# Assignment 3

Anirudha Ramesh (aramesh3)

October 26, 2022

Please look at the code submitted along-side. There are numerous comments describing what we are doing and answering/elaborating on some answers as well as having the necessary visualizations.

1. Part 1 : Bilateral Filtering
   Various hyper-parameters for Abase.
   The change in sigma s considering kernel size has been limited throughout for faster compute is not very prominent. But with increase in sigma s the patch without edges will get blurrier.
   The difference images in general retain edges very well by taking the edge values close to 0 but other values are not 0. With increase in sigma s the non edge values slightly increase.
   With increase in sigma r, we get higher resolution in detail and have more edges and better edges. This is particularly visible in the threads at the top of the lamp. This is reflected in the difference image as well.

   All difference images are scaled 100x to be more visible (all through this report).

   We pick sigma s to be 40 and sigma r to be 0.05. lambda is 0.01 though this and all following methods.

Figure 1: Variation over sigma s : 20, 40, 75

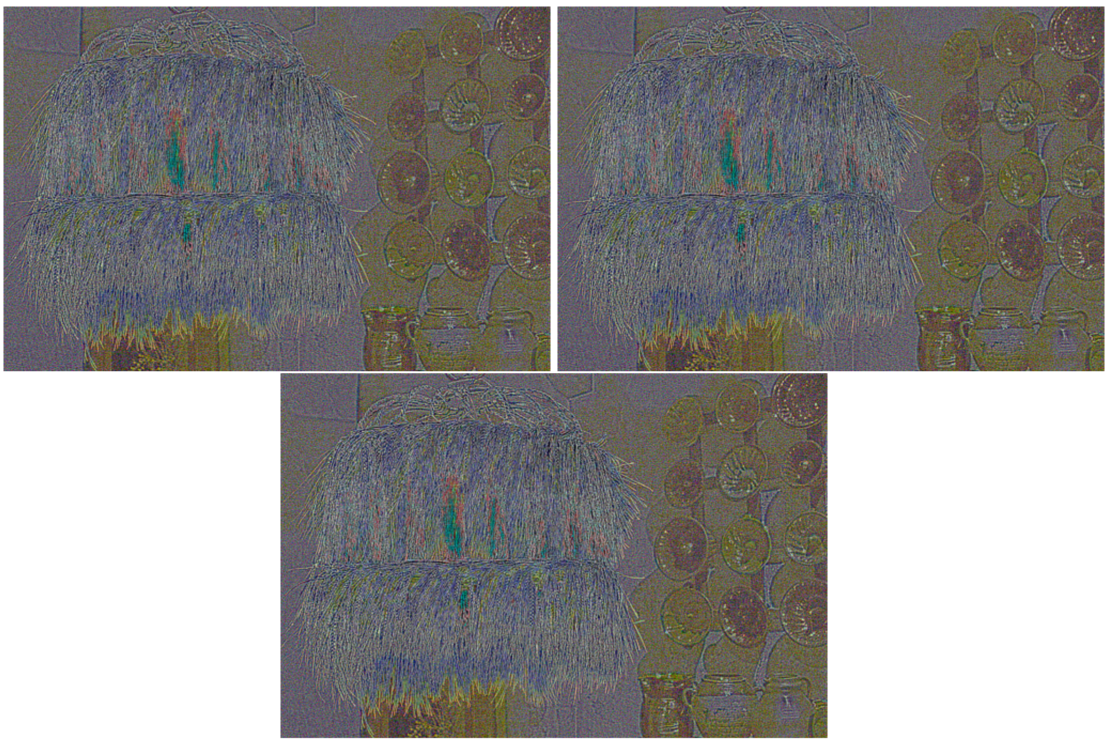


Figure 2: image1 (amb img) -image2 Variation over sigma s : 20, 40, 75

Figure 3: Variation over sigma r : 0.05, 0.1, 0.2

Figure 4: image1 (amb img) -image2 Variation over sigma r : 0.05, 0.1, 0.2

Various hyper-parameters for ANr.
The variation of sigma s and sigma r follow similar trends here but we pick a higher sigma s with lower kernel size. sigma r should be a lot lesser according to the orignal paper, but this creates weird artifacting.
Compared to Abase we can preserve edges better while more aggressively blurring the noisy regions without loss in detail. This is apparent in the background, and more dimly lit portions of background and objects in the background.
We pick sigma s to be 100, and sigma r to be 0.05.

