```python
"""
Image Processing and The Scanning Tunneling Microscope
"""

from math import *
import pylab as pl
import numpy as np


def I(partial_x, partial_y, phi):
    return -(np.cos(phi) * partial_x + np.sin(phi) * partial_y) / np.sqrt(
        pow(partial_x, 2) + pow(partial_y, 2) + 1)


def main():
    altitudes = pl.loadtxt(’altitude.txt’)
    # print(altitudes.shape)
    partial_x = np.zeros(altitudes.shape)
    partial_y = np.zeros(altitudes.shape)
    size_y, size_x = altitudes.shape
    for y in range(altitudes.shape[0] - 1):
        for x in range(altitudes.shape[1] - 1):
            partial_x[y][x] = (altitudes[y][x + 1] - altitudes[y][x]) / 30000
            partial_y[y][x] = (altitudes[y + 1][x] - altitudes[y][x]) / 30000
    for x in range(altitudes.shape[1] - 1):
        partial_x[size_y - 1][x] = (
            altitudes[size_y - 1][x - 1] - altitudes[size_y - 1][x]) / 30000
        partial_y[size_y - 1][x] = (
            altitudes[size_y - 1][x] - altitudes[size_y - 2][x]) / 30000
    for y in range(altitudes.shape[0] - 1):
        partial_x[y][size_x - 1] = (
            altitudes[y][size_x - 1] - altitudes[y][size_x - 2]) / 30000
        partial_y[y][size_x - 1] = (
            altitudes[y + 1][size_x - 1] - altitudes[y][size_x - 1]) / 30000
    intensity = np.zeros(altitudes.shape)
    for y in range(size_y):
        for x in range(size_x):
            intensity[y][x] = I(partial_x[y][x], partial_y[y][x], pi / 4)
    pl.set_cmap(’Greys’)
    pl.imshow(intensity)
    pl.show()
    altitudes = pl.loadtxt(’stm.txt’)
    partial_x = np.zeros(altitudes.shape)
    partial_y = np.zeros(altitudes.shape)
    size_y, size_x = altitudes.shape
    for y in range(altitudes.shape[0] - 1):
        for x in range(altitudes.shape[1] - 1):
            partial_x[y][x] = (altitudes[y][x + 1] - altitudes[y][x]) / 2.50
            partial_y[y][x] = (altitudes[y + 1][x] - altitudes[y][x]) / 2.50
    for x in range(altitudes.shape[1] - 1):
        partial_x[size_y - 1][x] = (
            altitudes[size_y - 1][x - 1] - altitudes[size_y - 1][x]) / 2.50
        partial_y[size_y - 1][x] = (
            altitudes[size_y - 1][x] - altitudes[size_y - 2][x]) / 2.50
    for y in range(altitudes.shape[0] - 1):
        partial_x[y][size_x - 1] = (
            altitudes[y][size_x - 1] - altitudes[y][size_x - 2]) / 2.50
        partial_y[y][size_x - 1] = (
            altitudes[y + 1][size_x - 1] - altitudes[y][size_x - 1]) / 2.50
    intensity = np.zeros(altitudes.shape)
    for y in range(size_y):
        for x in range(size_x):
            intensity[y][x] = I(partial_x[y][x], partial_y[y][x], pi / 4)
    pl.imshow(intensity)
    pl.show()


if __name__ == "__main__":
    main()
```