Video Flashing Reduction

Background and Algorithm

March 2023

v1.0

Important: This document includes a sample video that contains sequences of flashing effects.

Contents

Introduction

Background

Video content with bright, repetitive, high-contrast flashing may be uncomfortable for some viewers. In addition, more serious conditions, such as photosensitivity and photosensitive epilepsy, may be triggered by bright, flashing video content. In this paper, we describe an algorithm to detect and mitigate potentially uncomfortable video flashing.

Data on the precise stimulus attributes that produce discomfort in viewers are few and fragmentary. One widely cited report describes the fraction of symptomatic patients who experience photo-paroxysmal response (PPR) EEG waveforms as a function of the frequency of a flashing stimulus (G. F. Harding & Jeavons, 1994).

PPR are considered a precursor of epileptic seizures. A plot of those data are shown in Figure 1. They show that the incidence is highest between 10 and 50 Hz. Other data show that these responses increase with area, contrast, and luminance (G. Harding & Fylan, 1999; Trenité, 2006; Wilkins et al., 1980).

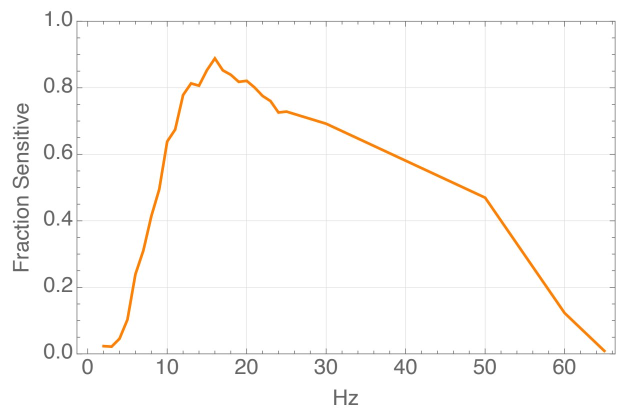

Figure 1. Frequency of PPR as a function of frequency of stimulation (Harding & Jeavons, 1994).

Not all discomfort is confined to clinical conditions. In a recent study, observers rated their discomfort with sinusoidally flashing uniform fields of various frequencies (Gentile & Aguirre, 2020). The contrast was approximately 0.9, and the luminance about 150 nits. Results are shown in Figure 2. The points show that discomfort (on a 10-point scale) was greatest for frequencies between 8 and 30 Hz. The blue curve shows the sensitivity of the Universal Flicker Metric (UFM), a metric we developed to predict sensitivity to temporal luminance fluctuations at arbitrary display sizes and luminances. This rough agreement, and the fact that the UFM can adapt to size and luminance, encouraged us to use it as the basis of our video flashing metric. The UFM is discussed further below.

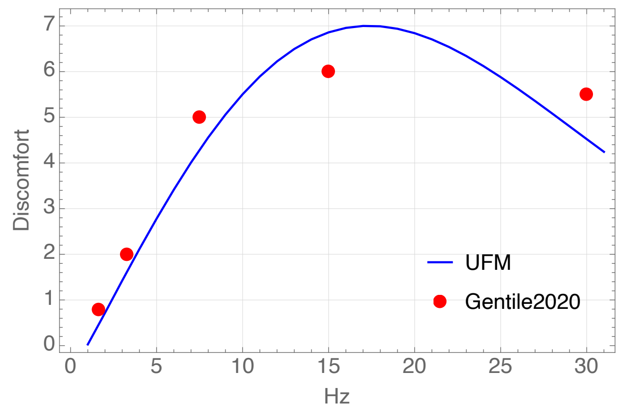

Figure 2. Data from Gentile and Aguirre (2020) (red points) and the sensitivity function of the Universal Flicker Metric (blue curve).

General Description of the Algorithm

The algorithm processes a stream of video frames, and identifies those frames or groups of frames that are likely to cause discomfort. The algorithm achieves this outcome by first computing the average luminance of each frame, and then applying filters to the sequence of frames to select energy at frequencies that are known to present a risk of discomfort. We then accumulate the energy over a brief interval of time, and convert that accumulated energy into a risk of discomfort, using a risk-mapping function.

In addition, for users that opt in to this feature, the algorithm actively reduces the risk through a mitigation process that primarily reduces the luminance and contrast of the video during the identified segments.

Universal Flicker Metric

The Universal Flicker Metric is a tool for evaluating the visibility of temporal modulations of luminance (Watson & Agaoglu, 2018). It is generally useful as a tool for evaluating flicker in displays, but also serves as a general model of human sensitivity to variations in luminance over time. An example of the predictions of the metric for sinusoidal variations is shown in Figure 2 above. In general, the UFM accepts as input a record of luminance over time, as well as an indication of the illuminated area, and outputs a measure of the visibility of fluctuations from the mean in units of Just Noticeable Difference (JND).

Technical Challenges

Because the algorithm is designed to operate in real time with minimal delay, all operations are causal and work only with the current frame and past results. As a consequence, it isn't possible to mitigate the potentially risky frames before they are detected. Thus, in a sequence of flashes, the first few may be unmitigated. Within the constraints of an effective method, we have tried to keep delays to a minimum.

Because visual sensitivity to flicker varies substantially with adapting luminance and flickering area, it is necessary to adopt filters that reflect the current size and luminance. However the adapting luminance, at least, can vary by a large factor over the course of a video. Our solution to this challenge is to select a set of five filters, suitable for a large range of luminance, and interpolate from the energy of their outputs based on the current adapting luminance.

Detailed Description of the Algorithm

Algorithm Architecture

Video Flashing Reduction (VFR) includes both a Video Flashing Metric (VFM) to quantify the risk of discomfort, and an algorithm for mitigating the risk. VFM is designed to accommodate either real-time streaming video or off-line, non-real-time video processing that can be used to annotate a video file. An overview of the VFR architecture is shown in Figure 3. The algorithm has two types of input: static, which are provided once per video or session, and dynamic, which consist of the stream of frames.

