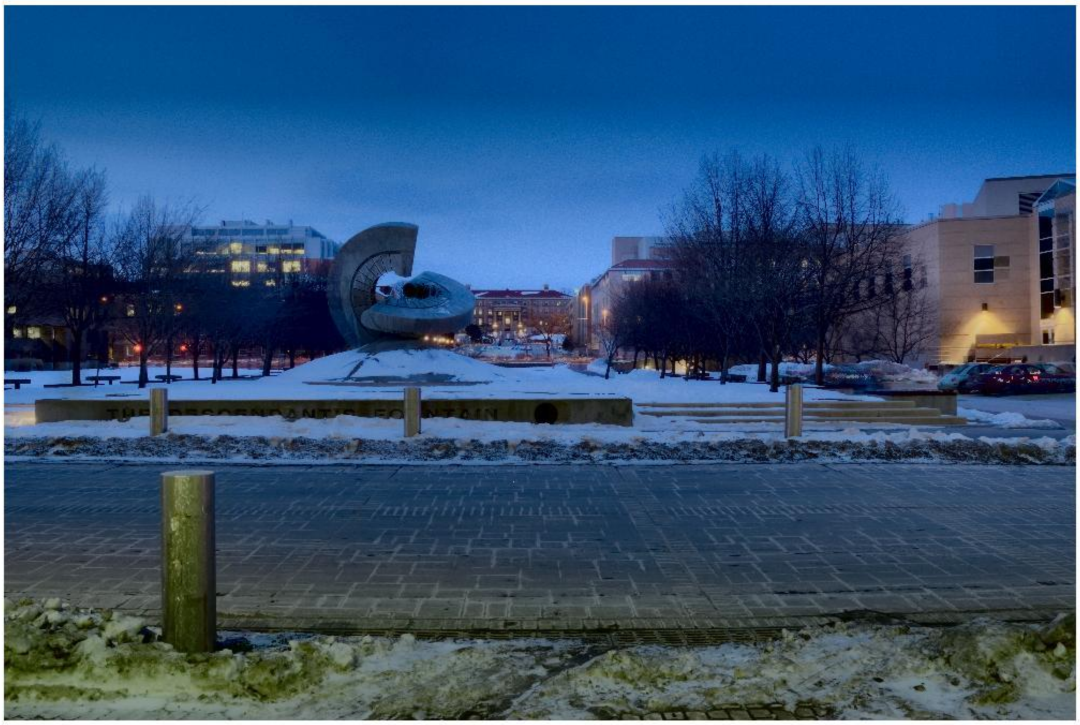
Assignment 1: High Dynamic Range Imaging

Anmol Mohanty

Rosaleena Mohanty



Assignment 1: High Dynamic Range Imaging

Anmol Mohanty

Rosaleena Mohanty

TABLE OF CONTENTS

1. **Project Description:**

   The ultimate goal of taking photographs on an imaging device is to try to match the scene to what a human eye is able to see and perceive. Using HDR (high dynamic range) images, as the name implies, a "dynamic range" can be added to the photographs. This is done by exposure bracketing of the same scene and combining the information from the different exposures, which then is tone-mapped to display it in the visible range. Here, our aim is to produce a single photograph from multiple inputs that would have a desired luminance in accordance with the human eye perception.

2. **Input Datasets**

   Our Dataset:
   The **Descendant's Fountain** in front of Engineering Hall with Univ Ave and Agriculture hall in the background, during dusk

   Image Acquisition:
   Camera Model: Nikon D3300
   F/Stop = f/20. ISO – 3200
   Number of Exposures: 15 (1/3200 -> 15s)

   Other Datasets: In addition to using the acquired pictures, we also used downloaded datasets from the web that have been used for HDR analysis.

3. **Implementation Details**

   All algorithms were implemented on MATLAB.
   The basic algorithm is based on the Debevec's algorithm with the following steps in order:

   a. Image acquisition
   b. Solving for of camera response
   c. Producing the radiance map
   d. Tone-map algorithm

4. Bonus

a. Bilateral Tone-mapping

In order to display the image, we need to implement a local tone-mapping algorithm. We used a version of the algorithm presented in Durand 2002. The steps as follows:

- Input is the RGB values of radiance
- Compute the intensity used a weighted average
- Compute the chrominance channels
- Compute the log of the intensity
- Apply bilateral filter and compute the detail layer
- Apply an offset and a scale to the base
- Reconstruct the log intensity
- Put the colors back
- Apply a gamma compression

b. Exposure Fusion

An interesting approach that aims at producing a single well-balanced image from multiple exposures in the exposure fusion method**.** This is done without computation of camera response. Although the output image is high quality and captures details from all the exposures, it cannot be considered an HDR image since there is no computation of an intermediate radiance map. This could be considered as an advantage over HDR as no radiance map implies no need of tone-mapping, thus, making the algorithm faster. , The demerit is that this method cannot extend the dynamic range of the original pictures, but produce a well-exposed image for display purposes. The code implemented was found to be based on the paper on exposure fusion, referenced at the end.

Steps involved are:

- Decompose input images into Laplacian pyramids (band-pass filter at multiple scales)
- Gaussian pyramid of the weight maps is created
- These maps are blended together channel-wise to form a fused pyramid
- Final pyramid is collapsed to obtain the fused image.

.

c. Image Alignment

Please note that we have been able to implement this part partially only. But the process yielded a lot of learning and also we were able to salvage lot of useful functions like the threshold_bitmap_computer, XOR, ADD, bitmap_shifter, RGB2gray convertor and a way to find the difference between various offsets. These files have been accordingly attached with the project source code in the image_align directory as a standalone program.

Image alignment is the process of aligning the input source images with each other to compensate for any shake while operating the camera. It is observed however that the Debevec algorithm is fairly robust and is in general immune to small translations or rotations while capturing the image.

Brief Description: Essentially the program takes 2 input images and gives how much the offset of 1 image (current) is from the other image (reference image). The algorithm implemented is the ward's algorithm the brief overview of which is presented below. The input images are converted into gray scale (imp as the program would be efficient running on 1 channel only), then the median/percentile bitmap image is computed by averaging; this enables the code to leave out very minor changes in edges and hence operate much faster without sacrificing accuracy.

The file image_alignment is the core file which reads the images and calls the wrapper to the shifter program which can be used to passed any other programmable values to the core shift computer file called Actualshift_computer.m which computes bitmap and does the necessary processing (again please note that this portion wasn't completed fully, the program yields spurious values of shifts and needs more work).

Original Image

Grayscale variant

Some intermediate outputs: - (Median threshold bitmap image)

MTB of original image

MTB of original image

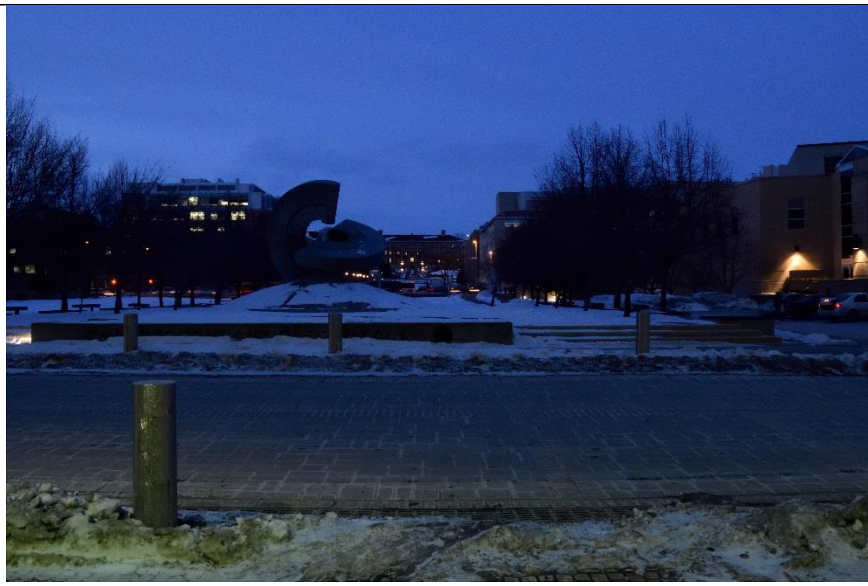
MTB of original image

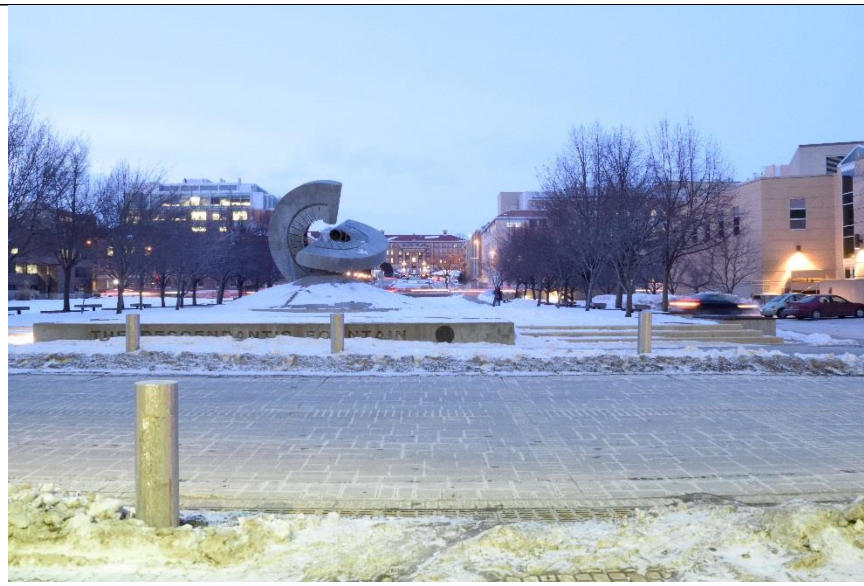
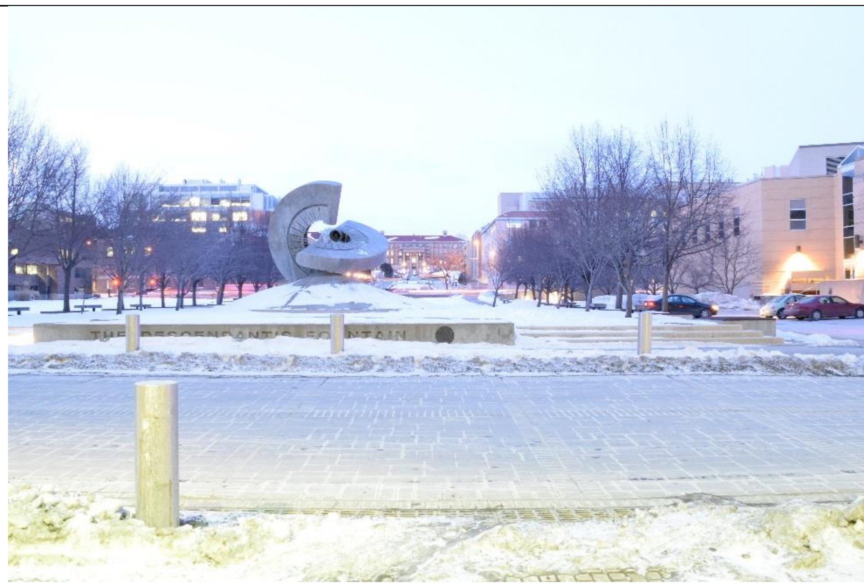
Final reconstructed average image of the scene.

