EXAMINING AUTOEXPOSURE FOR CHALLENGING SCENES

BEIXUAN YANG

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

JANUARY 2024

Abstract

Autoexposure (AE) is a critical step cameras apply to ensure properly exposed images. While current AE algorithms are effective in well-lit environments with unchanging illumination, these algorithms still struggle in environments with bright light sources or scenes with abrupt changes in lighting. A significant hurdle in developing new AE algorithms for challenging environments, especially those with time-varying lighting, is the lack of platforms to evaluate and improve AE algorithms and suitable image datasets. To address this issue, we have designed a software platform to allow AE algorithms to be used in a plug-and-play manner with the dataset. In addition, we have captured a new 4D exposure dataset that provides a complete solution space (i.e., all possible exposures) over a temporal sequence with moving objects, bright lights, and varying lighting. Our dataset and associate platform enable repeatable evaluation of different AE algorithms and provide a much-needed starting point to develop better AE methods. We examine several existing AE strategies using our dataset and show that users prefer a simple saliency method for challenging lighting conditions.

Acknowledgements

I want to express my very profound gratitude to my supervisor, Dr. Michael S. Brown, for his significant support and guidance throughout my master's program. His expertise and encouragement facilitated the success of this research and thesis writing.

I wish to thank Dr. Marcus A. Brubaker for serving on my thesis committee and providing helpful feedback and constructive suggestions.

I am particularly grateful to Saikiran Kumar Tedla, who delivered valuable support and brilliant ideas in every stage of my research work.

I would also like to thank Rajat Ajayakumar and Dr. Abdullah Abuolaim for all of the resources and technical support they provided. Also, thanks to all the participants in the user study for their time and very important contribution.

Finally, I am deeply thankful to my lovely family and friends for their love, trust, and support throughout the journey. With them, the completion of this work was possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem

Autoexposure (AE) is an essential component used in modern camera's image processing pipeline. AE allows a camera's exposure settings to be automatically determined. This technology has revolutionized the field of photography by simplifying the exposure process. AE enables users without extensive photographic knowledge or skills to shoot well-exposed images. As the usage of cameras becomes more and more prevalent, the demand for instantly outputting high-quality images and videos has increased sharply. In addition, computer vision tasks, such as object detection, recognition, and tracking, heavily rely on a camera's ability to take high-quality images. Among several factors that affect image quality, auto exposure plays a significant role in producing clear scene content, which in turn enhances the performance of computer vision algorithms.

The work in this thesis is focused on developing and analyzing autoexposure for scenes

with challenging lighting. As previously mentioned, a camera's AE subsystem determines capture-time settings to ensure a properly exposed image. Exposure is attributed to the lens aperture, shutter speed, and sensor ISO gain; however, some parameters are often fixed at capture time. For example, this project focuses on the most common form of AE, known as aperture-priority AE, where the aperture (and ISO) are fixed to avoid changes in the depth of field. In aperture-priority, AE algorithms are responsible for choosing the appropriate shutter speed for the next image capture based on information gleaned from the current captured image. AE is inherently a dynamic process, where the algorithm constantly predicts changes in shutter speed to account for a changing environment. While AE is a long-standing and well-researched problem in low-level computer vision, AE algorithms, including those on commercial cameras, still struggle in challenging lighting environments.

Two primary factors make scenes challenging for AE algorithms. The first is when the scene's dynamic range surpasses the capability of the imaging sensor. In such cases, the AE algorithm must determine which part of the scene will be over-exposed or under-exposed. For this problem, it is easy to construct 3D datasets comprised of a stack of 2D images of a stationary scene captured under intense lighting with different shutter speeds. Several such exposure datasets exist and provide a complete solution space—assuming the fixed lighting and objects. The second challenging factor for AE is when there is an abrupt change in the environment lighting, such as lights switching on and off or objects moving under bright lights. Capturing datasets for this scenario is more complicated, given the temporal nature of the changing scene. To the best of our knowledge, there are no 4D datasets that provide a complete solution space over a dynamically changing environment.

This later case is the impetus for our work. Specifically, we have captured a 4D dataset using a stop-motion setup.

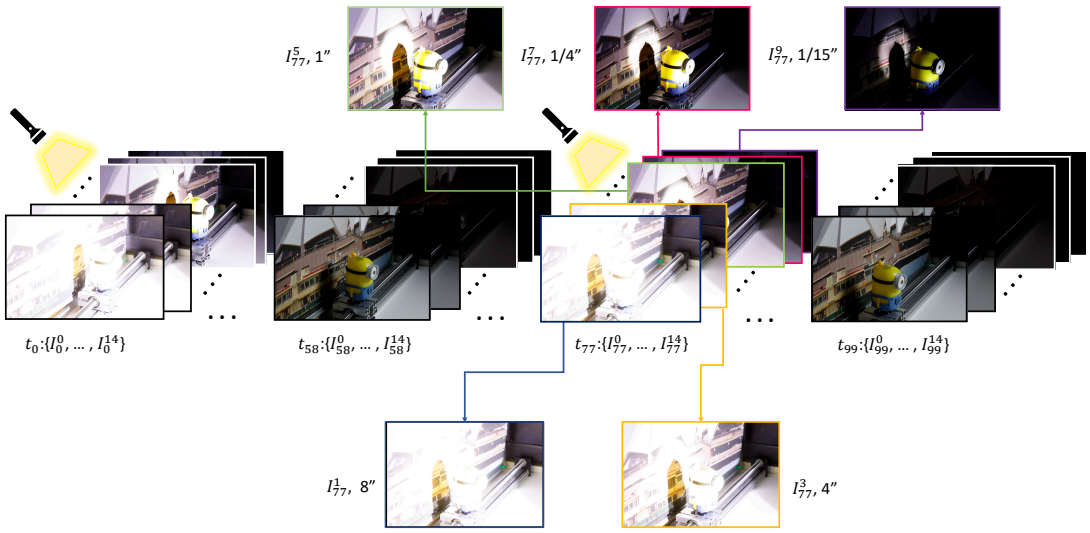

Figure 1.1: 4D exposure data of scene 9 in our dataset. The four dimensions refer to (1) time, (2) exposure setting (i.e., shutter speed), and (3&4) the 2D coordinates of the RAW sensor image associated with each (1) and (2). This scene consists of an abrupt change in lighting caused by a flashlight turning on and off.

1.2 Thesis Contribution

The work in this thesis is related to work performed for a project at York University. The project was a joint effort between fellow student SaiKiran (Sai) Telda and myself. Sai led the effort in capturing a dataset comprising nine (9) scenes. Each scene simulates 100 temporal time steps, with 15 exposures at each time step. Scenes are carefully constructed to cover a range of challenging conditions, including objects moving in front of intense lights, scenes with highly reflective materials, and scenes with rapid lighting changes.

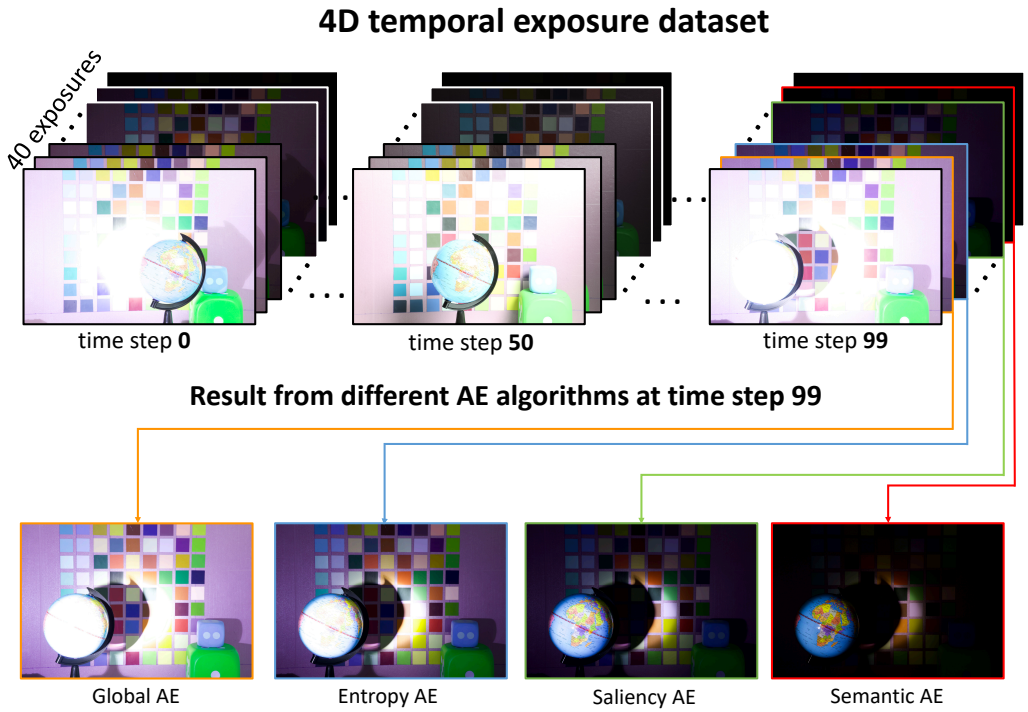

Figure 1.2: An example scene from our new 4D exposure dataset. A moving object (globe) has a light source that is periodically turned on and off, causing an abrupt change in lighting. Using our dataset and AE platform, we can evaluate different AE algorithms. Here, we show four different AE algorithms that select different exposures as their solution for a particular time step.

Figure 1.1 shows an example of a scene in the dataset. While I did not capture the dataset, I was involved in the processing of the images to make them suitable for use with the AE experiments described in this thesis.

My main role was to develop the software platform that enables different AE algorithms to operate on the captured dataset. The platform provides a basic framework to deploy an AE algorithm and test its performance. One critical feature of our platform is that it allows different AE algorithms to be run with the same input and starting conditions (e.g., starting in an over-exposed or under-exposed condition). Figure 1.2 shows that dif-

ferent AE algorithms select notably different shutter speeds at the same time steps. The platform allows output videos to show what a user would see on a camera for a particular AE algorithm and starting point. Given the mixed results of the different AE methods, we also performed a user study to determine the preferences of these different AE algorithms when used in challenging scenes. While our evaluation represents early work, most users preferred our AE algorithm based on a simple saliency method to determine regions of interest in the image. In the following chapters, we detail the construction of our dataset, an API designed to allow AE algorithm testing, the details of the various AE algorithms tested, and the results from our user study.

Chapter 2

Background

This chapter provides a background of general concepts concerning exposure for cameras. Existing works related to autoexposure (AE) algorithms are also discussed.

2.1 Understanding exposure

Image exposure is controlled through the optical system's aperture value (f-stop), shutter speed, and the ISO setting [1]. Since the aperture affects the depth of field blur and ISO amplifies the noise, most AE methods focus on adjusting shutter speed with the aperture (f-stop) and ISO fixed. We note that the ISO might change between the different scenes in our dataset but remains fixed for a given scene. In a working camera system, the AE algorithm is applied early in the camera pipeline and works directly with images in the RAW Bayer pattern format [2].

The core idea of the AE algorithms is to drive the mean of the RAW Bayer image histogram to a desired target value as shown in Figure 2.1. The main difference between

AE algorithms is how they select or weight different pixels in the image to construct the histogram used by the AE algorithm. In particular, many AE algorithms attempt to remove pixels considered "outliers" when computing their histogram. This can be as simple as excluding all dark and bright pixel values below/above some threshold or based on scene analysis to exclude pixels found in certain regions.

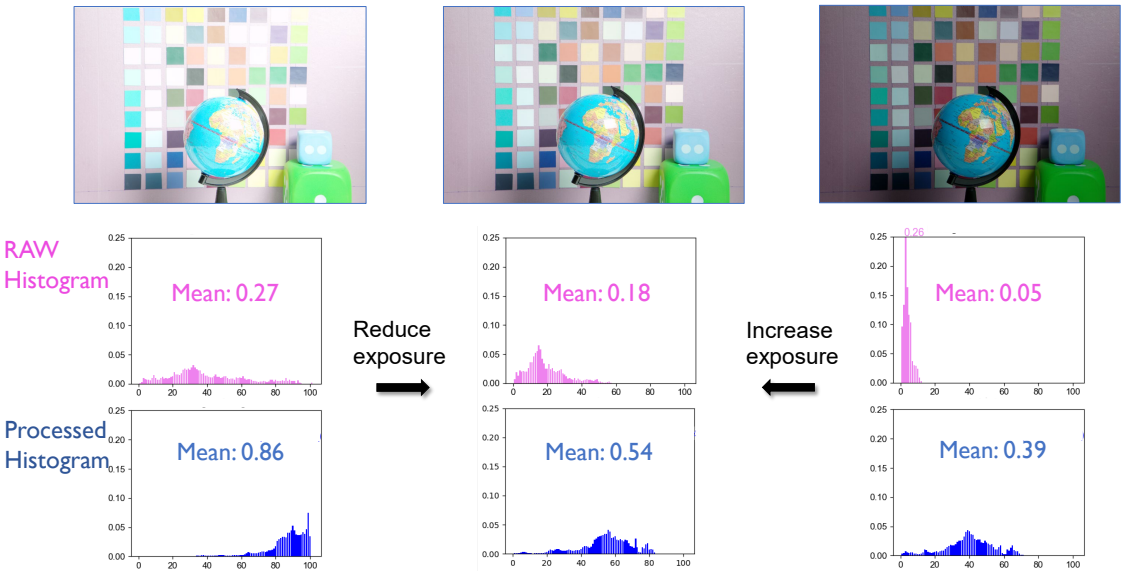

Figure 2.1: The core idea of the AE algorithms is to drive the mean of the RAW Bayer image histogram to a desired target value - 'key' (0.18 in this case). When the RAW histogram mean is higher than the key (the left side image), the amount of exposure should be reduced. And it should be increased while the mean is low (the right side image). The image in the middle is considered to be the optimal result based on the target key value. Besides the RAW histograms, the corresponding processed image histograms are displayed for reference. (Note: The pixel values are normalized from 0 to 1. The images shown are processed images. All the pixel values discussed are RAW image values.)

