National Tsing Hua University
Department of Electrical Engineering
EE6620 Computational Photography, Spring 2021

Homework #1

High Dynamic Range Imaging

Assigned on March 17, 2021

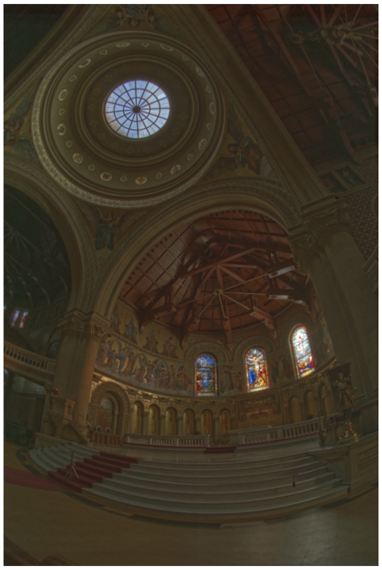
Due by April 14, 2021

Homework Description

Modern cameras are unable to capture the full dynamic range of commonly encountered natural scenes. High-dynamic-range (HDR) photographs are generally achieved by capturing multiple standard-exposure images, often using exposure bracketing, and then merging them into a single HDR image. Also, to view the HDR image on an ordinary low-dynamic-range (LDR) display, tone mapping operation from HDR to LDR on images is required. In this assignment, you will implement the whole HDR photography flow, including **image bracketing**, **camera response calibration**, **white balance**, and finally, **tone mapping**, to visualize your results.

GlobalTM

LocalTM



EdgeLocalTM

Table 1: Scores

Items		Score
Implementation (65%)	Camera Response Calibration	20%
	Global Tone Mapping	10%
	Local Tone Mapping	15%
	Edge-Preserving Filter	15%
	White Balance	5%
Experiment& Free Study (35%)	Experiments	20%
	Free Study	15%

1 Implementation

In this section, you will implement the algorithms for the following five problems. For each function in problems, at least one test function is provided in test_HW1.py for you to check the correctness. You can develop your functions step by step with these unit tests and maintain the correctness during the experiment. You do not have to pass all of the tests to get a perfect score. We will evaluate your correctness based on the result files in each problem. Save your result in folder /result with the same name of the corresponding file in folder /ref. (For example, p2_gtm.png in problem 2.)

1.1 Camera Response Calibration (20%)

Camera response calibration estimates the radiometric response of a camera. In this assignment, you will implement the debevec method [1] to recover a high dynamic range radiance map from a set of bracketing images.

The debevec method consists of 3 steps:

a. Describe the objection function in matrix form.

$$O = \sum_i \sum_j w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j)]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2. \tag{1}$$

In the objection function, Z_{ij} is the intensity measurement for pixel i on image j. Δt_j is the exposure time of the image j, and E_i is our target radiance of location i. $g(z)$ is the camera response which maps the measured intensity Z_{ij} to exposure $E_i \Delta t_j$. The objection function connects data term and smoothness term with a Lagrange multiplier λ. A curve shape prior $w(z)$ is applied in debevec method to fit the curve assumption:

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

Besides, we should also apply a unit exposure assumption by letting $g(127) = 0$. We will describe the equation and prior in the matrix form $Ax = b$. The relation between the objection function and the matrix is shown in Figure 1.

For data term, we would like to fill the matrix with

$$[w(Z_{ij}),\ -w(Z_{ij})] \cdot [g(Z_{ij}),\ \ln E_i] = w(Z_{ij}) \ln \Delta t_j, \tag{3}$$

for measurement $(Z_{ij}, \Delta t_j)$.

And for smoothness term, we would like to fill the matrix with

$$[w(z),\ -2w(z),\ w(z)] \cdot [g(z-1),\ g(z),\ g(z+1)] = 0, \tag{4}$$

for intensity z.

