

K-means Clustering Based Tone-Mapping Operator For High Dynamic Range Video

by

Negar Taherian
Bachelor of Engineering in Electrical Engineering
Ryerson University, 2015

A thesis presented to Ryerson University
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering
In the program of Electrical and Computer Engineering
Toronto, Ontario, Canada 2017

©Negar Taherian, 2017

K-means Clustering Based Tone-Mapping Operator For High Dynamic Range Video

Negar Taherian

Master of Applied Science in Electrical Engineering
Department of Electrical and Computer Engineering
Ryerson University, 2017

Abstract

The field of high dynamic range (HDR) imaging deals with capturing the luminance of a natural scene, usually varying between 10^{-3} to 10^5 cd/m^2 and displaying it on digital devices with much lower dynamic range. Here, we present a novel tone mapping algorithm that is based on K-means clustering. Our algorithm takes into account the color information within a frame and using k-means clustering algorithm it builds clusters on the intensities within an image and shifts the values within each cluster to a displayable dynamic range. We also implement a scene change detection to reduce the running time of our algorithm by using the cluster information from the previous frame for frames within the same scene. To reduce the flicker effect, we proposed a new method that multiplies a leaky integer to the centroid values of our clustering results. Our algorithm runs in $O(N \log K + K \log K)$ for an image with N input luminance levels and K output levels. We also show how to extend the method to handle video input. We display that our algorithm gives comparable results to state-of-the-art tone mapping algorithms. We test our algorithm on a number of standard high dynamic range images and video sequences and provide qualitative and quantitative comparisons to a number of state-of-the-art tone mapping algorithms for videos.

Acknowledgments

I would like to thank all the people who made this thesis possible.

First of all, my supervisor Dr. Dimitri Androutsos for giving me the opportunity to pursue my graduate studies in the Department of Electrical and Computer Engineering at Ryerson University and for guiding me through the process. Second, I would like to thank my family especially my loving husband who provided his unconditional support over the course of this degree. I also thank God for giving me the strength to finish this work. In the end, I like to dedicate this work to the loving memory of my father who I know would have been proud if he were with us.

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis may be made electronically available to the public.

TABLE OF CONTENTS

List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Objective	2
1.2.1 Contribution	3
1.3 Thesis Outline	3
2 Background	4
2.1 HDR-to-LDR Tone Mapping Operators	4
2.2 Global Tone Mapping Operators	5
2.2.1 Histogram Equalization for Tone Mapping Operator	6
2.2.2 Gradient Domain based Tone Mapping Operator	7
2.2.3 Photographic Tone Reproduction	7
2.2.4 Adaptive Logarithmic Mapping	8
2.3 Local Tone Mapping Operators	9
2.3.1 Globally Optimized Linear Windowed Tone Mapping	9
2.3.2 Display Adaptive Tone-mapping	10
2.3.3 Tone-mapping using Bilateral Filtering	11
2.3.4 Tone mapping using K-means clustering	11
2.4 Edge-Aware Filtering	13

2.4.1	Bilateral filter	13
2.4.2	Weighted Least Squares (WLS) filter	14
2.4.3	Local Laplacian Filter	15
2.5	HDR Video Tone Mapping	17
2.5.1	Local HDR Video Tone Mapping	17
2.6	Clustering	19
2.6.1	The C-Means clustering Algorithm (CMA)	20
2.6.2	Hierarchical Clustering Algorithms	23
2.6.3	Fuzzy C-Means Clustering	24
2.7	Chapter Summary	24
3	Proposed Algorithm	26
3.1	Introduction	26
3.1.1	Finding the number of clusters used in K-means	26
3.1.2	Calculating the intensity channel (Dimensionality Reduction)	29
3.1.3	Taking the log of intensity channel	30
3.1.4	Performing K-means algorithm	30
3.1.5	Adjusting the color based on K-means results	31
3.1.6	Applying Gaussian filter for smoother local factors	33
3.2	Video Processing	33
3.2.1	Determining the first frame of a scene	35
3.2.2	Video Flicker Removal	35
3.3	Chapter Summary	36
4	Results and Analyses	37
4.1	Results on HDR Images	37
4.2	Results on HDR Video	41
4.2.1	Comparison With Other Methods	41
4.2.2	Subjective Study	44
4.3	Chapter Summary	46

5	Conclusions and Recommendations	49
5.1	Recommendations for Future Work	50
	Bibliography	57

List of Figures

2-1	HDR image (bottom) made out of 4 LDR images (top) with different exposures. (Image courtesy of Wikipedia)	5
2-2	Before (left) and after (right) the application of HE on an image . . .	7
2-3	Durand’s method [15] (shown on the top right) and Reinhard’s method [2] (shown on the top left) and Fattal’s method [11] (shown on the bottom left) faithfully reproduce the visual appearance. (Image courtesy of ESPL-LIVE HDR Image Quality Database[16], [17])	9
2-4	Bilateral Filter. illustrates how the weights are computed for one pixel near an edge. adapted from Paris et. al [23]	14
2-5	A comparison between edge aware filters. Top left image is the input image before any filters, image on (top right) is Bilateral filtered version, image on (bottom left) is weighted least squares filtered version, image on (bottom right) is local laplacian filtered version[30].	16
2-6	The visual trade-off between emphasizing spatial contrast (a, b) and temporal contrast (a, c). While in both settings frame 38 remains the same (a), frame 265 can be adjusted to either maintain spatial (b) or temporal contrast (c). Figure adapted from[3]	18
2-7	Classifications of Clustering Methods	20
2-8	K-means Example. Adapted from wikipedia.	22
2-9	Top image shows data before using Fuzzy C-means clustering, top right shows clustered data after 8 iterations, bottom left image shows clustered after 37 iterations.	25

3-1	Flow chart of our proposed method	27
3-2	(a,b) a frame of a video before tone mapping with its histogram (c,d) intensity channel of frame and its histogram (e,f) frame color regions based on K-means (g,h) Tone mapped frame and its histogram.	32
3-3	(Top left) Original HDR image (Top right) Tone mapped version without smoothing local factors (Bottom left) Tone mapped version with Gaussian filter with sigma equal to 1/1200 of the bigger side of the frame (i.e. 713 pixels) (Bottom right) Tone mapped version with Gaussian filter with sigma equal to 1/300 of the bigger side of the frame (i.e. 713 pixels)	34
3-4	EMD distance between frames of a video with 189 frames. The red eclipse shows the maximum value of EMD which is 1.853 at the frame 114.	36
4-1	(a) image tone mapped with Drago method[14],(b) image tone mapped with Mantiuk et al. [19], (c) image tone mapped with Reinhard et al. [5], (d) image tone mapped with our proposed method. One can see that the compared methods suffer from over-saturation, color artifacts and loss of detail.	38
4-2	(a) image tone mapped with Drago method [14],(b) image tone mapped with Mantiuk et al. [19], (c) image tone mapped with Reinhard et al. [5], (d) image tone mapped with our proposed method. One can see that the compared methods suffer from over-saturation, color artifacts and loss of detail.	39
4-3	The result of running our Algorithm on a number of HDR images. HDR radiance maps courtesy of Debevec [60] and Mantiuk [59]	40
4-4	The figure shows two frames (left and right, respectively) of the output from our method, (from top to bottom) the exhibition sequence, the hallway2 sequence, the students sequence, the window sequence. The results are best viewed on screen	42

4-5	Comparison different current TMO methods capable of processing videos. For each method, two representative frames and plot of mean pixel values (tone mapped) and log-mean luminance (HDR) over time. Note that the HDR plots are shifted and compressed by an exponent for presentation. The image is taken from [3] the method (e) referred to as 'Our method' represents their proposed method.	43
4-6	Frame 80 (top left) and frame 265 (top right) from Hallway video sequence. (bottom left) Mean intensity value of each frame in Hallway video sequence before and after being tone-mapped by our method. .	45
4-7	Comparison of our method (a) with Ramseyetal.[2004] [36] (gray) and Gastal et al. [2011] (recursive filtering version) [27] (blue) combined with Boitard et al.'s [2014] [46]segmentation based temporal coherency method (both provided by Ronan Boitard). Also Aydin et al. (red)[3]coherent temporal tone-mapping for HDR videos is shown. This Images represent the mean intensity value of frames in Hallway video sequence after being tone-mapped by the following methods.	45
4-8	Average ratings based on the subjective study	46

List of Tables

1.1	Ambient luminance for some lighting conditions from [1].	1
4.1	The tested HDR sequences from[62].	41
4.2	: List of tone mapping operators included in our survey. Processing refers to either global processing that is identical for all the pixels within a frame or local processing that may vary spatially. This table is adapted from [44]	48

Chapter 1

Introduction

1.1 Motivation

The luminance of a natural scene often has a high dynamic range (HDR), varying between 10^{-3} to 10^5 cd/m², that unlike digital displays can be handled by the human visual system [1]. In digital imaging, typically 8-bits per color channel are used resulting in 2^{24} number of distinct colors. Although this seems like a large number, it is not enough to represent the range of luminances found in the real world. For instance, from starlight to sunlight the real world contains more than 10 log units (orders of magnitude) of dynamic range which is defined as the ratio of the maximum to the minimum luminance. The luminance levels of several typical scenes are given in Table 1.1 for illustration purposes.

Table 1.1: Ambient luminance for some lighting conditions from [1].

Condition	luminance(cd/m ²)
Starlight	10^{-3}
Moonlight	10^{-1}
Indoor lighting	10^2
Sunlight	10^5

The dynamic range in natural scenes may vary depending on several factors such as scene content and lighting conditions. However, on average, most real world scenes

contain at least 4 log units of dynamic range (1:10000) [2]. A standard digital display has a lower dynamic range (LDR) of about 10^2 cd/m². In the absence of displays that can show the full dynamic range of a scene as captured the need for tone mapping operators arises. The field of tone mapping in high dynamic range imaging deals with displaying of images or videos with wide dynamic range on devices that have much lower dynamic range[2]. Different tone mapping operators (TMOs) emphasize on preserving different attributes during this process. For instance, while some operators aim to reproduce all details, others strive to maintain the global contrast instead of details. More information on these operators will be provided in the next chapter.

Today, many camera devices have built-in functionality for acquiring HDR images, and this resulted in the growth of HDR field. Particularly regarding video technologies that keep improving the viewer's experience by for example going from HD to 4K, or increasing the frame rates in cinemas.

As modern high-end cameras have the ability to capture high dynamic range (HDR) videos the research on finding a tone mapping operation as means of visualization and artistic expression has further increased. Precisely, it is expected to reduce the dynamic range of a captured HDR video in a fashion that its visual content is maintained and that no artifacts are introduced[3].

Although there have been some efforts in recent years to address the following shortcomings, still the majority of HDR video tone mapping operators can not achieve both of the goals mentioned earlier.

1.2 Objective

The objective of this thesis is to develop a new HDR tone-mapping operator that works on HDR videos. We also take a look into HDR image tone-mapping as a foundation for video tone-mapping.

1.2.1 Contribution

We propose a tone mapping operator that can perform local processing based on the colors present locally and provides an output free from ghosting and flickering. Aforementioned is to achieve sufficient dynamic range compression in all circumstances while maintaining a right level of detail and contrast. The proposed HDR tone mapping operator is also efficient in processing a large amount of data in abbreviated run time. It also needs no parameter tuning, and there is no need for calibrating the input data. This work intends to solve tone mapping problem with clustering techniques. To be precise, we want to cluster input intensity levels from a larger range to a set with much smaller range. We perform clustering in three dimensions and illustrate how unlike other methods that only work with luminance channel, we can achieve more natural looking results.

1.3 Thesis Outline

Following the introductory chapter, the remainder of this thesis is organized as follows; Chapter 2 reviews several TMOs, for both HDR image and video. It will review some of the most well-known clustering methods with more emphasis on K-means clustering that prepare the necessary background knowledge for later chapters. Chapter 3 elaborates on the structure of the algorithm with a focus on the improvements over the previous frame works. Chapter 4 presents the results obtained using the proposed algorithm. Chapter 5 concludes the thesis and discusses future work.

Chapter 2

Background

This chapter starts with a literature review of tone mapping operators (TMO) for HDR images, followed by (TMO) for HDR video. We finish the chapter by discussing clustering techniques as means to reduce dynamic range in both images and videos.

2.1 HDR-to-LDR Tone Mapping Operators

High dynamic range (HDR) image¹ tone mapping has been studied extensively over the years, and there is a comprehensive overview of the field by Reinhard et al. [2]. High dynamic range (HDR) images represent greater accuracy of luminance levels in natural scenes than standard low dynamic range (LDR) images [5]. They can be directly acquired with HDR imaging devices [2] or created by fusing differently exposed images of the same real scene, as illustrated in Fig. 2.1. The latter method, exposure fusion, aims to directly create a detailed LDR image from a set of bracketed exposures. That is, in this pipeline, the generation of the HDR image is bypassed. For instance, the HDR mode on iPhone 4 and above models utilizes an exclusive algorithm of this type[6].

HDR-to-LDR TMOs facilitate display of HDR images on viewing devices with lower dynamic range. The goal of TMOs is to compress the dynamic range of HDR

¹The term "dynamic range" for images is defined as the ratio between the lightest and darkest pixels [4].



Figure 2-1: HDR image (bottom) made out of 4 LDR images (top) with different exposures. (Image courtesy of Wikipedia)

images while preserving their structural detail and natural appearance. Tone mapping algorithms can be divided into two categories of global and local operators. It is worth mentioning that results of global and local TMOs can be considerably different and some people tend to like one over the other based on their aesthetic preferences.

2.2 Global Tone Mapping Operators

Global TMOs are essentially point-wise luminance transformations like sigmoid function or histogram based algorithms that apply same mapping function across the image[7]–[9]. They produce an LDR images that are spatially consistent and are often not very computationally expensive algorithms. Their downside is that they fail to provide fine detail in local areas particularly in HDR images that contain both light and dark areas meaning that they do not cope well with huge contrast ratios. Examples for sigmoid functions used for tone-mapping purpose Gamma mapping and

log-normal mapping. Gamma mapping takes the following formula

$$Y = X^{\frac{1}{\gamma}}$$

where X and Y represent HDR and tone mapped LDR images, respectively. Log-normal mapping utilizes the logarithmic function to boost lower luminance levels and to compress higher luminance levels. The tone mapped image is computed by

$$Y = \frac{\log_2^X - x_{min}}{x_{max} - x_{min}}$$

where again X and Y represent HDR and tone mapped LDR images, respectively. x_{min} and x_{max} are the minimum and maximum pixel values of \log_2^X respectively. The downside of it is that log-normal mapping often produces blanched images with blurred detail and strange appearance.

