HW03

Question 2: Choose a 3-D resolution granularity, perform voxel filter (or box grid filter) to down-sample all the 3D point cloud points to the 3D voxel space points, and visualize the result points

In [1]:
```
import numpy as np
import argparse

filename   = str("bin_files/002_00000001.bin")
pointcloud = np.fromfile(filename, dtype=np.float32)
pointcloud = pointcloud.reshape([-1,4])

print('LiDAR data loaded as a variable pointcloud')

str1= str('\nLidar data file : ') + str(filename) + str('\nSize of pointclo
ud data = ') + str(pointcloud.shape)
print(str1)
```

```
LiDAR data loaded as a variable pointcloud

Lidar data file : bin_files/002_00000001.bin
Size of pointcloud data = (92246, 4)
```

In [2]:
```
def voxel_downsample(pointloud,leaf_size):
    import numpy as np
    import pcl

    # Convert numpy array to pcd format ref:https://github.com/Sirokujira/p
ython-pcl/blob/rc_patches4/examples/official/Filtering/VoxelGrid_160.py
    p = pcl.PointCloud(np.array(pointcloud[:,0:3], dtype=np.float32))

    # voxel downsampling from pcl lib.
    sor = p.make_voxel_grid_filter()
    sor.set_leaf_size(leaf_size,leaf_size,leaf_size)
    cloud_filtered = sor.filter()

    # Convert back pcd format pointcloud to numpy array
    a = np.asarray(cloud_filtered)      # NumPy view on the cloud
    return a
```

In [8]:
```
def visualize_3d(pointcloud,cloud_color,Point_size):
    import pptk
    import numpy as np

    # Extract first three points as x y z inputs and 4th for reflectivity v
alue
    P = pointcloud[:,0:3]

    a = pointcloud.shape[0]
    R = np.ones((a))*20

    if pointcloud.shape[1]==4:
        R   = pointcloud[:,3]

    # define color channels
    rgb = np.ones((P.shape))*cloud_color    # for grayish effect [200,200,2
00]

    rgb[:,0] = rgb[:,0]*(255-R)/255
    rgb[:,1] = rgb[:,1]*(255-R)/255
    rgb[:,2] = rgb[:,2]*(255-R)/255

    # Visualize point cloud
    v = pptk.viewer(P)
    v.attributes(rgb / 255, R)
    v.set(lookat = [0,0,0])
    v.set(point_size=Point_size)  #for better visualization point_size = 0.
001
```

In [9]:
```
leaf_size = 0.20
voxel_pointcloud = voxel_downsample(pointcloud,leaf_size)

str_voxel_print  = str('Pointcloud downsampled by ') + str(100-float(int(vo
xel_pointcloud.shape[0]*10000/pointcloud.shape[0]))/100)+str(' percentage')
print(str_voxel_print)

visualize_3d(voxel_pointcloud,[200,200,200],0.001)
```

```
Pointcloud downsampled by 77.58 percentage
```

In [71]:
```
voxel_pointcloud.shape
```

Out[71]:
```
(8062, 3)
```