2.2 Related work

This section discusses AE algorithms and datasets.

AE methods. While manual exposure control is possible for still photography, robust AE is needed when capturing video or photographs in dynamic environments [3, 4]. Many AE algorithms are proprietary [5, 6] because camera manufacturers implement AE in hardware due to the need for low latency [4]. AE algorithms in the literature fall into two broad categories: (1) content-agnostic AE and (2) semantic AE.

Content-agnostic AE approaches do not explicitly consider the scene content but instead examine heuristics derived from images to determine the appropriate shutter speed [7–10]. These methods examine the global mean of the image's histogram to determine a change in the shutter speed, resulting in the current mean being changed to some target value [11, 9, 12]. Many AE methods are variants on this basic idea but modify how the histogram is constructed to avoid pixels in over-exposed and under-exposed regions contributing to the histogram [11, 13–15]. Many works also treat AE as a model-predictive control problem with the goal of fast correction after a poor exposure, based on histogram heuristics [16–18].

Two popular approaches for content-agnostic AE instead of simple histogram moments are to measure optimal exposure based on entropy and gradient. Zhang et al. [19] proposed a method where the best exposure was the shutter speed that maximized the entropy of the histogram. This approach requires capturing multiple exposures to find the maximum entropy. Follow-up works attempt to reduce the search time [20, 21] and settle for local maximums in entropy. The gradient-based methods are introduced in [22, 23], which claim a higher gradient approves more details captured.

Semantic AE methods explicitly consider scene content by weighting different image regions related to their semantics. Many consumer cameras emphasize scenes with faces [24]. Yang et al. [3] apply reinforcement learning to create personalized semantic AE based on user preference. Onzon et al. [4] recently proposed a method to train semantic AE jointly with the task of object recognition. This method is ideal for machine vision but needs to be optimized for perceptual quality. Many methods in the computer vision literature attempt to improve a poorly exposed image through post-processing to improve the perceptual quality of a capture image [25–29]. However, these methods do not directly improve the basic AE algorithm used to capture the image in the first place.

AE Datasets. Several existing datasets have been collected for exposure-related problems. Most of these datasets contain only a single static scene with varying exposure, often with the goal of post-processing exposure correction [30–32]. Similarly, datasets have been captured with varying exposure with the goal of constructing a fused HDR image [33–36]. Recent work by [37] provided a dataset with intensive lighting for evaluating object detection and classification. While these can be used to evaluate AE on a static scene, none of these datasets include a temporal dimension.

Video datasets with multiple exposures to examine video-HDR methods have been captured [35, 38–40]. These existing datasets, however, often have limited exposure sampling (often only two exposures) and do not include scenes with harsh lighting or abrupt changes in lighting. Moreover, AE algorithms are implemented in the low-level camera hardware and are therefore applied directly to RAW sensor data. As a result, existing static and video-based datasets are unsuitable for evaluating AE algorithms.

As discussed in Chapter 1, the lack of sufficient datasets is the impetus of our work.

The following chapters describe our dataset collection and AE evaluation platform. Representative content-agnostic and semantic AE algorithms are evaluated using our system. Our dataset and platform even allow us to propose a simple semantic algorithm using fast saliency detection that shows promising performance.

Chapter 3

4D Temporal Exposure Stack Dataset

A detailed description of the 4D temporal exposure stack dataset, including the capturing process, the reprocessing, and the final format of the dataset, is illustrated in this chapter.

3.1 Capture Setup

We captured a temporal exposure dataset with four dimensions: $time \times exposure \times height \times width$. Our capture environment was adopted from the stop-motion setup proposed by Abuolaim et al. [41] for 4D auto-focus capture (see Figure 3.1 for an image of our setup). All scenes were captured in a dark room with two controllable 90000-lumen key lights and a softbox-diffused light source. The lights use DC power to prevent the flickering effect caused by alternating currents. The softbox light was used as a constant illumination, while the key lights provided intense lighting (high dynamic range) and could be turned on and off. Two linear stage actuators allow us to move objects and key lights. An Arduino/Genuino microcontroller synchronized scene motion, lights, and camera cap-

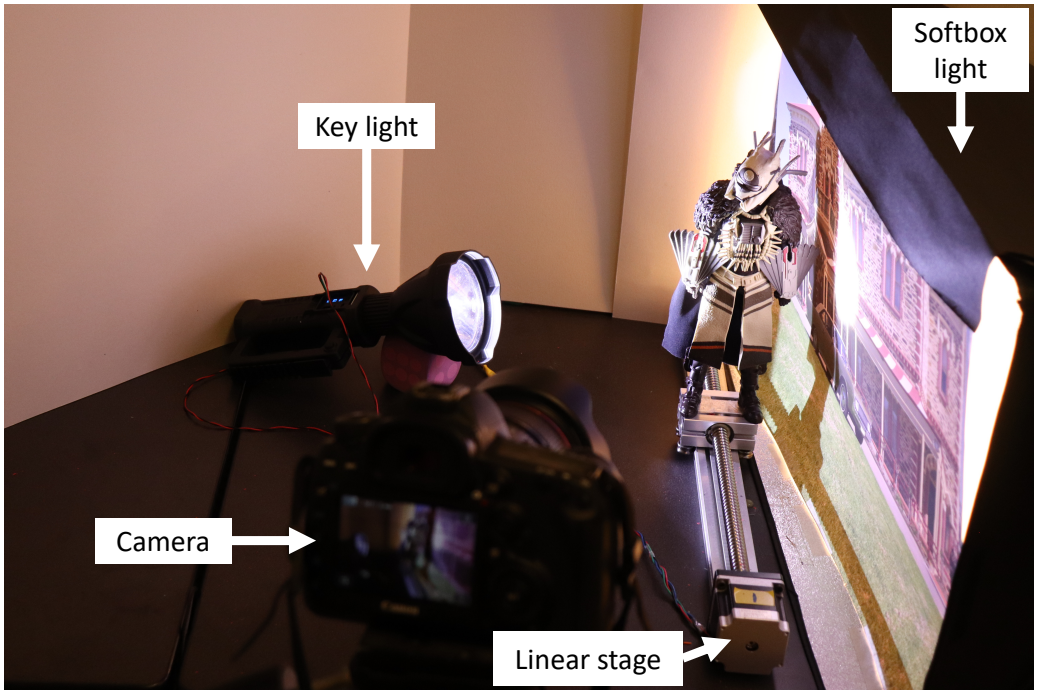

Figure 3.1: Our dataset capture setup with controllable lights, camera, and motion stage.

ture. Images were captured with a Canon EOS 5D Mark IV in RAW format (6744×4502). Since we are interested in aperture-priority AE, we fixed the camera aperture (f/14) and ISO (100).

Our setup enabled us to capture scenes with dynamic lighting but have access to a full exposure stack at each time step. Each scene contained 100 simulated time steps $t_0 - t_{99}$. For each time step t, we captured 15 exposure stack images $I_t^0 - I_t^{14}$. The shutter speed ranged from $\frac{1}{500}$ and 15 seconds—that is 12.87 EV of steps. We added interesting scene dynamics by turning on key lights for certain time steps of our scenes. Scene 9 is shown in Figure 3.2 as an example. In this scene, the key light is on between $t_{20} - t_{39}$ and $t_{60} - t_{79}$, simulating sudden change in lighting.

3.2 Dataset Format

Our initial AE evaluation found that 15 exposure steps did not provide sufficient granularity for smooth exposure changes. Instead, we found that 40 exposure levels evenly sampled between the shortest and longest exposure in provided a better emulation of exposure adjustment in a real AE system. To expand the initial 15 exposures to 40, specifically, any exposure not already part of the original 15 exposures is interpolated based on its two nearest neighboring exposure images in the original sequence. The interpolation procedure assumes a linear relationship between the exposure time and the image pixel value [18]. To visualize the RAW images, we also process the RAW images to have a corresponding 4D sRGB dataset (also $100 \times 40 \times height \times width \times 3$) using the camera pipeline provided in [42].

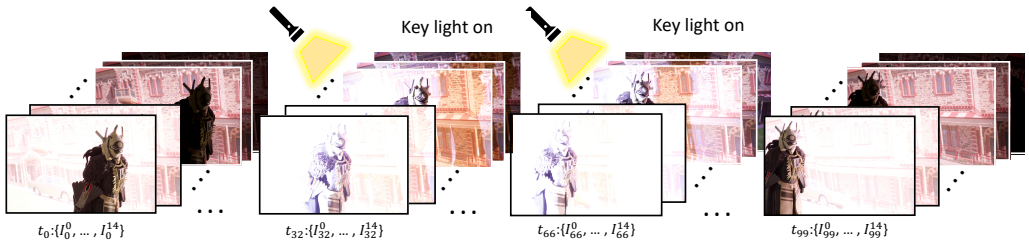

Figure 3.2: 4D Exposure Dataset (Scene 9). The four dimensions refer to (1) time, (2) exposure, and (3&4) the 2D coordinates of the RAW sensor image associated with each (1) and (2). Each scene is characterized by a combination of challenging lighting conditions and objects. In this scene, a flashing light source is simulated by turning on the key light between t_{20}-t_{39} and t_{60}-t_{79}.

Scene ID	1	2	3	4	5	6	7	8	9
Example image									
Back light	✗			✓	✗				✓
Moving light	✗	✓			✗		✓		✗
Flashing light	✗		✓				✗	✓	
Reflective objects	✓			✗		✓		✗	
Preferred objects	✗			✓					

Table 3.1: The nine scenes (image sequences) in our AE dataset.

Figures 3.3-3.8 show examples of different scenes from our dataset; for each scene, we split the exposure stack across two figures.

3.3 Summary

In total, we captured nine scenes intended to provide a range of challenging conditions for AE algorithms. Each scene had challenging lighting conditions: backlight, moving light, and/or sudden changes in lighting. In addition, some scenes contained reflective objects (i.e., a mirror) or preferred objects (i.e., faces). Table 3.1 shows each scene and the corresponding lighting conditions and objects. The following chapters introduce the platform and our experiments evaluating AE methods on this dataset.

Figure 3.3: Scene 1 exposure stack (15s - $\frac{1}{2}$s) at 6 different time steps.

