A Novel Flashing Video Content Detector

Nezih Topaloglu

10 September 2022

Abstract

Photosensitive epilepsy (PSE) is a condition where seizures are triggered with fast luminance variations and pulsating lights, known as epileptogenic visual content. Television broadcasts, internet video streams and digital games are common sources of epileptogenic visual content. In this study, an automatic flashing video detection tool is developed. The tool is compliant with the recent WCAG 2.3.1 guidelines. The algorithm is verified on a set of synthetically generated benchmarking videos. It is shown that the frequency, area and intensity criteria of the WCAG 2.3.1 guidelines are correctly checked.

Keywords: flashing content detection, photosensitive epilepsy, digital video processing

Introduction

Epilepsy is a serious disease affecting 3.4 million people in the United States of America [1]. When the epileptic seizures are triggered by contrasting or flashing lights, this type of epilepsy is called photosensitive epilepsy (PSE). Around 3% of people with epilepsy have PSE [2]. Among natural and artificial sources of flashing or contrasting lights, television, internet videos and video games are considered as the most common seizure triggers. PSE usually develops between the ages of 9 and 15 years old [3]. Videos that have intense flashing content can be harmful to people with PSE. A famous example is the 38th episode of the 1st season of Pokemon [4], which was broadcasted in 1997 in Japan, and caused 685 seizures [3]. In addition to people with PSE, it is a well-known fact that fast changing images and flashes cause discomfort to people not having PSE. Therefore it is beneficial to implement automatic flashing detection to broadcasted or streamed videos, as well as offline movies and video games.

Based on the clinical studies on PSE [5-8], several guidelines on broadcasted content were announced [9-11]. The first guideline was released by the Independent Television Commision (ITC) in 2001 and updated in 2012 [9]. Later, the guideline was adopted by ITU-R and released as a recommendation document in 2005 [10]. These guidelines were prepared specifically for TV broadcasting. In today's world, high-contract and widescreen devices are very common, which increases the risk of PSE triggering due to broadcasted videos, compared to early 2000s. In addition, although having small screen sizes, mobile devices also pose a serious risk as their nit levels are high and the viewing distances are very short.

As a result of these developments, more strict guidelines were proposed by the Web Accessibility Initiative of the World Wide Web consortium. The latest of these guidelines, known as Web Content Accessibility Guidelines (WCAG) 2.3.1, was released in 2018 [11]. These guidelines provide the most up-to-date criteria for detecting flashing content in videos.

Marking a video as safe or harmful based on guidelines requires an automatic video processing software. The research on this topic is very limited and very few scientific papers are available in the literature [12-14]. The first study in the field was published by Clippingdale et al. in 1999 [12]. This study included a short description of the automatic video analyzer tool they developed. In 2015, Carreira et al. developed an algorithm that detects flashing video content, using ITU-R and ITC guidelines [13]. Although their algorithm is simple and efficient, it is based on the simple flashing area criterion of ITU-R and OfCom, which states that the combined area of flashes occurring concurrently should not occupy more than 25% of the total screen area. Therefore their method is not applicable to check WCAG 2.3.1 area criterion, which is more strict and challenging. It also lacks detecting alternating checkerboard type flashes, which will be explained later in this document. In 2021, Kothari et al presented a neural-network based flashing detector [14]. This method is not considered as a real alternative, as using a learning-based method to a fully-deterministic problem is not a feasible choice. Commercial [15-16] and free [17] softwares are available, but their underlying algorithms are unknown and no benchmarking results have been published.

In this study, we develop a flashing detection and quantification algorithm based on WCAG 2.3.1 guidelines. The algorithm is based on application of the intensity criteria in a pixel-by-pixel basis, and then applying the flashing area criterion and flashing frequency criterion on in order to detect flashing content. The algorithm is validated using benchmarking videos.

