# Computational Photography - CS 445
## Project 4: Image Base Lighting

by

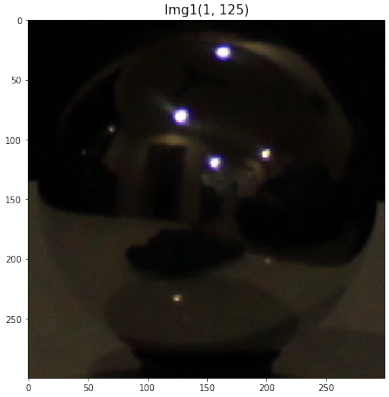
Deepti Sharma (deeptis2@illinois.edu)

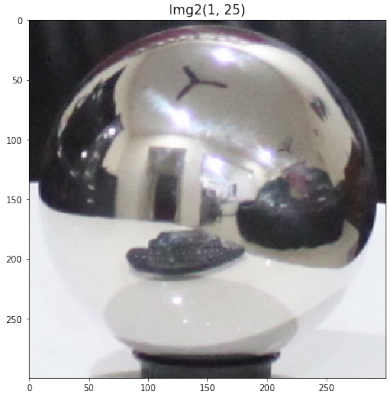
## Table of Contents

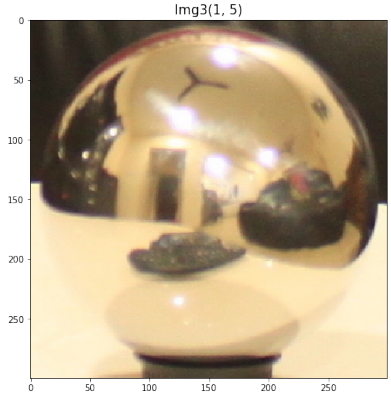
# Deliverables

## 1. Recovering HDR Radiance Maps (50 pts)

**Input Images:**




### Naïve:

Scaled the images by dividing each one with its exposure times respective and doing and average of the resultant images. The results are good enough.

### Filtered

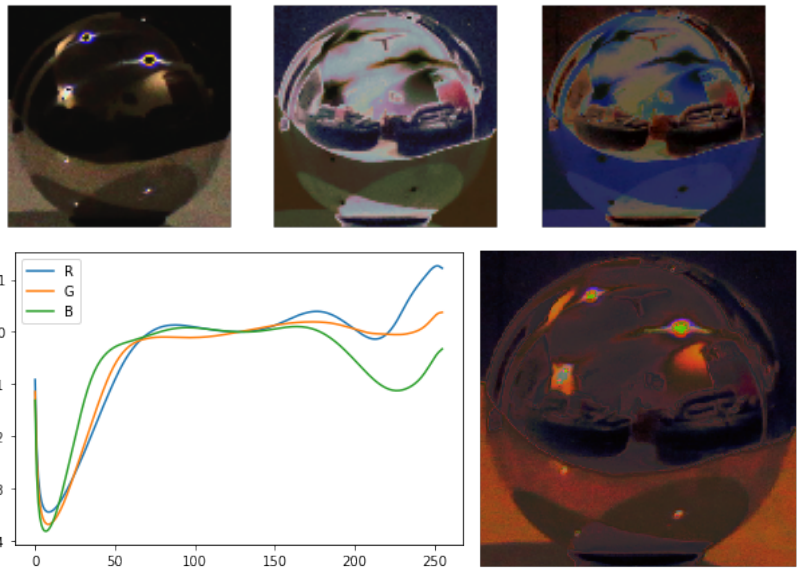
Scaled the images by dividing each one with its exposure times respective. Additionally used a. weight function (w = lambda z: float(128-abs(z-128)) , also excluded values z < 5 and z > 250. The result looks much better.