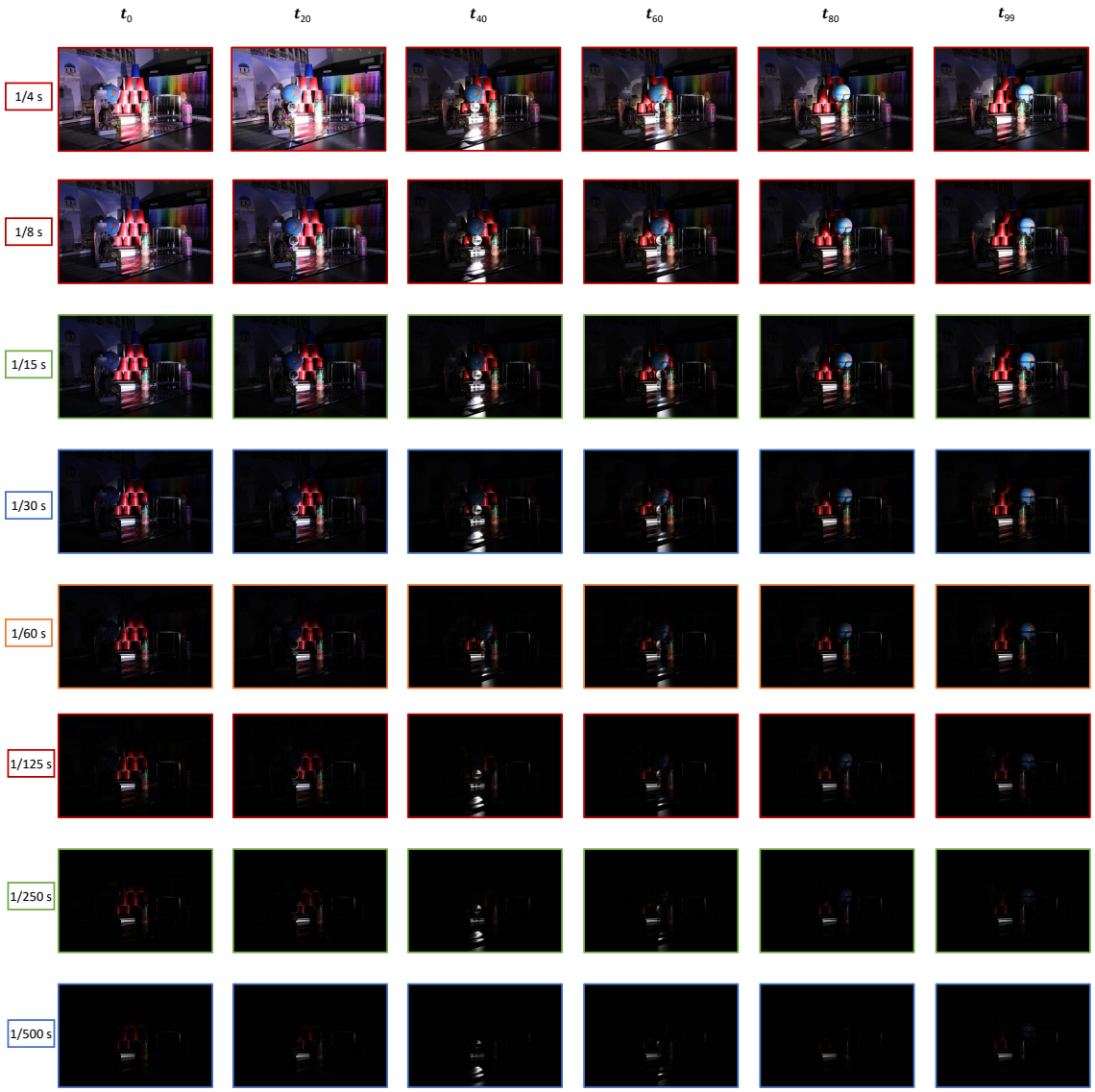

Figure 3.4: Scene 1 exposure stack ($\frac{1}{4}$s - $\frac{1}{500}$s) at 6 different time steps.

Figure 3.5: Scene 6 exposure stack (15s - $\frac{1}{2}$s) at 6 different time steps.

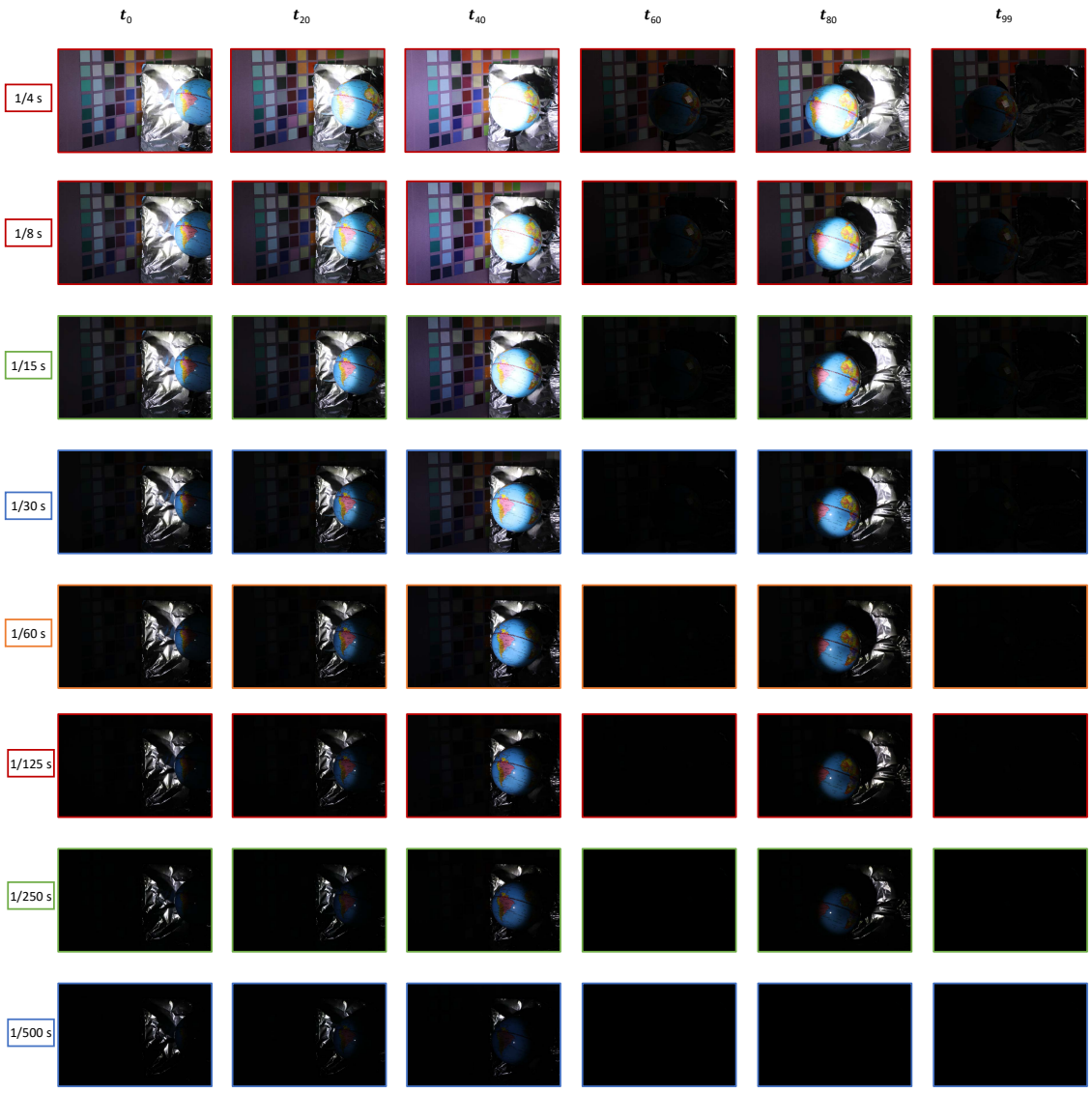

Figure 3.6: Scene 6 exposure stack ($\frac{1}{4}$s - $\frac{1}{500}$s) at 6 different time steps.

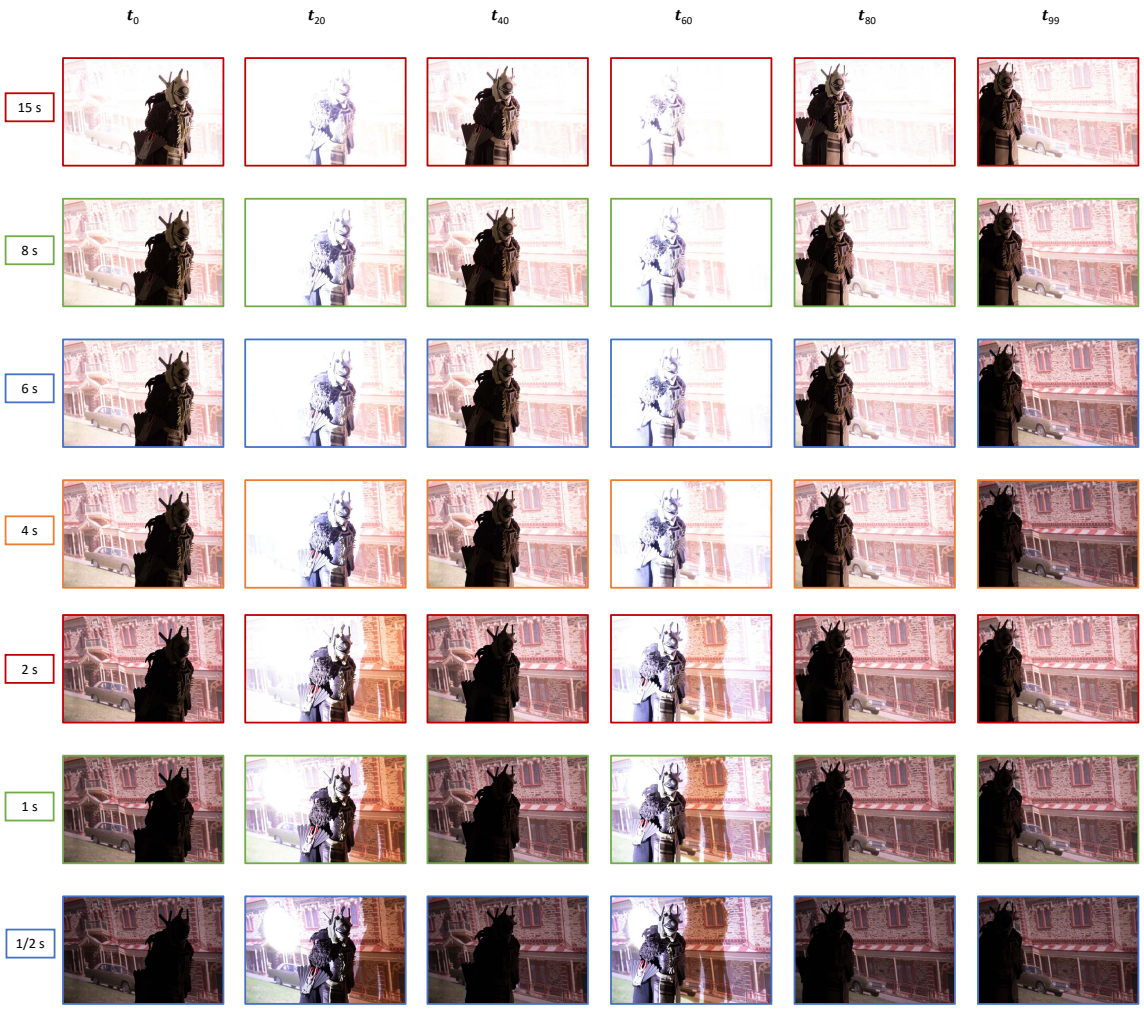

Figure 3.7: Scene 9 exposure stack (15s - $\frac{1}{2}$s) at 6 different time steps.

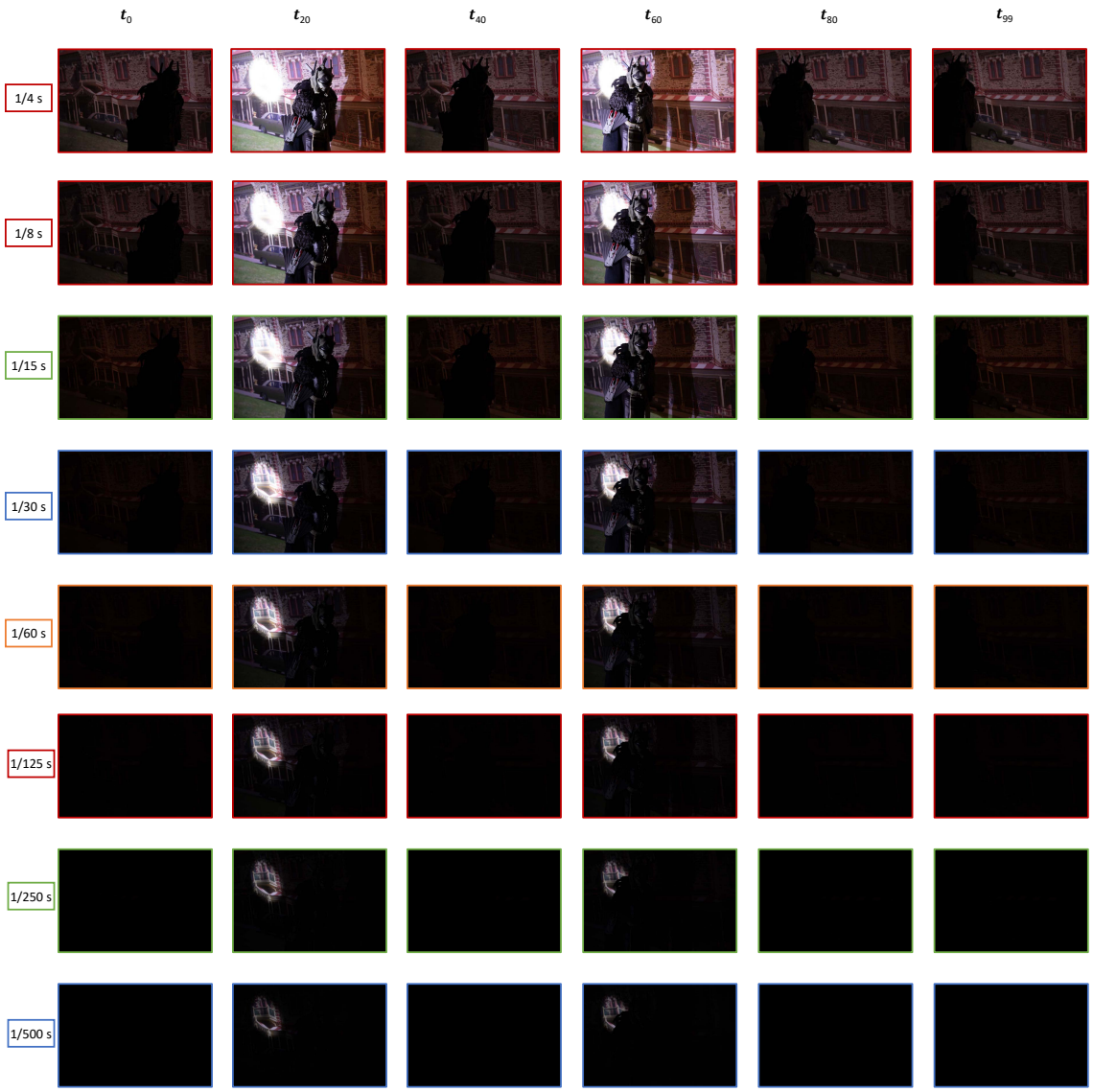

Figure 3.8: Scene 9 exposure stack ($\frac{1}{4}$s - $\frac{1}{500}$s) at 6 different time steps.