Figure 5: Variation over sigma s : 50, 100, 150

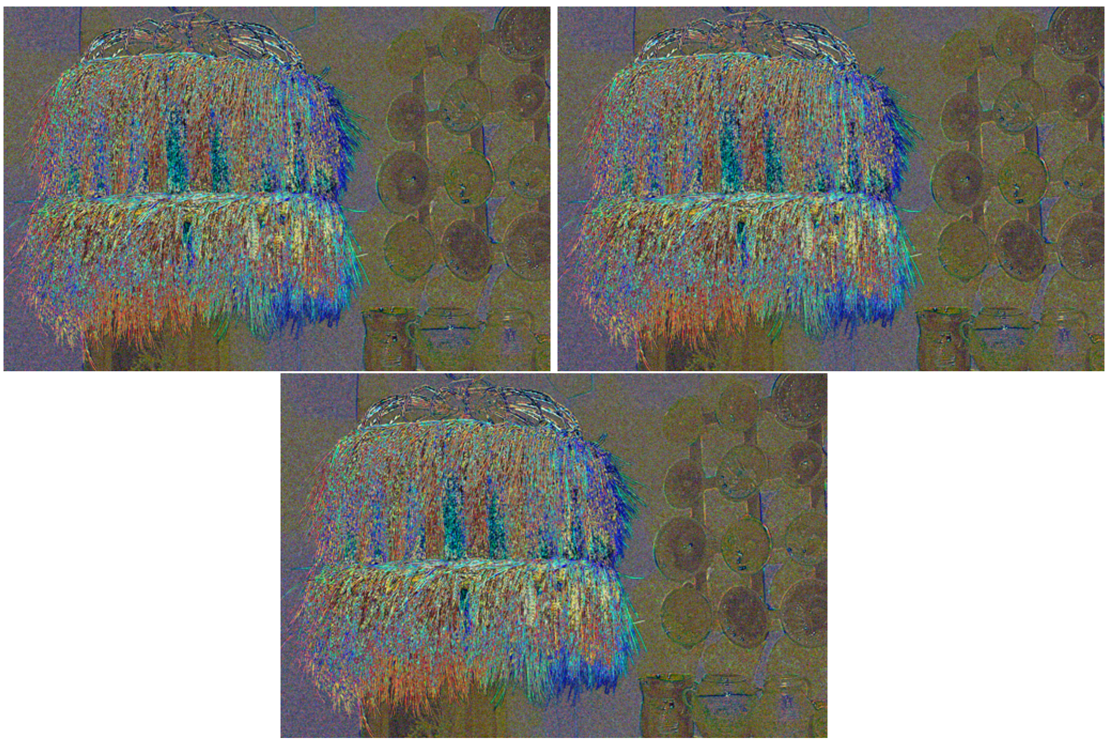


Figure 6: image1 (amb img) -image2 Variation over sigma s : 50, 100, 150

Figure 7: Variation over sigma r : 0.05, 0.1, 0.2

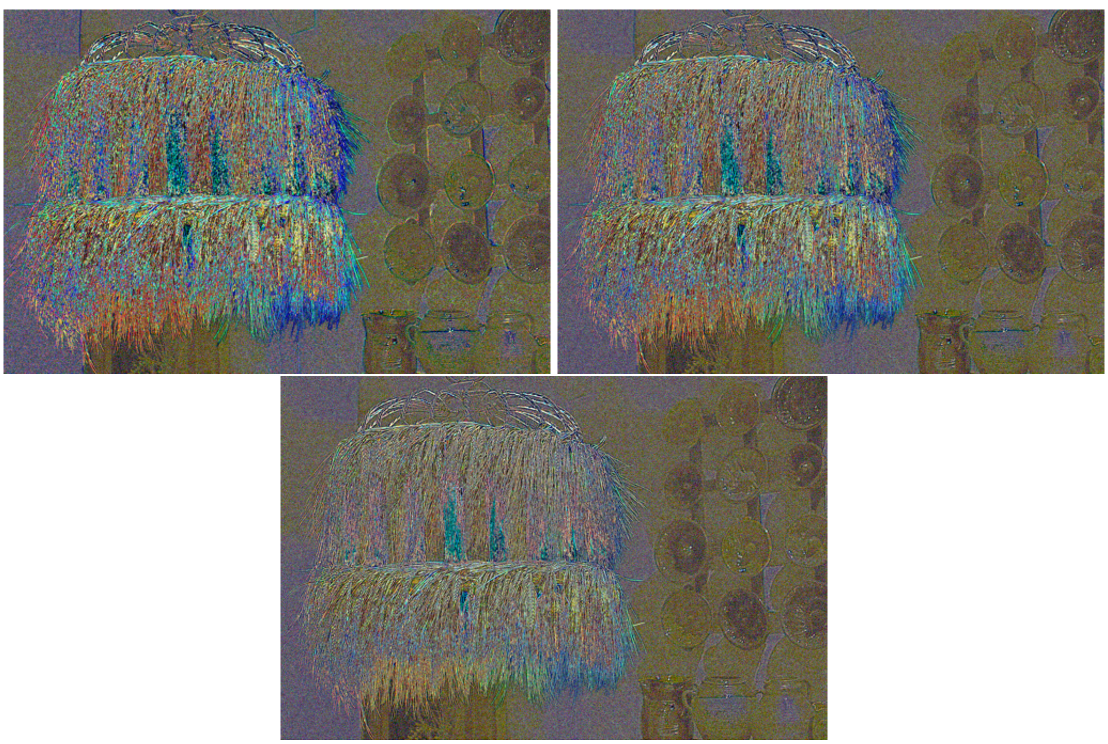


Figure 8: image1 (amb img) -image2 Variation over sigma r : 0.05, 0.1, 0.2

Various hyper-parameters for ADetail.
The variation of sigma s and sigma r follow similar trends here but we pick the same sigma s with same kernel size as Abase for Fbase. We tuned this for hyper-params across levels (combinatorial explosion here we are) but it is not reasonable to have all of them in the report. Changing eps however does make a difference as we would expect. We show some of those changes below to understand the maximum eps we can have (more robust to small irregularities) without losing detail.
With increase in eps our detail transfer is less successful and our blurry patches from ANr continue to remain so.
