3D Data Processing

Point Clouds Processing
Filtering

Hyoseok Hwang

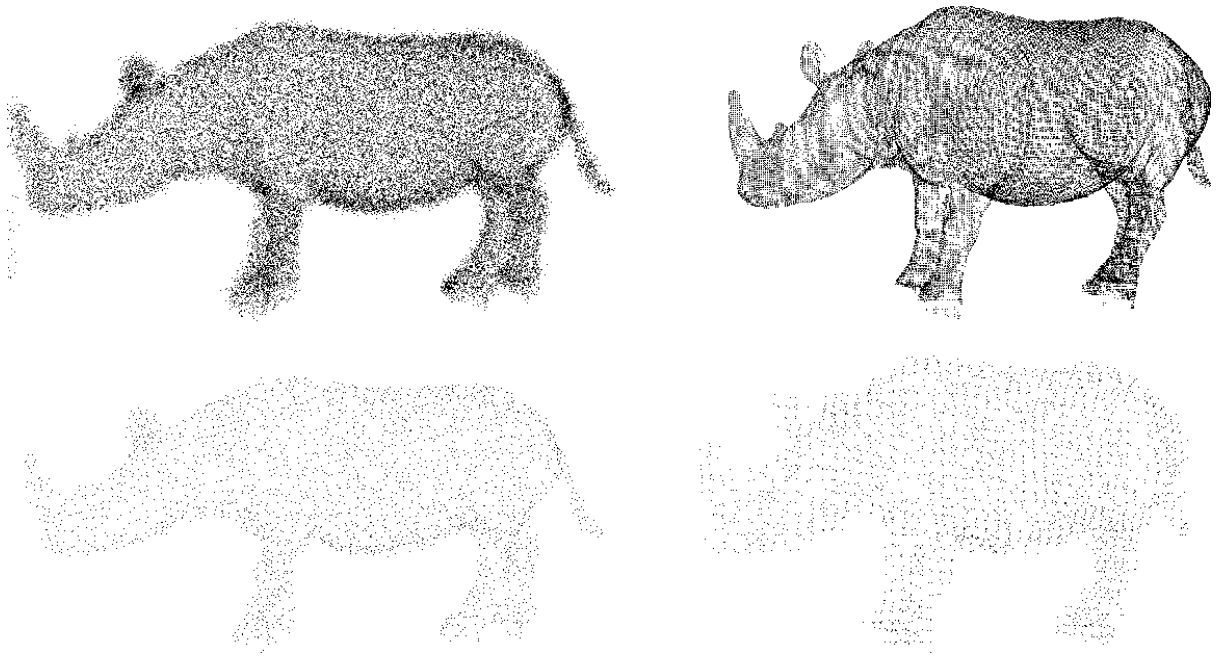
Lectures are based on Open3D functions

Today

- Filtering
  - Down-sampling
    - Random sampling
    - Uniform sampling
    - Voxel-based sampling
  - Cropping
  - Outlier removal
    - Radius outlier removal
    - Statistical outlier removal

Down-sampling

- Down-sampling
  - Reducing the number of points.
  - For reducing the running time of the processing step
  - To select an exact number of points for training



Down-sampling

- Down-sampling methods
  - Random sampling
    - Select random points from input point clouds
  - Uniform sampling
    - Selects points uniformly regarding their order
  - Voxel-based sampling
    - Create a 3D Voxel grid of which size is H x W x D
    - Each voxel includes the points that belong to the same intervals regarding the 3 axes.

Down-sampling

- ply_point_cloud = o3d.data.PLYPointCloud()
- pcd = o3d.io.read_point_cloud(ply_point_cloud.path)
- Print(pcd)
- Print(np.as array(pcd.points))