The paper starts with a technical description of the WCAG 2.3.1 guidelines. The proposed detection algorithm is explained in detail in Section 3. The validation of the algorithm, using custom-made benchmarking videos is presented in the results and discussion section (Section 5). Finally, conclusions are drawn in Section 5.

Features of Flashing Content

The WCAG 2.3.1 guidelines [11] are implemented in this study. The guidelines provide criteria in characterizing two different type of flashes:

- General Flash: A flash that is characterized by a pair of opposing changes in relative luminance of 0.1 or more, where the relative luminance of the darker image is below 0.8. A pair of opposing changes is defined as an increase followed by a decrease, or vice versa. Relative luminance is defined as

  $$L = 0.2126R + 0.7152G + 0.0722B \quad (1)$$

  where R, G and B are the red, green and blue components of the CIE 1931 color space. Conversion from the default RGB color space to CIE 1931 color space can be found in [17]

  The relative luminance change requirement can be expressed as

  $$\left|L_2 - L_1\right| > 0.1 \quad (2)$$

  where L_1 and L_2 are the relative luminances of two temporally consecutive pixels. The requirement related to the intensity of the darker frame is given as

  $$min\left(L_1, L_2\right) < 0.8 \quad (3)$$

- Red Flash: A flash that is characterized by a pair of opposing transitions involving a saturated red. Either the beginning or the end of the transition should satisfy the red saturation (RS) requirement, given as

  $$max\left(RS_1, RS_2\right) \geq 0.8 \quad (4)$$

where the subscripts denote the states of transition. The red saturation is given by

$$RS = R/(R + G + B) \quad (5)$$

In addition to the above requirement, the absolute value of the change in red luminance (RL) should be greater than 20. This condition can be expressed mathematically as

$$|RL_2 - RL_1| > 20 \quad (6)$$

where the subscript denotes one state of the transition. The red luminance is defined as

$$RL = 320\, ReLU(R - G - B) \quad (7)$$

where ReLU is the rectified linear unit function, defined as $ReLU(x) = max(0, x)$.

In addition to the luminance and red saturation based criteria described above, the additional criteria based on flashing area and flashing frequency are defined:

- Area criterion: The flashing content is considered as safe if the combined area of flashes occurring concurrently occupies no more than 25% of any 10 degree visual field [18]. The 10 degree visual field is approximately a rectangle of height and width of *H/3* and *W/3*, where *H* and *W* are the height and width of the video frame.
- Frequency criterion: The flashing content is considered to be safe if there are no more than three general flashes and/or red flashes within any one-second period [19]. In terms of transitions, the number of successive (consecutively opposite) transitions should be at most six, in order for the flashing content to be safe.

Detection Algorithm

General Approach

The basic flowchart of the flashing detection algorithm is given in Fig. 1. In this algorithm, the transitions are determined first in pixel level. Afterwards, it is determined whether a transition is a successive transition or not. In order for a transition to be successive, it has to be in the opposite direction of the previous transition. After determining the opposing transitions in pixel level, the flashing area criterion is checked, and after that check, the frequency check is made at the frame level. The details of the algorithm are explained next.

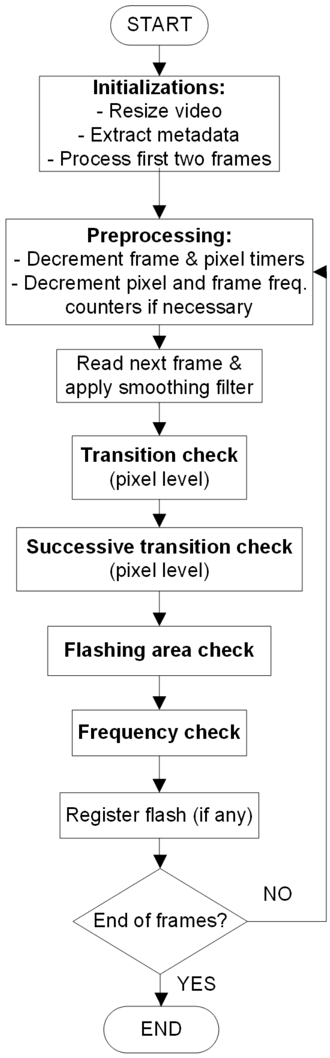

Figure 1: The detection algorithm flowchart

Initialization

Prior to frame processing, the video file is resized to a standard size to avoid performance problems. The video metadata (height, width and frame-per-second (fps)) are then obtained. In order to detect an extremum in luminance, at least three consecutive frames should be read. Therefore, the first two frames are processed, before entering the frame processing loop.