We pick eps to be 0.001.

Figure 9: Variation over eps : 0.001, 0.02, 0.1, 0.5

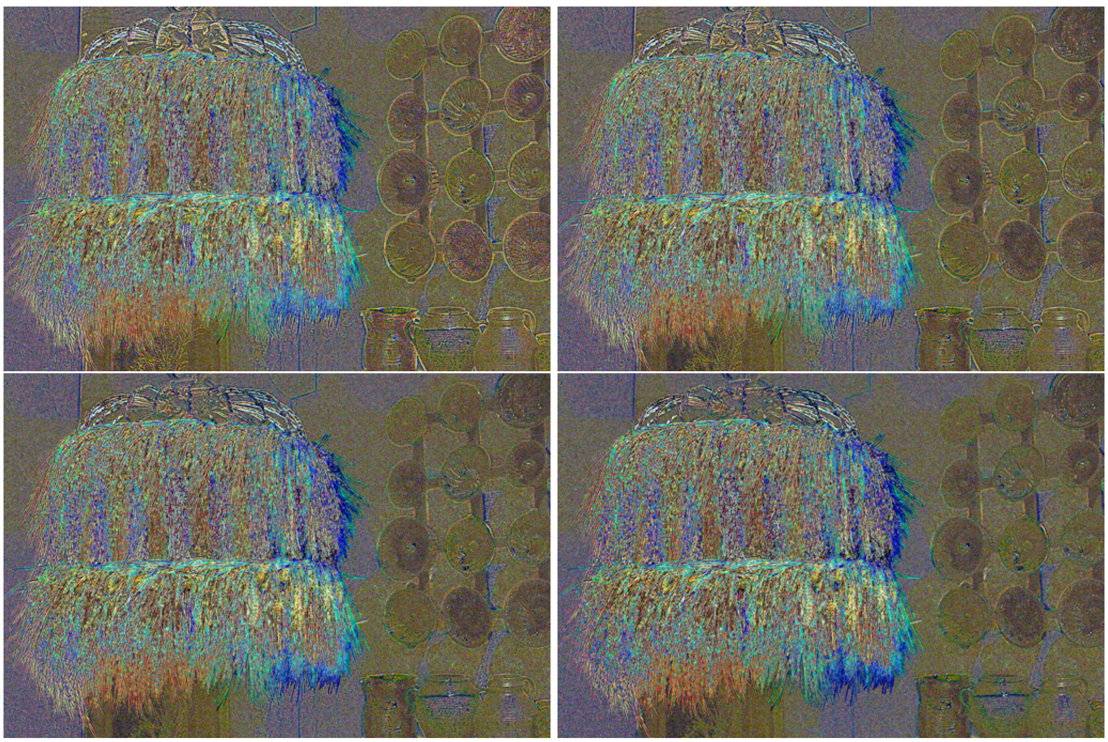


Figure 10: img1(amb) - img2 Variation over eps : 0.001, 0.02, 0.1, 0.5

Various hyper-parameters for AFinal (AMasked).
We try various combinations of morphological operations to get our best results. We conclude that the best way to do this is to have an opening (remove noise), followed by a closing (fill holes), followed by a further dilation (expand the speckles), and finally a gaussian blur to make mask smooth.
We demonstrate the change of some of the hyperparams involved in this below.
We choose shadow_threshold = 0.0005, speckle_thershold = 0.9.
opening_kernel = np.ones((3,3),np.uint8) clears noise
closing_kernel = np.ones((8,8),np.uint8) fills holes
dilation_kernel = np.ones((20,20),np.uint8) dilation
For final gaussian smoothing we use a kernal of size 21 with sigma=75.

