogram completion (GPU→CPU); a per-frame fence gates the GPU dispatch on LUT readiness (CPU→GPU). Per-stage computational cost is dominated by the fused extract-and-unsharp pass at 2.56 ms (Table 3).

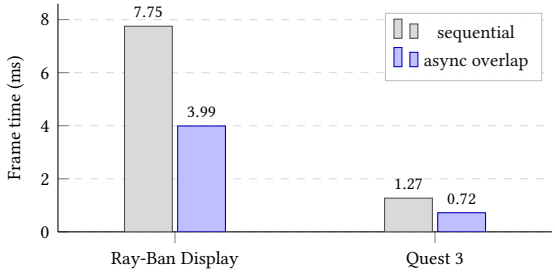


Fig. 13. Per-frame compute time of *AROMA* at *full* configuration, 640×640, on the Meta Ray-Ban Display (Snapdragon AR1 Gen 1) and the Meta Quest 3 (XR2 Gen 2, Adreno 740). Sequential pays the per-frame QP solve in series with GPU dispatch; async overlap hides it behind the GPU work. Both fit within the Ray-Ban Display's 33.3 ms 30 Hz frame budget with >29 ms left for application rendering.

Table 3. Per-stage computational cost of *AROMA* on the Meta Ray-Ban Display at 640×640, *full* configuration. The GPU stages sum to 3.34 ms; with the CPU solve, the strict-serial total is 4.81 ms. Async overlap brings end-to-end to 3.99 ms (Figure 13). The CPU QP solve uses the supra-threshold weighting of Eq. (21); the simpler flat formulation of Eq. (14) costs 1.40 ms on the same A55 cluster (−5%), the trade-off being a perceptually weaker contrast objective. Memory is the per-stage scratch buffer; input and output textures (∼4.9 MiB combined) are excluded as shared across stages.

Stage	Time (ms)	Memory
Extract + unsharp	2.56	∼3 KiB
Histogram	0.30	∼16 KiB
Apply tone curve	0.48	∼0.5 KiB
Tone-curve QP solve (CPU)	1.47	∼36 KiB
End-to-end (async)	**3.99**	—

We also implemented *AROMA* with Halide [Ragan-Kelley et al. 2013]. On the Ray-Ban Display's Adreno GPU, the Halide path runs in 14.6 ms at 640×640, dominated by the weighted histogram; per-pass Halide times are not directly comparable to Table 3's GPU-timestamp values (Halide's profiler is wallclock-based), so we report only end-to-end. Halide's mobile-GPU back-end is autoscheduler-driven and historically tuned for CPU and NVIDIA CUDA targets; reaching the Adreno's peak throughput required roughly 2× the source-line count in the hand-written Vulkan path (≈4.9 kLoC of C++ host plus GLSL compute shaders, vs. ≈2.3 kLoC of Halide generators). The gap reflects backend-specific tuning investment, not a fundamental algorithmic difference; Halide remains attractive for its portability, even if a hand-written path wins on this particular GPU family.

Implementation details. For our on-device implementation, we used N=32 tone-curve segments (33 LUT nodes), 32 histogram bins with the per-tile weighting and shaping rule of Section 5.3, λ=0.01, $g_{\max}$=1.3, 32×32-pixel zonal tiles (stride 32), a 30-tap temporal box filter (per Section 5.4, at 30 Hz, this filter spans 1 s, suppressing visible frame-to-frame fluctuations but remaining responsive to scene

changes), and the edge-stopping bilinear upsample of Figure 6 with σ=1.0 in $\log_2$ tone-value units. The active-set solver is cold-started each frame (no inter-frame warm-start), converges at $\varepsilon_{\text{rel}} = 10^{-6}$ on residual feasibility, and uses a 200-iteration safety cap that is not reached on real content. Reported numbers are medians over 100 timed frames after 30 warm-up frames at pinned clocks (*tone-curve QP* pinned to the Cortex-A55 perf cluster). The bench harness consumes a preloaded RGBA8-sRGB `VkImage`, matching the format of an Android Surface fed by UI components or a compositor-rendered swapchain — in either case GPU-resident — so these timings exclude only the system compositor and OS computational overhead, not the in-pipeline computational cost of the image-read.