Chapter 4

Platform GUI

In this chapter, we describe the functions and features of the platform GUI that processed the dataset introduced in Chapter 3 to evaluate AE algorithms and visualize the results.

4.1 Introduction

A Python-based AE evaluation platform is developed to work with our dataset. Figure 4.1 shows the platform's interface and the basic workflow of testing an AE algorithm. On the top right of the window, two drop-down menus are provided for the scene and algorithm selection. Users can adjust the parameters for the AE algorithms described in Chapter 5. The user can also select the starting shutter speed as the input to the AE algorithm. After the parameters for an AE algorithm are set, the AE algorithm can be applied, where it will select a single exposure per time-step that can be played back in our GUI or saved to a video.

Our platform also allows a visual dataset exploration and an AE solution. Sliders allow

navigation of the exposure and temporal dimensions. For example, the upper-half image in Figure 4.1 displays the 16th image (shutter speed 2/5 s) of the 36th frame in scene 1 of our dataset. This image could either be selected manually by adjusting the sliders or have been selected by an AE algorithm as the optimal frame for this time step. The second part of the visualization shows three additional plots. These plots are displayed to help the user understand the AE algorithm's performance. In particular, the user can see a different combination of plots when operating the interface. An example is shown in Figure 4.1. After taking the user's input in "Step 3," and running the "semantic" algorithm described in Section 5.1, the GUI displays the plots, which allows the user to view the RAW image histogram, the processed image(sRGB) histogram and the current frame stack mean values.

The user may pause the algorithm at any frame and adjust the vertical slider (to visualize an image with another EV from the same frame); then, the plots will switch to the mode as in the bottom part of Figure 4.1. In this case, the user sees the difference between the current image and the one selected by the AE algorithm in terms of the RAW histogram and mean.

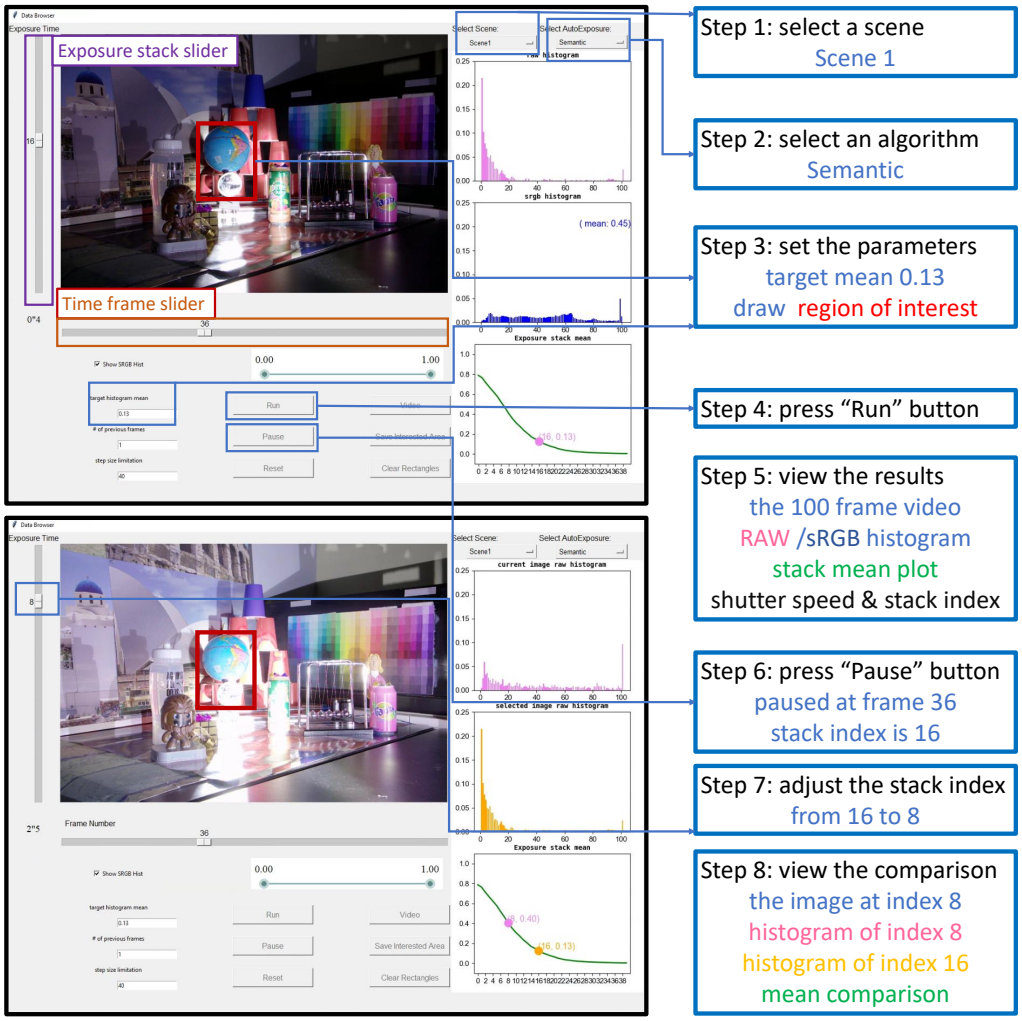

Figure 4.1: The basic steps for using our AE platform. The user selects a scene and an algorithm. The parameters of the AE algorithm can be adjusted. After the AE algorithm runs, the platform plays the output images and the corresponding plots at 10 FPS. The user causes “pause” at any time frame to adjust the exposure stack slider for comparison. Image histograms for each frame are also shown.

4.2 Summary

We produced the 4D temporal dataset and then designed a platform GUI that examines and visualizes AE algorithms on our dataset or similar datasets. The next chapter discusses our experiment of evaluating AE methods and designing a new AE algorithm on the 4D dataset by the platform.

Chapter 5

AE Algorithms

Up to this point, we implemented six AE algorithms to apply the algorithm evaluation on our dataset and platform; two content-agnostic and two semantic AE algorithms out of the seven were selected for the user study (details in Chapter 6). The first AE was a content-agnostic global algorithm driving the mean to a target value [12]. The second content-agnostic AE focused on entropy maximization [19]. Next, we implemented a semantic AE that uses manually drawn bounding boxes to give preference to specific regions. Finally, we developed our own semantic AE guided by a fast saliency method. We discuss the concept and the settings of these four AE methods in this chapter, and brief descriptions of the other two are also provided. They are an alternative version of the *global* algorithm which partitions the image into grids and defines the outliers based on the mean of grids, and a gradient approach [23] which prefers the image with maximum gradient.

Most AE algorithms (with the exception of the entropy methods and the gradient approaches) can be modeled with two steps shown in Figure 5.1.

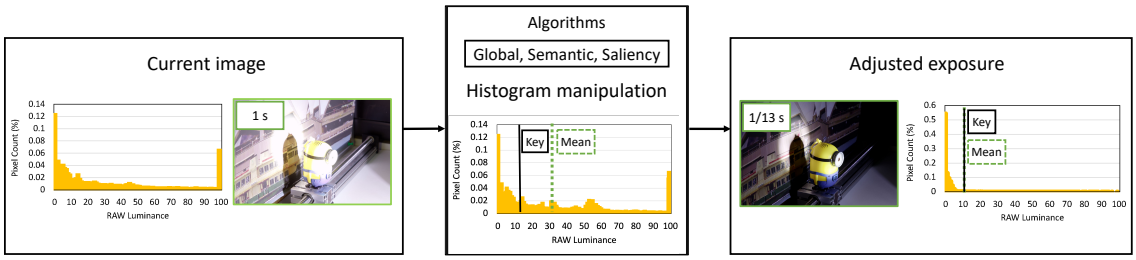

Figure 5.1: The basic two steps for most AE algorithms. First, each AE algorithm uses different criteria to decide how to construct a histogram from the current image frame. Second, the mean of the histogram is compared to a user-specified key. The exposure is adjusted either longer or slower so the next frame will better match the key.

1. Histogram manipulation: each AE algorithm determines which pixels in the image contribute to the image histogram.

2. Exposure modification: the shutter speed is adjusted such that the histogram's mean is shifted towards a user-defined target (referred to as the 'key').

Each of these steps is discussed in more detail in the following.

5.1 Histogram manipulation

A histogram H can be represented by a vector of RAW pixel values $H_p = [p_1, p_2, ..., p_n]$ and corresponding weights $H_w = [w_1, w_2, ..., w_m]$. Each corresponding weight can be considered how important a pixel is to the AE algorithm used. Typically, AE histogram manipulation consists of changing the histogram by modifying H_w.

We look closely at three types of AE algorithms that perform histogram manipulation. The *global* algorithm uses the full image histogram. The *semantic* algorithm only uses the

histogram within a specified area. We introduce our own method: a *saliency* AE algorithm that uses a saliency map to build the histogram weights. An example of each algorithm in our user study is shown in Figure 5.3.

The *global* method weights all pixels equally. Our *semantic* method is similar to the *global* method but uses only pixels within a defined bounding box to construct the histogram. In our work, we have manually drawn this bounding box to emulate different detectors, such as a face detector or object tracking [24]. Finally, we introduce a simple *saliency* method, where we first run a fast saliency [43] detector on the image. The saliency map is thresholded and pixels above the threshold contribute more to the final histogram. If no pixel is labeled salient, the saliency method reduces to the global algorithm where all pixels have equal importance.

After using the algorithm-specific weighting function, all algorithms implemented a histogram clipping of saturated pixels. In our user study, this was done by zeroing the weights of all pixels with intensity greater than 0.9. A user may customize the settings as needed with our platform.

For the alternative *global* algorithm, an input RAW image is broken up into non-overlapping grid patches. When a grid mean falls into the outlier range, a certain percent (user-defined) of pixels randomly selected in such grid patch will be weighted as 0 to reduce the impact from this over-exposed or under-exposed area. Figure 5.2 illustrates a comparison between the grid-based *global* approach and the *semantic* method.

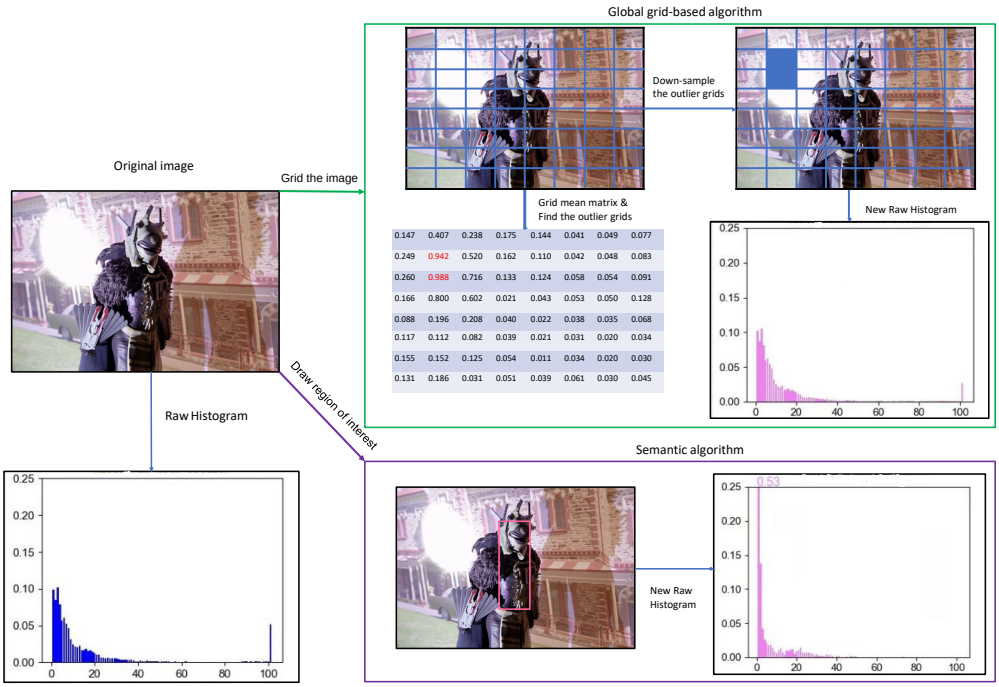

Figure 5.2: An illustrative example of the "global grid-based" and "semantic" algorithms. The global grid-based algorithm divides the image into an 8×8 grids. Any grid region with a mean value (intensities normalized between [0,1]) that is greater than 0.9, is considered as an outlier grid. As a result, only 20% of the pixel values in this grid will be added to the histogram (i.e., the grid is down-weighted in terms of histogram contribution). The semantic AE algorithm builds a histogram using only the pixels within the boundary of interest.