Preprocessing and Reading Next Frame

After the initialization step, a preprocessing step, which is applied before a new frame is read, is performed. The aim of this step is to update the timers and counters, which store the number of successive transitions and the number of frames until each successive transition expires, both in pixel and frame level.

After preprocessing, the next frame is read, and a uniform filter is applied to smooth the image. The objective of smoothing is to prevent any false alarms due to fine and balanced transitions, such as white noise or alternating checkerboard type pattern with "squares" smaller than 0.1 degree of visual field. This exception is indicated in WCAG 2.3.1 guidelines [11]. Applying a smoothing step is an effective way to suppress these fine transitions. It is a recommended preprocessing step in ITU-R guidelines [10].

Transition Check

For every pixel location, the transition check subroutine uses the luminance value of three temporally consecutive pixels (denoted as $L_{i,j}^{k-1}$, $L_{i,j}^{k}$ and $L_{i,j}^{k+1}$), where *i* and *j* denote the pixel row and column index. The first step is to determine whether an extremum (a peak or valley) in frame number *k* is present. The relative luminance values are calculated using Eq. (1). In order to detect an extremum, the sign of relative luminance change from *k-1* to *k* and the sign of relative luminance change from *k* to *k+1* are compared. If $sgn\left(L_{i,j}^{k+1} - L_{i,j}^{k}\right)sgn\left(L_{i,j}^{k} - L_{i,j}^{k-1}\right) = -1$, then it is concluded that there is an extremum. In Fig. 2 an example of three consecutive luminance values of a pixel, forming a peak is shown.

The next step is to determine whether the transition that ends at the extremum exceeds the WCAG 2.3.1 intensity thresholds given in the previous section. For this purpose, the two states of transition should be used. The final state of transition is $L_{i,j}^{k}$, but the initial state does not have to be $L_{i,j}^{k-1}$, as the transition can start earlier then frame *k-1*. Therefore, a separate variable, $L1st_{i,j}$ stores the luminance at the beginning state of the transition. Using $L1st_{i,j}$ and $L_{i,j}^{k}$, the intensity criteria of flash are checked, and if the pixel satisfies intensity criteria, the corresponding transition ($Tr_{i,j}$) is set as 1 or -1, depending on the transition direction. For instance, if the transition shown in Fig. 2 satisfies the flash intensity criteria, then $Tr_{i,j}$ is set as 1, as the transition from $L_{i,j}^{k-1}$ to $L_{i,j}^{k}$ is positive.

