Data Collection

Figure 1 below shows three different exposure images of the same scene as well as one background image without the mirrored object. The camera used to capture these images was an iPhone 11 camera. The scene is set indoors and is of a table with various items. The cropped images of the three exposures are shown in figure 2 with solely the mirrored object. There is some alignment error of the cropped images which can be seen later in the final HDR images but, for the purpose of this project, it is negligible. Each of the images in figure 2 were also resized to be 500 x 500 pixels for easier computation.

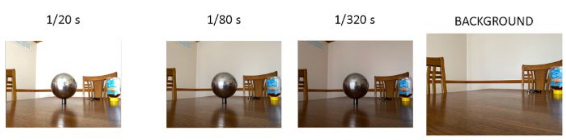

FIGURE 1: LDR and BACKGROUND images

FIGURE 2: LDR cropped images

Naive merging

In Naïve Merging, the goal is to take the LDR (Low Dynamic Resolution) images and to do a simple average of each of the respective pixels to create an HDR (High Dynamic Resolution) image. Each image was first scaled by dividing by its exposure time to put the images in the same intensity domain. Each pixel intensity is then averaged across the three images to get the final naïve HDR image. Figure 3 shows, in a broad sense, how HDR imaging captures a greater dynamic range than what a normal photograph would have with just a simple LDR image at one exposure. Figure 4 shows the final naïve HDR image created from Figure 2 along with the log irradiance images, or how much light is coming into the camera for that exposure.

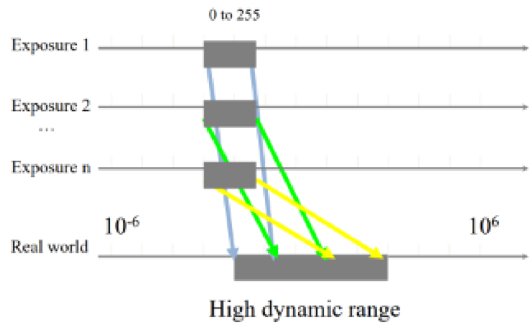

FIGURE 3: LDR to HDR

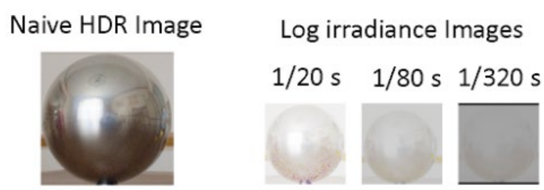

FIGURE 4: Naive merging

Exposure Correction

Another method to make a HDR image is to exclude certain parts of the image that are under or over exposed from the average calculation process. A weighted function was used to do exactly this:

w = lambda z: 128 - np.abs(z - 128)

The variable "z" is the pixel intensity.

The weighting function was applied to each pixel intensity for every image. That output was then multiplied by the pixel value at the given exposure and scaled by the exposure time. Finally, a sum of all the weights was used to get a weighted average, resulting in the final HDR image. Figure 5 shows the final HDR images along with the log irradiance images.

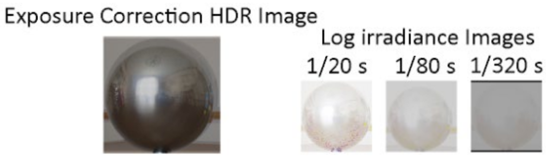

FIGURE 5: Exposure Correction

Response Function estimation

Response function estimation maps out an irradiance value for each pixel value in each channel. Then an HDR image is created using these values.

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

The equation above is how the HDR image is calculated using the mapped out irradiance values g(Zij). The irradiance values were first solved for each image using 100 random pixel values from various points in the image. Once that was calculated, it was used in the equation above. w(Zij) is again using the weighted function previously mentioned and ln(delta tij) is the natural log of the exposure time. Figure 6 shows the mapped out irradiance values for each pixel value (0 to 255) and channels (R,G,B). The plot is mostly smooth and continuously increasing, which is preferred.

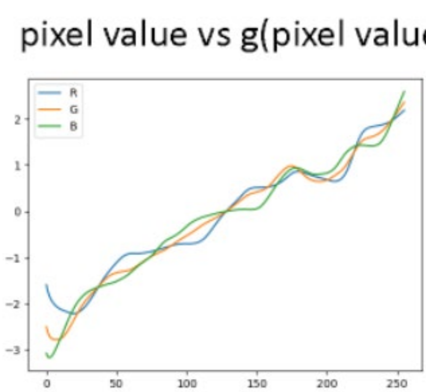

FIGURE 6: g(Z) plot

Figure 7 shows the resulting HDR image along with the log irradiance images.