5.2 Exposure modification

The mean of a histogram H is given by a weighted average where H_w is normalized so all weights add to 1. The camera-specific key is typically a RAW value that maps to half-brightness (0.5) in sRGB after being passed through the camera-specific image signal

processor (ISP). The goal of exposure modification is to bring the mean of a histogram to the camera-specific key. Since exposure has a linear relationship to the RAW image, the exposure modification can be calculated as a scale between the key and the current mean of the histogram. In our implementation, each algorithm's modified histogram is driven to the key by adjusting the shutter speed up or down.

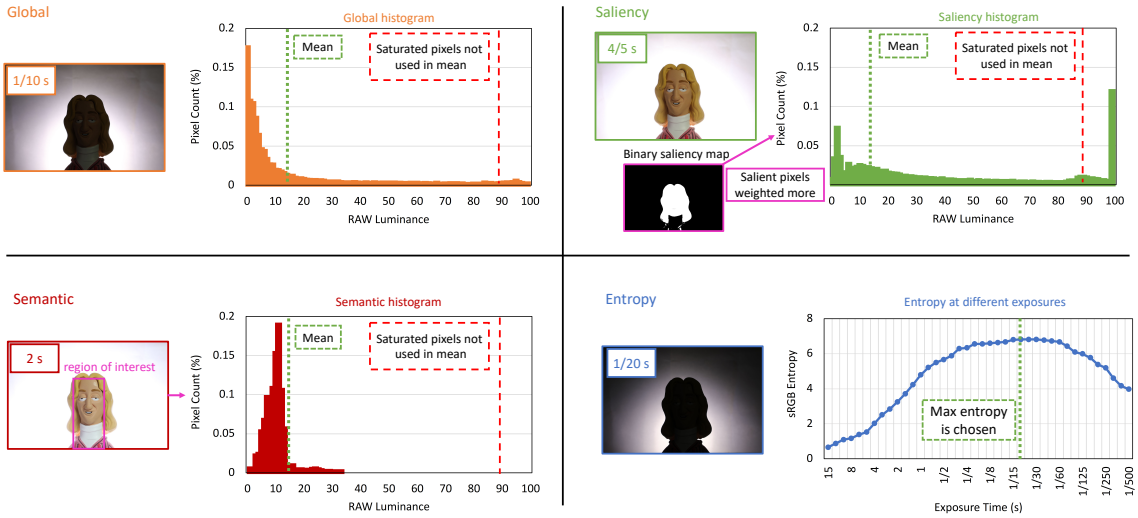

Figure 5.3: An overview of the four AE algorithms used in this work. A global AE (top left) that uses the whole image to build the histogram. A saliency AE (top right) that weights salient pixels with higher weight. A semantic AE (bottom left) that uses the bounding box we provide to build the histogram. Finally, an entropy AE (bottom right) that chooses the exposure with maximal entropy. Here, we show the images chosen by all of these algorithms for scene 4 at time step 59.

5.3 Entropy AE

We also implemented the *entropy* AE by Zhang et al. [19]. Since this method operates on a post-processed image, we perform our entropy calculation in the post-processed sRGB space. For each time step, we compute the exposure that maximizes the entropy across the exposure stack. An example of this entropy maximization is shown in Figure 5.3. At each time step the maximum entropy results in an "optimal" sequence of exposure indices. This is possible because our dataset gives us access to all exposures for any time step. Additionally, with our implementation, we are advantaging this method because normally a local search would be required to find the maximal entropy image.

5.4 Gradient AE

A gradient AE method adjusts camera exposure using the gradient information. We implemented a gradient AE based on the concept introduced by Shim in [23], which adjusts camera exposures to maximize the gradient information of the captured images. We excluded it from the user study because the output was less satisfactory than the other algorithms. A possible reason for the poor performance is that we used the default value for the hyper-parameters without tuning them. Future work could be done to test the gradient methods with different hyper-parameters.

5.5 Settings

Here we discuss the settings for these algorithms we have chosen for the user study. In particular, we first provide details to the *global*, *semantic*, and *saliency* AE algorithms. These methods all use the same AE mechanism of adjusting the shutter speed based on histogram means and its relationship to a target value (key). These methods differ in how they construct (or manipulate) the image histogram. This is followed by a discussion on the *entropy* AE method that uses a different strategy to decide on the optimal AE value.

Histogram Manipulation AE Recall that a histogram H is represented by a combination of its pixel values H_p and corresponding weights H_w.

- *Global* AE gives all pixels within the image an equal weight.
- *Semantic* AE uses the hand-drawn bounding boxes provided for each frame within a scene. All pixels within the bounding boxes are given a weight of 1; pixels outside the bounding boxes are given a weight of 0.
- *Saliency* AE uses the last time step's saliency map to generate weights for the current time step's histogram. To generate the salient map, we use the fast saliency [43] detector on the sRGB image produced in the last time step. This produces a saliency map where each pixel has an associated value between 0 and 1. Then, we build a binary saliency map where any pixels above γ are considered salient; γ can be interpreted as the "sensitivity" of the saliency detector. Salient pixels were given a weight of β, and the remaining pixels were given a weight of 1. For our implementation, we set $\gamma = 0.1$ and $\beta = 14$. Figure 5.4 contains two examples of the

binary saliency maps we generated. Additionally, for the first time step in a scene, we consider no pixels to be salient.

After using the algorithm-specific weighting function, all algorithms implemented a histogram clipping of saturated pixels. This was done by zeroing the weights of all pixels with an intensity greater than 90% of the maximum intensity value in the image. Figures 5.5-5.7 visualize three continuous stacks of the weighted binary saliency maps with the outliers weighted as 0. It involves an abrupt change of lighting. As shown in Figure 5.6, a flashlight was turned on at time step 70. By using the saliency map of time step 69, which stores the possible interested area (i.e., the globe), it is able to find the optimal exposure value given in the red-bordered image (EV26), which shows a large jump from EV5 (the optimal image of time step 69 as shown in Figure 5.5).

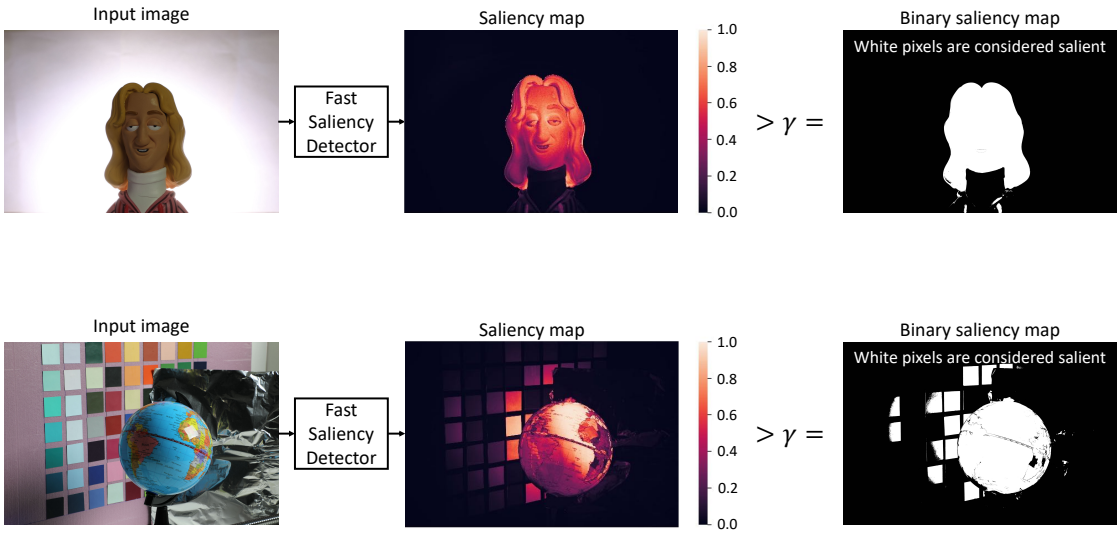

Figure 5.4: Example binary saliency maps generated from two images in our dataset. Each binary saliency map is generated by a fast saliency method [43] thresholded with $\gamma = 0.1$.

Figure 5.5: Stack of binary saliency maps-scene 6 time 68. The saliency map generated from the optimal image of time 68 is shown as hot map. The right side of the image shows the weighted (based on the outliers of time 69 images) binary saliency map paired with the time 69 images. The pair in red boundary is the optimal image chose by the saliency AE method of time 69 by applying the binary saliency map.

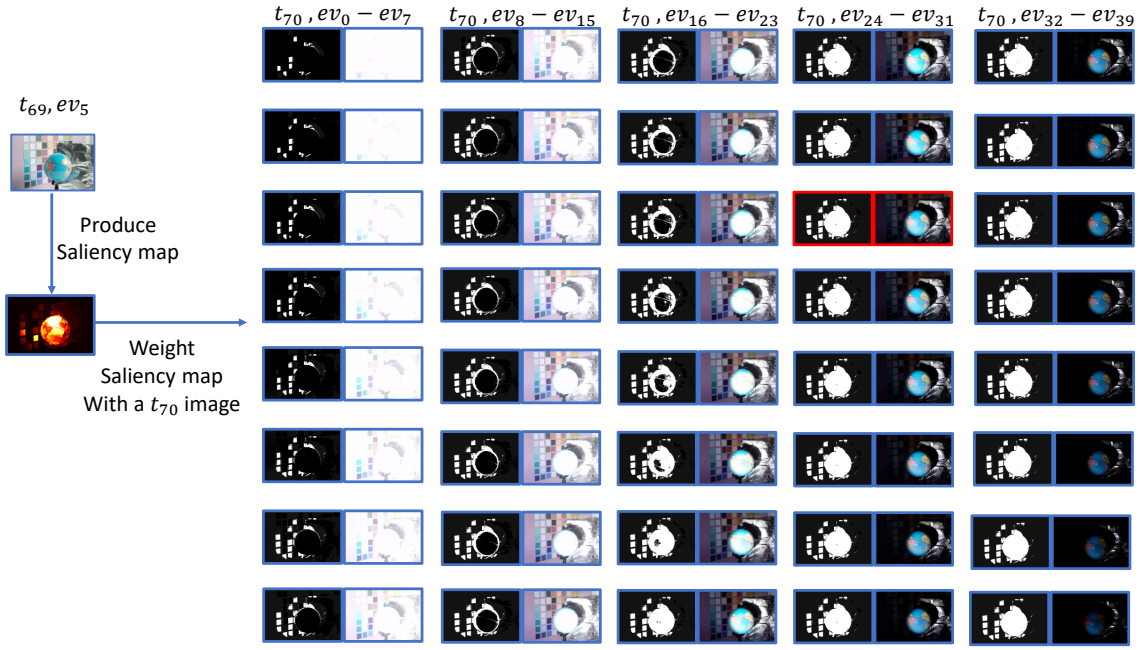

Figure 5.6: Stack of binary saliency maps-scene 6 time 69. The saliency map generated from the optimal image of time 69 is shown as hot map. The right side of the image shows the weighted (based on the outliers of time 70 images) binary saliency map paired with the time 70 images. The pair in red boundary is the optimal image chose by the saliency AE method of time 70 by applying the binary saliency map.

