University of California Santa Cruz

CMPE264 Proyect 1: High Dynamic Range (HDR) System

Aaron Hunter
Carlos Espinosa
November 14, 2018

Introduction

This report documents a manual approach to creating a high dynamic range, or HDR, photograph. HDR is a technique used to display high contrast images by combining a series of two or more photos at different exposure values taken of the same scene in order to boost the signal to noise ratio, or SNR, of the entire image. Since most monitors can't project HDR images, the images are compressed using a tone mapping function which reduces the global contrast in a way that allows all areas of the image to retain significant detail, in other words it reduces the overall contrast ratio of the image, while preserving the relative values.

First we sought out a scene that would demonstrate the technique effectively. Our initial thoughts were to get images during the daytime of the interior of a church with colorful stained glass windows receiving sunlight. The challenge then would be to allow the very bright window to be displayed alongside interior details of the building and have both be clearly visible. The state of religion being what it is, however, all the churches we visited on an early Saturday morning were locked. There is one chapel, however, that is maintained as an historic site by the Holy Cross Church. It is known as the Mission Chapel and dates back to the mid 1700's when Spain occupied the central coast of California. This chapel turned out to be an ideal subject, at least from the technical standpoint. Although it doesn't have the colorful stained glass we were seeking, we realized after looking at images on our camera that there was a faint cross made by the individual panes of glass in each window. The effect is so faint that we didn't notice it at first. The interior of the chapel is dimly lit with small incandescent lights mounted in a heavy iron candelabra. On the walls are old religious paintings that were difficult to see with the naked eye, much less photograph well. Thus we found our challenge. Create an image of the interior that contains both a directly sunlit window and a dark religious painting that allows the details of both to be seen.

The following sections will detail two approaches to generate an HDR image. In the first section we evaluate the compression scheme that our camera uses when generating a JPEG file. Although the camera is capable of producing "raw" images, that is, images that are linear over the entire range of the sensor, we will attempt to reverse engineer the non-linear process used by the camera to produce a JPEG from the linear sensor. The second section shows each of the images collected for our project. We will select three images from the collection and generate a linear version of each. In the third section we generate two different composite images, one which simply selects non-saturated pixels based upon their exposure, in other words, it combines the images based upon their SNR within the scene. We then create a second composite image using an average of the non-saturated pixels. Finally, we use the Matlab tone mapping function to generate a compressed image for display on a standard monitor.

1 Camera Radiometric Calibration

In preparation for this report we evaluated two cameras: a Sony Rx100 and Fujifilm X-E1. The compression scheme for the X-E1 appeared to be a smoother function so we chose that camera for the project. Initially we took images in daylight at a gain value of ISO 200. However, the pictures in the chapel required a higher gain to span a broad range of exposure values, so we repeated the process at ISO 800. The target image is a piece of standard printer paper, backed by a piece of white cardboard. We added the backer to minimize the transparency of the paper. We took photos of the target in shade using a tripod at a range of exposure times from 1/60 sec to 1/4000 sec at f8 with the white balance set to daylight. We cropped a 100 × 100 pixel area of the picture and averaged over the cropped image to measure its average color. The cropped images are presented in fig(1).

If a camera does not use a compression scheme for its JPEG image processing, a plot of the average color against the exposure time will be linear. For our camera, however, there is a strong non-linearity as seen in fig(2). What is notable about this camera is that the three colors behave very similarly and that the values are very close to one another depending on which portion of the original image we chose. We tried two different methods for linearization, a polynomial fit and a power fit. Although a cubic polynomial fits the data perfectly, between the data points the fit is suspect, so we chose a power law fit instead. Its poor fit is at least repeatable between the color channels. The fitting parameters found for the exponential term for each channel are shown in table (1). To determine these values we used the Matlab `lsqcurvefit` function specifying a fit to the data of the form $B' = g_0 * B^{1/g}$, where B' is the recorded brightness of a given pixel, B is the actual value measured by the sensor, g_0 and g are the fitting parameters. This is a standard least squares fit that minimizes the residual between the data and the prediction. Because we only care about the non-linear factor, we can safely ignore the proportionality constant when calculating the linear image values.