## HDR Estimates

This one has been inconsistent. When I solve for g, all color channels do not increase consistently for different images. To make it g curve smooth, I tried increasing the value of lambda from 1 to 10

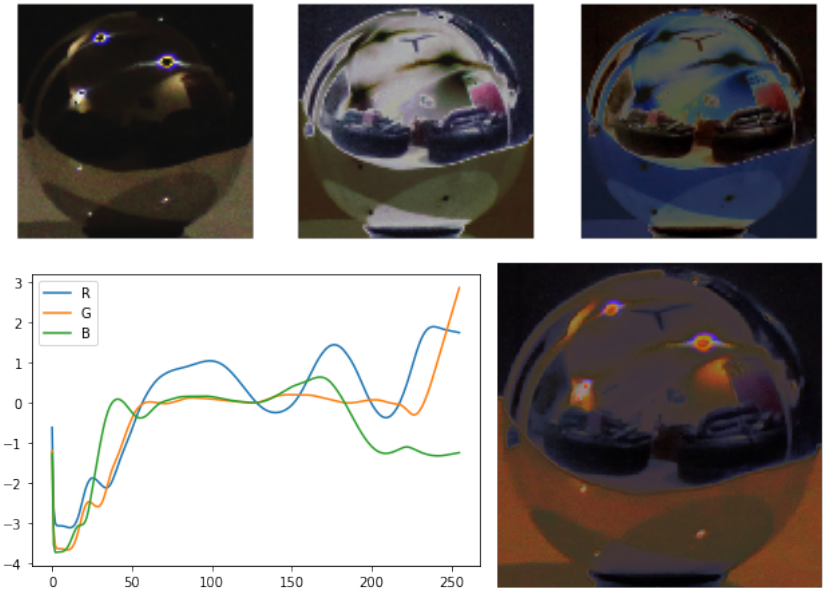
### With Lambda =1

```
Log Irradiances min:  -28.509755868807936  max: 627.9938738867953
```



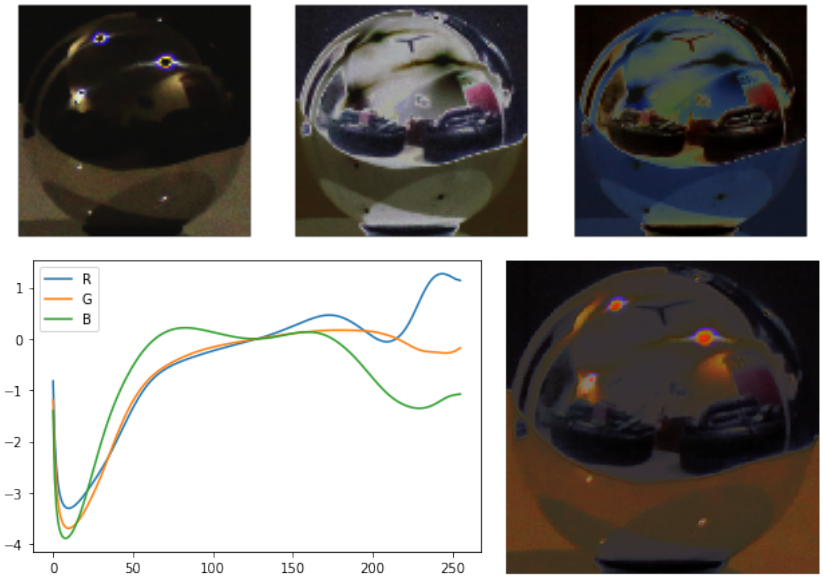

### With Lambda =5

```
Log Irradiances min:  -35.68689512393159  max: 618.0241583772918
```



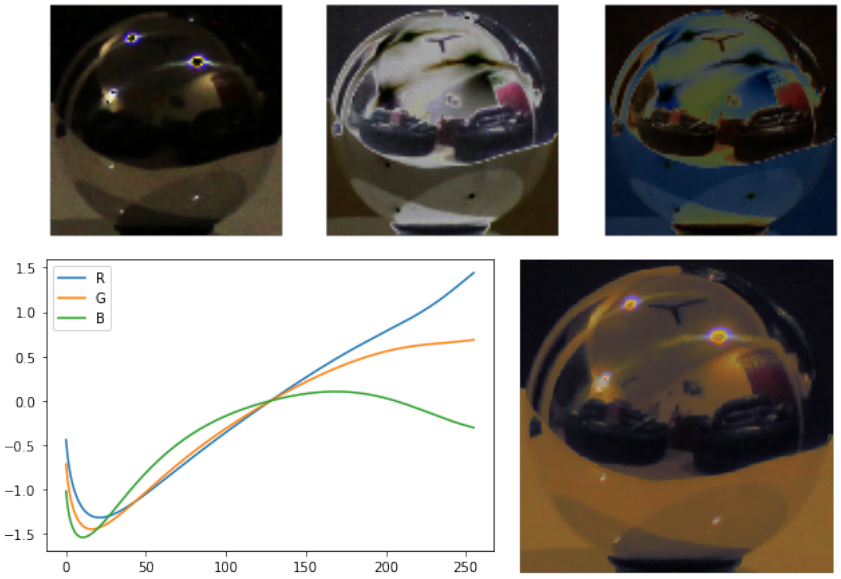
## With Lambda =10

```
Log Irradiances min:  -30.734501543128722  max: 618.0241583753722
```



