

```

1
2 import os
3 import image_util
4 from matplotlib import pyplot as plt
5 import numpy as np
6
7 BASE_DIRECTORY = './images'
8
9
10
11 if __name__ == '__main__':
12
13     image = image_util.read( os.path.join(
14         BASE_DIRECTORY, 'fluo_shading.png'))
15     image_util.show(image) #show image with
16     background
17     M = image.shape[0]
18     N = image.shape[1]
19
20     import numpy as np
21     # Coefficients of the equations
22     coefficients = np.array([
23         [1, 26, 18],
24         [1, 136, 236],
25         [1, 493, 496],
26     ])
27
28     # Constants on the right-hand side of the
29     equations
30     constants = np.array([15, 52, 126])
31
32     # Solve the system of equations
33     solution = np.linalg.solve(coefficients,
34         constants)
35     variables = (np.floor(solution)).astype(int) #
36     solution.astype(int) #np.round(solution, decimals=0)
37
38     print("Solution for coefficients a0, a1, a2:")
39     print(solution)

```

```

37     print(variables)
38
39     background_image = []
40
41     for x in range(0, M):
42         rows = []
43         for y in range(0, N):
44             background_intensity = solution[0] + (
45 solution[1]*x) + (solution[2]*y)
46             rows.append(background_intensity)
47             image[x,y] = abs(image[x,y] -
48 background_intensity)
49             background_image.append(rows)
50
51     image_util.show(background_image)
52
53     image_util.show(image) #image without background
54
55     # min intensity in first square
56     square_1 = [] # contains all intensities in that
57     square
58     for x in range (0, int(M/2)):
59         for y in range (0, int(N/2)):
60             square_1.append([image[x,y], (x,y)])
61
62     print(min(square_1))
63
64     square_2 = [] # contains all intensities in that
65     square
66     for x in range(0, int(M / 2)):
67         for y in range(int(N / 2), N):
68             square_2.append([image[x, y],(x,y)])
69
70     print(min(square_2))
71
72     square_3 = [] # contains all intensities in that
73     square
74     for x in range(int(M / 2), M):
75         for y in range(0, int(N / 2)):
76             square_3.append([image[x, y], (x, y)])

```

```
73
74     print(min(square_3))
75
76     square_4 = [] # contains all intensities in
that square
77     for x in range(int(M / 2), M):
78         for y in range(int(N / 2), N):
79             square_4.append([image[x, y], (x, y)])
80
81     print(min(square_4))
82
83
84
85
86
87
```