CSCE 689 - Computational Photography

Programming Assignment 4

Deadline: Apr. 12th

1 background

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 Overview

We want to estimate the camera response function from several LDR images with different exposures. It is highly recommended that you read Sections 2.1 and 2.2 in Debevec and Malik 1997 to help understand this process. Below is a summary.

The observed pixel value Z_{ij} for pixel i in image j is a function of unknown scene radiance and known exposure duration:

$$Z_{ij} = f(E_i \Delta t_j). \tag{1}$$

E_i is the unknown scene radiance at pixel i, and scene radiance integrated over some time $E_i \Delta t_j$ is the exposure at a given pixel. In general, f might be a somewhat complicated pixel response curve. We will not solve for f, but for $g = \ln(f^{-1})$ which maps from pixel values (from 0 to 255) to the log of exposure values (equation 2 in Debevec's paper):

$$g(Z_{ij}) = \ln(E_i) + \ln(t_j). \tag{2}$$

Solving for g might seem impossible (and indeed, we only recover g up to a scale factor) because we know neither g or E_i. The key observation is that the scene is static, and while we might not know the absolute value of E_i at each pixel i, we do know that the value remains constant across the image sequence.

To make the results robust, we want to consider two additional things:

- We expect g to be smooth. Debevec adds a constraint to our linear system which penalizes g according to the magnitude of its second derivative. Since g is discrete (defined only at integer values from g(0) to g(255) we can approximate the second derivative with finite differences, i.e.,

  $$g''(x) = (g(x-1) - g(x)) - (g(x) - g(x+1)) = g(x-1) + g(x+1) - 2g(x). \tag{3}$$

  We will have one such equation for each integer in the domain of g, except for $g(0)$ and $g(255)$ where the second derivative would be undefined.

- Each exposure only gives us trustworthy information about certain pixels (i.e. the well exposed pixels for that image). For dark pixels the relative contribution of noise is high and for bright pixels the sensor may have been saturated. To make our estimates of E_i more accurate we need to weight the contribution of each pixel according to Equation 6 in Debevec. An example of a weighting function w is a triangle function that peaks at $Z = 127.5$, and is zero at $Z = 0$ and $Z = 255$.

With these two considerations, we can obtain g by solving the following equation:

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} [w(z) g''(z)]^2 \tag{4}$$

Here, we have a total of $N + Z_{\max} - Z_{\min} + 1$ unknowns. In our case, $Z_{\min} = 0$ and $Z_{\max} = 255$ and, thus, we have $N + 256$ unknowns. Given the fact that we have NP equations, we need to choose a sufficiently large N to be able to reliably solve the above equation, i.e. $NP \gg N + 256$ or $N \gg 256/(P-1)$. However, choosing a very large N could slow down the optimization process. Therefore, you should choose a reasonable N, e.g., $N \approx 5 \times 256/(P-1)$.

3 Details

This assignment has two major parts. The first part is to calibrate the camera and calculate the camera response function (CRF) using the process, explained above. 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 throughout the stack to perform the optimization.
- Write the triangle function defined in Eq. 4 of the Debevec's paper.
- Perform the optimization. This is provided in the starter code (`gsolve.m`). You just have to provide appropriate inputs to the function.

In the second part of the assignment, you use the calculated CRF to reconstruct an HDR image given a series of input LDR images. This can be done by implementing Eq. 6 in Debevec's paper. Once you obtain the HDR image, you have to display it by tonemapping the image. You need to do this in two different ways. Using the global operator, which we discussed in the class, i.e., $L/(1+L)$, and using MATLAB's built-in local tonemapper, i.e., `localtonemap()`.

Starter code (in MATLAB) along with the images can be downloaded from here. The package includes 4 scenes, which have been captured with two different cameras. "0_Calib_Chapel" is captured with a Canon 35mm SLR and the rest are captured with a Canon 5D Mark iv. You have to first use the two calibration sets (the one with "Calib" in the name) to obtain the CRF for each camera (CRF0 for 35mm SLR and CRF1 for 5D Mark iv). Then use CRF0 to reconstruct HDR image (and the global and local tonemapped versions) for "0_Calib_Chapel" scene and use CRF1 to do the same for the rest of the scenes.

In addition to this, use CRF1 to reconstruct the HDR image (and the global and local tonemapped versions) for "0_Calib_Chapel". Discuss your observation.

4 Deliverables

Your project should be in a folder and submitted in zip format (called "`firstname_lastname.zip`") 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.
- A folder named "Results" containing 10 tonemapped images. Tonemapped images using global and local operators for each scene (8 images) as well as two images (local and global tonemappers) when using CRF1 to reconstruct an HDR image for "0_Calib_Chapel".

- A write up describing the algorithm used to implement the assignment. Please discuss any problem you faced when implementing the assignment or any decisions you had to make. For the first part of the assignment, you have to show a plot like Fig. 7 (d) for each CRF (two plots, one for CRF0 and one for CRF1). For the second part, you use the calculated CRFs to reconstruct HDR image for each scene. Note that you should use CRF0 to reconstruct "0_Calib_Chapel", while CRF1 to reconstruct the rest of the scenes. For each scene, show the images obtained using the local and global tonemappers. Discuss which one's better and why. Finally, show the two images by using CRF1 on "0_Calib_Chapel". Discuss your observation and explain why do you think it happens. *Make sure you write your name on top of the report.*

5 Ruberic

- Camera calibration - 50 pts
- HDR reconstruction - 20 pts
- Tonemapping - 10 pts
- Write up - 20 pts

6 Acknowlegements

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