

```
import numpy as np
from math import cos, sin, pi
'''
```

Samuel Haberkorn
CS2300 - Computational Linear Algebra
Project 4
12/04/2020

Questions:

1)

Yes, Every point can be drawn back to another point located on the image plane

2)

No. Because projection matrices are not full rank.

'''

```
# Test data from assignment handout
MATRICES = [[10, 10, 10], [-10, 10, 10], [10, -10, 10], [-10, -10, 10]]
FOCAL_LENGTHS = [1, 5]
```

```
def main():
```

```
    # iterate over test data and calculate the image
    for matrix in MATRICES:
        for focal_length in FOCAL_LENGTHS:
            print(f"The image projection of point {matrix} and focal length {focal_length} is "
                  f"{calculate_image(matrix, focal_length)}")
        # add a case for rotating the matrix with f=2
        print(f"The image projection of point {matrix} rotated pi/4 radians and focal length 2 is "
              f"{calculate_image(matrix, 2, pi / 4)}")
    print('\n')
```

```
def calculate_image(point, f, rotation=None):
```

```
    homo_point = convert_to_homogeneous(point)
```

```
    # if we supplied a rotation, do the rotation using another function
```

```
    if rotation is not None:
```

```
        homo_point = rotate_point(homo_point, rotation)
```

```
    # matrix to calculate the image
```

```
    focal_matrix = [[f, 0, 0, 0],
                    [0, f, 0, 0],
                    [0, 0, 1, 0]]
```

```

# multiply the projection matrix by the point.
mat = np.matmul(focal_matrix, homo_point)

# normalize the point and round answer to 2 decimal points. Need to round bc of rotation
issues
image_points = np.divide(mat, mat[2]).round(2)
return image_points[:-1]

def rotate_point(point, theta):
    # Rotate the homogeneous point around the z-axis. There is no translation so T = 0
    rotation_matrix = [[cos(theta), sin(theta) * -1, 0, 0],
                       [sin(theta), cos(theta), 0, 0],
                       [0, 0, 1, 0],
                       [0, 0, 0, 1]]
    return np.matmul(rotation_matrix, point)

def convert_to_homogeneous(matrix):
    # function to convert to homogeneous point. We don't necessarily need to copy matrix, but
    we did
    t = matrix.copy()
    t.append(1)
    return t

if __name__ == '__main__':
    print("Samuel Haberkorn\nProject 4\n-----")
    main()

```