CMPT 414 Project Report

Recovering High Dynamic Range Radiance and Localized Tone-Mapping Algorithm

Student:

Qiyue Le

Zhu Xi

Contents

Introduction

The range of light we experience in the real world is vast, spanning approximately ten orders of absolute range from star-lit scenes to sun-lit snow. Over four orders of dynamic range may appear in a single scene from the dark shadows to highlights. However, the range of light we can reproduce on our print and screen display devices spans at best about two orders of absolute dynamic range. On the other hand, either with film or an electronic imaging array when we photograph a scene, and digitize the photograph to obtain a two-dimensional array of “luminance” values. These values are rarely the true measurement of relative radiance in the scene. Instead, there is usually an unknown, nonlinear mapping that determines how radiance in the scene becomes pixel values in the image. In a typical characteristic curve (Film response – Exposure curve) of a film is the presence of a small response with no exposure and saturation at high exposures. Digital cameras are facing the same difficulties as they are designed to mimic the response characteristics of film, anticipate nonlinear responses in the display device, and often to convert 12-bit output from the CCD’s analog-to-digital converters to 8-bit values commonly used to store images. As with the film, the most significant nonlinearity in the response curve is at its saturation point, where any pixel with a radiance above a certain level is mapped to the same maximum image value.

The problem that how to capture and reproduce the realistic scene has actually two aspects of difficulties. The limited dynamic range of film and digital camera makes the photographer have to choose the range of radiance values that are interest and determine the exposure time suitably. However in the scenes that spans very high dynamic range often have extreme differences in radiance values that are impossible to capture without either under-exposing or saturating the film. On the other hand, even if we have got the nearly exact measurement of the radiance in the scene, it is impossible to display such high dynamic range scene on a low dynamic range device.

This discrepancy leads to the radiance recover problem and tone reproduction problem. To cover the full dynamic range in a scene with a high dynamic range radiance without any specialized devices, one can take a series of photographs with different exposures. This then poses problems that how should we combines these photos with different exposure to into a composite radiance map and how should we map measured/simulated scene luminance to display luminance and produce a satisfactory image?

This project aims to solve both of the problems. To implement the radiance mapping function that can recover the response function covering the whole dynamic range from the set of photos taken with varying, known exposure values; and display subject satisfactory high dynamic range images using local optimal tone mapping algorithms.

Reconstruct radiance map

The reconstruction process is used to recover high dynamic range radiance maps from multiple photographs with different exposure time. And the recovered radiance maps can be applied the tone mapping method to the get the final high dynamic range photo in 8-bit colors.

According to the Image Acquisition Pipeline, Unknown nonlinear mappings can occur during exposure, development, scanning, digitization, and remapping. Based on the scanning and digitization processes of camera, the pixel value Z of each pixel is calculated by certain formula f of original exposure X at the pixel. And exposure X can be simply computed by exposure time Δt and irradiance E. After knowing the above physical process, we can conclude the following equations:

$$Z = f(X), \qquad X = E\Delta t$$

From these two equations, we can get every pixel value Z_{ij} where i is a spatial index over pixels and j indices over exposure times Δt_j. And we also assume that f is monotonic which means f is invertible. So we can rewrite the above as:

$$Z_{ij} = f\left(E_i \Delta t_j\right) \qquad (1)$$

$$f^{-1}(Z_{ij)} = E_i \Delta t_j$$

Taking the natural logarithm of both sides, and then define g as $\ln f^{-1}$. We then have:

$$\ln f^{-1}\left(Z_{ij}\right) = \ln E_i + \ln \Delta t_j$$

$$g\left(Z_{ij}\right) = \ln E_i + \ln \Delta t_j \qquad (2)$$

Next step is to recover the function g and the irradiances E_i that best satisfy the set of equations arising from Equation 2 in a least-squared error sense. We note that recovering g only requires recovering the finite number of values that $g(z)$ can take since the domain of Z, pixel brightness values, is finite. Letting Z_{min} and Z_{max} be the least and greatest pixel values (integers), N be the number of pixel locations and P be the number of photographs, we formulate the problem as one of finding the $(Z_{max} - Z_{min} + 1)$ values of $g(Z)$ and the N values of $\ln E_i$ that minimize the following quadratic objective function:

$$O = \sum_{i=1}^{N} \sum_{j=1}^{P} \left[g\left(Z_{ij}\right) - \ln E_i - \ln \Delta t_j\right]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2 \qquad (3)$$