Color	g
Red	2.4678
Green	2.5483
Blue	2.4473

Table 1: Exponential fit parameter g for each channel of the Fujifilm X-E1 camera

The fit of the data for each channel is shown in fig(3. It is clear that the linearization is not perfect, particularly for high brightness values. If we reverse the linearization process we obtain the approximately linear data shown in fig(4. Although still retaining a convex shape the curvature relative to the recorded brightness values, B' is reduced.

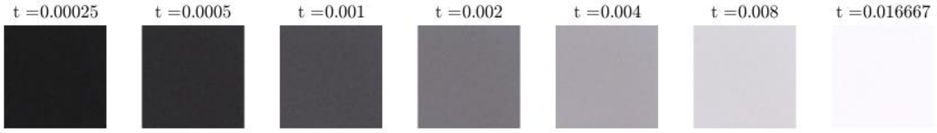

Figure 1: Fujifilm X-E1 white calibration images and exposure time. All images taken at ISO 800. These data span 67× exposure range

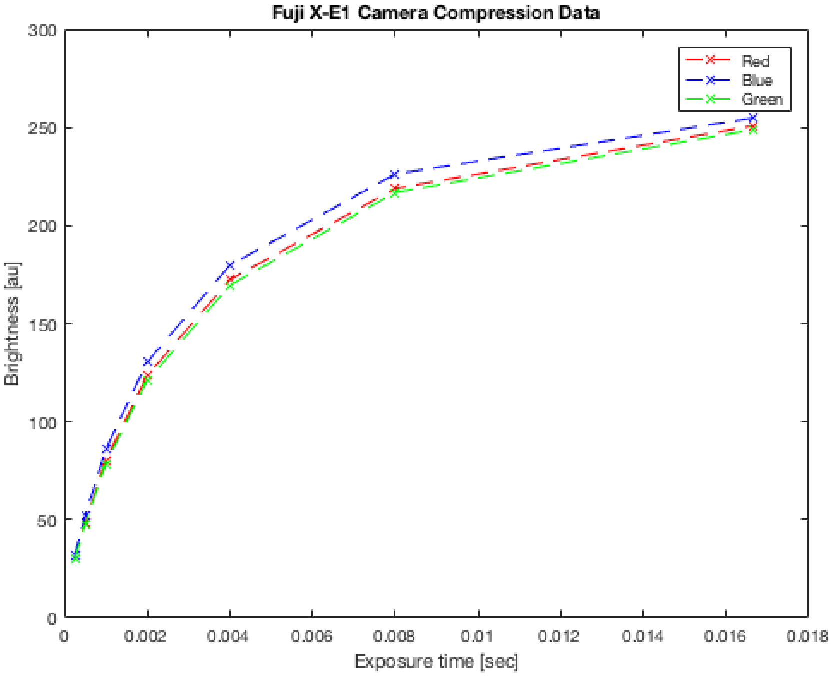

Figure 2: Fujifilm X-E1 brightness values versus exposure time at ISO 800.