11 Conclusions and Discussion

AROMA is a tone mapping framework designed for AR, where strong ambient light and limited display luminance severely reduce effective contrast and can distort intended content appearance. We combine an ambient-aware global tone curve minimizing perceived contrast distortion with an adaptive local detail enhancement and a color correction step, aiming to preserve the source's intended appearance within the constraints of the display and environment. We validated our approach in a subjective study, where *AROMA* was consistently preferred over no mapping, with improvements comparable to a 45–59% increase in display luminance, indicating that similar perceived quality may be achievable at lower display power. *AROMA* 's lightweight design is tailored for real-time display pipelines on power- and compute-constrained HMDs.

Limitations. OST-AR displays are limited in their ability to reproduce deep blacks and high contrast. *AROMA* can improve the legibility and quality of content. However, since ambient light cannot be removed from the display (without complex spatial modulators), no method can restore the original appearance of images seen with perfect black levels. Our method assumes a uniform ambient image when computing the required compensation because low-powered ambient light sensors (ALS) do not provide image data. A highly textured background could make the AR image less legible (due to contrast masking); however, this effect is partially mitigated by the defocus blur of the human eye [Kim et al. 2025]. Moreover, our experiment further validated the method's effectiveness in the presence of $1/f$ noise. Furthermore, our method does not consider the higher-level appearance phenomena, such as simultaneous contrast (caused by ambient light), or the perception of transparency and scission [Murdoch 2025].

References

Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. 1996. Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference*, Vol. 4. Society of Imaging Science and Technology, 238–245.

Maliha Ashraf, Rafal Mantiuk, and Alexandre Chapiro. 2023. Modelling contrast sensitivity of discs. *Electronic Imaging* 35, 10 (Jan. 2023), 246–1–246–8. doi:10.2352/EI.2023.35.10.HVEI-246

Maliha Ashraf, Rafał K Mantiuk, Alexandre Chapiro, and Sophie Wuerger. 2024. castleCSF — A Contrast Sensitivity Function of Color, Area, Spatio-Temporal frequency, Luminance and Eccentricity. *Journal of Vision* 24, 5 (2024). doi:10.1167/jov.24.4.5

Yancheng Cai, Ali Bozorgian, Maliha Ashraf, Robert Wanat, and Rafał K. Mantiuk. 2024. elaTCSF: A Temporal Contrast Sensitivity Function for Flicker Detection and Modeling Variable Refresh Rate Flicker. In *SIGGRAPH Asia 2024 Conference Papers*. ACM, Tokyo Japan, 1–11. doi:10.1145/3680528.3687586

Alexandre Chapiro, Dongyeon Kim, Yuta Asano, and Rafał K Mantiuk. 2024. AR-DAVID: Augmented Reality Display Artifact Video Dataset. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–11.

J. Chen, S. Paris, and F. Durand. 2007. Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 103.

Kenneth Chen, Dongyeon Kim, Yuta Asano, Alexandre Chapiro, Qi Sun, and Rafał K Mantiuk. 2026. Adapting Quality Metrics to Tone Mapping. In *SIGGRAPH 2026 Conference Papers*. 1–11.

Kenneth Chen, Nathan Matsuda, Jon McElvain, Yang Zhao, Thomas Wan, Qi Sun, and Alexandre Chapiro. 2025. What Is HDR? Perceptual Impact of Luminance and Contrast in Immersive Displays. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers*. 1–11.

Kenneth Chen, Thomas Wan, Nathan Matsuda, Ajit Ninan, Alexandre Chapiro, and Qi Sun. 2024. Pea-pods: Perceptual evaluation of algorithms for power optimization in xr displays. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–17.

