Michael Doescher
Eugene Yoong
Due Thursday February 19, 2015
Computer Vision: Project 1

High Dynamic Range Imaging

High dynamic range imaging is the process of taking many images captured at different exposure levels and combining them to generate one image which better matches what human perception would generate when viewing the same scene. Cameras have limited dynamic range in the intensities of light that they are able to capture, this means that light of high intensity reaching the detector may saturate the detector; and therefore, light of any intensity exceeding the maximum threshold of the camera detector will all map to the same signal. The same process occurs for light of low intensity, and the camera is unable to capture contrast variations in regions of very dark or very intense light exposure. Additionally, most camera detector elements exhibit a nonlinear response to incident light so that changing the light intensity does not result in a proportional change in detector output. Traditional photography allows users to set the shutter speed and the camera aperture to capture the best image possible given the limitations of the detector dynamic range.[1-4]

The series of images below were captured at different shutter speeds:

These images have been combined into a single dynamic range image:

Debevec's High Dynamic Range Imaging:[2,4] We have implemented Debevec's method for generating high dynamic range radiance maps[2,4]. This algorithm uses multiple photographs of the same scene taken at different exposure levels and then uses these to determine the camera response function for each color channel. Once the camera response is in hand the algorithm uses the response function to combine the images into a single image such that each pixel is assigned a value that is proportional to the true radiance of the corresponding in the real-world scene. Generating the camera response function requires solving a linear system of equations to generate a map between the detector response which is a function of both scene irradiance and shutter speed to the true scene irradiance. Construction of the radiance map is then the process of summing over corresponding pixels in each image the irradiance determined by this mapping function from the captured pixel value.[2,4]

Ward's MTB Image Alignment:[5] Because high dynamic imaging requires correspondence in each image at the level of each pixel we used a tripod to mount the camera and attempted to take each image without moving the camera. Unfortunately, some of our images did exhibit some minor shifts in camera position (particularly the outside images captured during extremely cold weather due to shivering). To deal with this camera movement we have implemented Ward's Median Threshold Bitmap image alignment algorithm to align all of our images so as to generate the best possible high-dynamic range image possible. Ward's algorithm is quite straightforward; the goal is to determine a number of pixels to shift each image along both the x and y axis to bring the image into alignment with another image. Consider a pair of images. First the images are transformed into grayscale so as to only deal with a single color channel. Then a series of recursive imaging resizing operations decrease the dimensions of each image by half at each step. These images are then converted to median threshold bitmaps by determining the median value of the grayscale pixel values and then remapping the image to a bitmap with 1 representing a pixel greater than or equal to the median and 0 representing a gray scale value less than the median. The images are then subjected to an xor operation which will evaluate to 0 if the pixels are identical and then summed. Then one image is translated by 1 pixel along the 8 possible direction and the bitmap is again xored and summed. The translation resulting in the smallest sum represents the best possible alignment. This alignment vector is then passed back to the recursive function that called it and is used to begin the alignment process for the next larger level.[5]

Tone-mapping: Once an absolute radiance map has been generated the resulting map must be again transformed to a format appropriate to an output device such as a monitor, or printer which also may have limited dynamic range. The goal is to generate an image as close as possible to what a human would perceive when viewing the original scene. In addition to experimenting with Matlab's native tone-mapping algorithm we also explored tone-mapping algorithms available on the web. Ultimately, we ended up implementing three of these algorithms.

Reinhard's Photographic Tone Reproduction:[6] We first implemented an algorithm by Reinhard et. al (found in the included Matlab file simplerein.m). This algorithm takes its inspiration from the dodge and burn techniques used by traditional film development and printing techniques.[6] This algorithm produces realistic renderings from absolute radiance maps by initially scaling the hdr image data using the log-average luminance, and then applying an automatic 'dodge and burn' technique to compress the dynamic range to match that of an output device. 'a' is used to represent the 'key value' which is used to map the middle-gray value of the displayed image to the key value of the original scene. Typical values set to a to 0.18 and then vary this value up to 0.36 and down to 0.045 to achieve appropriate scaling. An adaptation permits extremely high values, for example from highlights, to 'burn out' in a graceful way by adjusting their equation for log-average luminance to incorporate a linear component:

$$\bar{L}_w = \frac{1}{N} \exp\left(\sum_{x,y} \log\left(\delta + L_w(x,y)\right) \right)$$

$$L(x,y) = \frac{a}{\bar{L}_w} L_w(x,y)$$

$$L_d(x,y) = \frac{L(x,y)\left(1 + \frac{L(x,y)}{L^2_{\text{white}}}\right)}{1 + L(x,y)}$$

Following the global scaling, Reinhard's method the algorithm applies an automatic local 'dodge and burn'. This is much like selecting a key value of for each pixel and takes advantage of a local region around each pixel determined by the difference of Gaussian profiles with different radii.