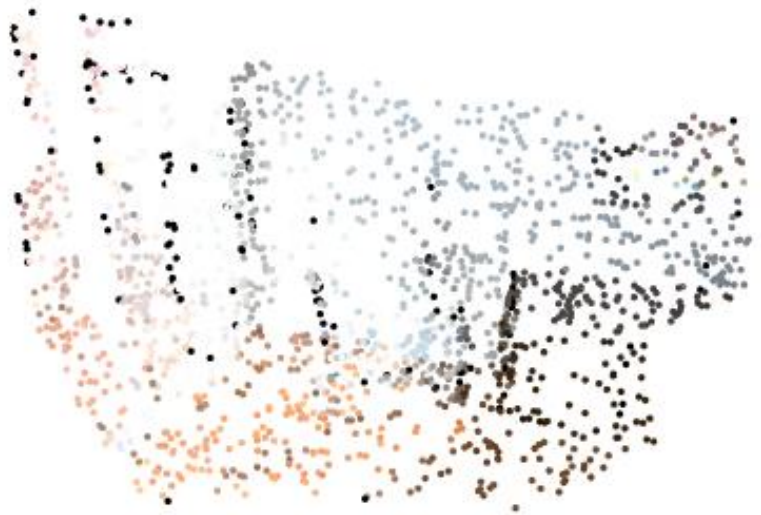
```
PointCloud with 196133 points. [[0.65234375
0.84686458 2.37890625] [0.65234375
0.83984375 2.38430572] [0.66737998
0.83984375 2.37890625] ... [2.00839925
2.39453125 1.88671875] [2.00390625
2.39488506 1.88671875] [2.00390625
2.39453125 1.88793314]]
```

PCD: http://www.open3d.org/docs/0.14.1/python_api/open3d.geometry.PointCloud.html

Down-sampling

- Random_down_sample(sampling_ratio=0.01)
  - Sampling_ratio: The ratio of the number of sampled points to the number of total points.
  - Return: open3d.geometry.PointCloud

```
random_pcd =
pcd.random_down_sample(sampling_ratio=0.01)
o3d.visualization.draw_geometries([random_pcd])
print(random_pcd)

PointCloud with 1961 points.
```



Down-sampling

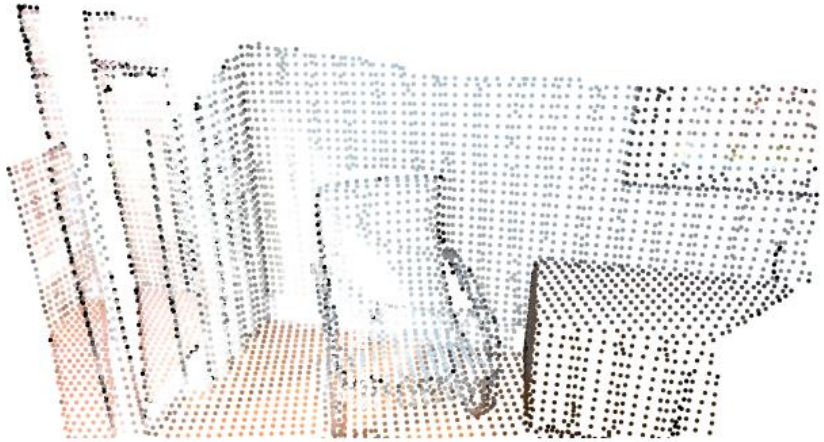
- uniform_down_sample(every_k_points=10)
  - every_k_points: Samples every kth point in the set of all point clouds.
  - Return: open3d.geometry.PointCloud

```
uniform_pcd =
pcd.uniform_down_sample(every_k_points=10)
o3d.visualization.draw_geometries([uniform_pcd])
print(uniform_pcd)

PointCloud with 19614 points.
```

Down-sampling

- voxel_down_sample(voxel_size=0.04)
  - voxel_size(v) : Samples one point in every v x v x v voxel grid.
  - Return: open3d.geometry.PointCloud

```
voxel_pcd = pcd.voxel_down_sample(voxel_size=0.04)
o3d.visualization.draw_geometries([voxel_pcd])
print(voxel_pcd)

PointCloud with 19614 points.
```



