

```
1: """
2: Image Processing and The Scanning Tunneling Microscope
3: """
4:
5: from math import *
6: import pylab as pl
7: import numpy as np
8:
9:
10: def I(partial_x, partial_y, phi):
11:     return -(np.cos(phi) * partial_x + np.sin(phi) * partial_y) / np.sqrt(
12:         pow(partial_x, 2) + pow(partial_y, 2) + 1)
13:
14:
15: def main():
16:     altitudes = pl.loadtxt('altitude.txt')
17:     # print(altitudes.shape)
18:     partial_x = np.zeros(altitudes.shape)
19:     partial_y = np.zeros(altitudes.shape)
20:     size_y, size_x = altitudes.shape
21:     for y in range(altitudes.shape[0] - 1):
22:         for x in range(altitudes.shape[1] - 1):
23:             partial_x[y][x] = (altitudes[y][x + 1] - altitudes[y][x]) / 30000
24:             partial_y[y][x] = (altitudes[y + 1][x] - altitudes[y][x]) / 30000
25:     for x in range(altitudes.shape[1] - 1):
26:         partial_x[size_y - 1][x] = (
27:             altitudes[size_y - 1][x - 1] - altitudes[size_y - 1][x]) / 30000
28:         partial_y[size_y - 1][x] = (
29:             altitudes[size_y - 1][x] - altitudes[size_y - 2][x]) / 30000
30:     for y in range(altitudes.shape[0] - 1):
31:         partial_x[y][size_x - 1] = (
32:             altitudes[y][size_x - 1] - altitudes[y][size_x - 2]) / 30000
33:         partial_y[y][size_x - 1] = (
34:             altitudes[y + 1][size_x - 1] - altitudes[y][size_x - 1]) / 30000
35:     intensity = np.zeros(altitudes.shape)
36:     for y in range(size_y):
37:         for x in range(size_x):
38:             intensity[y][x] = I(partial_x[y][x], partial_y[y][x], pi / 4)
39:     pl.set_cmap('Greys')
40:     pl.imshow(intensity)
41:     pl.show()
42:     altitudes = pl.loadtxt('stm.txt')
43:     partial_x = np.zeros(altitudes.shape)
44:     partial_y = np.zeros(altitudes.shape)
45:     size_y, size_x = altitudes.shape
46:     for y in range(altitudes.shape[0] - 1):
47:         for x in range(altitudes.shape[1] - 1):
48:             partial_x[y][x] = (altitudes[y][x + 1] - altitudes[y][x]) / 2.50
49:             partial_y[y][x] = (altitudes[y + 1][x] - altitudes[y][x]) / 2.50
50:     for x in range(altitudes.shape[1] - 1):
51:         partial_x[size_y - 1][x] = (
52:             altitudes[size_y - 1][x - 1] - altitudes[size_y - 1][x]) / 2.50
53:         partial_y[size_y - 1][x] = (
54:             altitudes[size_y - 1][x] - altitudes[size_y - 2][x]) / 2.50
55:     for y in range(altitudes.shape[0] - 1):
56:         partial_x[y][size_x - 1] = (
57:             altitudes[y][size_x - 1] - altitudes[y][size_x - 2]) / 2.50
58:         partial_y[y][size_x - 1] = (
59:             altitudes[y + 1][size_x - 1] - altitudes[y][size_x - 1]) / 2.50
60:     intensity = np.zeros(altitudes.shape)
61:     for y in range(size_y):
62:         for x in range(size_x):
63:             intensity[y][x] = I(partial_x[y][x], partial_y[y][x], pi / 4)
```

```
64:     pl.imshow(intensity)
65:     pl.show()
66:
67:
68: if __name__ == "__main__":
69:     main()
```