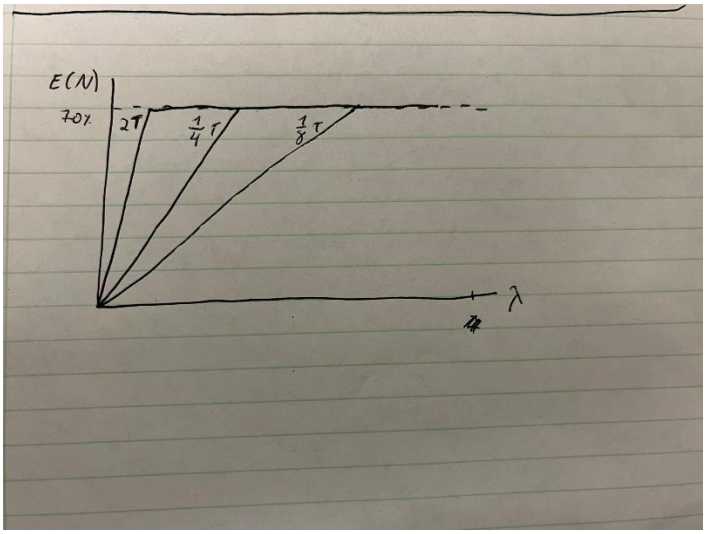
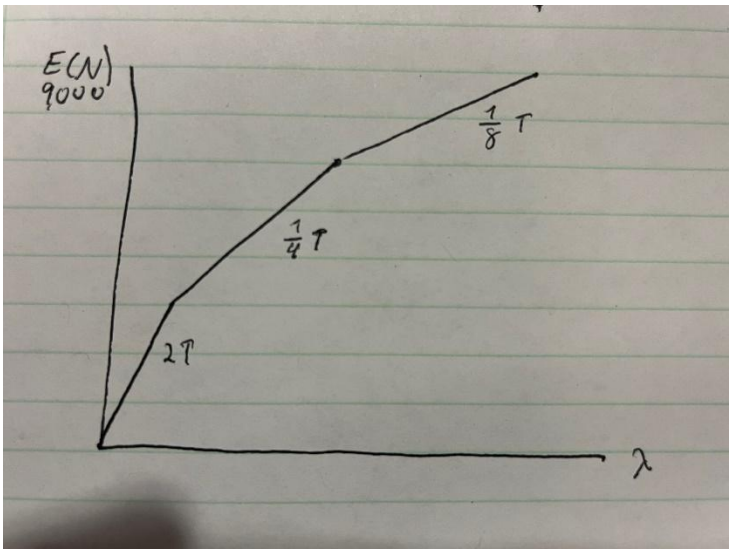
(a) [5 points] Suppose a CCD camera sensor has pixels with full well capacity of 30,000 electrons, and a quantum efficiency of 70%. We capture three images with this sensor using exposure times 1/8, 1/4 and 2 seconds. Sketch the pixel response curves for these three exposure times. Make sure you annotate interesting features about these curves such as their slopes and saturation points.

One interesting feature we can see involves saturation. We see that the longer the exposure time, the steeper the slope. This means that since the exposure time is longer, the saturation limit will be reached later, creating a brighter image.



(b) [5 points] A rudimentary method of exposure bracketing simply adds up the pixel values from each individual low-dynamic-range exposure, effectively increasing the saturation limit. For example, the CCD sensor in part (a) will have an effective saturation limit of 3x30000=90000 electrons when three different exposure times are merged together. Draw the response curve for this scheme.



(c) [10 points] You are given three grayscale PNG images captured with exposure times 1/8, 1/4 and 2 seconds. What do you notice about the dark/bright regions in the shorter/longer exposure time images? Write a script in your programming language of choice to merge them and display the HDR image.

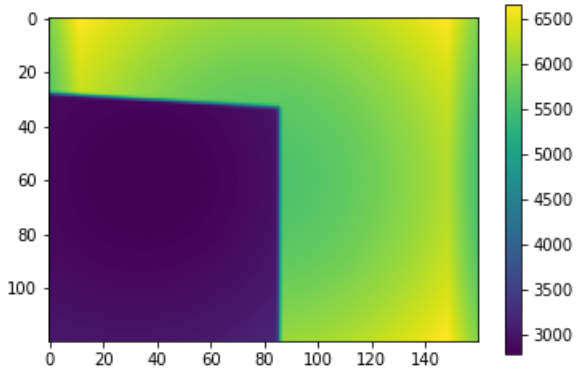
One thing we can see from all 3 images is the level of detail in the dark spots. For example, in the $1/8^{th}$ exposure time, we notice there are more detail in the clouds compared to the 2 second exposure time where we can hardly see the borders of the clouds.

(d) [10 points] Another method for exposure bracketing is the so called "last sample before saturation" method. Starting with a sequence of exposures, each pixel is assigned a value which is closest to saturation but not saturated. For example, if the 8-bit pixel values for three different (increasing) exposure times are {10, 110, 255}, then that pixel is assigned a value of 110 in the final HDR image. Draw the response curve for this scheme. Implement it for the three images in part (c) and display the resulting image. What artifacts do you notice with this method?

Some things I notice about this image are that the image is much darker compared to the last. Also, if looked very closely we'll see that there are certain spots that seem to be missing detail which could come from the pixel averaging methods used to create the image.

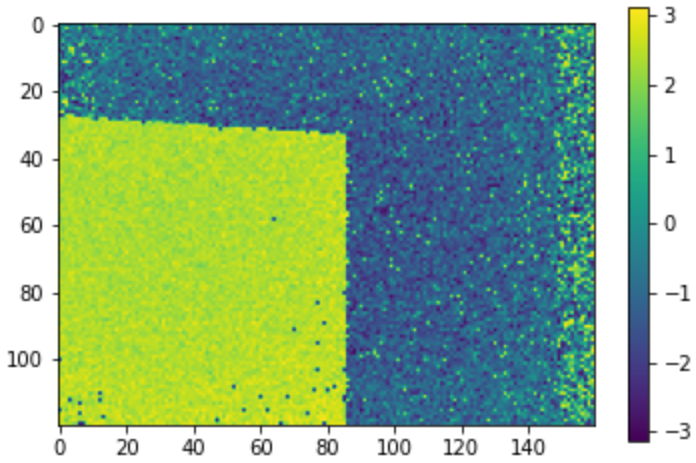
a) [5 points] display the ground-truth depth map of this scene. What is the x-y resolution of the depth image? Does the depth map agree with the appearance of the RGB color image in Fig. 1 above? What is the average distance of the rectangle patch that's closer to the camera? What is the average distance to the wall in the background? (Use appropriate regions-of-interest to compute these averages. Note that these depths are in the unit of millimeters. Show a colorbar with your depth map plots.)

The below image is the mapping of the file. The x-y resolution is about 160x120 and is fairly accurate to the RGB representation of the original image.



And since we know the units of depth are in millimeters, we can assume the average depth of the frontmost surface is about 3,000 millimeters while the back of the surface is about 6,000 millimeters (about twice the distance from the camera).

(b) [10 points] Recall that a CW-ToF camera estimates depth by measuring the *phase shift* of the intensity of light that is transmitted into the scene by a laser. We measured the phase shifts at each pixel in the scene above using a ToF camera with modulation frequency of 20MHz. What is the unambiguous depth range at this modulation frequency? Estimate the depth map and display it in the unit of meters at each pixel.



Although, we were not able to unwrap the depth values, we are able to estimate the interval for the frequency by dividing by out intervals of 2pi to range the depth image to 10pi.