Frédo Durand and Julie Dorsey. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics* 21, 3 (July 2002), 257–266. doi:10.1145/566654.566574

Gabriel Eilertsen, Rafał K. Mantiuk, and Jonas Unger. 2015. Real-time noise-aware tone mapping. *ACM Transactions on Graphics* 34, 6 (oct 2015), 1–15. doi:10.1145/2816795.2818092

Mark D Fairchild. 2013. *Color Appearance Models*. John Wiley & Sons.

Eduardo S. L. Gastal and Manuel M. Oliveira. 2011. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics* 30, 4 (2011), 1. doi:10.1145/2010324.1964964

MA Georgeson and GD Sullivan. 1975. Contrast constancy: deblurring in human vision by spatial frequency channels. *The Journal of physiology* 252, 3 (1975), 627–656.

Sofie R Herbeck, Michael J Murdoch, and Christopher A Thorstenson. 2024. Adjusting transparency toward optimizing face appearance in optical see-through augmented reality. *Journal of Perceptual Imaging* 7 (2024).

Robert W. G. Hunt. 2004. *The Reproduction of Colour* (6th ed.). John Wiley & Sons.

ITU-R. 2002. Parameter values for the HDTV standards for production and international programme exchange. *Recommendation ITU-R BT. 709-5* (2002).

ITU-R. 2011. Recommendation ITU-R BT.1886: Reference electro-optical transfer function for flat panel displays used in HDTV studio production. Recommendation. https://www.itu.int/rec/R-REC-BT.1886.

ITU-R. 2016. Recommendation ITU-R BT.2100-0: Image parameter values for high dynamic range television for use in production and international programme exchange. Recommendation. https://www.itu.int/rec/R-REC-BT.2100.

ITU-R. 2022. ITU-R BT.2408-4: Operational practices in HDR television production. https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-BT.2408-4-2021-PDF-E.pdf. Table 1: Recommended HDR Reference White for different nominal peak displays.

Dongyeon Kim, Maliha Ashraf, Alexandre Chapiro, and Rafał K Mantiuk. 2025. Supra-Threshold Contrast Perception in Augmented Reality. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

Mario Kleiner, David Brainard, and Denis Pelli. 2007. What's new in Psychtoolbox-3? (2007).

J.J. Kulikowski. 1976. Effective contrast constancy and linearity of contrast sensation. *Vision Research* 16, 12 (jan 1976), 1419–1431. doi:10.1016/0042-6989(76)90161-9

Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. 1997. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics* 3, 4 (1997), 291–306.

Wei Li, Midori Tanaka, and Takahiko Horiuchi. 2026. Mitigation of Color Appearance Shifts in Augmented Reality Under Aligned Conditions. *Journal of the Society for Information Display* (2026).

Rafał Mantiuk, Scott Daly, and Louis Kerofsky. 2008. Display adaptive tone mapping. In *ACM SIGGRAPH 2008 papers*. 1–10.

R. Mantiuk, R. Mantiuk, A. Tomaszewska, and W. Heidrich. 2009. Color correction for tone mapping. *Computer Graphics Forum* 28, 2 (April 2009), 193–202. doi:10.1111/j.1467-8659.2009.01358.x

Rafal K Mantiuk, Param Hanji, Maliha Ashraf, Yuta Asano, and Alexandre Chapiro. 2024. ColorVideoVDP: A Visual Difference Predictor for Image, Video and Display Distortions. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Rafał K. Mantiuk, Karol Myszkowski, and Hans-peter Seidel. 2015. High Dynamic Range Imaging. In *Wiley Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1–42. http://doi.wiley.com/10.1002/047134608X.W8265

Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafał K Mantiuk. 2021. Active Sampling for Pairwise Comparisons via Approximate Message Passing and Information Gain Maximization. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2559–2566.