The first term ensures that the solution satisfies the set of equations arising from Equation 2 in a least squares sense. The second term is a smoothness term on the sum of squared values of the second

derivative of g to ensure that the function g is smooth.

Moreover, we need an extra weighting function $w(z)$ to emphasize the smoothness and fitting terms toward the middle of the curve. So a simple function would be:

$$w(z) = \begin{cases} z - Z_{min} & for\ z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{min} - z & for\ z \geq \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \tag{4}$$

Equation 3 now becomes:

$$O = \sum_{i=1}^{N}\sum_{j=1}^{P}\{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2$$

Where N is number of sample pixels randomly picking all through P photographs.

Typically (Z_{max} – Z_{min}) = 255

Minimizing O is a straightforward linear least squares problem. We use singular value decomposition (SVD) method to solve it in both Matlab and OpenCV.

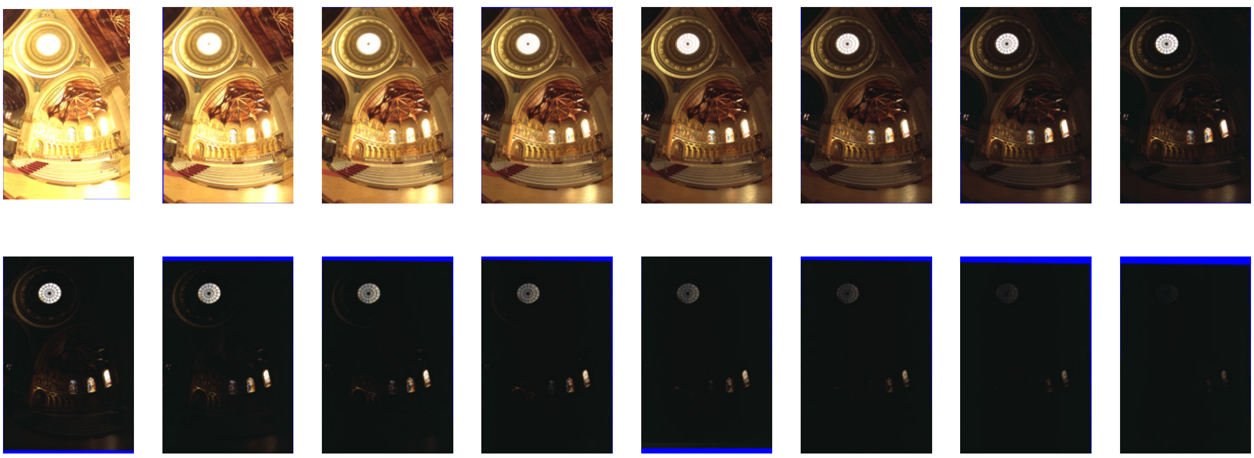
From all equations above, we obtain the final equation to compute each E_i:

$$\ln = g(Z_{ij)} - \ln \Delta t_j \tag{5}$$

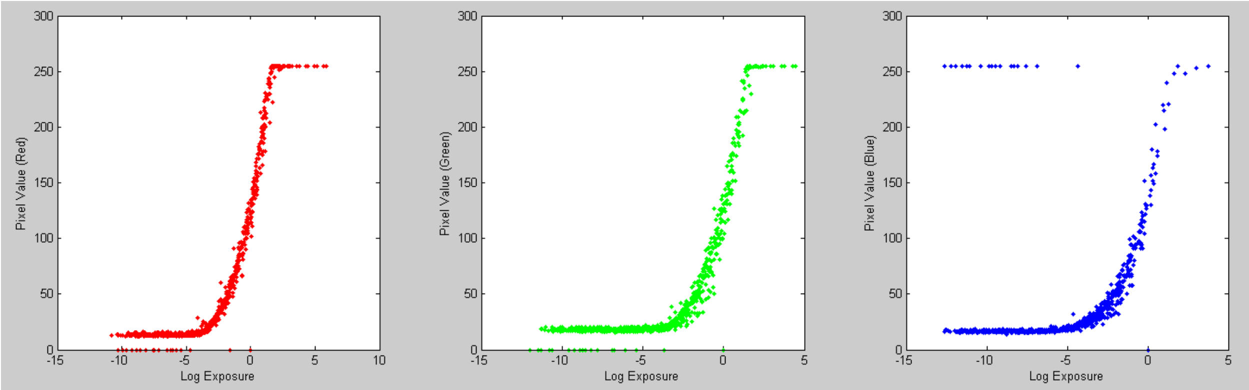
$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})} \tag{6}$$

The following is our test result:

These are 16 images taken by digital camera with exposure time 32 sec to 1/1024 sec:

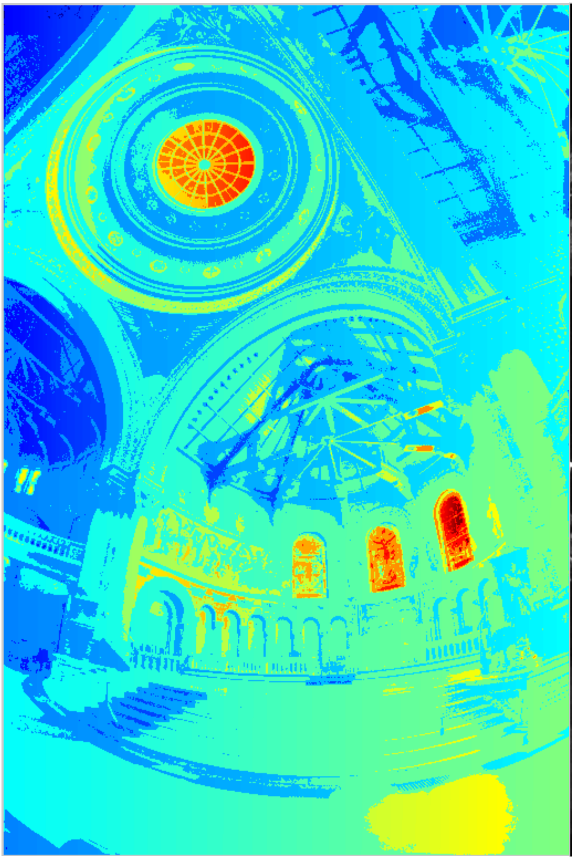


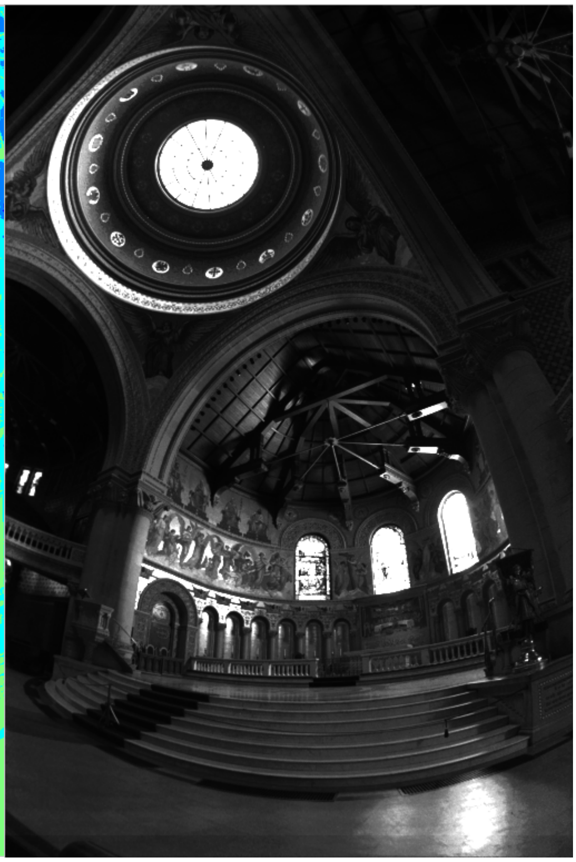
Here are the reconstructed radiance map of RGB channels, of which the x-axis is logarithm exposure $\ln(E_i \Delta t_j)$, and y-axis the pixel value Z_{ij}:



All three channels are fairly consistent with each other. Some weird points in the blue channel are caused by the blue margin in some original photos.

And this is the final recovered radiance map:

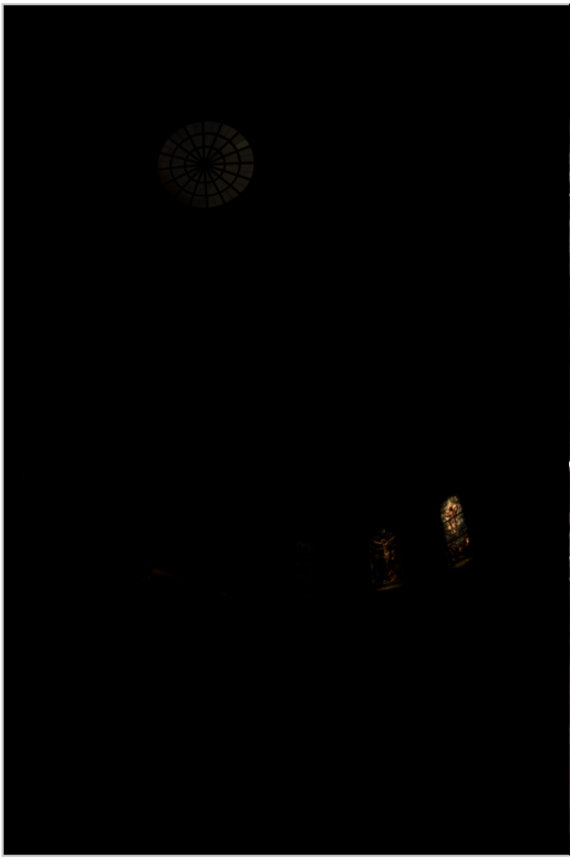




False-color mapped of grayscaled version of radiance map

With grayscale interval [0,1]

(all radiance value >1 becomes 1)



Linear scaled colored radiance map

Pixel value range of each channel is [0,1]

(all radiance value >1 becomes 1)

After implementing the algorithm in both Matlab and OpenCV, we observed the same from both implementations. However solving least square problem (SVD) in OpenCV is much slower than the same solving process in Matlab.

The accuracy of the radiance map of this algorithm relies on how well is the alignment among all photos. We tested some original photos with slightly different filming angles. It would end with ill-constructed radiance map in some color channel. And the final HDR image will have some unnatural color.

Tone Reproduction

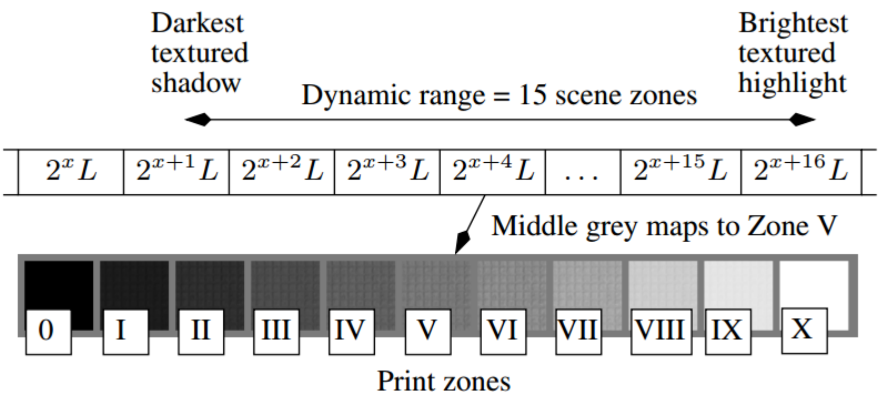
From the radiance recovering we have got an image stored with FP32 RGB, values of which can range from extremely small (less than 0.01) up to hundreds of thousand that representing the dynamic range spanning more can 2 order of magnitude that the normal display device can display.

