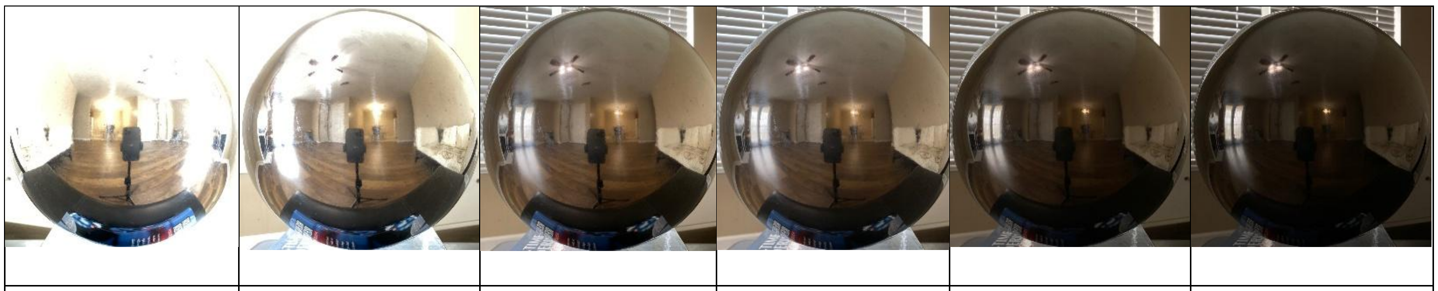
Name (netid): Byeong Kyu Park (bkpark2)
CS 445 - Project 4: Image Based Lighting

Complete the claimed points and sections below.

Total Points Claimed	**[195] / 210**
Core	
1. Recovering HDR maps	
a. Data collection	[20] / 20
b. Naive HDR merging	[10] / 10
c. Weighted HDR merging	[15] / 15
d. Calibrated HDR merging	[15] / 15
e. Additional HDR questions	[10] / 10
2. Panoramic transformations	[10] / 10
3. Rendering synthetic objects	[30] / 30
4. Quality of results / report	[10] / 10
B&W	
5. Additional results	[20] / 20
6. Other transformations	[20] / 20
7. Photographer & Tripod removal	[10] / 25 (used cv2.inpaint internally)
8. Local tone-mapping operator	[25] / 25