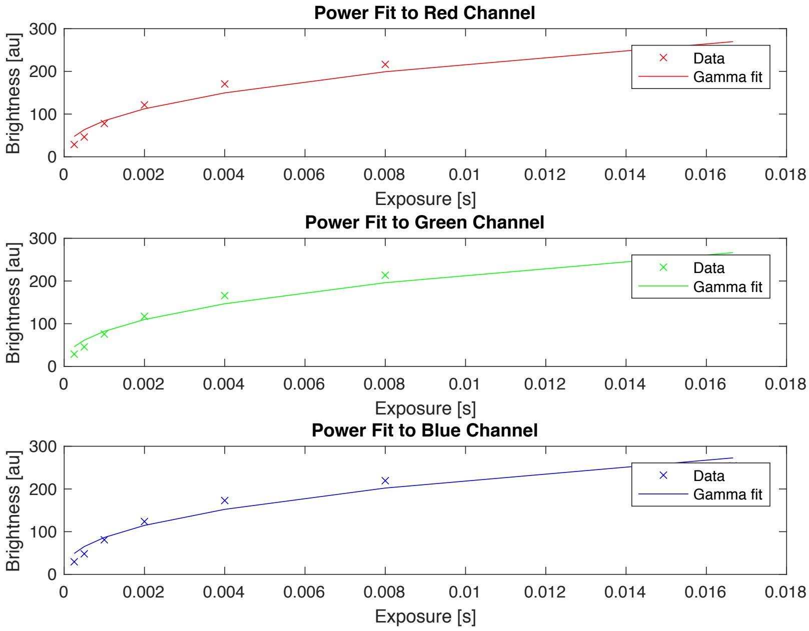

Figure 3: Power fit to the exposure data for each color channel

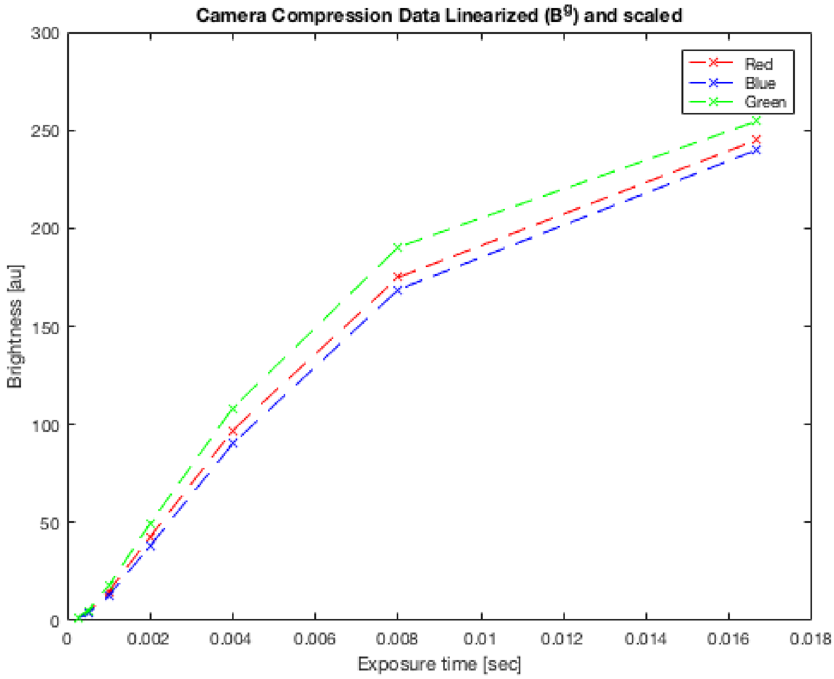

Figure 4: Linearization results on exposure data for each channel. Note that this fit to the camera data is not particularly good and the resulting linearization isn't particularly linear. The important thing to note is that the response is more linear than the compressed data.

2 Picture Stack Acquisition

The photographs that we took inside the chapel are shown in fig(5). In them the completely black areas demonstrate the pixels that are saturated and the histogram shows the brightness values once the saturated pixels are removed. The darkest image is exposed such that the cross in the window is visible and not saturated. It is taken at ISO 800 at 1/1000 of a second. Each subsequent photograph is taken at twice the time of the previous one, up to the brightest photo in the collection taken at 1/60 second. To determine which images to use for our composite image, we evaluated each for their content. The first picture selected is the one with no saturated pixels and clearly shows the faint cross made by the window panes. The brightest image is also selected because it is exposed properly for the interior. For the middle image we chose the one taken at 1/125 second because of the detail seen in the window sill itself.

Fig(6) shows the three images selected with their respective histograms after linearization but before dividing by exposure ratio.

Fig(7) shows the histograms for each color channel of the linearized photographs. Fig(8) shows the histograms for the second and third images chosen once their overall brightness is reduced by the ratio of their exposure time versus the base image exposure time.