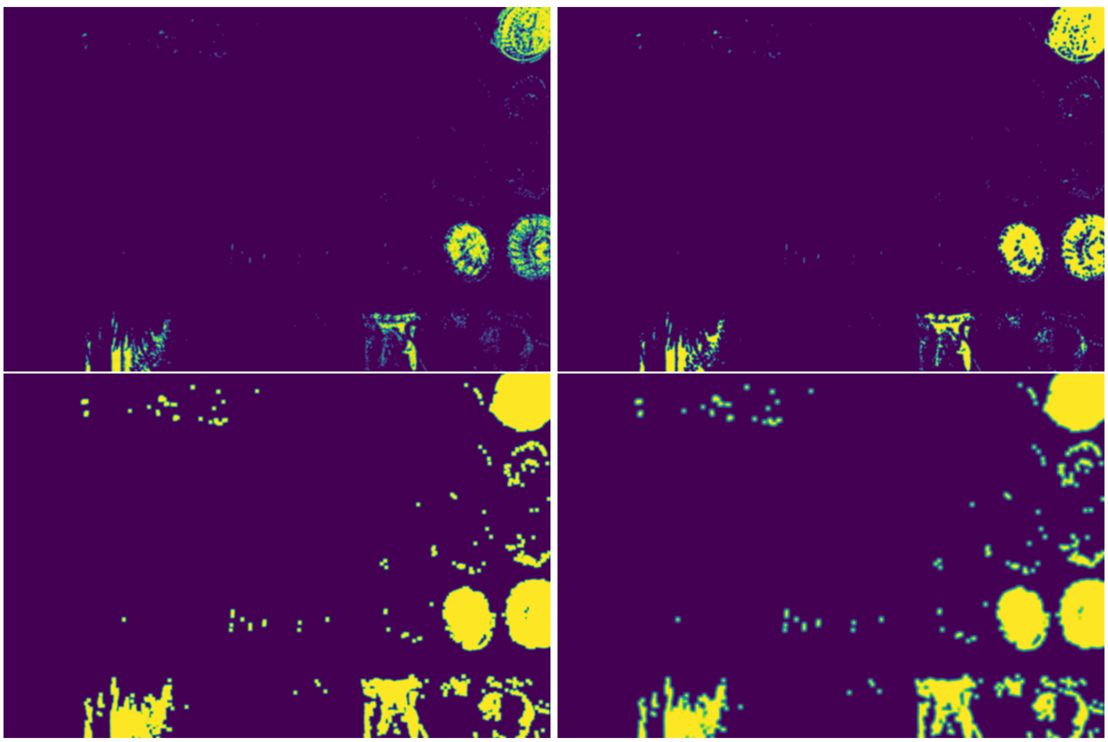


Figure 11: First has just opening, second opening-closing, third opening-closing-dilation, larger final gaussian blur. First three have smaller gaussian blur to show the difference.

Figure 12: First has just opening, second opening-closing, third opening-closing-dilation, larger final gaussian blur. First three have smaller gaussian blur to show the difference.

