CA4007 Computer Graphics and Image Processing

Assignment #1 - Camera Model

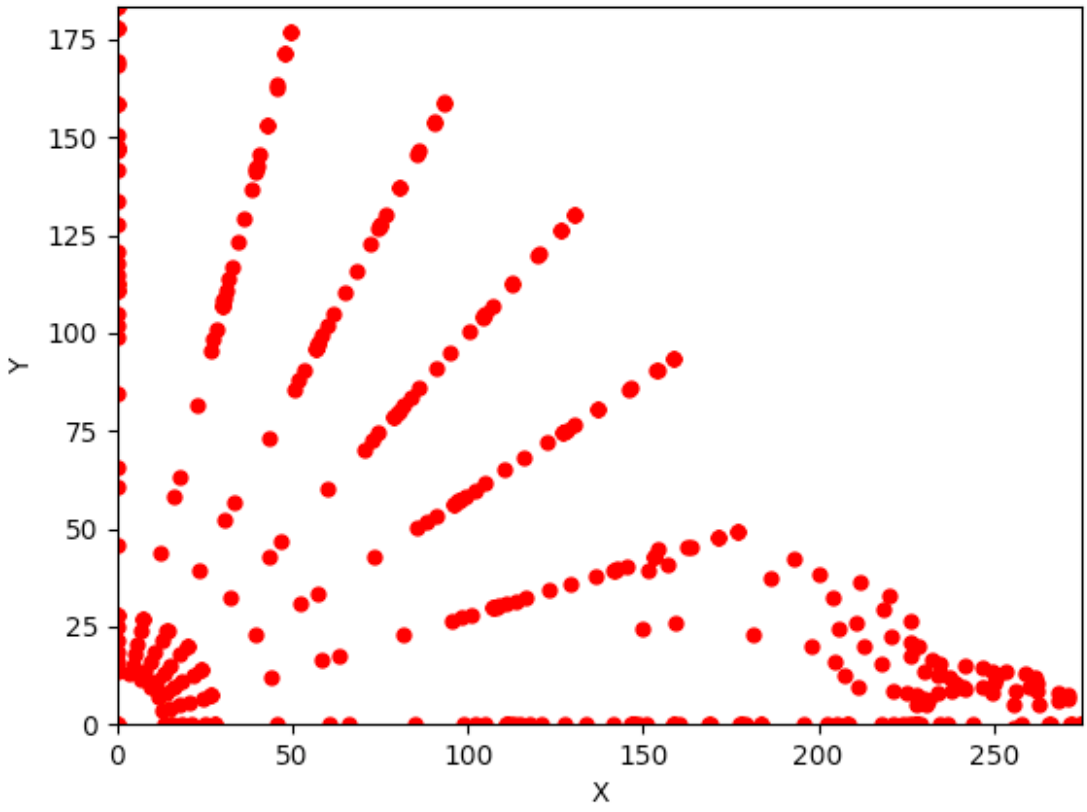

Figure 1: Projection with principle at [0, 0]

Objective 1

For the first objective, our goal is to perform a projection with the camera placed at location [0, 0, 10]. As seen in our result for this objective in figure 1 above, only the pixels with positive coordinates are visible. This is because we have specified that **c = [cx, cy] = [0, 0]**.

To begin this objective, we first need to define our intrinsic matrix (K), known as the internal camera matrix. This matrix will allow us to represent points in 3D space in coordinates with respect to the camera frame where the z-axis is the optical axis.

```
k = np.array([[1000, 0, 0, 0], [0, 1000, 0, 0], [0, 0, 1, 0]])
```

Now we define our extrinsic matrix (R|t), which takes the form of a rigid transformation matrix: a 3 × 3 rotation matrix in the left-block, and 3 × 1 translation column-vector in the right:

```
rt = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 10], [0, 0, 0, 1]])
```

Given the intrinsic and extrinsic parameters, the complete camera matrix model (M) can be presented in homogeneous coordinates by computing the dot product of our intrinsic matrix (K) and our extrinsic matrix (R|t):

```
m = k.dot(rt)
```

But before we can plot the point cloud model of the teapot we must first make sure the numpy array is eligible for matrix multiplication. We do this by first inserting a column of ones into the teapot numpy array, and then transposing the matrix so that it is eligible to be multiplied with our camera matrix (M).

```
teapot = np.insert(teapot, 3, 1, axis=1)
```

```
teapot = np.transpose(teapot)

result = m.dot(teapot)
```

