Capturing and grouping SDR frames containing sections of HDR from a video feed to artificially expand the dynamic range of SDR screens

Rinke Schreuder, Elmar Eisemann, Ruben Wiersma
TU Delft

Abstract

In this paper, a method is proposed to artificially expand the dynamic range of screens with a limited dynamic range. This research is linked to a new film-making technology where, instead of using a green screen, the background of a scene is displayed on a screen in real time using a computer generated background. This provides real time lighting in the studio; however, due to the limited dynamic range of the screen, it can not fully replicate the brightness of light sources. Overcoming this problem involves capturing and synchronize frames that each display a small section of the wider dynamic range, defined as illumination maps. The method uses a pipeline in which the illumination maps are displayed on a monitor in a grouped order, which are then captured with a camera. The recording is processed by labeling the frames and selecting key frames. The key frames are then additively combined with compatible illumination maps, which result in a video of the full dynamic range. A program was developed as a proof of concept, providing expected results. For various recording inputs, It was also found that the implemented program discarded a lot of the frames of the recordings. A variation of the proposed method also yielded a slight speed-up, for practically the same results.
The proposed method provides a good starting point tackling the problem of artificially extending the dynamic range. The program used is a step in the right direction, but has flaws that limit its usefulness.

1 Introduction

1.1 Context

In recent film-making, there is a development where, instead of a green screen, an LED wall is used. The LED's display a computer-generated background in real-time. This technology has been used in the productions of 'The Mandalorian' and the remake of 'The Lion King', both by Disney[1, 2]. This LED wall has the benefit that it is not necessary to key out the green colour and insert a background of the scene. It also provides somewhat more realistic reflections compared to a green screen, where the reflected green has to be edited out and replaced with computer generated reflections.
There are also some downsides to using a screen as the background. The screen only has a Standard Dynamic Range (SDR), causing high dynamic range (HDR) images to lose detail in the displayed images. The full brightness of light sources and reflections of the image can not be represented on the screen. For example, if a scene is set in a sunny environment the sun can appear as a dimmed white on the screen. In these sunny scenes, close-ups can still be shot with this screen as the background, as the brightness is close enough, while it is better to capture wide-shots on a separate set outside.
Furthermore, the setup for this technology is very expensive. It is not a viable setup for low-budget projects. Cheaper monitors usually have SDR and lack built-in features to synchronize with a camera. Therefore it is important to look at the problems of using an LED screen and to make an attempt at solving certain parts of those problems.

HDR

The problem that needs to be solved is how to artificially expand the dynamic range of an SDR screen. This can not be done by increasing the brightness values of the image, as objects which should have a low illumination will also get an increased brightness value. Therefore the dynamic range should be split into sections of brightness. These sections are defined as illumination maps and can be amplified separately to increase the dynamic range. A camera is needed to capture each illumination map accurately, after which the frames are processed. This will be the goal of this paper.
Solving the problem involves building a software program. This program should be able to display the illumination maps at a desired frame rate and it should process the camera recording.

1.2 Problem description

The problem

The screens used in the film production are not able to display the full dynamic range of the image. Therefore a

method must be created to display parts of the dynamic range separately and then capture, synchronize and merge these ranges.

The following are formal descriptions of some relevant terms.

Images
The original HDR image will be defined as I_{HDR}. Then illumination maps can be defined as SDR elements of the original image $I_{SDR}^{0...n}$. These elements each have a weigth w, which is used to adjust the strength of the element. The equation for these is shown in Equation 1. Here n represents the number of illumination maps. The images captured by the camera are defined as images $I_{CAM}^{0...t}$ and the reconstructed image is $\hat{I}_{HDR}$.

$$I_{HDR} = \sum_{i=1}^{n} w_i \cdot I_{SDR}^i \tag{1}$$

The problem can now be defined as follows: The camera should produce t frames, then for each SDR image i there exists a frame j, such that $I_{CAM}^j = f(I_{SDR}^i)$. Here, $f(x)$ represents some function that transforms the image based on the ISO, shutter speed and aperture settings on the camera and whether there is an object in front of the screen. This function should still uphold the relationship between frames, i.e. $f(I^i) - f(I^{i+1}) \sim I^i - I^{i+1}$.