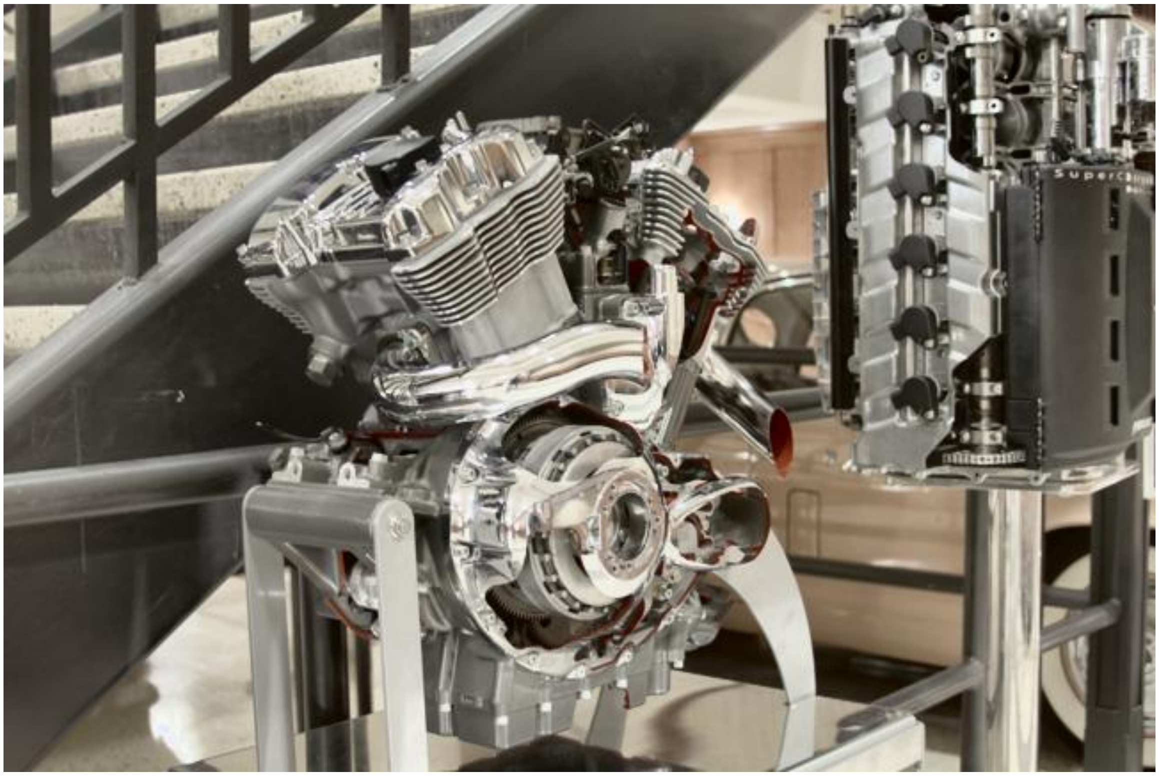
We did not implement this portion of the algorithm ourselves, but we did experiment with implementations available on the Internet and found the results to be very good. If we had been able to spend more time on the project this is the feature we would have loved to implement next. Our implementation of the global portion of Reinhard's photographic tone reproduction algorithm is found in our Matlab function named simplerein.m.

Reinhard's Dynamic Range Reduction:[7] We have implemented Reinhard's Dynamic Range Reduction Algorithm which can be found in the included Matlab file reinhard.m) This tone mapping algorithm draws from the physiology of the human visual system to visualize high dynamic range images on devices with limited dynamic ranges like monitors and printers in a way that maintains the brightness, visibility, contrast, and appearance of the original scene. Human photoreceptors automatically adjust to the level of illumination, to incorporate this model the algorithm interpolates for each pixel between the pixel intensity and the average scene intensity using only the luminance channel. The algorithm brings in a number of user specified parameters including a key value which can be used to estimate the user specified 'm' parameter which controls the contrast. The 'f' parameter is used to control the intensity in the resulting image. The algorithm also allows for chromatic adaptation through their 'c' parameter and finally 'a' is used to for light adaptation. Details of the algorithm are found within the paper.[7]

