SAKSHI TAPARIA
2017MT10748

ASSIGNMENT 1

The HDR images **"memorial.hdr"** and **"vinesunset.hdr"** have been selected from "Paul Debevecs HDR page" to test all the tone mapping algoritms applied in the assignment.

The assignment has been implemented in **Python-3**. Open CV library has been used for image I/O and Numpy library has been used to preform all array operations.

Algorithm 1: Taking HDR images as input

```
import numpy
import cv2

# "-1" is required to access HDR images as "float" arrays
# without clipping pixel intensities in displayable range.

img = cv2.imread('memorial.hdr', -1)
img_arr = numpy.array(img)
```

Linear Re-scaling

The formula used for linear re-scaling or linear interpolation is as follows -

$I_n = (I_o - Low_i) * \frac{(High_o - Low_o)}{(High_i - Low_i)} + Low_o$

Linear re-scaling was performed on R, G and B channels taking Low_i as the global minimum and $High_i$ as the global maximum pixel intensity.

Three ranges were taken for Low_o and $High_o$ for both the images.

1. **0 to 50**
2. **0 to 255 (visible range)**
3. **0 to 500**

The outputs of linear re-scaling were displayed after applying the following **Gamma correction** function on every pixel intensity of R G B channel separately.

$$\gamma(u) = \left\{ \begin{array}{ll} 12.92 * u, & \text{for } u \leq 0.0031308 \\ 1.055 * u^{1/2.4} - 1.055, & \text{otherwise} \end{array} \right\}$$

Some observations are -

1. When image is re-scaled so the brightest pixels are mapped to the highest displayable intensity, the rest of the image becomes extremely dark.
2. When image is re-scaled by larger values, details in darker regions become visible, but those in brighter regions are lost.
3. **Gamma correction** function works well only when **the image is dim** i.e. when pixel intensities are close to zero and not when the image is already bright.
4. In almost every range of linear re-scaling, some detail or the other is invariably lost.

Figure 1: Memorial.hdr: Re-scaling between 0-50 with (right) and without (left) correction

Figure 2: Memorial.hdr: Re-scaling between 0-255 with (right) and without (left) correction

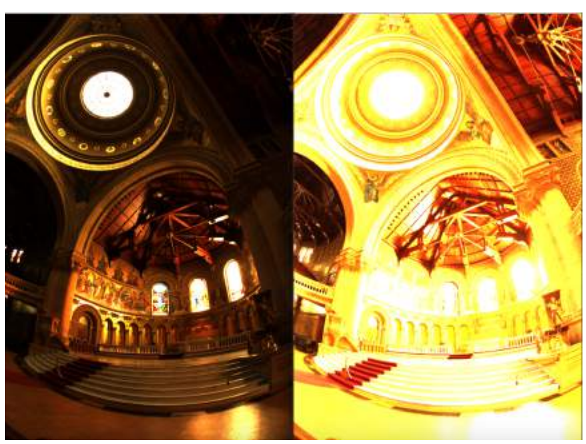

Figure 3: Memorial.hdr: Re-scaling between 0-500 with (right) and without (left) correction

Figure 4: Vinesunset.hdr: Re-scaling between 0-50 with (right) and without (left) correction

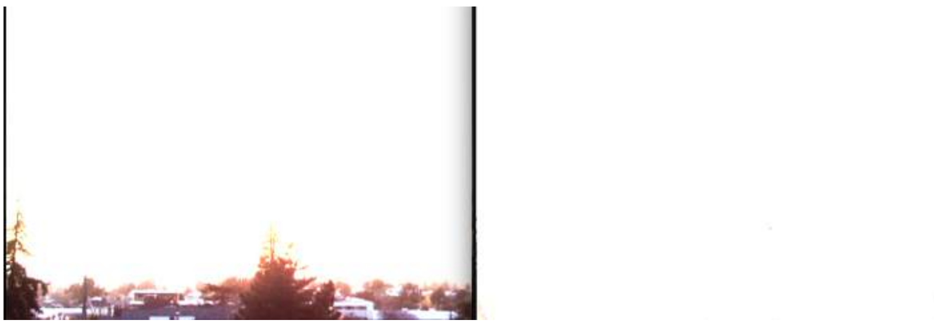

Figure 5: Vinesunset.hdr: Re-scaling between 0-255 with (right) and without (left) correction

Figure 6: Vinesunset.hdr: Re-scaling between 0-500 with (right) and without (left) correction

Logarithmic Re-scaling

As a baseline tone mapping algorithm, re-scaling is performed in log - luminance domain. Luminance of a pixel is given by -

$L = 0.299 * R + 0.587 * G + 0.114 * B$

Log - luminance of a pixel if given by -

$$LogL = \left\{ \begin{array}{lc} log(\delta), & \text{L} = 0 \\ log(L), & \text{otherwise} \end{array} \right\}$$

where, δ = minimum value of luminance

This value is added to L whenever L = 0 to to **avoid the singularity** that occurs if black pixels are present in the image.

