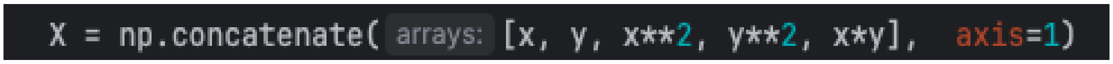
Neta Nakdimon
322216128
BS.c Computer Science

# Vignetting – Ex4

1. get_index_matrix_function:

   Constructs a matrix $X$, which is essential for generating the calibration map used to correct vignetting in images. It starts by creating grid arrays for $x$ and $y$ coordinates of each pixel using **np.meshgrid()** and reshapes them into column vectors. These coordinates, along with their squared $(x^2, y^2)$ and cross-product $(xy)$ terms, are combined using **np.concatenate().** This approach captures the radial characteristics of vignetting, enabling a more precise modeling of the brightness variations across the image.



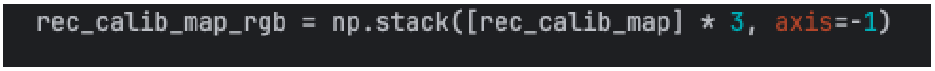
```
X = np.concatenate( arrays: [x, y, x**2, y**2, x*y],  axis=1)
```

2. get_calib_coeffs fuction:

   Calculates the calibration coefficients $(b)$ for a given calibration image(calib_map) These coefficients provide a compact way to represent the calibration map. The function starts by reshaping the calibration map into a column vector $(y)$.
   Then it builds the feature matrix $X$ using get_index_matrix() function. Finally it applies the least squares method(**np.linalg.lstsq**) to solve $b$, which are the coefficients that minimize the difference between the modeled and actual calibration map.

3. fix_raw_im:

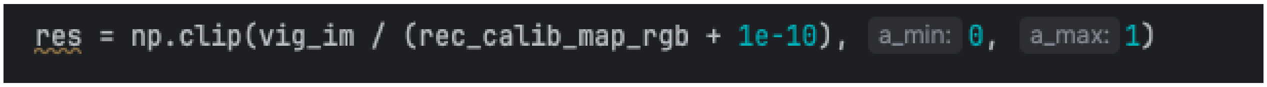
   Designed to correct the vignetting effect in an image(vig_im) using the calibration coefficients $(b)$ computed during the calibration step. The process begins by normalizing the input image to ensure the pixel values are in the range [0,1]. Then the function uses the get_index_matrix() to construct the feature matrix $X$. The calibration map is reconstructed using the formula $X@b$. This reconstructed map is reshaped into 2D array(rec_calib_map) to align with the image's pixel layout. Since the vignetting correction applies equally to all color channels, the 2D calibration map is replicated



```
rec_calib_map_rgb = np.stack([rec_calib_map] * 3, axis=-1)
```

across the 3 RGB channels using **np.stack**.

Finally, the input image is corrected by performing element-wise division of the normalized image by the replicated calibration map(rec_calib_map_rgb) with an addition of a small number(1e+10) to avoid dividing by zero. The **np.clip** function is used on the division to ensure that the corrected image (res) has pixel values constrained within the valid range [0, 1], preventing any out-of-bound values that may result from the division operation. The function returns the corrected image (res) and the 2D calibration map(rec_calib_map).