Blended frames
As mentioned previously, certain frames will blend. This means that $I_{SDR}^i = I_{CAM}^j$ does not hold exactly. A blended frame is a frame that is a combination of two illumination maps. An equation to represent blended frames can be found in the OpenCV documentation[3, 4]. The applied case is shown in Equation 2.

$$I_{CAM}^j = f((1-\alpha) \cdot I_{SDR}^i + \alpha \cdot I_{SDR}^{i+1}) \tag{2}$$

Difference
The key frame I_K of a subset $I_{CAM}^{k...l}$, where $0 \leq k < l \leq t$ is determined by image difference. The difference is based on a pixel-wise comparison, that is found in the paper by Koprinska and Carrato [5]. The function is shown in Equation 3. $D(i,j)$ represents the difference between an image i and image j. P is a pixel at screen coordinates x and y. X and Y represent the screen dimensions. The key frame is the frame I_{CAM}^j ($k \leq j \leq l$) that represents a subset for I_{SDR}^i. Key frames $I_K^{0...n}$ are defined to be compatible if they can be combined to form $\hat{I}_{HDR}$, i.e. $\sum_{i=1}^{n} w_i \cdot I_K^i = \hat{I}_{HDR}$.

$$D(i,j) = \frac{\sum_{x=1}^{X} \sum_{y=1}^{Y} |P_i(x,y) - P_j(x,y)|}{XY} \tag{3}$$

1.3 Contribution

This research will focus on the capturing, the synchronizing and the merging of frames to artificially extend the dynamic range of a screen with SDR. The idea is to select key frames from a feed of captured illumination maps, and then merging the compatible frames, to recreate the original image. The key frames will be selected based on which is most similar to or has the least difference from the illumination map cluster. This method will be compared to a recording of a feed where the maps are already merged, which is used as a baseline. This way, it can be shown if using the 'select-and-merge' provides a significant difference in terms of increasing the dynamic range.

2 Related works

The main method of this research is inspired by the proposed implementations for temporal video segmentation of Koprinska & Carrato [5] and Sokeh et al. [6]. These papers focus on the processing of distinguishing elements in a video, such as boundary shot detection. The paper by Koprinska & Carrato specifically provided the function for calculating the difference. The papers also provided the idea of using key frames.
There are several papers that attempt to expand the dynamic range using several LDR images to obtain a HDR image, each with their own method.
Firstly, the paper by Sun et al. [7] uses a method using so-called disparity maps, but focuses mostly on HDR involving fast movement.
Secondly, Jinno & Okuda [8] combine exposures and estimate the 'irradiance' value for each pixel. The focus here is to reduce ghosting artifacts in an attempt to obtain motion-blur-free HDR images. They also propose a weighting scheme for fusing multiple images.
Thirdly, in the research by Vavilin & Jo [9] a similar method is presented, directed at images with fast motion. Their method combines three LDR exposures to create an HDR image. Besides a weighted fusion, they also include error maps.
Lastly, Grossberg & Nayar [10] present a method which focuses on which exposures should be combined. They provide proof that simple summation is sufficient for combining exposures.
A paper by Petković et al. researches the possibility of using synchronization to solve the problem of displaying and capturing sections of an image [11]. Their method is used for implementations such as 3d body scanning using hardware. They compare different synchronizations, software implementation being the most relevant. This paper is linked to determining the display sequence.

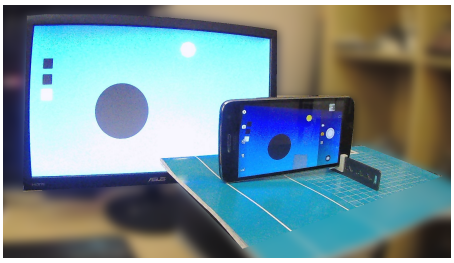
3 Method

This section will describe the process used for solving the research questions. The setup is described, explaining all components that are involved. This section will also explain the pipeline that is used for getting results and how the program should work. It also discusses a small variation of the pipeline.