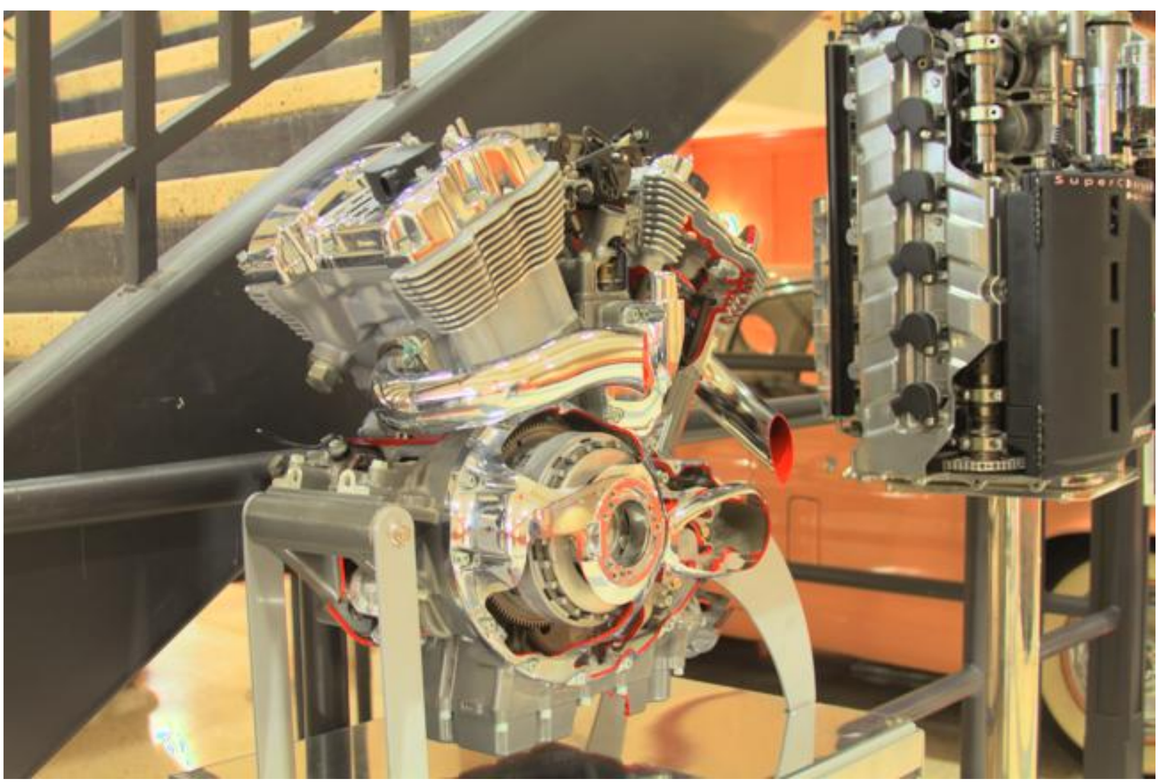
Drago's Adaptive Logarithmic Mapping: Finally, we implemented an adaptive logarithmic mapping algorithm by Drago et. al.[8] This algorithm may be found in the Matlab file drago.m, included with our submission. Drago's algorithm is quite straightforward in its foundational concept; it is simply the logarithmic compression of luminance values. The work varies the logarithmic bases to preserve details and contrast and then uses gamma correction to improve contrast in dark regions. The user specified parameters to this algorithm are the bias which is used to determine the logarithmic base, exponent, and contrast terms. Drago's algorithm uses the Yxy colorspace, so conversion functions were also written to convert to Yxy from rgb and back again.

Below are some example images prepared with the various algorithms we experimented with:

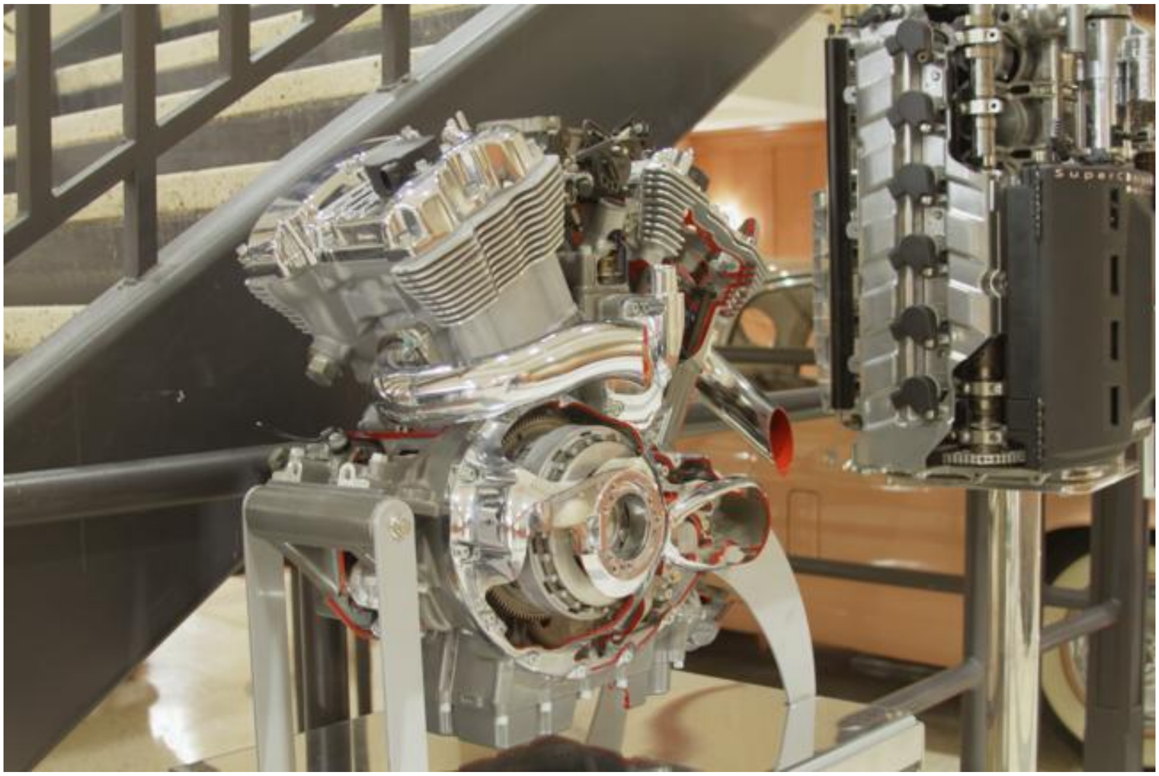
Matlab’s Native Tonemapping Algorithm:



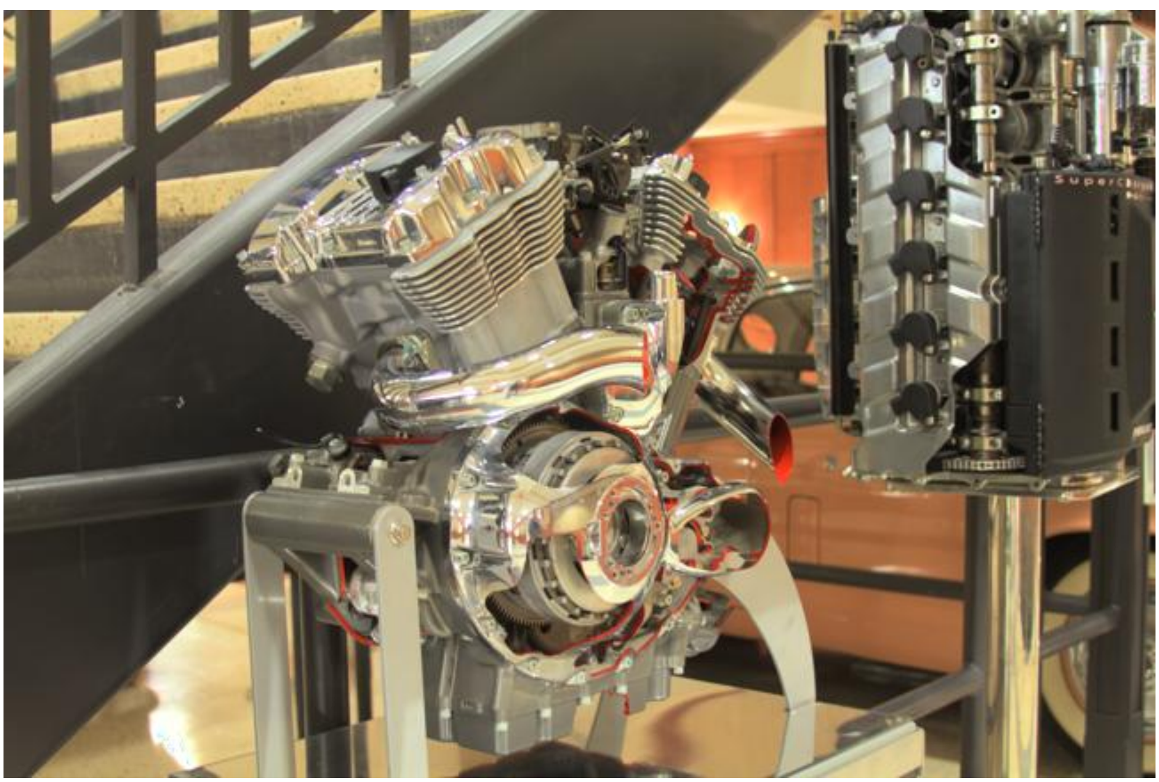
Simple Reinhard Tonemapping:



Reinhard’s Tonemapping Algorithm



Drago Tonemapping:



Usage: Our code requires a directory containing a series of images of the same scene taken with different shutter speeds. Also, in that directory our code requires a .txt file containing the length of the exposure as a double for each image. Please record the exposure values in an order corresponding to the images listed in alphabetical order.

In the main.m script, you will need to specify the name of the directory as well as the image file-type extension (e.g. 'jpg') please see the commented examples for the prerequisite format. The code first reads the images and the file with exposure data and then runs Ward's image alignment algorithm. Please set the depth of the recursive calls where indicated in the code. Note that the images will shift by a maximum of 2^(depth-1) pixels so you can fine tune the maximum shift based on the magnitude of movement and the image size. The middle section of the code in main.m then runs Debevec algorithm to generate a high dynamic range image – no user intervention is required here. At the end of the main function you have the option to select the tone-mapping algorithm to use for displaying the image. Examples using the native Matlab tone-mapping algorithm as well as calling our tone-mapping functions (simplerein, reinhard, or drago) appear as comments in the code; simply remove the comment to engage one of these algorithms for displaying the hdr image.

Contributions: A large portion of the development occurred as paired programming where both of us were working at the same computer looking over the code. We met nine evenings for approximately two hours each evening of the course of the project to review code and take pictures. Eugene was responsible for the initial drafts of most portions of the code including: image loading, solving for the radiance map, radiance mapping to create the hdr image, and the tone-mapping algorithms. Michael was responsible for Ward's MTB image alignment algorithm and substantial revisions to the code to enable working with image data structures instead of working directly with the image files in the image loading and image sampling functions. We meet twice and took photographs of scenes near the computer science building and in mechanical engineering.

GitHub commit logs may be found at:
https://github.com/mdoescher/Computer_Vision/commits/master.

References

1. High Dynamic Range Imaging. (n.d.). In *Wikipedia*. Retrieved from en.wikipedia.org/wiki/High-dynamic-range-imaging.
2. Debevec, P.E., Malik, J. (1997). High Dynamic Range Radiance Maps from Photographs. SIGGRAPH 1997. Retrieved from vsingh-www.cs.wisc.edu/cs766-12/lec/debevec-siggraph97.pdf.
3. Forsyth, D.A., Ponce, J. (2012). *Computer Vision: A Modern Approach 2nd ed.* Prentice Hall: Boston.
4. Szeliski, R.(2010). *Computer Vision: Algorithms and Applications*. Springer. Retrieved from szeliski.org/Book/.

5. Ward, G. (2003). Fast Robust Image Registration for Compositing High Dynamic Range Photographs from Hand-Held Exposures. Jgt 2003. Retrieved from pages.cs.wisc.edu/~lizhang/courses/cs766-2008f/projects/hdr/jgtpap2.pdf.
6. Reinhard, E. Stark, M., Shirley, P., Ferwerda, J. (2002). Photographic Tone Reproduction for Digital Images. SIGGRAPH 2002. Retrieved from pages.cs.wisc.edu/~lizhang/courses/cs766-2012f/projects/hdr/Reinhard2002PTR.pdf.
7. Reinhard, E., Devlin, K. (2005). Dynamic Range Reduction inspired by Photoreceptor Physiology. IEEE TVCG 2005. Retrieved from pages.cs.wisc.edu/~lizhang/courses/cs766-2012f/projects/hdr/Reinhard2005DRR.pdf.
8. Drago, F., Myszkowski, K., Annen, T., Chiba, N. (2003). Adaptive Logarithmic Mapping for Displaying High Contrast Scenes. Eurographics 2003. Retrieved from pages.cs.wisc.edu/~lizhang/courses/cs766-2012f/projects/hdr/Drago2003ALM.pdf.