Log has been taken with respect to **base 10**. Hence, in order to achieve **1:100 dynamic range**, the log - luminance values must have a **range of 2**.

To achieve this, the maximum and minimum log - luminance values are determined. **Mean is defined as the average of the maximum and minimum** value. Finally, the log - luminance values are **linearly re-scaled from** $(mean - 1)$ **to** $(mean + 1)$ to get a dynamic ratio 1:100 as well as prevent the image from converting into extremely bright or extremely dull image.

Algorithm 2: Logarithmic Re-scaling bounds

```
min_L = numpy.amin(logL)
max_L = numpy.amax(logL)

# mean determines the bounds of linear rescaling
mean = (min_L + max_L)/2

# we need 1:100 pixel range
logLnew = linear_scale(logL, mean-1 , mean+1)
```

In order to undo the log - luminance and extract values of R G B channels, we assume that $Colour_{old}/L_{old} = Colour_{new}/L_{new}$, i.e. **the colour to luminance ratio is constant**.

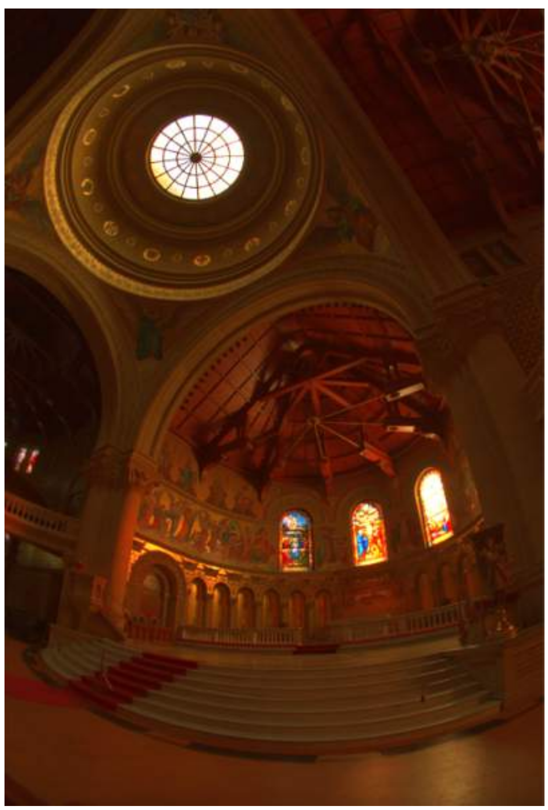

Figure 7: Memorial.hdr: Log - luminance re-scaling

Figure 8: Vinesunset.hdr: Log - luminance re-scaling

Details in the image become weaker because logarithmic re-scaling indiscriminately compresses both large-scale intensity variations and local contrast. To counteract this effect, spatial-domain image enhancement techniques have been implemented.

It is **better to apply the gray-scale techniques on luminance** values of pixels rather than on the R, G and B channels separately as **separate scaling of each colour channel can introduce overall colour distortion.**

Histogram Equalization

Histogram Equalization technique is a **nonlinear operation** that modifies the initial distribution of the pixels in the HDR image to produce a **uniform distribution** over the range of the LDR display.

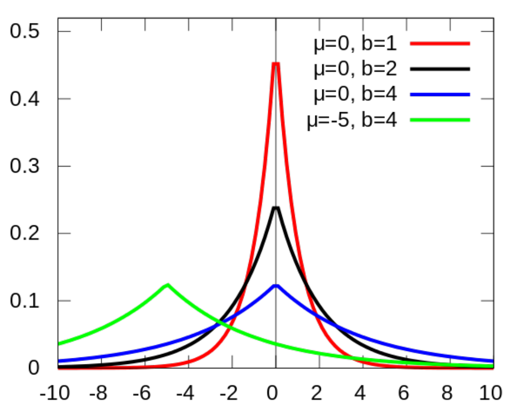

Figure 9: Laplacian function

Probability density function of the HDR pixel values usually behaves in **Laplacian** manner and **manipulating the linear values of luminance causes heavy distortion** because of the insignificant relation to the visual perception.