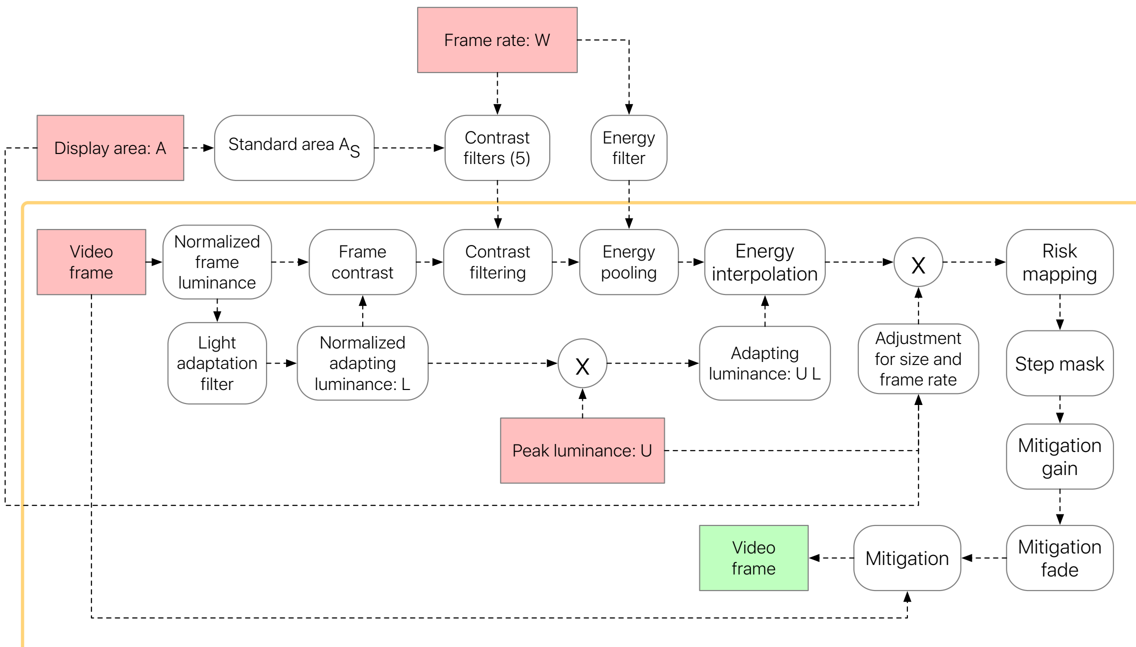

Figure 3. Architecture of the Video Flashing Reduction algorithm. The orange box encloses frame operations. Pink boxes are inputs; green is output.

Static Inputs

There are three static, one-time or per-session inputs: U, the unit luminance of the display in nits, A, the area of the video in square degrees, and W_0, the frame rate of the video in Hz.

Unit Luminance

The unit luminance U is equivalent to the peak white luminance, produced when the normalized luminance is 1. Normalized luminance is the luminance of a pixel divided by the peak white luminance.

Standard Sizes

The VFM employs a set of three standard sizes: 6, 20, and 45 degrees, corresponding roughly to the longer dimensions of Apple Watch, iPhone, and MacBook displays. These values are derived from the UFM.

Standard Luminances

Likewise VFM employs five standard luminances: 0.2, 1, 10, 150, and 500 nits (cd m^{-2}). These values are also derived from the UFM.

Standard Frame Rates

The VFM handles a set of 7 standard frame rates: 24, 25, 30, 50, 60, 90, 120 Hz. If the input frame rate W_0 differs from these rates, we select the nearest standard frame rate W. We call W the video frame rate.

Video Area and Size

The area A in square degrees depends on the area of the display in square meters A_m, and the viewing distance in meters d:

$$A = A_m \left(\frac{360 \text{ArcTan}[2d, 1]}{\pi} \right)^2 \tag{1}$$

The video area A is converted to a size S_0, based on the formula:

$$S_0 = \sqrt{1.6\, A} \tag{2}$$

We then find the nearest standard size S to S_0. We call S the video size.

Filter Selection

The VFM is provided with a set of 105 digital FIR filter kernels. These correspond to a separate kernel for each standard frame rate, standard size, and standard luminance. These kernels are based on the UFM. Consult the Appendix for more details on the construction of these filters.

At the start of a session, a set of five kernels is selected, written h_k, corresponding to the video frame rate W and size S, as defined above. The subscript k is an index to the five standard luminances. An example set of five kernels is shown in Figure 4. The kernels may have different lengths, which we write N_k.

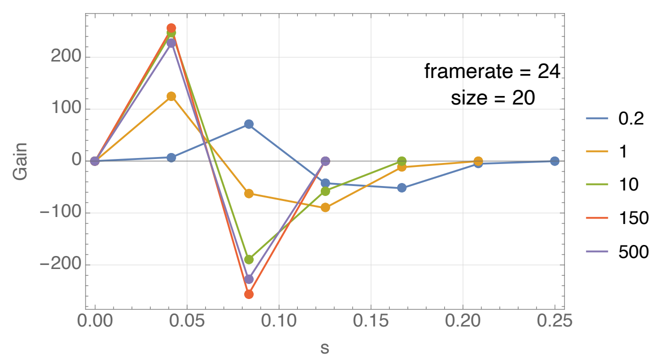

Figure 4. Five filter kernels for W= 24 Hz, S = 20 deg.

Frame Processing

The remaining inputs are the video frames. Frame processing is a continuous streaming process, and in general we refer to frame number i as the current frame. A number of process steps involve previous steps. The process of including previous steps is handled by means of buffers of appropriate length, that are generally initialized to zero.

Normalized Average Luminance

The first step in the process is to convert each video frame to a normalized average luminance $l(i)$, where i is the frame number. Normalized means that the maximum numerical luminance value in the video container is 1. Conversion to luminance depends on the specific encoding of the digital video.

Adapting Normalized Luminance

The adapting normalized luminance $a(i)$ is computed by a recursive IIR (infinite impulse response) exponential filter, based on $l(i)$ and $a(i\text{-}1)$:

$$a(i) = \mu\ l(i) + \left(1 - \mu\right)a(i-1)\,. \tag{3}$$

The rate of change of the adaptation level is determined by the parameter μ, which depends on the parameter τ (in seconds) and the frame rate W:

$$\mu = 1 - e^{-\frac{1}{\tau W}} \tag{4}$$

The adaptation level must be initialized at $i = 0$ (the frame before the first), and we set $a(0) = l(1)$.

