CS 435 - Computational Photography

Assignment 2 - High Dynamic Range (HDR) Images

Introduction

Digital cameras are unable to capture the full dynamic range of real scenes (especially those with sunlight). We can capture the full dynamic range of these real scenes by taking multiple exposures.

In this assignment we'll look at combining images taken at different exposure lengths to form a *high dynamic range* (HDR) image. In addition, we'll look at how to convert an HDR image to a standard dynamic range (SDR) image.

Grading

Theory Questions	20pts
Plotting pixel value vs log exposure	15pts
Finding Response Curves	30pts
Generating HDR and Tonemapped Images	35pts
TOTAL	100pts

Table 1: Grading Rubric

1 Theory Questions

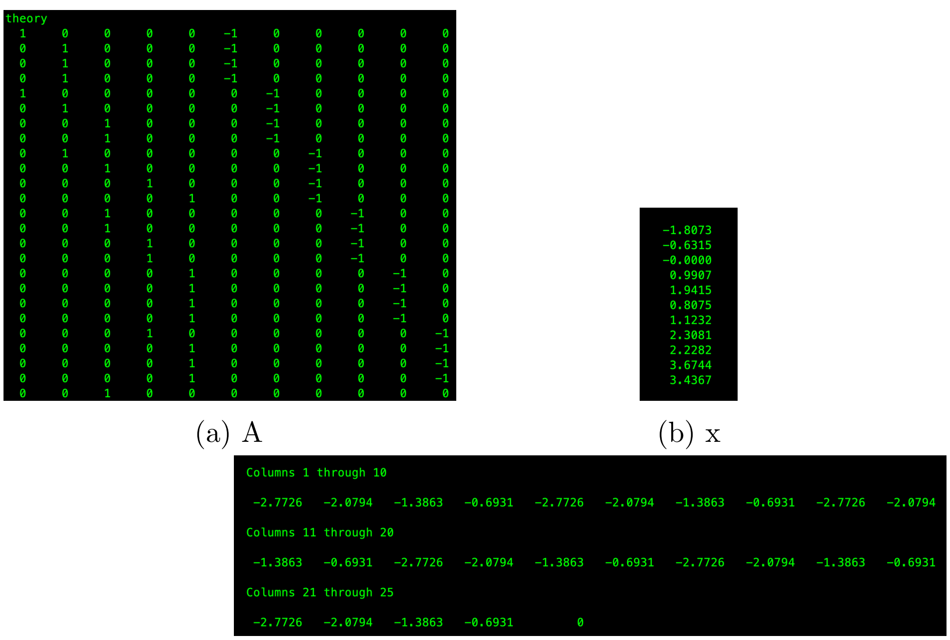
Let's imagine that our camera can capture up to 5 possible values for a pixel. We take four pictures of the same scene with exposure lengths of $\frac{1}{16}, \frac{1}{8}, \frac{1}{4}$ and $\frac{1}{2}$ seconds. The table below shows the pixel intensity of six pixels in this image for each of these four exposure lengths:

$\Delta t =$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$
Pixel 1	0	1	1	1
Pixel 2	0	1	2	2
Pixel 3	1	2	3	4
Pixel 4	2	2	3	3
Pixel 5	4	4	4	4
Pixel 6	3	4	4	4

1. (2pts) What is the value of each pixel if you just averaged their value across images?

Pixel 1	3/4
Pixel 2	5/4
Pixel 3	5/2
Pixel 4	5/2
Pixel 5	4
Pixel 6	15/4

2. (10pts) What is the A matrix and the x and b vectors necessarily to find the elements of the log irradience function? Since A will be a relatively large matrix, feel free to place its elements in a matrix in Matlab, then include a screenshot of the matrix in your submission.



```
theory
  1     0     0     0     0    -1     0     0     0     0     0
  0     1     0     0     0    -1     0     0     0     0     0
  0     1     0     0     0    -1     0     0     0     0     0
  0     1     0     0     0    -1     0     0     0     0     0
  1     0     0     0     0     0    -1     0     0     0     0
  0     1     0     0     0     0    -1     0     0     0     0
  0     0     1     0     0     0    -1     0     0     0     0
  0     0     1     0     0     0    -1     0     0     0     0
  0     1     0     0     0     0     0    -1     0     0     0
  0     0     1     0     0     0     0    -1     0     0     0
  0     0     0     1     0     0     0    -1     0     0     0
  0     0     0     0     1     0     0    -1     0     0     0
  0     0     1     0     0     0     0     0    -1     0     0
  0     0     1     0     0     0     0     0    -1     0     0
  0     0     0     1     0     0     0     0    -1     0     0
  0     0     0     1     0     0     0     0    -1     0     0
  0     0     0     0     1     0     0     0     0    -1     0
  0     0     0     0     1     0     0     0     0    -1     0
  0     0     0     0     1     0     0     0     0    -1     0
  0     0     0     0     1     0     0     0     0    -1     0
  0     0     0     1     0     0     0     0     0     0    -1
  0     0     0     0     1     0     0     0     0     0    -1
  0     0     0     0     1     0     0     0     0     0    -1
  0     0     0     0     1     0     0     0     0     0    -1
  0     0     1     0     0     0     0     0     0     0     0
```

(a) A

```
-1.8073
-0.6315
-0.0000
 0.9907
 1.9415
 0.8075
 1.1232
 2.3081
 2.2282
 3.6744
 3.4367
```

(b) x

```
Columns 1 through 10

 -2.7726   -2.0794   -1.3863   -0.6931   -2.7726   -2.0794   -1.3863   -0.6931   -2.7726   -2.0794

Columns 11 through 20

 -1.3863   -0.6931   -2.7726   -2.0794   -1.3863   -0.6931   -2.7726   -2.0794   -1.3863   -0.6931

Columns 21 through 25

 -2.7726   -2.0794   -1.3863   -0.6931         0
```

(c) b

Figure 1: New Images

3. (3pts) Using Matlab, what are the values of elements of the log irradience function for this scenario?

$$g = \begin{bmatrix} -1.8073 \\ -0.6315 \\ -0.0000 \\ 0.9907 \\ 1.9415 \end{bmatrix}$$

4. (5pts) Finally, what is the irradiance for each pixel using the log irradience function you computed in the previous step?

$$g(Z) = \begin{bmatrix} 2.6256 & 4.2545 & 2.1273 & 1.0636 \\ 2.6256 & 4.2545 & 4.0000 & 2.0000 \\ 8.5091 & 8.0000 & 10.7728 & 13.9389 \\ 16.0000 & 8.0000 & 10.7728 & 5.3864 \\ 111.5114 & 55.7557 & 27.8778 & 13.9389 \\ 43.0911 & 55.7557 & 27.8778 & 13.9389 \end{bmatrix}$$

2 (15 points) Plotting pixel value vs log exposure

On BBlearn you have been provided with a directory, *memorial*. This directory contains a file *images.txt* that provides a list of images in that directory, as well as their exposure lengths. Your first task will be to parse the image.txt file to get the list of file names and associated exposure times, and then load all the images in that directory.

Next, select *three* pixel locations (random or strategically chosen) and plot the values in their *red channel* as a function of Δt_j, where Δt_j is the exposure length for image j. This is akin to plotting the *log irradiance* as a function of the exposure length, but with an identity log irradiance function.

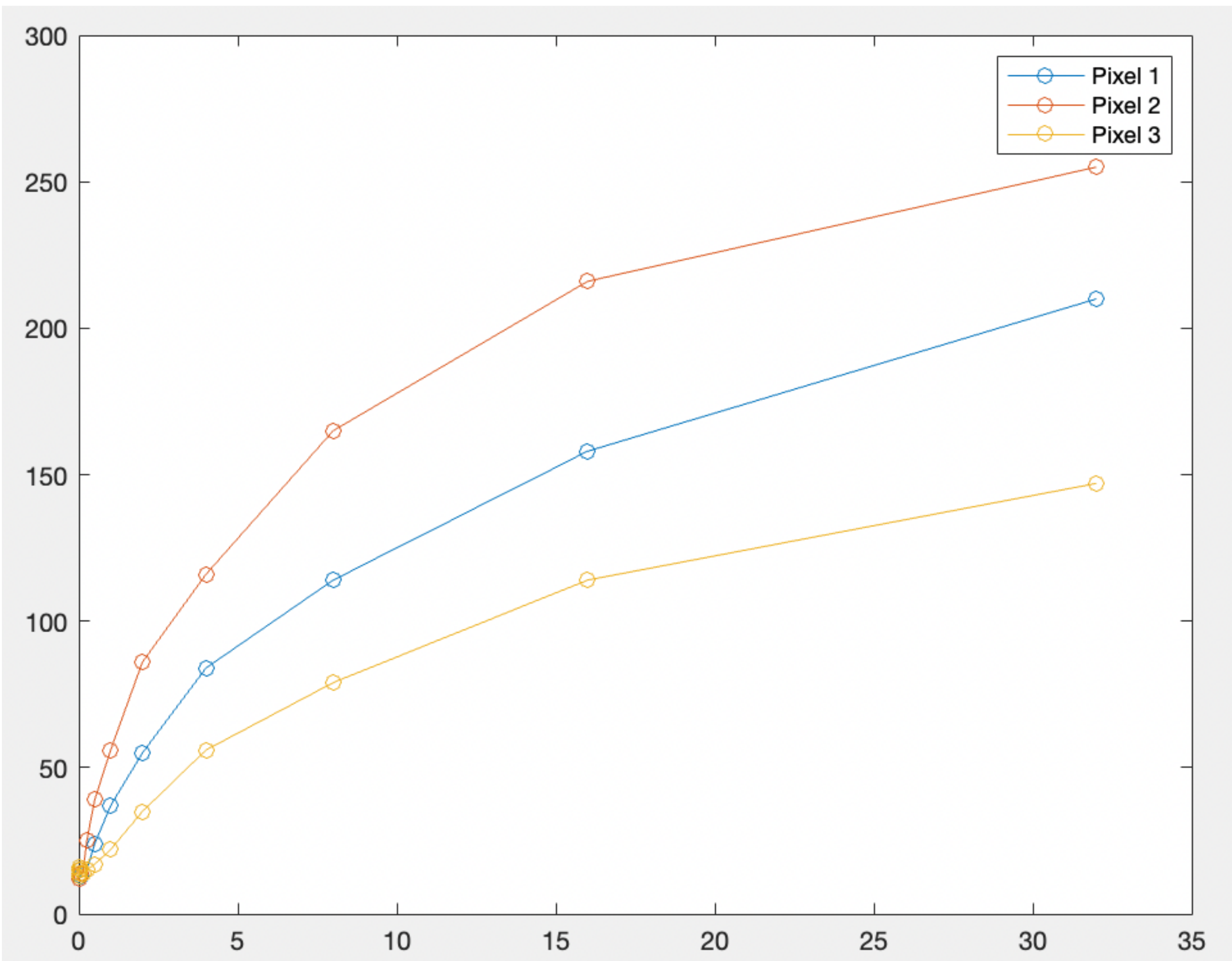


Figure 2: Observed red intensity vs Exposure Length

3 (30 points) Finding and Plotting a Log Irradance Function

Using the technique discussed in class, find the log irradiance function $g(z_{ij})$ for the *red* color channel. Then repeat the plot from the previous section (so you're plotting $ln(R_i)$ vs $exposure$). You image should look something like Figure 3.

NOTE: The more pixels you use to solve the system the better. That being said, the more pixels you use, the larger the matrix to invert will becomes. Therefore experiment with how many pixels to use.