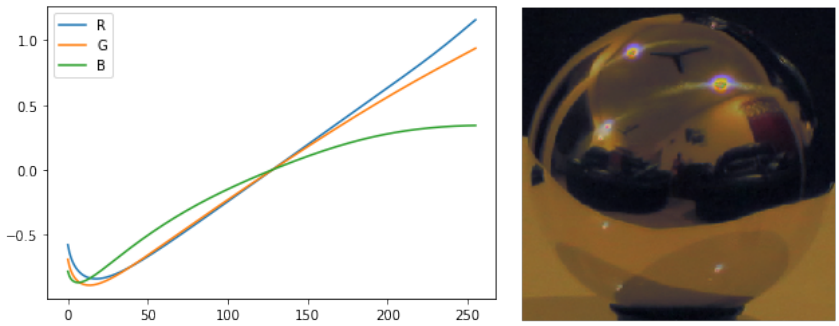

## With Lambda=50

```
Log Irradiances min:  0.0  max: 618.0241583771647
```



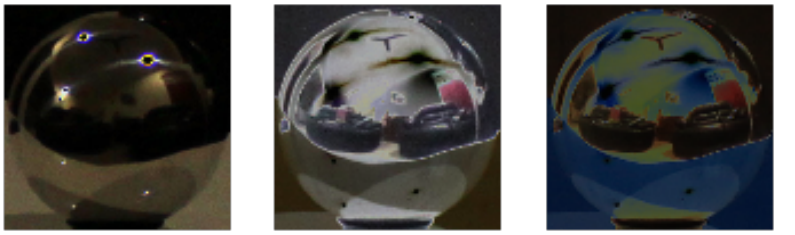
## With lambda=100

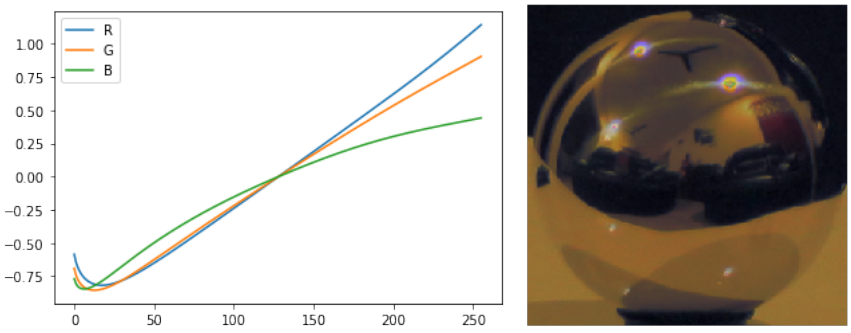
```
Log Irradiances min:  0.0  max: 618.0241583808596
```




## With lambda=110

```
Log Irradiances min:  0.0  max: 618.0241583657507
```

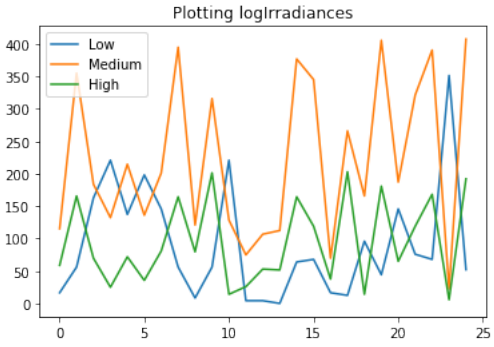


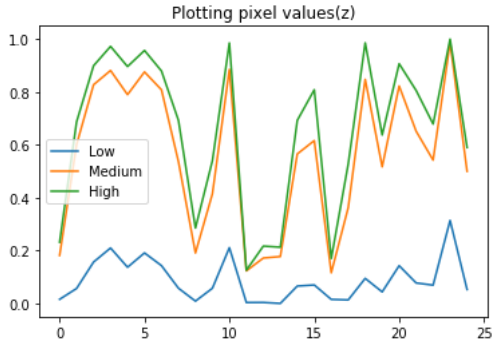


## Irradiance Discussion

g(z) = log (irradiance) + log(exposure time)
log (irradiance) = g(z) - log(exposure time)