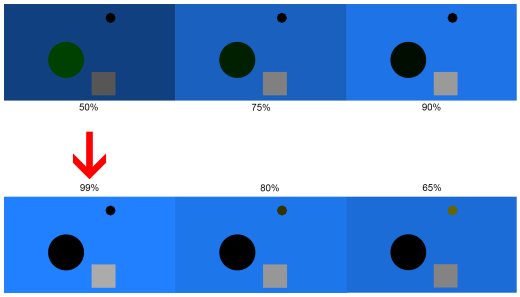
3.1 Pipeline

The pipeline consists of displaying, capturing, selecting and merging. For this research, a simple setup is used with a computer monitor and a camera. The monitor is used to

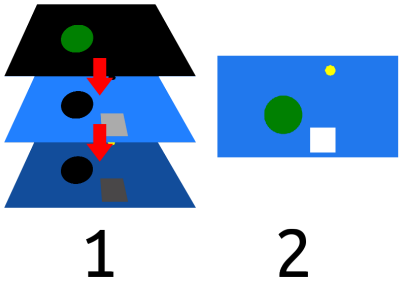
display $I_{SDR}^{1...n}$, while the camera captures $I_{CAM}^{1...t}$ from the monitor. The video recordings will then be used as input for the program. From this input, $I_K^{1...n}$ are selected to be merged into a frame $\hat{I}_{HDR}$. The pipeline is shown in Figure 1.



(a) Steps 1 & 2: Frames are displayed on the monitor sequentially. Meanwhile the camera captures the screen.



(b) Step 3: The selection of key frames from a sub-list of frames. The most similar frame is chosen. In this scenario the fourth frame is most similar to the second illumination map.



(c) Step 4: The merging of key frames (1) to obtain the complete image (2).

Figure 1: Pipeline process

Display and Capture

To be able to transfer the image data displayed on the screen to the camera accurately, the frames will be put in a grouped sequence. As an example, groups that are each one frame per map results in the sequence $(I^{1,t}, I^{2,t}, ...I^{n,t}, I^{1,t+1}, I^{2,t+1}, ..., I^{n,t+1}, ...)$. $(n, t$ indicates the n'th illumination map of the original HDR frame $t)$ and for groups that are 2 frames per map it results in $(I^{1,t}, I^{1,t}, I^{2,t}, I^{2,t}, ..., I^{n,t}, I^{n,t}, ...)$. The sequence influences how much data can be accurately transported, because if the display and the camera are not synchronized, the camera will capture blended frames. To give the transfer of each frame more precision, the display of the frame on the screen could be displayed multiple times.

The display rate of the screen will be a set rate r_d. The camera will be set at a default capture rate $r_c = 60fps$. This rate is selected as it is a commonly used speed and is available on most cameras. The standard display rate will be at a fraction of the capture rate, for example $r_c = 3r_d$. The reason to display at a lower rate than the capture rate is to guarantee that there is at least one frame that is not blended. Higher display rates, such that it can not not guaranteed that there is an non-blended frame, will also be tested. The use of the different display rates will be compared in the results.

The Program

To determine exactly how the frames should be displayed, a brute force program was created. In this program, r_d can be set manually. After the program has concluded displaying the maps, it continues with the next steps of the pipeline. It takes in the captured video and applies the following.

Key frame selection

The frames $I_{CAM}^{0...t}$ are assigned to groups for which the difference $D(I_{SDR}^i, I_{CAM}^j)$ is minimal. To accurately measure D, the frame and the map should be aligned as much as possible. Ideally this is done by setting the camera on a tripod to stabilize the camera. This is not always possible, thus a software solution serves as a substitute to align images. For simplicity, this alignment will only be a manual translation or a manual selection of certain Regions of Interest (ROIs), which can be compared separately. Then the sum of the differences is measured: $\sum_{k=1}^{m} D(ROI_k(I_{SDR}^i), ROI_k(I_{CAM}^j))$, where m represents the amount of ROIs.
After the subsets $(I_{CAM}^{0...k}, I_{CAM}^{k...l}, ..., I_{CAM}^{x...t})$ are formed, the key frame I_K^i from each subset i is selected to represent I_{SDR}^i. The key frame is chosen using $I_K^i = min_{k \leq j \leq l}(D(I_{SDR}^i, I_{CAM}^j))$.