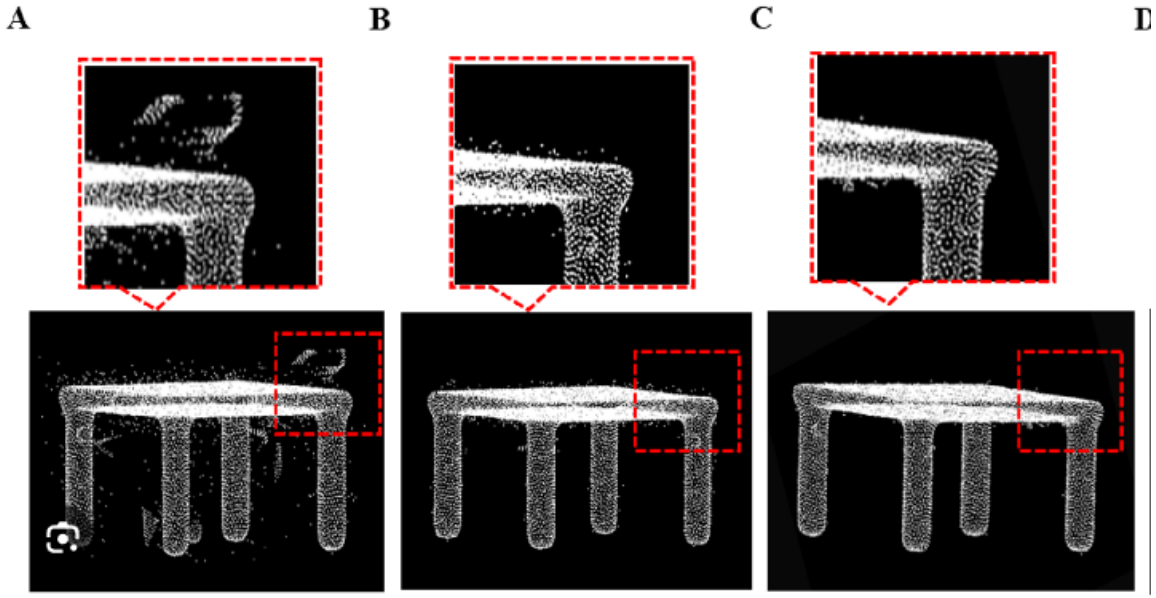
Pass-through filter

- applies constraints on the input data
- Crop
  - Create 3D bounding box
  - Apply crop to PCD

```
# Create bounding box:
bounds = [[2, 2.8], [1.4, 2.4], [0.8, 1.6]]  # set the bounds
bounding_box_points = list(itertools.product(*bounds))
bounding_box =
o3d.geometry.AxisAlignedBoundingBox.create_from_points(o3d.utility.Vector3dV
ector(bounding_box_points))  # create bounding box object

# Crop the point cloud using the bounding box:
pcd_croped = pcd.crop(bounding_box)
```