Luminance Contrast

The contrast $c(i)$ is computed as:

$$c(i) = \frac{l(i)}{a(i)} - 1\,. \tag{5}$$

Filter Responses

The contrast waveform $c(i)$ is then convolved with the five selected kernels h_k to produce the five responses $r_k(i)$:

$$r_k(i) = \sum_{j=1}^{N_k} c\left(i - j + 1\right)h_k\left(j\right)\quad . \tag{6}$$

Energy Pooling

At the next stage, we compute the local energy of the each response. The response $r_k(i)$ is raised to the power α, convolved with an energy kernel h_e and the α root taken to compute the local energy $e_k(i)$:

$$e_k(i) = \left[\sum_{j=1}^{N_e} \mid r\left(i - j + 1\right) \mid^{\alpha} h_e\left(j\right)\right]^{\frac{1}{\alpha}}\quad . \tag{7}$$

Typically the exponent $\alpha = 2$. Note that we call this energy, but it is actually the square root of energy. N_e is the length of the energy kernel.

The energy pooling kernel h_e is defined as the PDF of a Gamma probability density function with shape parameter κ and scale parameter λ (seconds):

$$h_e(t) = \frac{e^{-\frac{t}{\lambda}}\lambda^{-\kappa}t^{-1+\kappa}}{\Gamma(\kappa)} \tag{8}$$

The sampled discrete version of the kernel must be constructed based on the frame rate W. An example is pictured in Figure 5.

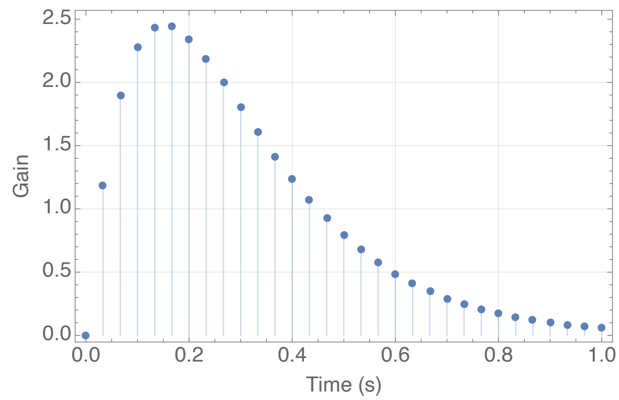

Figure 5. Energy pooling kernel for W = 24 Hz, ϰ = 2, λ = 0.15.

Energy Interpolation

We determine the final energy $e(i)$ by interpolating among the five energies $e_k(i)$, based on the value of the adapting normalized luminance $a(i)$ times the unit luminance U, relative to the standard luminances of the kernels. The interpolation is linear in the space of energy versus log adaptation level, as shown in Figure 6. This interpolation solves the problem of how to adapt the filter kernel to the current adaption luminance.

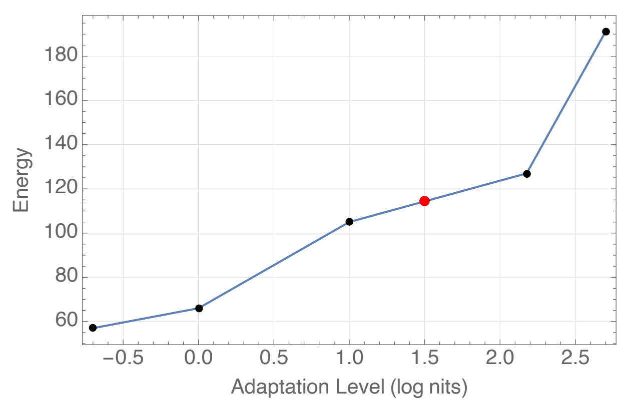

Figure 6. Energy interpolation. The dark points show energies on frame i for each of the five kernels h_k, designed for each of the five standard luminances. The dark line shows the linear interpolation, and the red point is the interpolated response at the current adaptation level.

Energy Adjustment

Following interpolation, we multiply by a term g, an adjustment factor determined by the input size S_0 and a global gain G:

$$g = \left(\frac{S_0}{S}\right)^{2c_A} G W^{-1/\alpha} \tag{9}$$

This term adjusts for the difference between the actual size and the standard size, and adjusts for different frame rates in subsequent energy calculations. The parameter G is a global gain parameter, and α is the energy pooling exponent described in the previous section.

Risk Mapping

We can map the energy to a bounded quantity that reflects the risk of discomfort. The energy computed above is bounded, in practical terms and using default parameters, to about 4,000. But most video produces values of less than 500. We have adopted a particular mapping that suppresses small values, and compresses large values. The mapping is based on a Weibull probability density, with location parameter μ, shape parameter ν, and scale parameter ξ. The mapping to risk q can be written:

$$q(e) \; = \; 100 \begin{pmatrix} 1 - e^{-\left(\frac{e-\mu}{\xi}\right)^{\nu}} & e > \mu \\ 0 & \text{otherwise} \end{pmatrix} \tag{10}$$

This mapping is illustrated in Figure 7 for the default parameter values.

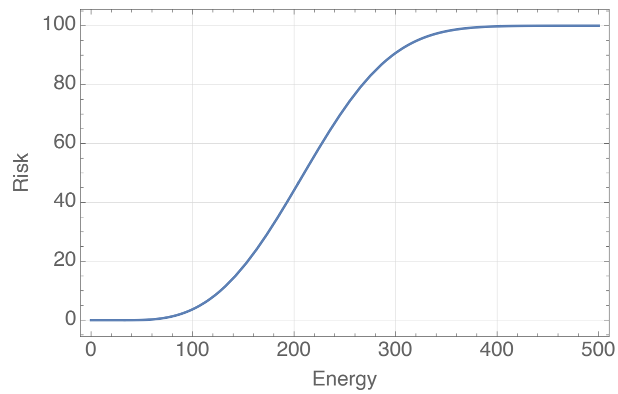

Figure 7. Risk Mapping Function with location parameter μ = 33, shape parameter ν = 3, and scale parameter ξ = 200.

Step Masking

The algorithm developed above correctly estimates risk as the supra-threshold magnitude of the contrast energy. But it may respond to transients that, while highly visible, are less likely to lead to discomfort. In particular, scene cuts may involve large, abrupt step changes in mean luminance. To ameliorate this effect, we include an additional process that we call *step masking*.

Step masking involves a parallel process, similar to the energy calculations above, but differing in that normalized cross-correlation is used in place of convolution.