A toy example `toy()` shows a matrix with three measurements. You can also check the matrix construction with example /ref/p1_A.npy and /ref/p1_b.npy.

b. Solve the least square problem with function `numpy.linalg.lstsq()`.

c. Estimate the radiance map from response curve. The final radiance for each pixel i can be estimated by averaging measurements from all images j:

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

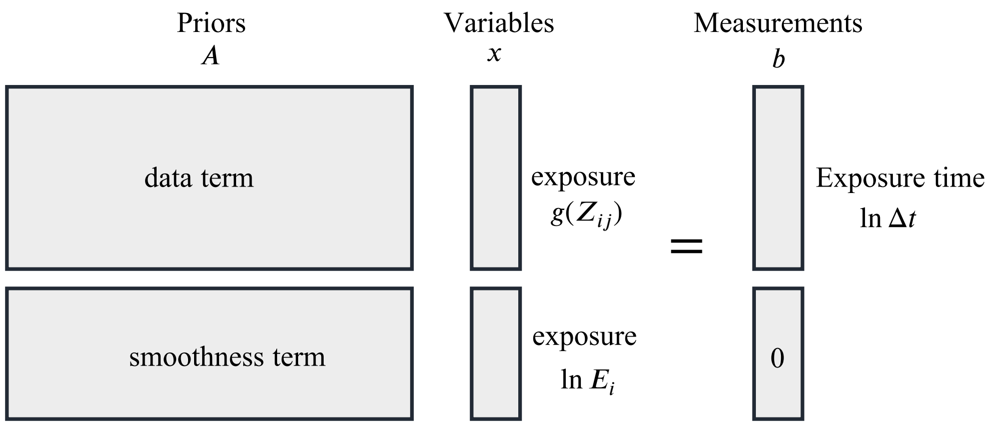


Figure 1: Matrix Form

Complete the debevec method in functions `estimateResponse()` (steps a, b) and `constructRadiance()` (step c) in cr_calibration.py. Test sequences in /ref/p1_* are single channel contents for you to check your function. Function `wholeFlow()` contains example flow using these two functions to construct radiance from bracketing images. Your functions should pass unit tests (test_HW1.py) to get a full score.

1.2 Global tone mapping (10%)

Global tone mapping compresses the contrast of intensity by scaling in log domain as shown in eq (6)

$$\begin{aligned} \log_2 \hat{X} &= s(\log_2 X - \log_2 X_{max}) + \log_2 X_{max}, \\ X_{max} &= \max_{i,j}(X(i,j)), \end{aligned} \tag{6}$$

where X is the radiance, and $\hat{X}$ is the compressed result. Note that this operation should be performed on each of the color channels. Besides the tone mapping operation, we should also perform gamma correction to convert linear RGB to nonlinear color space before saving

$$X' = \hat{X}^{1/\gamma}. \tag{7}$$

In tone mapping, the contrast is reduced in log domain and s here is a parameter to adjust the contrast. On the other hand, γ in nonlinear correction is fixed to commonly used `2.2`.

Complete function `globalTM()` in tm.py. Your result for /TestImg/memorial.hdr should be similar to /ref/p2_gtm.png (PSNR>45dB), under the default setting `s=1.0` to get a full score.

1.3 Local tone mapping with Gaussian filter (15%)

Local tone mapping performs contrast compression only on the base layer to preserve details while balancing the luminance of different regions. A reference flow to achieve this goal is listed as below. Complete functions `localTM()` and `gaussianFilter()` in tm.py. Your result for /TestImg/vinesunset.hdr should be similar to /ref/p3_ltm.png (PSNR>45dB), under the default setting `{N, sigma_s(`σ_s`),scale}={35, 100, 3}` to get a full score. Note that you should use symmetric padding in the filtering.

Local Tone Mapping Flow:

a. Separate Intensity map(I) and Color Ratio(C_r, C_g, C_b) for radiance (R, G, B).

$$I = \text{avg}(R, G, B),$$

$$C_x = X/I,\ for\ X \in \{R, G, B\}.$$

b. Take log of intensity

$$L = \log_2 I.$$

c. Separate the detail layer (L_D) and base layer (L_B) of L with Gaussian filter,

$$L_B(i,j) = \frac{\sum_{k,l} L(i+k, j+l)w(k,l)}{\sum_{k,l} w(k,l)}, \tag{8}$$

$$w_{Gaussian}(k,l) = \exp\left(-\frac{||(0,0)-(k,l)||^2}{2\sigma_s^2}\right),\ k,l \in [-N/2, N/2] \tag{9}$$

$$L_D = L - L_B. \tag{10}$$

eq (8) express the convolution between an image L and symmetric filter w. Indices i, j indexes the pixel of an image on location (i,j). Indices k, l span a filter with $N \times N$ window size. σ_s^2 in eq (9) is the variance of the Gaussian filter.

d. Compress the contrast

$$L'_B = (L_B - L_{max}) * \frac{scale}{L_{max} - L_{min}}, \tag{11}$$

$$L_{min} = \min_{i,j} L_B(i,j), \tag{12}$$

$$L_{max} = \max_{i,j} L_B(i,j). \tag{13}$$

where $scale$ is a parameter for contrast adjustment, different scenes may have different settings. The value around 2~15 would look good for most of scenes.

e. Reconstruct intensity map with adjusted base layer and detail layer,

$$I' = 2^{L'_B + L_D}. \tag{14}$$

f. Reconstruct color map with adjusted intensity and color ratio,

$$C = C_x * I', \; for \; X \in \{R, G, B\}. \tag{15}$$

g. Apply gamma correction.

1.4 Edge-Preserving Filter (15%)

Replace Gaussian filter in local tone mapping with bilateral filter. Bilateral filter on image L can be expressed as

$$w_{bilateral}(L,i,j,k,l) = \exp\left(-\frac{||(0,0)-(k,l)||^2}{2\sigma_s^2} - \frac{||L(i,j)-L(i+k,j+l)||^2}{2\sigma_r^2}\right), \tag{16}$$

where σ_s^2 and σ_r^2 represent the spatial and range variances of the bilateral filter. Again, indices k, l are bounded in $[-N/2, N/2]$ window. Complete function `bilateralFilter()` in tm.py. Your result for /TestImg/vinesunset.hdr should be similar to /ref/p4_ltm.png (PSNR>45dB), under default setting `{N, sigma_s(`σ_s`), sigma_r(`σ_r`), scale}={35, 100, 0.8, 3}` to get a full score.

1.5 White balance (15%)

White balance is the process of removing unrealistic color casts, so that the objects which appear white in person are rendered white in your photo/display. In this part, we scale the color ratios such that the "known to be white" region has equal average intensity in three color channels. To achieve this, color channels G and B are scaled with R_{avg}/G_{avg} and R_{avg}/B_{avg} respectively based on the average value measured in the "known to be white" region:

$$X_{avg} = \text{avg}_{(i,j)\in\Omega_{ktbw}} X \tag{17}$$

$$X' = X * (R_{avg}/X_{avg}), \; for \; X \in \{G, B\} \tag{18}$$

G, B, and G', B' here stand for the radiance before and after the white balance. We will evaluate the result with white balance plus global tone mapping. Complete function `whiteBalance()` in tm.py. Your result with global tone mapping should be similar to /ref/memorial_global_wb.png (PSNR>60dB), under the default setting `{y_range, x_range}={(457, 481), (400, 412)}` to get a full score.

2 Experiment& Free Study (35%)

In this section, we are going to explore the HDR imaging flow using the tools we built in Section 1. There are two parts of problems: Experiments and Free Study. Finish problems in experiments and choose at least two interesting topics of HDR imaging for your free study. You should answer the problems by following the two-step research flow: **Assumption** and **Justification**. Based on your observation, repeatedly propose an assumption, and justify your idea by experiments. After you reach a satisfying result, clearly describe your ideas and justification in the report. Grading criteria for this part are based on the clarity of the report and rigorousness of the justification. Note that you can discuss as more topic as you want in free study; however, we will consider both breadth and depth in grading.

2.1 Experiments (20%)

a. (5%) A conceptual figure (Figure 2) in textbook [2] shows how the bracketing images can recover the radiance of a scene. What is the meaning of this figure? Try to plot a same figure for memorial image set. How do you plot the figure and why do you choose to do so?

b. (5%) What's the meaning of scaling factor in 1.2 and 1.3? You can start with the distribution and the transfer function.

c. (10%) Photograph a bracketing set of exposures for a designed scene and perform the HDR imaging flow. How do you choose the scenes, photograph setup, and parameters in the flow? and why? Do you apply any additional steps? Why?