2.2.1 Histogram Equalization for Tone Mapping Operator

One global TMO is the technique developed by Larson et al [7] which is a histogram adjustment method to perform tone mapping by accounting human visual sensitivity, color sensitivity, and visual acuity. In image processing Histogram Equalization (HE) is a popular technique utilized to increase the global contrast of an image. Histogram Equalization (HE) is a process that spreads the pixel intensities over the available range and hence increasing the global contrast of the image. Figure 2-2 illustrates the application of an HE. To accomplish this cumulative distribution function is used. This function is used to expand bins associated with many pixels and shrink bins with smaller pixel density (see Figure 2-2). Larsen et al.[7] build the histogram of luminance values in log domain since the human visual system (HVS) is almost log linear when photoreceptors are fully responsive [10]. Next using Cumulative distribution function they redistribute logarithmic HDR luminance values. Later they transfer the values back to the luminance domain and normalize them to the range [0, 1] which are then gamma encoded using the BT.1886 and quantized on

the targeted bit-depth. This TMO struggles in tone reproduction of relatively large areas with small luminance range like sky or night scene since most of the available dynamic range gets assigned to a large bin when using HE. As a result, applying histogram equalization will usually lead to not efficiently utilizing the colorspace, due to discretization effects.

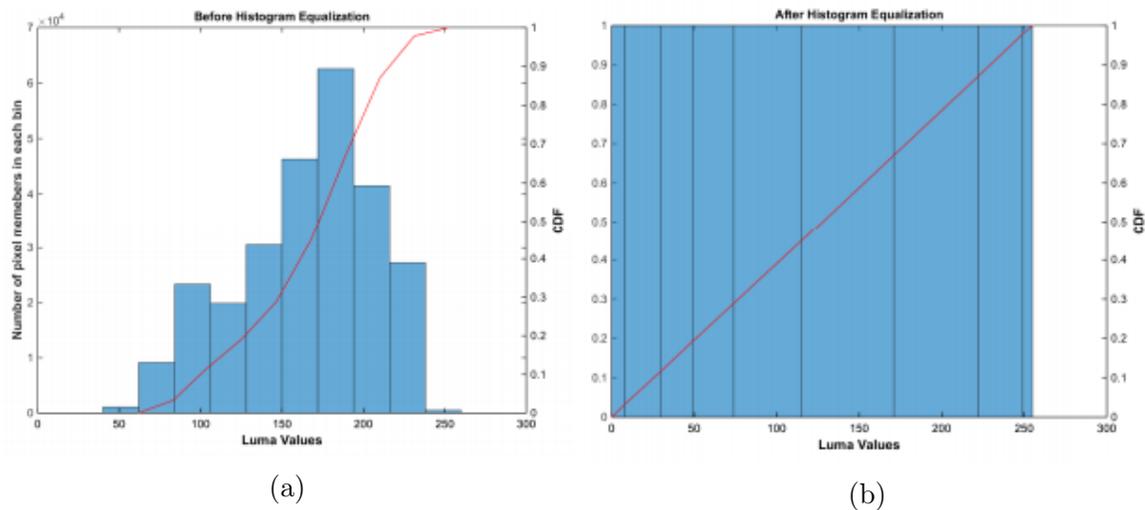


Figure 2-2: Before (left) and after (right) the application of HE on an image

2.2.2 Gradient Domain based Tone Mapping Operator

Another global TMO is the technique by Fattal et al. [11] which uses the gradient domain² to reduce the dynamic range of the image. They use the fact that in gradient domain large values indicate a great change in the pixel values of the image. And small values correspond to fine details. Their algorithm reduced large gradients without changing their direction and kept the smaller gradients the same. This way they were able to reconstruct a reduced dynamic range image.

2.2.3 Photographic Tone Reproduction

Another well-known global TMO is Reinhard's method [5] which is considered to be among the best TMOs on several independent subjective tests [12], [13]. In [5] Rein-

²Gradient domain image processing is a type of digital image processing that operates on the differences between neighboring pixels, rather than on the pixel values directly.

hard et al. stimulated dodging and burning technique used in traditional photography allowing different exposures across the image to be printed. They first approximated the key ³ of a scene as the log average of luminance and scaled the luminance based on the key value. They assign high key values to lower luminance values, and vice versa to convey detail. They further compressed high luminance by the function $f(x) = x/(1 + x)$. Finally, they adjust the resulting tone mapped LDR image locally to improve the detail further.

2.2.4 Adaptive Logarithmic Mapping

In [14], Drago et al. recommended a global tone mapping function based on adaptive logarithmic compression of luminance. To preserve detail, they used logarithmic functions with varying bases ranging from log2 to log10. The log10 is utilized for the brightest image pixels, for the rest, the logarithm base is changed between the base of 2-10 as the function of their luminance. Perlin bias power function is used for interpolation between the logarithm bases to provide better steepness control of the resulting tone mapping curve. The following formula was used to map the pixel values

$$L_d = \frac{L_{dmax} \cdot 0.01}{\log_{10}(L_{wmax} + 1)} \cdot \frac{\log(L_w + 1)}{\log \left(2 + \left(\left(\frac{L_w}{L_{wmax}} \right)^{\frac{\log(b)}{\log(0.5)}} \right) \cdot 8 \right)}$$

L_d is the displaying value of each pixel. L_{dmax} is the maximum luminance capability of the displaying medium. L_w is the luminance value of pixel and L_{wmax} is the maximum luminance value of image. The parameter of bias function denoted by b is calculated using Perlin bias power function.

Figure 2-3, illustrates the results produced by these methods. As depicted Reinhard's method[2] and Drago's method[14]generate results with less noticeable artifacts.

³A scene's key is an indicator of how light or dark the overall impression of a scene is [2].

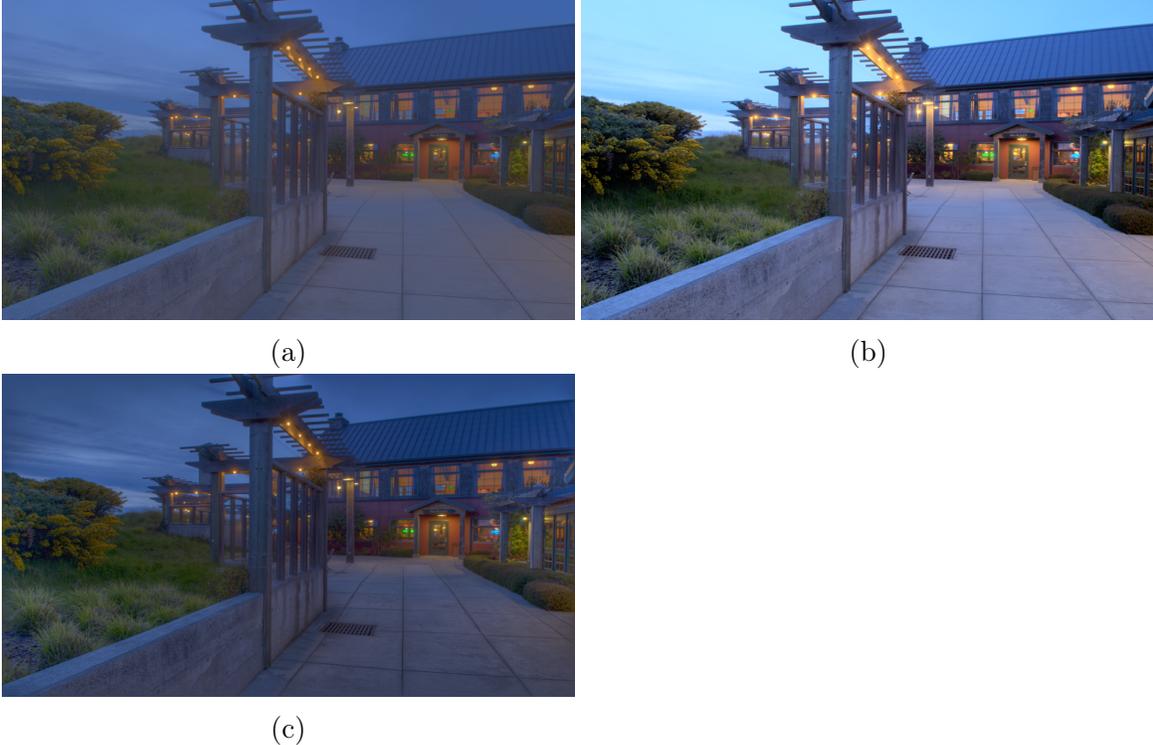


Figure 2-3: Durand’s method [15] (shown on the top right) and Reinhard’s method [2] (shown on the top left) and Fattal’s method [11] (shown on the bottom left) faithfully reproduce the visual appearance. (Image courtesy of ESPL-LIVE HDR Image Quality Database[16], [17])

2.3 Local Tone Mapping Operators

To increase local contrast, local tone mapping algorithms usually apply local filtering. Local TMOs are designed based on the way the human visual system (HVS) responds to local scene luminance and contrast.

2.3.1 Globally Optimized Linear Windowed Tone Mapping

In [18], Shan et al. proposed a method that performs local tone mapping on overlapping windows in an image to reconstruct the image radiance. They categorized the transition of the scene as smooth or sharp and based on that they use a local approach that preserves the local properties of the image. In this method, they directly process the image radiance instead of decomposing the image to different layers or segmenting it. As a result, the algorithm does not cause problems such as halo

effects that are caused by layer decomposition. They compress the strong edges by preserving the small details and impose an optimization problem which combines a set of local-based constraints. The algorithm is defined as a linear mapping of HDR radiance map to LDR within small radiance groups on the image called windows. The mapping is formulated with the following primary linear function:

$$I^l(j) = p_i I^h(j) + q_i, \quad j \in w_i$$

where p and q are linear function parameters, I^l is the compressed pixel value, I^h is the original pixel value, and w_i is the window i . The problem is essentially defined as an objective function minimization:

$$f = \sum_i \left(\sum_{j \in w_i} (I^l(j) - p_i I^h(j) - q_i)^2 + \epsilon c_i^{-2} (p_i - c_i)^2 \right)$$

The term $c_i^{-2} (p_i - c_i)^2$ is squared relative error of guidance map c_i which is contributed to the objective function in order to avoid trivial solution to p_i and q_i : 1 and 0. ϵ is weight of guidance error term. The problem is solved by first setting the partial derivatives of p_i and q_i to zero and calculating the optimal I_i^l by solving the resulting linear system. The window size is set to 3×3 by default.

2.3.2 Display Adaptive Tone-mapping

One other well known local tone mapping algorithms is the display adaptive TMO by Mantiuk et al. [19]. This operator also takes into account the visual properties of the display device and the human visual system.

This operator tries to solve an optimization problem that essentially updates the tone mapping parameters to minimize the difference between the HVS model response of the original image and the displayed image. They also take into account the display properties and viewing conditions. They model the display device with the following formula:

$$L_d(L') = (L')^\gamma \cdot (L_{max} - L_{black}) + L_{black} + L_{refl}$$

where L_d is displayed luminance, L' is the pixel value in range 0 - 1, γ is gamma value of the display (default 2.2), L_{max} and L_{min} are the maximum and minimum luminance value of the display and L_{refl} is the ambient light reflected from the display surface which affects the minimum luminance value of the display. L_{refl} is calculated with the following formula:

$$L_{refl} = \frac{k}{\pi} E_{amb}$$

where E_{amb} is the ambient luminance value in lux and k is the reflectivity of the display panel.

2.3.3 Tone-mapping using Bilateral Filtering

In [15] Durand et al., first extracted the edges of an HDR image using a bilateral filter [20]. Then using linear scaling in the logarithmic domain, they reduce the dynamic range. Finally, they obtain LDR image by adding back the extracted edges. To put it simply they decompose the base layer from the detail layer and then add back the tone mapped base layer to the detail layer. In principle, our method is also based on break down of base and detail layer by utilizing edge-aware filters. Hence we review edge-aware filtering in the subsequent section from a tone mapping aspect.

2.3.4 Tone mapping using K-means clustering

In [21] they addressed the tone mapping problem by using K-means to cluster the image into regions and applying proper gamma correction⁴ to each segment, according to the mean value of each region. In the proposed algorithm, the luminance component of the HDR radiance map is first filtered by a bilateral filter. Then using a logarithmic function the filtered radiance map is as globally assigned for global contrast enhancement. Then using K-means algorithm, they divide the bilateral filtered luminance into regions. Next, they set the display gamma value automatically according to the mean value of each region. Then, the tone of HDR image is reproduced

⁴Gamma correction, or often simply gamma, is the name of a nonlinear operation used to encode and decode luminance or tristimulus values in video or still image systems.

by a local TM method with adaptive gamma value.

In [22] they also address the tone-mapping problem with the use of K-means clustering. They use dynamic programming to solve the K-means clustering and find the global optimum. They claim that their algorithm runs in $O(N^2K)$ for an image with N input luminance levels and K output levels.

In this paper they use $K = 256$ for their K-means clustering, corresponding to 8-bit output, and they claim that this value can be set to any output quantization level desired, but 256 performs the best. They first start by estimating the intensity channel which is the maximum of the of the three channels and claim that maximum of three channels generates the best result with their method. Next, they cluster the intensity channel into K desired levels. Based on the clusters they then calculate a transform function for each intensity value. Following is the steps of their algorithm.

-
- 1: Given a high bit color input image: I_{in}
 - 2: Calculate the intensity channel I_{gr} of I_{in} .
 - 3: Take the log to get $I_{log} = \log I_{gr}$
 - 4: Calculate the histogram $h(s)$ of I_{log} .
 - 5: Find centers c_l using their suggested K-means algorithm.
 - 6: Estimate $F(s) : u_i \rightarrow c_l$ using nearest neighbours.
 - 7: $I_{out}^{ch} = F(I_{out}^{ch})$ or $I_{out}^{ch} = \frac{I_{ch}}{I_{gr}} \cdot F(I_{gr})$.
-

They claim that the performance of their algorithm is comparable to the state-of-the-art tone mapping algorithms and with the significant benefit of the minimal need for parameter setting. In their case, user is not able to automatically choose some parameters to achieve a better effect in local color and contrast; factor above is desirable in some HDR imaging applications.