Michael J Murdoch. 2025. Transparency and Scission in Augmented Reality. *Electronic Imaging* 37 (2025), 1–3.

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (2nd ed.). Springer, New York, NY. doi:10.1007/978-0-387-40065-5

S. Paris and F. Durand. 2009. A fast approximation of the bilateral filter using a signal processing approach. *International journal of computer vision* 81, 1 (2009), 24–52.

Eli Peli, Jian Yang, Robert Goldstein, and Adam Reeves. 1991. Effect of luminance on suprathreshold contrast perception. *Journal of the Optical Society of America A* 8, 8 (aug 1991), 1352. doi:10.1364/JOSAA.8.001352

Maria Perez-Ortiz and Rafal K Mantiuk. 2017. A Practical Guide and Software for Analysing Pairwise Comparison Experiments. *arXiv preprint arXiv:1712.03686* (2017).

Maria Perez-Ortiz, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafał K Mantiuk. 2019. From Pairwise Comparisons and Rating to A Unified Quality Scale. *IEEE Transactions on Image Processing* 29 (2019), 1139–1151.

Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. 1987. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing* 39, 3 (Sept. 1987), 355–368. doi:10.1016/S0734-189X(87)80186-X

Charles Poynton. 2012. *Digital video and HD: Algorithms and Interfaces*. Elsevier.

Yizhou Qian, Sung-Chun Chen, En-Lin Hsiang, Hajime Akimoto, Chih-Lung Lin, and Shin-Tson Wu. 2024. Enhancing a Display's Sunlight Readability with Tone Mapping. *Photonics* 11, 6 (2024), 578. doi:10.3390/photonics11060578

Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. 2013. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Seattle, Washington, USA) *(PLDI '13)*. Association for Computing Machinery, New York, NY, USA, 519–530. doi:10.1145/2491956.2462176

Christophe Schlick. 1995. Quantization Techniques for Visualization of High Dynamic Range Pictures. In *Photorealistic Rendering Techniques*. Springer, 7–20.

SMPTE. 2014. High dynamic range electro-optical transfer function of mastering reference displays. *SMPTE ST* 2084, 2014 (2014), 11.

SMPTE. 2016. SMPTE ST 2094-10:2016 - Dynamic Metadata for Color Volume Transform — Application #1. Available from https://www.smpte.org/.

Li Song, Yankai Liu, Xiaokang Yang, Guangtao Zhai, Rong Xie, and Wenjun Zhang. 2016. The SJTU HDR Video Sequence Dataset. In *Proceedings of International Conference on Quality of Multimedia Experience (QoMEX)*. 100.

Taimoor Tariq, Nathan Matsuda, Eric Penner, Jerry Jia, Douglas Lanman, Ajit Ninan, and Alexandre Chapiro. 2023. Perceptually Adaptive Real-Time Tone Mapping. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

C. Tomasi and R. Manduchi. 1998. Bilateral filtering for gray and color images. In *International Conference on Computer Vision*. Narosa Publishing House, 839–846. doi:10.1109/ICCV.1998.710815

Robert Wanat and Rafal K Mantiuk. 2014. Simulating and compensating changes in appearance between day and night vision. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* 33, 4 (2014), 147. doi:10.1145/2601097.2601150

Krzysztof Wolski, Fangcheng Zhong, Karol Myszkowski, and Rafał K. Mantiuk. 2022. Dark stereo: improving depth perception under low luminance. *ACM Transactions on Graphics* 41, 4 (2022), 1–12. doi:10.1145/3528223.3530136

Lili Zhang, Romain Bachy, and James Hillis. 2023. *Egocentric Luminance Statistics*. Poster. Reality Labs, Meta Platforms Inc. https://sites.google.com/view/lili-zhang-us/neti-poster

Lili Zhang and Michael J Murdoch. 2021. Perceived transparency in optical see-through augmented reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 115–120.