Estimating the Camera Response Function of an iPhone 15 Pro to create new HDR Images

Audrey Lee [1] Lycia Tran [1]

[1]Electrical Engineering, Stanford

Motivation

High Dynamic Range (HDR) images are images that balance lighting within an image to more accurately match what is perceived by the human eye. In recent years, the average person uses their mobile phone to capture images, and while most phones seem to have their own "HDR Mode", these HDR images do not seem to be significantly better or more balanced. Our project aims to estimate the camera response function of the most accessible digital camera – your smartphone – and create our own HDR image pipeline to produce balanced images that can be more interchangeable to suit different people's tastes in a final image. [2]

Related Work

From Debevec's method on recovering the camera response function, we wanted to see if we could recover the camera response of an iPhone given its limited range of exposure times and camera settings. [3]

Debevec utilizes a property relating the exposure times to the digital pixel value on what you see in the camera. From this property, we can then say that there is a non-linear mapping relating the exposure X to the digital value you see on the camera Z. This value Z is our camera response function. To find Z, we can first find the inverse of that camera response function g: $Z_{ij} = f(\Delta t_j E_i)$

$$\ln f^{-1}(Z_{ij}) = \ln \Delta t_j + \ln E_i \tag{1}$$

And since g can be defined as the natural log inverse of Z, we get

$$g(Z_{ij}) = \ln \Delta t_j + \ln E_i \tag{2}$$

We can then put this inverse camera response function into a properly weighted irradiance values equation. We can establish a scale factor w to estimate the pixel values relative to the minimum and maximum pixel values. If our z values are closer to the minimum or maximum, g(Z) will have a steeper curve, so we can introduce a smoothing factor λ to fit the data more. This results in the following objective function

$$O = \sum_{i=1}^{N}\sum_{j=1}^{P}\{w(Z_{ij})[g(Z_{ij}) - \ln \Delta t_j - \ln E_i]\}^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1}[w(z)g^H(z)]^2 \tag{3}$$

We can solve for g and E_i by framing it as a linear least squares problem to minimize our objective equation and obtain our irradiance mapping and inverse camera response function.

Project Theory

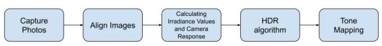

Figure 1. Flowchart Diagram of Implementation

As shown in Figure 1, our pipeline involves aligning images we took from capturing photos with different exposure times, calculating the camera response function for this iPhone camera, creating our own HDR image pipeline, and creating our own tone mapping methodology.

After capturing photos at different exposure times, we can estimate the camera response function by utilizing a property between photochemical and electrons where the exposure X is defined as the product of irradiance E and exposure time Δt, and then obtain our inverse camera response function g and irradiance mapping E to get Low Dynamic Range (LDR) images. [3]

Using these LDR images and exposures, we can merge these images into an HDR image. To do this we compute a weight or confidence value that pixel is well exposed.

$$w_{ij} = exp(-4\frac{(I_{lin_{ij}} - 0.5)^2}{0.5^2}) \tag{4}$$

Using these weights, we want to find a good estimation, $\hat{X}$, for our "true image", X. To this we have to minimize, O, which is the difference between our estimation, $\hat{X}$, at our different exposures, t_k and our linearized LDR images, I_{lin_k}.

$$O = \sum_k w_k(log(I_{lin_k}) - log(t_k\hat{X}))^2 \tag{5}$$

We can calculate the derivative of O to find our linearized HDR Image, $\hat{X}$:

$$\hat{X} = exp(\frac{\sum_k w_k(log(I_{lin_k}) - log(t_k)}{\sum_k w_k} \tag{6}$$

We can now use different tone mapping algorithms to map our linearized HDR image into our final HDR image. To do this we start with the simple equation below where s is our scaling factor and γ is used for gamma correction

$$I_{HDR} = (s * I_{HDR_{lin}})^\gamma \tag{7}$$

From this we chose to add a bias term, b, to see if that would improve our final HDR image. We use bias term used in Drago's tonemapping [1], but applied directly to our linear HDR pixel values giving us the following equation:

$$I_{HDR} = (s * I_{HDR_{lin}}{}^{\frac{log(b)}{log(0.5)}})^\gamma \tag{8}$$

Implementation and Results

Using Debevec's method [3], we first loaded the following set of images with the following camera settings: ISO 500, 24 mm, f/1.8. We took 6 images of the same scene using the following exposure times: 1/1 s, 1/5 s, 1/13 s, 1/25 s, 1/60 s, 1/100 s as shown in Figure 2.

Figure 2. Original set of images with the following exposure times: 1/1 s, 1/5s, 1/13 s, 1/25 s, 1/60 s, 1/100 s.

Since we are limited in the amount of compute power and memory we have, we sampled the same 50 evenly spaced pixels from each of our original images. These pixels are the values we used for solving equation 5 The only remaining unknown we have besides $g(Z_{ij})$ and E_i is our smoothing term, λ. To find λ, we solved equation 5 for various different λ values as seen in the plot in Figure 3.

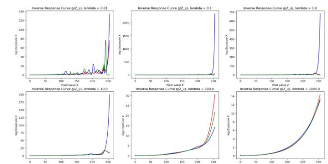

Figure 3. Inverse Camera Response Function, $g(Z_{ij})$ for different smoothing factors, λ

From the plots, we see that with $\lambda = 1000$, we get a smooth response, so this is the camera response curve we will use to recover our linear LDR images. For each of our original images, we can map our pixel values (Z_{ij}) to our linear LDR pixel values to get the following images shown in Figure 4.

Figure 4. Linearized LDR Images

We can use these linearized LDR images to compute our linearized HDR images using equations 4, 5, and 6. With this linearized HDR image, we can now apply some tonemapping algorithms like Drago's tonemapping [1], Mantiuk's tonemapping [4], and our tonemapping algorithm shown in Figure 5.

Figure 5. Tonemapped Images

We also used another set of images to create another HDR image. We used the camera response function we found earlier to map these images to linearized LDR images to create our HDR image. We also applied the same tonemapping as was done on the previous images, but we also chose to modify our previous tonemapping by adding a bias term in our original tonemapping. By adding the bias term we are able to see slight improvements in our resulting HDR image.

We now compare our new tonemapping to Drago's tonemapping, Mantiuk's tonemapping, and finally Apple's HDR image pipeline. These images can be seen in Figure 6.

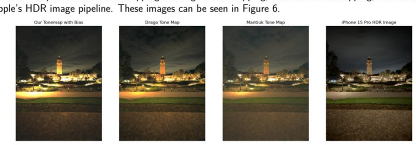

Figure 6. Comparision of Different HDR Image Pipelines

References

[1] T. Annen N. Chiba E. Drago, K. Myszkowski.
Adaptive logarithmic mapping for displaying high contrast scenes.

[2] Whitson Gordon.
How to use hdr mode when taking photos.

[3] Jitendra Malik Paul E. Debevec.
Recovering high dynamic range radiance maps from photographs.

[4] Louis J Kerofsky Rafal Mantiuk, Scott Daly.
Display adaptive tone mapping.

[5] Claude E. Shannon.
A mathematical theory of communication.
Bell System Technical Journal, 27(3):379–423, 1948.