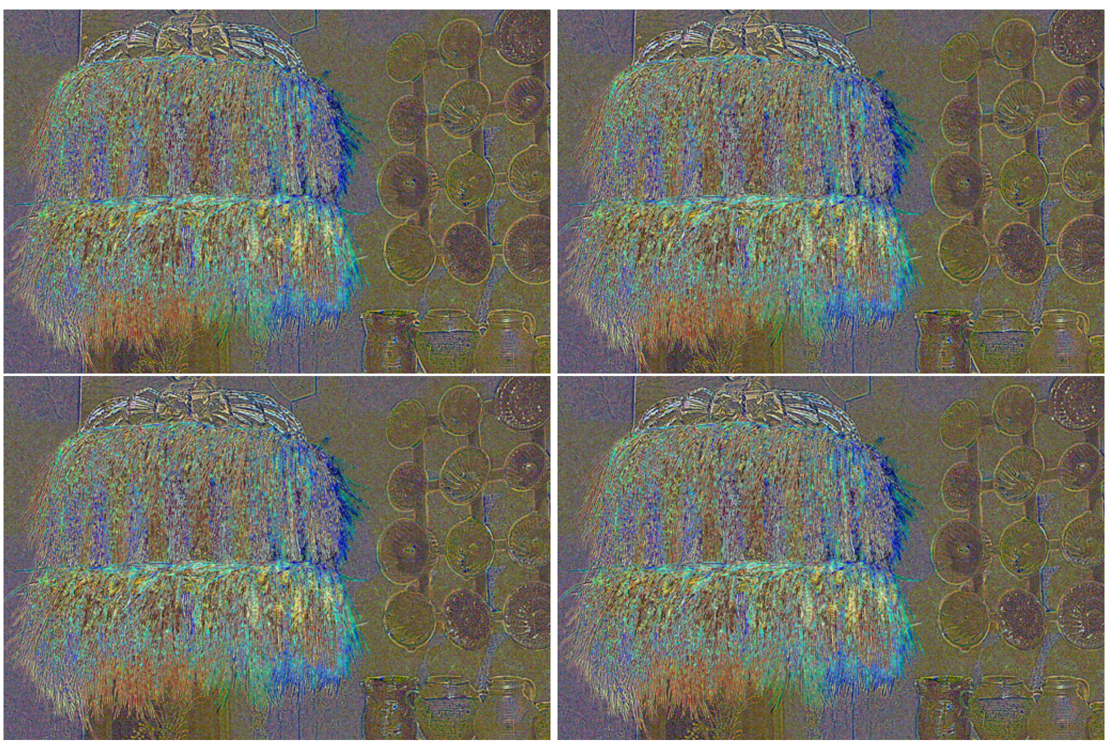


Figure 13: First has just opening, second opening-closing, third opening-closing-dilation, larger final gaussian blur. First three have smaller gaussian blur to show the difference.

Final picks are below,

Figure 14: These are in order Abase, ANr

We see that Abase by itself has made the image far less noisy in the background as compared to the unfiltered image.
ANr clears the background noise very well, but affects some of the detail in the other region as well, for example, it blurs out some of the edges in the lamp.
Adetail adds this detail back, and this image looks better than Abase both in terms of noise reduction in the darker background, as well as detail preservation. We do still have some shadows and specularities to deal with which have been introduced by the flash image's edges.
AFinal (AMask) attempts to do so, but it makes very little difference overall visually though we do see some blurring along the flash-origin shadows and edges (shadows and speckles near some of the bowls, and some shadows around the lamp).
In the end, though AFinal is the best, Adetail is close enough to be considered good enough.

Figure 15: These are in order Adetail, AFinal(AMask)

2. Gradient Domain Processing

   We show the effects of sigma and tau variation below.
   We select sigma to be 40 and tau to be 0.7.

Figure 16: sigma 0.001, 10, 40, 200. With increase in sigma image seems to get brighter in general, but the difference between 10 and 200 isn't visible.

Figure 17: tau 0.5, 0.7, 0.9. With increase in tau image seems to get brighter, but more faded.

Gradient plots are below.