5. Results and Comparison

5.1.Results

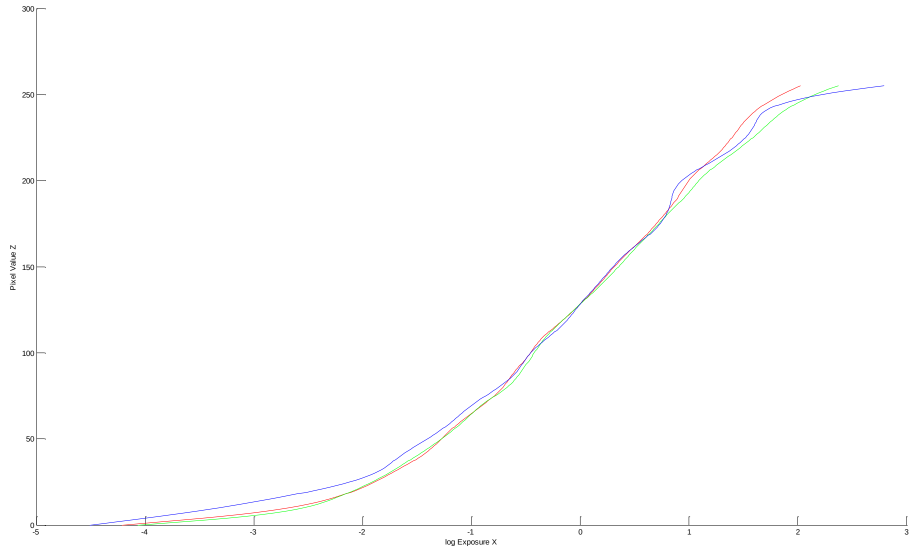
Step 1: Image Acquisition

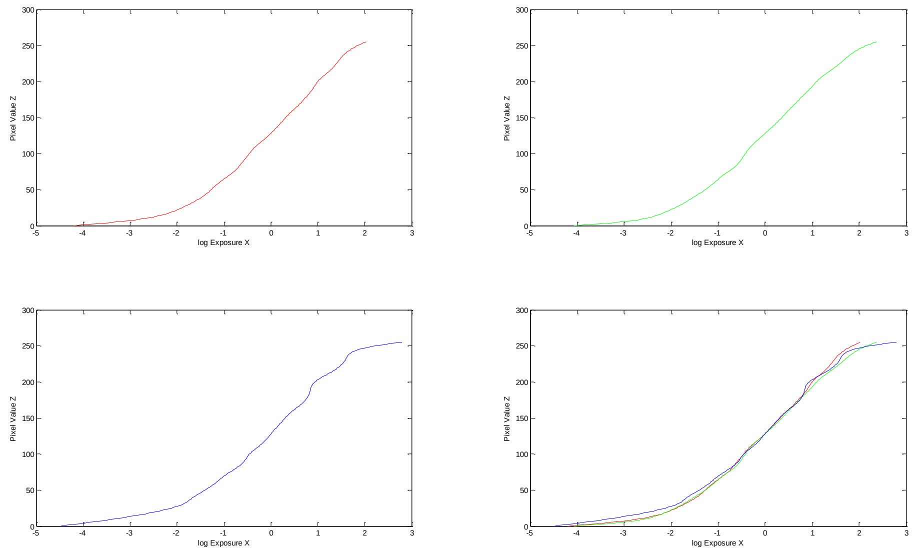
Exposure Time (seconds)	Input Image Acquired (only 5 shown here)
1/50	
1/8	

1	
2	

4	

Step 2: Computing Camera Response Function:





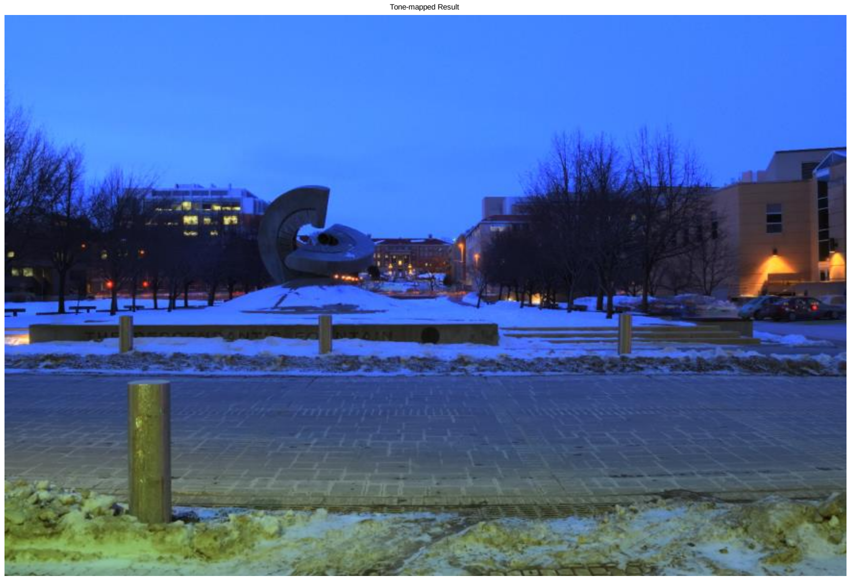
Using Debevec's equations, the camera response for each channel are shown above separately as well as plotted together. Depending on the smoothing parameter 'lambda', a smooth or wiggly response was obtained.

Step 3: Producing the Radiance Map:

HDR Radiance Map

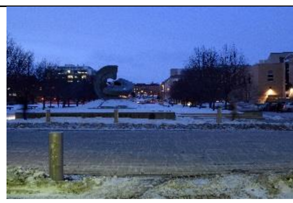
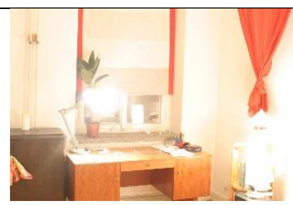
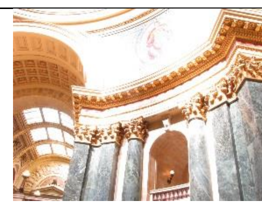
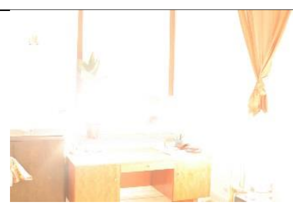
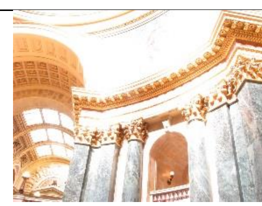
Step 4: Tone-mapping

Tone-mapped Result

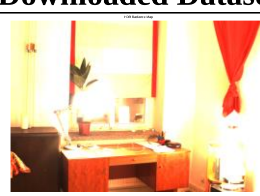
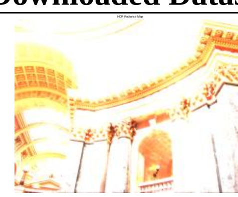