The tone reproduction problem was first defined by photographers. Often their goal is to produce realistic “renderings” of captured scenes, and they have to produce such renderings while facing the limitations presented by slides or prints on photographic papers.

The method we are using combining global tone mapping algorithm and a tone reproduction algorithm presented by Erik Reinhard in the paper “Photographic Tone Reproduction for Digital Images”. This algorithm mimic the reproduction procedure of conventional photo prints, including zone mapping of the image middle-grey key, global dynamic range compression combining with local Dodging-and-burning algorithm that can darken or lighten specific area of the image to keep the detailed info of local area.

Zone System

A zone is defined as a Roman numeral associated with an approximate luminance range in a scene as well as an approximate reflectance of a print. There are eleven print zones, ranging from pure black (zone 0) to pure white (zone X), each doubling in intensity, and a potentially much larger number of scene zones.



The basic idea is that the key of an image should maps to the middle-key, which is Zone V or middle zone of the all luminance zones presents in the image. In this way we can set the tonal range of the output image based on the scene’s key value to produce overall satisfactory image print.

We view the log-average luminance as a useful approximation to the key of the scene. This quantity $\overline{L_w}$ is computed by:

$$\overline{L_w} = \exp\left(\frac{\sum_{x,y} \log(\delta + L_w(x,y))}{N}\right)$$

Where $L_w(x,y)$ is the “world” luminance for pixel (x,y), N is the total number of pixel in the image and δ is a small value to avoid the singularity that occurs of pure black pixels are present in the image. If the scene has normal-key we would like to map this to middle-key of the displayed image, typically 0.18 on a scale from zero to one. The mapping is done by the following equation:

$$L(x,y) = \frac{a}{\overline{L_w}} L_w(x,y)$$

Where $L(x,y)$ is a scaled luminance and $a = 0.18$, which is called the "key value" of the image because it relates to the key of the image after applying the above scaling. We can choose different a values to map images with low-key and high-key. The value of a can vary from 0.045 up to 0.72 depending on the scene's key.

The mapping results are shown as follows, where the left most image is the original image produced by the radiance mapping.

Original image   *a = 0.09*   *a = 0.18*   *a = 0.36*

We can see that for $a = 0.36$ the image is too bright to preserve most of the detail and for $a = 0.09$ the image is too dark instead. For different images we should choose suitable a values to get satisfactory result.

We find that typically the key-value may fluctuate within the range of 0.09 to 0.36. Usually we don't take photos with extremely high or low key.

Global Tone mapping

After Zone system key mapping we can now apply tone mapping algorithm to fine-tune the tone of the extreme high and extreme low luminance regions. In traditional photography this issue is dealt with by compression of both high and low luminance. However, modern photography has abandoned these "s"-shaped transfer curves in favor of curves that compress mainly the high luminance. A simple tone mapping operator with these characteristics is given by:

$$L_d(x,y) = \frac{L(x,y)}{1 + L(x,y)} \tag{a}$$

This operator cause high luminance to be scaled by approximately $1/L$, while scale low luminance by 1. It create a graceful blend between the scaling for low luminance and high luminance. It will bring all luminance into displayable range but the result is not the best and sometimes not desirable

The equation (a) can also be extended to allow high luminance to burn out in a controllable fashion:

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

Where L_{white} is the smallest luminance that will be mapped to pure white. By default we set L_{white} to the maximum luminance in the scene.