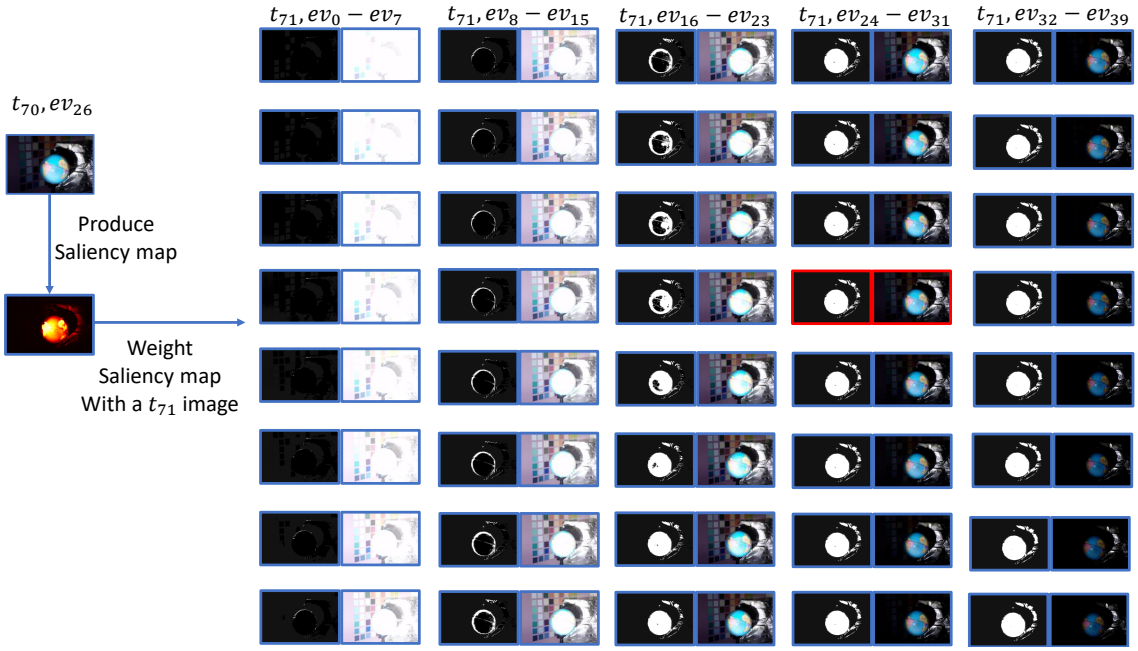

Figure 5.7: Stack of binary saliency maps-scene 6 time 70. The saliency map generated from the optimal image of time 70 is shown as hot map. The right side of the image shows the weighted (based on the outliers of time 71 images) binary saliency map paired with the time 71 images. The pair in red boundary is the optimal image chose by the saliency AE method of time 71 by applying the binary saliency map.

Entropy AE The Entropy AE algorithm attempts to measure the amount of information present in the image (i.e., the image's entropy). Entropy is high when there is a wide range of pixel values, indicating a wide range of variation in brightness levels. Conversely, entropy is low when the pixel values are concentrated in a narrow range. Unlike the other methods, this method targets the entropy of the sRGB (processed) images. Figure 5.8 shows an example where the Shannon entropy function has been applied to the full stack of sRGB images. The image with maximum entropy is considered optimally exposed. Note that our implementation of entropy AE is *advantaged* because a typical implementation

needs to perform a local search to find the maximum entropy image [19–21]. However, we search over the full AE stack of sRGB mages in our implementation. This means our entropy method is more accurate as it avoids local optima. We note that this method could not be designed in practice since having the full stack of sRGB at each time step would not be possible. However, we include this method to gauge its performance.

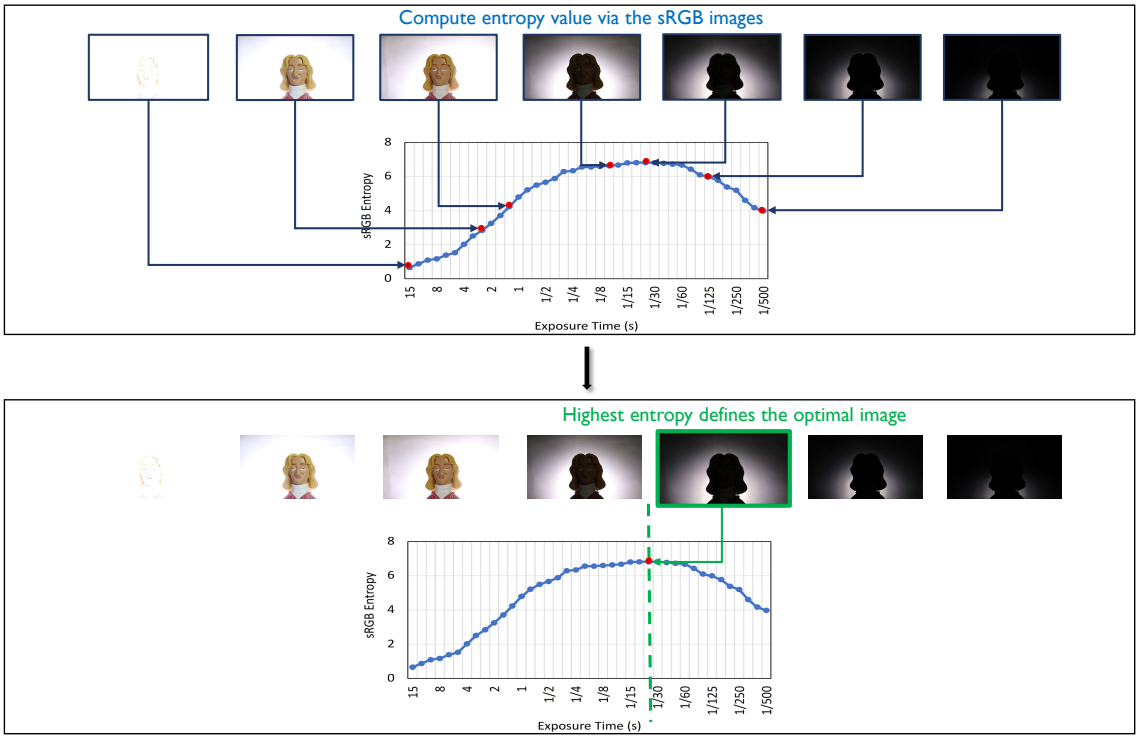

Figure 5.8: Example entropy-based AE algorithm applies on a time frame in scene 4. The Shannon entropy function is applied to the full stack of sRGB images, and the image with maximum entropy is considered optimal.

5.6 Summary

This chapter discussed the main concepts and procedures of the AE algorithms we evaluated with our dataset and platform. The results of our experiments are shown in the following chapter.

Chapter 6

Experiments

As mentioned in Chapter 4, one benefit of our dataset and platform is the ability to evaluate different AE algorithms on the same input and in a repeatable manner. Since each algorithm has different criteria about which image pixels or pixel intensities should contribute to the determination of the exposure, we performed a user study to see if users have a preference. In addition, we ran the methods on full-size, 1/8, and 1/40 scaled input images. We saw only minor changes to the output of the different AE algorithms among the different image sizes, even at a 1/40 scale as described in Section 6.3.

6.1 User study

In challenging lighting conditions in a temporal sequence, the output of different AE is subjective. To better understand what users' prefer in such conditions, we conducted a forced-choice user study to compare four AE algorithms on the nine scenes from our dataset. We evaluated the algorithms by compiling overall user preference scores for each

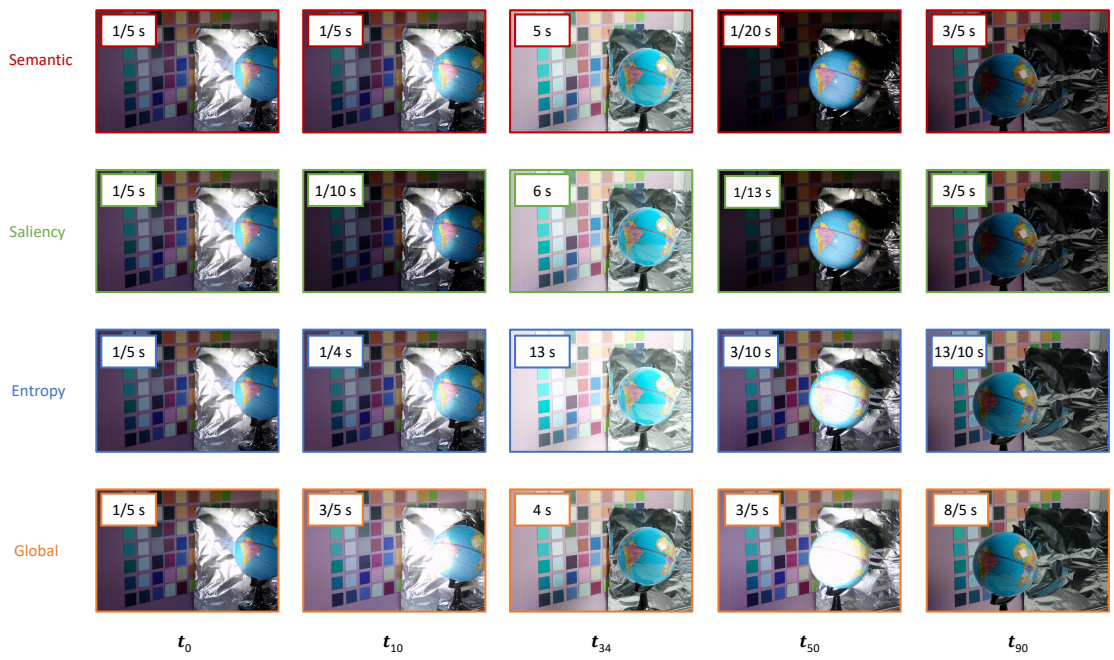

Figure 6.1: The frames chosen by different AE algorithms for scene 7 at five different time steps. Initially, all AE algorithms started at the same exposure but then quickly diverged.

algorithm. The user study falls under ethics approval granted for image/video comparisons. Proper procedures regarding disclosure and anonymity were followed.

To perform our study, we generate simulated videos for all nine scenes in our dataset with four AE algorithms (*global*, *semantic*, *saliency*, and *entropy*). Videos using semantic AE used the bounding boxes provided with the dataset for all frames. For saliency AE, the first-time step assumes no salient pixels. All other time steps used the saliency map generated from the previous frame's post-processed sRGB image. Figure 6.1 shows an example output for the different methods on a particular scene.

For each time frame, we calculate the optimal index that brings the algorithm's modified histogram mean closest to the key. Then, we smooth the indices using a history of

four-time steps to have smooth transitions as the exposure changes; this was inspired by Aboulaim et al. [41], who showed users had a strong preference for smoothness when viewing videos.

Our study involved 33 people. The age of participants ranged between 22 and 39 (mean = 24.4, standard deviation = 3.3). The cohort included 22 males and 11 females. Participants performed the experiments in a dark room using the same 15" MacBook Pro.

Participants were asked to sit centered in front of the computer screen and read the instructions. Then, participants did 54 forced-choice comparison trials; this took 10-15 minutes per participant. In addition, the trial order was randomized for each participant.

For each trial, a participant would view two videos rendered with different AE algorithms synchronously. Videos were 10 seconds long (10 FPS) and looped until the user selected a preferred video—that is, forced choice. Participants selected their preferred video by using the left and right arrows on the keyboard. The option to select a video was enabled only after watching the video fully once.

This study utilized a 4×9 within-subjects design with the following independent variables and levels: AE algorithm:*semantic*, *saliency*, *entropy*, *global*; Scene: 1,2, ..., 9.

The dependent variable measured was user preference. The average number of votes for an AE algorithm is between 0 and 3 because each algorithm is only compared in 3 trials per scene. User preference normalizes the average number of votes by 3, so the metric is between 0 and 1.

Each scene requires six trials to have all pairings between the four AE algorithms. Thus, the total number of trials was 33 participants $\times$ 9 scenes $\times$ 6 trials = 1782.

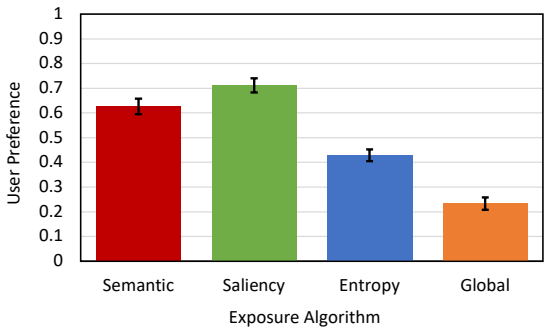

Figure 6.2: The average preference of AE algorithms across all scenes. Error bars represent 95% confidence interval bars.

6.2 User study results

The average preference of all algorithms across the scenes was 0.71 for *saliency*, 0.63 for *semantic*, 0.43 for *entropy*, and 0.23 for *global*. Figure 6.2 shows a comparison between the methods with 95% confidence interval bars.

An ANOVA was also conducted and it showed that the effect of the algorithm on preference was statistically significant ($F_{3,128} = 203.1, p < .00001$). Additionally, the post hoc Fisher LSD and Bonferonni-Donn comparison tests resulted in a significant difference between all pairs of algorithms.

In addition, Figure 6.3 shows a breakdown of the preference of algorithms per scene. The saliency AE performed well across all scenes except scene 9. Most of the time, we found that users were willing to have over/under-exposed backgrounds if it meant a foreground object (often moving) was well exposed. However, for scene 9, users preferred a result where the foreground object (a figurine) was dimly lit so they could see the background; thus, saliency and semantic AE performed poorly.