2.4 Edge-Aware Filtering

Based on the study by Mi-lanfar [2013] on the use cases of edge aware filters HDR image tone mapping is one of the common utilization of these types of filters since the establishment of edge-aware filtering based tone mapping [Durand and Dorsey 2002]. Nevertheless, the use case of edge-aware filters in HDR image tone-mapping is often brief since the primary focus of these works is reducing the dynamic range; consequently, it is not straightforward to conclude how properly an edge-aware filter is performing on HDR tone mapping. Some of the most well-known edge aware filters used for the tone mapping application is as follows

2.4.1 Bilateral filter

The bilateral filter [20] is famous to create halo⁵ and ringing⁶ artifacts in the resulting image. It replaces each pixel with a weighted average of its neighbours. It works based on the difference in the value of neighbouring pixels with the fundamental idea that for a pixel to influence another pixel, it should have not only a nearby location but also a similar value. Therefore, each neighbouring pixel gets weighted based on two components of distance and intensity. The combination of these two elements assures that only nearby similar pixels contribute to the final result. The bilateral filter, denoted by $BF[\hat{u}]$, is defined by:

$$BF [I]_p = \frac{1}{W_p} \sum_q G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q$$

where normalization factor W_p ensures pixel weights sum to 1.0:

$$W_p = \sum_q G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q$$

⁵In digital image processing halo refers to a light line around the edges of image.

⁶In digital image processing ringing refers to artifacts that appear as oscillation at a fading rate around a sharp transition in the input.

Parameters σ_r and σ_s will specify the amount of filtering for the image I . Equation (3) is a normalized weighted average where G_{σ_r} is a spatial Gaussian weighting that decreases the influence of distant pixels, G_{σ_s} is a range Gaussian that decreases the influence of pixels q when their intensity values differ from I_p .

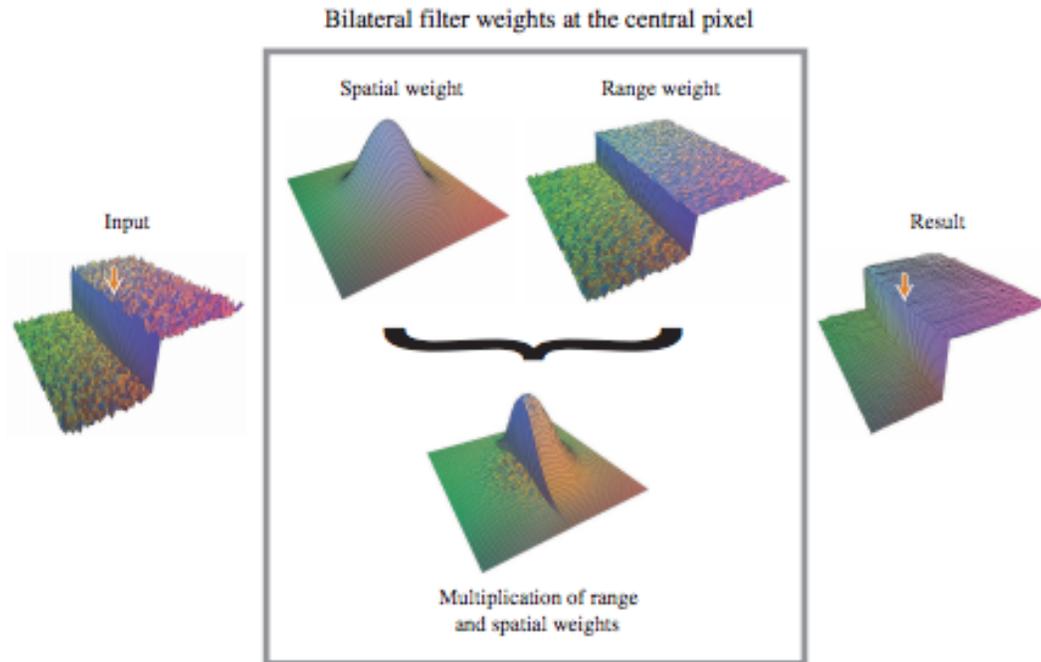


Figure 2-4: Bilateral Filter. illustrates how the weights are computed for one pixel near an edge. adapted from Paris et. al [23]

2.4.2 Weighted Least Squares (WLS) filter

This filter was originally used to reduce ringing artifacts while de-blurring images in the presence of noise [24]. By minimizing the function whose, data term penalizes the distance between the original and filtered image it can prevent halo artifacts. The disadvantage of this filter is that it has to solve large numerically-challenging linear systems [25] that involve conjugate gradients [26]. That makes it a computationally expensive, but it still is one of the go-to methods for generating high-quality results [25], [27]. WLS smoothing is uniquely defined as the solution of the following linear

system

$$I_p = J_p + \sum_{q \in N_4(p)} a_{pq}(J_p - J_q)$$

where I_p and J_p are pixel values of the input image and the smoothed output image, respectively, $N_4(p)$ is the 4-neighborhood around pixel p , and a_{pq} are the smoothness weights as defined in [28].

$$a_{pq} := \frac{\lambda}{|I_p - I_q|^\alpha + \epsilon}$$

The authors solve the equation system using a multi-resolution preconditioned conjugate gradient solver.

2.4.3 Local Laplacian Filter

Introduced by Paris et al. [26], these filters produce high-quality results with detail enhancement with no halo effects. However, this filter suffers from long running time, on the order of a minute per megapixel with a single thread. Hence, it requires a parallel implementation. While recently a faster implementation became available [29], the temporal stability aspect is still an open question, and it is not clear how to extend the method to the temporal domain.

Local Laplacian filters define the output image O by constructing its Laplacian pyramid $L[O]$ coefficient by coefficient. The computation of each coefficient is independent of the others. To estimate $L[O](x, y)$, the Laplacian coefficient at level l and position (x, y) , they first apply a simple pixel-wise filter to the input image then compute a Laplacian pyramid of this transformed image, and finally use the (l, x, y) coefficient in that pyramid as the value of the output coefficient $L[O](x, y)$. First, they process input image I with a point-wise nonlinearity $r(\hat{u})$ that depends on $g = G[I](x, y)$, the coefficient of the Gaussian pyramid at level l and position (x, y) . Intuitively, $r(I)$ is an image that looks like the desired result where the intensity I is close to g . For instance, to increase the amount of detail, we apply a local S-shaped

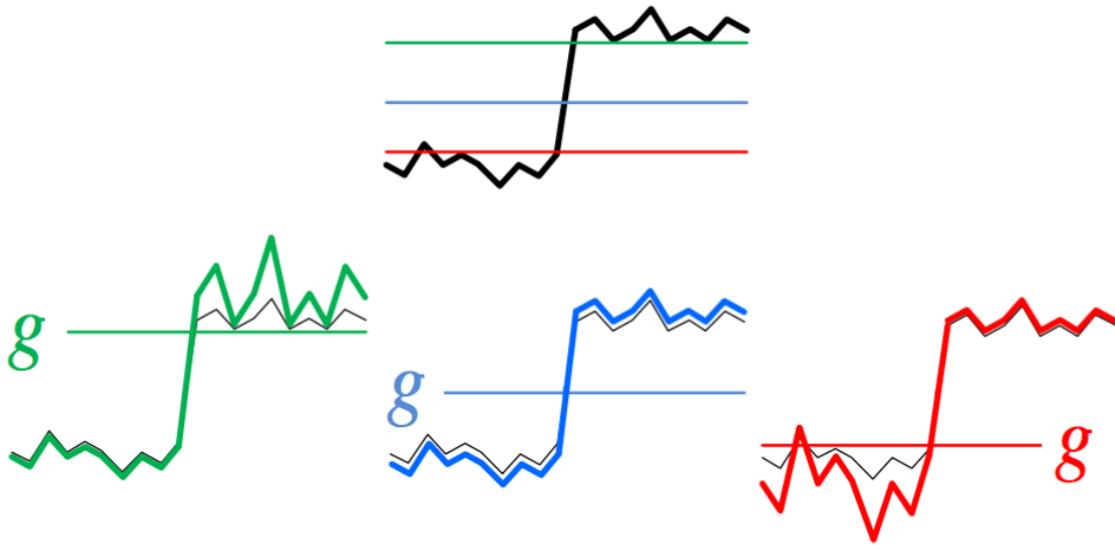


Figure 2-5: Effect of a detail enhancement ($0 \leq \alpha < 1$) remapping function r with several reference values g near an edge. Details are enhanced for values similar to the reference value, but not for values far from it. adapted from [29].

tone curve ⁷ centered on g which makes I values close to g farther away from it, and leaves more distant values unchanged. By combining the results from various g values, they obtain the final output. Then, given $r(I)$ for a particular g value corresponding to the position (x, y) and scale h they build the Laplacian pyramid of that transformed image, that is $L[r(I)]$.

2.5 HDR Video Tone Mapping

Tone mapping is also necessary for HDR video. This field has received rising interest over the last years, considerably as HDR video content is gaining popularity [31]–[34]. The reason for the absence of much work in the field of HDR video tone mapping was the insufficient HDR video data content available to research community. However, by the advent of high-end cameras capable of capturing HDR videos, this matter has

⁷The Tone Curve represents all the tones of your image. The bottom axis of the Tone Curve is the Tone axis: the line starts with Shadows at the left-most end and ends with Highlights at the rightmost end. The middle is the Mid-tones. The Y axis represents lightness of the tones. Darker tones are at the bottom of the Y axis; tones become brighter as you move up the axis.



Figure 2-6: A comparison between edge aware filters. Top left image is the input image before any filters, image on (top right) is Bilateral filtered version, image on (bottom left) is weighted least squares filtered version, image on (bottom right) is local laplacian filtered version[30].

been addressed[35].

Some of the first extensions to video apply TMOs image wise, and including a temporal component enabling them to process HDR video and avoid flickering [19], [31], [36]. Among some global HDR video operators are [19], [37]–[41] that produce results with good temporal coherence but low spatial contrast. There are also local operators like **benoit2009spatio**; **reinhard2012calibrated**, [42], [43] that keep a high contrast but produce more temporal artifacts. One can refer to Eilertsen et al.[44] for comprehensive information on these operators.

Some other works [31], [35], [36] focus on extending the TMO's tone curve [5] with temporal filtering to estimate a temporally coherent key value. In these TMOs the temporal coherence can easily be improved by ensuring the temporal smoothness in the TMO parameter space [45]. But similar other global TMOs, these methods are essentially limited to local contrast reproduction.

In work by Boitard et al. [46], they segment each video frame to two-four parts and apply a global tone curve to each segment separately. By doing this, they achieve local adaptation at the cost of more complicated processing on video. They claim to have achieved local brightness coherency based on their study that measures subjective preference rather than evaluating temporal artifacts and local contrast generation.

2.5.1 Local HDR Video Tone Mapping

While using HDR tone mapping, there is unavoidably some scene contrast loss as the dynamic range of input gets decreased significantly. As a result, there is always a compromise between having a superior or inferior contrast in the scene. If one opts out for a coarse scale contrast; as a result, the luminance difference between large image regions, as well as highlights and shadows become more obvious at the cost of the visibility of the fine scale details. On the other hand having a better contrast result in "flattening" effect.

There is also the trade-off between spatial and temporal contrast. The former is when we use the dynamic range of each frame separately, such that each frame has sufficient brightness to generate most of the scene details. It may come with the cost of losing the perception of a scene change from a brighter space to a darker one. The latter will result in a certain amount of the temporal contrast at the cost of less visible spatial details during such transitions from brighter to darker scenes. Figure 2-6 shows these two visual trade-offs when using either of these methods.

The trade-offs mentioned above are context dependent and ultimately artistic decisions, and no tone mapping approach is inherently better than others in all possible situations. As a result, it is more desirable if the tone-mapping operator enables the user to choose spatial contrast as well as temporal contrast. But the most impor-

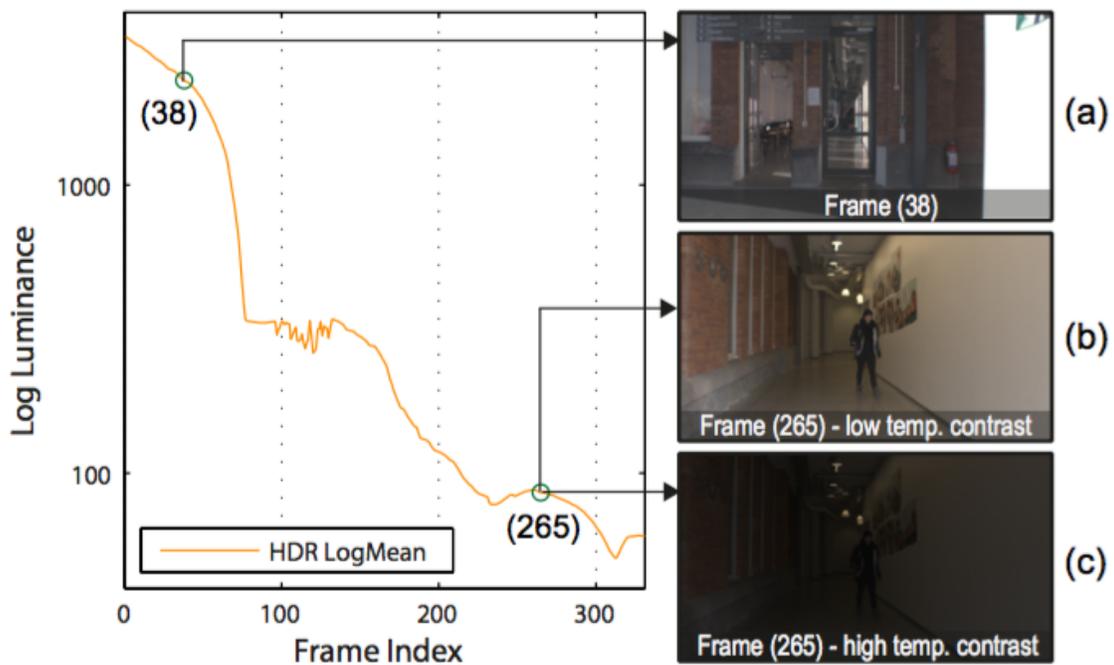


Figure 2-7: The visual trade-off between emphasizing spatial contrast (a, b) and temporal contrast (a, c). While in both settings frame 38 remains the same (a), frame 265 can be adjusted to either maintain spatial (b) or temporal contrast (c). Figure adapted from[3]

tant factor is that the method should keep some level of quality that avoids visually noticeable artifacts. Therefore, video tone mapping is especially challenging because the temporal dimension emerges as a new source of artifacts. In fact, high temporal frequency artifacts such as brightness flickering, camera noise, as well as any temporal inconsistencies of the TMOs are immediately noticeable because of the human visual system's properties. Even minimal amounts of ghosting and flickering in the tone mapped HDR videos are unacceptable in practice [44].