Outside of the frame processing loop, after kernels are selected, we compute the norm of each kernel:

$$\overline{h}_k = \sqrt{\sum_{j=1}^{N_k} h_k^2(j)} \; . \tag{11}$$

Within the frame processing loop, we compute the norm of the luminance contrast over the preceding N_k frames:

$$\overline{c}_k(i) = \sqrt{\sum_{j=0}^{N_k-1} c^2\left(i-j\right)} \; . \tag{12}$$

We then cross-correlate the contrast waveform $c(i)$ with the five selected kernels h_k to produce the five normalized cross-correlations $n_k(i)$:

$$n_k(i) = \frac{\sum_{j=1}^{N_k} c(i-j+1) h_k(N_k - j + 1)}{\overline{h}_k \; \overline{c}_k(i)} . \tag{13}$$

Note that this expression is similar to the convolutions described by Equation 6, but the kernel isn't reversed in time before multiplication, and the result is normalized by the kernel and contrast norms.

The energies of these correlations are then pooled, as with the energies above, through convolution with the energy kernel:

$$f_k(i) = \left[\sum_{j=1}^{N_e} \mid n_k(i-j+1) \mid^{\alpha} h_e(j) \right]^{\frac{1}{\alpha}} . \tag{14}$$

We again interpolate among the five correlation energies, to yield a final correlation energy $f(i)$. Finally, if this quantity for a frame is below a threshold δ, then the risk is masked to zero.

Mitigation

Mitigation Transfer Function

To reduce the risk of discomfort posed by flashing video, we use a simple reduction of contrast. Specifically, we use a transfer function described by:

$$l_{\text{out}} = a(1 - f_C) f_L + l_{\text{in}} f_C f_L \tag{15}$$

where l_{in} and l_{out} are the input and output luminances, and f_C and f_L are the contrast and luminance mitigation factors, and a is the adapting luminance for that frame. Examples of this transfer function are shown in Figure 8.

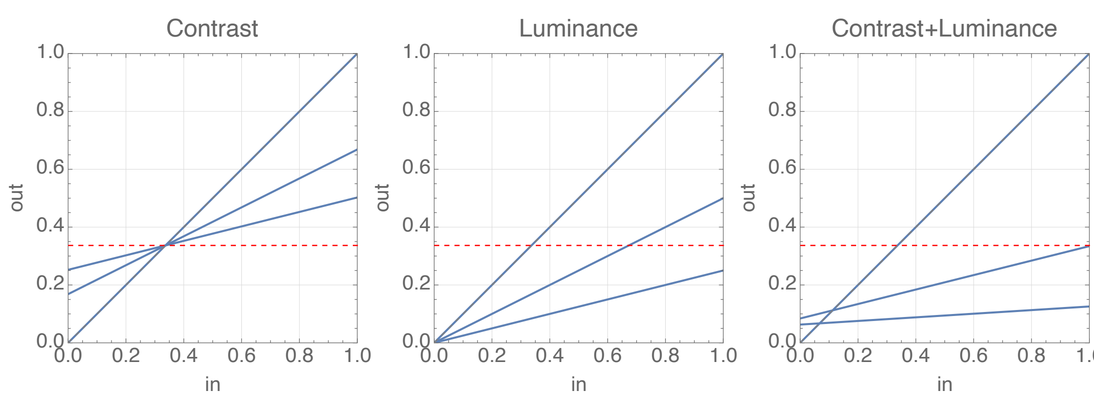

Figure 8. Illustration of mitigation of contrast, luminance, and both. In each case we show values of 1, 0.5, and 0.25.

Currently the mitigation is applied on a frame-by-frame basis. Note that the transfer operates on linear normalized luminance values, and for an RGB color image acts equally and independently on the three color channels. An illustration of the effects of f_C and f_L on a single frame is shown in Figure 9.

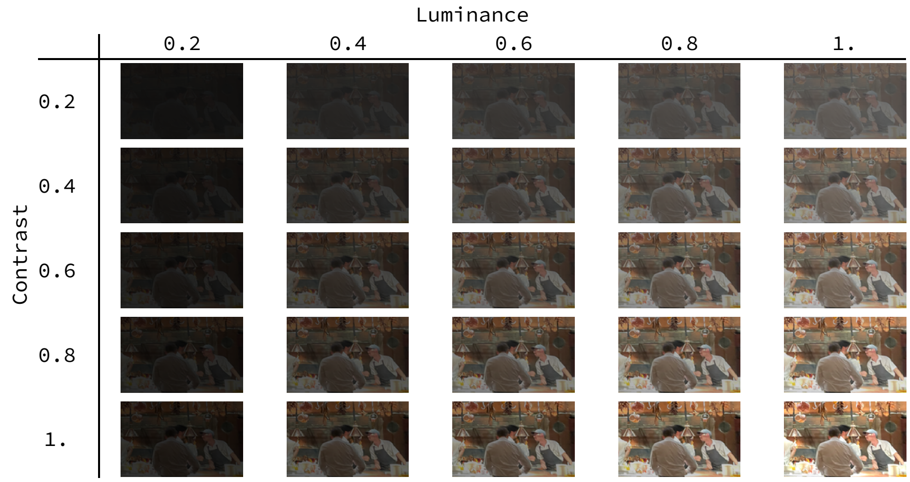

Figure 9. Effect of contrast and luminance mitigation factors on a single video frame. The adapting luminance was assumed to be 0.33.

In Figure 10 we show an animation in which the contrast and luminance mitigation factors are separately varied between 0 and 1.

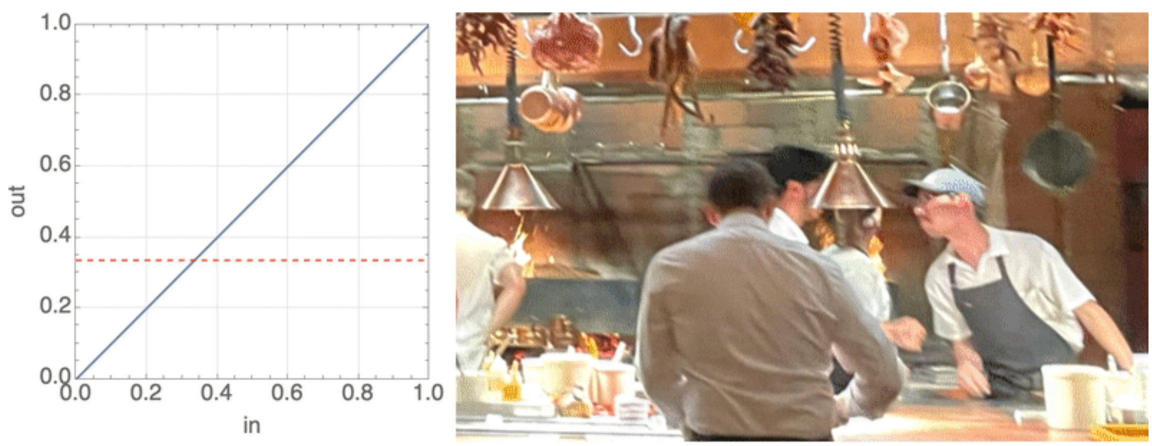

Figure 10. Animation showing the effect of contrast and luminance mitigation.