Therefore, to take into account the human sensitivity function to light, it is better to operate on the logarithmic luminance values instead.

Since the pixel values are continuous and not discrete, therefore a **bin size of N*100** is used to plot the histogram where N is the total number of pixels in the image.

Histogram equalization has been applied on luminance as well as log of luminance for both the images as shown below.

Figure 10: Memorial.hdr: Histogram equalization of luminance (left) and log - luminance (right) values

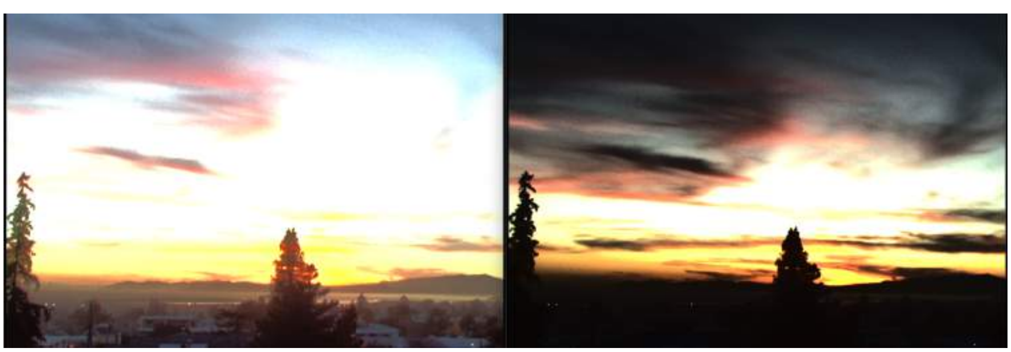

Figure 11: Vinesunset.hdr: Histogram equalization of luminance (left) and log - luminance (right) values

However, histogram equalization in either case does not produce satisfactory results on the HDR images in this case.

Sharpening Filter

The input image is **convoluted** with the **Laplacian Filter** that uses second derivative for image sharpening and also takes the diagonal values into account.

$$Laplacian = \left\{ \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array} \right\}$$

Since the matrix has a positive center coefficient, we add the mask obtained by Laplacian Filter to the original image for generating the output. The Laplacian Filter sharpens the details considerably but the **output is a rather noisy sharpened image.**

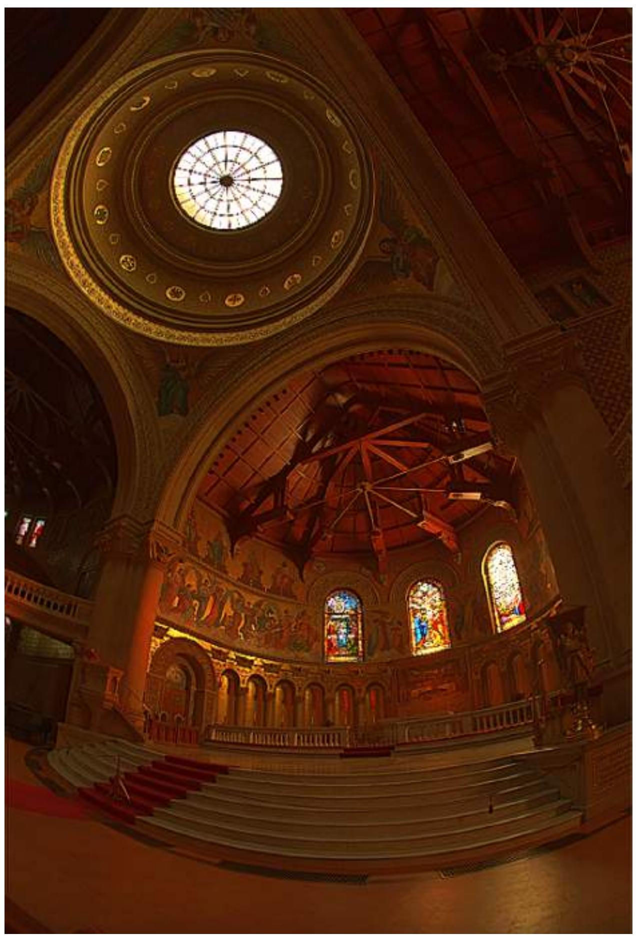

Figure 12: Memorial.hdr: Image sharpening to log-luminance re-scaled image

Figure 13: Vinesunset.hdr: Image sharpening to log-luminance re-scaled image

Mean and Median Filter

Two immediate ways to reduce the noise are to use a median filter or a smoothing filter. A **3*3 median filter** and a **weighted average smoothing filter** has been used on the sharpened image to reduce the noise.

$$Smoothing = \left\{ \begin{array}{ccc} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{array} \right\}$$

Figure 14: Vinesunset.hdr: Sharpening followed by smoothing (left) and median filtering (right)

Figure 15: Memorial.hdr: Sharpening followed by smoothing (left) and median filtering (right)

Sharpening filter tends to produce a noisy image. Further, **median filtering** a non-linear process that **can also remove the image features**.