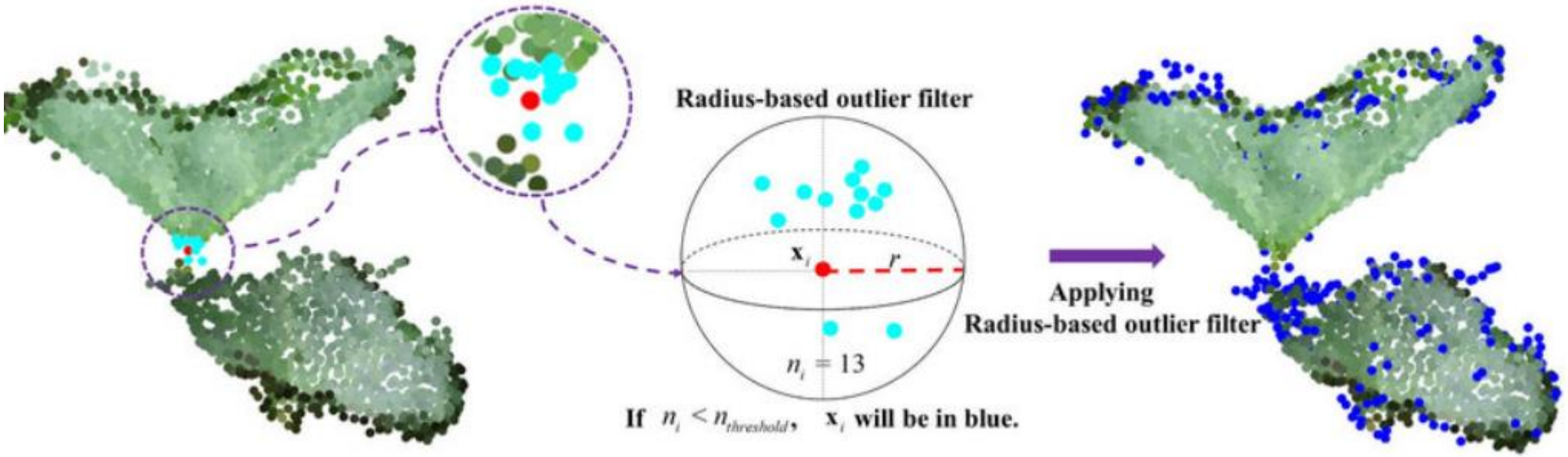
Outlier removal

- Remove noisy points
  - Radius outlier removal
  - Statistical outlier removal



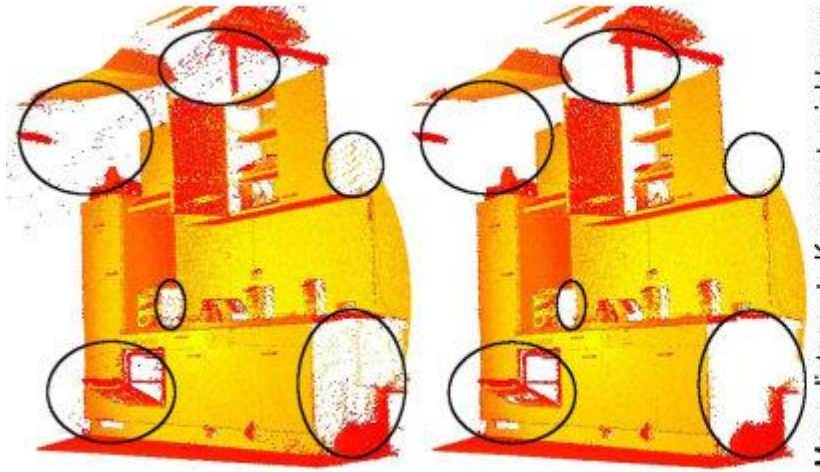
Outlier removal

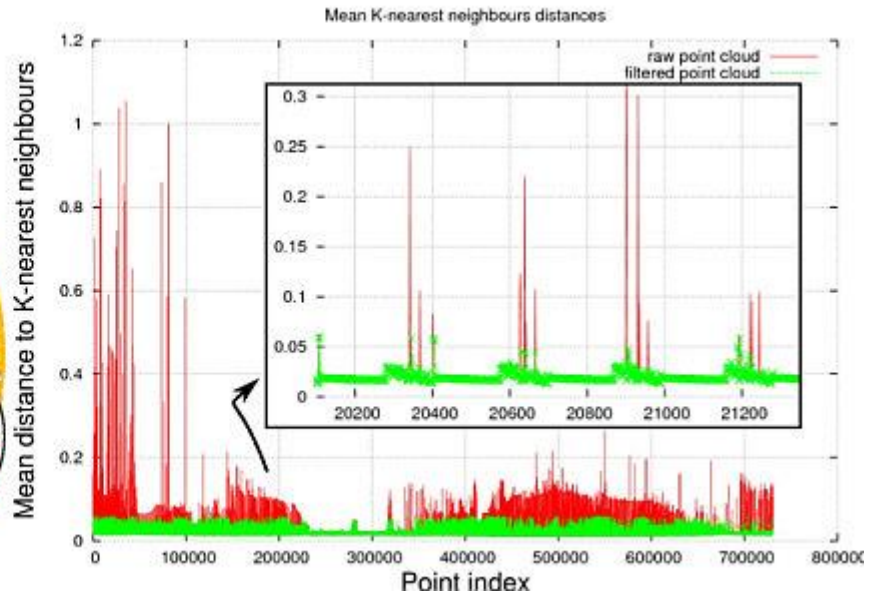
- Remove noisy points
  - Radius outlier removal
    - Remove points if the number of neighborhood points in radius r is less than the threshold