Mitigation Strength

The mitigation strength is given by:

$$m = \frac{M}{2} \log_{10}\left[\mathrm{Max}\left[1, q(i)\right]\right] \tag{16}$$

where M is the mitigation gain in the range [0,1], and $q(i)$ is the frame risk. Because risk is in the range [0,100], the log is in the range [0,2], and mitigation strength is in the range [0,1].

Mitigation Fading

Additionally, to provide a gentle recovery from mitigation, when mitigation declines we adjust m with an exponential decay over time, using a timescale (in units of frame) of μ_m:

$$m(i) = \mu_m\, m + \left(1 - \mu_m\right) m(i-1)\,. \tag{17}$$

The rate of change of the mitigation strength is determined by the parameter μ_m, which depends on the parameter τ_m (in seconds) and the frame rate W:

$$\mu_m = 1 - e^{-\frac{1}{\tau_m W}} \tag{18}$$

The mitigation strength must be initialized at $i = 0$ (the frame before the first), and we set $m(0) = 0$.

Mitigation Factors

We have set the mitigation factors for contrast and luminance in each frame according to the formula:

$$\{f_C\,, f_L\,\} = 1 - m\ \{w_C\,,\ w_L\,\} \tag{19}$$

where m is the mitigation strength, f_C and f_L are the contrast and luminance mitigation factors, and w_C , w_L are the contrast and luminance mitigation weights. The weights are in the range [0,1], and so the factors will lie in the range [0,1].

Example: High Contrast Pulse Train

In this section we illustrate the steps in the algorithm with a synthetic sequence. While it is unlikely to occur in typical video, a frame sequence that is particularly risky is an an extended sequence of alternating bright and dark frames. The risk increases when the alternation rate is near to the peak of visual sensitivity (see Figure 2). In this example, we use a frame rate of 24 Hz and alternating frames of bright (normalized luminance = 1) and dim (0.2), yielding a signal at 12 Hz. The minimum luminance is 100 nits, and the maximum 500 nits. There are a total of 36 pulses, over an interval of 3 seconds.

The processing steps are illustrated in Figure 11. The adaptation level (panel 2) begins at the lower luminance but rises as the flashes begin, and subsides once they cease. The contrast (panel 3) is initially very high, but subsides somewhat as adaptation level rises. The responses (panel 4) are also very large for the first pulses, but decline somewhat. The energy (panel 5) rises to a level of around 500, and remains high until the pulses cease. The risk (panel 6) rises quickly to the maximum of 100 and remains there until after the pulses cease. In panel 7, we show the mitigated luminance after the first few pulses the contrast is greatly attenuated. In panels 8 and 9, we show the mitigated energy and risk.

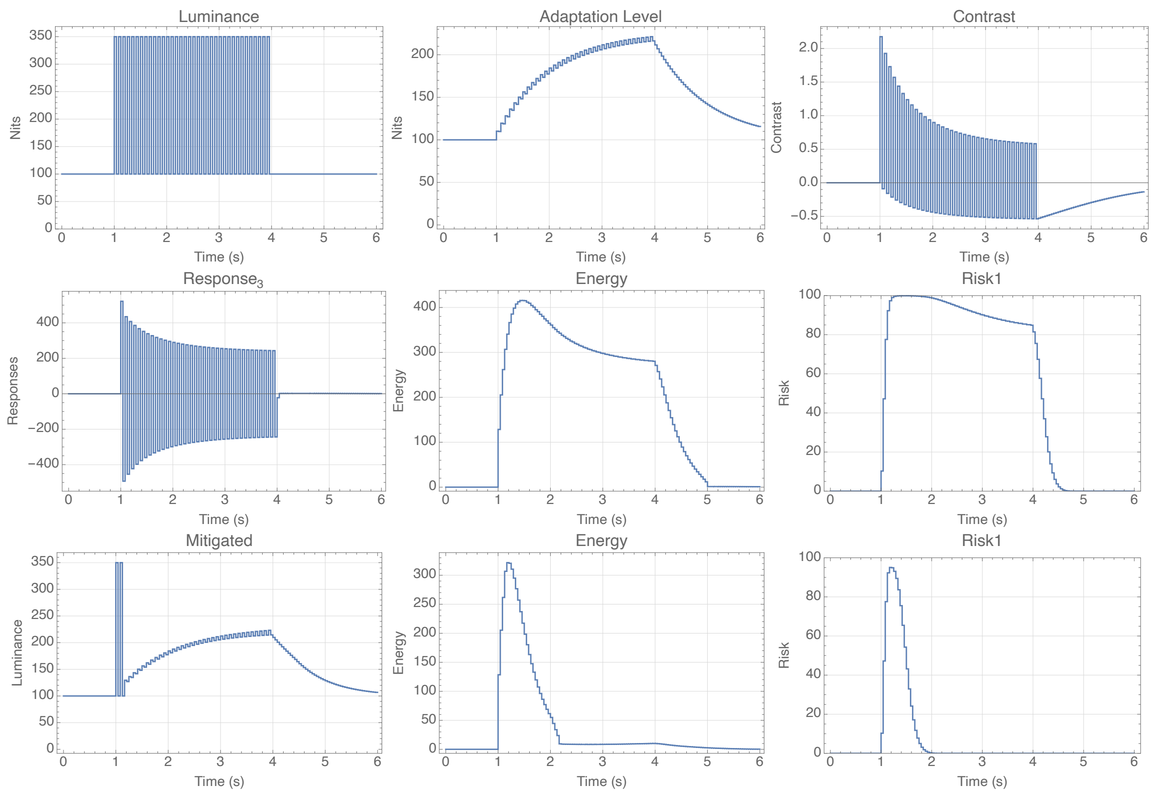

Figure 11. Example of processing stages for input of high contrast pulse train.