2.6 Clustering

Color clustering is an important process of representing true colors of an image using a smaller palette of colors that could be displayed on the screen. This idea was prevalent during the 1980s and 1990s when the displays had low dynamic range and colors could not be represented by higher bits. As a result, they had to map a 24-bit depth per pixel, meaning 256 discrete intensity levels for each color channel of the image to a smaller number of colors in a color map.

Color clustering usually consists of four stages. The first step is sampling the original image and computing the image histogram for obtaining color statistics meaning the number of distinct colors and their frequencies. The second phase is designing the colormap, which is choosing the best possible set of colors to represent the color statistics. The third step maps each color in the original image to a representative color in the colormap. The fourth phase replaces the original color with a representative color.

Natural images are composed of a vast number of distinct colors. Hence, the quantizers could choose a set of K representative colors (8- 256) from input images in such a way that a difference between reproduced K -color images and original N -color images are as little as possible. Mean square error (MSE) is normally used to represent the difference.

The clustering-based algorithms perform clustering the color space into K -desired clusters. The methods involve an initial selection of colormap followed by repeatedly

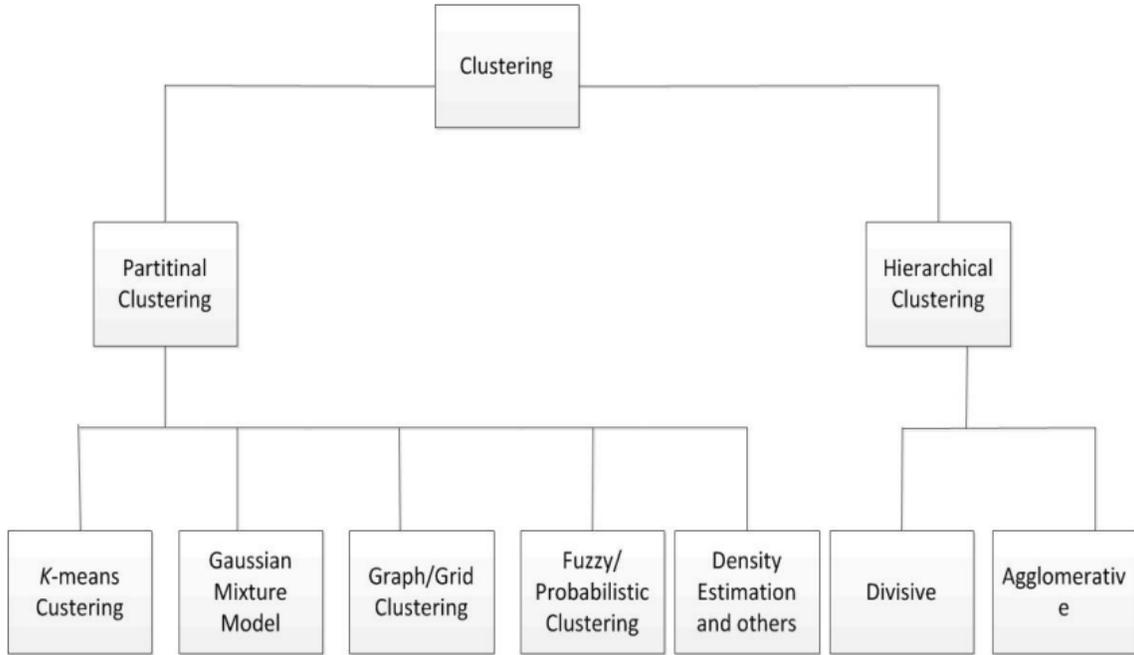


Figure 2-8: Classifications of Clustering Methods

updating cluster representatives.

2.6.1 The C-Means clustering Algorithm (CMA)

A classical approach to cluster input pixels is the (K-means) clustering algorithm (CMA) [47]–[50]. The KM algorithm is inarguably one of the most widely used methods for data clustering [48]. Given a dataset $X = \{x_1, x_2, x_3, \dots, x_N\} \in R^D$, the objective of KM is to partition X into K exhaustive and mutually exclusive clusters $S = \{S_1, S_2, S_3, \dots, S_N\}$, $\bigcup_{k=1}^K S_k = X$ $S_i \cap S_j = \emptyset$ for $1 \leq i \neq j \leq k$ by minimizing the sum of squared error (SSE):

$$SSE = \sum_{k=1}^K \sum_{x_i \in S_k} \|x_i - c_k\|_2^2$$

where $\|\cdot\|_2$ denotes the Euclidean (L2) norm and c_k is the center of cluster S_k calculated as the mean of the points that belong to this cluster. This problem is known to be NP-hard even for $K = 2$ [51] or $D = 2$ [52], but a heuristic method

developed by Lloyd [53] offers a simple solution. Lloyd’s algorithm starts with K arbitrary centers, typically chosen uniformly at random from the data points [54]. Each point is then assigned to the nearest center, and each center is recalculated as the mean of all points assigned to it. These two steps are repeated until a predefined termination criterion is met. The pseudocode for this procedure is given in Algo. (1) (bold symbols denote vectors). Here, $m[i]$ denotes the membership of point x_i , i.e. index of the cluster center that is nearest to x_i .

Convventional K-means Algorithm.

input: $X = \{x_1, x_2, \dots, x_N\} \in R^D$ ($N \times D$ input data set)

output: $C = \{c_1, c_2, \dots, c_N\} \in R^D$ (K cluster centers)

Select a random subset of C of X as the initial set of cluster centers;

While: termination criterion is not met **do**

for ($i = 1; i \leq N; i = i + 1$) **do**

Assign x_i **to the nearest cluster;**

$$m[i] = \operatorname{argmin}_{k \in \{1, 2, \dots, K\}} \|x_i - c_k\|^2$$

end

Recalculate cluster centers;

for ($k = 1; k \leq K; k = k + 1$) **do**

Cluster S_k **contains the set of points** x_i **that are nearest to the center** c_k :

$$S_k = \{x_i | m[i] = k\}$$

Calculate the new center c_k **as the mean of the points that belong to** S_k

$$c_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} x_i ;$$

end

end

The complexity of KM is $O(NK)$ per iteration for a fixed D value. For example, in color quantization applications $D = 3$ since the clustering procedure is often performed in three dimensional color spaces such as RGB or CIEL*a*b* [52]. The KM algorithm has the advantage of having a linear time complexity. It is also guaranteed to terminate with a

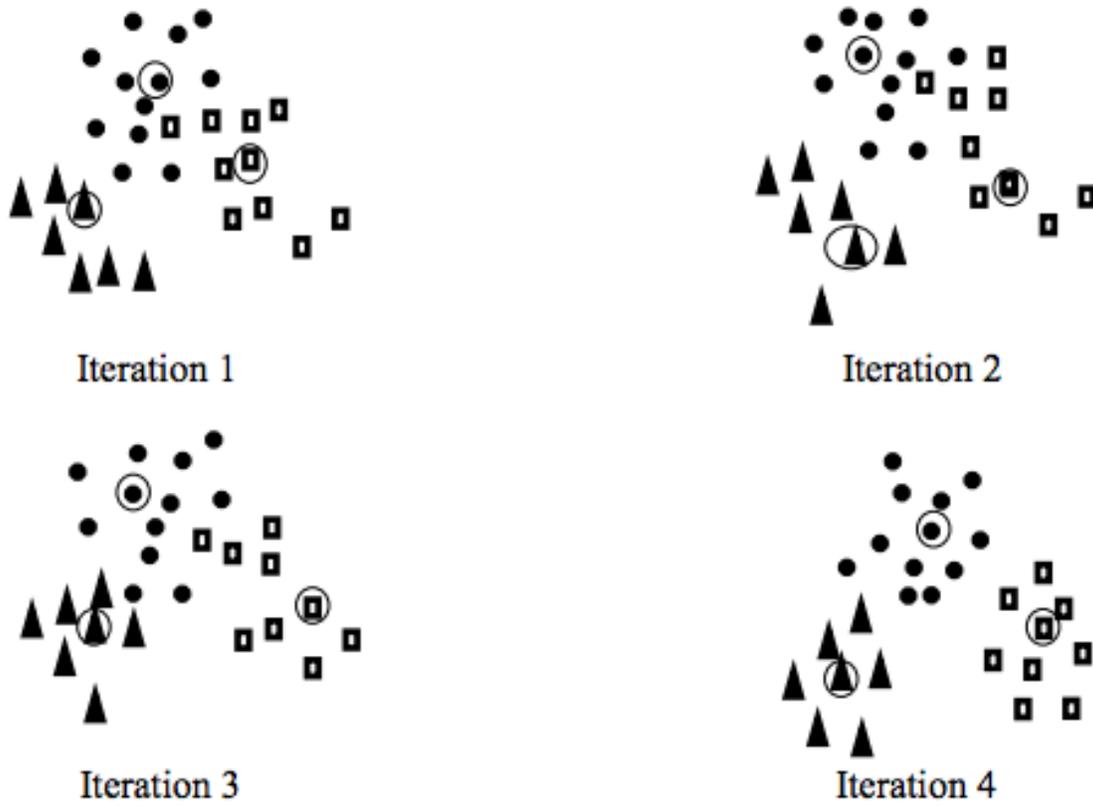


Figure 2-9: K-means Example. Adapted from wikipedia.

quadratic convergence rate [55]. The main disadvantages of KM are that it often stops at a local minimum[59] and that its results dependent on the initial choice of the cluster centers.

2.6.2 Hierarchical Clustering Algorithms

Hierarchical clustering techniques are the second most important clustering methods. Just like K-means clustering, these methods are relatively old compared to many clustering algorithms but are still used widely. There are two essential approached to create hierarchical clustering:

Agglomerative: Starts with each point as individual clusters and merges the closest pair of clusters at each step. This requires of defining the notion of cluster proximity. following is the basic steps for an agglomerative clustering algorithm.

Basic agglomerative hierarchical clustering algorithm.

- 1: Compute the proximity matrix, if necessary.
 - 2: **repeat**
 - 3: Merge the closest two clusters
 - 4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
 - 5: **until** Only one cluster remains
-

There are three different ways to complete step 3, by using single-linkage, complete-linkage or average-linkage proximity matrix. In single-linkage clustering (also called the connectedness or minimum method), the distance between clusters is equal to the shortest distance between any of their members. If there exists a similarity between distances, the similarity between clusters is considered to be the greatest similarity from any members of one cluster to any member of the other cluster. In complete-linkage clustering (also called the diameter or maximum method), the distance between two clusters is considered to be the greatest distance between any of their members. In average-linkage clustering, the distance between clusters is equal to the average distance between members of clusters.

Divisive: Starts with one all-inclusive cluster, and at each step splits a cluster until

only singleton clusters of individual points remain. At each point, we need to decide which cluster to split and how to split it. Divisive methods are not available and rarely have been applied.

The main weaknesses of agglomerative clustering methods are that they do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects. They also can never undo what was done previously.

2.6.3 Fuzzy C-Means Clustering

Fuzzy c -means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method was first developed by Dunn in 1973 [56] and improved by Bezdek in 1981 [57] is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension center of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center.

2.7 Chapter Summary

This chapter carried out a comprehensive literature review on various tone-mapping operators on HDR image and video. Some of the most well-known edge aware filters used for HDR tone-mapping were also reviewed. We also examined the Bilateral filter in depth which is used in our proposed method. This chapter, also, provides the required knowledge needed about clustering methods including K-means clustering. As a result, this chapter provides highly relevant information to our study in later chapters. From the next chapter, we will focus on constructing our new tone mapping operator using K-means clustering.

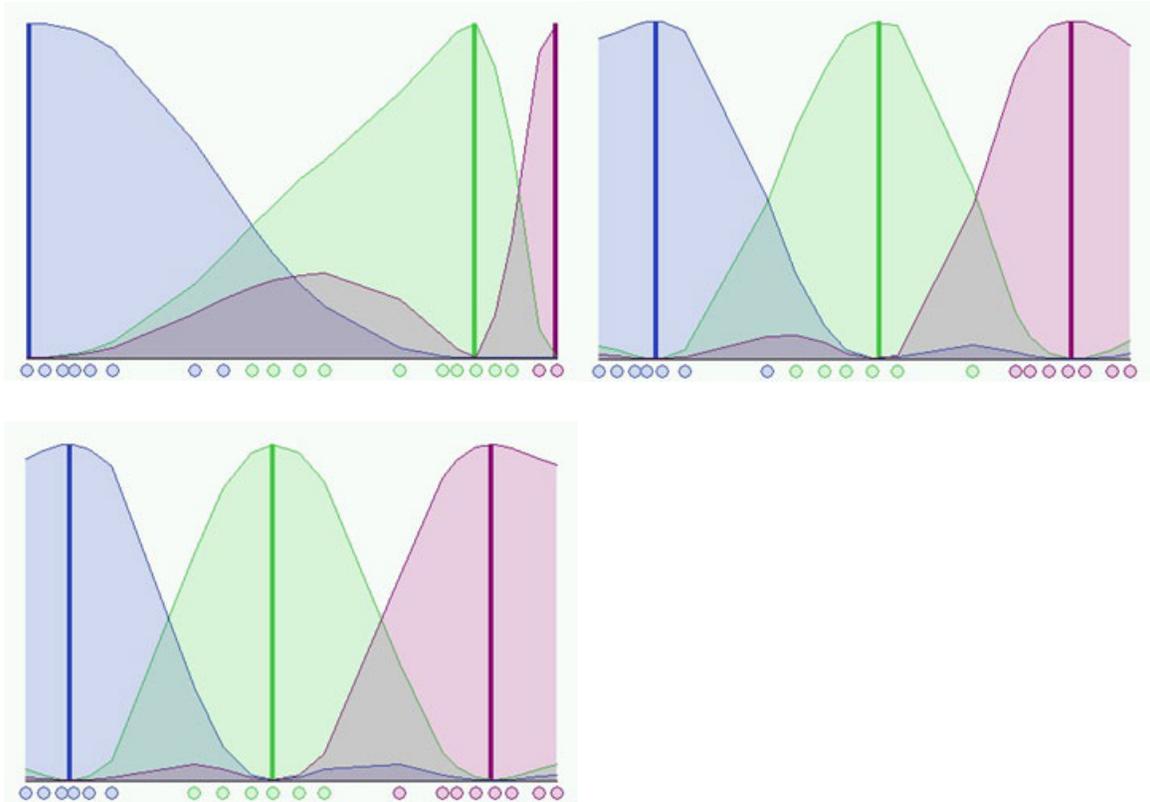


Figure 2-10: Top image shows data before using Fuzzy C-means clustering, top right shows clustered data after 8 iterations, bottom left image shows clustered after 37 iterations.