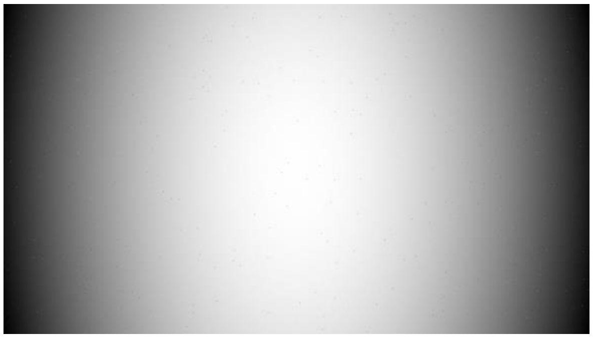


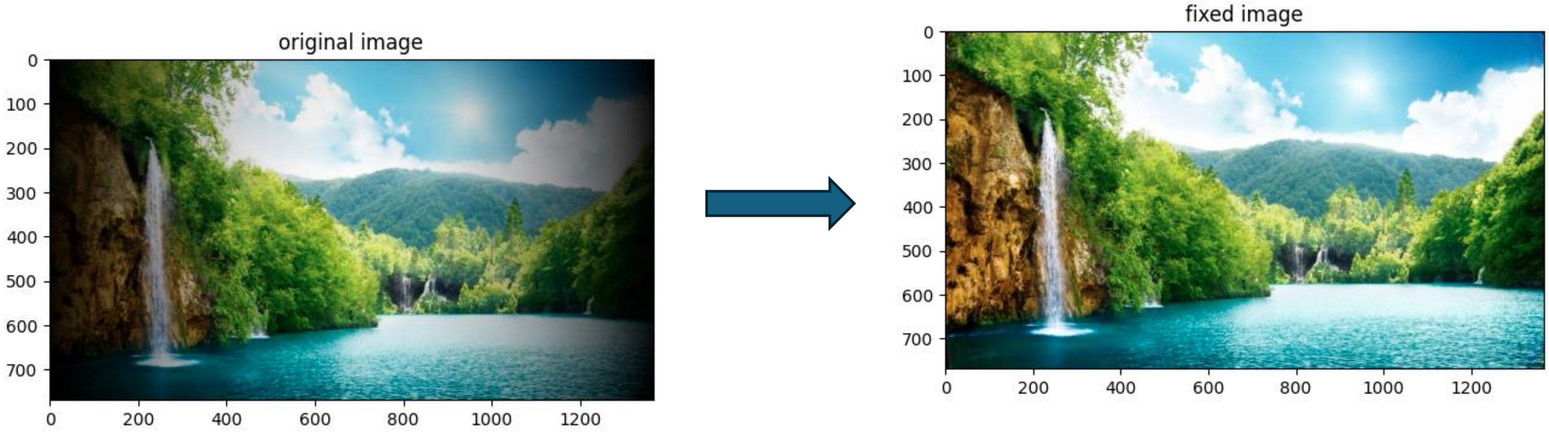
```
res = np.clip(vig_im / (rec_calib_map_rgb + 1e-10), a_min: 0, a_max: 1)
```

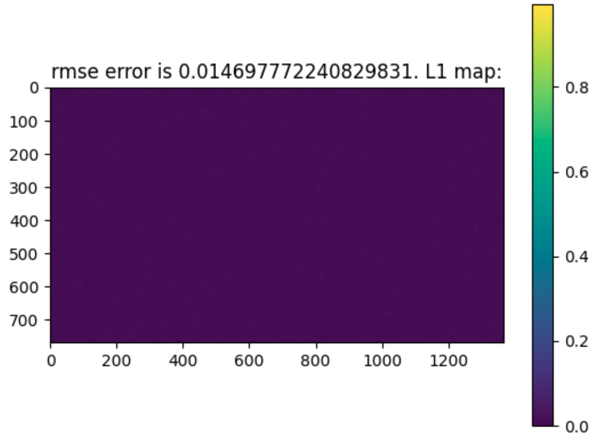
## Results:

The results attached below demonstrate successful correction of the vignetting effect. The fixed images show an even brightness distribution, eliminating the darkened areas near the edges. The overall brightness is now well-balanced across the pixels. Furthermore, the RMSE values achieved are consistently low, staying below 0.0148, indicating accurate reconstruction of the calibration map. The L1 error maps also show minimal differences, with values close to 0.0 for all images, highlighting the effectiveness of the solution. This implementation provides a reliable method for addressing the vignetting problem, offering a practical solution for the camera startup to automatically enhance image quality.