5.2 Comparisons

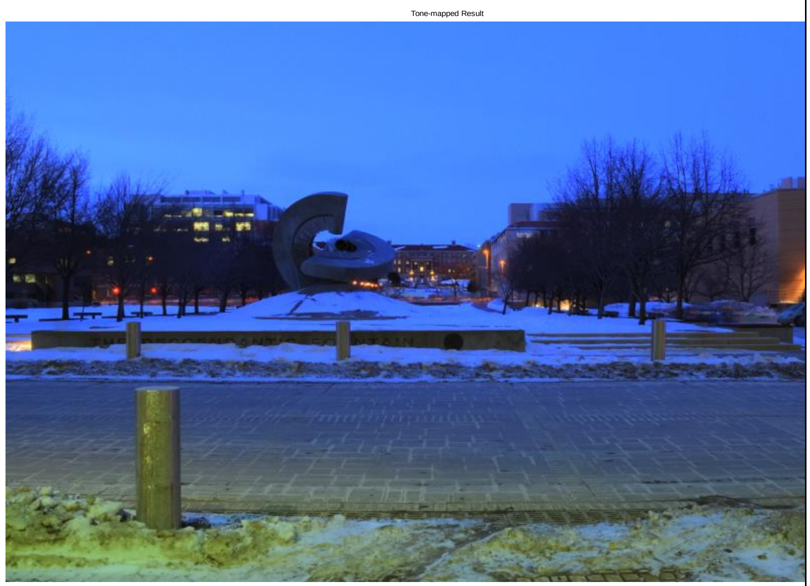
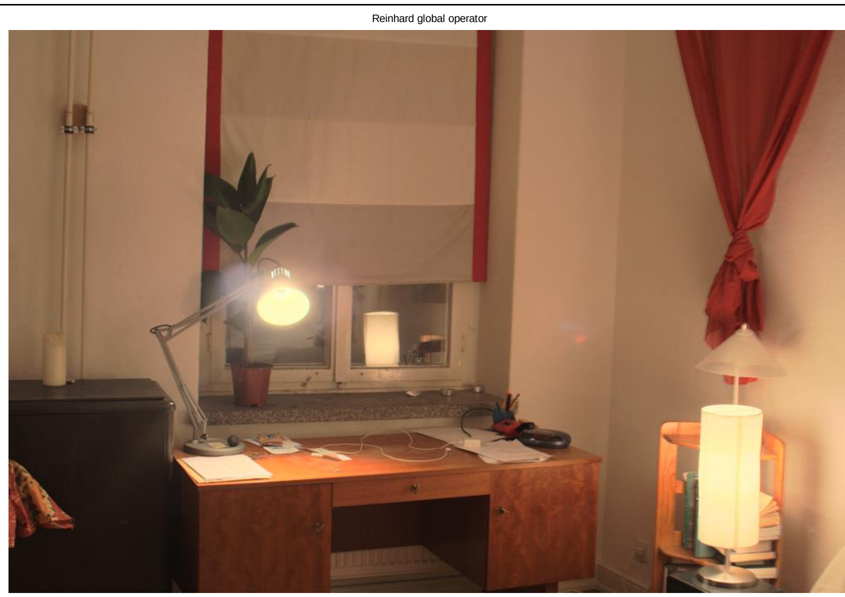
a. Datasets used for comparison

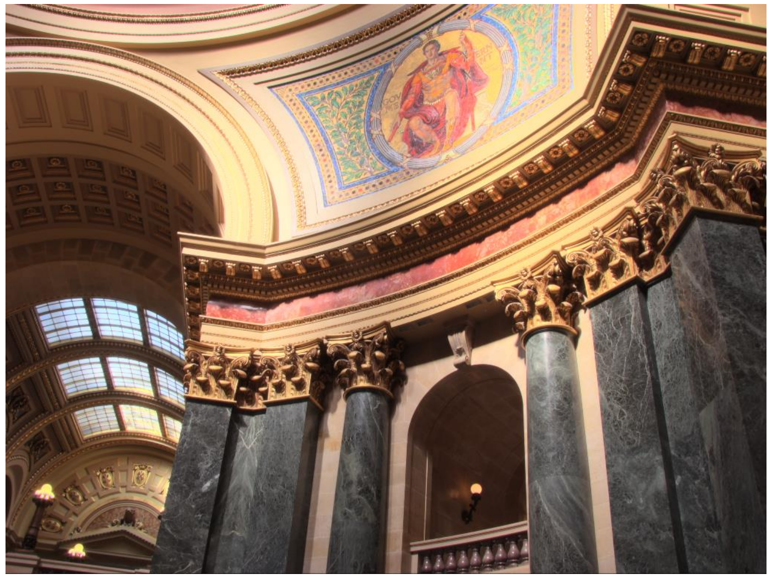
Our Dataset	Downloaded Dataset	Downloaded Dataset
		