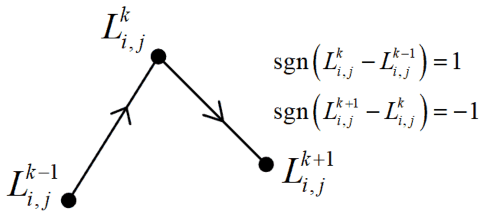

Figure 2: An extremum example in pixel i,j

A similar procedure is applied for determining transitions satisfying red flash criteria. This time, red luminance (RL) and red saturation (RS) values are required for three frames, as well as their values at the first state. The RL and RS values are calculated using Equations (7) and (5), accordingly. Therefore the following values are used: $RL_{i,j}^{k-1}$, $RL_{i,j}^{k}$, $RL_{i,j}^{k+1}$, $RS_{i,j}^{k-1}$, $RS_{i,j}^{k}$, $RS_{i,j}^{k+1}$, $RL1st_{i,j}$ and $RS1st_{i,j}$. First, it is checked whether an extremum is obtained via $sgn\left(RL_{i,j}^{k+1} - RL_{i,j}^{k}\right)sgn\left(RL_{i,j}^{k} - RL_{i,j}^{k-1}\right) = -1$. Once an extremum is detected, the intensity criteria, given by Equations (4) and (6) are checked. If those criteria are satisfied, the transition of the corresponding pixel (1 if $sgn\left(RL^{k}_{i,j} - RL^{k-1}_{i,j}\right) = 1$, -1 if $sgn\left(RL^{k}_{i,j} - RL^{k-1}_{i,j}\right) =- 1$) is set.

Successive Transition Check

The pixels that exhibit up or down transition were determined in the previous section. However, in order to determine whether the transition of a pixel is a successive transition, it has to be compared with the latest transition status, denoted as $Latest_{i,j}$. If $Latest_{i,j} = 1$, then a down transition is a successive transition and if $Latest_{i,j} =- 1$, then an up transition is a successive transition. This comparison is made by multiplying $Latest_{i,j}$ and $Tr_{i,j}$. When a successive transition is detected, a new timer is set for that pixel. The set value of a timer is the fps and the timer value decrements as a new frame is processed. If the timer value reaches zero, it means one second has passed after the transition corresponding to this frame. In this case, the transition does not contribute to a flash anymore, so the timer is removed. The number of timers of a pixel (*PixelFCount*) shows how many successive transitions the pixel has made within the last second; hence it is the frequency counter of that pixel. There are also timers and frequency counters at the frame level, denoted as *FrameTimer* and *FrameFCount*.

The algorithm for pixel-level successive transition check is shown in Fig. 3 as a flowchart. First, the presence of a transition is checked. If a transition is not an opposing transition, then the corresponding timer of the pixel is restarted. However, since it is not an opposing transition, *PixelFCount* is not incremented. It should be noted that even if a transition is opposing, it may not be counted as a successive transition, if the *PixelFCount* is greater than *FrameFCount*. This is due to the concurrent flashing requirement of pixels, in order to be counted as a flash. For instance, if there are three successive frame transitions within the last second, then any *PixelFCount* should not be above four.

Finally, if an opposing transition satisfies the frequency counter requirement described above, then it is registered as a successive transition. In this case, the latest transition direction is updated, the frequency counter of the pixel is incremented, and the timer is restarted.

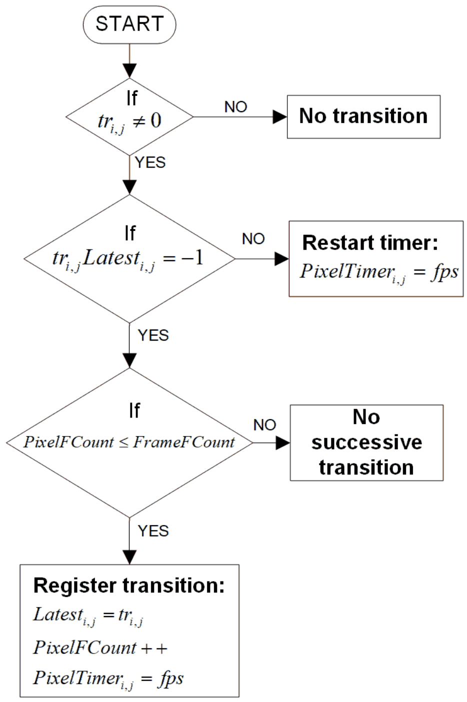

Figure 3: Successive transition detection in the pixel level

Flashing Area and Frequency Check

As explained in the previous section, each pixel has an assigned *PixelFCount* value, showing the number of successive transitions within the last second. In order to register a successive transition in the frame level, the area check should be performed. To ensure that the transitions are occurring concurrently, the pixels with the same value of *PixelFCount* values should be counted. For instance, if there are three successive transitions registered in the frame level (i.e. *FrameFCount* = 3), then area check should be performed to the pixels having *PixelFCount* = 4, as the previous pixel successive transitions were already counted. This algorithm is shown as a flowchart in Fig. 4.