Figure 16: Sharpening filter alone (left), after adding median filter (center) and after smoothing (right)

Hence, among the techniques implemented, **sharpening using Laplacian filter followed by smoothing using weighted average** produces the best results.

Photographic Tone Reproduction

As per the definitions given in the research paper, **Memorial.hdr** is an image with **11 zones** whereas **Vinesunset.hdr** is an image with **7 zones**.

Initial luminance mapping

L_d values were calculated for every pixel at position (x, y) using the formula -

$$L_d(x, y) = \frac{L(x, y) * (1 + \frac{L(x,y)}{L_{white}^2})}{1 + L(x, y)}$$

where L_{white} has been set to L_{max} which is the maximum luminance value in the image.

Scaled luminance values have been calculated for both the images by varying the key factor value to produce different set of results as shown below.

Figure 17: Vinesunset.hdr: Key value (a) = 0.09

Figure 18: Vinesunset.hdr: Key value (a) = 0.18

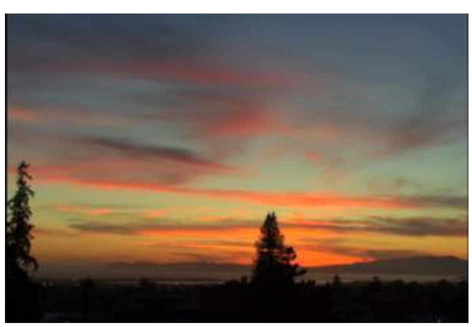

Figure 19: Vinesunset.hdr: Key value (a) = 0.36

Figure 20: Memorial.hdr: Key value (a) = 0.09

Figure 21: Memorial.hdr: Key value (a) = 0.18

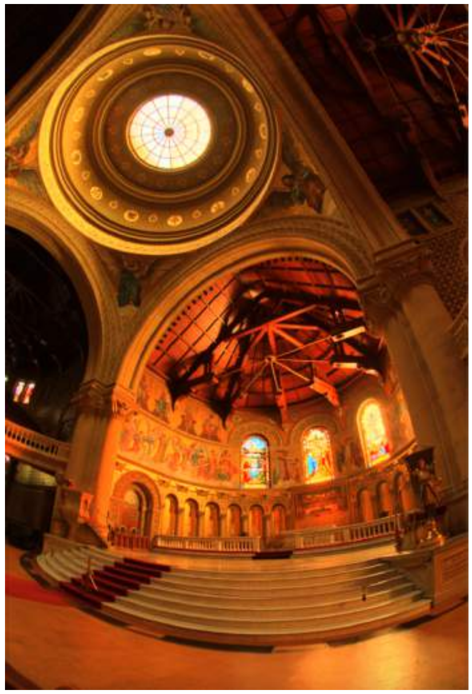

Figure 22: Memorial.hdr: Key value (a) = 0.36

Based on the image outputs, **key value of 0.36** has been taken while implementing the next part.

Automatic dodging and burning

As per the research paper, error bound epsilon has been set to 0.05 and free parameter ϕ has been set to 10 in order to obtain a **sufficiently sharp image.**

Circular convolution has been implemented using FFT. The principle used is that **circular convolution of functions in the Spatial domain are same as the product of functions in the Fourier domain.**

Algorithm 3: Convolution using FFT

```
def compute_V (L_array, R_array):

    L = numpy.fft.fft2(L_array)
    R = numpy.fft.fft2(R_array)

    return ( numpy.fft.ifft2(L*R) )
```

s_m varies at discrete scales starting from 1 pixel and increasing by a **factor of 1.6** to reach up to 450 pixels.