Chapter 3

Proposed Algorithm

3.1 Introduction

The proposed tone mapping algorithm constructs the tone mapped frames using their HDR versions as input. The proposed local TM algorithm segments an image into a number of local regions according to the luminance of initial global mapping. Our algorithm consists of the following steps:

- 1- Finding the number of clusters used in K- means.
- 2- Calculating the intensity channel.
- 3- Taking the logarithm of intensity channel.
- 4- Performing K-means algorithm.
- 5- Adjusting the color based on K-means results.
- 6- Applying a Gaussian kernel for smoother local factors.

3.1.1 Finding the number of clusters used in K-means

One of the difficulties of using k-means is choosing the number of clusters, K . The basic K-means is an incredibly straightforward and efficient algorithm. However, to determine the proper number of K one has to have prior knowledge of data. The correct choice is usually vague, and the solutions depend on shape and distribution of data and the clustering resolution needed based on the application. The number of K should strike a balance between maximum accuracy (i.e., assigning each data point to its own cluster) or maximum

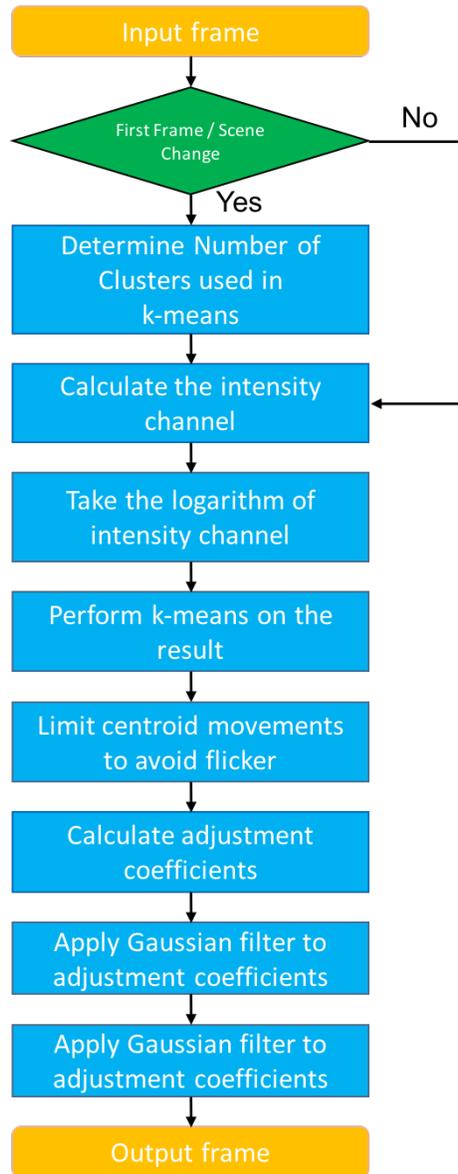


Figure 3-1: Flow chart of our proposed method

compression (i.e., using a single cluster for all data points). It is best to choose the appropriate value of K based on prior knowledge of data. Otherwise, there are several categories of methods for making this decision. One way of selecting K is by examining gap statistic[58].

The Gap statistic is a standard method for determining the number of clusters in a set of data. The idea is to compare the within cluster dispersion to its expectation under an appropriate null reference distribution [58], i.e., a distribution with no apparent clustering. It standardizes the graph of $l \log(W_k)$, where W_k is the within-cluster dispersion and is defined as

$$W_k = \sum_{k=1}^K \frac{1}{2n_k} D_k$$

where D_k is

$$D_k = \sum_{i,i' \in C_r} d_{ii'}$$

$d_{ii'}$ is the distance between samples i and i' within cluster C_r . This distance in our case is measured by squared Euclidean distance that is $d_{ii'} = \sum_j (x_{ii} - x_{i'j})^2$. W_k decreases monotonically as the number of clusters k increases. For the calculation of the Gap function, Tibshirani et al. (2001) proposed to use the difference of the expected value of $\log(W_k^*)$ of an appropriate null reference and the $\log(W_k)$ of the data set,

$$Gap_n(k) = E_n^* \{ \log W_k \} - \log W_k$$

Then, the proper number of clusters for the given data set is the smallest k such that

$$Gap_n(k) \geq Gap_n(k+1) - s_{k+1}$$

where s_k is the simulation error calculated from the standard deviation $sd(k)$ of B Monte Carlo replicates $\log(W_k^*)$ according to the equation $s_k = \sqrt{1 + 1/B} sd(k)$.

The data set we used in Gap statistic is the probability density estimate of 'a*b*' channels of the frame in the CIELAB color space¹. We use 'a*b*' channels since the color information exists in the 'a*b*' space. Therefore, we convert the frame from RGB color space to CIELAB color space. We use the recommended number by Gap statistic as the number of clusters used in K-means algorithm. Since CIELAB color space is a very close representation of

¹The Lab color space describes mathematically all perceivable colors in the three dimensions L for lightness and a and b for the color opponents green-red and blue-yellow[wikipedia].

human perception, thus the information in 'a' and 'b' components can be used to make accurate color predictions.

The reason behind choosing Gap statistic rather than opting out for a k-means implementation that automatically detects the number of Ks is that the data that we are clustering is the intensity channel of an image. Whereas, we determine the number of clusters by the information obtained from the 'a*b*' space. As a result, we see fit to use methods such as gap statistic to find the number of clusters before using K-means. Any other method that determines the number of clusters in a data set can also be utilized in the following context.

For the implementation of Gap statistic, we used Matlab build in function "eval clusters." This function creates a clustering evaluation object containing data used to evaluate the optimal number of data clusters. This function takes in the list of the number of clusters to evaluate. Since Gap statistics is a relatively slow algorithm, we limit our list to the numbers {8,16, 32, 64, 128, 256, 512}. The reason behind it being slow is that it generates N samples from a uniform distribution over the same range as the original data, B times. In each b = {1, 2, . . . , B} iteration, it then runs the same clustering algorithm with your candidate k values. This means that on top of clustering original data K times you have to cluster KB data sets, each of sample size N. When p and N are not significant, it's pretty tractable. However, it can grow out of control rather quickly.

3.1.2 Calculating the intensity channel (Dimensionality Reduction)

The frames of videos that we are working with are all in RGB color space. Other color spaces can be used here as well. There are various ways to approach working with color images, including working with different color spaces. Our goal is to estimate the image contrast, there are many ways to quantify the contrast of an image. One way is to use the root mean square (RMS) of the image. This approach has been found to match with human perception of image contrast. RMS contrast is defined as the standard deviation of the image pixel intensities,

$$RMS = \sqrt{\frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M I_{ij(R)}^2 + I_{ij(G)}^2 + I_{ij(B)}^2}$$

where I_{ij} is intensity at (i,j) pixel of the image with size M by N , and the subscripts represent the Red, Green and Blue channel. We then use the RMS contrast as input to K-means clustering algorithm. The intensity channel can also be represented by maximum of the three channels or as their weighted sum. Since in our approach we are trying to adjust the image contrast we found that our final result using RMS estimation of image are better.

3.1.3 Taking the log of intensity channel

Typically, image processing operations are performed using real based algebra, which proves its limitations under particular circumstances, like upper range overflow. For avoiding such situations, nonlinear techniques have been developed. Such examples are the LIP (logarithmic image processing) models. LIP models are commonly used in many image processing applications due to their rigorous mathematical properties and similarities with the human visual system. As a result, we take the logarithm of intensity channel to reduce the dynamic range of intensity channel. Since logarithm is not defined at 0, we add a epsilon to the intensity channel and then take the logarithm. The epsilon added depends on the brightness of the frame being processed. Since we do not want too much contrast, for each frame, we add a fraction of the mean pixel value of the context to the logarithm. We choose a tiny fracture that varies based on the brightness of the frame. This additive value will allow the user to have visual feedback at interactive rates while the additive parameter is being changed.

3.1.4 Performing K-means algorithm

By performing K-means algorithm we want to solve the following problem.

$$\min_{c_1, c_2, \dots, c_k} \sum_{i=0}^N \sum_{j=0}^M x_{ij} \| p_i - c_j \|^2$$

c_j is the centroid of cluster j

$$x_{ij} \in \{0, 1\} \forall i, j$$

The binary variable x_{ij} indicates whether or not point i is assigned to cluster j . Symbols p_i and c_j denote the coordinates of i th point and centroid of j th cluster, respectively. They are both located in \mathbb{R}^d , where d is the dimensionality of data points. The initial centroid

values are chosen randomly from the intensity values of the frame. To cluster, intensity values get mapped to one of the K initial centroid depending on their Euclidean distance from them. This way each intensity value gets assigned to a group. After all values to a group, the center value of each cluster gets updated with the mean of that cluster. This step gets repeated depending on the number of iterations of K- means. Through each iteration, the intensity values assigned to the center of each cluster gets updated as follows:

$$c_{\{1,2,\dots,k\}} = \frac{\sum_{i=1}^n x_{ij} p_{ij}}{\sum_{i=1}^n x_{ij}}$$

3.1.5 Adjusting the color based on K-means results

Now that we have labeled the image pixels using the results of K-means algorithm, we try to adjust the intensity of each group consistently. We divide the intensity spectrum [0,1] into K section:

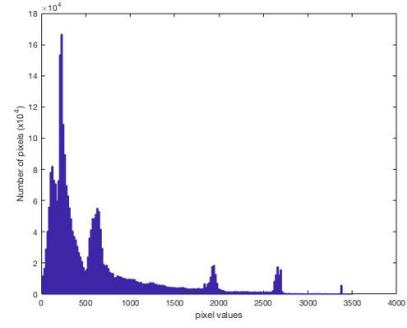
$$\left[0, \frac{1}{k}\right] \left[\frac{1}{k}, \frac{2}{k}\right], \dots, \left[\frac{k-1}{k}, 1\right]$$

Then we map each cluster into one of the sections, by sorting them based on their centroids we calculated using K-means algorithm. We assign the first section of intensity spectrum (i.e. [0,1/K]) to the centroid with the smallest value, the second section to centroid with second lowest value and so on until all centroids get assigned to an intensity section based on their value. Now we have K clusters and K sections of intensity spectrum. Since we want to keep the shape of intensity spectrum for each cluster and as centroids represent the mean of each cluster we need to calculate a scaling factor for pixels of every cluster so that the centroid of that cluster gets mapped to the center of the corresponding spectrum section. For example, for the first cluster, let's assume the centroid is c_1 , and the corresponding spectrum section is [0,1/K]. The center of this spectrum section is $1/(2 \times K)$. Therefore, we multiply all pixels in this cluster to $1/(2 \times K \times c_1)$ factor. Figure 1 illustrates the steps explained in previous sections for a frame with the value of $K = 15$.

As it is shown in Figure 3-2 each color cluster shifts based on the adjustment coefficient we calculate for it. This adjustment results in a tone mapped frame.



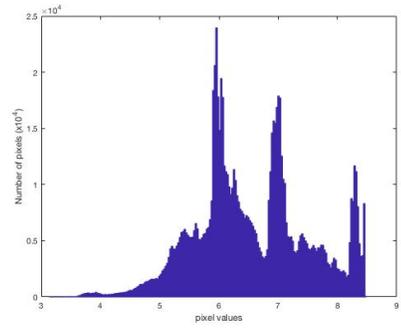
(a)



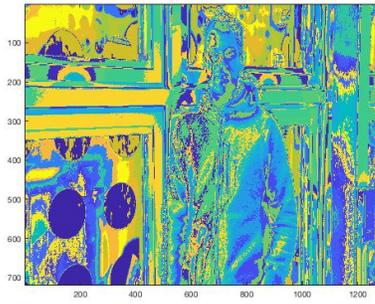
(b)



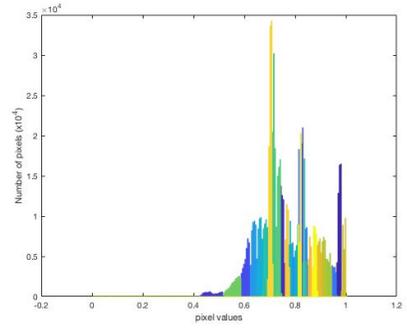
(c)



(d)



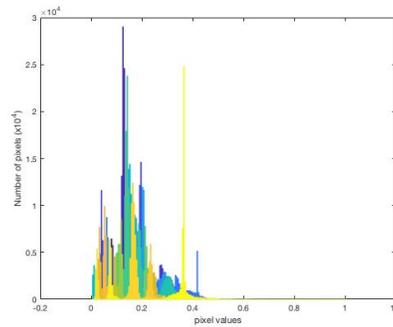
(e)



(f)



(g)



(h)

Figure 3-2: (a,b) a frame of a video before tone mapping with its histogram (c,d) intensity channel of frame and its histogram (e,f) frame color regions based on K-means (g,h) Tone mapped frame and its histogram.

3.1.6 Applying Gaussian filter for smoother local factors

We apply a Gaussian filter on the calculated adjustment coefficients to smooth out the transition between image region boundaries. This step reduces the artifacts caused by localizing areas with different colors. The sigma used in this process is based on the frame size. We use $1/1200$ of the larger side of the frame as our sigma value. To reduce the artifacts that can result from this process, we choose a small sigma value for Gaussian kernel. Otherwise, bigger sigma value will result in halo artifacts around boundaries of different regions of the frame. Figure 3-3 Shows the difference between not using a Gaussian filter and using it with two different sigma value. As depicted in Fig 1. (a) not using a smoothing filter causes lines separating various color regions to appear in the image. Additionally, as shown in Figure 3-3 using a larger sigma results in the bright boundaries in some areas of the picture. Therefore, as discussed, we need to lessen the sigma value to achieve a smoother looking image. The bilateral filter can also be used for this purpose. We chose Gaussian since it is much faster than Bilateral filter and results are essentially the same.

3.2 Video Processing

For video processing, our algorithm goes through the following steps in addition to the one explained earlier:

- 1- Determining the first frame of each scene.
- 2- Video Flicker removal.

