Project 1: High Dynamic Range Imaging

Emily Busche
Giang Doan

Introduction

High dynamic range (HDR) images have much larger dynamic range than traditional images' 256 brightness levels. In addition, they correspond linearly to physical irradiance values of the scene. Hence, they have many applications in vision.
In this project, we build an application that can assemble an HDR image from a set of images. Step 1, images will be captured in different shutter speeds. Step 2, we use Ward's MTB algorithm to align the images. Step 3, we find the response function which will be used to convert the whole image. Step 4, we use Debevec's algorithm to construct HDR radiance map then write result to a HDR image. Finally, we use Drago's tone mapping algorithm to convert HDR image to a regular image.

1. Capture Images (Emily and Giang)

We took pictures of three scenes (17 images of each scene): 1 at Bascom Hall and 2 at the Capitol. Then we decided to choose the 2nd scene.

Figure 1: Bascom Hall – The first scene

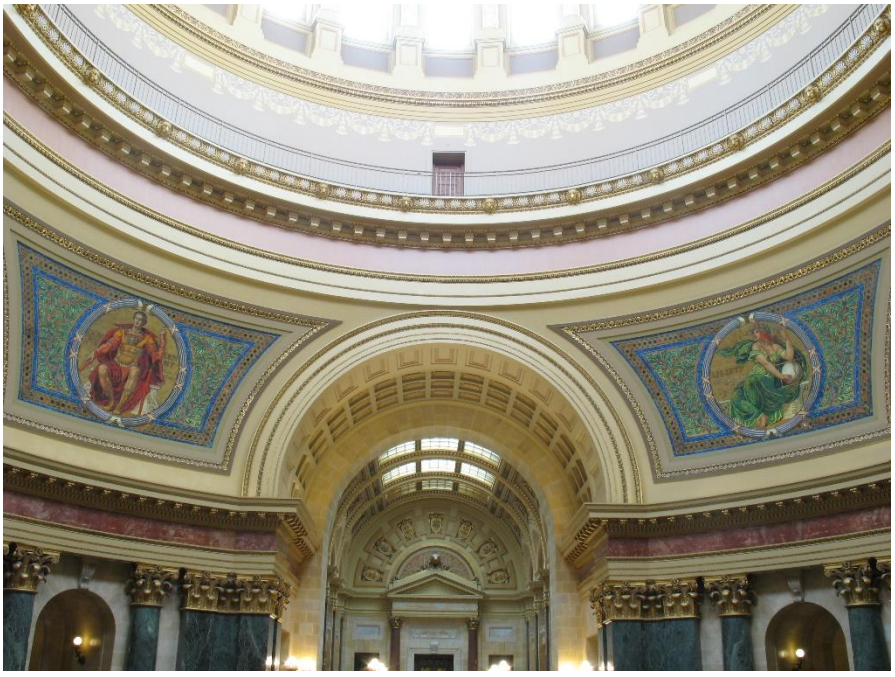

Figure 2: State Capitol Building – The 2nd scene

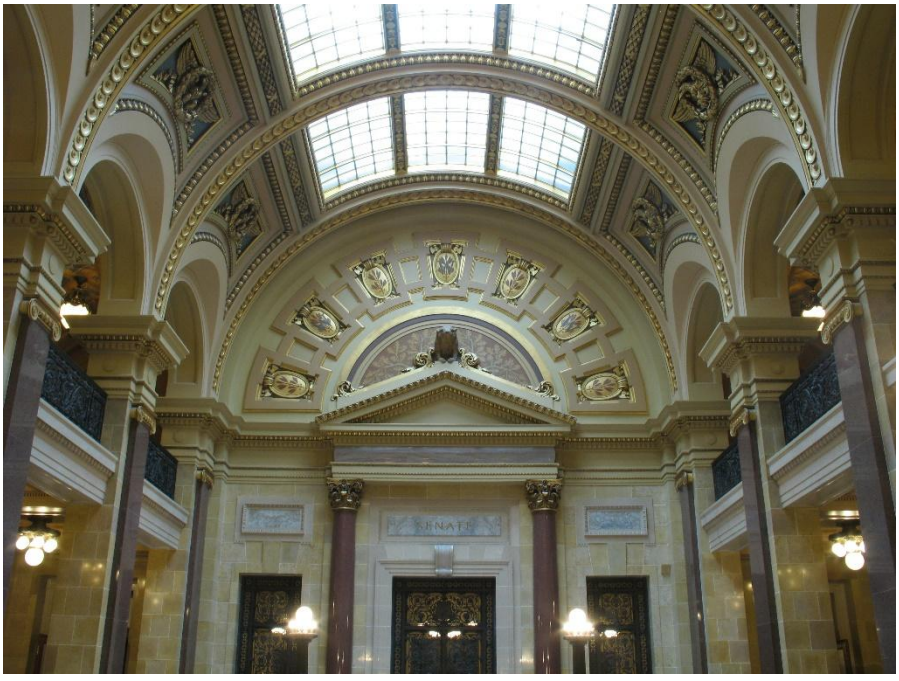

Figure 3: State Capitol Building – The 3rd scene

2. Write a program to assemble an HDR image

2.1 Import photos (Emily)

We read in the images read in the jpg images and store them in a four dimensional array. We also extract the exposure time from those images. We develop 1 MATLAB script:

- **readInImages.m**: reads in and stores images and exposure times

2.2 Sample the images (Emily)

To acquire enough pixels to extrapolate the image response function, we divide the image into a grid using parameters entered in the main MATLAB script. To ensure we get pixels from throughout the image, a pixel is randomly sampled from each square in the grid. For this step, we develop 1 MATLAB script:

- **sample.m**: samples and stores pixels throughout the input images

2.3 Solve for the image response function (Emily)

We develop a weight function. This, the log of the exposure times, and the sampled pixels are used in **gsolve.m** to discover the image response function. **gsolve.m** is run on each color channel**.** For this step, we developed the MATLAB script:

- **weightingFunction.m:** determines the weights to be used in gsolve

2.4 Write HDR image (Giang)

We implement the HDR algorithm in Paul E. Debevec's paper [1]. We use a set of images as input, then build a radiance map. Result is written to a Radiance RGBE file (.hdr). We develop 2 MATLAB scripts:

- **convertToHDR.m**: build a radiance map from a set of images
- **write_rgbe.m**: write HDR image to disk

3. Develop radiance map using tone mapping. (Giang)

We load our radiance map into Photomatix Pro 5.0, then tone map it into a usual image.

Figure 4: Image created by tone mapping our radiance map in Photomatix

4. Bonus

4.1 Image alignment (Giang)

For image alignment, we implement the MTB algorithm in [3]. After the image alignment, the positions of the pictures are the same. We tested the algorithm in 2 images:

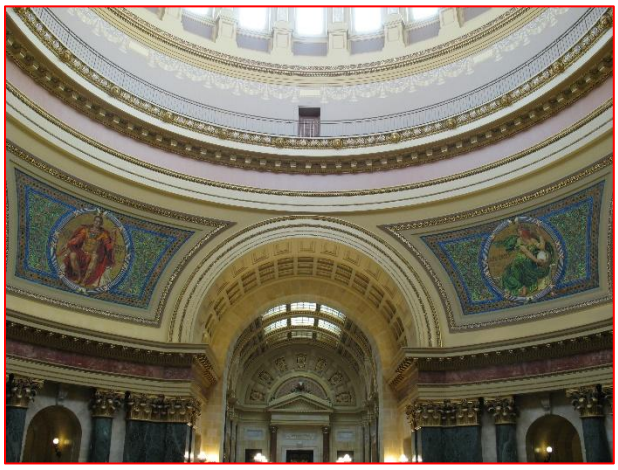

Figure 5a: Picture A before aligned

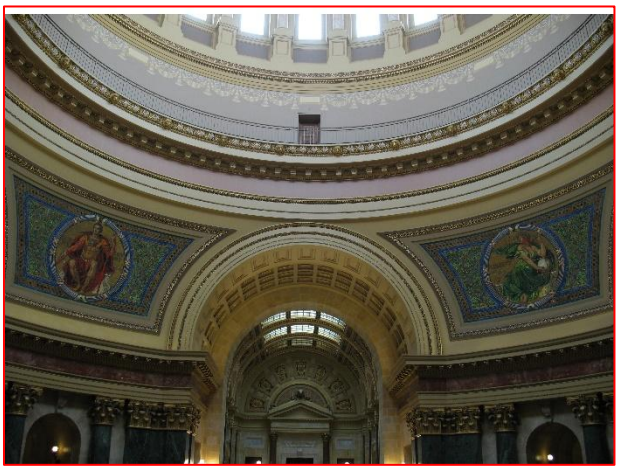

Figure 5b: Picture B

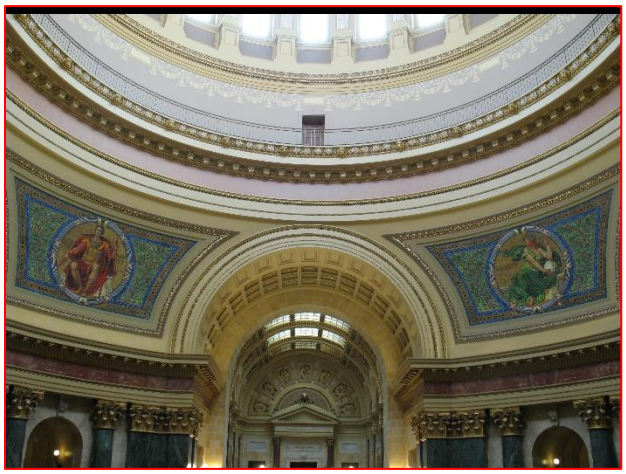

Figure 5c: Picture A after aligned

As we see, picture B has a white gap at its top (the white line). Hence, picture A is moved down after aligned (the black line). We develop 1 MATLAB script:

- **alignImages.m**: build a radiance map from a set of images

4.2 HDR tone mapping (Emily)

For tone mapping, we follow the algorithm described by Drago, Myszkowski, and Annen [2]. We ran it on the algorithm on scenes 2 and 3.

Figure 6a: the 2nd scene after tone mapping

Figure 5b: the 3rd scene after tone mapping

We implement the algorithm using following the MATLAB scripts:

- **dragoToneMapping.m**: the main script that implements Drago's algorithm on an HDR image by performing calculations and using other scripts
- **luminance.m**: calculates the luminance of every pixel the HDR image
- **logMean.m**: returns the logarithmic average of the luminance
- **changeLuminance.m**: updates the image with the new luminance values
- **gammaDrago.m**: gamma corrects the updated image

5. References

1. Paul E. Debevec, Jitendra Malik, Recovering High Dynamic Range Radiance Maps from Photographs, SIGGRAPH 1997.

2. F. Drago, K. Myszkowski, T. Annen, N. Chiba, Adaptive Logarithmic Mapping for Displaying High Contrast Scenes, Eurographics 2003.

3. Greg Ward, Fast Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures, jgt, 2003.