The result is shown as follows:

Original image *Equation (a)* *Equation (b)*

You can see that both equation (a) and equation (b) cannot preserve the details in high or low luminance regions. As we simply blend the luminance into the displayable range, you can see that the edges in the window area is barely viewable. Also the details in the dark is not distinct as well. In other scenes with large high luminance regions using this global tone mapping will loss huge amount of details in the image.

The problem of this global tone mapping algorithm is that it will apply the equation to each pixel in the image. With loss of locality may sometimes create undesired artifacts results. Details in the high or low luminance region are not preserved well. It worth to mention that the global tone mapping is applied to scenes that have a low dynamic range (i.e., $L_{max} < 1$), the effect is a subtle contract enhancement. This can be used as enhancement of images with low contrast. This will be mentioned later.

Automatic dodging-and-burning

Dodging-and-burning is typically applied over an entire region bounded by large contrasts. The difficulty of automatic dodge-and-burning is to identify the size of local region. It can be estimated by using a measure of local contrast, which is computed at multiple spatial scales, often implement by Difference of Gaussians.

Different with the normal Gaussian convolution operator, this Gaussian operator is implemented to be circularly symmetric to find the suitable size of the local area:

$$R_i(x,y,s) = \frac{1}{\pi(\alpha_i s)} exp\left(-\frac{x^2+y^2}{(\alpha_i s)^2}\right) \tag{c}$$

Where x, y is the relative position to the center of the operator window; s is the scale of the operator. Analyzing an image using such operator amounts to convolving the image with these Gaussians, resulting in a response Vi as function of image location, scale and luminance distribution L:

$$V_i(x,y,s) = L(x,y) \otimes R_i(x,y,s) \tag{d}$$

The smallest Gaussian profile will be only slightly larger than one pixel and therefore the accuracy with which the above equation is evaluated, is important. We perform the integration in terms of the error function to gain a high enough accuracy without having to resort to super-sampling.

$$V(x,y,s) = \frac{V_1(x,y,s) - V_2(x,y,s)}{2^{\phi}a/s^2 + V_1(x,y,s)} \tag{e}$$

This constitutes a standard difference of Gaussians approach, normalized by $2^{\phi}a/s^2 + V_1$ where $2^{\phi}a/s^2$ is the term prevents V from becoming too large where V approaches zero. The free parameters a and ϕ are the key value and a sharpening parameter respectively. For computational convenience the center size of the next higher scale is the same as the surround of the current scale. The center-surround ratio is chosen to be 1.6, which results in the DOG model to resemble the LOG.

Usually the DOG operator is used to find the edges in the image. This technique can also be applied here is that the local high contrast is indicating that it is bounded an edge. If we are hitting an edge, then it should be the current measuring scale should provide the size of the local region.

Equation (e) is computed to establish a measure of locality for each pixel, which amounts to finding a scale s_m of appropriate size. This scale may be different for each pixel, and the procedure for its selection is the key to the success of our dodging-and-burning technique. To choose the largest neighborhood around a pixel with fairly even luminance, we threshold V to select the corresponding scale s_m.Starting at the lowest scale, we seek the first scale

s_m where

$$|V(x, y, s_m)| < \epsilon$$

Given a chosen scale for a pixel, we observe that $V1(x, y, s_m)$ may serve as a local average for that pixel. Hence, we can convert equation (a) to the tone reproduction operator by replacing L with V_1 in the denominator:

$$L_d(x, y) = \frac{L(x, y)}{1 + |V_1(x, y, s_m(x, y))|}$$

This function constitutes our local dodging-and-burning operator. The luminance of a dark pixel in a relatively bright region will satisfy $L < V_1$, so this operator will decrease the display luminance L_d, thereby increasing the contrast at that pixel. This is akin to photographic "dodging". Similarly, a pixel in a relatively dark region will be compressed less, and is thus "burned". In either case the pixel's contrast relative to the surrounding area is increased. Choosing the right scale s_m is of crucial importance. If s_m is too small, then V_1 is close to the luminance L. On the other hand, with too large s_m it will cause dark rings around bright areas.

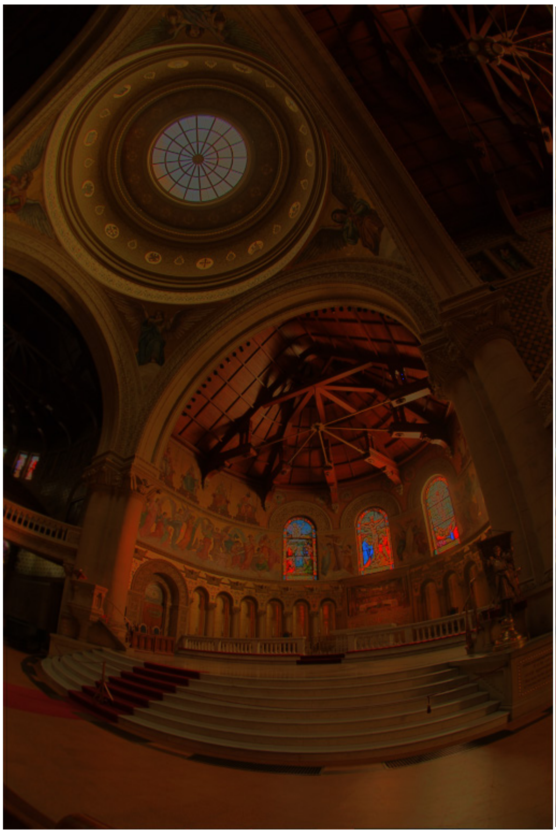
We may fine-tune the result with different input of the parameters. We have tried to find a set of parameter values that is suitable for most of the images, but it doesn't really work well. Each image may require settings with a little bit difference.

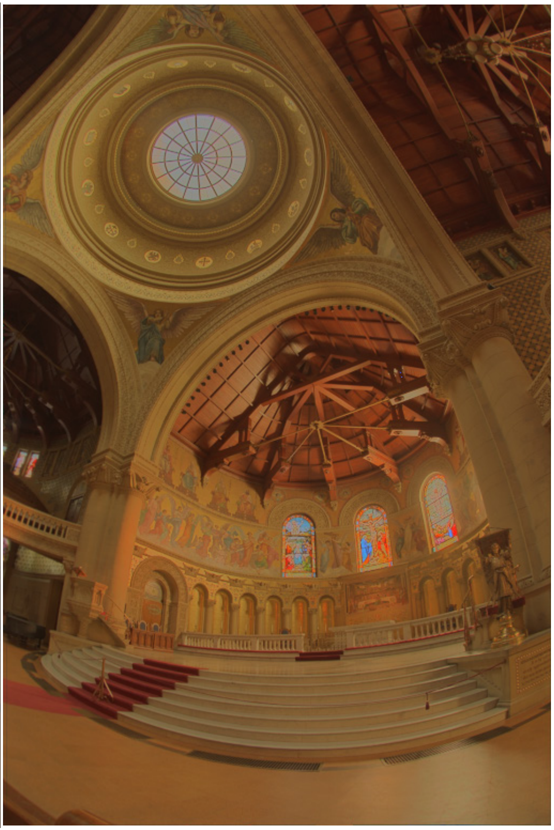
Implementation

The implementation in Matlab and openCV almost shows the same result. The only difference is that for equation (c) we are using the meshgrid function in Matlab that may result in asymmetric x, y grid for the Gaussian kernel, while in openCV we are implement the grid output function ourselves, results of which are always symmetric. Comparing with the openCV version and Matlab version we found no difference in the tone mapping procedure. The main difference in the results is still in the sampling part of the radiance recovery.