		
		
		
		

b. Radiance Map obtained from different datasets

Our Dataset	Downloaded Dataset	Downloaded Dataset
		

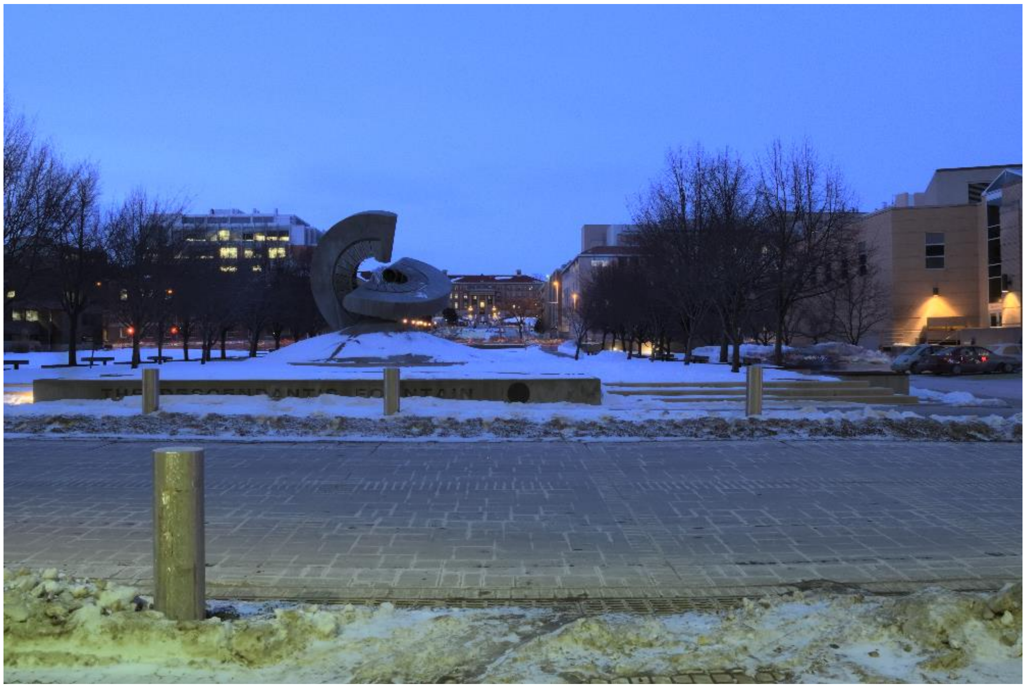
c. Tone-mapped HDR image

Our Dataset	
Downloaded Dataset	

Downloaded Dataset	Reinhard global operator 

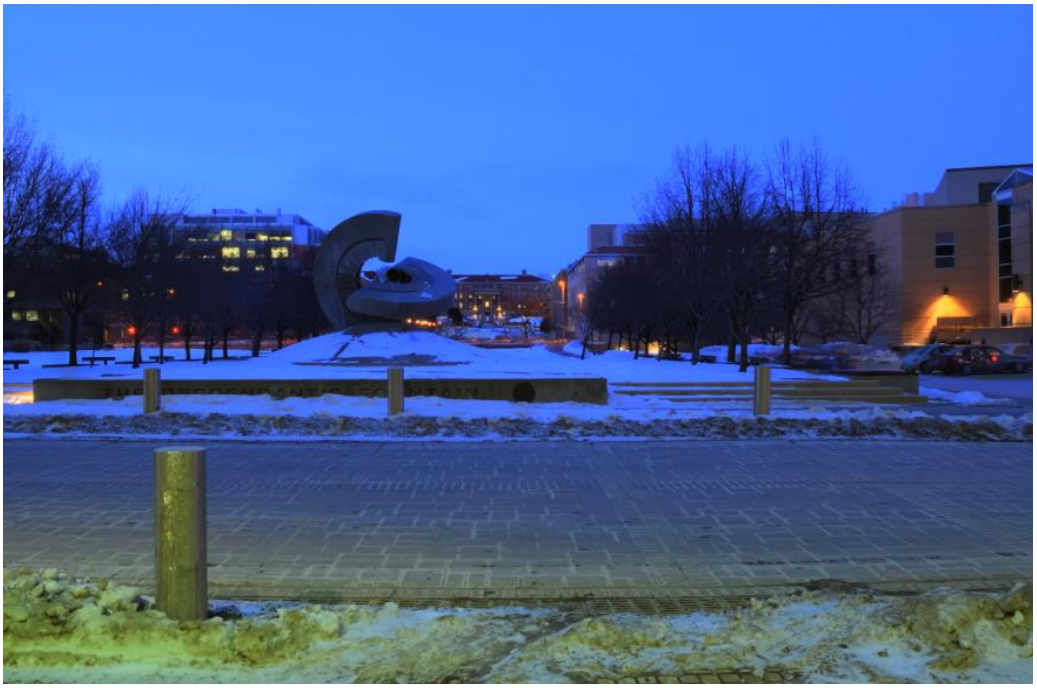
e. Tone-mapped output from different methods:
   Reinhard tone-map