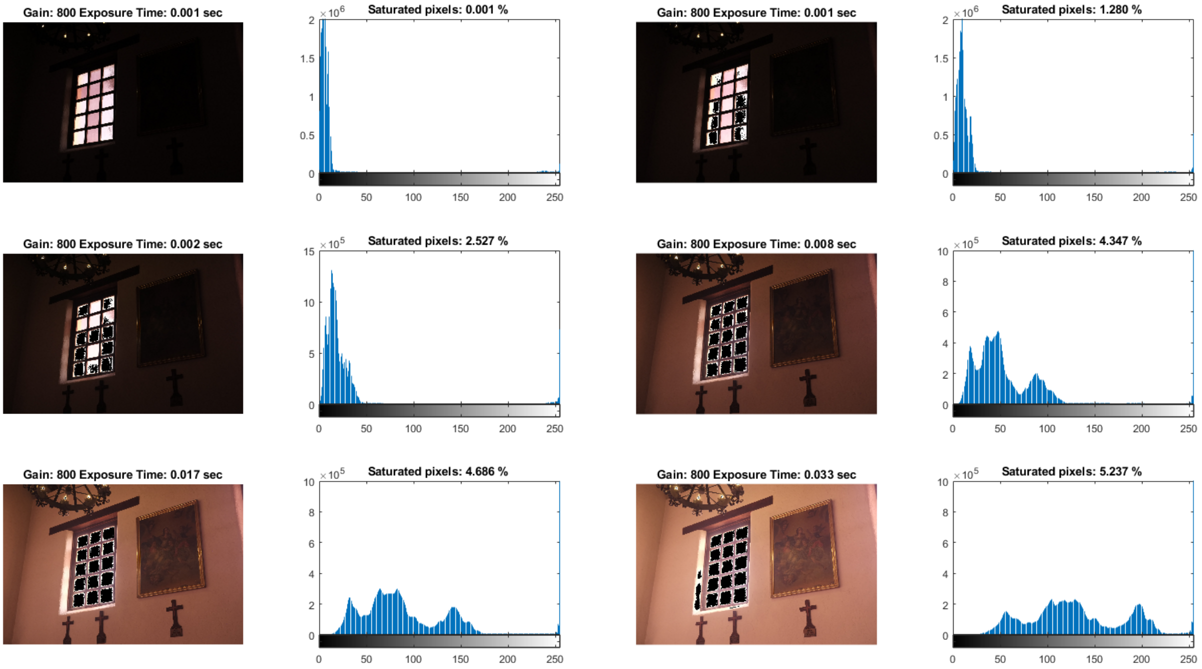

Figure 5: Picture Stack with ISO gain fixed at $G = 800$. The first picture selected is the top left image. The second is the middle right, and the third is bottom right.

