**PBI Report**

**PDPM IIITDM Jabalpur**

**Internship at Canon India Pvt. Ltd**

**Topic - High Dynamic Range (HDR) imaging and Tone mapping algorithms**

**External Mentor – Radhish Ayyappan**

**Internship Duration – 16th May to 15th November**

**Report for Duration – 16th May to 15th June**

|  |  |
| --- | --- |
| **From** | **To** |
| **Ravi Jain** | **Dr. Ayan Seal** |
| **B.Tech, 3rd year**  **Computer Science and Engineering** | **Assistant Professor Computer Science and Engineering** |
| **2013167** |  |
| **2013167@iiitdmj.ac.in**  **ravithats.it@rediffmail.com** | **ayan@iiitdmj.ac.in** |

**HDR Image Generation –**

The task of this project is to deal with the **computational bottlenecks of HDR imaging**.

We first understand how an HDR image is generated.

Libraries of **pfstools** have been used in order to handle HDR images.

pfstools package is a set of command line programs for reading, writing and manipulating high-dynamic range (HDR) images and video frames. It includes also Qt and OpenGL HDR image viewers.

pfs in not just another format for storing HDR images, but is rather an attempt to integrate the existing HDR image formats by providing a simple interface for exchanging data between applications.

pfstools is a base set of tools and more advanced functionality can be found in related packages, such as pfstmo(tone-mapping) or pfscalibration (recovery of camera response curve and merging multi-exposure LDR image).

**Main features of pfstools –**

* pfstools try to handle properly colorimetry of images, not neglecting physical meaning of color data.
* The filters can work on a single image as well as on a sequence of frames.
* It includes a set of loaders and savers for most of the popular HDR file formats. There is no need to write one's own loader for Radiance's RGBE or link with several libraries to load OpenEXR images.
* Because of its modular architecture, owning to use of UNIX pipes, the suite of tools is quite flexible. For example, to load an HDR image, tone map it, apply gamma correction and save it as a jpeg file, one can issue:
  + pfsin memorial.hdr | pfstmo\_drago03 | pfsgamma 2.2 | pfsout memorial.jpeg
* pfs files can contain not only color information but also additional channels, like depth map, flow field or alpha channel.
* Integration with GNU Octave and matlab. It includes functions to read and write HDR frame from/to Octave/matlab. This way, it's easy to write one's one tone mapping operator using a high level math language.
* It includes a convenient viewer for HDR images and any data that fits into pfs files.
* Its stands for portable floatmap streams.
* C++ is the programming language of this API
* All programs in pfstools package are named to facilitate a tab-completion in a UNIX shell. Therefore the names of programs always start with 'pfs' prefix. Then follows a name of the program or name of the program group followed by the actual name of the program. Some example of groups are: pfsin\* for reading files, pfsout\* for writing files, and pfstmo\* for tone mapping.

These tools can be found at <http://pfstools.sourceforge.net/>

The initial task was to understand use of these tools and install this library on my system. To understand basics of HDR images I went through the High Dynamic Range videos of course <https://classroom.udacity.com/courses/ud955/lessons/3567488563/concepts/34981591450923>

In this process I learnt a few **LINUX commands** –

**CMake** - CMake can handle in-place and out-of-place builds, enabling several builds from the same source tree, and cross-compilation. The ability to build a directory tree outside the source tree is a key feature, ensuring that if a build directory is removed, the source files remain unaffected.

**yum** - Yellowdog Updater Modified

**grep** - Used to search for lines of text that match one or many regular expressions, and outputs only the matching lines.