Example: Detection and Mitigation of Video Flashing

Here we show an example of a video clip with significant video flashing. We show both original and mitigated video, along with a plot of the energy and mitigation strength over the time course of the video.

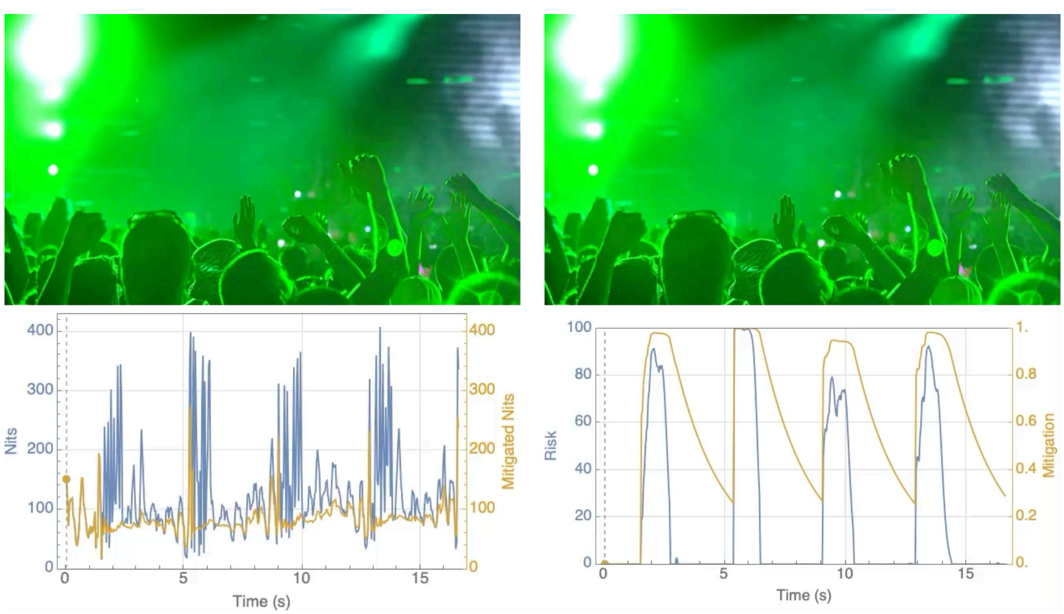

Figure 12. Illustration of the Video Flashing Reduction algorithm. The original and mitigated video are shown, along with a plots of the original and mitigated luminance, and the resulting risk and mitigation strength.

Future Work

As we develop a deeper understanding of the specific video features that induce discomfort, and how they can be mitigated, and as we gain more experience with our current algorithm, we anticipate future enhancements that will be made available as part of the usual software upgrade process. We expect those enhancements will also be documented.

Appendices

Appendix: Contrast Filters

The metric relies on contrast filters that process the stream of frame luminance values. Each filter is an finite impulse response (FIR) defined by a kernel (a list of numbers). There is a distinct kernel for each combination of average luminance, display size, and frame rate. In this section, we describe the creation of these filters.

The filters are based on the filters defined by the Universal Flicker Metric. These filters were created by fitting a particular function to the contrast sensitivity data collected in a user study (Watson & Agaoglu, 2018). That function represents a particular filter design, and is specified by five parameters: {g, α, β, a, b, c}. The filter can be expressed as either an impulse response h, or a transfer function H. Both are continuous functions.

Note that video at a frame rate W will have temporal frequencies up to $W/2$. If the kernel has an odd number N of samples, then its total duration will be $D = N/W$. The MTF=|DFT| of that kernel will specify magnitudes at a discrete set of frequencies with spacing $d_W = 1/D = W/N$, from 0 to $d_W (N\text{-}1)/2 = (W/2)(N\text{-}1)/2$. We call that *$mtf_0$*.

One approach to creating an FIR kernel would be to sample the impulse response at discrete points in time that are multiples of the frame rate. We call that *ker_1*. However, the *mtf_1* of *ker_1* may not resemble *mtf_0*. This typically happens when the frame rate is low, because the samples are too far apart in time.

However, we can perturb the filter parameters to produce a *ker_2* whose *mtf_2* resembles *mtf_0*. We have found that this generally produces a good match between *mtf_2* and *mtf_0*, so long as certain conditions are met. Figure 13 shows this process for one example at 24 Hz frame rate.

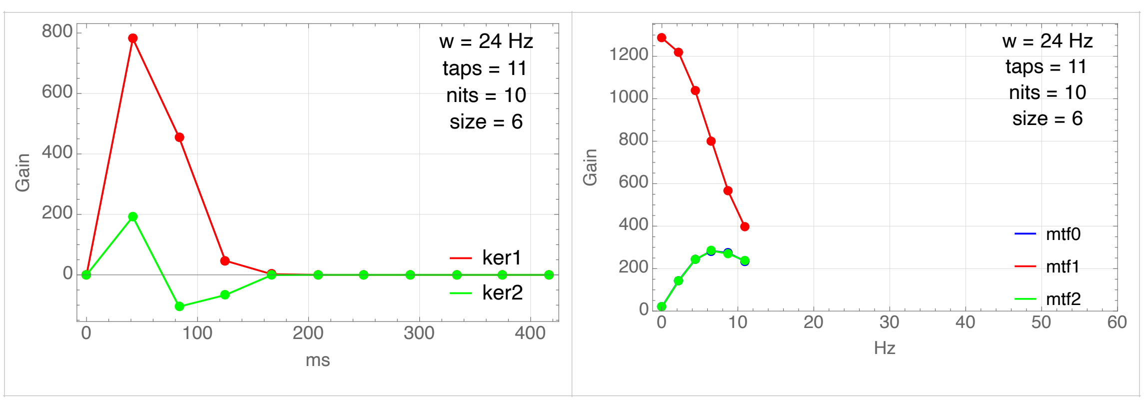

Figure 13. Construction of UFM filter kernels. The individual plots are, on the left, ker1 (red): the initial estimate of the kernel; ker2 (green): final estimate of the kernel; on the right, mtf1 (red) the initial MTF; mtf1: the final MTF; mtf0 (blue): the MTF of the UFM.

In the UFM, we defined filter parameters for three sizes: 6, 20, and 45 degrees, and for 5 mean luminances: 0.2, 1, 10, 150, and 500 nits. (The sizes refer to the longer dimension of a rectangle whose other dimensions was 1.6 times shorter.) Here we create filters for that same array of nits and size, and also for a set of seven possible frame rates: 24, 25, 30, 50, 60, 90, and 120 Hz. This results in a total number of kernels of 3 x 5 x 7 = 105 kernels.

The required kernel length is determined by the frame rate, and by the duration of the impulse response, which increases as luminance is reduced and as size is reduced. The longest kernel is required for the frame rate of luminance of 0.2 nits, and size of 6 degrees, of about 250 ms. At 120 Hz, that results in a kernel of length 31. Table 1 below shows the required kernel length when terminal values less than 10^{-4} are dropped.

size = 6

nits \ frame rate	24	25	30	50	60	90	120
0.2	7	9	10	15	18	26	31
1	7	8	9	14	16	24	31
10	6	5	7	12	13	20	27
150	4	5	5	7	9	11	15
500	4	4	5	6	9	10	13

size = 20

nits \ frame rate	24	25	30	50	60	90	120
0.2	7	7	8	14	17	24	31
1	6	7	8	14	15	22	28
10	5	5	5	9	11	16	22
150	4	4	5	6	7	11	13
500	4	4	5	5	6	9	12

size = 45

nits \ frame rate	24	25	30	50	60	90	120
0.2	6	7	8	13	15	23	29
1	6	6	7	12	14	20	25
10	5	5	5	7	9	13	17
150	4	4	4	6	6	9	12
500	3	4	4	5	6	10	11

Table 1. Kernel lengths.

The complete set of kernels is shown in Figure 14. The MTFs of these kernels are shown in Figure 15, along with the target MTF derived from the transfer function of the UFM.

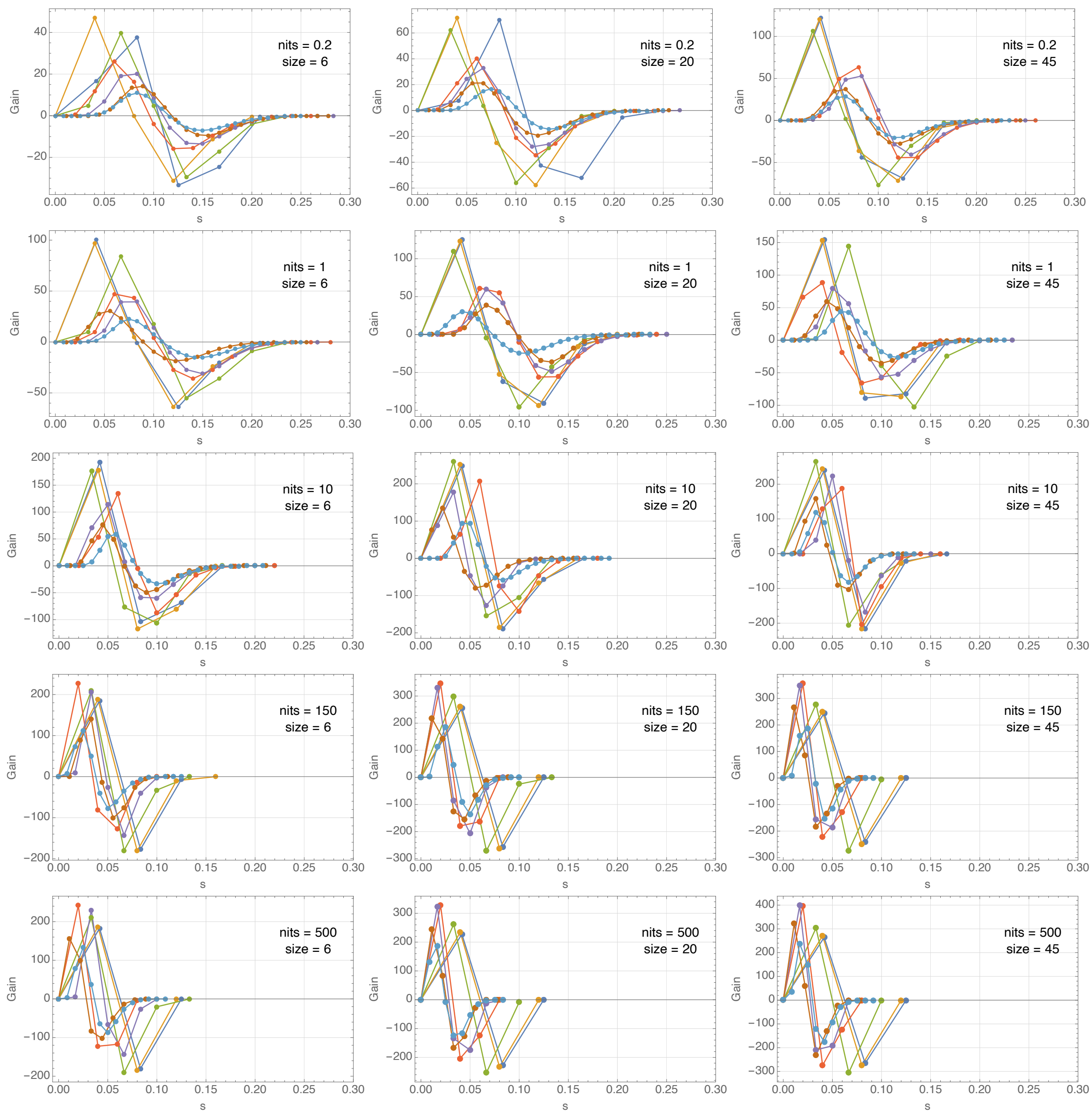

Figure 14. Complete set of 105 kernels based on UFM. Within each panel, the different kernels correspond to different frame rates.