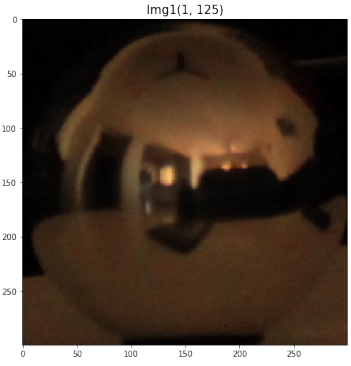
As we can see, when the values are too high or two low in either of the exposure images, they do not align. Where the z values are more distributed, log irradiance values tend to converge.

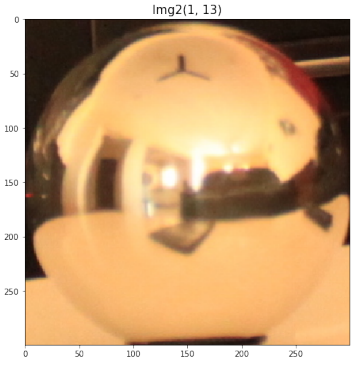


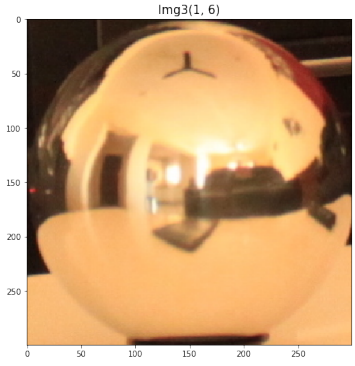




## IBL HDR Results Details:

### More results:



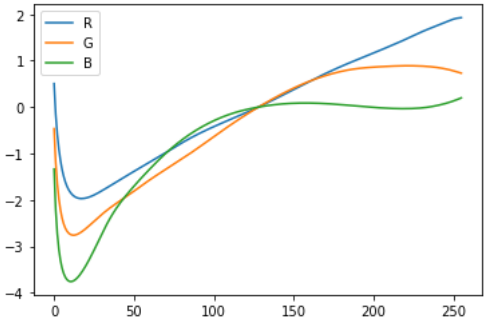



Irradiances LDR Images

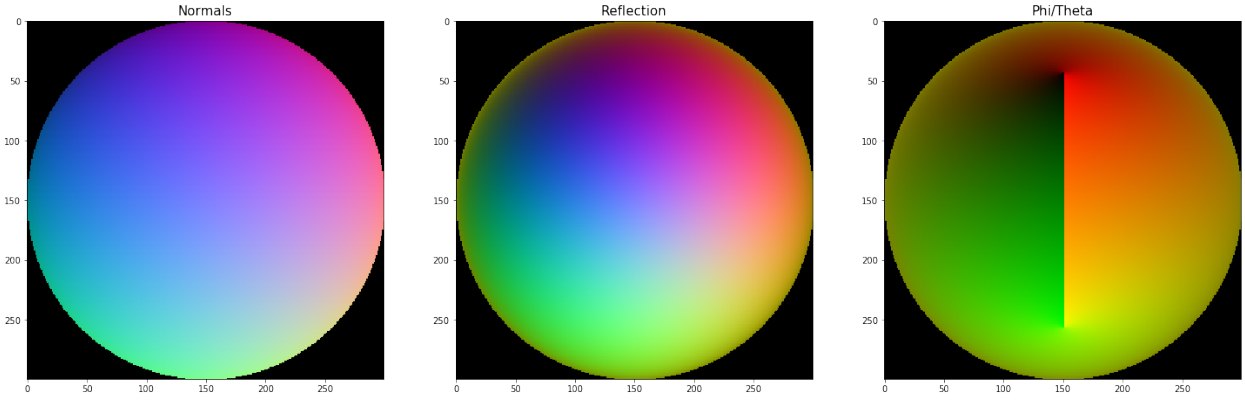
Naive Filtered HDR Estimates

Irradiances Estimated (gsolve with lambda = 10)

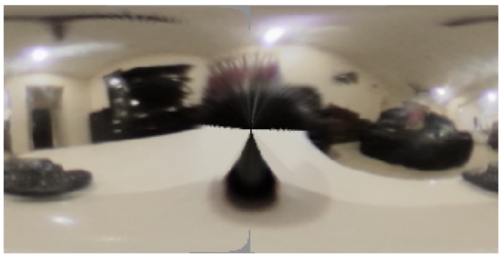


## 2. Panoramic transformations (10 points)

Using Filtered HDR for Panoramic transforms




Using Provided function



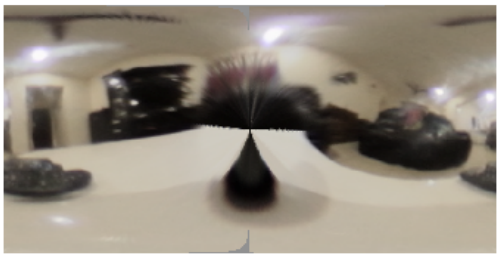
Made 1 line change (phi_ball += math.pi/2 )

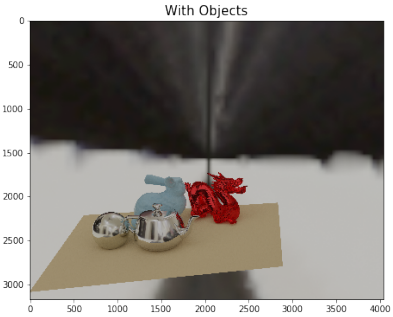
## 3. Rendering synthetic objects into photographs (30 pts)

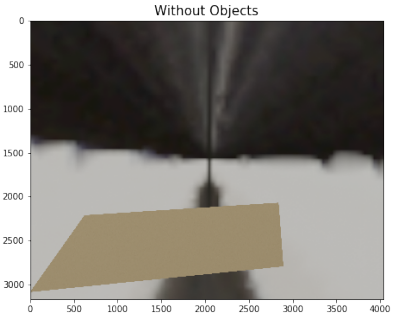
### Results1

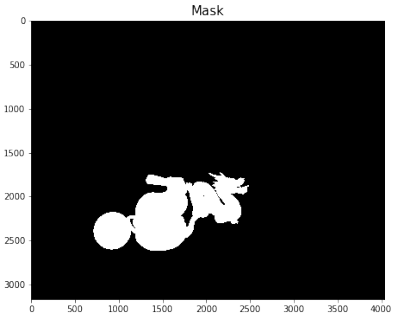
composite = M*R + (1-M)*I + (1-M)*(R-E)*c

HDR and Equirectangular Image

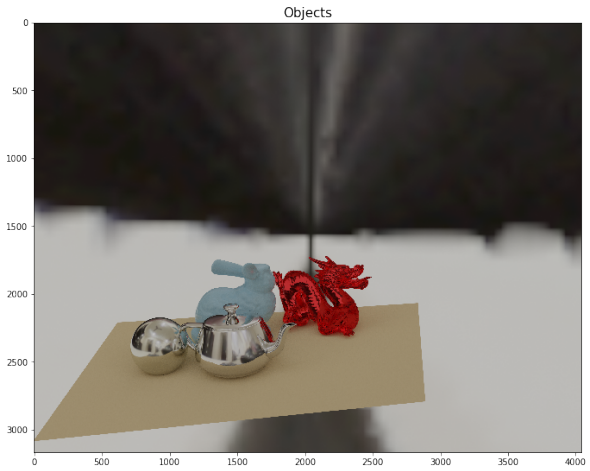








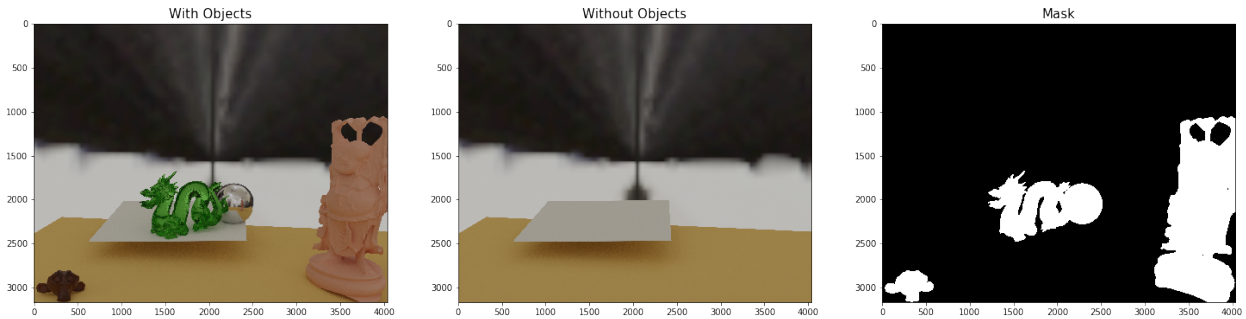


# Bells & Whistles (Extra Points)

## Additional Image-Based Lighting Result (20 pts)

### Results2

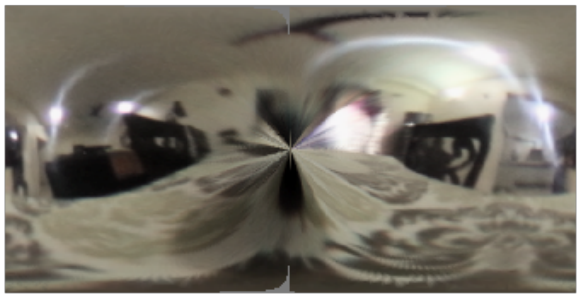
Object placement with two planes

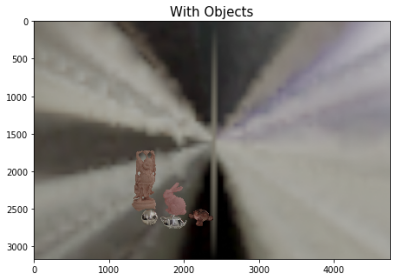


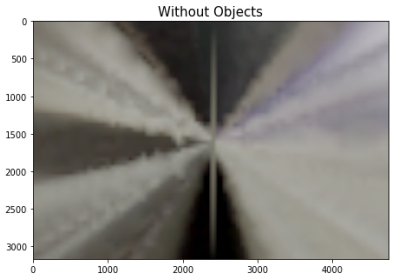


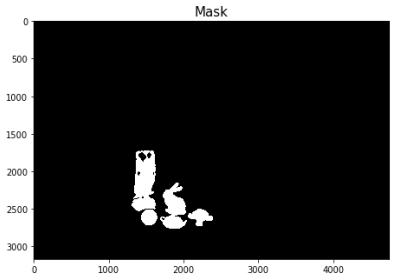
## Results3

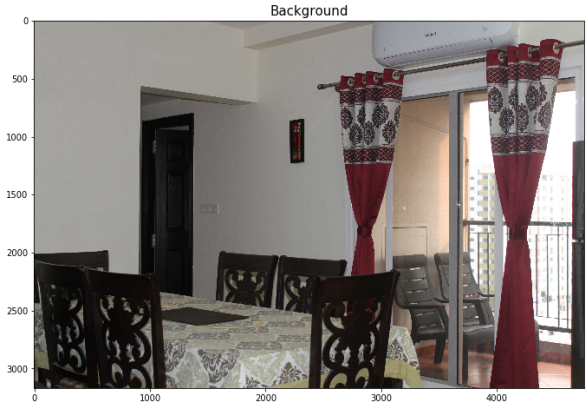
Object placement was done using a plane but due to textured suface, removed the plane before rendering to include texture reflections