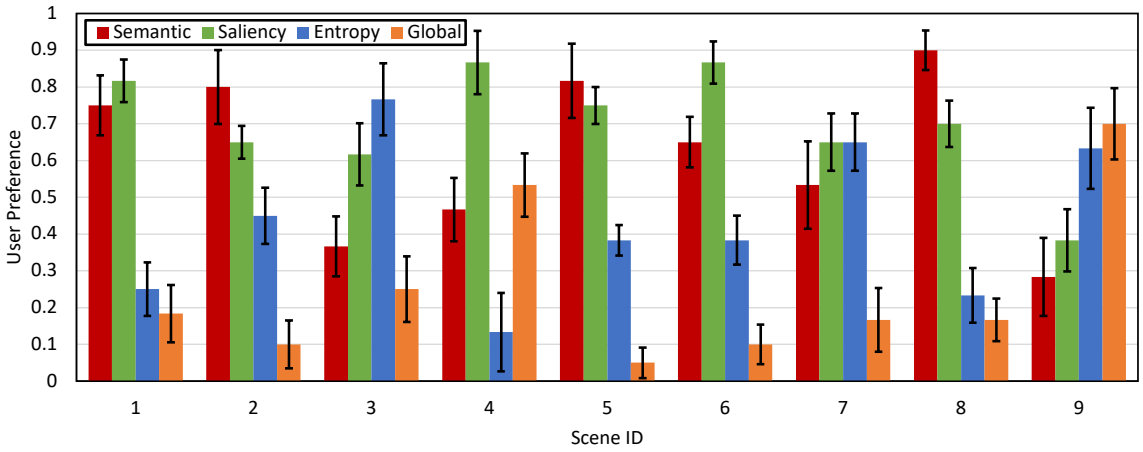

Figure 6.3: The average preference of AE algorithms shown per scene. Error bars represent 95% confidence interval bars.

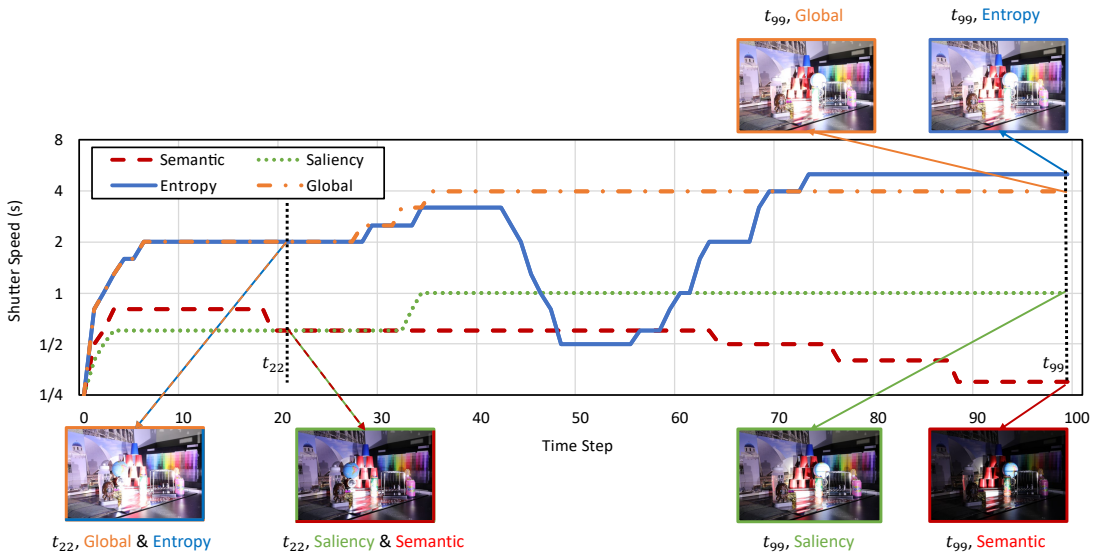

Figure 6.4: Scene 1 time plot visualizing how AE algorithms choose different exposures at various time steps. This scene contains a moving mirror (high reflective object). Additionally, we visualize the images chosen by the AE algorithms at time steps 22 and 99.

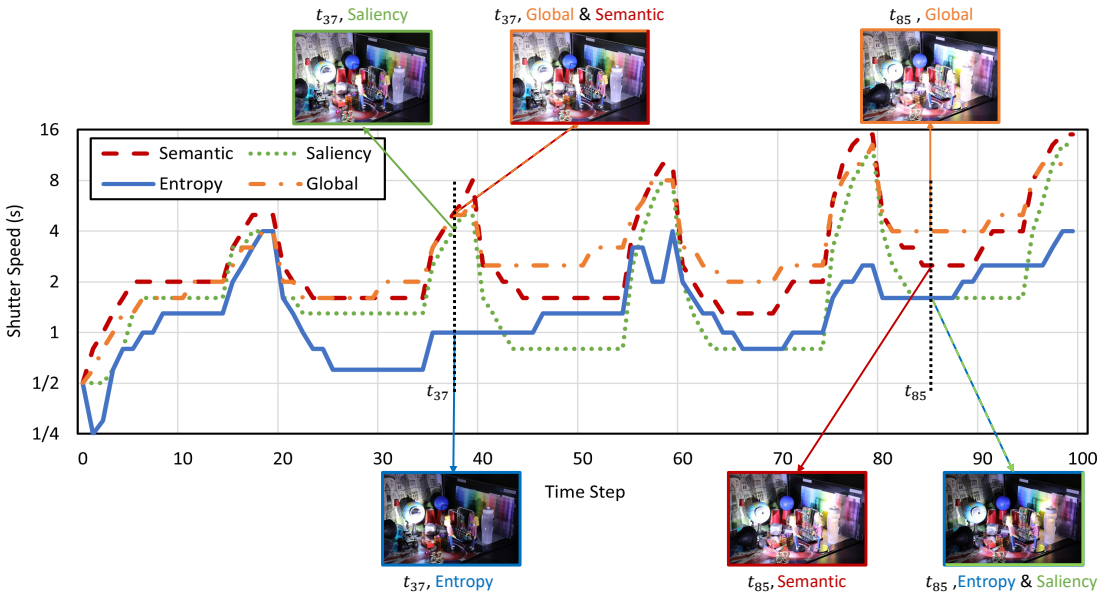

Figure 6.5: Scene 3 time plot visualizing how AE algorithms choose different exposures at various time steps. This scene contains a moving, flashing light and a mirror (high reflective object). Additionally, we visualize the images chosen by the AE algorithms at time steps 37 and 85.

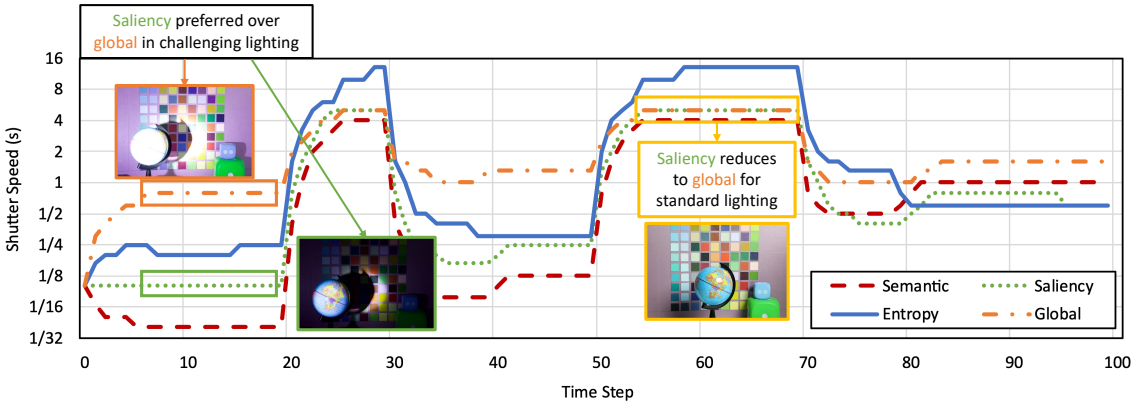

Figure 6.6: Scene 5 time plot visualizing how AE algorithms choose different exposures at various time step. In challenging lighting conditions, saliency AE chooses an exposure that prioritizes the object at the expense of the background. However, in standard lighting, saliency and global AE choose the same exposure.

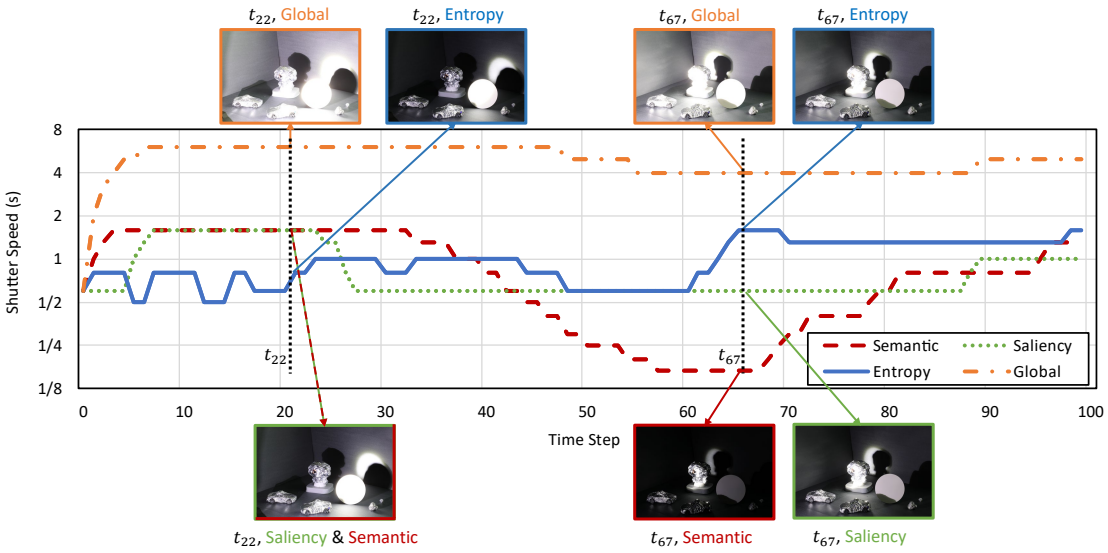

Figure 6.7: Scene 7 time plot visualizing how AE algorithms choose different exposures at various time steps. This scene contains a moving high-lumen light and several highly reflective objects (doll, ball, and cars). Additionally, we visualize the images chosen by the AE algorithms at time steps 22 and 67.

There was a clear preference for the saliency and semantic algorithms because they prioritized the objects participants were more interested in viewing. Objects of interest typically are moving, familiar (e.g., face), or centered in the frame. To better understand why saliency AE had higher user preference than other methods, we generated a plot of exposures selected by different AE algorithms on different scenes. We noticed that the global method and saliency method chose similar exposures at time steps with less challenging lighting. However, for time steps with challenging conditions (e.g. bright globe), saliency AE chooses an exposure with more poorly exposed pixels that keeps the preferred object at a proper exposure. Figures 6.4-6.7 show examples of this phenomenon in 4 scenes.

Even though our implementation of *entropy* AE was advantaged, it could have performed better. This might indicate entropy is not a good metric to use for AE algorithms designed for human viewing.

6.3 Scale the input image size

Our platform allows the image to be resized. Most camera AE systems use subsampled RAW images. We found that reducing the image size to 168×112 resulted in an average EV change of 0.08 across all frames, scenes, and AE algorithms (i.e., minimal impact). See the experiment below with AE algorithms run at different sizes. Figure 6.8 gives a comparison of evaluating the *Global* and *Saliency* method on two scales of scene 6. As shown in the figure, miner differences are presented between the three scales for both methods. We computed the errors with downsized (1/8 and 1/40) input images compared

to the full-size input images over the 9 scenes and 4 methods (*semantic*, *saliency*, *entropy*, *global*), and displayed them in Table 6.1. Where "EV Error" stands for the absolute error of the exposure value between the scaled and the full-size input image, and "EV Distance" indicates the absolute value of the indices difference in the 40 EV stack between the scaled and the full-size input image. We conclude that the output errors with downsized input images are not considerable. As a result, we found it possible to perform AE algorithms on the downsized images (down to 168×112 in our experiment) to reduce memory overhead and improve computational performance.