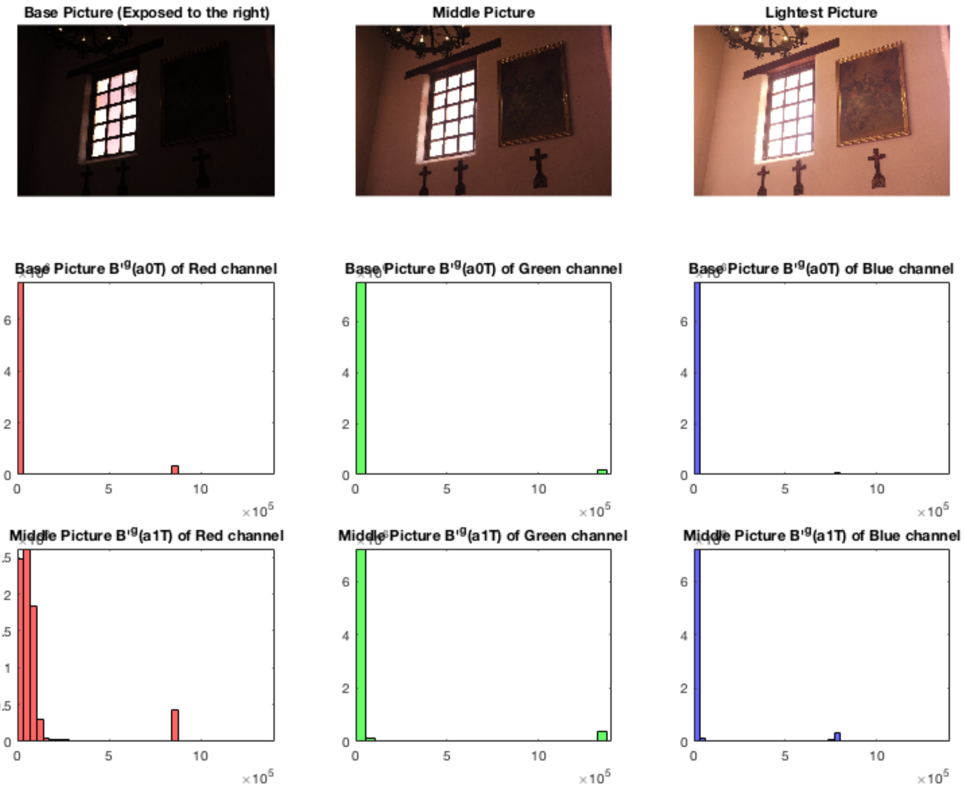

Figure 6: Image stack used for the HDR image and histograms

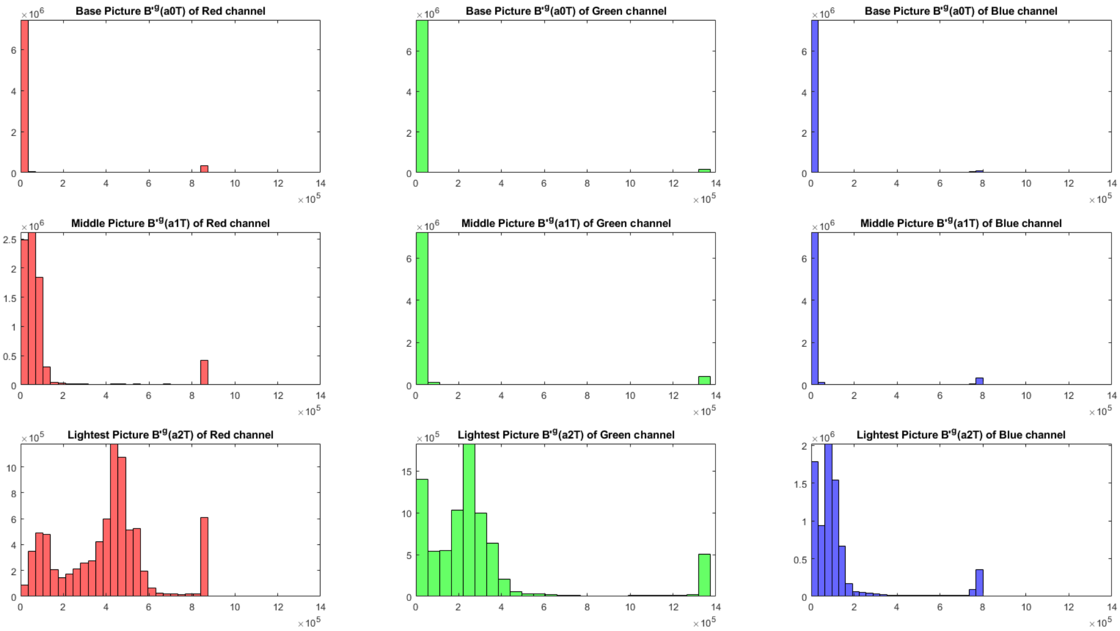

Figure 7: Histograms for the linearized images

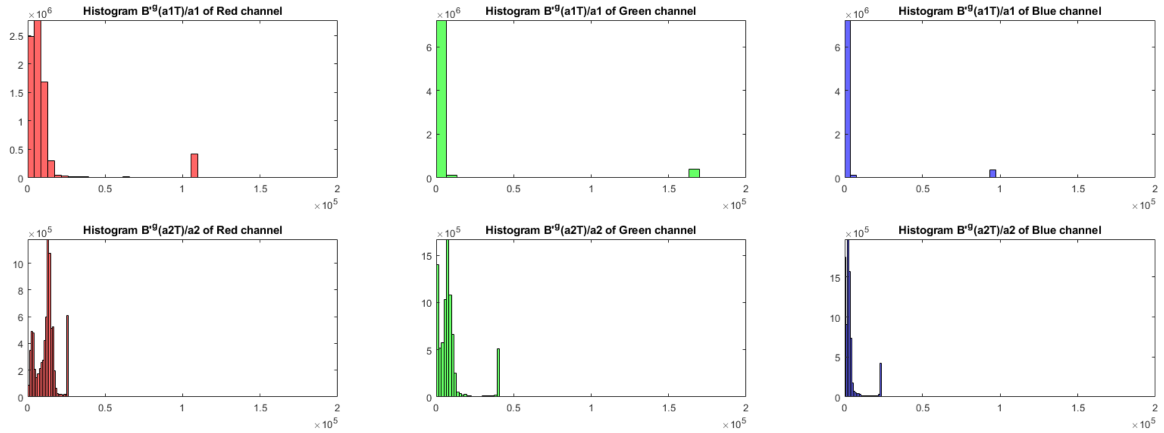

Figure 8: Histograms for the linearized images, adjusted for exposure times. Image 2 brightness values have been divided by $a_1 = 8$ and image 3 values have been divided by $a_2 = 33$

3 Composite Image Creation

In this section we explore two different methods for combing the linearized images selected in the previous section. The first method selects the pixels with the highest SNR between the three images and uses them for the composite image. The second one doesn't discard the non-saturated pixels. Instead it takes the average of the pixels from the three images with non-saturated pixels.

3.1 Composite Algorithm 1

In the first composite image, we selected the pixels from image 1 where the brightness $B \geq E_{max}/a_1$ where $a_1 = 1000/125 = 8$ and E_{max} is the brightest value in image 1. This is because all the pixels from image 2 and image 3 are saturated above this exposure value. We performed this with the use of a mask applied to image 1. We then created a mask from image 1 where $B < E_{max}/a_2$ (to have the correct brightness in the final image) and applied it to image 3. these pixels were then divided by a_2 and represent the pixels with the highest SNR in this portion of the image. Finally, we took the inverse