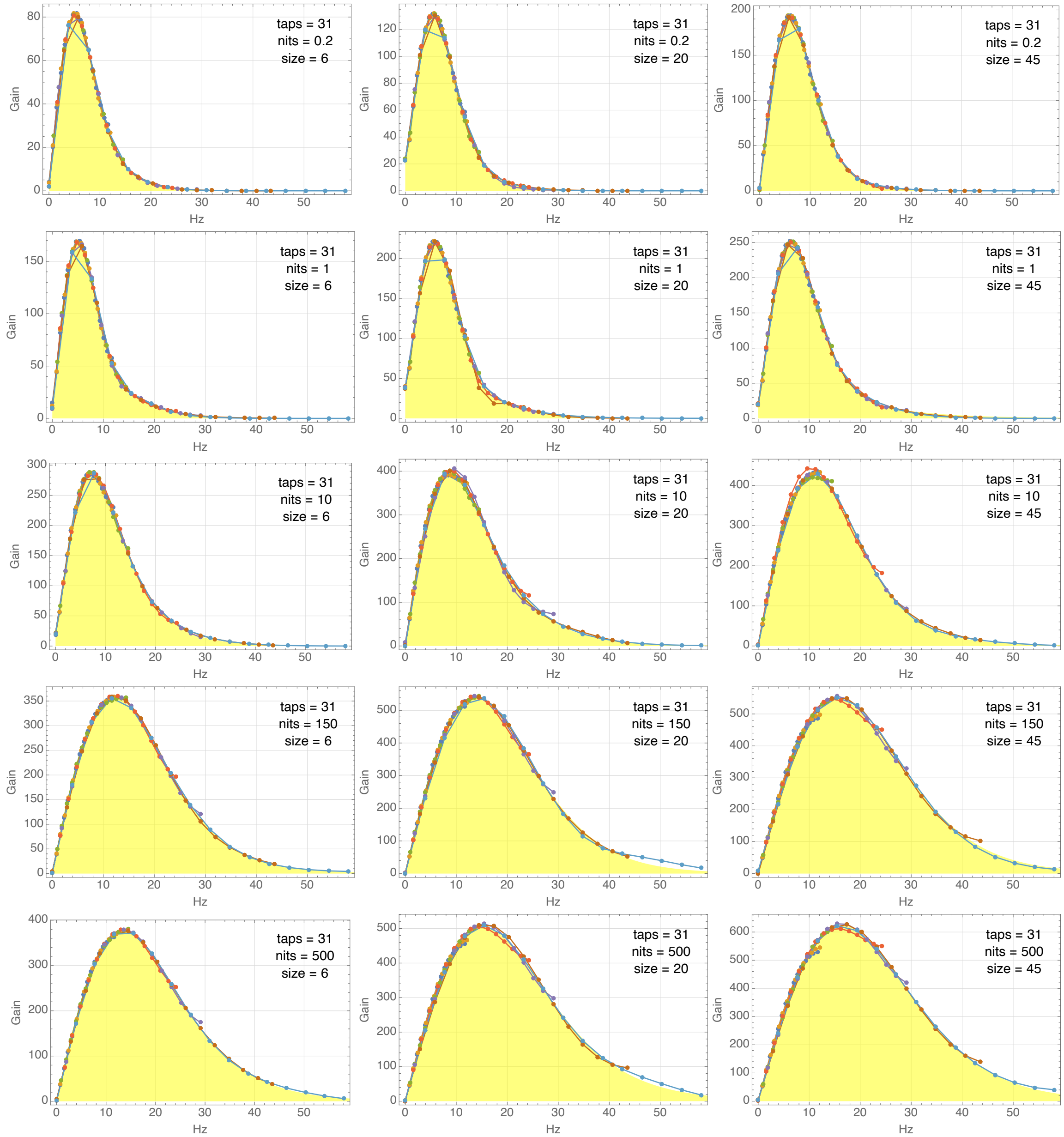

Figure 15. MTFs of all UFM kernels (colored points), compared with the MTF of the UFM itself (yellow-filled region).

Appendix: Notation

The following is a summary of notation used in this document, in order of appearance. Where relevant default values and units are provided.

Symbol	Default	Definition	Unit
U	500	Peak display luminance	nits
A	1265.63	Display area	degree^2
W		Frame rate of selected kernel	Hz
W_0		Input frame rate	Hz
A_m		Display area	meters^2
d		Display viewing distance	meter
S_0		Input display size	degree
S		Kernel display size	degree
h_k		Kernel for *k*th standard luminance	
N_k		Length of kth kernel	
i		Frame number	
$l(i)$		Frame normalized average luminance	
$a(i)$		Frame adapting normalized luminance	
μ_L		Adapting luminance filter coefficient	
τ_L		Adapting luminance time constant	seconds
$c(i)$		Frame contrast	
$r_k(i)$		Frame response of kth filter	
h_e		Energy kernel	
α	2	Energy exponent	
λ	0.15	Scale of energy kernel	seconds
κ	2	Shape of energy kernel	
$e_k(i)$		Frame energy of kth kernel	
$e(i)$		Frame energy	
g		Adjustment factor	
G	1	Global gain parameter	
c_A	0.263	Area effect on sensitivity	

Symbol	Default	Definition	Unit
$q(\mathrm{i})$		Frame risk	
ν	3	Risk map shape parameter	
ξ	200	Risk map scale parameter	
μ	33	Risk map threshold parameter	
$n_k(i)$		Normalized cross-correlation for kernel k	
$f_k(i)$		Cross-correlation energy for kernel k	
$f(i)$		Cross-correlation energy	
δ	1.8	Cross-correlation energy threshold	
f_C		Contrast mitigation factor	
f_L		Luminance mitigation factor	
m		Mitigation strength	
M	1	Mitigation gain	
μ_m		Mitigation filter coefficient	
τ_m	2	Mitigation time constant	seconds
w_C	1	Contrast mitigation weight	
w_L	0.5	Luminance mitigation weight	

Bibliography

Gentile, C. P., & Aguirre, G. K. (2020). A neural correlate of visual discomfort from flicker. Journal of Vision, 20(7), 11-11.

Harding, G., & Fylan, F. (1999). Two visual mechanisms of photosensitivity. Epilepsia, 40(10), 1446-1451.

Harding, G. F., & Jeavons, P. M. (1994). Photosensitive epilepsy: Cambridge University Press.

Trenité, D. G. K.-N. (2006). Photosensitivity, visually sensitive seizures and epilepsies. Epilepsy research, 70, 269-279.

Watson, A. B., & Agaoglu, M. (2018). Universal Flicker Metric: Apple.

Wilkins, A., Binnie, C., & Darby, C. (1980). Visually-induced seizures. Progress in neurobiology, 15(2), 85-117.