Merging

Next the selected frames are going to be merged. The merging is the weighted addition of the key frames into one image, following from Equation 1. For this paper, the focus is not to find the weights, therefore the weights will be held constant: $w_i = 1$ for all i. It is also proved in [10] that a simple summation of exposures combines all information of the individual exposures without loss. The output of the program is a video of the sequence of merged images $\hat{I}_{HDR}^{0...x}$.

3.2 A variation on the pipeline

The suggested program compares every frame with every illumination map on multiple Regions of interest. There is a small improvement that can be made in this step. The idea is that the features of the image itself are not considered as ROI, instead markers are manually placed. Each marker represents one of the each maps. These markers are black, except for one white one that is unique to the map. An

example is shown in Figure 2. In this case, it is no longer required to make comparisons with every illumination map for each frame, as only the markers need to be checked.

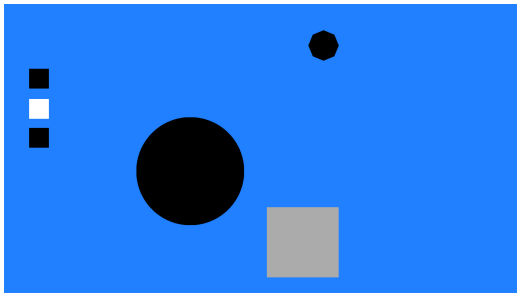


Figure 2: The variation method uses separate squares as ROIs instead of the shapes of the image. The white square signifies the current map.

4 Experimental Setup and Results

Physical setup

For testing, the camera used is a phone camera, recording at a Full HD resolution. The monitor is has a resolution of Full HD with Standard Dynamic Range, which are standard settings for most commercially available computer monitors. Note that not every monitor uses the same colour range and that phones can vary in camera quality.

Software setup

The program is implemented in C++ using the image editing library OpenCV [3]. The OpenCV library provides a wide range of image operations. These are useful for displaying the illumination maps and editing individual frames from the video footage. It is also able take the result frames and output them as a video.

4.1 Implementation analysis

This section contains an analysis of the created program. It focuses mostly on the merging algorithm used, as this contains the actual work for the research.

Program

The program contains two parts: the displaying of the illumination maps and the processing of the camera recording. Here, the latter will be explained and this part will be referred to as the program.

Before running the program, the ROIs are manually selected. This is done by selecting parts of an image. The program starts by looking for the first illumination map. It scans through the frames, calculates if the current frame is detected as the required map. It does this by calculating the difference of the frame and each map, and selecting the 'minMap' that has the minimum difference. It halts if the 'minMap' is equal to the first map (this can be calculated using the difference function or simply by comparing pointer addresses). This starting step is purely done to provide a constant start for merging frames in later steps. Afterwards, the program continues by executing the pipeline as described. The program can also run the method variation, which only requires a different selection of ROIs.

One issue with the current program is how it deals with the frames that are appear in an incorrect sequence. Currently, if a frame j representing I^{x}_{SDR} appears before a frame i representing I^{x+1}_{SDR}, where i should have appeared before j, they are discarded. From any subset $I^{k...l}_{CAM}$, only one key frame is selected. Given a subset that is 4 frames large, 3 frames will be discarded. This means that a lot of frames from the recordings will not be used in the final product.

Complexity

The merging program goes through every frame of the input video. Therefore the program has at least a linear time complexity. In this brute force approach, for each frame every illumination map is compared. The comparisons are made on the specific ROIs.

Let $N = |\#frames|$, $M = |\#illumination\_maps|$ and $R = |\#ROI|$. It is assumed that every ROI contains a constant amount of pixels to compare. Combining this, the theoretical time complexity becomes $O(NMR)$. Although this is not a great time complexity, it is good enough for small videos with a limited amount illumination maps.

In the variation of the method a small improvement is made, since the ROI are not checked on every illumination map. Instead, each ROI is immediately linked to an illumination map. Thus the ROIs are only compared for every frame. This reduces the time complexity to $O(NM)$ (or the equivalent $O(NR)$).

Mocked input

The input should be a video feed of illumination maps. However, these need to be generated first. Instead, the illumination maps will be simulated. They are mocked by creating the same image several times, each with different intensities. Each mocked map should show different elements of detail. An example of mocked maps is shown in Figure 3. The mock images represent the maps $I^{1,2,3}_{SDR}$, each with an exposure, I^{1}_{SDR} for a long exposure, I^{2}_{SDR} a medium exposure and I^{3}_{SDR} a short exposure.