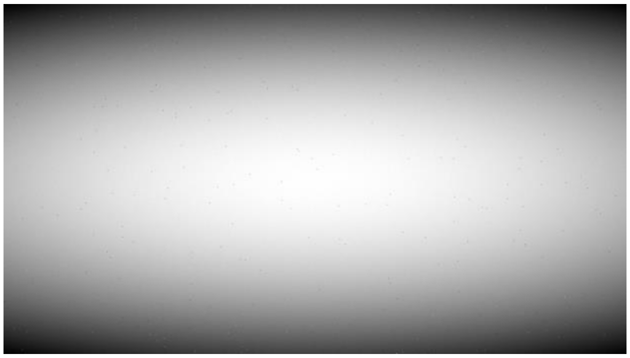
Calibration image – camera 1

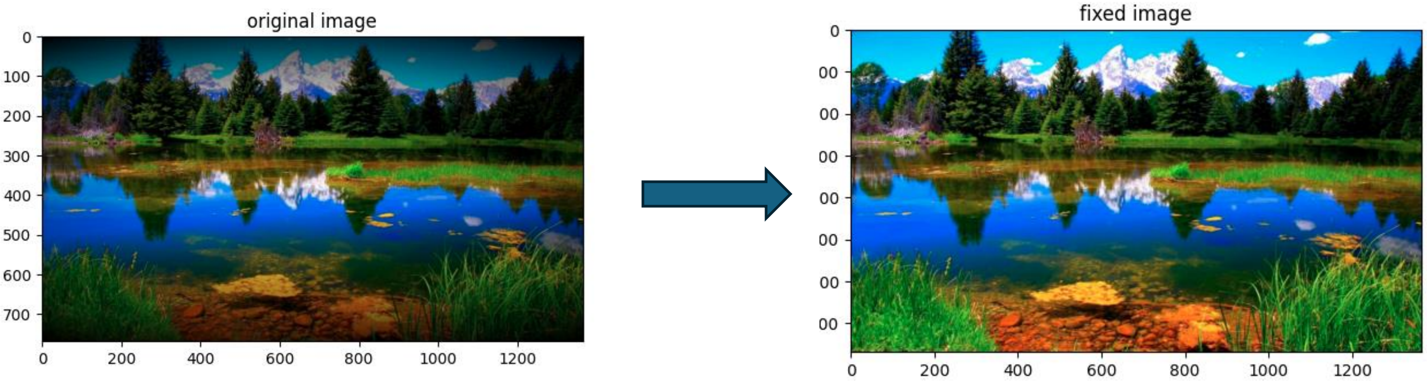


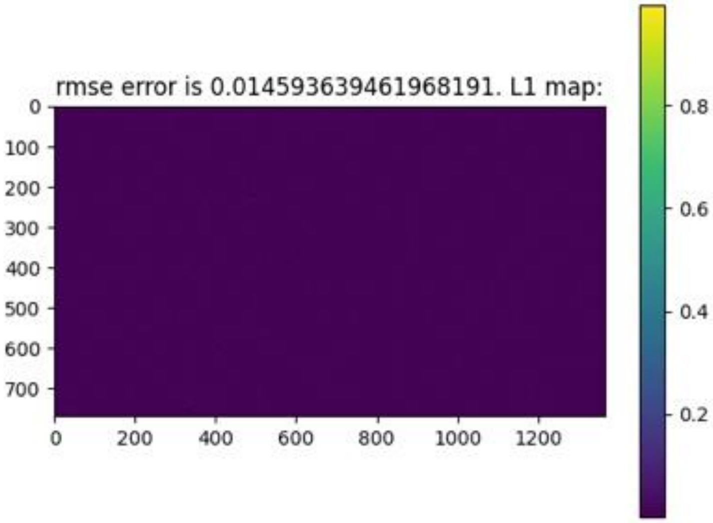



rmse error is 0.014697772240829831. L1 map:

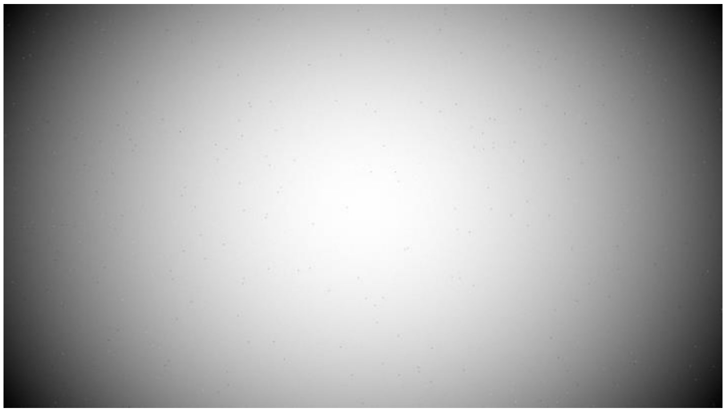
Calibration image – camera 2

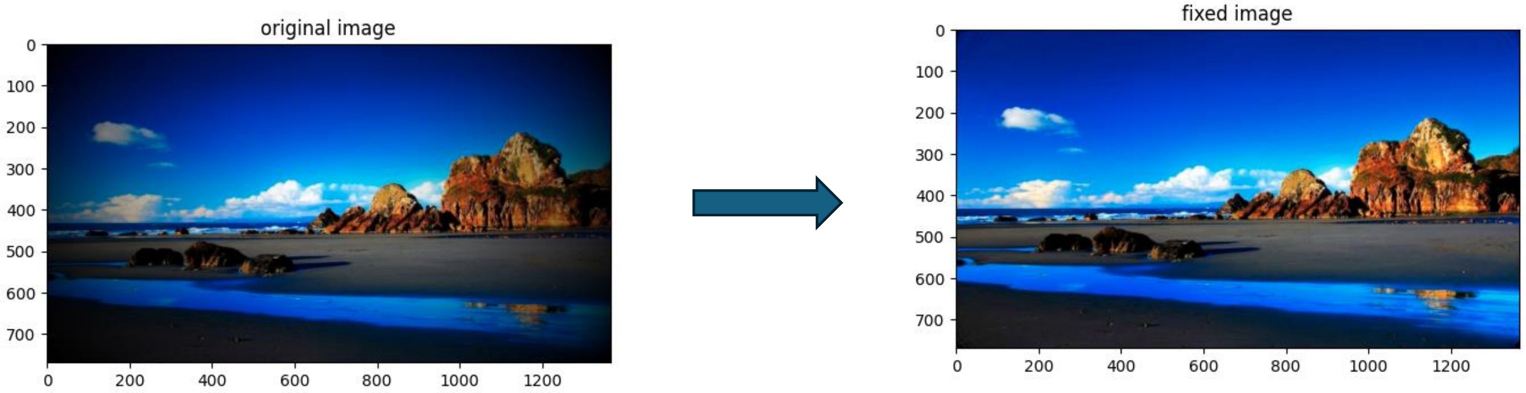


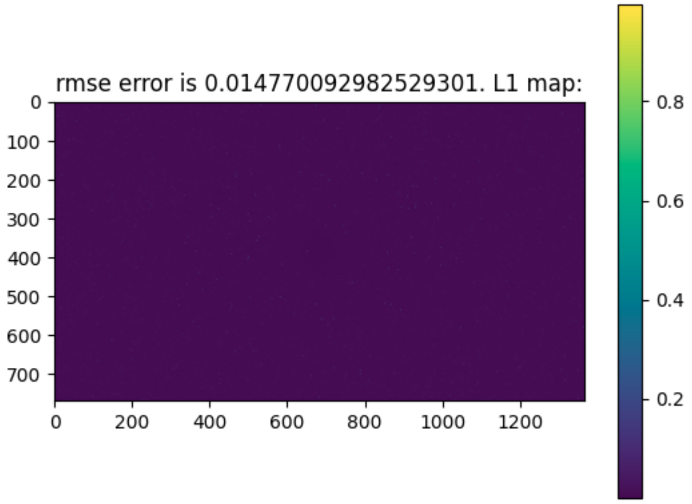

original image
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
fixed image
0
00
00
00
00
00
00
00
0
200
400
600
800
1000
1200


rmse error is 0.014593639461968191. L1 map:
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
0.8
0.6
0.4
0.2

Calibration image – camera 3




original image
fixed image


rmse error is 0.014770092982529301. L1 map: