CSCE 448/748 - Computational Photography

Programming Assignment 6

Deadline: Apr. 24th

1 Goal

Modern cameras are unable to capture the full dynamic range of commonly encountered real-world scenes. In some scenes, even the best possible photograph will be partially under or over-exposed. Researchers and photographers commonly overcome this limitation by combining information from multiple exposures of the same scene. You will write software to automatically combine multiple exposures into a single high dynamic range radiance map, and then convert this radiance map to an image suitable for display through tone mapping.

2 Starter Code

Starter code (in Python) along with the images can be downloaded from here. The package includes 2 scenes, which have been captured with two different cameras. "Chapel" is captured with a Canon 35mm SLR and "Office" is captured with a Canon 5D Mark iv. Note that, some of the expected results are included in the "Results" folder.

3 Task 1

Here, you will calculate the camera response function (CRF) from a set of images captured with different exposure times. Two scenes are provided in the "Images" folder. You need to obtain the CRFs for both these cases, and plot the CRFs like Fig. 7 (d) of Debevec's paper. Generally you have to do the followings to estimate the CRF:

- Read the images and their corresponding exposures (already done in the starter code).
- Randomly select N pixels to perform the optimization. Note that, you use the same N randomly selected positions for all the images in the stack. Choosing a very large N could slow down the optimization process. Therefore, you should choose a reasonable N, e.g., $N \approx 5 \times 256/(P-1)$, where P is the number of images in the stack.
- Write the triangle function defined in Eq. 4 of the Debevec's paper and discussed in the class.
- Perform the optimization. This is provided in the starter code (`gsolve` function for Python). You just have to provide appropriate inputs to the function. Note that, the dimension of the inputs are as follows: `Z` ($N \times P$), `B` ($P \times 1$), l (scalar, the value provided in the main file), `w` (256×1).

4 Task 2

In this task, you use the calculated CRF of each scene to reconstruct the radiance of the scene. This can be done by implementing Eq. 6 in Debevec's paper, which was discussed in the class. Once you obtain the radiance, you need to tonemap it to be able to show the results. For this, you will be implementing a global and local tonemapper.

Global – Here, you will obtain a tonemapped image using gamma compression as follows:

$$T = \left(\frac{E}{\max(E)}\right)^{\gamma} \tag{1}$$

Choose a γ value (less than 1) that produces the best result in each case.

Local – You will implement a simple local tonemapper (similar to the one discussed in the class) as follows:

- Your input E is linear RGB values of radiance.
- Compute the intensity (I) by averaging the color channels.
- Compute the chrominance channels: (R/I, G/I, B/I)
- Compute the log intensity: $L = \text{log2}(I)$
- Filter that with a Gaussian filter: $B = \text{filter}(L)$. Larger standard deviations result in tonemapped images with more details. Standard deviation of 0.5 to 2 seem reasonable.
- Compute the detail layer: $D = L - B$
- Apply an offset and a scale to the base: $B' = (B - o) * s$
  - The offset is such that the maximum intensity of the base is 1. Since the values are in the log domain, o = max(B).
  - The scale is set so that the range of output base is dR, i.e., s = dR / (max(B) - min(B)). Values around 4 or 5 for dR should look fine.
- Reconstruct the log intensity: $O = 2^{(B'+D)}$
- Put back the colors: $R', G', B' = O * (R/I, G/I, B/I)$
- Apply gamma compression. Without gamma compression the result will look too dark. Values around 0.5 should look fine (e.g. result$^{0.5}$).

Your HDR images have zero values which will cause problems with log. You can fix this problem by replacing all zeros by some factor times the smallest non-zero values.

5 Write Up

For both scenes, you should show the estimated CRF and the global and local tonemapped version of the HDR image (radiance). Describe how you implemented the assignment. Discuss any problem you faced when implementing the assignment or any decisions you had to make.

6 Graduate Credit

There are no additional requirements for graduate credit for this project.

7 Deliverables

Your entire project should be in a folder called "`firstname_lastname`". This folder should be zipped up and submitted through e-campus. Inside the folder, you should have the followings:

- A folder named "Code" containing all the codes for this assignment. Please include a README file to explain what each file does if you add any other files to the starter code.
- A report in the pdf format. **Make sure you write your name on top of the report. Also make sure the pdf file is under 5 MB.**

Make sure you exclude all the results and original images from your submission.

8 Checklist

Make sure you can check all the items below before submitting your assignment. You will lose 5 points for each item that cannot be checked.

- □ The folder is named properly (“firstname_lastname”). Note _ between first and last name. The folder structure should be exactly as follows

  “firstname_lastname.zip”

  - “firstname_lastname”
    - ∗ “Code”
    - ∗ Report.pdf
- □ Inside the root folder, there is a folder called “Code” that contains your source code. Also make sure the report is in the root folder.
- □ The folders “Images” and “Results” are not included (you only submit your codes and a report).
- □ Name written on top of the report.
- □ The report is in pdf format and the file is under 5 MB.

9 Ruberic

Total credit: [100 points]

[30 points] - CRF estimation
[20 points] - Radiance reconstruction
[10 points] - Global tonemapping
[30 points] - Local tonemapping
[10 points] - Write up

10 Acknowlegements

This project is partially based on James Hays Computational Photography course with permission.