Outlier removal

- Remove noisy points
  - Statistical outlier removal
    - removes points that are further away from their neighbors
    - For each point the mean distance from it to all its neighbors is computed.
    - if the mean distance of the point is outside an interval defined by the global distances mean and standard deviation then the point is an outlier.





https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical_outlier.html

Outlier removal

- Generate noisy image

```
#create noisy image
random_pcd_noisy = pcd.voxel_down_sample(voxel_size=0.01)
mat_pcd= np.asarray(random_pcd_noisy.points)
for i in range(0, mat_pcd.shape[0], 50):
  mat_pcd[i,:] = mat_pcd[i,:] + (np.random.random((1,3)) - 0.5) * 0.5
random_pcd_noisy.points = o3d.utility.Vector3dVector(mat_pcd)
o3d.visualization.draw_geometries([random_pcd_noisy])
```

Outlier removal

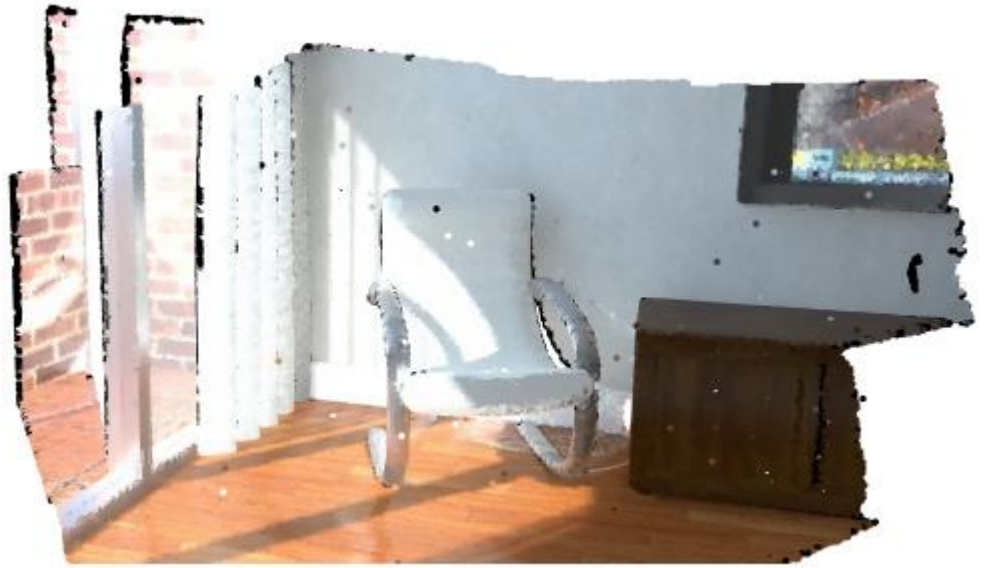
- Radius outlier removal
  - remove_radius_outlier(nb_points=16, radius=0.05)
    - nb_points: pick the minimum amount of points that the sphere should contain.
    - Radius: defines the radius of the sphere that will be used for counting the neighbors.

```
pcd_rad, ind_rad =
random_pcd_noisy.remove_radius_outlier(nb_po
ints=16, radius=0.05)
outlier_rad_pcd =
random_pcd_noisy.select_by_index(ind_rad,
invert=False)
o3d.visualization.draw_geometries([outlier_r
ad_pcd])
```

Outlier removal

- Statistical outlier removal
  - remove_statistical_outlier(nb_neighbors=20, std_ratio=2.0)
  - nb_neighbors: how many neighbors are taken into account in order to calculate the average distance for a given point.
  - Std_ratio: setting the threshold level based on the standard deviation of the average distances across the point cloud.

```
pcd_stat, ind_stat =
random_pcd_noisy.remove_statistical_outlier
(nb_neighbors=20, std_ratio=1.0)
outlier_stat_pcd =
random_pcd_noisy.select_by_index(ind_stat,
invert=False)
o3d.visualization.draw_geometries([outlier_
stat_pcd])
```



Thank you