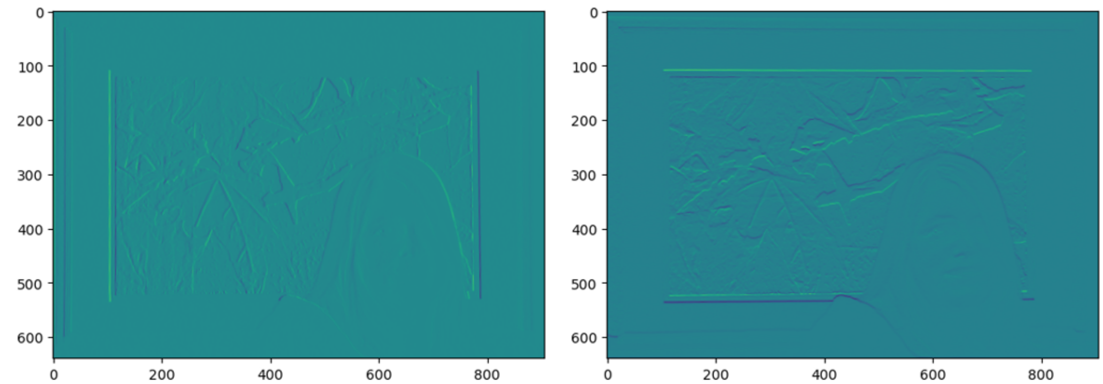



Figure 18: a grad x, and a grad y

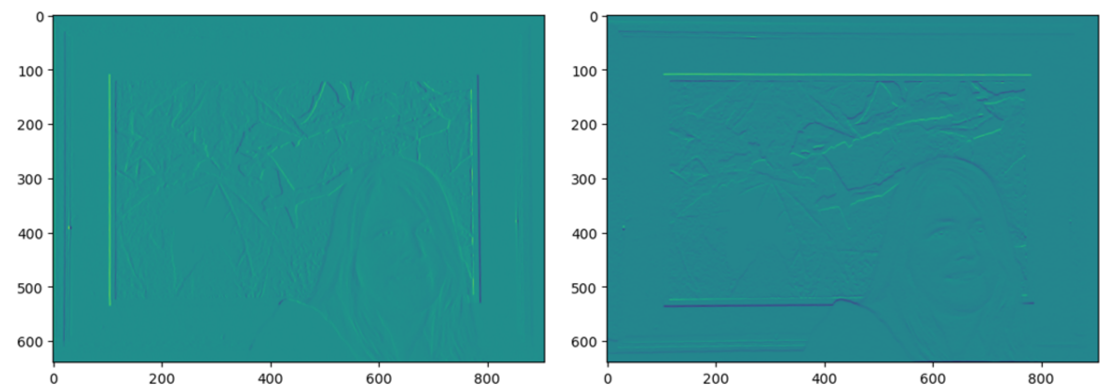



Figure 19: phi dash grad x, and phi dash grad y

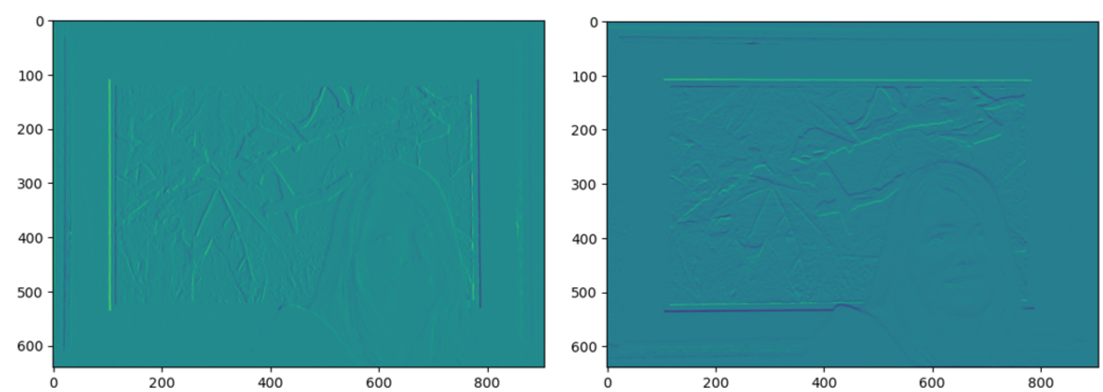



Figure 20: phi star grad x, and phi star grad y

Boundary conditions experiments.
We experiment along both size and origin of boundary.
We can see the experiments below.

Boundary with ambient image produces a very dark image, and as we go towards flash, our image becomes brighter. We therefore choose the flash image for boundary here.
We experiment with larger boundary sizes, but find that this doesn't seem to particularly improve the image. If anything, it creates a flash style bright streak in the bottom left parts of the frame of the painting.

The final image we pick has sigma 40, tau 0.7, flash initialization with boundary size 1. This is the first image in both the figures below. We can see that the face is overall considerably brighter than the ambient image, and details have been transferred correctly from the flash image without fundamentally altering the color pallette of the image or the creation of other major flash artifacts. We do have a slight highlight streaking that changing just these hyper-parameters does not seem to remove.

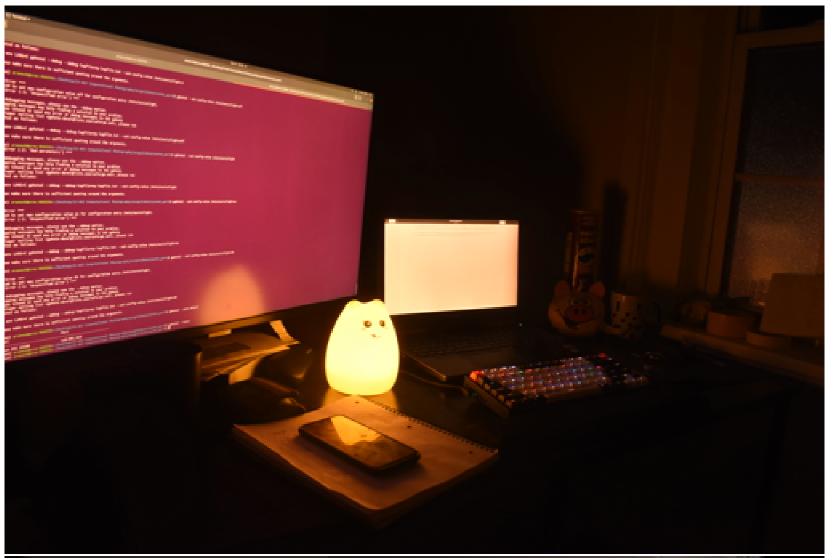
Figure 21: Boundary origin is ambient image, flash image, amb-flash mean image

Figure 22: Boundary size is 1, 25

3. Custom Flash/No-Flash pairs

   Bilateral Filtering.