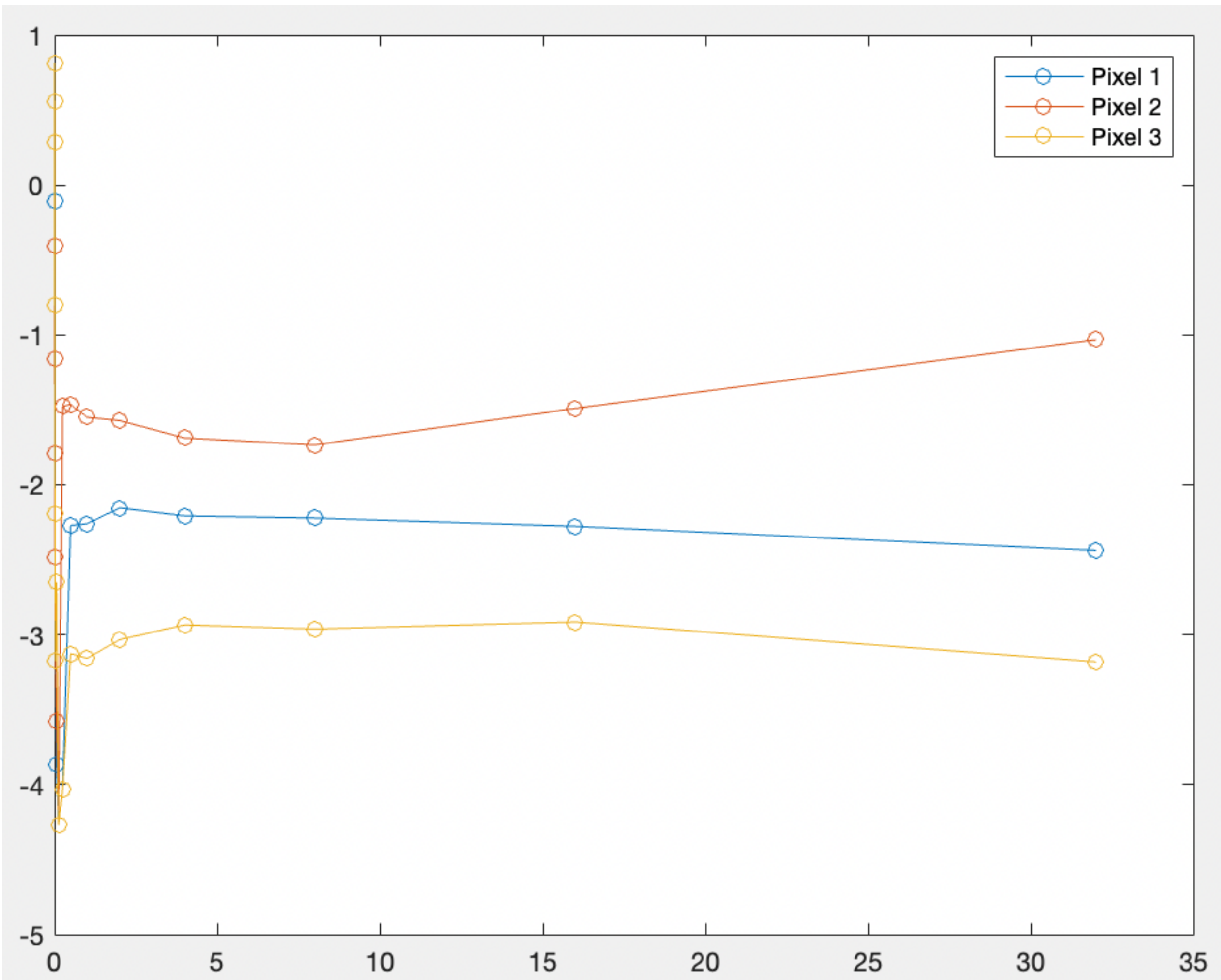


Figure 3: Log Irradiance vs Exposure Length with learned log irradiance function

4 (40 points) Generate HDR Image and Tonemap

Now we're going to find the log irradience function for each of the color channels, use them to create our HDR image, then tonemap back to a uint8 image!

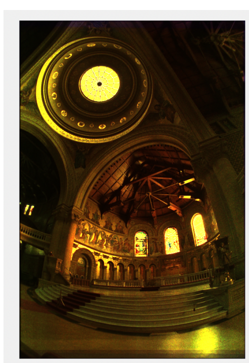
Your process is as follows:

1. First, find the log irradience function for each color channel (like you did in the previous section).
2. Next, for each color channel, go through all the pixel locations and compute the new pixel value to be the average of the pixel's irradiance values from the different exposure length images (making use of the associated channel's log irradiance function, and that image's exposure time). As a quick reference, from the lecture slides, the equations to do this are:

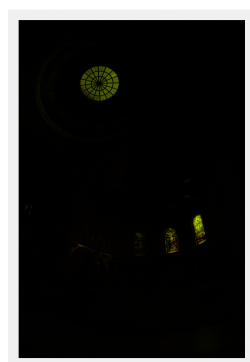
$$ln(R_i) = \frac{1}{P}\sum_{j=1}^{P}(g(z_{ij}) - ln(\Delta t_j)) \tag{1}$$

$$R_i = e^{ln(R_i)} \tag{2}$$

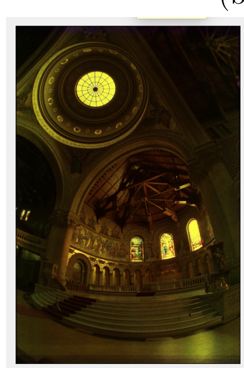
3. Finally, let's look at the affect of tonemapping our HDR image prior to converting to a full-range SDR image. Generate two images, one that just map the min/max range of the HDR content to the [0,255] range, and one that first tonemaps via the function $f(x) = \frac{x}{1+x}$, then maps the new min/max range to [0,255].



(a) HDR Image



(b) Naive SDR Image



(c) SDR w/ Tone-mapping

Figure 4: New Images

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF writeup that includes:
   (a) Your answer to the theory question(s).
   (b) Your plot for Part 2.
   (c) Your plot for Part 3.
   (d) Your HDR and Tonemapped images for Part 4.
2. A README text file (**not** Word or PDF) that explains:
   (a) Features of your program
   (b) Name of your entry-point script
   (c) Any instructions on how to run your script
3. Your source files.