The result is then transposed again back to its original state, where the 2D points for x and y are then computed by dividing each of them by the z coordinate.

```
result = np.transpose(result)

x = np.divide(result[: , 0], result[: , 2])

y = np.divide(result[: , 1], result[: , 2])
```

The axis is then labeled and the resulting 2D image is plotted.

```
ax.plot(x, y, 'r.', markersize=10)

ax.set_xlabel('X')

ax.set_ylabel('Y')

plt.show()
```

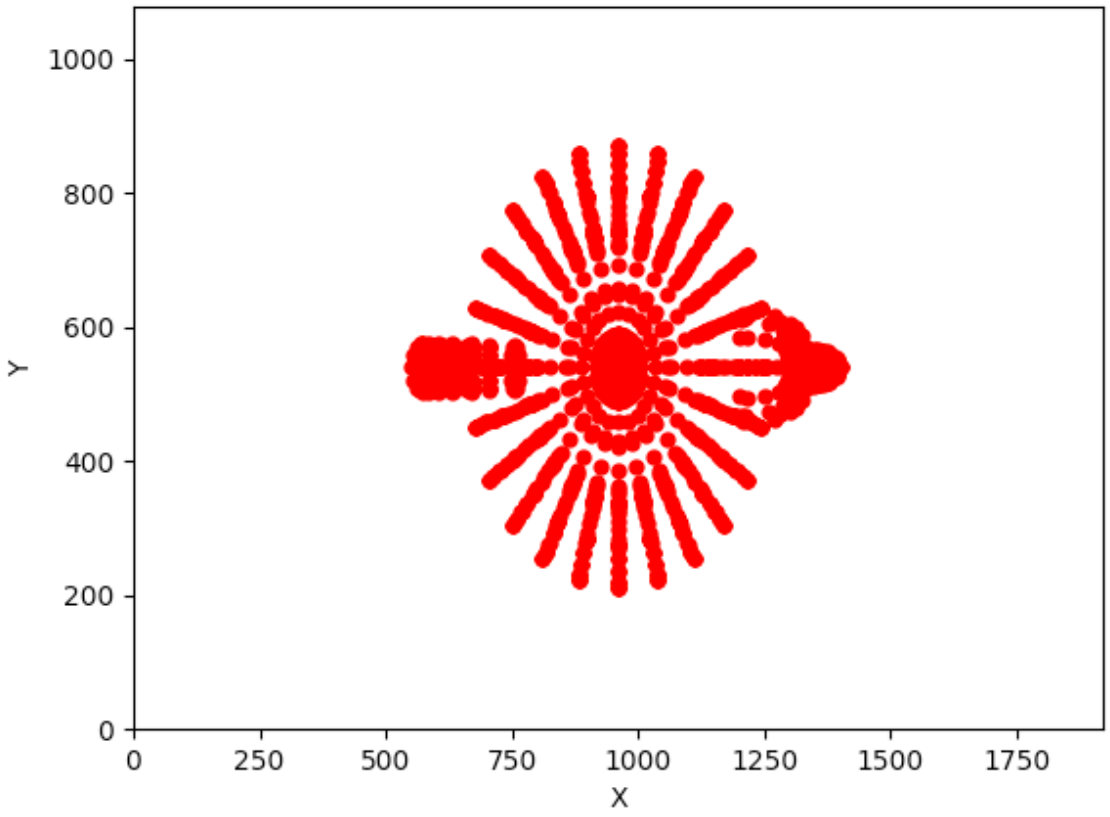

Figure 2: Projection with image resolution=1920 × 1080

Objective 2

For the second objective, our goal is to modify the intrinsics matrix (K) in order to capture an image with the resolution 1920 × 1080 pixels. As we are told that **c = [cx, cy]** is where the optical axis intersects the image plane, the values for both **cx** and **cy** were changed as follows:

k = np.array([[1600, 0, **960**, 0], [0, 1800, **540**, 0], [0, 0, 1, 0]])

These values were approximated with half the width and height of the image.

As you can see, the focal lengths of x and y were also changed, in the hopes of achieving a better match to the solution for this objective.

The axis limits for x and y were also adjusted:

ax.set_xlim(0, 1920)

ax.set_ylim(0, 1080)

References

Dr. Hossein Javidnia, CA4007 Computer Graphics and Image Processing Notes, Source: https://loop.dcu.ie/pluginfile.php/4292791/mod_resource/content/1/Lecture%233.pdf