Reinhard global operator



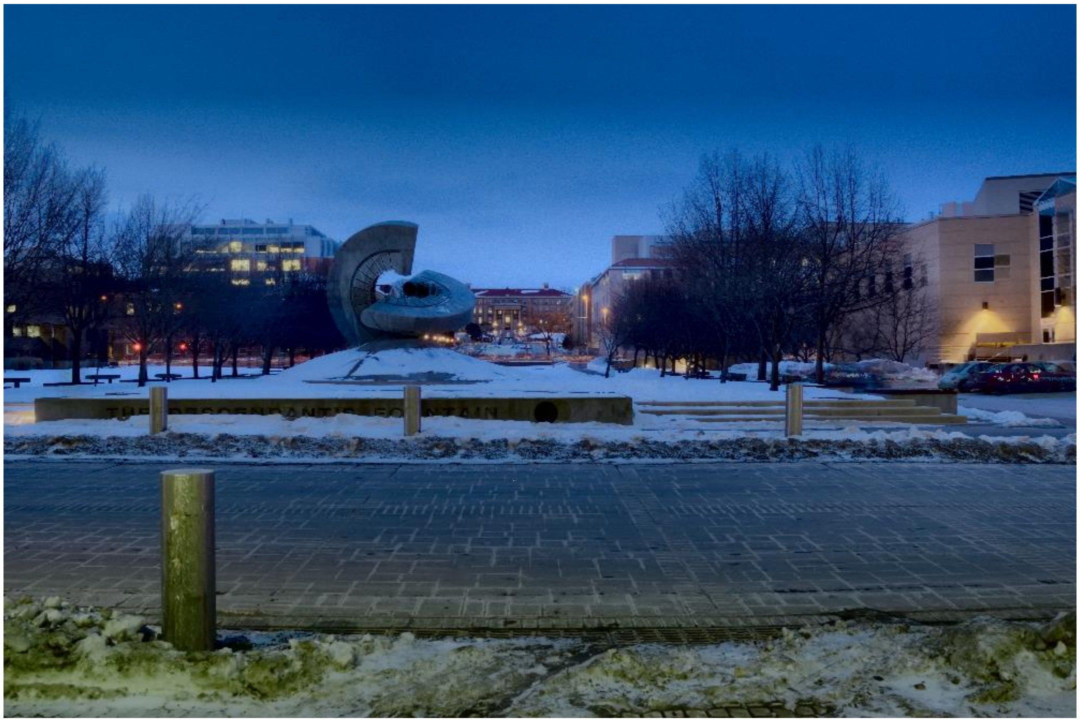
Bilateral tone-map

Tone-mapped Result



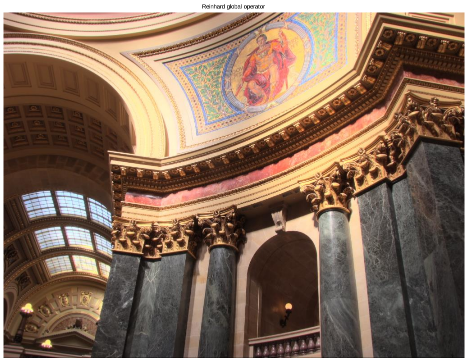
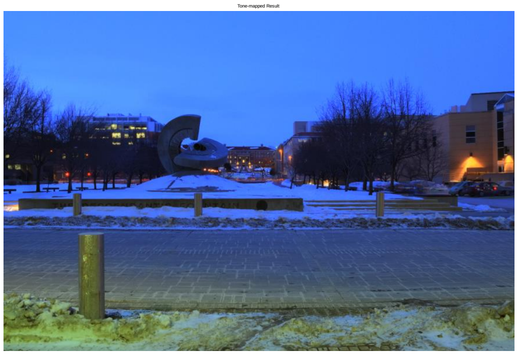
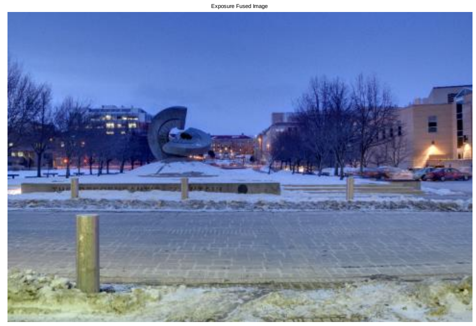
In-built MATLAB tone-map

Ib-built MATLAB tonemap



f. Comparison between tone-mapped result and exposure fusion:

While HDR images are more pleasing by covering a dynamic range, exposure fused images look more realistic.