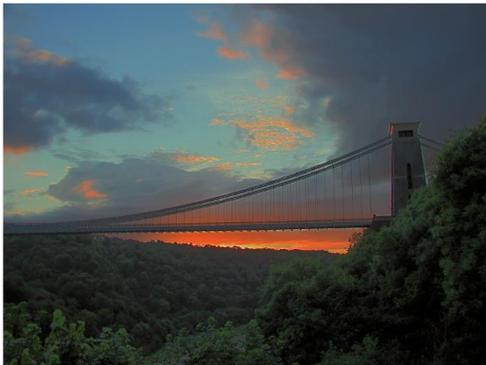
Our algorithm would go through all the steps explained in section 3.1 if the frame processed is the first frame of a new scene. Otherwise, in order to reduce the running time of our algorithm for the frames following the first in each scene, we use the number of Ks computed in the first frame as the number of clusters used for the next frames. Also for finding the pixels attributed to each cluster, we will not go through the same number of iterations in K-means algorithm as the first frame. As in the frames following the first for each scene, the clusters will not change drastically. As a result, by feeding the centroid values of the previous frame as the seeds to K-means for the next frame we reduce the number of iterations used for frames afterwards.



(a)



(b)
(d)



(c)



Figure 3-3: (Top left) Original HDR image (Top right) Tone mapped version without smoothing local factors (Bottom left) Tone mapped version with Gaussian filter with sigma equal to $1/1200$ of the bigger side of the frame (i.e. 713 pixels) (Bottom right) Tone mapped version with Gaussian filter with sigma equal to $1/300$ of the bigger side of the frame (i.e. 713 pixels)

3.2.1 Determining the first frame of a scene

The first frame is chosen by its grey value distribution. We monitor this value for all the frames in the video sequence, and a significant change indicates a scene change. We used Earth Mover's Distance (EMD) to measure this change. The EMD distance d is calculated using cumulative histogram distribution of two consecutive frames. Considering H_1 and H_2 as the cumulative histogram distribution, then

$$d = \sum_1^N |H_1(i) - H_2(i)|$$

where N is the number of bins in the histograms. To quantify the difference between two distributions, we can measure how far the grains of sand have to be moved so that the two distributions coincide exactly. EMD is the minimal total ground distance traveled weighted by the amount of sand moved (called flow). EMD makes sure that shifts in sample values are not penalized excessively. Studies report improved performance when comparing various histogram descriptors with EMD over the χ^2 statistic and the L2 norm.

To decide on a threshold of which if it is passed the scene is considered to have changed, we ought to have the information of cumulative histogram distribution of all the frames. As a result, this scheme can be used in an offline setting, so that we can find out the EMD distance between all the frames, then based on the mean value of EMD distance we set a threshold for d to represent scene change. Figure 3-4 shows EMD distance between every two frames of a video with 189 frames. Here we set the limit for d to be 1.3 which is the mean value of EMD distance between the frames.

3.2.2 Video Flicker Removal

Our tone mapping system seeks to reduce flicker² in tone mapped HDR video by using a leaky integrator applied to the value of seeds used in K-means algorithm after the first frame of each scene. As mentioned before to reduce the running time we pass in the centroid value of clusters of the previous frame as the seed values to K-means for the subsequent frames.

²Flicker is a visible fading between cycles displayed on video displays.

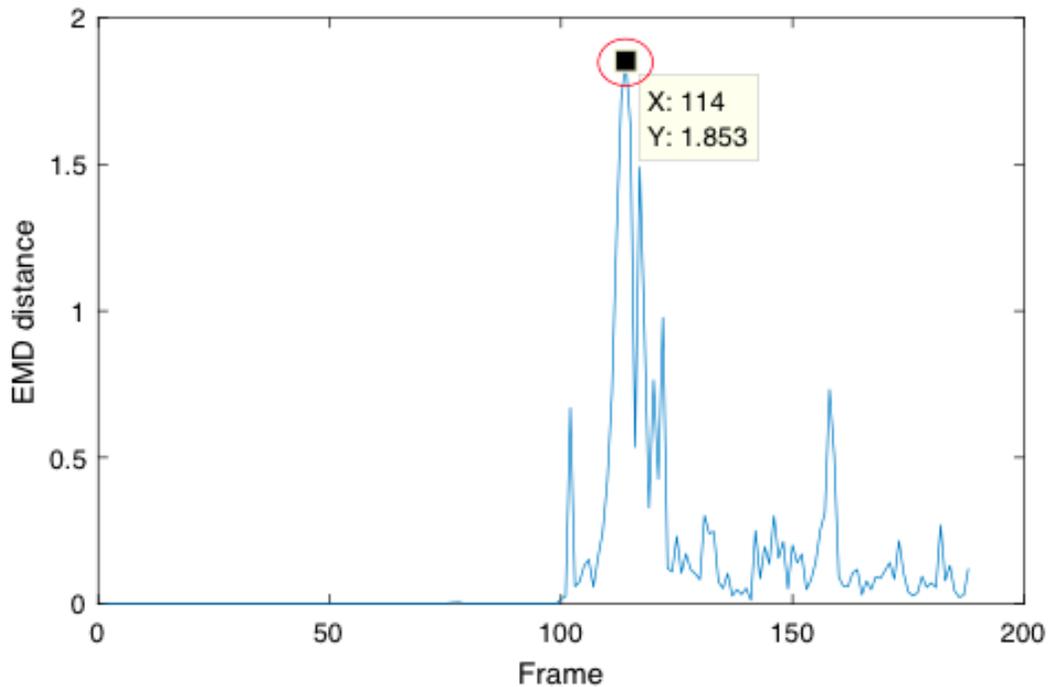


Figure 3-4: EMD distance between frames of a video with 189 frames. The red eclipse shows the maximum value of EMD which is 1.853 at the frame 114.

The update equation for centroids is as follow:

$$centroids = newcentroids \times \alpha + centroids \times (1 - \alpha)$$

The parameter α is $\in [0,1]$ and is set based on the speed of changes occurring in video frames. This method reduces the rapid frame to frame changes in the parameters of the TMO thus giving the TMO algorithm temporary memory.

3.3 Chapter Summary

This chapter detailed the steps taken in the proposed algorithm. It first covers the steps required for tone-mapping a single frame or image and next it covers additional steps required for processing video sequences. For better representation of details taken we also provide images describing each step. The algorithm was developed with MATLAB2017a. In the next chapter, we will present the results obtained using our method, and we will compare these results with the other TMO operators.

Chapter 4

Results and Analyses

This chapter presents the results obtained by the proposed method in the previous chapter. The results in this chapter were generated using the default parameters presented in chapter 3. We divide this chapter into two sections. The first section presents the results obtained for HDR images and compares them with the results of other well-known methods. The second section presents the results for HDR videos.

4.1 Results on HDR Images

We have tested our method on some HDR input images and compared the results with a number of standard tone-mapping algorithms. The images were taken from R. Mantiuk[59] and P. Debevec [60]. The Debevec's database comprises of HDR images that were used in most of the well-known HDR tone-mapping methods.

We used the HDR image tool Luminance HDR [61] to do the processing for generating other methods results. [61] is an open source graphical user interface application that aims to provide a workflow for HDR imaging. The supported HDR format images that can be used in this HDR image tool are OpenEXR (extension: `exr`), Radiance RGBE (extension: `hdr`), Tiff formats: 16bit, 32bit (float) and LogLuv (extension: `tiff`), Raw image formats (extension: various), PFS native format (extension: `pfs`). The HDR images we used in this process were all in the format of Radiance RGBE. This tool contains implementations of a number of tone mapping algorithms most of which were mentioned in chapter 2. when producing results we used the default parameter settings provided by Luminance HDR. It is

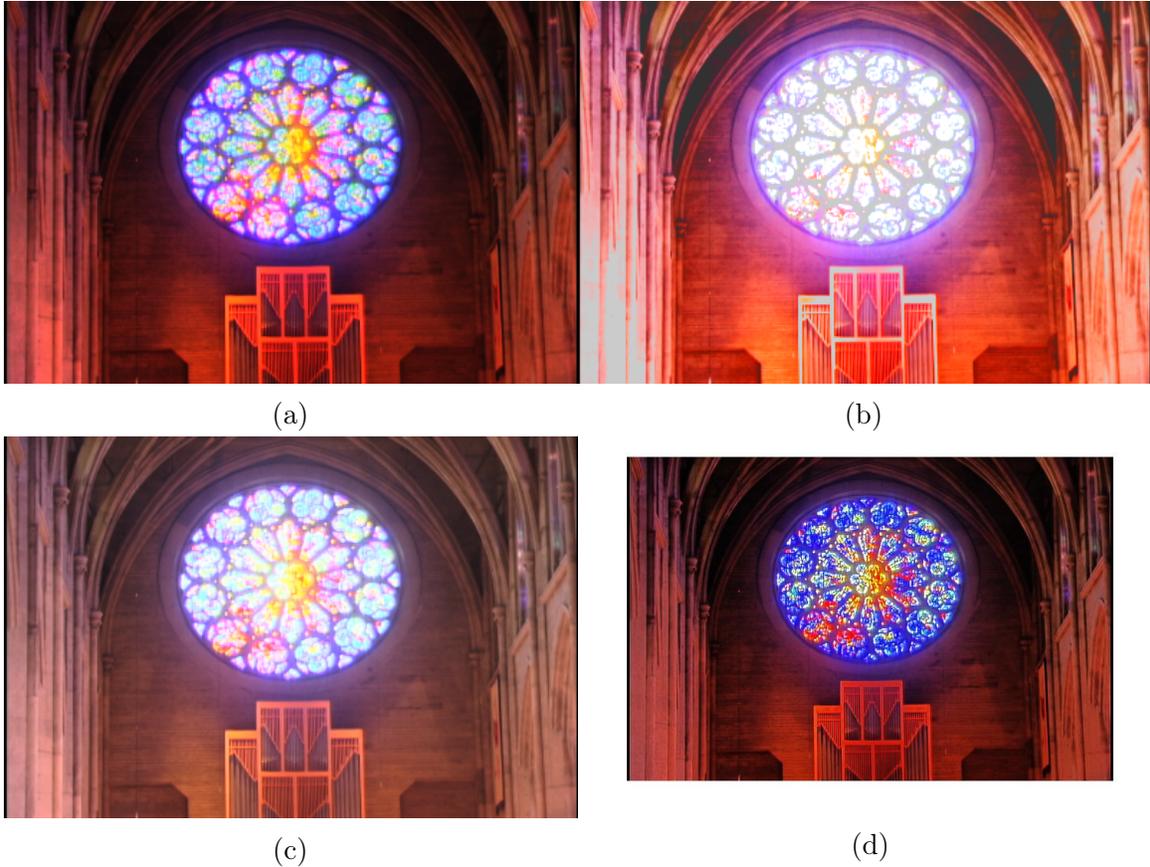


Figure 4-1: (a) image tone mapped with Drago method[14],(b) image tone mapped with Mantiuk et al. [19], (c) image tone mapped with Reinhard et al. [5], (d) image tone mapped with our proposed method. One can see that the compared methods suffer from over-saturation, color artifacts and loss of detail.

possible that in some instances better results can be obtained by tweaking the parameters manually. This fact also holds for our method; as a result, we used the default parameters mentioned in the previous chapter. We compared our method to the methods of [5], [14], [15], [19]. Of these, we found that [19] and [5] and [14] gave significantly better results over the set of test images. Our results were very much comparable to Drago and [5]. In Figure 4-1, and Figure 4-2, we show some examples of the results obtained by other methods and our suggested method. We can see that the compared methods display problems with over saturation, loss of detail resolution and color artifacts. In Figure 4-3, the output of our algorithm is shown for a number of different input HDR images.

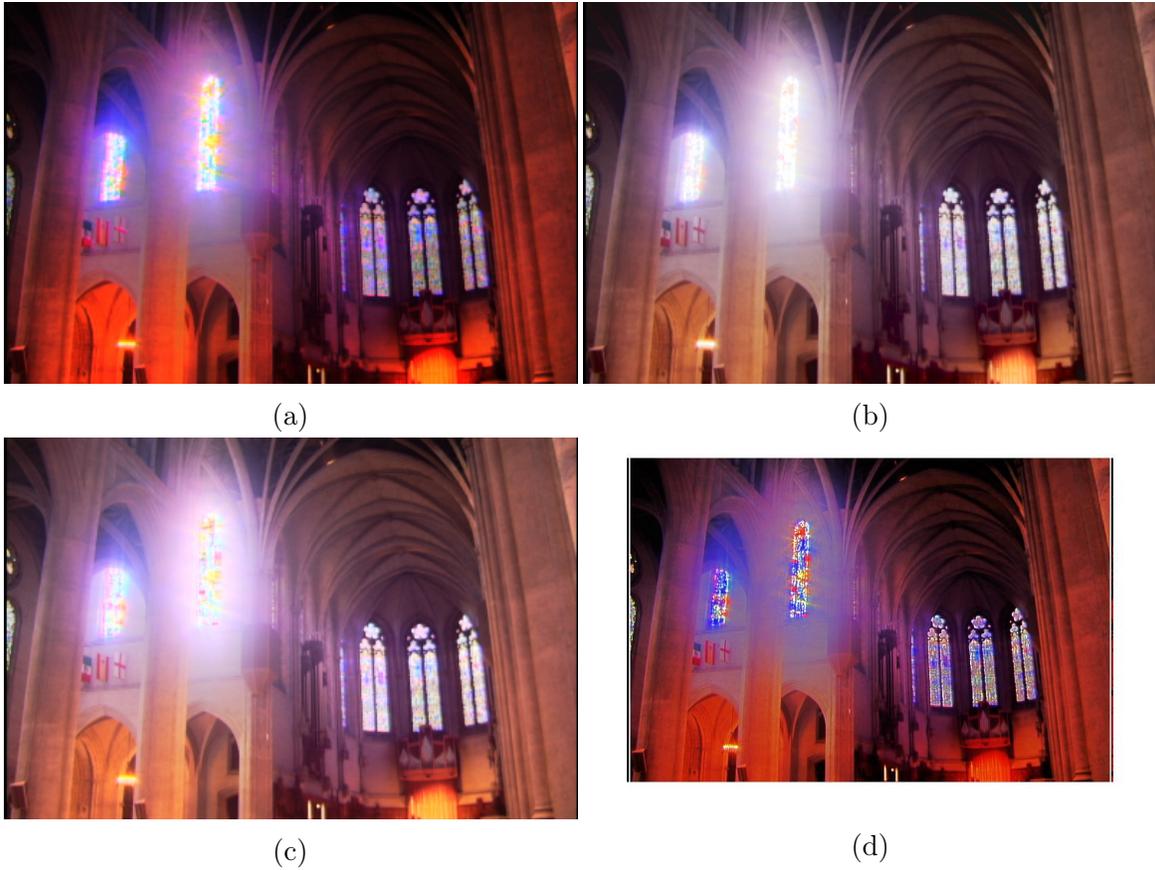


Figure 4-2: (a) image tone mapped with Drago method [14],(b) image tone mapped with Mantiuk et al. [19], (c) image tone mapped with Reinhard et al. [5], (d) image tone mapped with our proposed method. One can see that the compared methods suffer from over-saturation, color artifacts and loss of detail.