**insmod** - install loadable kernel module

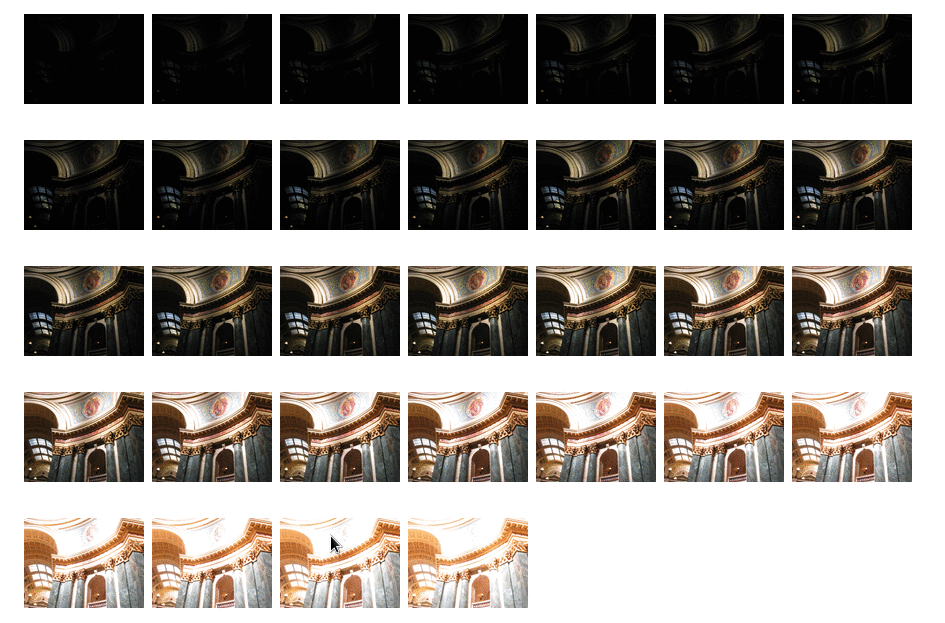
After the installation of these libraries was complete, the following details were considered.

* **Image Set for generation of HDR image**
* **How pfstools can be used to** 
  + **Estimate Response Curve**
  + **Generate HDR Images**
* **Tone mapping**
* **OpenCV vs pfstools**

**Image Set**

* Standard Image Sets were used, these include images with varying exposures.
* Images containing interior parts of Madison Capitol were considered.
* Furthermore high definition images of MacBeth Color Checker and Toy Shuttle were taken into account.
* Six such different image sets have been operated upon.

**Sample Image Set (1)**



**Sample Image Set (2)**



**HDR Imaging**

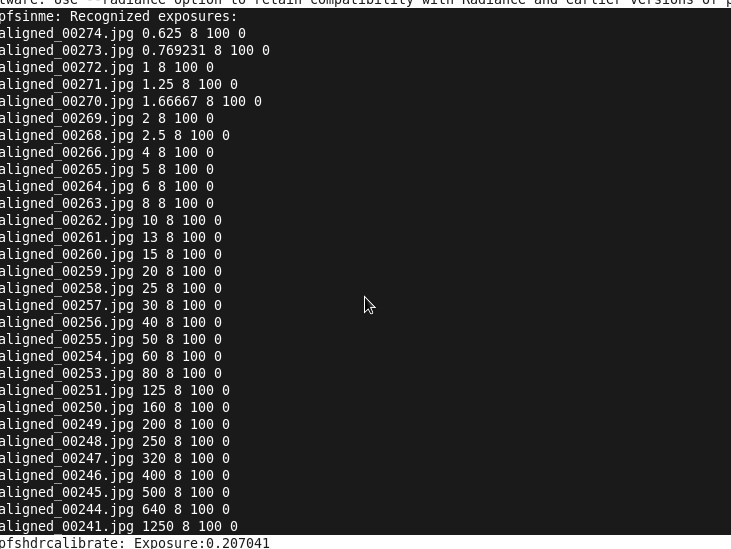
* Using certain pfstools commands these 'jpg' format images are converted to hdr image.
* pfshdrcalibrate − Create an HDR image or calibrate a response curve from a set of differently exposed images supplied in PFS stream.
* pfsinme \*.JPG | pfshdrcalibrate -v -s response.m | pfsout xyz.hdr
* Recover the response curve from set of all JPEG files in the current directory and save it to response.m file.
* This command considers the exposure time, aperture value, focal length and other properties of each image in order to generate an HDR picture.