Debevec's Algorithm	**Exposure Fusion**
Reinhard global operator 	Exposure Fusion 
Reinhard global operator 	Exposure Fusion 
Tone-mapped Result 	Exposure Fused Image 

6. Miscellaneous learning

We had a lot of frustrations, setbacks and epiphanies while working on this project. It was an intensely challenging and yet immensely fun learning experience. This is a summary of some of the miscellaneous things we learnt from the project.

1. Never trust your hands despite how deft or cautious you are. We took the photos on a DSLR mounted on a tripod and despite our best attempts to be careful there was some noticeable mis-alignment in the series of images that we took.

2. So we moved to the USB controlled setup as suggested in the project page. The software used was 'digicamcontrol' which is an excellent, if somewhat confusing piece of program. It has a host of features including a live view-finder which allows one to sample the image before actually capturing it. The latency of the LVF was very good and usable.

3. One mistake which we made was we disabled the option to take both RAW and JPG photos which was offered by the software, and hence what we were left with were Nikon's proprietary .NEF files (Nikon Electronic Format) which were unusable in Matlab. We tried converting them to JPG using free off-the shelf convertors but were disappointed with the outputs which seemed to have a unmistakable pinkish tinge. We then had to go for the full Nikon Software package for Nikon DSLR's which is a huge piece of software and we ended up losing valuable time in the process.

4. Another interesting issue we bumped upon and resolved quite accidentally was the naming convention of the files which changed our program's state to an undesirable value. Our file names were supposed to contain the exposure value in filename*_Nr_Dr format in which the exposure value would be Nr/Dr in seconds. We faithfully appended the _Nr_Dr combo to the end of the file names generated by the DSLR but unfortunately the DSLR nomenclature was of the form DSC_536*.jpg. So the exposure values which was being computed for individual images was 536/Nr instead of Nr/Dr which led to most of the exposure values around the 536 mark(since Nr of exposures are mostly 1) and for quite some time we felt that our algorithm had gone bad. Fortunately by debugging using the code involving radiance image, tone map, and individual camera response functions (which looked very weird by the way) we were able to nail down the issue and subsequently rectify it.

5. Another learning point was perhaps using the given camera would have been a wiser choice, because you see going with the DSLR in hope of taking better shots we ended up with relatively super huge image files and gargantuan resolutions and it took Matlab quite a while to run the full algorithm on our image set. So we had to downscale the image to lower resolution to run faster. Professor Singh's advice that camera size doesn't matter and smaller size would actually be ideal turned out to be true.

6. We thought we had the perfect image for shooting, but unfortunately the environment in which we shot wasn't ideal as there were some cars moving in the background with their headlights on. But the algorithm turned out to be robust and no artifacting was visible in the final output images.

7. A final learning point was that if the conditions are dark the camera's output would artificially interpolate to fill in the dark areas with a post-processing function which might not be ideal. For example in our case, we went in evening and the camera was aggressively filling in the sky (which was mostly dark by this time) with an artificial blue.

7. Individual Contributions

Anmol Mohanty: Image acquisition, image alignment implementation, miscellaneous learning, final write-up

Rosaleena Mohanty: Basic Debevec's algorithm, bilateral tone-map, exposure fusion, final write-up

8. References

HDR:
Debevec, P. E., and Malik, J. 1997. Recovering high dynamic range radiance maps from photographs. In Proc. ACM SIGGRAPH 97, T. Whitted, Ed., 369–378.

Reinhard Tonemap:
Erik Reinhard, Mike Stark, Peter Shirley and Jim Ferwerda, 'Photographic Tone Reproduction for Digital Images', ACM Transactions on Graphics, 21(3), July 2002 (Proceedings of SIGGRAPH 2002).

Bilateral Filter tone-map:
Durand, F., and Dorsey, J., 2002. Fast bilateral filtering for the display of high-dynamic-range images. ACM Transactions on Graphics, 21(3), pp. 257-266.

Exposure Fusion:
T. Mertens, J. Kautz, and F. V. Reeth. Exposure fusion. In PG ’07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, pages 382–390, Washington, DC, USA, 2007. IEEE Computer Society.

Datasets:
http://pages.cs.wisc.edu/~csverma/CS766_09/HDRI/hdr.html