Madeline Burton
Sarah Bouwman
Andrew Hoelscher

HDR Photography and Tone Mapping
CS 4670 – Computer Vision

With this project, we tried to address on of the major problems in modern photography. When you view a scene with the naked eye, your eye moves around, and so you absorb multiple different levels of light into your retina, often adjusting so that you will receive more light from the darker areas of a scene, and less from the brighter areas. This dynamic scene viewing allows us to see details in a scene which includes both light and dark areas. However, a camera is only able to take in a certain amounts of light as it takes a picture. You can vary this with exposure time, but images with a high exposure time and a bright light source will often appear very bright and washed out in some areas, whereas images with a low exposure time will often be too dark overall, with points of the image that appear normal. To combat this, we felt that the best method would be to take multiple images of different exposures, as has been done in several papers, and see what kind of results we would get. This problem is becoming a popular one in modern photography (mostly due to the fantastic images that are generated), and so we decided it would be an interesting one to tackle for our final vision project.

There have been several papers published on the subject of HDR. For initial reference, we looked at two books, *Computer Vision: Algorithms and Applications* by Richard Szeliski, and *High Dynamic Range Image Reconstruction* by Alsa M. Sa, Paulo Cezar Carvalho and Luiz Velho. These books were a good introduction to these topics, and they referred us to the paper "Recovering High Dynamic Range Radiance Maps from Photographs", by Paul Debevec and Jitendra Malik. This paper was what we based the majority of our HDR algorithm off of. It included written MATLAB code for part of the algorithm, which we used as a pseudocode basis for our own section. The tonemapping algorithm we applied in order to reduce our HDR image to a displayable image was described in "Photographic Tone Reproduction for Digital Imaging" by Erik Reinhard, Michael Stark, Peter Shirley and James Ferwerda.

The algorithm we implemented has two main parts. The first part was taking the three input images, each with a different exposure time, and creating a radiance map of the scene. This will allow us to find the adjusted irradiance at every pixel determined by the weighted images. To do this, we have to find g(Z_{xy}), which is related to the response curve f by maps pixel intensities to radiances. By finding g, we can use the equation $g(Z_{ij})=\ln E_i + Dt_j$ to map these intensities to find the new pixels. To find g, we need to minimize O, in the following equation, where the first term in the equation is based off of the earlier equation, and allows us to solve the equations as a least squares problem. The second term is a smoothness term, which ensures that g is a smooth function.

$$O=\sum_{i=1}^{N}\sum_{j=1}^{P}[g(Z_{ij})-\ln E_i-Dt_j]+\lambda\sum_{z=Z_{\min}+1}^{Z_{\max}-1}g''(z)^2$$

This equation can be minimized by solving it as an overdetermined system of linear equations. We represented this equation as a matrix of values, and minimized it using the CvSolve function. From here, we can determine the radiance at each pixel from the value ln E, which is calculated from this equation.

$$\ln E_i=\frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij})-\ln Dt_j)}{\sum_{j=1}^{P} w(Z_{ij})}$$

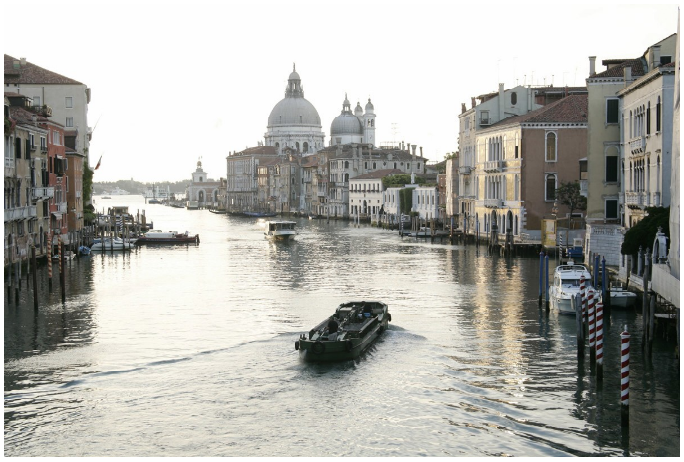
From this new mapping we can derive an HDR image.