**Command Execution**



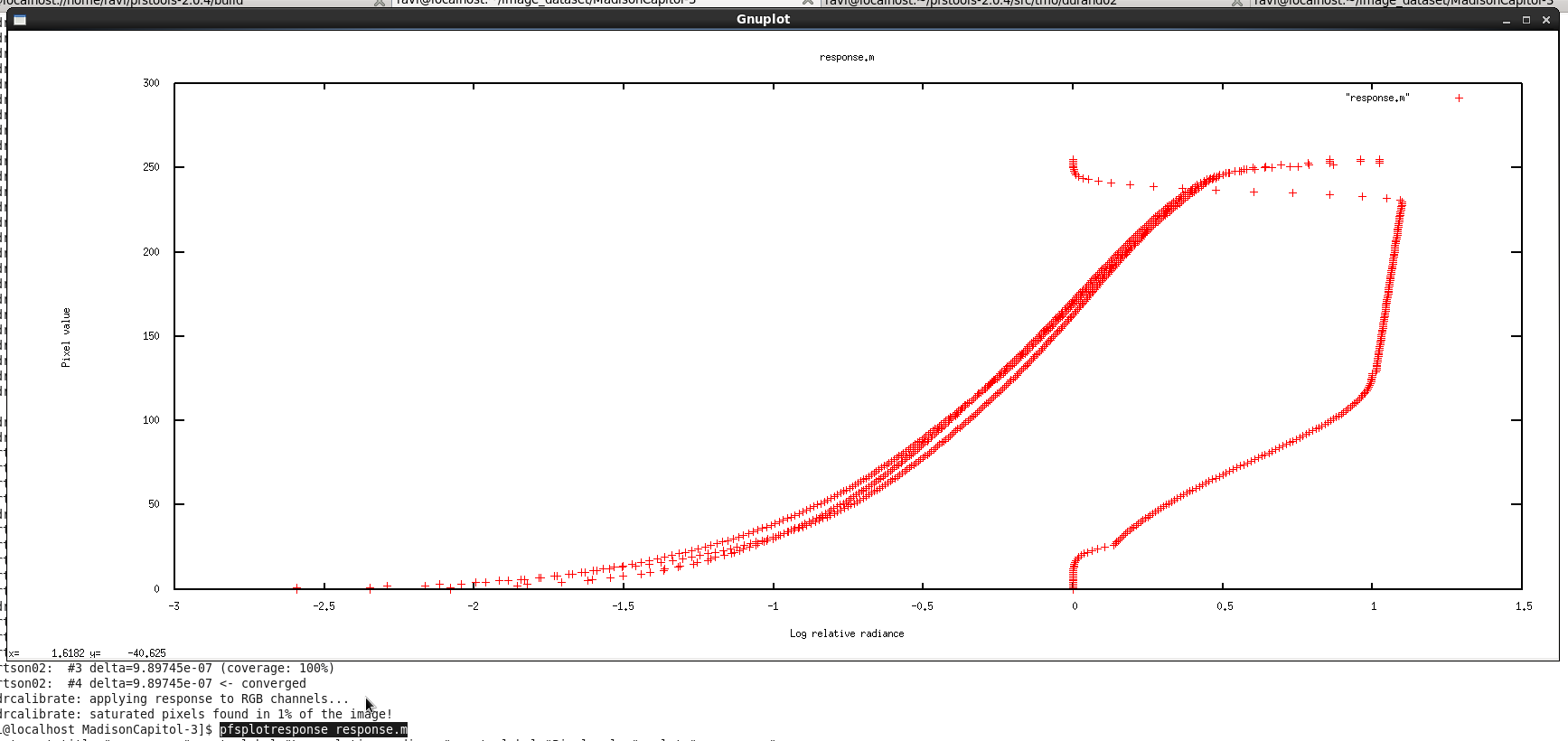
* This command considers jpg format files, generates response curve 'response.m' and saves the hdr image in 'xyz.hdr'.