of the two masks and generated yet another mask by multiplying them together. This mask was then applied to image 2 and the resulting image was divided by a_1, again, to have the correct brightness. The resulting image represents the pixels with the highest SNR between E_{max}/a_2 and E_{max}/a_1. The histogram of the final image is found in fig(9). The composite image is just the sum of the individual masked images. The algorithm for the code generation is shown below for clarity.

```
 ...
 mask = im1>= E_max/a(2);
mask1 = mask;
...
%mask the lowest exposure pixels
mask = im1 < E_max/a(3);
mask3 = mask;
...
%mask two is the inverse of mask1 and mask3
mask2 = not(mask1) .* not(mask3);
...
```

3.2 Composite Algorithm 2

For the second composite image we take an average of the non-saturated pixels in a given region. Clearly, the values from image 1 above E_{max}/a_1 can only come from image 1. For the pixels that fall between E_{max}/a_2 and E_{max}/a_1 we take the average of the pixels from image 1 and image 2. For the remaining pixels the average of all three images is taken and applied. The code for this is demonstrated in the following snippet. The histogram of the resulting image is found in fig(9)

```
 ...
mask = im1>= E_max/a(2);
mask1 = mask;

mask = im1 < E_max/a(3);
mask3 = mask;

mask2 = not(mask1) .* not(mask3);

Image1 = (im1.*mask1);
Image2 = (im1.*mask2 + im2.*mask2/a(2))/2;
Image3 = (im1.*mask3 + (im2.*mask3)/a(2) +(im3.*mask3)/a(3))/3;
...
```