Algorithm 4: s values list

```
s_values = numpy.array( [1, 1.6, 2.56, 4.10, 6.55, 10.49, 16.77, 26.84,
42.95, 68.72, 109.95, 175.92, 281.48, 450.39] )
```

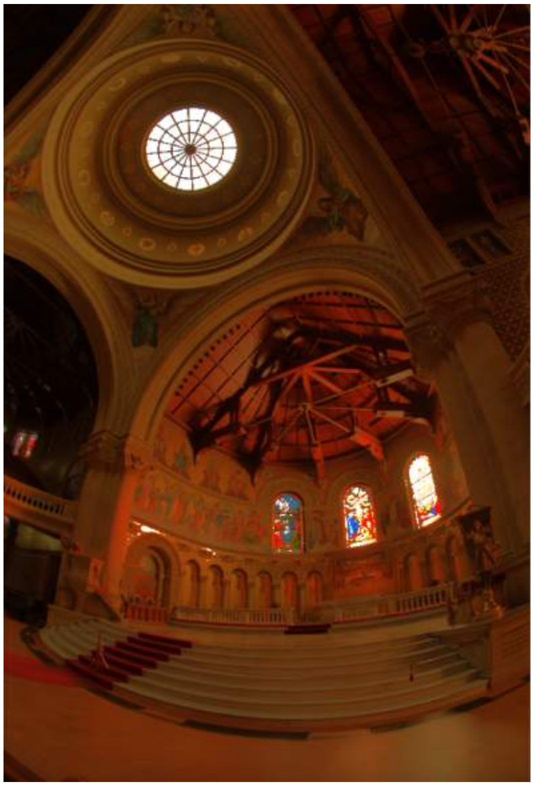

Figure 23: Memorial.hdr: Image obtained after automatic dodging and burning

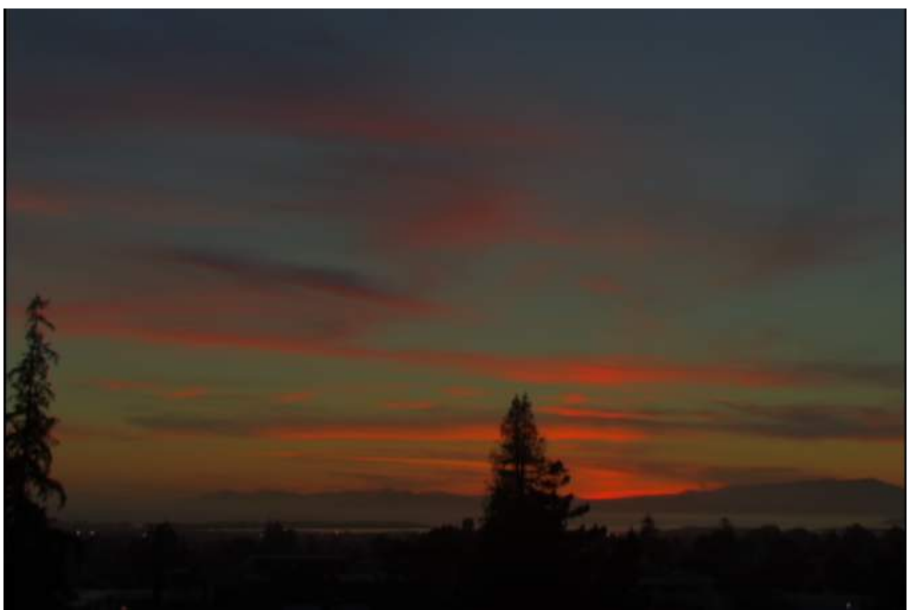

Figure 24: Vinesunset.hdr: Image obtained after automatic dodging and burning

Comparison of results

Figure 25: Memorial.hdr: Sharpening and logarithmic re-scaling vs Automatic dodging and burning

For HDR images with **high zone ranges**, **automatic dodging and burning is necessary** to bring out all the details. For instance, automatic dodging and burning brings out the contrast between the memorial roof and its shadow better than all the previous approaches.

Figure 26: Vinesunset.hdr: Sharpening and logarithmic re-scaling vs Automatic dodging and burning

However, in case of HDR images with **low zone ranges**, automatic dodging and burning is unnecessary. In this case, re-scaling in log - luminance domain and **initial tone mapping approach** of the research paper **works well**.

References

1. COL783 (Digital Image Processing), lecture notes and assignment 1.
2. Gonzalez and Woods, Digital Image Processing, 3rd ed.
3. Reinhard et al., Photographic Tone Reproduction for Digital Images.
4. A. Husseis, A. Mokraoui and B. Matei, "Revisited histogram equalization as HDR images tone mapping operators".
5. G. Guarnieri, S. Carrato, G. Ramponi, "Nonlinear mapping for dynamic range compression in digital images".
6. Fattal et al., Gradient Domain High Dynamic Range Compression.