   We select the following flash-no flash pair for our bilateral filtering.
   Our bilateral filtering works exceptionally well. This is more visible in the full resolution submission. Our edges are preserved, and enhanced in comparison to the ambient image, and we reduce dark noise considerably as well.



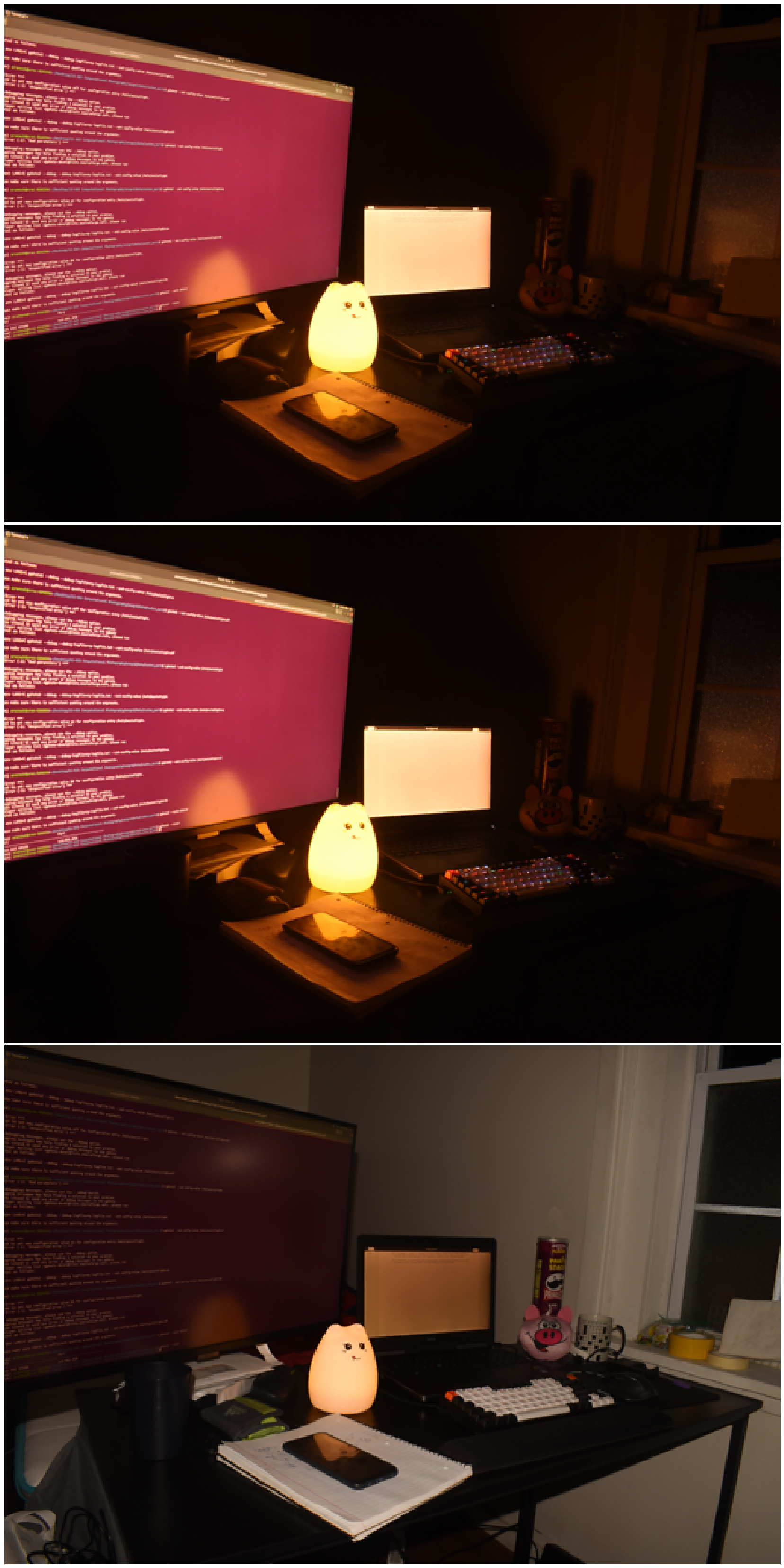


Figure 23: Abase, ANr, Fbase

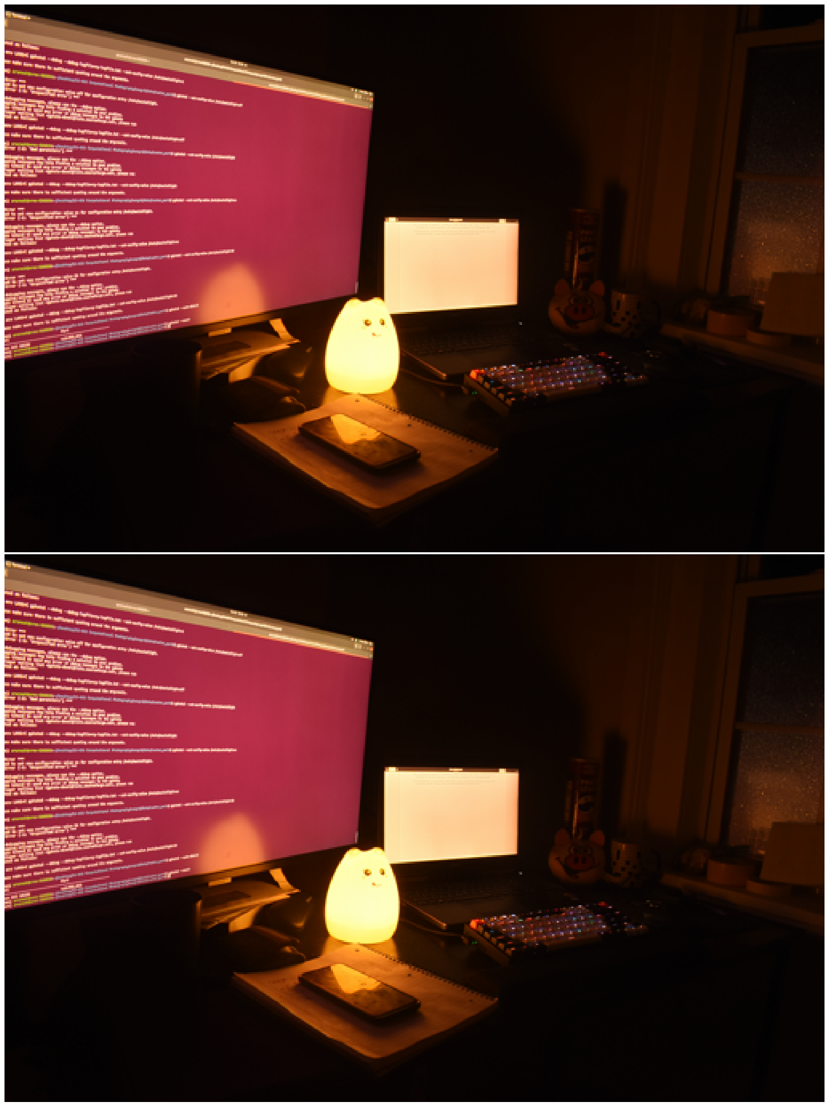


Figure 24: Adetail and Afinal

Hot-Spot removal
We select the following flash-no flash pair for our flash removal.

We enhance this image by giving greater detail in some of the dimly lit areas like behind the monitor (wall and red bedspread), and dimly lit objects on the desk, while simultaneously completely removing the flash on the screen. We would also like to get the wires of the earphones towards the bottom left, but this area remains slightly dim (still brighter than before).

Figure 25: Final output

4. Bonus : Reflection Removal in Flash/No-Flash photography

While the code has been written, the scenarios we managed to capture could not exactly fulfill the role required. (Near window, from a higher height because of apartment location, tripod height, lack of flash control through gphoto2 etc)
Regardless we submit those results below, but hope the TAs can look through the code for credit.
Instead we demonstrate our code on the images from the original paper.
We use sigma and tau with values of 0.1 and we set the dierlicht boundary conditions on the H image. To accommodate crop alignment issues, we set a larger boundary. In creation of H we also use a slightly larger percentage of the flash image over the ambient image.
Our generated image has the color from the flash image without the reflection, but at the same time there is a slight halo-ing blur.

Figure 26: We had trouble creating a flash reflection.


Figure 27: ambient, flash, and merged

5. Bonus : GradientShop

   We've implemented and verified most of the subparts involved in gradientshop. The applications implemented are saliency sharpening and stylize filter.
   It is easy to extend this to all other applications as well since the code is structured in a easy to extend way.
   One key deviation from this relates to edge length calculation which we do in binary based on canny edge filter, which we use to find continuous edges. If we find any continuous edge, we assign its length to be max, all else are 0. Despite all other subparts seeming to work, our results shift very little from our input, despite gradient descent convergence. We believe this may be due to the variation in length implemented. The 2-point numerical approximation of derivative may also be a reason.
   Please refer to the jupyter notebook printout corresponding to part 5 to see the subcomponent code and tests.