4.2 Results

Visual results

After running the entire program using the first recording as input, we get a result as seen in Figure 4. This frame shows that the details of each different illumination map are indeed combined correctly. However it is very overexposed across the image, due to external light leaking onto the screen, which is increased when overlaying the key frames. Using another recording, where the camera was set to capture setting the exposure -2 and using automatic ISO settings. It was made sure that there was no other light source falling onto the screen, as it would give the image an uneven brightness. A correct frame result is seen in Figure 5. Comparing this image to the 'standard' image seen in Figure 6, this image shows the circle, sun and square clearly, with a deeper blue sky.

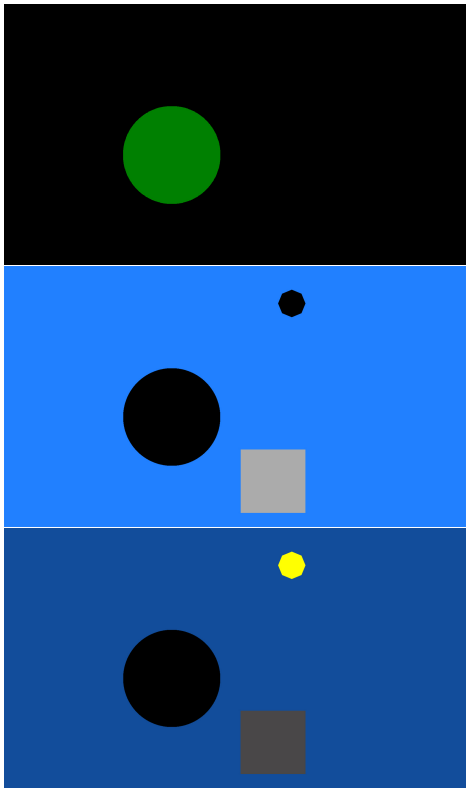


Figure 3: Mocked Illumination Maps, representing long, medium and short exposures respectively.

Displaying at a higher rate also provides some good frames such as shown in Figure 7a, but is less consistent. For example, using a display rate of 40 fps has certain frames that did not merge together very well, like shown in Figure 7b.
This indicates that there was some mismatch in the similarity of the illumination map and that means it became more likely for frames to blend. One noteworthy aspect is that the original recording of this display rate already contained frames that were out of order. This means that certain maps were not visible for long enough for the camera to be able to capture them at all.

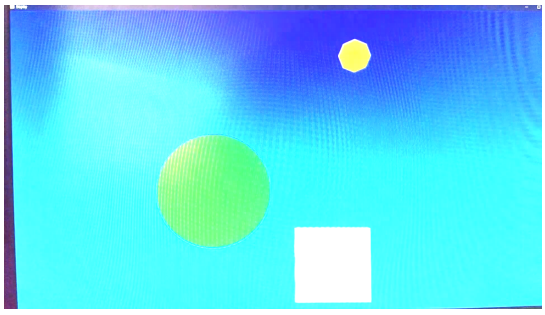


Figure 4: One example frame after processing a stabilized video. Display rate of 20 fps.

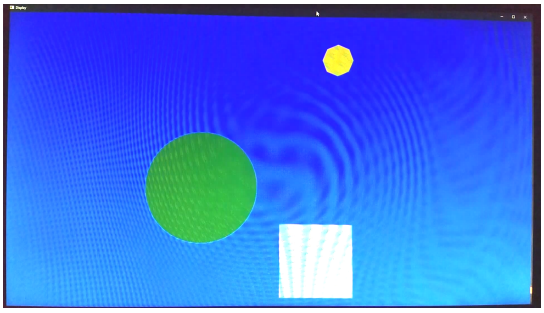


Figure 5: A similar resulting frame using a lowered brightness setting on the camera. Display rate of 20 fps.