**Image Recognition**



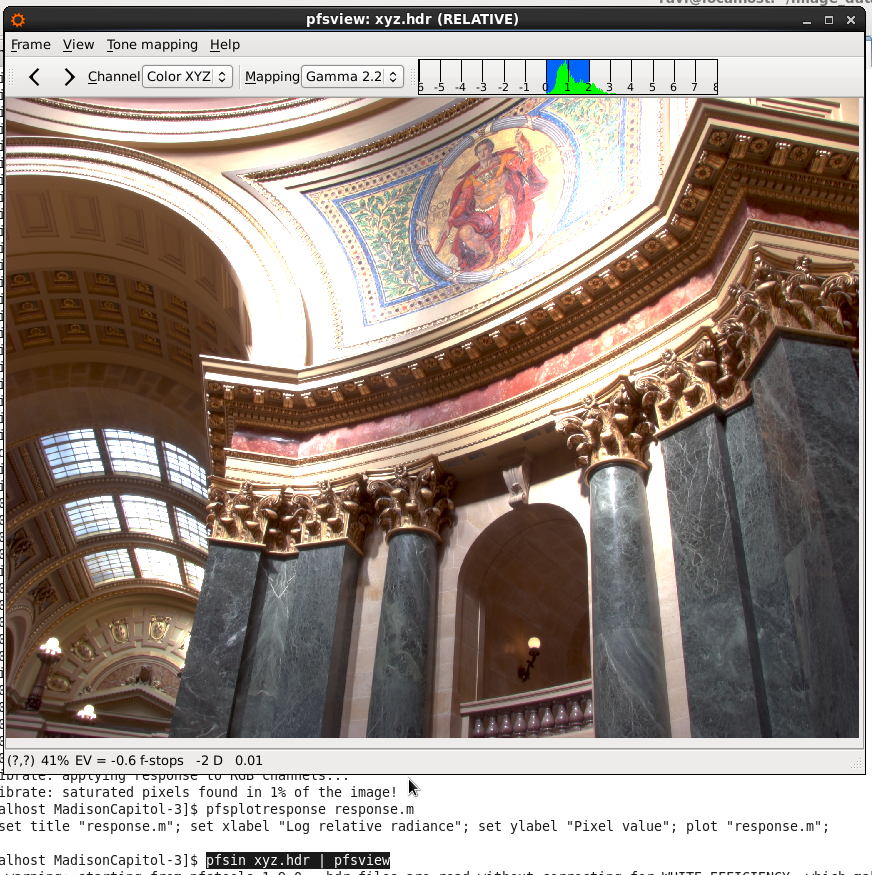
* We see the images being processed as their exposure times are considered to calibrate them into an HDR.
* The above screenshot displays image name, exposure time, f-number, ISO speed ratings and contrast values of each image in the respective order.

**Estimating Response Curve**



* The above screenshot is obtained when the response curve is plotted for the generated HDR image.
* The three almost overlapping lines represent Red, Green and Blue.
* The plot considers log relative radiance vs pixel values.
* pfstools command used is ‘pfsplotresponse response.m’

**Resulting HDR Image**



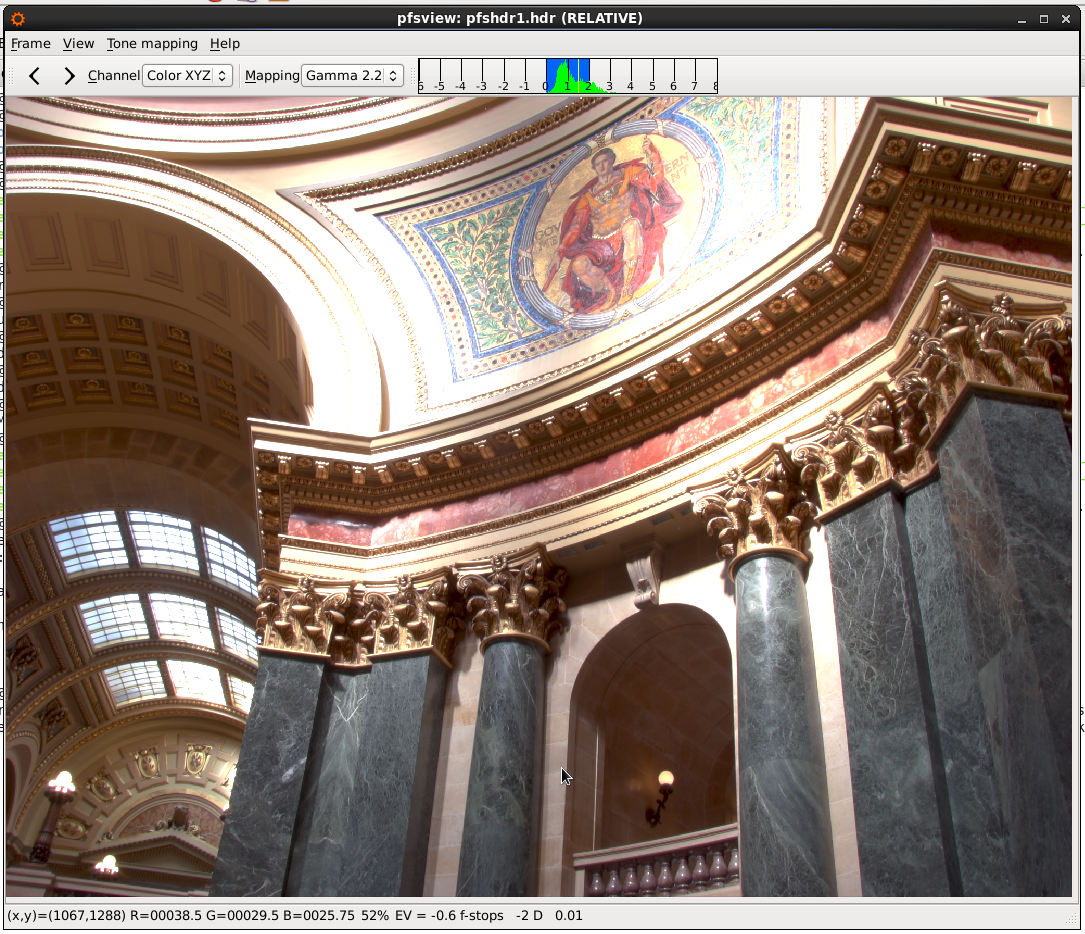
* As a result of the above calibration, this HDR image is generated.
* Its luminance range can be viewed in the image.
* This is the HDR image for Sample Image Set (1).
* The command used to view this image is ‘pfsin xyz.hdr | pfsview’ where ‘xyz.hdr’ is the name of this file.

After the generation of HDR image, next task was to tone map this image and generate a lower resolution png image, which can be mapped to 256 pixel values.

**Tone Mapping**

* Tone mapping is a technique used in image processing and computer graphics to map one set of colors to another to approximate the appearance of high-dynamic-range images in a medium that has a more limited dynamic range.
* Here we use durand tone mapping operator in order to generate a png image from the HDR picture.
* We analyze the time taken by pfstools in order to do so.

**Sample HDR to png conversion**

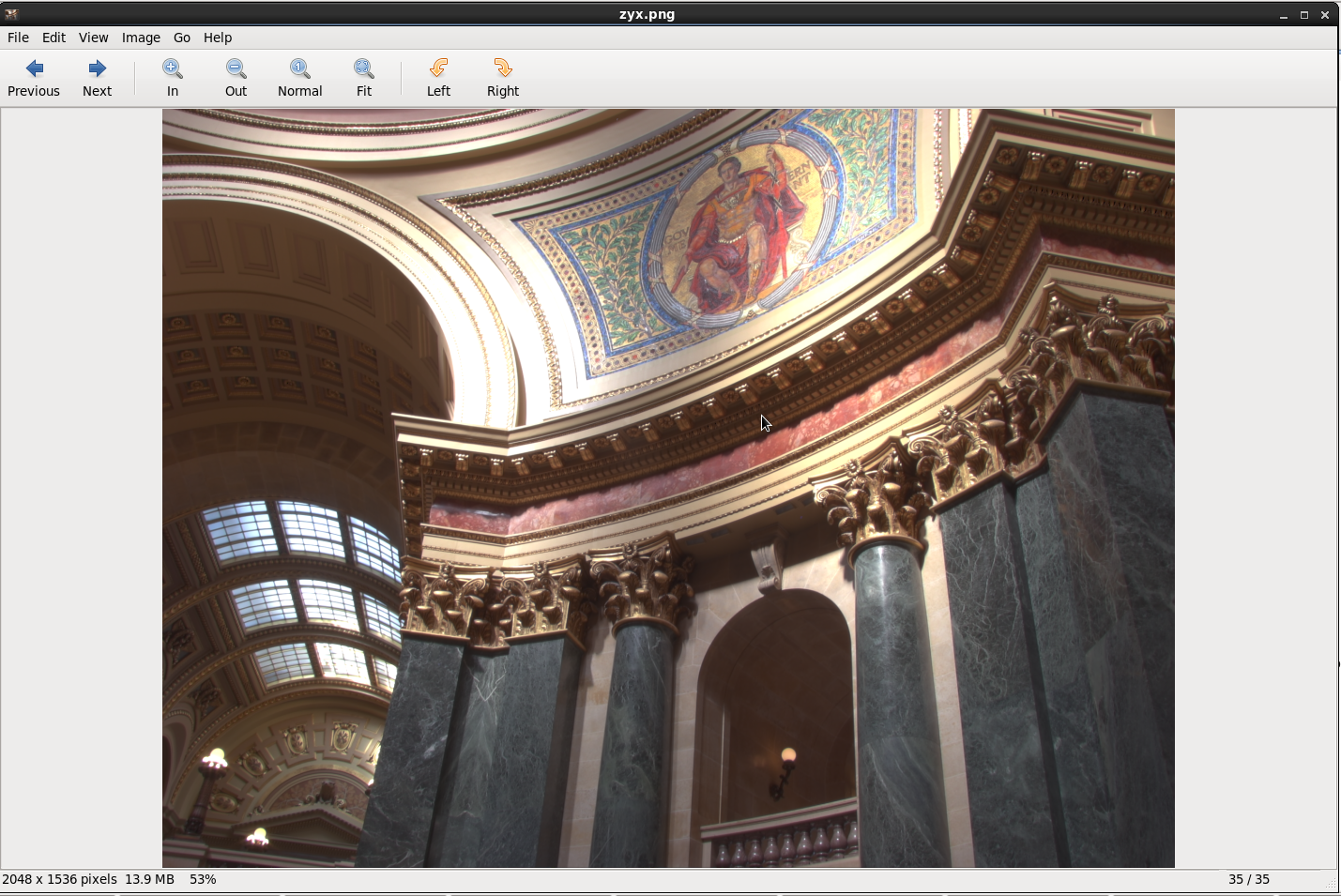
Madison Capitol HDR image

**Command Execution**



* This command takes in as input the above mentioned hdr image 'pfshdr1.hdr', applies durand tone mapping, gamma correction and gives 'zyx.png' as output.

**Tone Mapped Image**



* This is the resulting lower resolution tone mapped image.

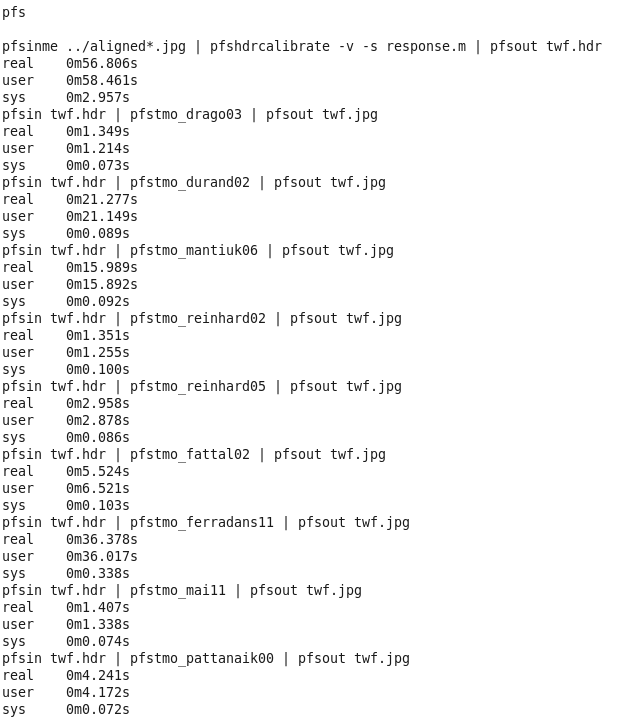
**Comparison**

* We take a set of jpg images and apply OpenCV , pfstools commands in order to compare the two.
* The standard image set described above considering the interior of Madison Capitol has been used.

**OpenCV vs pfstools**



* This image describes the time taken by OpenCV commands for different purposes, for example to compute response curve is 9.408 sec, for exposure fusion is 13.737 sec etc.
* Several tone mapping styles such as drago, durand, reinhard and matiuk are performed and their computation time is noted.



* This image describes the time taken by pfstools commands for different purposes, for example for drago tone mapping is 1.349 sec, for durand tone mapping is 21.277 sec etc.
* Several tone mapping styles such as drago, durand, reinhard , matiuk, pattanaik, mail, ferradans and fattal are performed and their computation time is noted.
* The list of pfstools commands can be found at http://pfstools.sourceforge.net/man\_pages.html

**Observations**

* pfstools take a lot more time to calibrate and generate an HDR image.
* Tone mapping in pfstools using drago, reinhard operators is much faster.
* pfstools supports more methods of tone mapping than OpenCV.

**Tools Used**

* OpenCV
* pfstools
* ImageMagick
* QT viewer for displaying hdr images
* OpenGL

**Conclusion**

* We successfully generate HDR image from a set of jpg image files using pfstools.
* The HDR image was then converted into a lower resolution image by tone mapping.
* **The next steps are to observe what features in these tone mapping algorithms of pfstools take more time and to reduce their time complexities in order to achieve faster results which I will be working on in the coming days.**

**Python Programming Task**

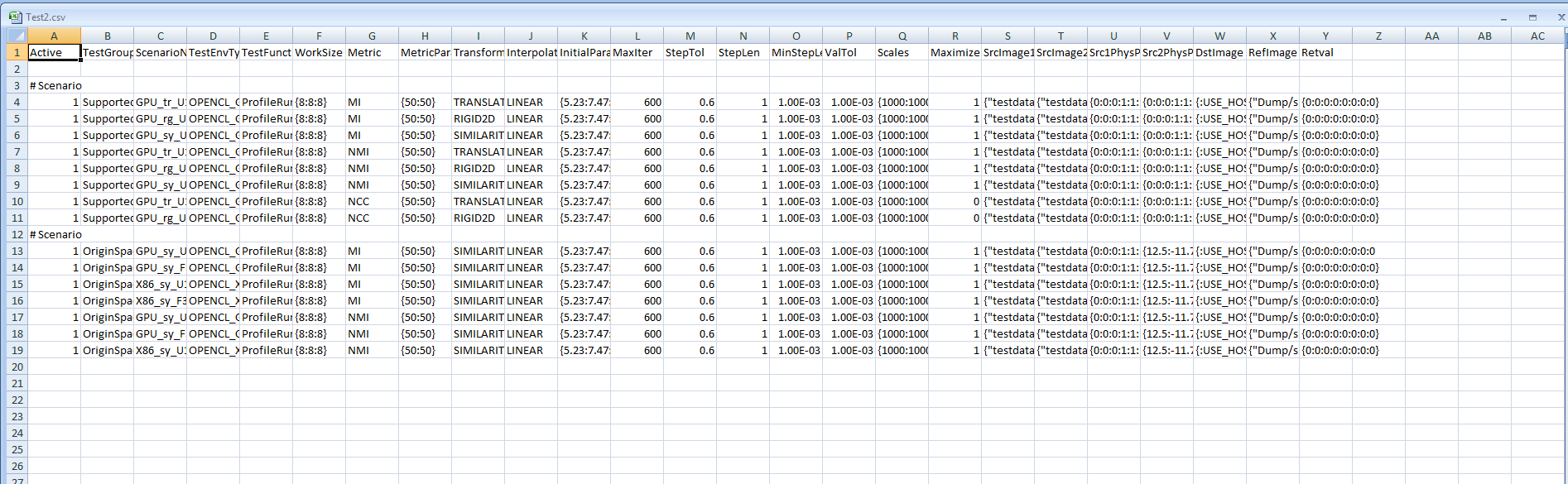
During the first week of internship I was given a task to write a program in python, which would

* Take a csv (comma separated values) file as input that contains information related to different scenarios.
* Create separate csv file for each scenario.
* Create a excel file which would contain embedded objects.
* These embedded objects will be linked to the generated csv files and display them on different excel worksheets.