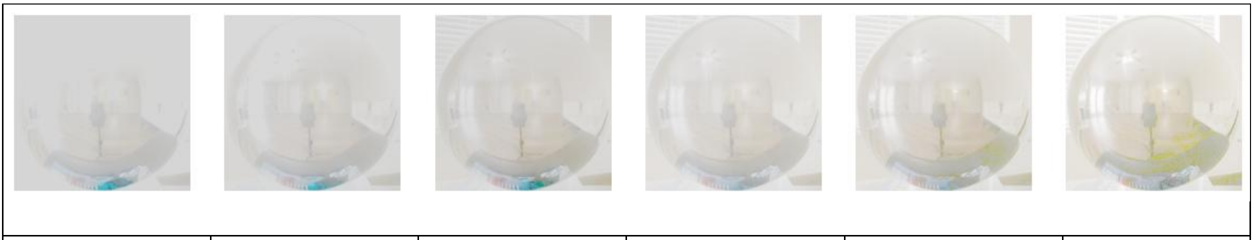
1. Recovering HDR maps

(a) LDR images



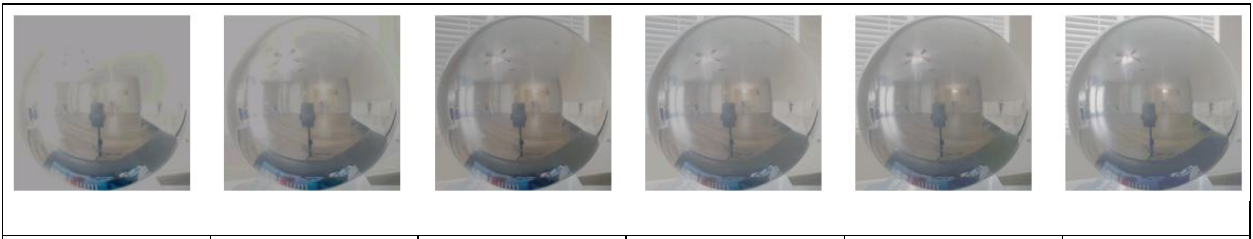
1/30s	1/40s	1/60s	1/120s	1/210s	1/442s

(b) Figure of rescaled log irradiance images from naive method



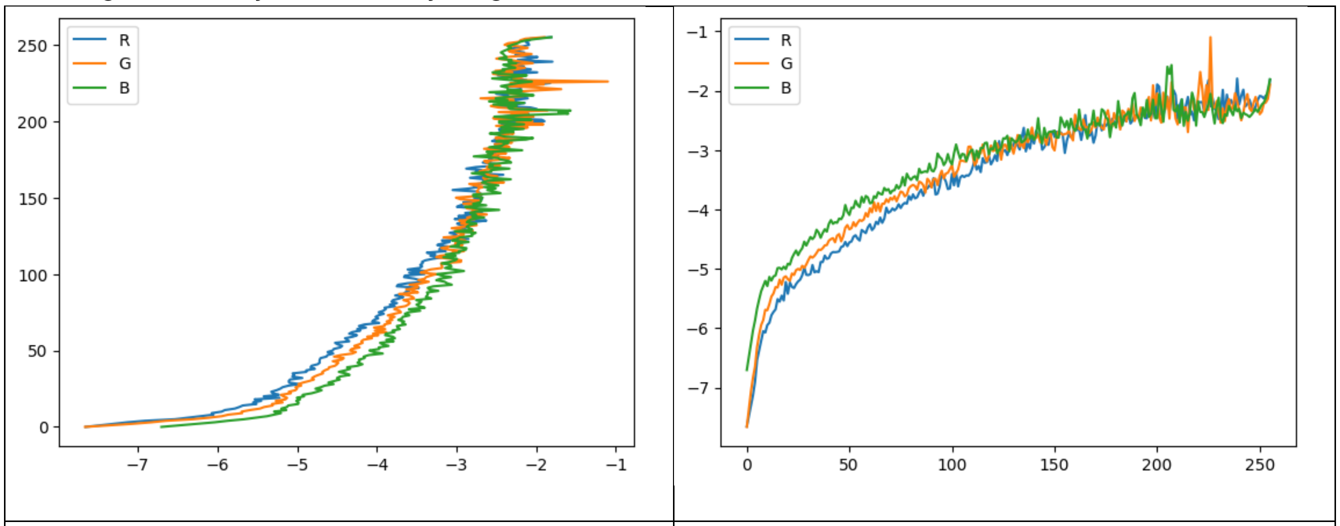
1/30s	1/40s	1/60s	1/120s	1/210s	1/442s

(c) Figure of rescaled log irradiance images from calibration method



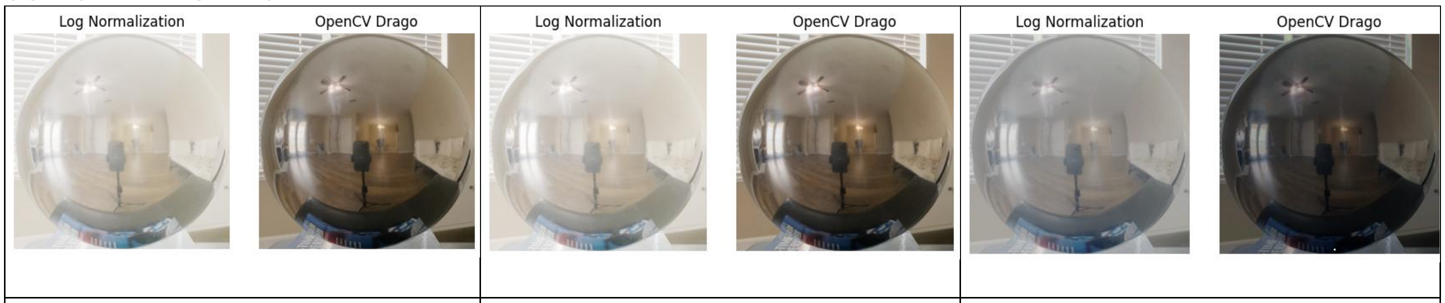
1/30s	1/40s	1/60s	1/120s	1/210s	1/442s

(d) Plots of g vs intensity and intensity vs g

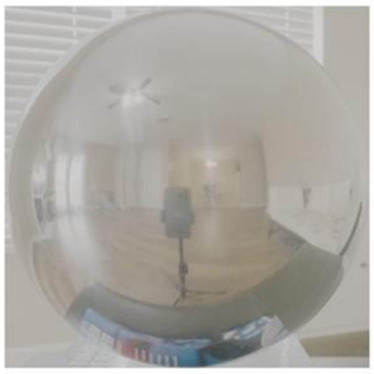
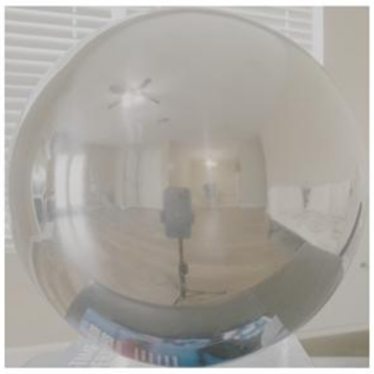
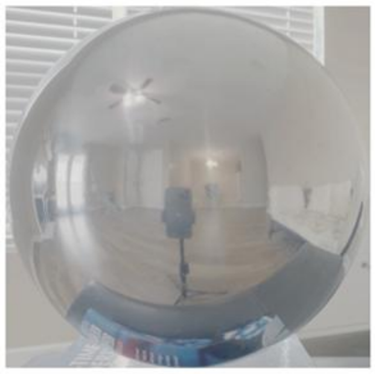


g vs intensity	**intensity vs g**

(e) Figure comparing the three HDR methods



naive	**weighted**	**calibrated**

		
naive	**weighted**	**calibrated**

- (f) Text output comparing the dynamic range and RMS error consistency of the three methods

```
     naive:       log range =  7.403     avg RMS error =  0.335
  weighted:       log range =  7.466     avg RMS error =  0.312
calibrated:       log range =  8.222     avg RMS error =  0.265
```

- (g) Questions

1. For a very bright scene point, will the naive method tend to over-estimate the true brightness, or under-estimate? Why?

The naive method will underestimate the true brightness.

In longer exposure images, more bright points get clamped to the maximum intensity that the LDR sensor can express, causing the true brightness to be lost (forever).

While averaging across exposures might mitigate the issue for moderately bright points, it isn’t robust enough for most extremely bright ones. For example, assume the true radiance is 100 and you take photos at 1/150s, 1/100s, and 1/50s. In an ideal linear response, these exposures might yield intensities of roughly 0.67, 1.0, and 2.0. However, since 2.0 exceeds the sensor’s limit, it’s clamped to 1.0. Averaging the values gives (0.67 + 1.0 + 1.0)/3 ≈ 0.89, which still underestimates the true brightness.

2. Why does the weighting method result in a higher dynamic range than the naive method?

Both the naive average and weighted methods assume that the recorded intensity is linearly proportional to the true radiance times the exposure time ($Z = R \cdot t$), but this relationship isn’t strictly linear. The weighted method improves on the naive approach by emphasizing “trustworthy” intensities—those in the midrange where the sensor’s response is approximately linear and less likely to be clamped. In other words, it approximates the true non-linear response by using segments of linear behavior, which better captures the broader range of actual intensities.

3. Why does the calibration method result in a higher dynamic range than the weighting method?

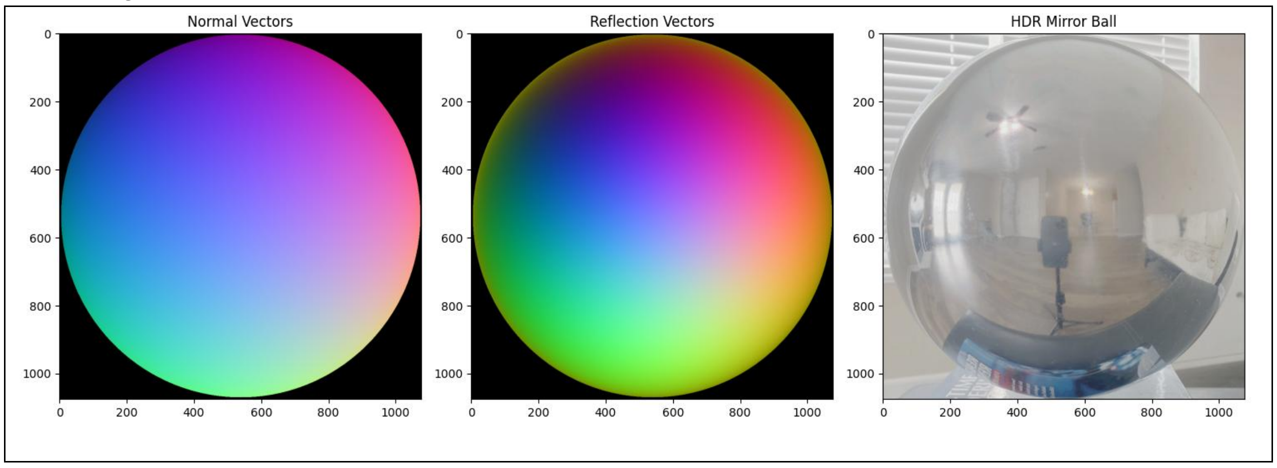
The calibration method yields a higher dynamic range compared to the weighting method because it does not assume a simple linear relationship between pixel values and radiance. Rather than using the relation $Z = R \cdot t$, the calibration method determines the actual response function of the camera, expressed as $Z = f(R \cdot t)$. By inverting this function to get $f^{-1}(Z) = R \cdot t$ and then taking the logarithm, we arrive at $\ln(f^{-1}(Z)) = \ln(R) + \ln(t)$. This process linearizes the relationship, allowing the recovery of true radiance values even in regions where the sensor would normally saturate. In contrast, the weighting method only trusts intensities in the midrange and cannot fully compensate for saturation, which limits its effective dynamic range.