The second part of this algorithm, tone mapping, in necessary because some of the radiance values are out of range of the displayable format, so we need to compress non-displayable images into a displayable format without losing local contrast. However, the tone mapping algorithm dealt with the luminance values instead of the radiances, so the first step we needed to make was to translate the radiance from RGB space to Yxy space, using a predetermined set of values that we looked up online. Once we had made this conversion, we use the luminosities Y to remap the luminance to a format that can be displayed. In our tone mapping algorithm we implemented a method called dodging and burning, a local operation, done per pixel, which allows for much better images. The technique is to find the largest blob centered on that pixel, and scale that pixel by the average luminance in that blob. This is done to better preserve contrast. We did this by convolving the image with a series of Gaussians of differently scaled widths, to create a list of convolved matrices. From here, we determine the appropriate scale for that pixel, and used the pixel value in the appropriately scaled image as our final value. Then we converted back to RGB space to display the image.

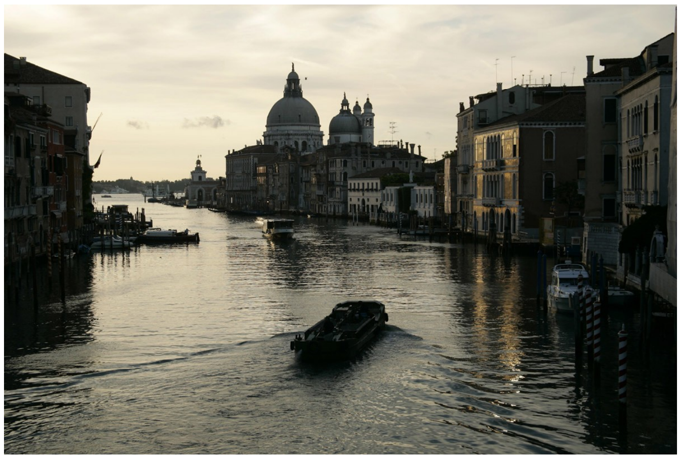
The results that we achieved with this algorithm are displayed below. The three input images for each test were found on www.hdrspot.com/examples.html.

Examples:

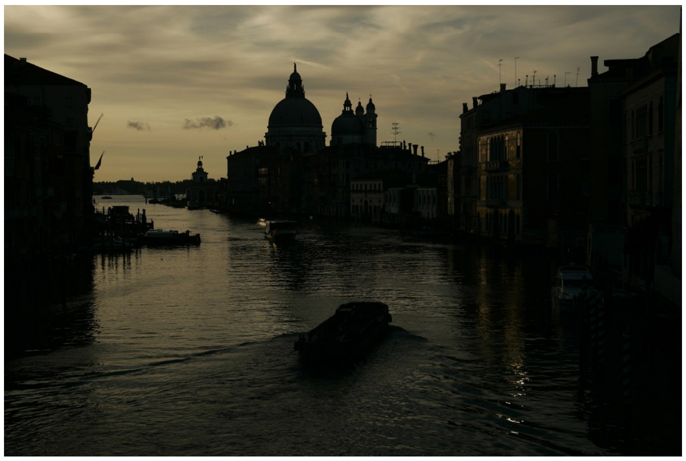
Original Image, exposure time 1/45s:



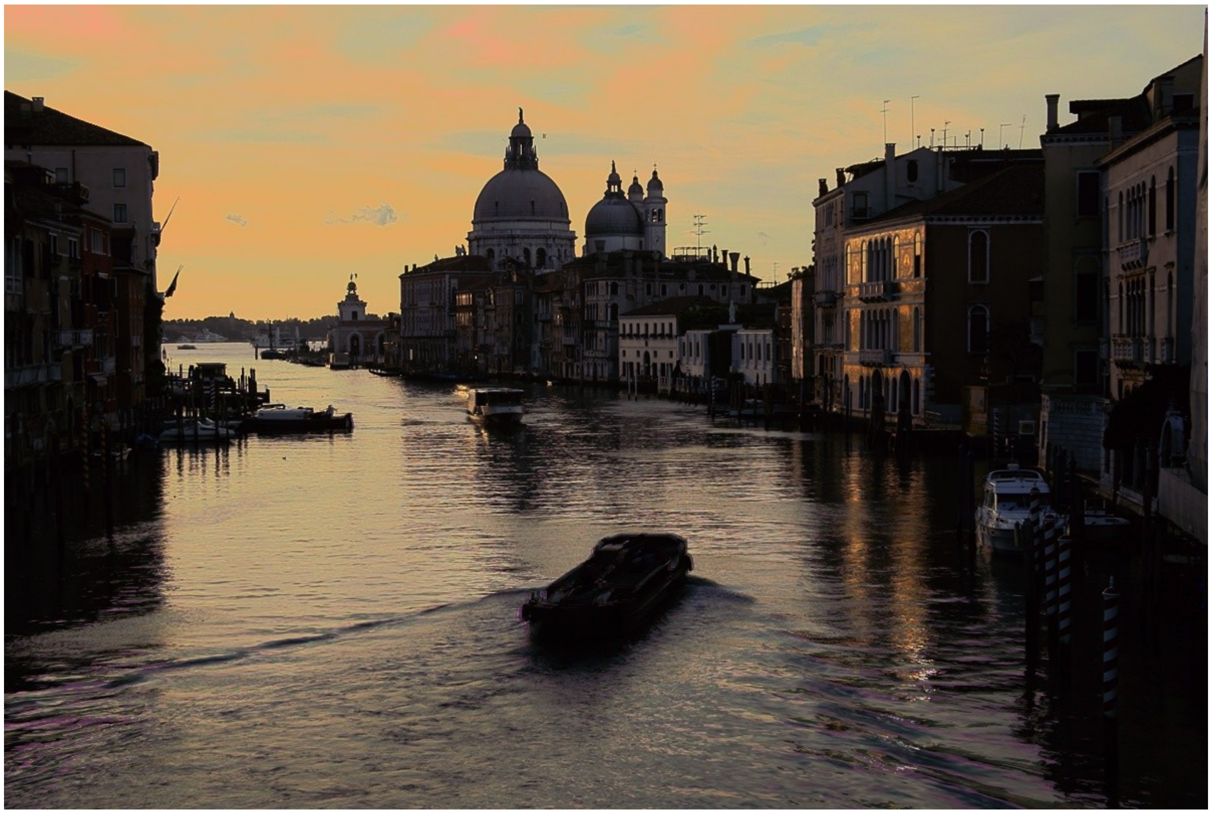
Original Image, exposure time 1/180s



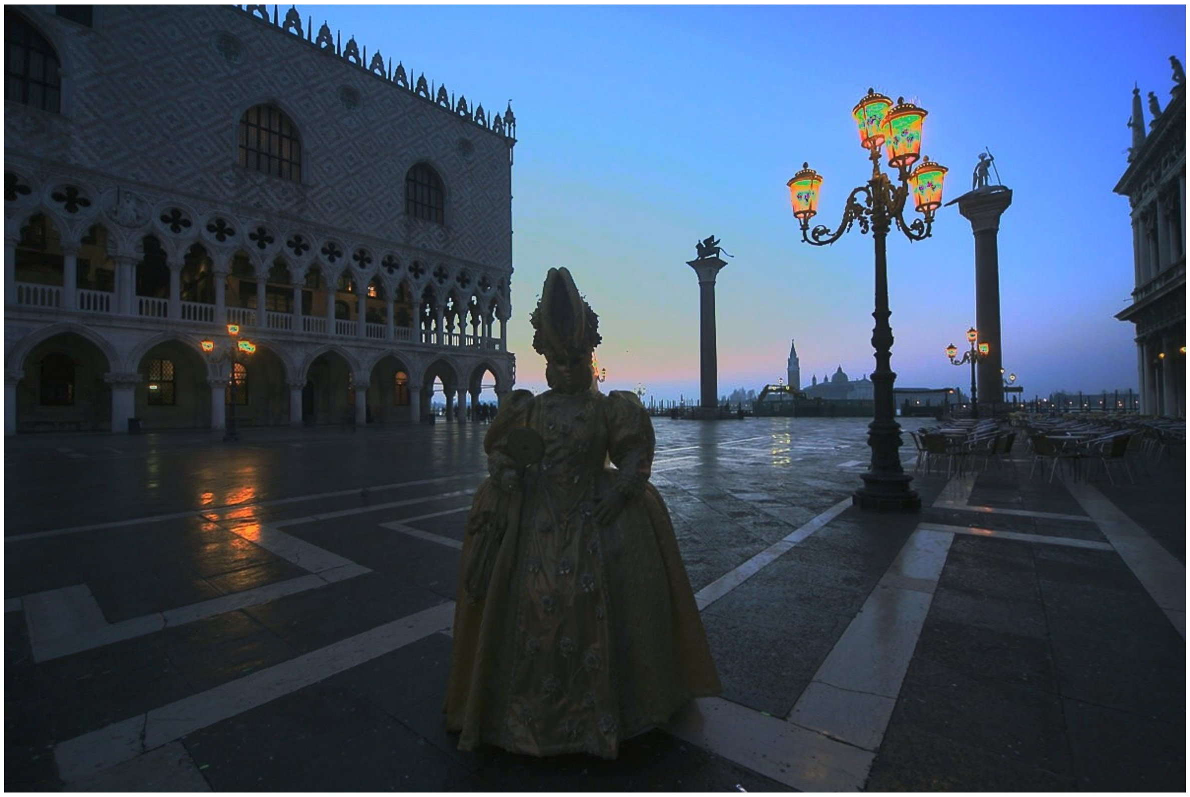
Original Image, exposure time 1/750s

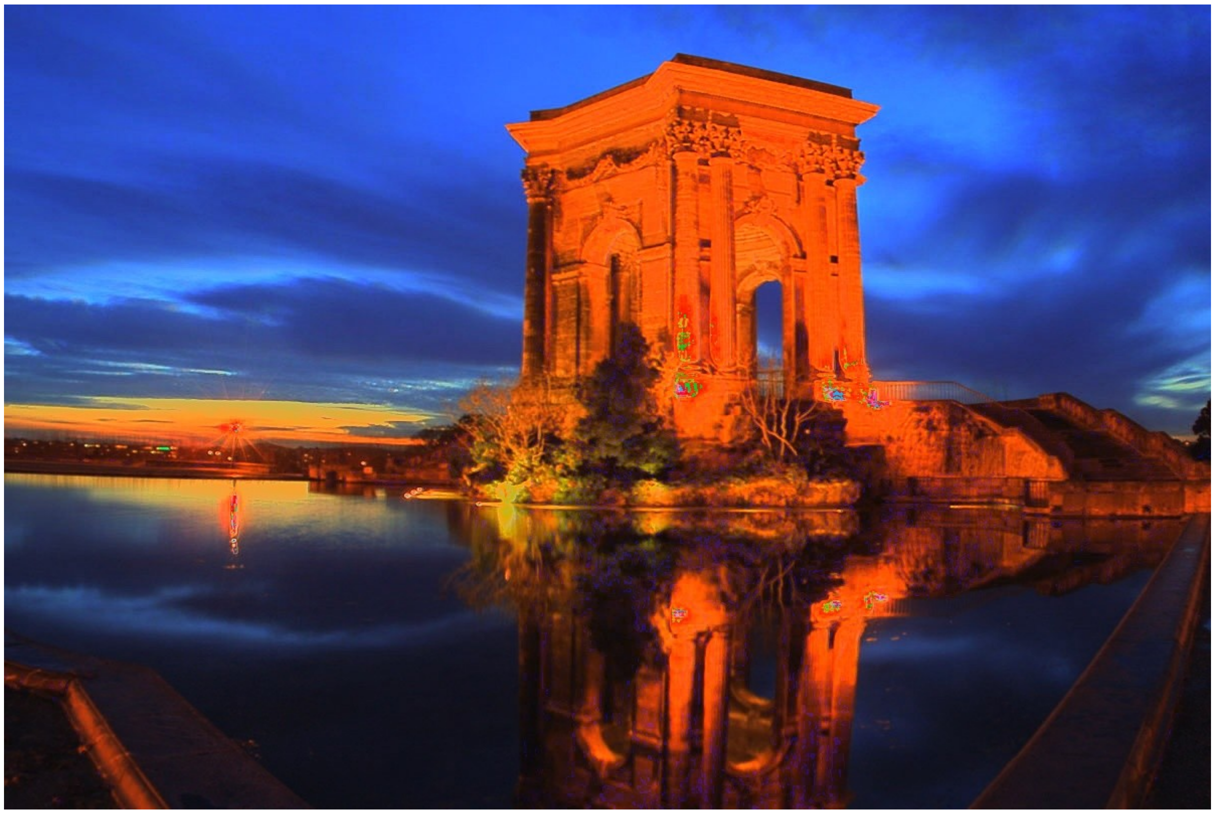


Output Image:

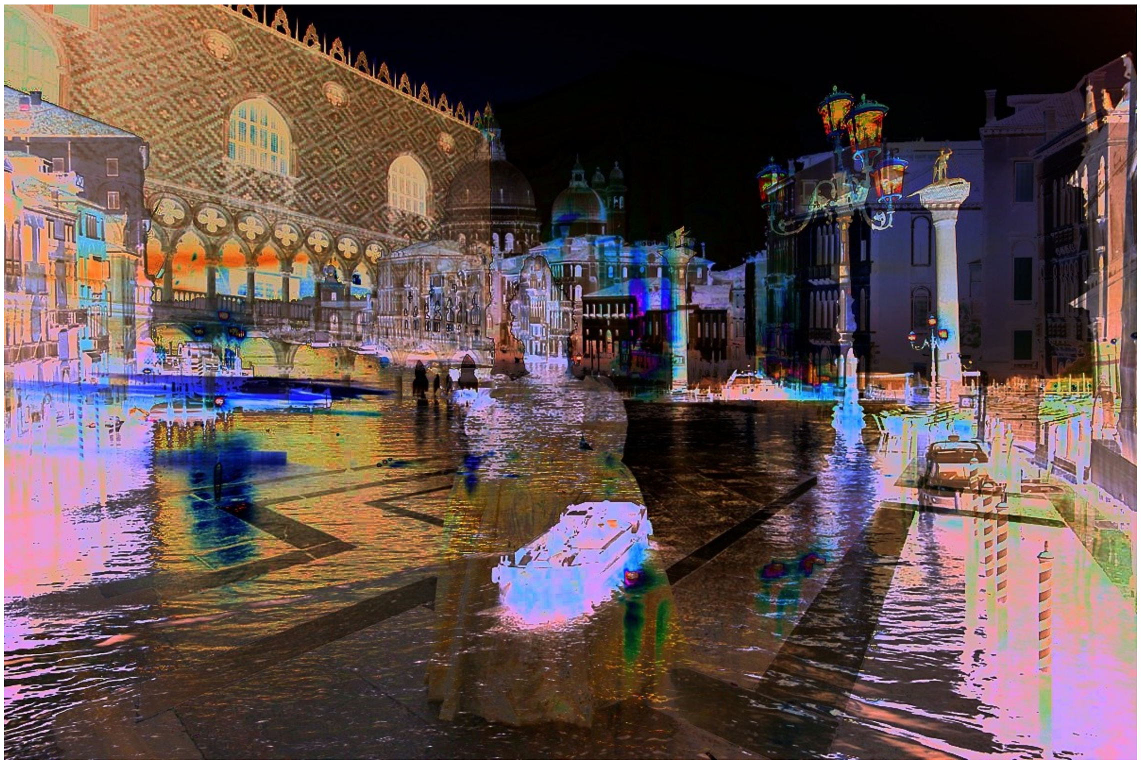


Other image results





Also, this is pretty cool. It's a combination of one exposure time per test image.



The breakdown of work was roughly as follows: Sarah began the project before Drew and Madeline, creating the basic structure for the HDR and tonemapping code. After finishing another project, Drew and Madeline joined Sarah to fine tune the code and debug. While Drew and Sarah spent their time with HDR and tonemapping, Madeline began to look in to implement our code on both the Nokia 900 and Android platforms. After extensive research, she decided that neither phone would realistically be able to support the project due to heap memory issues and joined Drew and Sarah for tonemapping.

In general, our HDR compares favourably to what we hoped for. However, it doesn't quite match the standards of commercial versions, as are popular in modern photography. The discrepancies include some burnout, artefacts around bright regions and some wash out around darker regions. These problems result not from the misinterpretation of the algorithm of bugs in the code, but from some parameter mistuning within our code , as well as the implementation in commercial software of extensions to the Debevec and Reinhart algorithms that are not included in the papers. Furthermore we recognise that the requirement that the images be aligned is needlessly restrictive, however we choose to focus on the HDR and tone mapping algorithm in preference to the image alignment, which we have implemented in previous projects. Overall though, our tone mapping was a great success, the implementation of dodging and burning really made our images clearer and a high lambda value for our smoothing removed a lot of the blocky artefacts from our images, as well as significantly decreasing the burnout we received on most images.

If we were to continue working on this implementation, we might attempt putting the application on the phone. This would be a sever optimization process, and might change the basic equations we were using. For instance, the utilization of fast Fourier transforms in the tone mapping code. Also, we could implement image alignment, such that the photographs would not need to be aligned before running the code. One final possibility is to extend the code such that it can take in more than three images for alignment.