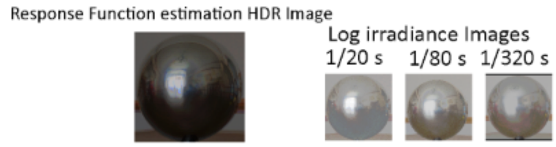

FIGURE 7: Response Function estimation

Irradiance Discussion

In the case of the images provided in Figure 2, the irradiance images seem to be different among the 3 methods used. For naïve, only a weighted average of the three images is done, without accounting for over or under exposed regions. The Response estimation function yields a representation based on most of the pixel values. The three methods would probably yield a similar irradiance image in the case of a photo with an already well distributed irradiance in low light conditions.

Panoramic Transformation

A panoramic transformation was created using the naïve HDR outputted image. First, the normal vector from the mirror ball is calculated. The equations for solving for the x, y and z coordinates are shown below. The assumption that is made is that the center of the image is the center of the image coordinates.

$$Nx = (u - center) / (width/2)$$
$$Ny = (v - center) / (height/2)$$
$$Nx^2 + Ny^2 + Nz^2 = 1$$

Once the Normal vector is solved for, the Reflection vector is solved for using the equation below. V, or the viewing vector, is determined to be [0,0,-1].

$$\begin{aligned} R &= -V+2*(dot(-V,N)*N - (-V)) \\ &= V - 2*dot(V,N)*N \end{aligned}$$

An illustration of how the different vectors look with respect to the ball are shown in Figure 8.

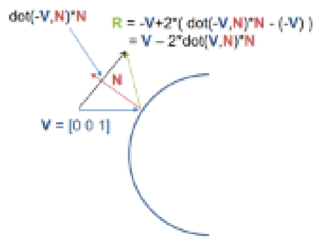

FIGURE 8: R, V, N vector calculations

The final panoramic image is created using the reflection vector R and the naïve HDR image. Figure 9 shows the equi triangle image (panoramic image output). Figure 10 shows the Normal, Reflection, and Phi-Theta images.

Equitriangle image

FIGURE 9

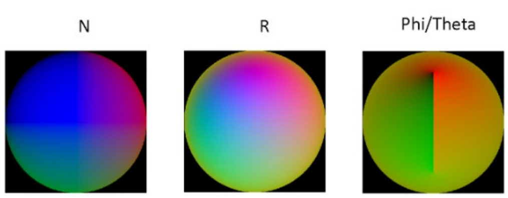

FIGURE 10

Image-based object relighting

The software Blender was used to add objects in the background image shown in Figure 1 and the panoramic image was used as the environmental textures for the objects added in the photo. Figure 11 shows the intermediate renderings, both with and without the rendered objects, and the object mask. Figure 12 shows the final result with original background image and final composited result.

FIGURE 11: Rendered Results

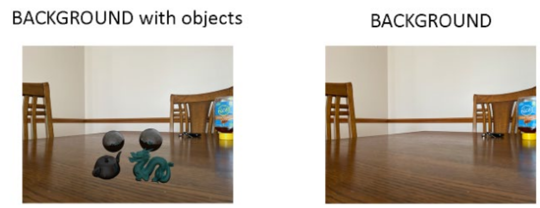

Figure 12: Final Result

Bells & Whistles

Additional Image-Based Lighting Result

Different objects were again added using the same HDR light map as seen in figures 11 and 12. Figure 13 shows the different renderings with and without the objects. Figure 14 shows the final results.

FIGURE 13: Rendered Results

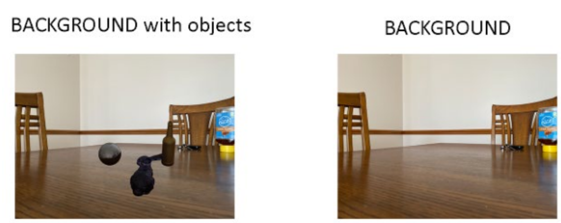

Figure 12: Final Result

Points

50/50 LDR-to-HDR merging

10/10 panoramic transformation and final equirectangular image and image-based light compositing result

30/30 Image-based object relighting result

10/10 quality of results and project page

10/20 Additional Image-Based Lighting Result

110/100 TOTAL