4. Why does the calibration method result in higher consistency, compared to the weighting method?

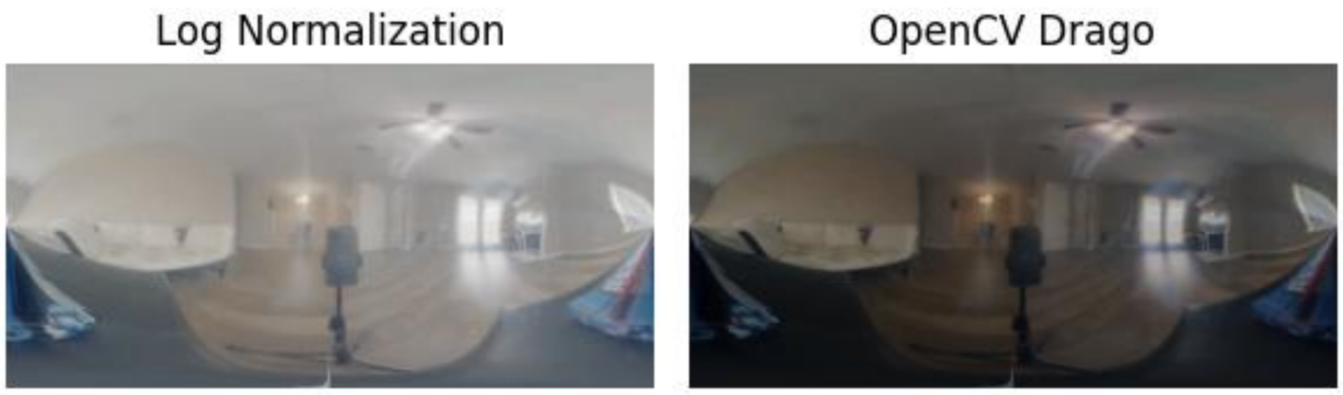
Instead of using a heuristic approach, the calibration method finds a solution that best explains all observations simultaneously through a global optimization approach. In addition, using a logarithmic transformation increases robustness to errors, and the introduction of constraints helps reduce noise.

2. Panoramic transformations

- The images of normal vectors and reflectance vectors

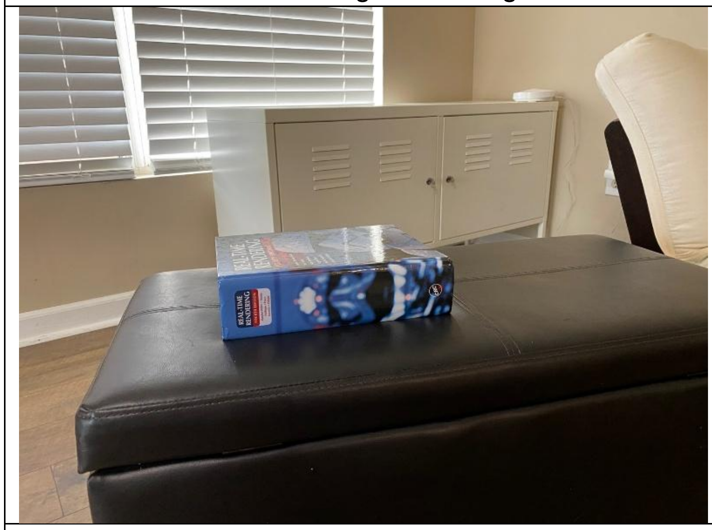
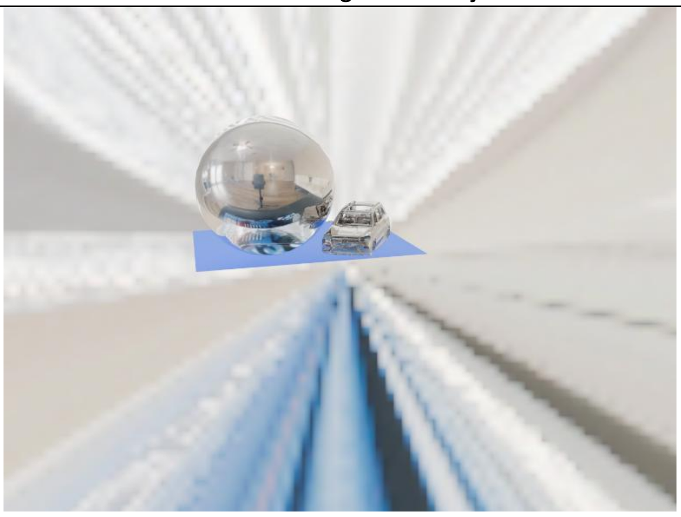
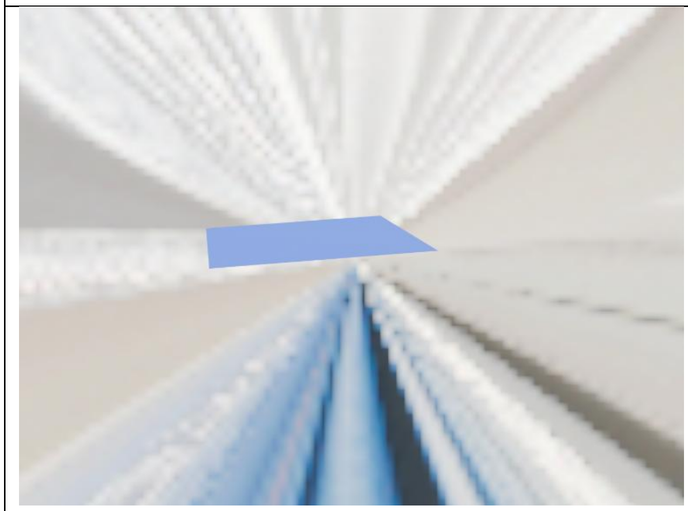
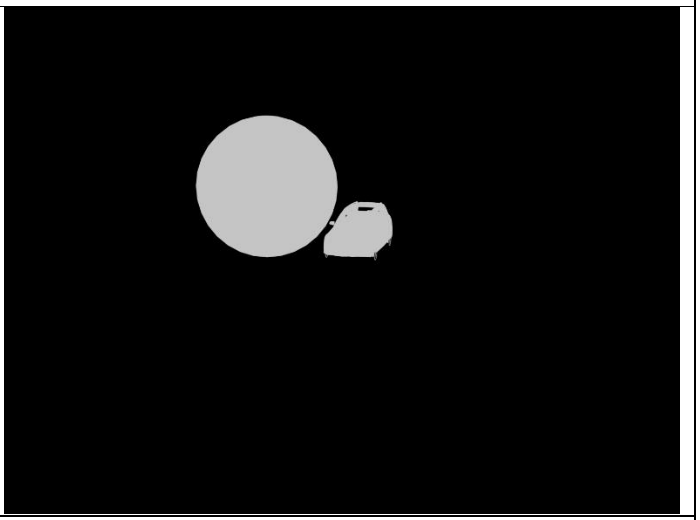


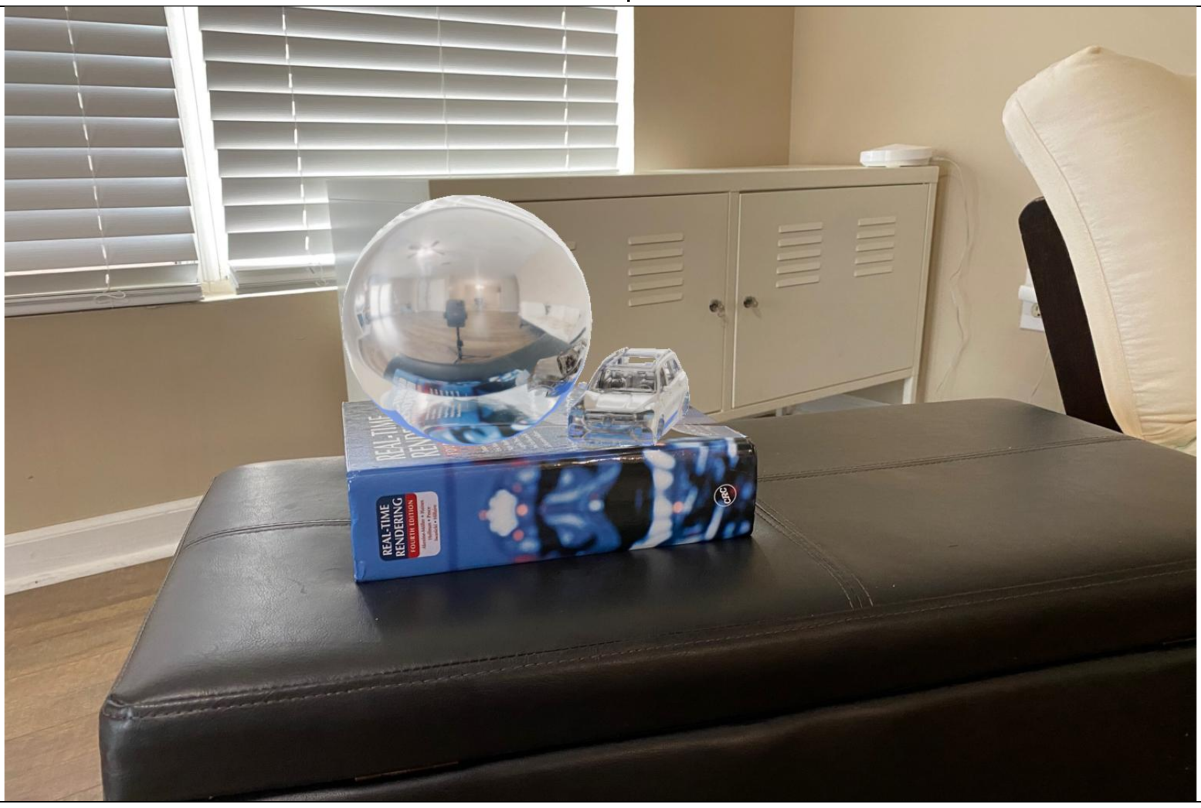
- The equirectangular image from calibration HDR result