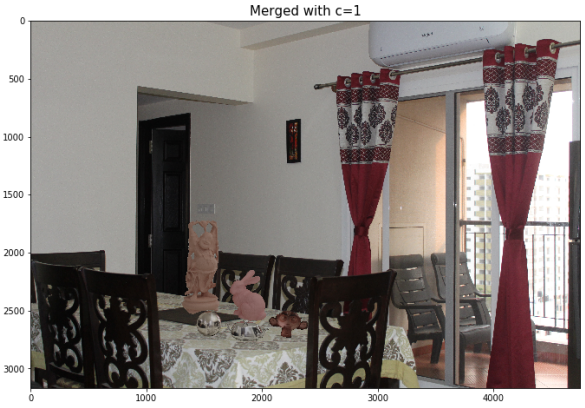




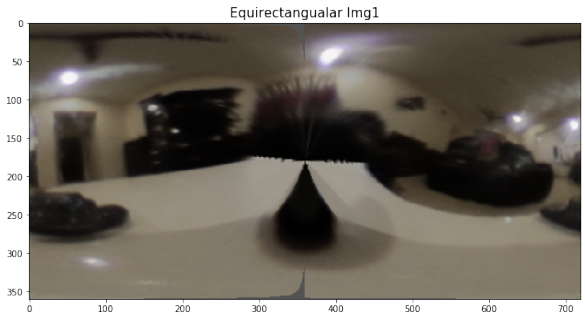



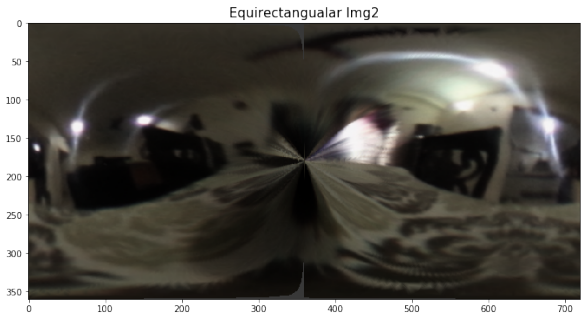



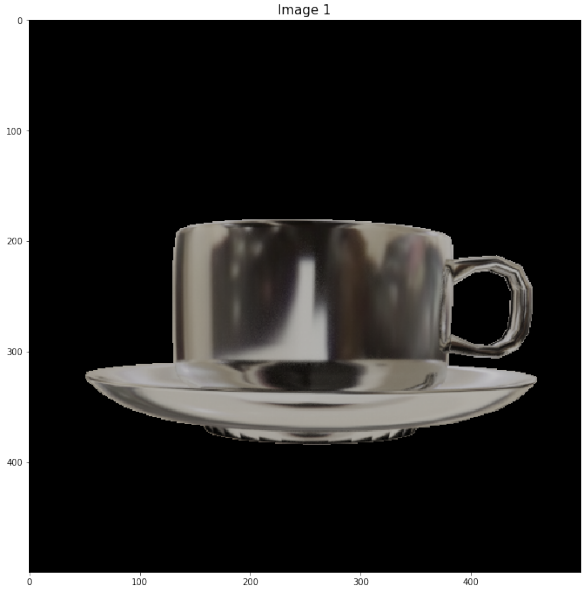


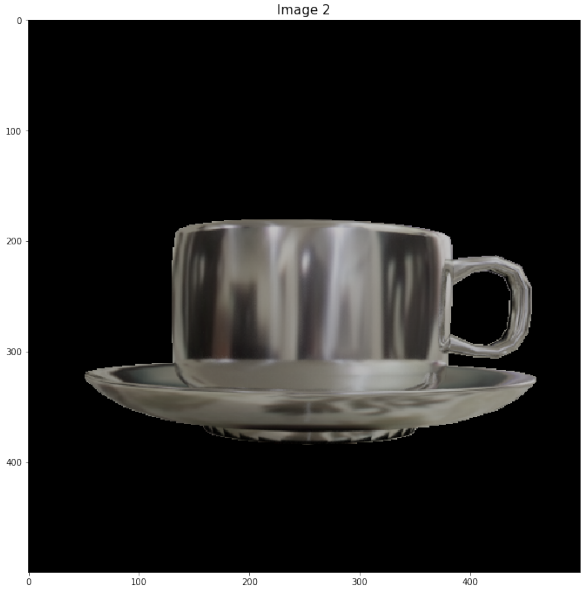




## Results 3: Just for Fun



## Local tone-mapping operator (30 pts)

For tone mapping, follow methods were applied.

### Rescaling and Reinhart

Rescaling and Reinhart Operator

Original Rescaled Reinhart

### Bilateral filter

Filter was applied to Rescaled Image, with Laplacian Edge Detector