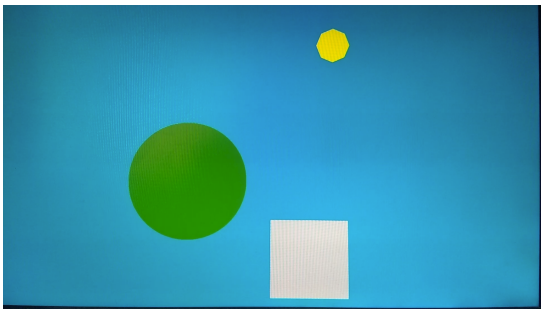
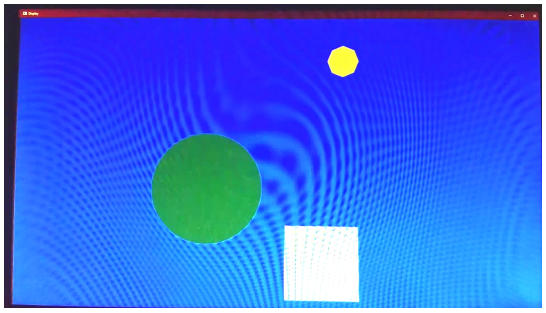


Figure 6: The 'standard' image. This is the result of merging the illumination maps before merging. This image is used as comparison, to verify the usefulness of the program. Display rate of 20 fps.

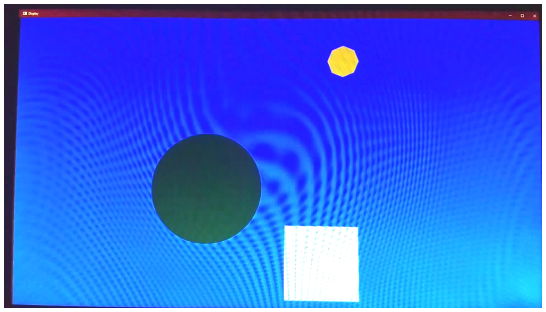
Other results

The program currently works well enough even if the ROIs are partially obscured by an object, such as in Figure 8. This will not work if the ROIs are completely blocked from the camera. For this method, where each shape represent an ROI, this means that these shapes can not be obscured by an object or actor in front of the screen. Furthermore, given an object that sligtly moves in front of the screen, the results are still similar. However there is a slight 'ghosting' of the object (Figure 9).
The variation on the program presented in the method, where the ROIs are separate squares, deals with this problem. Results for this variation are shown in Figures 10 and 11. The latter image shows the shapes being obscured completely.

One detail that is not immediately noticed in either method is how frames are discarded. The first way is that frames are discarded because they are not selected as a key frame. In the current program this is expected. The second way is that a frame is discarded if it is detected as 'out-of-order', which is unexpected. The data is shown in Table 1. The original method is represented with 'ori' and the variation with 'roi'.



(a) A correct frame from the output video.



(b) An incorrect frame from the same output. The circle does not have the correct colour.

Figure 7: Resulting frames using 40 fps display rate.

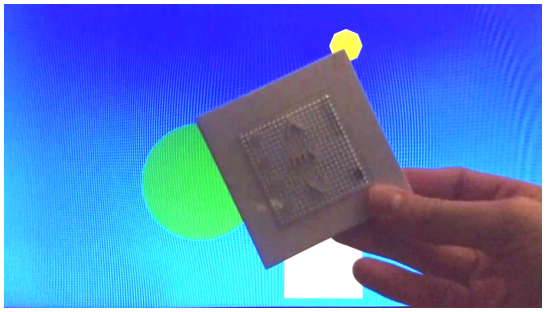


Figure 8: The output frame where an object is held partially in front of the details of the images. Display rate of 20 fps.

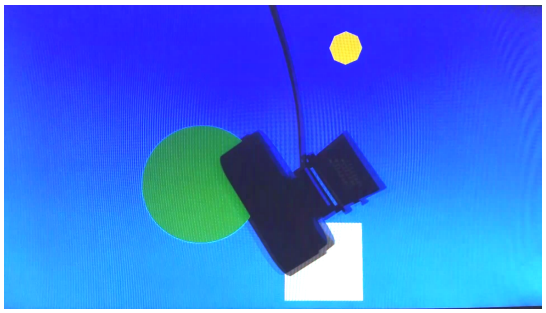


Figure 9: The webcam dangling in front of the images caused a slight ghosting effect. Display rate of 20 fps.