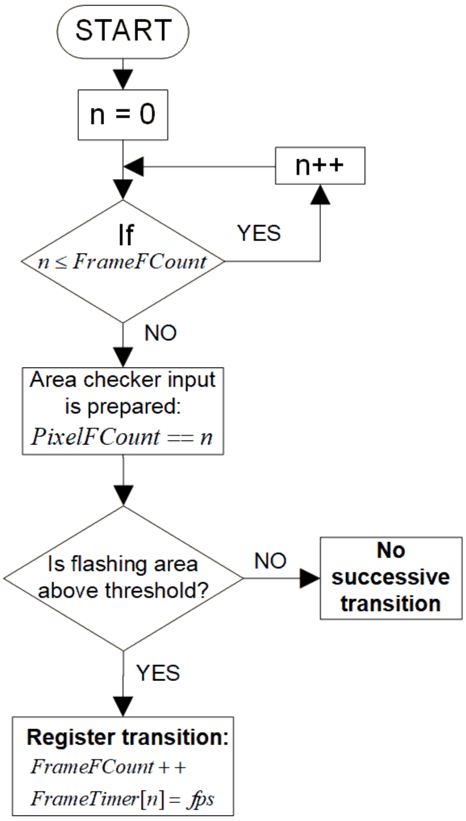

Figure 4: Successive transition detection in frame level

According to the flashing area criterion, the flashing area should be at least 25% of any 10-degree visual field. To test this criterion, a boolean array with True values corresponding to the pixels whose frequency counter exceeds frame frequency counter is prepared. This boolean array is fed to a uniform filter of kernel size that equals 10-degree visual field. If any element of the filter output exceeds 25% of the filter area, then it is concluded that the area threshold is exceeded. With this linear filter approach, an effective and high performance area check is performed.

After updating the frame-level number of successive transitions in the last one second (FrameFCount), the frequency is simply checked from the value of *FrameFCount*. If *FrameFCount* is greater than six, a flash is registered.

Finally, it should be noted that after processing each frame, all the timers in the timer arrays (*PixelFCount*, *FrameFCount*) are decremented, to reflect the progress of the frame to these timers. In addition, if a timer value becomes zero, the corresponding frequency count is decremented, as that transition is expired. This step is named as the preprocessing step (please refer to Fig. 1), as it is applied before reading the next frame.

Results and Discussion

To verify the flashing detector algorithm, a set of benchmark videos are generated [20]. The objective of the verification procedure is to verify the detection of each criterion separately. The properties and the resulting pass/fail status of the luminance flash detection benchmark videos are tabulated in Table 1. In Table 2, the features of three videos, prepared for red flash detection are tabulated. Each video essentially includes a rectangular flashing object or pattern, which flashes/flickers for two seconds. For each of the four criteria (frequency, luminance change, minimum luminance and flashing area), the green color indicates that the content is in the safe region, whereas the red color indicates that the threshold is exceeded. In order to register a video as risky, all the four thresholds should be exceeded, which occurs in videos 5, 7 and 11. All videos have the frame size of 480 by 360.

In order to verify the flashing area detection algorithm, videos 4, 5, 6 and 7 are prepared and tested. To identify their differences, the screenshots from these videos are given in Fig. 5. According to the 10-degree visual field sliding window calculation, the window size for a 480x360 frame should be 160x120. The flashing area of video 4 is 65x45, which is less than 25% of the sliding window area (0.25x160x120 = 4800). On the other hand, the flashing areas of videos 5 and 7 are 110x90, exceeding the area threshold. In video 7, the flash is of 2x1 alternating checkerboard type, i.e. while half of the flashing area makes a low-to-high transition, the other half is making a high-to-low transition. The algorithm labels this video as risky, which matches with the ground truth. The flashing area of video 6 is a rectangle of size 440x22, giving an aspect ratio of 20. Although the total flashing area (9680 pixel-sq) is greater than the threshold of 4800, this area does not constitute 25% of any 160x120 sliding window. Therefore, the algorithm did not register a flash in this video, again meeting with the expectations.

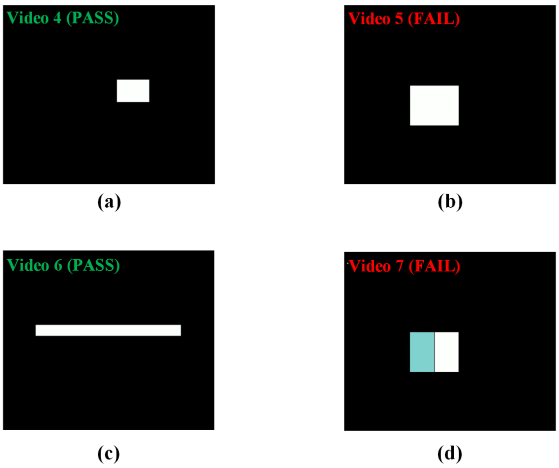

Figure 5: Screenshots from videos 4, 5, 6 and 7.

It is also tested whether the algorithm correctly implements the exception of fine and balanced transitions, described in the WCAG 2.3.1 document. For this purpose, a video with an alternating checkerboard pattern, where each square is 15 x 11 pixels, is created. A screenshot from this video is shown in Figure 6. For a video of size 480 x 360, a 0.1 degree visual field square corresponds to 16 x 12. Therefore, it is ensured that the square size is slightly lower than the 0.1 degree visual field. The algorithm successfully implements the exception and marks video 8 as safe.

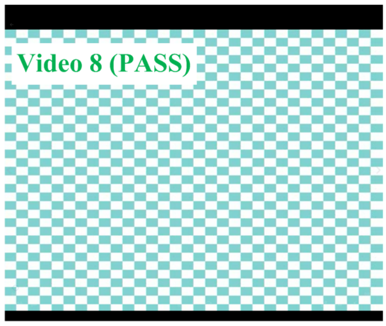

Figure 5: Screenshot from video 8.

Conclusion

A novel flashing content detection algorithm that complies with WCAG 2.3.1 guidelines is developed. The algorithm detects successive intensity transitions in the pixel level and effectively calculates the flashing area and frequency, using pixel and frame level frequency counters and timers. The accuracy of the algorithm is verified using eleven synthetic videos. The algorithm does not output any false positive or false positive results. It is concluded that the proposed algorithm can be implemented as an automatic epileptogenic visual content detection tool.

References

1. Zack MM, Kobau R. National and state estimates of the numbers of adults and children with active epilepsy — United States, 2015. *MMWR*. 2017;66:821–825. DOI: 10.15585/mmwr.mm6631a1.
2. Phososensitive Epilepsy, Epileptic Society, visited on 09/2022, URL: https://epilepsysociety.org.uk/about-epilepsy/epileptic-seizures/seizure-triggers/photosensitive-epilepsy
3. L. Carreira, N. Rodrigues, B. Roque and M. P. Queluz, "Automatic detection of flashing video content," 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX), 2015, pp. 1-6, doi: 10.1109/QoMEX.2015.7148104.
4. Pokemon Episode 38: Electric Soldier Porygon (no subtitles, raw), visited on 09/2022, URL: https://archive.org/details/PKMNEP38-ElectricSoldierPorygon-NoSubs
5. G.F.A. Harding, P.M. Jeavons, “Photosensitive epilepsy”, London, MacKeith Press, 1994.
6. A.J. Wilkins, “Visual stress”, Oxford University Press, 1995.
7. C.D. Binni, J. Emmett, P. Gardiner, “Characterising the flashing television images that precipitate seizures”, SMPTE, vol.111, pp. 32-39, 2002.