3. Rendering synthetic objects

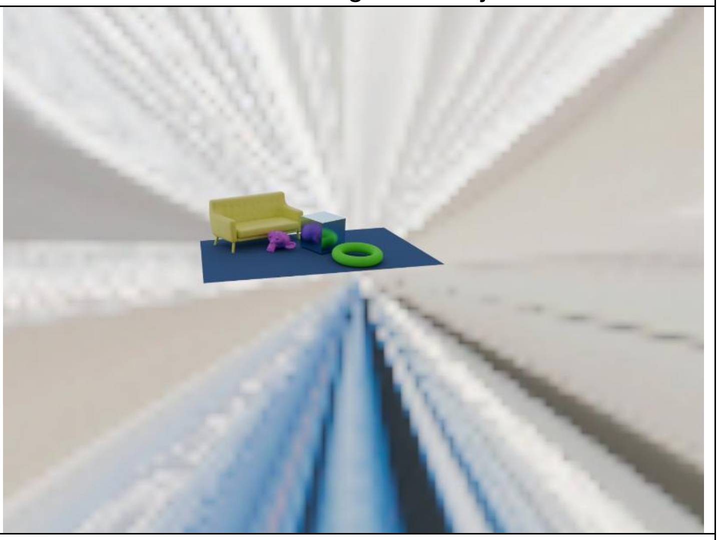
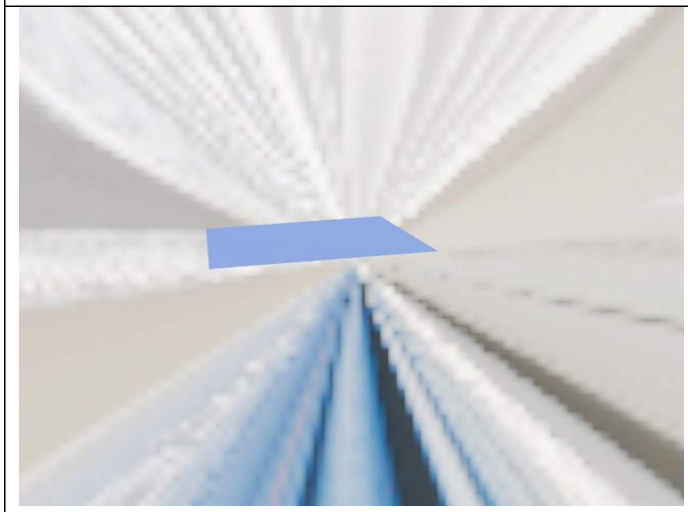
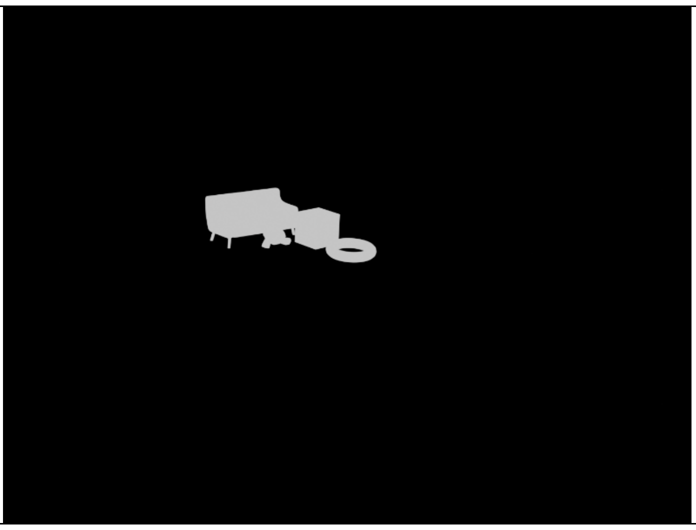
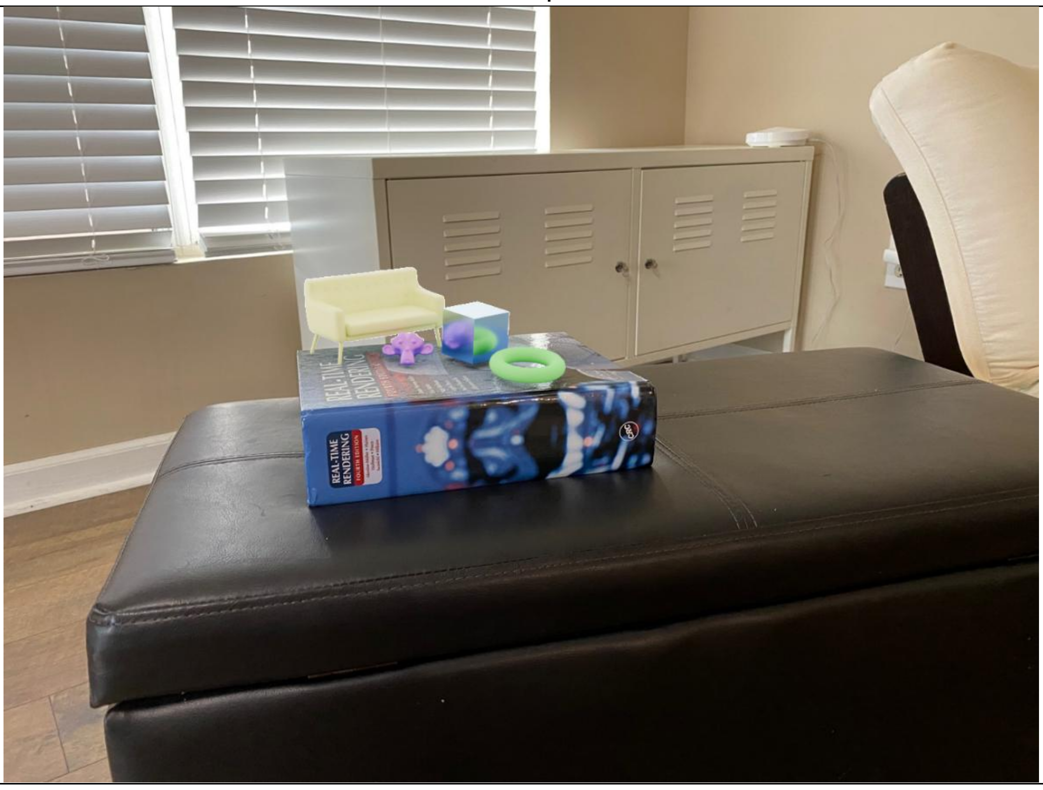
- Component images

Background image	Rendered image with objects
	
Rendered image with support plane	Rendered mask image
	

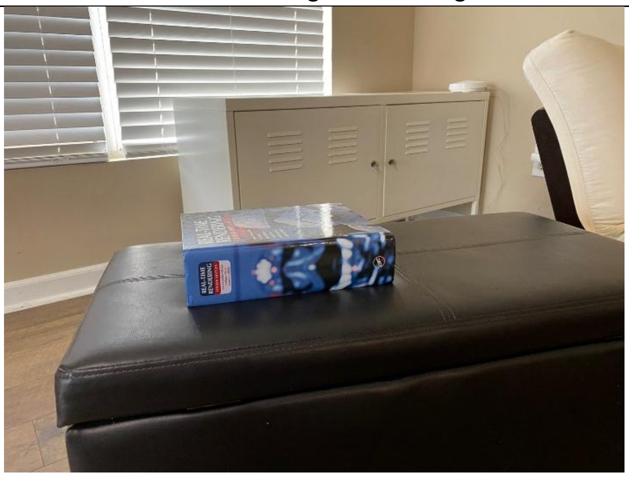
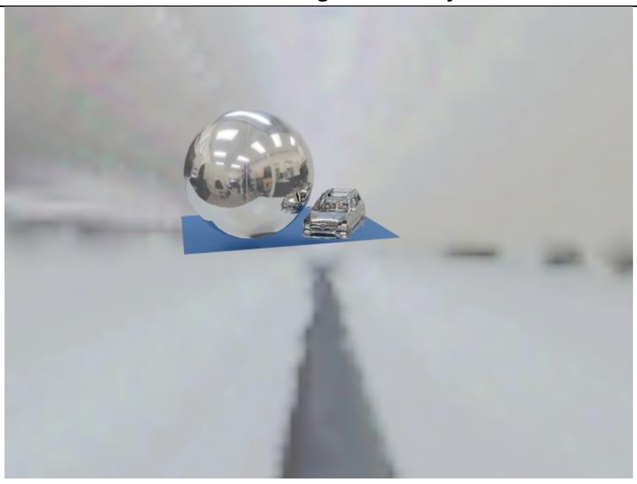
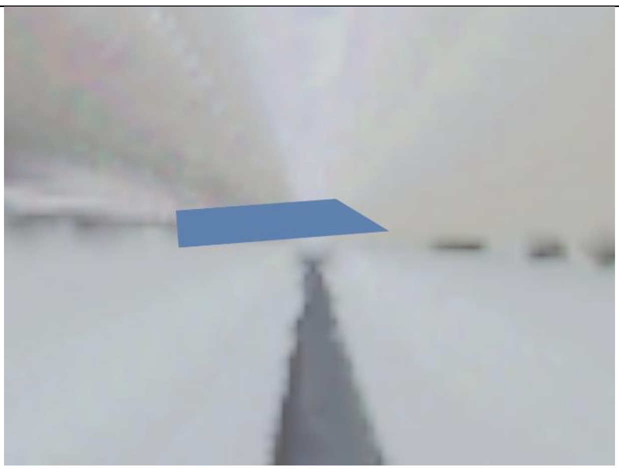
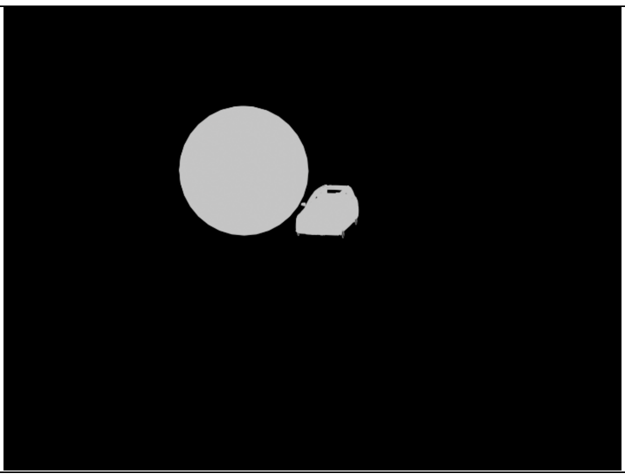
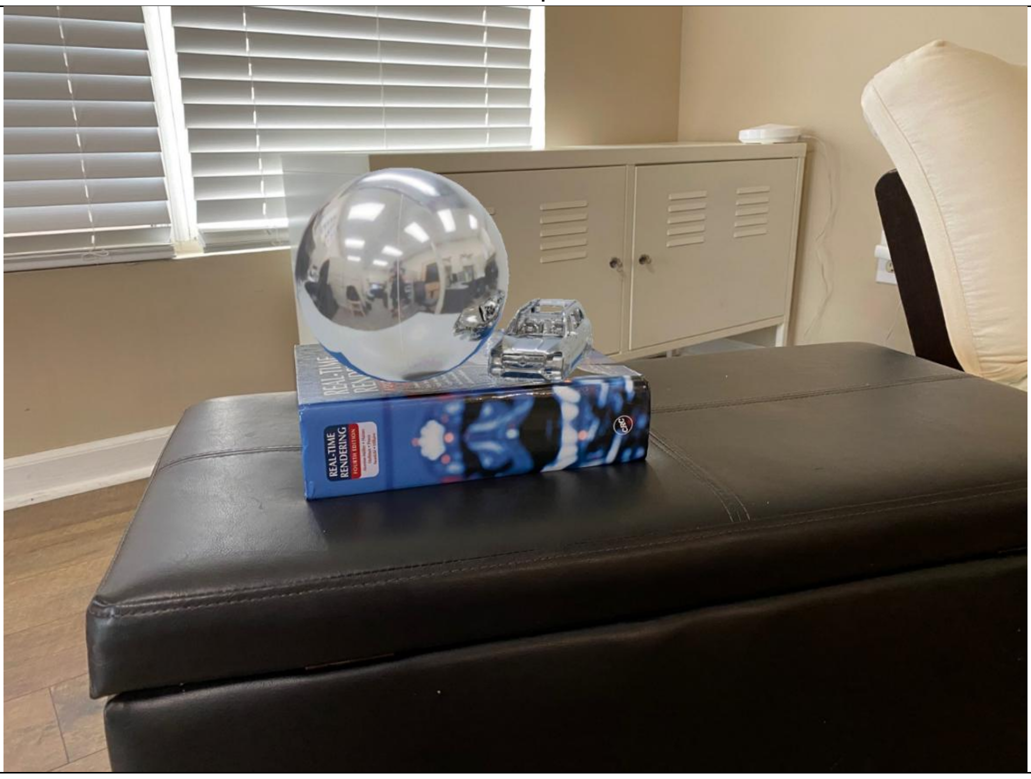
Final composited result


4. Additional results (B&W)

- New objects, same environment map

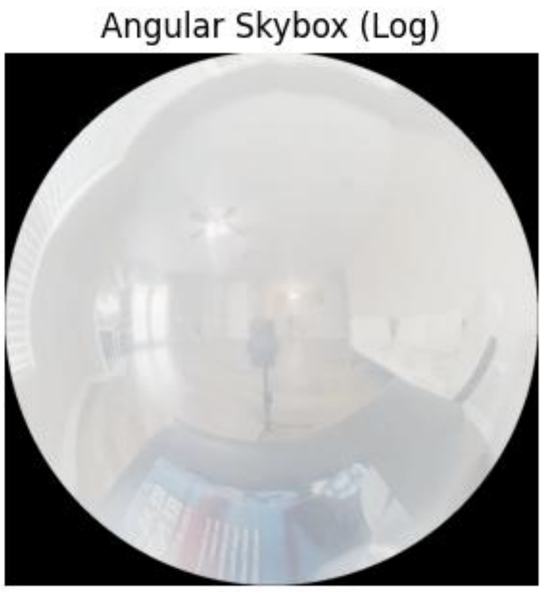
Background image	Rendered image with objects
	
Rendered image with support plane	Rendered mask image
	
Final composited result	
	

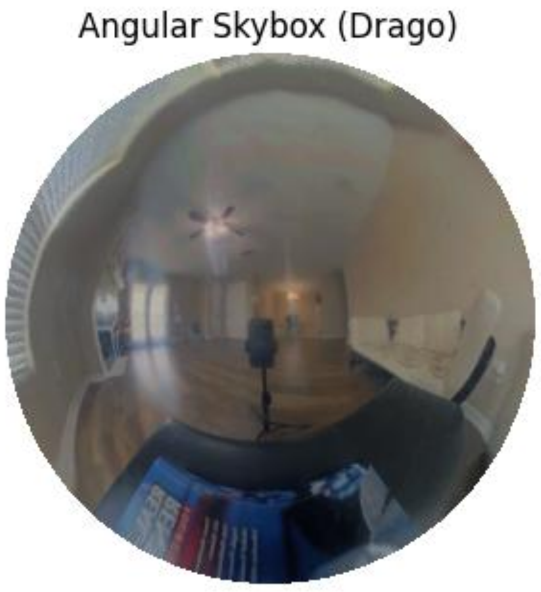
- New environment map, same objects

Background image	Rendered image with objects
	
Rendered image with support plane	Rendered mask image
	
Final composited result	
	

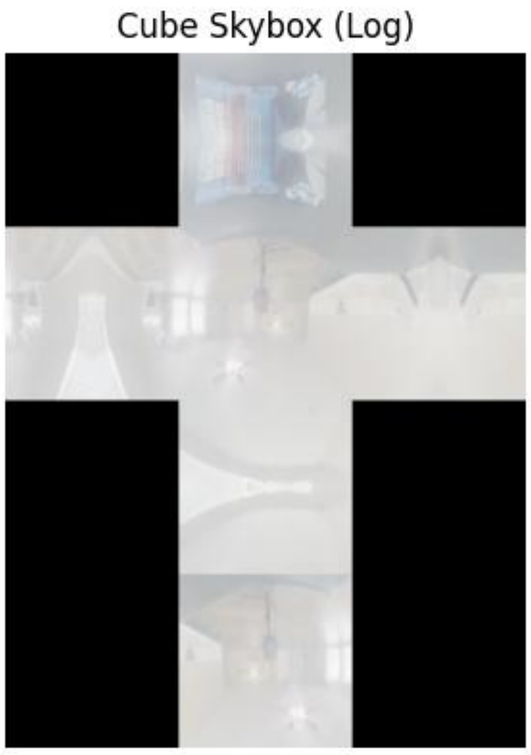
5. Other transformations (B&W)

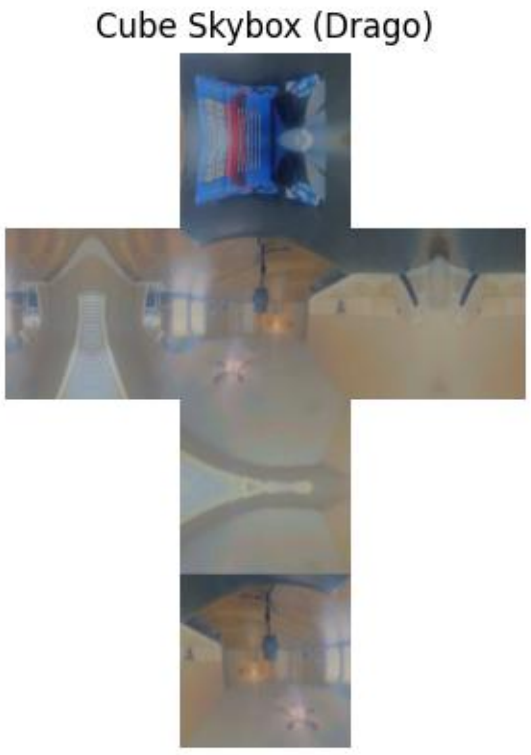
- Angular environment map





- Vertical cross environment map



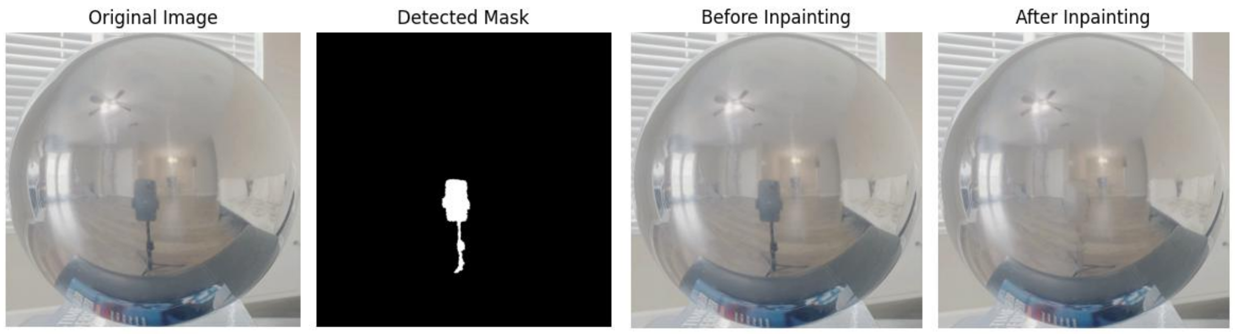


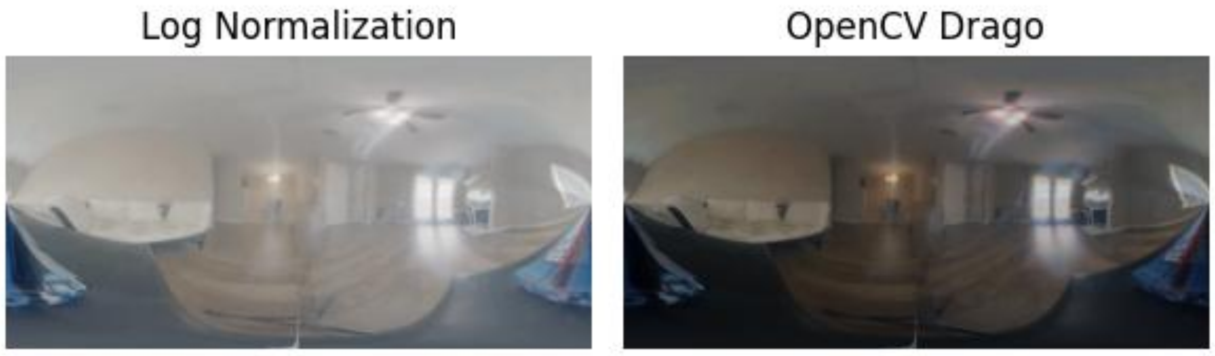
6. Photographer and tripod removal (B&W)