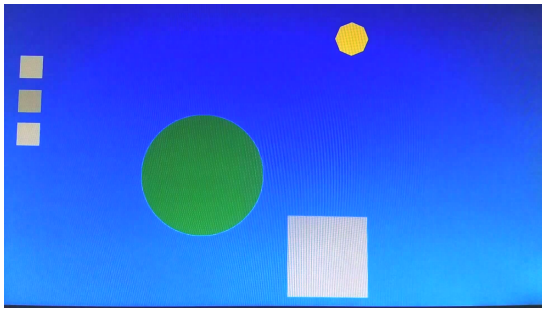


Figure 10: The output frame using the variation of the method. Display rate of 40 fps.

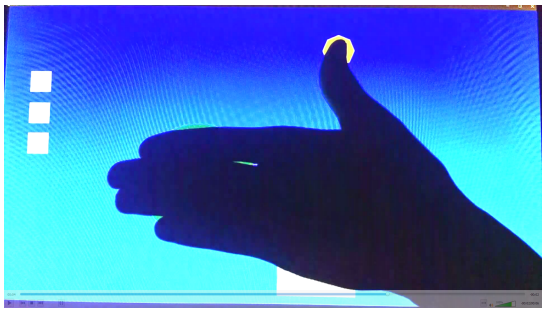


Figure 11: The output frame using the variation of the method, with the shapes completely obscured. Display rate of 20 fps.

Method	Display rate (fps)	Frames not used	Frames out of order	Total frames discarded	Total amount of frames
ori	20	410	18	428	882
ori	40	77	95	172	699
ori	60	101	158	259	717
roi	20	356	0	356	751
roi	40	76	102	178	699
roi	60	91	131	222	717

Table 1: The number of discarded frames for different display rates. The method is either the originally described method 'ori' or the variation of the method 'roi'. The expected and unexpected discarded frames are counted separately and combined, out of a total amount of frames (the length of the recording).

5 Responsible Research

5.1 Reproducibility

The ethical side of this research mainly focuses on the ability to reproduce the experiment. This is achieved through the pipeline, of which each step is given. The provided pipeline, steps given in the method and the description of the implementation show the process to achieve a similar result. The largest differences will be in the equipment used for displaying and capturing. Most monitors have a different colour space, and each phone has a different quality camera. This may cause some deviation in the results, however if similar equipment is used it is unlikely to significantly affect the outcomes.

5.2 Other concerns

Besides reproducibility, there are some concerns related to increasing realism in photography, which are important to consider. The intentions of improving the dynamic range of video are to create more realistic imaging using only equipment of a lower budget. The concern is that making a video feed more realistic could be used for more questionable purposes. As an example, with a subject like Deepfakes is already difficult to tell the differences between real and fake [12]. There are already concerns in the use of these type of videos and the more realistic they can be made, the more people will worry about their impact.

6 Discussion

The results of the program are very promising. The output images $\hat{I}_{HDR}$ were reconstructed using the correct frames, in both the original method and the variation. The colours of the result image are vibrant and clear, while the colours of the 'standard' image are dulled. This suggests that the key frames are selected correctly and that the compatible key frames are merged. Furthermore, the variation is able to deal with problems such as objects in front of the camera and it provides a slightly better time complexity.
The downside of the implemented program is that it does not correctly deal with all frames. In the ideal situation, no frames would be discarded. Currently it discards too many frames from the recording. With higher display rates, the program discards significantly more frames because they are out of order.
However the total amount of discarded frames is still highest for 20 frames per second, even though there are almost no unexpected discarded frames. This is likely because the each subset is larger, as each map is repeated several times, causing more frames to be discarded because they weren't used. Other than that, the total amounts of discarded frames remain similar between the methods.
One could argue that, given a screen with infinite resolution infinite dynamic range and a correctly stabilized camera could provide very high quality results without this solution. This is however not the goal of this research, as it focuses explicitly on the usage of low budget equipment.

7 Conclusion and Recommendations

7.1 Conclusion

The goal of this paper was to find a way to use a camera to capture, synchronize and merge HDR images using an SDR screen, and using that to artificially expand the dynamic range of that screen. The proposed method provides good results for simple situations and given the right setup. Therefore it can be said with reasonable confidence that the proposed method and the program work well as at least a proof of concept for the artificial expanding of the dynamic range. The implemented program does have significant limitations, such as the amount of frames it discards and the time and precision required to use the setup, which make it less applicable for larger scale projects.