For tone mapping the running speed of openCV version shows no primary advantage over the Matlab version, even with large images. We didn't profile the running speed but openCV is not as fast as we expected.

The final result is using gamma correction of 1.6 gamma value to correct the intensity response for display on a low dynamic range device as shown as follows:





Dodge-and-burning *After gamma correction*

The result is relatively satisfactory. You can see that after dodge-and-burning we can clearly see the edges in the window regions and some details in the dark area. However after the gamma correction we found that although the high dynamic range image can be displayed with no loss of detailed info on the image, the contrast of the image not very high resulting the image is not quite subjectively satisfactory.

We have tried to change the input parameter but the result did not become better. Definitely the problem is not with the algorithm, so we tried to enhance the contrast of the image by histogram equalization. It turns out that for a high dynamic range image the histogram equalization does not work well as it cannot capture the value larger than 1.0 and also cannot have a suitable upper bound for the luminance. The results shows that it will burn the whole image out.

Then it comes to us that the equation (b) of global tone mapping algorithm will result in enhancement of the contrast on the original image. While for the radiance reconstruction images the adjustment is not enough. It is probably because we cannot get the true value of the RGB values though we have the radiance map reconstructed. We have to use other functions like imadjust in Matlab to adjust contrast of the result. The final result combining the local doge-and-burning and the global tone mapping algorithm that gives a subjectively satisfactory image as well as preserving details on the image.

The following figure show the result of global tone mapping using equation (b) with gamma correction and the result of combining global tone mapping and local tone reproduction algorithm.

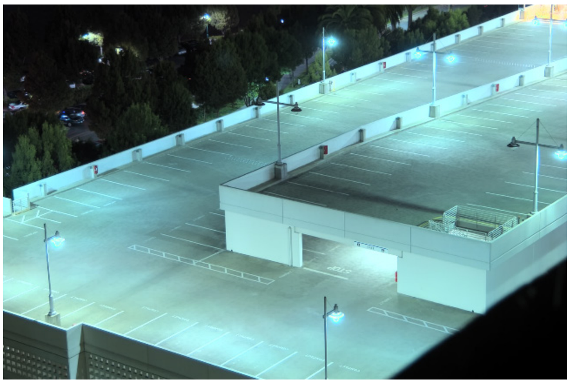
 

Global tone mapping with gamma correction *Combining global and local method*

Final Result

The final result is shown as follows. We show both the result of reconstruct radiance map from the original image as well as performing the tone reproduction on the HDR image directly.

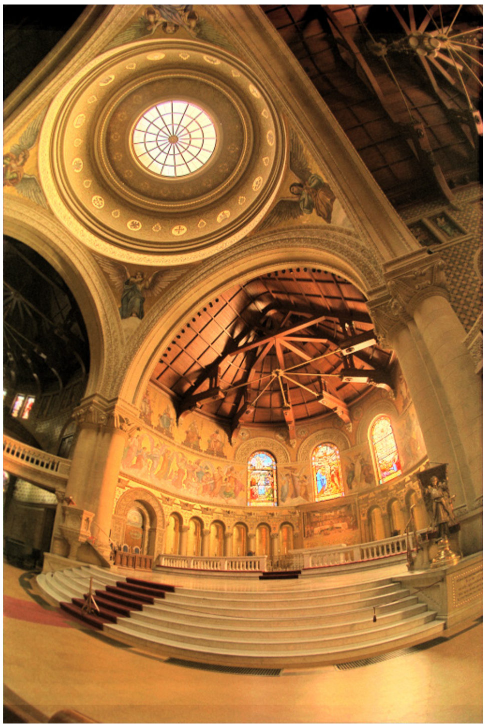
Reconstruct Radiance map from set of images



8 photos with exposure time from 8 to 30

3 photos with exposure time from 0.3 to 6.4

7 photos with exposure time from 0.00025 to 15



16 photos with exposure time from 0.001 to 32

Direct Tone mapping with HDR images