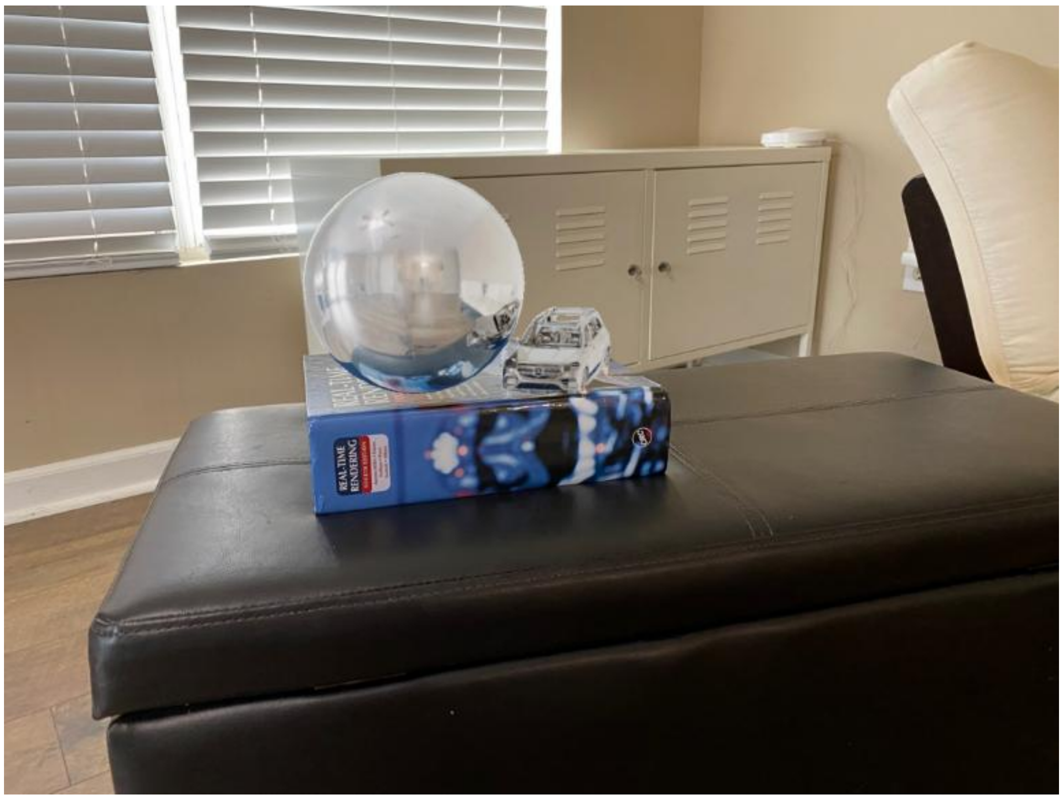
- Original LDR images

1/30s	1/40s	1/60s	1/120s	1/210s	1/442s

- Equirectangular without photographer(camera)





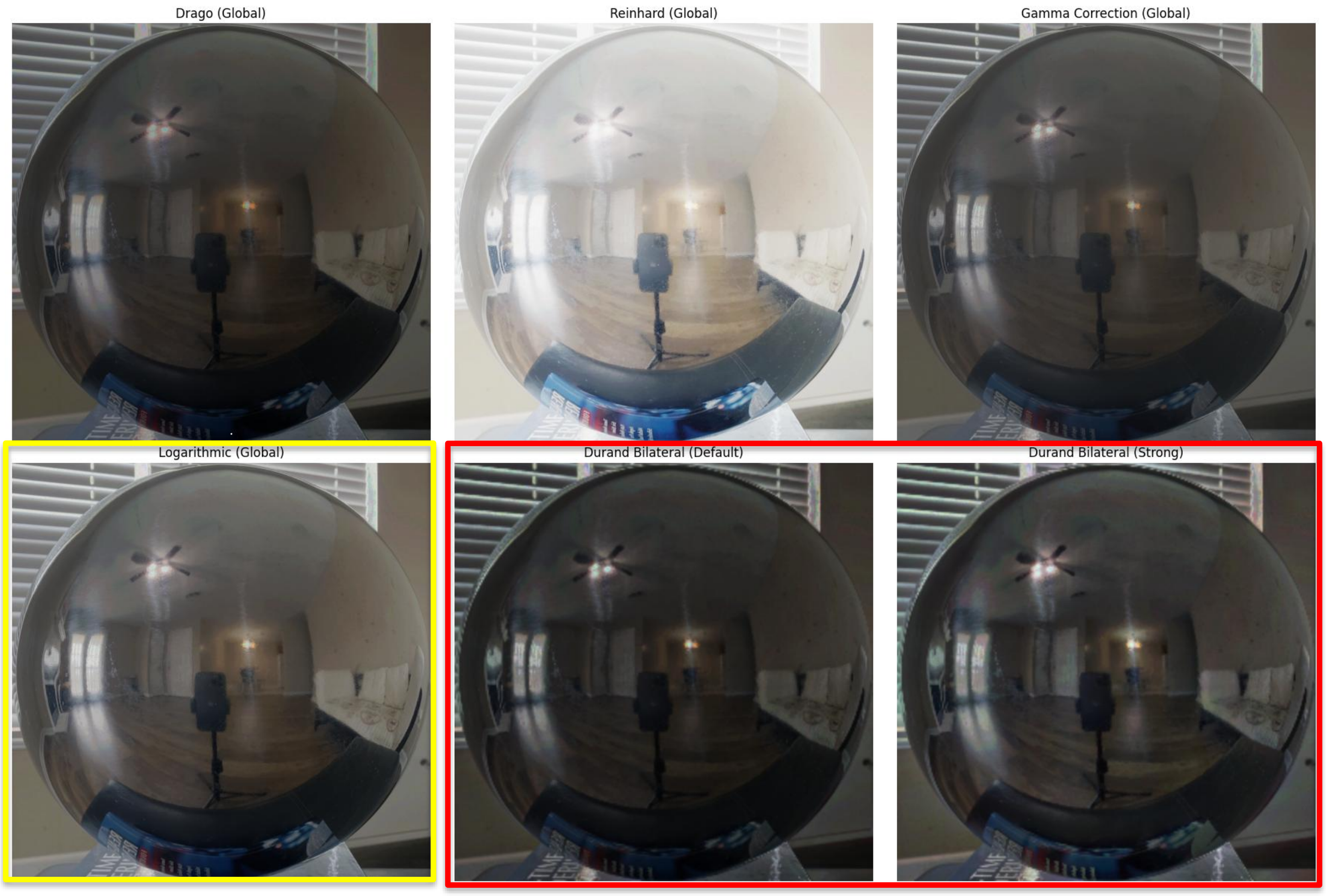


- Explanation

Manually set a bounding box surrounding the camera and tripod, then used GrabCut to segment them.
For each R, G, B channel, applied OpenCV's inpaint() method (since my custom method from Assignment 2 only worked for circular holes).
The inpainting used the GrabCut mask as a hint for where the object was located (based on the bounding box).

7. Local tone-mapping operator (B&W)

- tone mappings


Drago (Global)
Reinhard (Global)
Gamma Correction (Global)
Logarithmic (Global)
Durand Bilateral (Default)
Durand Bilateral (Strong)

- Explanation

First, we linearize the HDR image by taking the log base 10. Then, for each color channel, we apply a bilateral filter to the log image to extract the base (global) layer while preserving edges (local contrast). We compute the detail layer as the difference between the original and the base. After compressing the base layer, we recombine it with the detail layer to preserve local contrast and convert it back to linear space.

Acknowledgments / Attribution

[Objects]

1. Couch Sofa: https://free3d.com/3d-model/couch-sofa-742041.html
2. Car (Mercedes-Benz GLS 580 2020): https://free3d.com/3d-model/mercedes-benz-gls-580-2020-83444.html

[AI Tools]

Copilot (via VS Code)

: Assisted with summarizing functions, writing template code, and generating code comments.

[References]

1. Cube Mapping – Wikipedia: https://en.wikipedia.org/wiki/Cube_mapping
2. Mirror Ball, Angular Map & Spherical Map Comparison (PDF): https://horo.ch/docs/mine/pdf/Mb-Am-Sph.pdf
3. Paul Debevec – High Dynamic Range Imaging Research: https://www.pauldebevec.com/Research/HDR/