(a)



(b)



(c)



(d)



(e)

Figure 4-3: The result of running our Algorithm on a number of HDR images. HDR radiance maps courtesy of Debevec [60] and Mantiuk [59]

4.2 Results on HDR Video

4.2.1 Comparison With Other Methods

We tested our algorithm on a several test HDR videos obtained from [32], [33], [62]. These test videos were also used in other frameworks like in the evaluation in of tone mapped HDR video by [44]. An overview of the sequences can be found Table 4.1. The table shows the processing time for the tone mapping. As mentioned earlier our algorithm time complexity is $O(N \log K + K \log K)$. In the remainder of this section, we qualitatively compare our method's results on the publicly available HDR video data provided by [44]. Figure 4-4, displays the results obtained by our method on some frames of these HDR video sequences.

Table 4.1: The tested HDR sequences from [62].

Sequence	Resolution	Frames	time(s)
Window	720p	236	657
Hallway	720p	331	2289
Hallway 2	720p	351	1130
Students	720p	251	3242
Exhibition	720p	189	928

Figure 4-5, represents a comparison of the result of different methods for processing HDR videos. The result of all the mentioned methods are from the publicly available videos in [32], [44], [62]. The image is taken from [3] and the method (e) referred to as "Our method" represents their proposed method. Figure 4-5, also includes a plot of the mean brightness of HDR and tone mapped sequence for each method. For each method, only two frames of each sequence is presented since providing all frames is impossible. These frames display each method's ability of creating local contrast.

Local methods such as Retina Model [63] (a), Color Appearance [64] (b) and Virtual Exposures [43] (c) exhibit flickering brightness. This can be detected from their mean intensity plot. The Local Adaptation [42] (d) operator exhibits ghosting artifacts (See the wall at the bottom image in Figure 4-5,-d) since this method does not utilize motion paths. Like with the local image TMOs that create temporal artifacts [44], Bilateral TMO [15] mean intensity plot shows fluctuations over time. The average brightness plots of the remaining operators Temporal Coherence [41], Display Adaptive [19] and Camera TMO [44] display



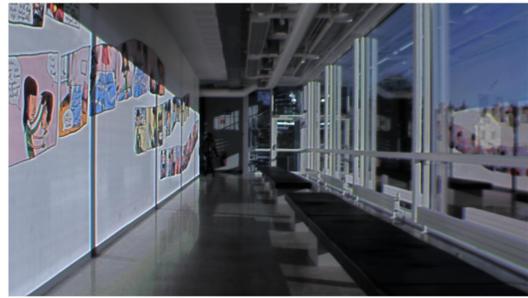
(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figure 4-4: The figure shows two frames (left and right, respectively) of the output from our method, (from top to bottom) the exhibition sequence, the hallway2 sequence, the students sequence, the window sequence. The results are best viewed on screen

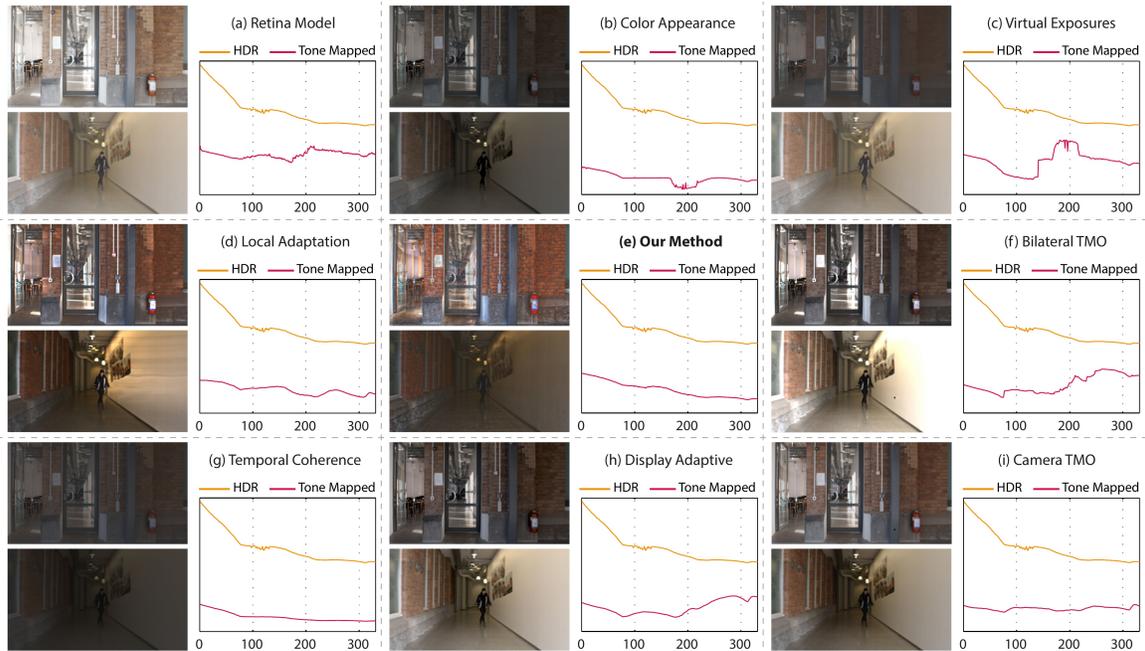


Figure 4-5: Comparison different current TMO methods capable of processing videos. For each method, two representative frames and plot of mean pixel values (tone mapped) and log-mean luminance (HDR) over time. Note that the HDR plots are shifted and compressed by an exponent for presentation. The image is taken from [3] the method (e) referred to as 'Our method' represents their proposed method.

less temporal artifacts. Their disadvantage is that they don't exhibit much local contrast as a result of their spatially-invariant processing.

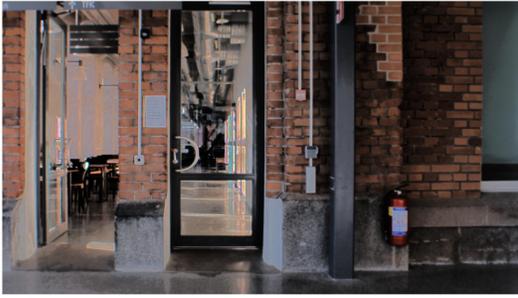
On the other hand, we configured our TMO represented in Figure 4-6b to improve local spatial contrast meanwhile that it also maintains the temporal contrast between the beginning and the end of the sequence. Our tone mapping result remains temporally coherent. Also as shown in Figure 4-6c our method does not exhibit flickering. Moreover, due to local filtering of some of the compared methods, they exhibit stronger local contrast than our method. Aforementioned at times can lead to artificial detail and cartoonish effects.

Figure 4-7, compares our results with the Boitard et al. [46] segmentation based temporal coherency method applied to Ramsey et al. [36] and the recursive domain transform filter [Gastal and Oliveira 2011] [27]. Also Aydin et al. [3] coherent temporal tone-mapping for HDR videos. As depicted Boitard et al.'s [2014b] method tends to underutilize the available display dynamic range and generates dark frames. The mean pixel value plots show fluctuations (notably between frames 100 and 130) even for the Ramsey et al. [36] version which is a global TMO. Our method also shows some fluctuations. Our result preserves the change from the bright exterior to the darker hallway by modulating the mean luminance accordingly, whereas this transition is diminished or reversed in Boitard et al.'s [2014b] [46] results. Aydin et al. [3] give in some areas slightly higher local contrast which in some cases results in the animated effect.

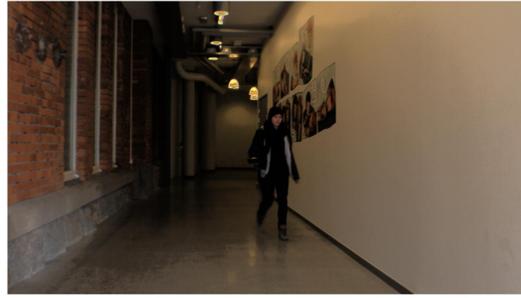
4.2.2 Subjective Study

While the qualitative comparison in the previous section is used as an overview and helps to put the many TMOs into perspective, it certainly does not capture all the aspects relevant to the evaluation of video tone mapping. For example, most operators tend to show camera noise in their results. As a result, we performed a subjective rating study to compare our method with others regarding their seeming quality.

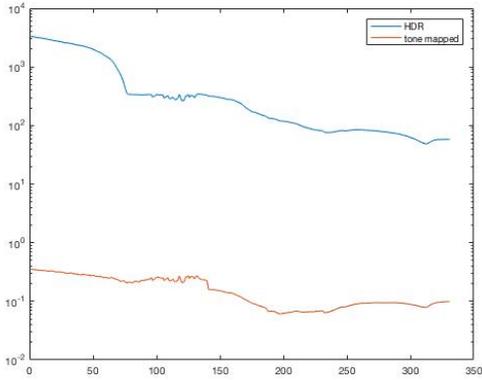
The study was conducted through Amazon Mechanical Turk. For the study, we used 5 HDR videos sequences each tone-mapped with twelve different tone-mapping operators. The tone mapping operators utilized for this study are listed in Table 4.2. We asked our subjects to rate each video on a scale of 0 to 10. With 0 representing worst quality and 10 representing the best quality. They were told to rate by how closely the displayed video clip



(a)

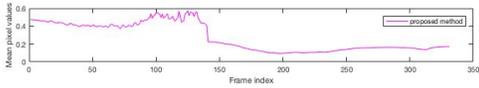


(b)

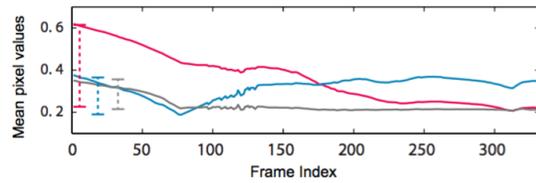


(c)

Figure 4-6: Frame 80 (top left) and frame 265 (top right) from Hallway video sequence. (bottom left) Mean intensity value of each frame in Hallway video sequence before and after being tone-mapped by our method.



(a)



(b)

Figure 4-7: Comparison of our method (a) with Ramsey et al. [2004] [36] (gray) and Gastal et al. [2011] (recursive filtering version) [27] (blue) combined with Boitard et al.'s [2014] [46] segmentation based temporal coherency method (both provided by Ronan Boitard). Also Aydin et al. (red) [3] coherent temporal tone-mapping for HDR videos is shown. These images represent the mean intensity value of frames in Hallway video sequence after being tone-mapped by the following methods.

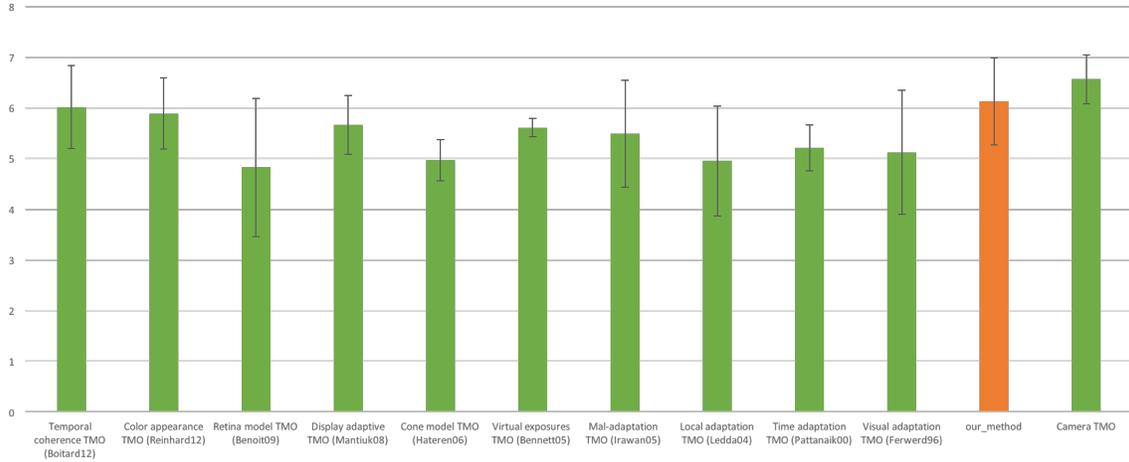


Figure 4-8: Average ratings based on the subjective study

matches the appearance of an actual scene they would expect to see in the real world and not look for a single feature, but to make each judgment based on overall impression. They were allowed to watch each video as many times as they wished and they were allowed to change their previous ratings. They were also asked to watch all the videos once before starting to rate them. The subjects were allowed a maximum time of 2 hours to complete each survey. The average length of completion for each survey was 12 minutes. For each video sequence, we collected data from 30 people. Table 4.2, represents the list of tone-mapping operators, excluding our method, used in the subjective study.

Figure 4-8, shows the average ratings obtained by our subjective study. To conclude, the findings of the subjective study as well as the qualitative comparison in previous section shows that our method is capable of producing results with reasonable local contrast and temporal coherence that are appealing to audience.

4.3 Chapter Summary

In this chapter, we presented the results obtained by our algorithm and compared them with results of other methods for both HDR images and videos. We also showed the advantage of our method over other existing methods for both image and video tone mapping. Furthermore, we conducted a subjective study to compare our results with results of other methods and showed that our method gives better or comparable results to other existing videos tone-mapping operators.