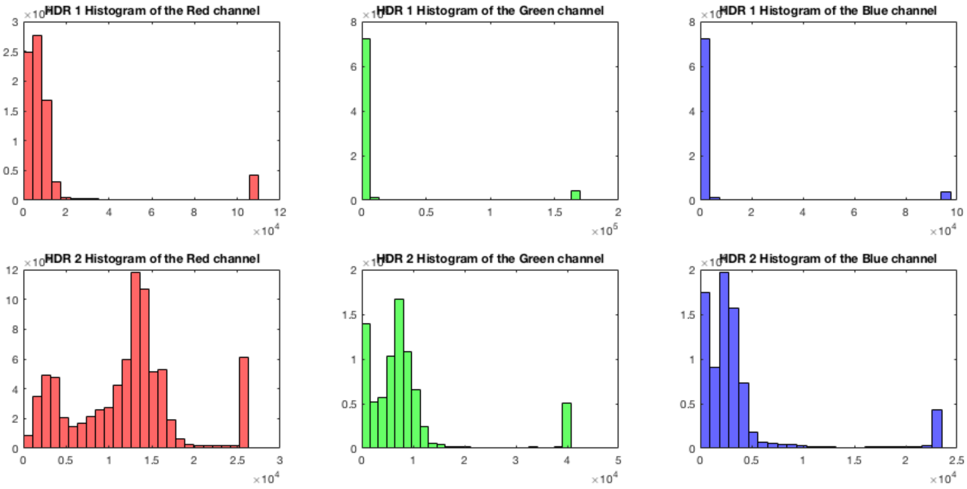

Figure 9: Histograms for the HDR composite images

4 Composite Image Reproduction

Now that we have the HDR images we need a way to represent them. If we naively convert them to 8 bit images we lose all the extra information that we added with the longer exposures. For this report we used the tone map function within Matlab to generate standard resolution images. Tone map functions by compressing the contrast of the scene while preserving the relative brightness values. Our initial attempts using the default parameters of the function resulted in a rather gray and flat image. We adjust the saturation parameters to increase the intensity of the color and adjusted the lightness of the image to increase slightly the contrast. The composite images are shown below. Fig(10 is generated is the result using algorithm 1 above. Fig(11) is the image generated using the same parameters for tone map as used for composite image 1.

The immediate thing to note about both images is that the cross made from the window is visible despite the dramatic contrast of the real world scene. Also the details in the painting are easily visible. This is actually quite remarkable because these details were difficult to see in person. The differences between the images is subtle. The second composite image has larger overall contrast despite using the same parameters to generate the tone map. More importantly though, the detail around the window area is improved. The color balance of both images is slightly off, perhaps due to the influence of the color of the incandescent lights located in the candelabra fixture.