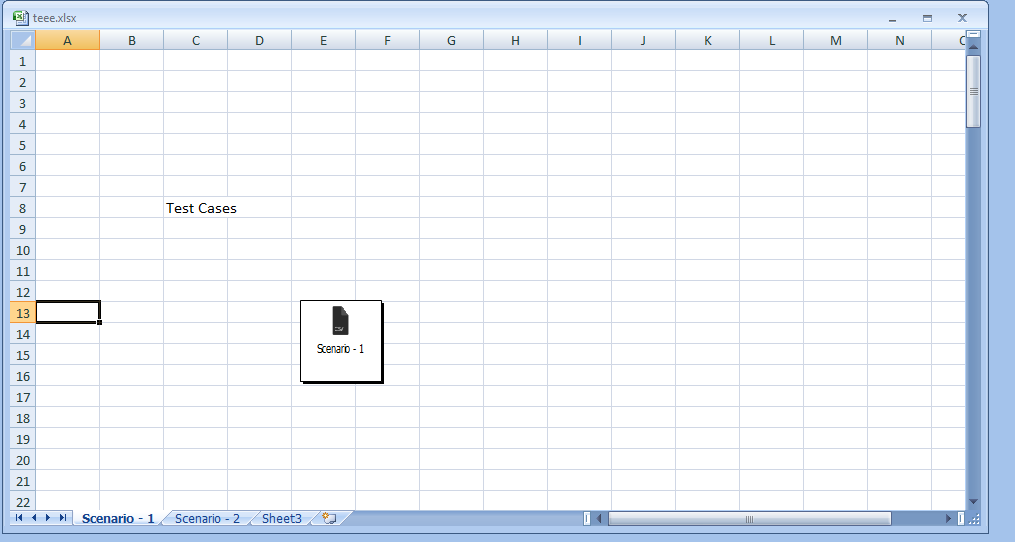
The Code written for this task is uploaded at <https://github.com/ravi2013167/coursera-site> with a file named final.py

The Screenshots of the output obtained are described below.

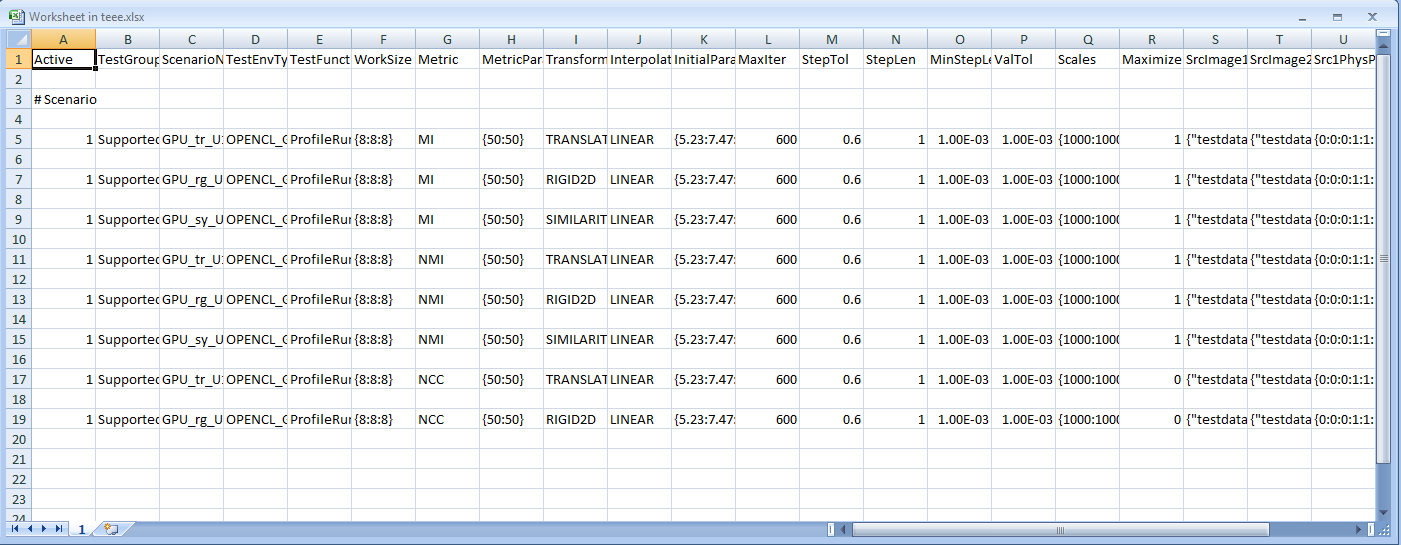
**Input csv file**



**The output excel file generated with embedded object**



**Preview of embedded object link (contents of a particular scenario)**



**Tools Used**

* Python-3.5.1
* MS Excel
* Sublime Text Editor

**References**

* http://pfstools.sourceforge.net/index.html
* http://pages.cs.wisc.edu/~csverma/CS766\_09/HDRI/hdr.html
* http://eidomatica.di.unimi.it/index.php/research/idb/yaccd2
* http://docs.opencv.org/3.0-beta/doc/tutorials/photo/hdr\_imaging/hdr\_imaging.html
* http://www.stackoverflow.com
* https://docs.python.org/3/library/csv.html