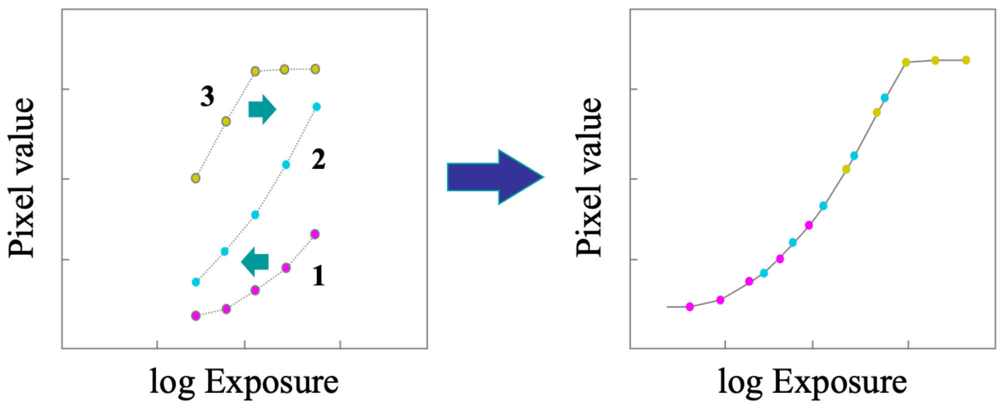


Figure 2: Radiometric Calibration

2.2 Free study (15%)

Find at least two interesting aspects of the HDR imaging flow as your research topics. You can also choose your research topics from the following problems.

1. Sampling in the camera calibration:
   What's the minimum amount of the sampling? How does the sample distribution affect the calibration result? A smoothness prior is applied in the objection function with a Lagrange multiplier. Can we ignore this prior? Does the sampling process affect the setting of smoothness prior? If so, how?
2. Prior in calibration:
   The authors apply a response curve assumption to constrain the curve shape. Is this assumption reasonable? What will happen if we ignore the assumption?
3. Camera response:
   Can we reuse the camera response for different bracketing image sets captured with a same camera? Is it possible to construct a radiance map using single image and a camera with known response?
4. Assumptions in tone mapping:
   Try to justify the assumption of these two methods (*i.e.* regional scaling facto, and halo effect).
5. Bilateral filter:
   There are three parameters $\{N, \sigma_s, \sigma_r\}$ in the bilateral filter. Is there any guidelines to properly select them?
6. Speed issue:
   The execution of bilateral filter is time-consuming. Is there any methods to accelerate it? For example, can we replace it with an approximation? or is there any acceleration algorithms?
7. Automatic white balance:
   In [3, 4] several automatic white balance algorithms are discussed. Choose one of them based on your scene. Why do you choose it? Does the result meet your expectation?
8. Order of algorithm:
   In problem 1-5, we applied the white balance before the tone mapping. Can we exchange the order Why and why not?

3 Deliverable

1. Source code in python. And please don't modify original function IO. Otherwise, your implementation will be considered as incorrect. Proper comments would be great.
2. Your image result for TA to examine your correctness. Put them in /result/.
3. Report in /{studentID}.pdf for Section II.
4. Compress your whole folder HW1/ in HW1_{studentID}.zip and submit to iLms.

Note that wrong file delivery or arrangement will get 5% punishment.

References

[1] P. E. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” in *ACM SIGGRAPH 2008 classes*, pp. 1–10, ACM, 2008.

[2] R. Szeliski, *Computer vision: algorithms and applications*, pp. 481–483. Springer Science & Business Media, 2010.

[3] E. Y. Lam, G. S. Fung, and R. Lukac, “Automatic white balancing in digital photography,” *Single-sensor imaging: Methods and applications for digital cameras*, pp. 267–294, 2008.

[4] G. Zapryanov, D. Ivanova, and I. Nikolova, “Automatic white balance algorithms fordigital stillcameras–a comparative study,” *Information Technologies and Control 1*, pp. 16–22, 2012.