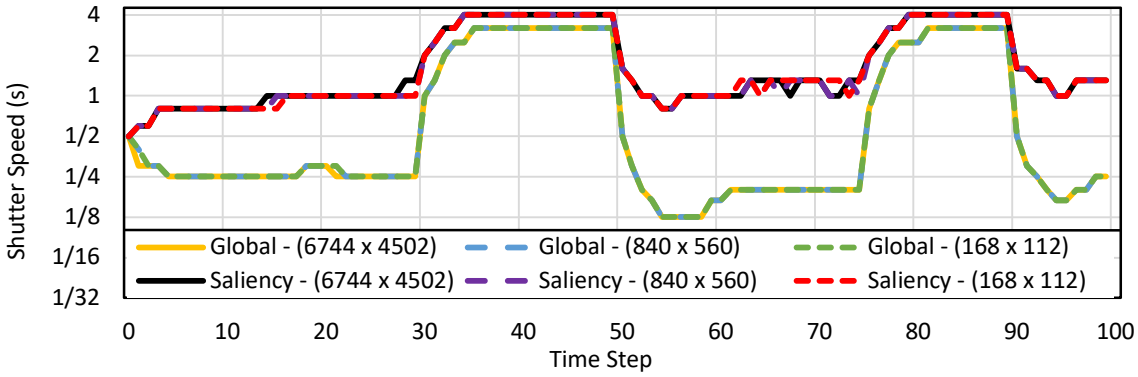

Figure 6.8: Global and saliency AE at different scales on scene 6.

6.4 Summary

This chapter presented the result of our experiments, including a user study to gauge preference for the different AE algorithms and the impact of input scale sizes.

The AE algorithms' output errors of 1/8 and 1/40 comparing to full size images									
Scene	Method	Global		Saliency		Semantic		Entropy	
	Scale / Diff	1/8	1/40	1/8	1/40	1/8	1/40	1/8	1/40
1	Avg EV Error	0	0.0032	0.1940	0.1472	0.0041	0.0441	0.2225	0.0779
	Avg EV Distance	0	0.01	0.53	0.45	0.01	0.13	0.69	0.24
2	Avg EV Error	0	0.0064	0.0613	0.1048	0302	0.0719	0.2582	0.2325
	Avg EV Distance	0	0.02	0.19	0.32	0.08	0.21	0.82	0.66
3	Avg EV Error	0	0.0191	0.0317	0.0351	0.0098	0.0270	0.2515	0.1721
	Avg EV Distance	0	0.06	0.1	0.1	0.03	0.08	0.79	0.53
4	Avg EV Error	0.0026	0.0220	0	0.0032	0	0	0.1982	0.0778
	Avg EV Distance	0.01	0.07	0	0.01	0	0	0.62	0.24
5	Avg EV Error	0	0.0041	0.0385	0.0941	0	0.0914	0.2859	0.2878
	Avg EV Distance	0	0.01	0.12	0.29	0	0.28	0.88	0.88
6	Avg EV Error	0	0.0074	0.0143	0.0357	0.0026	0.0192	0.3601	0.4206
	Avg EV Distance	0	0.02	0.05	0.11	0.01	0.05	1.09	1.26
7	Avg EV Error	0.0032	0.0068	0.2933	0.0893	0.0249	0.1069	0.0622	0.2125
	Avg EV Distance	0.01	0.02	0.88	0.26	0.08	0.31	0.19	0.64
8	Avg EV Error	0	0.0041	0.0082	0.0297	0.0026	0.0126	0.1421	0.1421
	Avg EV Distance	0	0.01	0.04	0.12	0.01	0.04	0.39	0.39
9	Avg EV Error	0	0	0.0249	0.0646	0	0.0	0.1972	0.2298
	Avg EV Distance	0	0	0.08	0.18	0	0	0.62	0.7

Table 6.1: The AE algorithms' output errors with downsized input images. "EV Error" stands for the absolute error of the exposure value between the scaled and the full size input image. "EV Distance" indicates the absolute value of the difference of indices in the 40 EV stack between the scaled and the full size input image.

Chapter 7

Conclusion

This final chapter summarizes the work covered in this thesis and potential future work.

7.1 Overall Summary

This thesis aimed to develop a platform for studying AE algorithms in challenging lighting environments. As part of this effort, a stop-motion setup captured a temporal sequence with an exposure stack at each time step. The 4D dataset provides all possible exposures over a simulated video.

A software platform was developed that allowed the testing of different AE algorithms. The platform allows users to visualize the algorithm's solution within the 4D dataset. The dataset consists of 36,000 images that emulate nine scenes, each with 100-time steps and each time step with 40 exposures. Our scenes include a variety of objects, object motion, and lighting configurations. Four of the implemented AE algorithms (*global*, *semantic*, *saliency*, and *entropy*) were tested on all scenes in our dataset. We used our platform to

produce videos from these algorithms' output for use in a user study to determine if there is a preference between methods. We found that users preferred semantic and saliency methods, where a region of interest was weighted more in the exposure decision. There was no significant difference between AE algorithms for time steps with relatively standard lighting conditions (e.g., Figure 6.6 between time steps 55-65).

However, challenging lighting conditions in our dataset demonstrated the strength of the saliency and semantic algorithms. Additionally, we found entropy to be a weak metric for AE algorithm design. Our dataset and platform will be a valuable contribution to the computer vision community for developing improved AE algorithms for challenging lighting conditions.

7.2 Future Work

One purpose of this project was to revitalize interest in exposure algorithm development. The AE algorithms we tested are implemented based on their basic concepts. Since our dataset and platform enable repeated output visualization over a temporal full exposure stack, researchers may further examine AE algorithms via specific settings as needed and tune parameters repeatedly for better performance. Furthermore, our dataset is a convenient resource for HDR-related research because it contains much more information than the existing HDR datasets mentioned in Section 2.2.

Although the dataset is large, more scenes with customer-defined conditions that apply to real-world applications could be captured under similar settings. For instance, to mimic the scenario of entering and exiting a tunnel or to include pattern detection targets such

as QR codes inside the challenge lighting scenes. More scenes would continue to help improve AE algorithms. In addition, this would help facilitate machine learning-based AE research, which requires a massive amount of data.

Bibliography

[1] S. Kelby, *The digital photography book*. Peachpit Press, 2015. 6

[2] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, pp. 34–43, 2005. 6

[3] H. Yang, B. Wang, N. Vesdapunt, M. Guo, and S. B. Kang, "Personalized exposure control using adaptive metering and reinforcement learning," *TVCG*, vol. 25, no. 10, pp. 2953–2968, 2018. 8, 9

[4] E. Onzon, F. Mannan, and F. Heide, "Neural auto-exposure for high-dynamic range object detection," in *CVPR*, 2021. 8, 9

[5] L. Yahiaoui, J. Horgan, S. Yogamani, C. Eising, and B. Deegan, "Impact analysis and tuning strategies for camera image signal processing parameters in computer vision," in *Irish Machine Vision and Image Processing*, 2018. 8

[6] J. B. Phillips and H. Eliasson, "The camera," in *Camera Image Quality Benchmarking*, 2018. 8

[7] S. Battiato, A. R. Bruna, G. Messina, and G. Puglisi, *Image Processing for Embedded Devices*. Applied digital imaging, 2010. 8

[8] S. Schulz, M. Grimm, and R.-R. Grigat, "Using brightness histogram to perform optimum auto exposure," *WSEAS Transactions on Systems and Control*, 2007.

[9] N. Kehtarnavaz, H.-J. Oh, I. Shidate, Y. Yoo, and R. Taluri, "New approach to auto-white-balancing and auto-exposure for digital still cameras," *SPIE*, vol. 4669, pp. 268–276, 2002. 8

[10] J. Torres and J. M. Menéndez, "Optimal camera exposure for video surveillance systems by predictive control of shutter speed, aperture, and gain," in *Real-Time Image and Video Processing*, 2015. 8

[11] T. Kuno, H. Sugiura, and N. Matoba, "A new automatic exposure system for digital still cameras," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 192–199, 1998. 8

[12] T. Jiang, K.-D. Kuhnert, D. Nguyen, and L. Kuhnert, "Multiple templates auto exposure control based on luminance histogram for onboard camera," in *IEEE International Conference on Computer Science and Automation Engineering*, vol. 3, pp. 237–241, 2011. 8, 26

[13] Y. Kinoshita, S. Shiota, and H. Kiya, "Automatic exposure compensation for multi-exposure image fusion," in *ICIP*, 2018. 8

[14] M. Cho, S. Lee, and B. D. Nam, "Fast auto-exposure algorithm based on numerical analysis," *SPIE*, vol. 3650, pp. 93–99, 1999.

[15] Q. K. Vuong, S.-H. Yun, and S. Kim, "A new auto exposure and auto white-balance algorithm to detect high dynamic range conditions using cmos technology," *Third International Conference on Convergence and Hybrid Information Technology*, 2008. 8

[16] Y. Su and C.-C. Jay Kuo, "Fast and robust camera's auto exposure control using convex or concave model," in *IEEE International Conference on Consumer Electronics*, 2015. 8

[17] S. Park, G. Kim, and J. Jeon, "The method of auto exposure control for low-end digital camera," in *International Conference on Advanced Communication Technology*, 2009.

[18] Y. Su, J. Y. Lin, and C.-C. J. Kuo, "A model-based approach to camera's auto exposure control," *Journal of Visual Communication and Image Representation*, vol. 36, pp. 122–129, 2016. 8, 13

[19] C. Zhang, Z. You, and S. Yu, "An automatic exposure algorithm based on information entropy," in *Sixth International Symposium on Instrumentation and Control Technology*, 2006. 8, 26, 31, 37

[20] J. Ning, T. Lu, L. Liu, L. Guo, and X. Jin, "The optimization and design of the auto-exposure algorithm based on image entropy," in *8th International Congress on Image and Signal Processing*, 2015. 8

[21] H. Lu, H. Zhang, S. Yang, and Z. Zheng, "A novel camera parameters auto-adjusting

method based on image entropy," in *RoboCup 2009: Robot Soccer World Cup XIII 13*, Springer, 2010. 8, 37

[22] M.-A. Bégin and I. Hunter, "Auto-exposure algorithm for enhanced mobile robot localization in challenging light conditions," *Sensors*, vol. 22, p. 835, 2022. 8

[23] I. Shim, T.-H. Oh, J.-Y. Lee, J. Choi, D.-G. Choi, and I. S. Kweon, "Gradient-based camera exposure control for outdoor mobile platforms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1569–1583, 2019. 8, 26, 31

[24] E. W. Jin, S. Lin, and D. Dharumalingam, "Face detection assisted auto exposure: supporting evidence from a psychophysical study," *SPIE*, vol. 7537, 2010. 9, 28

[25] L. Yuan and J. Sun, "Automatic exposure correction of consumer photographs," in *ECCV*, 2012. 9

[26] D. Guo, Y. Cheng, S. Zhuo, and T. Sim, "Correcting over-exposure in photographs," in *CVPR*, 2010.

[27] S. B. Kang, A. Kapoor, and D. Lischinski, "Personalization of image enhancement," in *CVPR*, 2010.

[28] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *CVPR*, 2011.

[29] N. Joshi, W. Matusik, E. Adelson, and D. Kriegman, "Personal photo enhancement using example images," *ACM TOG*, vol. 29, no. 2, p. 15, 2010. 9

[30] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister, "Image restoration using online photo collections," in *ICCV*, 2009. 9

[31] M. Afifi, K. G. Derpanis, B. Ommer, and M. S. Brown, "Learning multi-scale photo exposure correction," in *CVPR*, 2021.

[32] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong, "Zero-reference deep curve estimation for low-light image enhancement," in *CVPR*, 2020. 9

[33] H.-J. Kwon and S.-H. Lee, "Contrast sensitivity based multiscale base–detail separation for enhanced hdr imaging," *Applied Sciences*, vol. 10, no. 7, 2020. 9

[34] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-image hdr reconstruction by learning to reverse the camera pipeline," in *CVPR*, 2020.

[35] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep CNNs," *ACM TOG*, vol. 36, no. 6, pp. 1–15, 2017. 9

[36] B. Funt and L. Shi, "The effect of exposure on maxrgb color constancy," *SPIE*, vol. 7527, 2010. 9

[37] I. Morawski, Y.-A. Chen, Y.-S. Lin, S. Dangi, K. He, and W. H. Hsu, "Genisp: Neural isp for low-light machine cognition," in *CVPRW*, 2022. 9

[38] J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, and H. Brendel, "Creating cinematic wide gamut hdr-video for the evaluation of tone mapping operators and hdr-displays," *SPIE*, vol. 9023, 2014. 9

[39] G. Chen, C. Chen, S. Guo, Z. Liang, K.-Y. K. Wong, and L. Zhang, "HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset," in *ICCV*, 2021.

[40] A. Banitalebi-Dehkordi, M. Azimi, M. T. Pourazad, and P. Nasiopoulos, "Compression of high dynamic range video using the HEVC and H.264/AVC standards," in *QSHINE*, 2014. 9

[41] A. Abuolaim, A. Punnappurath, and M. S. Brown, "Revisiting autofocus for smartphone cameras," in *ECCV*, 2018. 11, 41

[42] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *CVPR*, 2018. 13

[43] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch, "Minimum barrier salient object detection at 80 fps," in *ICCV*, 2015. 28, 32, 33