Operator	Processing	Description
Visual adaptation TMO[37]	Global	Use of data from psychophysical experiments to simulate adaptation over time, and effects such as color appearance and visual acuity.
Time-adaptation TMO[38]	Global	Based on published psychophysical measurements. Static responses are modelled separately for cones and rods, and complemented with exponential smoothing filters to simulate adaptation in the temporal domain. A simple appearance model is also included
Local adaptation TMO[42]	Local	Temporal adaptation model based on experimental data operating on a local level using bilateral filtering.
Mal-adaptation TMO[39]	Global	Based on the work by Ward et al. [65] for tone mapping and Pattanaik et al.[38] for adaptation over time. Also extends the threshold visibility concepts to include maladaptation.
Virtual exposures TMO[43]	Local	Bilateral filter applied both spatially for local processing, and separately in time domain for temporal coherence.
Cone model TMO [40]	Global	Dynamic system modelling the cones in the human visual system over time. A quantitative model of primate cones is utilized, based on actual retina measurements.
Display adaptiveTMO [19]	Global	Display adaptive tone mapping, where the goal is to preserve the contrasts within the input (HDR) as close as possible given the characteristic of an output display. Temporal variations are handled through a filtering procedure
Retina model TMO [66]	Local	Biological retina model where the time domain is used in a spatiotemporal filtering for local adaptation levels. The spatiotemporal filtering, simulating the cellular interactions, yields an output with whitened spectra and temporally smoothed for improved temporal stability and for noise reduction

Operator	Processing	Description
Color appearance TMO [64]	Local	Display and environment adapted image appearance calibration, with localized calculations through the median cut algorithm.
Temporal coherence TMO [41]	Local	Post-processing algorithm to ensure temporal stability for static TMOs applied to video sequences. The authors use mainly Reinhard’s photographic tone reproduction, for which the algorithm is most developed. Therefore, the version used in this survey is also utilizing this static operator.
Camera TMO	Global	Represents the S-shaped tone curve which is used by most consumer-grade cameras to map the sensor-captured values to the color gamut of a storage format. The curves applied were measured for a Canon 500D DSLR camera, with measurements conducted for each channel separately. To achieve temporal coherence, the exposure settings are anchored to the mean luminance filtered over time with an exponential filter

Table 4.2: : List of tone mapping operators included in our survey. Processing refers to either global processing that is identical for all the pixels within a frame or local processing that may vary spatially. This table is adapted from [44]

Chapter 5

Conclusions and Recommendations

We have presented a local HDR video and image tone mapping algorithm that can significantly reduce the input dynamic range while preserving local contrast. Our key difference is the use of K-means clustering with color information of image available in the 'a' and 'b' channel of each frame in CIELAB color space. In all the K-means clustering algorithms presented previously for the tone-mapping purpose, they quantize colors and assign each pixel to the value of the mean of their clusters. Contrary our method shifts the pixel values based on the mean value of their cluster, and as a result, our method's output has more color variety.

Our algorithm gives a comparable result to state-of-the-art tone mapping algorithms with the large benefit of a minimal need for parameter setting. In most instances in HDR imaging, it is beneficial being able to tune the parameters manually to reach the most desirable results by the user. Our method allows the user to as mentioned earlier be able to adjust the output manually and get the desired effect in local color and contrast.

We also presented qualitative comparisons to the state-of-the-art and demonstrated that our results are comparable to them and in some cases are even better. The comparisons were made on challenging HDR sequences. In most cases the existing TMOs introduce a variety of distortions such as color saturation, blanched appearances and artificial edges in case of images; and would create ghosting and flickering artifacts in videos. While our method shows minimal sign of the following artifacts.

5.1 Recommendations for Future Work

The work on HDR video tone-mapping can be extended by introducing an objective quality assessment algorithm for video tone mapping operators. There have been a couple of subjective studies over the past few years for video tone-mapping operators, but still, no objective video quality assessments exist. This goal can also be achieved by extending the works on objective quality assessment of tone-mapped HDR images like the work by [13] known as TMQI.

This work can also be optimized regarding its speed. At the moment our algorithm takes quite a while to render a sequence of HDR video as mentioned in Table 4.1. We proposed a few techniques ourselves to reduce its running time. We suggest for further enhancing the rendering time one a look into optimizing K-means for tone mapping purpose. There has been some efforts by [22] in optimizing K-means for this purpose by using dynamic programming. One can refer to this work for further enhancing this area.

One other approach for enhancing our method is to use Fuzzy-C Means Clustering instead of K-means. As discussed in chapter 2 their main difference is that, in Fuzzy-C Means clustering, each point has a weighting associated with a particular cluster, so a point doesn't sit "in a cluster" and it has a weak or strong association to the cluster, which is determined by the inverse distance to the center of the cluster. Fuzzy C-means will tend to run slower than K-means since it's doing more work. Each point is evaluated with each cluster, and more operations are involved in each evaluation. K-Means just needs to do a distance calculation, whereas fuzzy C-means needs to do a full inverse-distance weighting. But the use of fuzzy C-means may be more helpful in our case by bringing more accuracy to the intensity association to a particular cluster and therefore producing better adjustment coefficients for re-adjusting pixel values.

Bibliography

- [1] B. A. Wandell, “Book review: Foundations of vision. by brian a. wandell,” *Journal of Electronic Imaging*, vol. 5, no. 1, pp. 107–107, 1996.
- [2] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.
- [3] T. O. Aydin, N. Stefanoski, S. Croci, M. Gross, and A. Smolic, “Temporally coherent local tone mapping of hdr video,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 196, 2014.
- [4] J. DiCarlo and B. Wandell, “Rendering high dynamic range images,” in *Proceedings of the SPIE: Image sensors*, vol. 3965, 2000, pp. 392–401.
- [5] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic tone reproduction for digital images,” *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 267–276, 2002.
- [6] “Iphone 4 tech specs,” <http://www.apple.com/iphone/iphone-4/specs.html>, Accessed: 2017-08-09.
- [7] G. W. Larson, H. Rushmeier, and C. Piatko, “A visibility matching tone reproduction operator for high dynamic range scenes,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291–306, 1997.
- [8] J. Duan, M. Bressan, C. Dance, and G. Qiu, “Tone-mapping high dynamic range images by novel histogram adjustment,” *Pattern Recognition*, vol. 43, no. 5, pp. 1847–1862, 2010.

- [9] T. Pouli and E. Reinhard, “Progressive histogram reshaping for creative color transfer and tone reproduction,” in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, ACM, 2010, pp. 81–90.
- [10] S. N. Pattanaik, M. D. Fairchild, J. A. Ferwerda, and D. P. Greenberg, “Multiscale model of adaptation, spatial vision and color appearance,” in *Color and Imaging Conference*, Society for Imaging Science and Technology, vol. 1998, 1998, pp. 2–7.
- [11] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” in *ACM Transactions on Graphics (TOG)*, ACM, vol. 21, 2002, pp. 249–256.
- [12] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi, “Evaluation of hdr tone mapping methods using essential perceptual attributes,” *Computers & Graphics*, vol. 32, no. 3, pp. 330–349, 2008.
- [13] H. Yeganeh and Z. Wang, “Objective quality assessment of tone-mapped images,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 657–667, 2013.
- [14] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive logarithmic mapping for displaying high contrast scenes,” in *Computer Graphics Forum*, Wiley Online Library, vol. 22, 2003, pp. 419–426.
- [15] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” in *ACM transactions on graphics (TOG)*, ACM, vol. 21, 2002, pp. 257–266.
- [16] A. C. B. D. Kundu D. Ghadiyaram and B. L. Evans, *Large-scale crowdsourced study for high dynamic range images*, IEEE Transactions on Image Processing, <http://users.ece.utexas.edu/~bevans/papers/2017/crowdsourced/index.html>, 2016.

- [17] —, *Espl-live hdr image quality database*, <http://signal.ece.utexas.edu/~debarati/HDRDatabase.zip>, 2016.
- [18] Q. Shan, J. Jia, and M. S. Brown, “Globally optimized linear windowed tone mapping,” *IEEE transactions on visualization and computer graphics*, vol. 16, no. 4, pp. 663–675, 2010.
- [19] R. Mantiuk, S. Daly, and L. Kerofsky, “Display adaptive tone mapping,” in *ACM Transactions on Graphics (TOG)*, ACM, vol. 27, 2008, p. 68.
- [20] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Computer Vision, 1998. Sixth International Conference on*, IEEE, 1998, pp. 839–846.
- [21] J. W. Lee, R.-H. Park, and S. Chang, “Local tone mapping using the k-means algorithm and automatic gamma setting,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, 2011.
- [22] M. Oskarsson, “Temporally consistent tone mapping of images and video using optimal k-means clustering,” *Journal of Mathematical Imaging and Vision*, vol. 57, no. 2, pp. 225–238, 2017.
- [23] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, *et al.*, “Bilateral filtering: Theory and applications,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009.
- [24] R. L. Lagendijk, J. Biemond, and D. E. Boekee, “Regularized iterative image restoration with ringing reduction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 12, pp. 1874–1888, 1988.
- [25] R. Fattal, “Edge-avoiding wavelets and their applications,” *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 22, 2009.
- [26] S. Paris, S. W. Hasinoff, and J. Kautz, “Local laplacian filters: Edge-aware image processing with a laplacian pyramid,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 68–1, 2011.

- [27] E. S. Gastal and M. M. Oliveira, “Domain transform for edge-aware image and video processing,” in *ACM Transactions on Graphics (ToG)*, ACM, vol. 30, 2011, p. 69.
- [28] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, “Interactive local adjustment of tonal values,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 646–653, 2006.
- [29] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, “Fast local laplacian filters: Theory and applications,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 5, p. 167, 2014.
- [30] P. Shao, S. Ding, L. Ma, Y. Wu, and Y. Wu, “Edge-preserving image decomposition via joint weighted least squares,” *Computational Visual Media*, vol. 1, no. 1, pp. 37–47, 2015.
- [31] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High dynamic range video,” in *ACM Transactions on Graphics (TOG)*, ACM, vol. 22, 2003, pp. 319–325.
- [32] J. Kronander, S. Gustavson, G. Bonnet, and J. Unger, “Unified hdr reconstruction from raw cfa data,” in *Computational Photography (ICCP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–9.
- [33] J. Petit and R. K. Mantiuk, “Assessment of video tone-mapping: Are cameras? s-shaped tone-curves good enough?” *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1020–1030, 2013.
- [34] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, “A versatile hdr video production system,” *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 41, 2011.
- [35] C. Kiser, E. Reinhard, M. Tocci, and N. Tocci, “Real time automated tone mapping system for hdr video,” in *IEEE International Conference on Image Processing*, IEEE Orlando, FL, 2012, pp. 2749–2752.

- [36] S. D. Ramsey, J. T. Johnson III, and C. Hansen, “Adaptive temporal tone mapping,” in *Proceedings of the 7th IASTED International Conference on Computer Graphics and Imaging*, 2004, pp. 124–128.
- [37] J. A. Ferwerda, S. N. Pattanaik, P. Shirley, and D. P. Greenberg, “A model of visual adaptation for realistic image synthesis,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, 1996, pp. 249–258.
- [38] S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg, “Time-dependent visual adaptation for fast realistic image display,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 47–54.
- [39] P. Irawan, J. A. Ferwerda, and S. R. Marschner, “Perceptually based tone mapping of high dynamic range image streams,” in *Rendering Techniques*, 2005, pp. 231–242.
- [40] J. Van Hateren, “Encoding of high dynamic range video with a model of human cones,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 4, pp. 1380–1399, 2006.
- [41] R. Boitard, K. Bouatouch, R. Cozot, D. Thoreau, and A. Gruson, “Temporal coherency for video tone mapping,” *SPIE Optical Engineering+ Applications*, pp. 84990D–84990D, 2012.
- [42] P. Ledda, L. P. Santos, and A. Chalmers, “A local model of eye adaptation for high dynamic range images,” in *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, ACM, 2004, pp. 151–160.
- [43] E. P. Bennett and L. McMillan, “Video enhancement using per-pixel virtual exposures,” in *ACM Transactions on Graphics (TOG)*, ACM, vol. 24, 2005, pp. 845–852.

- [44] G. Eilertsen, R. Wanat, R. K. Mantiuk, and J. Unger, "Evaluation of tone mapping operators for hdr-video," in *Computer Graphics Forum*, Wiley Online Library, vol. 32, 2013, pp. 275–284.
- [45] B. Guthier, S. Kopf, M. Eble, and W. Effelsberg, "Flicker reduction in tone mapped high dynamic range video.," in *Color Imaging: Displaying, Processing, Hardcopy, and Applications*, 2011, p. 78660C.
- [46] R. Boitard, R. Cozot, D. Thoreau, and K. Bouatouch, "Zonal brightness coherency for video tone mapping," *Signal Processing: Image Communication*, vol. 29, no. 2, pp. 229–246, 2014.
- [47] J. T. Tou and R. C. Gonzalez, "Pattern recognition principles," 1974.
- [48] G. J. Klinker, S. A. Shafer, and T. Kanade, "A physical approach to color image understanding," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 7–38, 1990.
- [49] M. Celenk, "A color clustering technique for image segmentation," *Computer Vision, Graphics, and image processing*, vol. 52, no. 2, pp. 145–170, 1990.
- [50] N. Goldberg, "Colour image quantization for high resolution graphics display," *Image and vision computing*, vol. 9, no. 5, pp. 303–312, 1991.
- [51] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "Np-hardness of euclidean sum-of-squares clustering," *Machine learning*, vol. 75, no. 2, pp. 245–248, 2009.
- [52] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," *Theoretical Computer Science*, vol. 442, pp. 13–21, 2012.
- [53] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [54] E. Forgy, "Cluster analysis of multivariate data: Efficiency versus interpretability of classification," *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.
- [55] M. E. Celebi, H. A. Kingravi, and F. Celiker, "Fast colour space transformations using minimax approximations," *IET Image Processing*, vol. 4, no. 2, pp. 70–80, 2010.

- [56] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” 1973.
- [57] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.
- [58] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.
- [59] http://pfstools.sourceforge.net/hdr_gallery.html, Accessed: 2017-08-09.
- [60] <http://www.pauldebevec.com/Research/HDR>, Accessed: 2017-08-09.
- [61] *Luminance hdr*, <http://qtpfsgui.sourceforge.net>, Accessed: 2017-08-09.
- [62] J. Kronander, S. Gustavson, G. Bonnet, A. Ynnerman, and J. Unger, “A unified framework for multi-sensor hdr video reconstruction,” *Signal Processing: Image Communication*, vol. 29, no. 2, pp. 203–215, 2014.
- [63] A. Benoit, D. Alleysson, J. Herault, and P. Le Callet, “Spatio-temporal tone mapping operator based on a retina model,” *Computational color imaging*, pp. 12–22, 2009.
- [64] E. Reinhard, T. Pouli, T. Kunkel, B. Long, A. Ballestad, and G. Damberg, “Calibrated image appearance reproduction,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 201, 2012.
- [65] G. Ward, “A contrast-based scalefactor for luminance display,”
- [66] R. S. S. Tominaga and A. Trémeau, *Computational color imaging*, 2013.