7.2 Recommendations

Blended frames

Even though the proposed method provided decent results, there is definitely room for improvement given more time. One suggestions is to create an implementation in which one can also use the frames that are currently being discarded. The blended frames will have to be reconstructed by measuring how much of each map is contained in the frame. The resulting frame will then be a more complicated mix of frames, however it ensures that less frames become redundant. In addition to dealing with blended frames, one could try to integrate the frames that are marked as out of order. As these are more common at higher display rates, it could mean an improvement on the final speed of the video.

Sequence

In the method the frames that are considered out of order are being discarded. The suggestion here is to still use these frames. In this case, the frame that should be in between the present frames will be replaced with an 'empty' frame. This empty frame will be represented by a black image and is inserted into the sequence. Using an empty frame should ensure that the sequence remains the same length.
As an example: a sequence that appears as $(I^{1,t}, I^{2,t}, I^{1,t}, I^{2,t}, I^{3,t})$. This becomes $(I^{1,t}, I^{2,t}, I^{2,t}, I^{3,t})$ with the original method where frames are discarded. Using the variation of the method, this same sequence will become $(I^{1,t}, I^{2,t}, Empty, I^{1,t}, I^{2,t}, I^{3,t})$, thus preserving the sequence length.
From this suggestion, it could continue by interpolating the frames. The empty frames can be filled or replaced with image data retrieved from frames surrounding them. This would lead to a completed sequence where all result frames contain the detail of each map.

Optimization

Another suggestion is to figure out if it is possible to create a solution which is optimal to this problem. This brute force method works fine for small amounts of frames and illumination maps, yet for larger data it could become unwieldy and impractical. Right now, all frames are being compared using the ROIs. Perhaps there is a solution where only a limited amount of ROIs have to be compared. Another idea is to check only every other frame, as an estimation or guess could be made based on the frame before and the frame after.

Camera Settings

Currently, the camera setting that was focused on the most was the exposure. The ISO and the aperture were left on automatic. It is recommended to attempt this method with different settings.

References

[1] Jay Holben. The Mandalorian: This is the way. https://ascmag.com/articles/the-mandalorian, 2020. [Accessed 2021-04-22].

[2] Devin Coldewey. How The Mandalorian and ILM invisibly reinvented film and TV production. https:

//techcrunch.com/2020/02/20/how-the-mandalorian-and-ilm-invisibly-reinvented-film-and-tv-production/, 2020. [Accessed 2021-05-03].

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] Ana Huamán. Adding (blending) two images using OpenCV. https://docs.opencv.org/master/d5/dc4/tutorial_adding_images.html. [Accessed 2021-06-24].

[5] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001.

[6] Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Superframes, a temporal video segmentation. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 566–571. IEEE, 2018.

[7] Ning Sun, Hassan Mansour, and Rabab Ward. Hdr image construction from multi-exposed stereo ldr images. In *2010 IEEE International Conference on Image Processing*, pages 2973–2976. IEEE, 2010.

[8] Takao Jinno and Masahiro Okuda. Multiple exposure fusion for high dynamic range image acquisition. *IEEE Transactions on image processing*, 21(1):358–365, 2011.

[9] Andrey Vavilin and Kang-Hyun Jo. Fast hdr image generation from multi-exposed multiple-view ldr images. In *3rd European Workshop on Visual Information Processing*, pages 105–110. IEEE, 2011.

[10] Michael D Grossberg and Shree K Nayar. High dynamic range from multiple images: Which exposures to combine. In *ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, volume 16, 2003.

[11] Tomislav Petković, Tomislav Pribanić, Matea Đonlić, and Nicola D'APUZZO. Software synchronization of projector and camera for structured light 3d body scanning. In *Proceedings of the 7th International Conference on 3D Body Scanning Technologies*, 2016.

[12] WGBH Educational Foundation. Deepfake Videos Are Getting Terrifyingly Real I NOVA I PBS. https://www.youtube.com/watch?v=T76bK2t2r8g, 2019. [Accessed 2021-06-1].