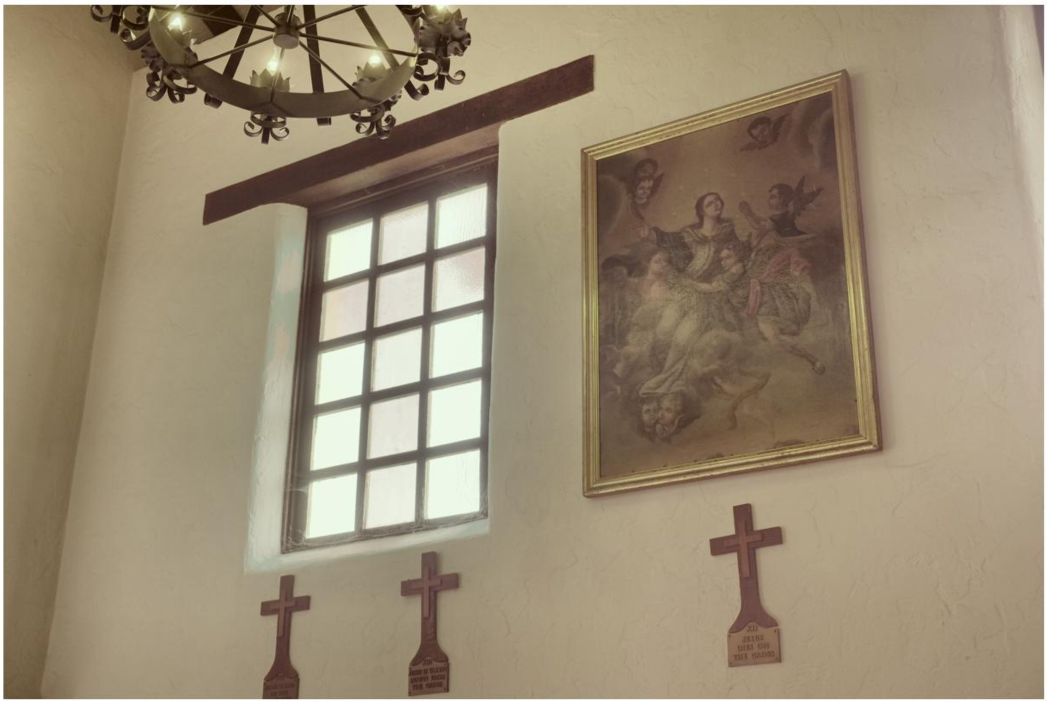

Figure 10: Composite image 1 after the tone map. Note the faint cross formed in the window and the detail in the painting on the wall.

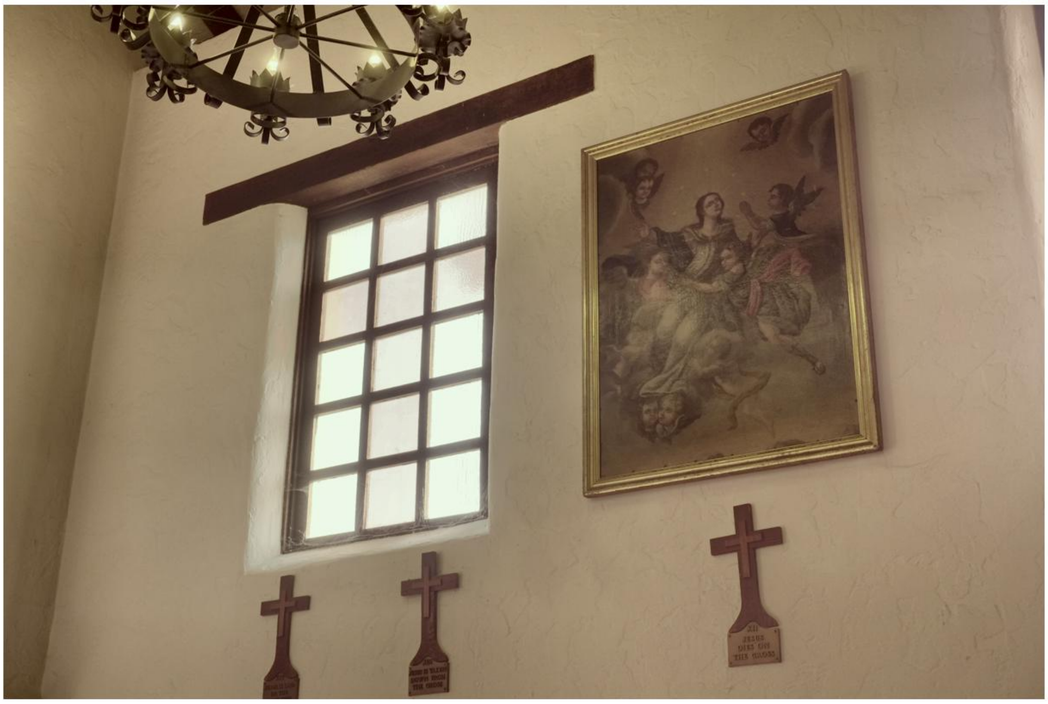

Figure 11: Composite image 2 after applying the tone map. Note the relatively higher contrast and greater detail, particularly around the window sill.

5 Conclusions

In conclusion we have demonstrated the principle of generating HDR images by creating composite images of a scene at multiple exposures. The images we produced are ones that look realistic but actually do not exist in the real world. In person the interior of the chapel was quite dark and the light from the window dominated our vision. Because of this the details of the painting were impossible to discern until we actually generate the HDR image itself. The slightly more sophisticated application demonstrated by the algorithm which uses all available information results in a visually more satisfying image, perhaps due to the greater amount of information encoded in the HDR image. More understanding of the tone map function would provide a better understanding of its output. There are probably even more sophisticated ways to add all the picture information, perhaps weighting the pixels by their SNR.