8. A.J. Wilkins, J. Emmett, G.F.A. Harding, “Characterizing the patterned images that precipitate seizures and optimizing guidelines to prevent them”, Epilepsia, vol. 46, pp. 1212-1218, 2005.
9. Ofcom, "Guidances Notes, section 2: harm and offence", Issue 10, 23 July 2012
10. “Guidance for the reduction of photosensitive epileptic seizures caused by television”, R. ITU-R BT.1702, ITU-R, 2005
11. “Understanding Success Criterion 2.3.1: Three Flashes or Below Threshold, visited on 09/2022, URL: https://www.w3.org/WAI/WCAG21/Understanding/three-flashes-or-below-threshold.html
12. S. Clippingdale, H. Isono, " Photosensitivity, broadcast guidelines and video monitoring", IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2, pp. 22 - 27, 1999
13. J. Kothari, A. Srivastava, “Flikcer -- A Chrome Extension to Resolve Online Epileptogenic Visual Content with Real-Time Luminance Frequency Analysis,” 2021, arXiv, doi: 10.48550/ARXIV.2108.09491,
14. "Harding Flash and Pattern Analyser", visited on 1/2015, URL: http://www.hardingfpa.com/
15. "Hitachi flicker check", visited on 09/2022, URL: http://www.hitachi-solutions.com/pdf/flickercheck.pdf
16. Photosensitive Epilepsy Analysis Tool (PEAT), Trace R & D Center, U. of Maryland, visited on 09/2022, URL: https://trace.umd.edu/peat/
17. International Standard, IEC 61966-2-1 Amendment 1, IEC 2003, visited on 09/2022, URL: https://www.sis.se/api/document/preview/562720/
18. G176: Keeping the flashing area small enough, visited on 09/2022, URL: https://www.w3.org/WAI/WCAG21/Techniques/general/G176
19. G19: Ensuring that no component of the content flashes more than three times in any 1-second period, visited on 09/2022, URL: https://www.w3.org/WAI/WCAG21/Techniques/general/G19
20. Luma test Cases, URL: https://zenodo.org/record/7049057

Table 1: Features of benchmark videos used in the verification of luminance flash detection algorithm

Video #	Freq.	Lum Change	Min. frame lum	Flash area	Ground Truth	Result of this work
1	< 3 Hz	> 10%	< 0.8	> threshold	**PASS**	**PASS**
2	> 3 Hz	< 10%	< 0.8	> threshold	**PASS**	**PASS**
3	> 3 Hz	> 10%	> 0.8	> threshold	**PASS**	**PASS**
4	> 3 Hz	> 10%	< 0.8	< threshold	**PASS**	**PASS**
5	> 3 Hz	> 10%	< 0.8	> threshold	**FAIL (LUM)**	**FAIL (LUM)**
6	> 3 Hz	> 10%	< 0.8	< threshold*	**PASS**	**PASS**
7	> 3 Hz	> 10%	< 0.8	> threshold**	**FAIL (LUM)**	**FAIL (LUM)**
8	> 3 Hz	> 10%	< 0.8	< threshold***	**PASS**	**PASS**

*: The flashing area has a high width/height ratio of 20.
**: The flashing area exhibits a 2x1 alternating checkerboard pattern.
***: The flashing area exhibits an alternating checkerboard pattern with each square less than 0.1 degree visual field

Table 2: Features of benchmark videos used in the verification of red flash detection algorithm

Video #	Freq.	Red L. Change	Max. Red saturation	Flash area	Ground Truth	Result of this work
9	> 3 Hz	> 20	< 0.8	> threshold	**PASS**	**PASS**
10	> 3 Hz	< 20	> 0.8	> threshold	**PASS**	**PASS**
11	> 3 Hz	> 20	> 0.8	> threshold	**FAIL (RED)**	**FAIL (RED)**