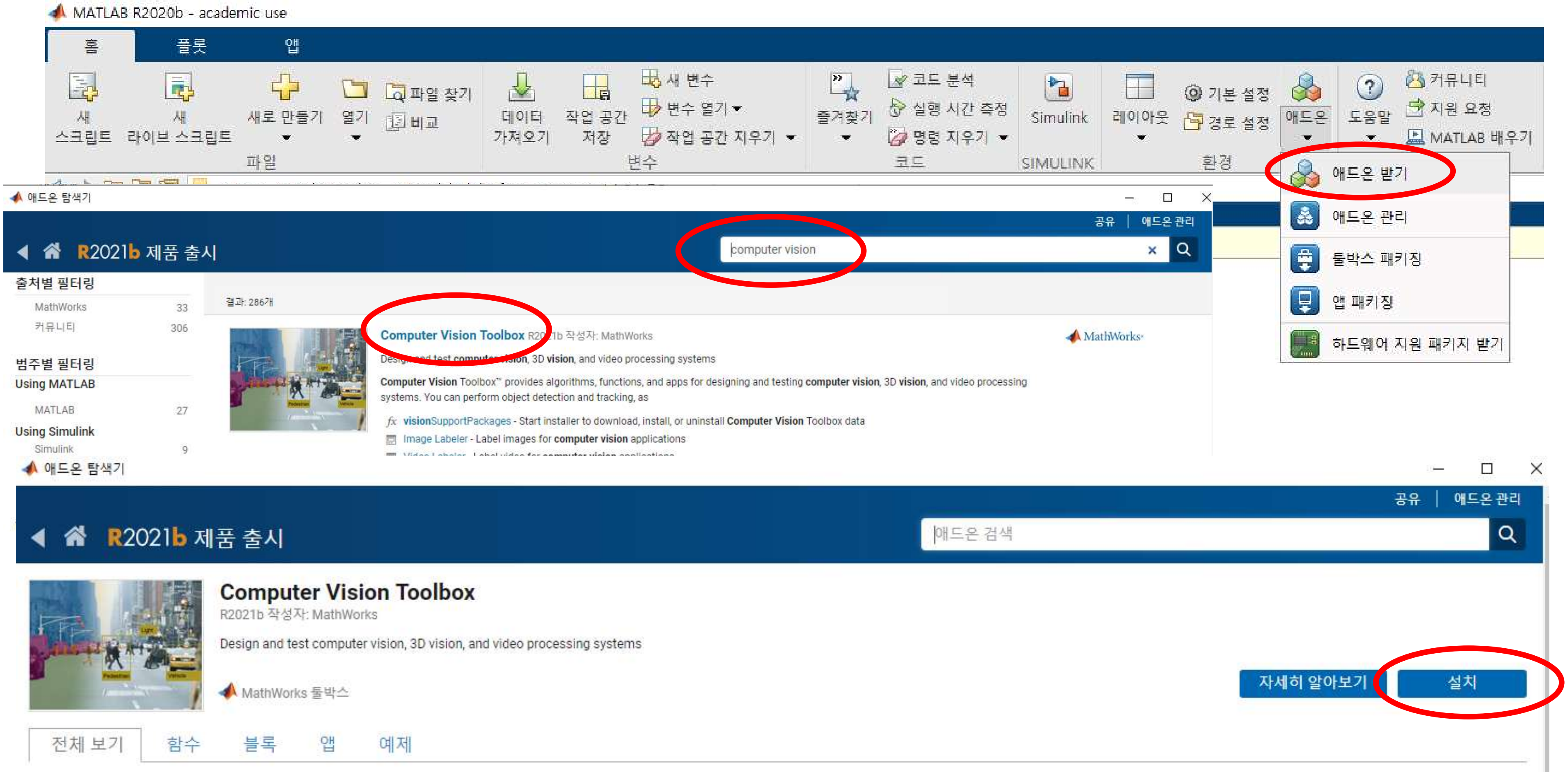
Matlab Point Cloud Processing

- Computer Vision Toolbox 안에 존재 -> 애드온 받기



Computer Vision Toolbox R2021b 작성자: MathWorks

Design and test computer vision, 3D vision, and video processing systems

Computer Vision Toolbox™ provides algorithms, functions, and apps for designing and testing computer vision, 3D vision, and video processing systems. You can perform object detection and tracking, as

visionSupportPackages - Start installer to download, install, or uninstall Computer Vision Toolbox data

Image Labeler - Label images for computer vision applications

Video Labeler - Label video for computer vision applications

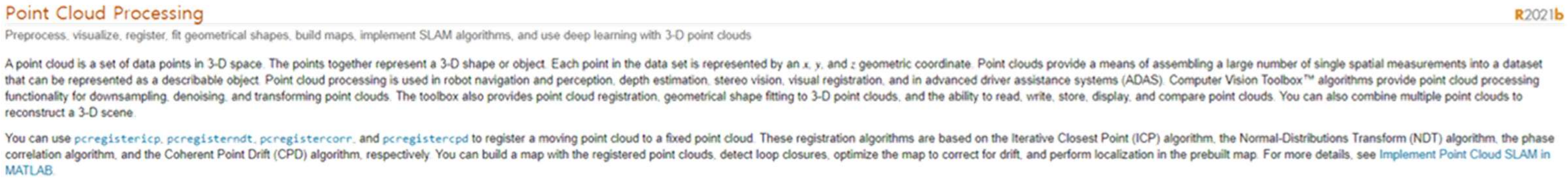
Computer Vision Toolbox

R2021b 작성자: MathWorks

Design and test computer vision, 3D vision, and video processing systems

Matlab Point Cloud Processing

자율주행센서



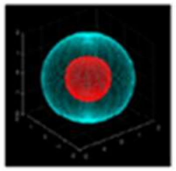
Point Cloud Processing

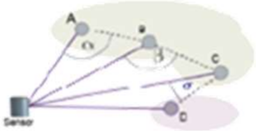
R2021b

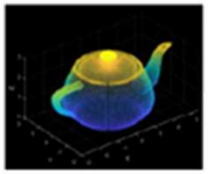
Preprocess, visualize, register, fit geometrical shapes, build maps, implement SLAM algorithms, and use deep learning with 3-D point clouds

A point cloud is a set of data points in 3-D space. The points together represent a 3-D shape or object. Each point in the data set is represented by an *x*, *y*, and *z* geometric coordinate. Point clouds provide a means of assembling a large number of single spatial measurements into a dataset that can be represented as a describable object. Point cloud processing is used in robot navigation and perception, depth estimation, stereo vision, visual registration, and in advanced driver assistance systems (ADAS). Computer Vision Toolbox™ algorithms provide point cloud processing functionality for downsampling, denoising, and transforming point clouds. The toolbox also provides point cloud registration, geometrical shape fitting to 3-D point clouds, and the ability to read, write, store, display, and compare point clouds. You can also combine multiple point clouds to reconstruct a 3-D scene.

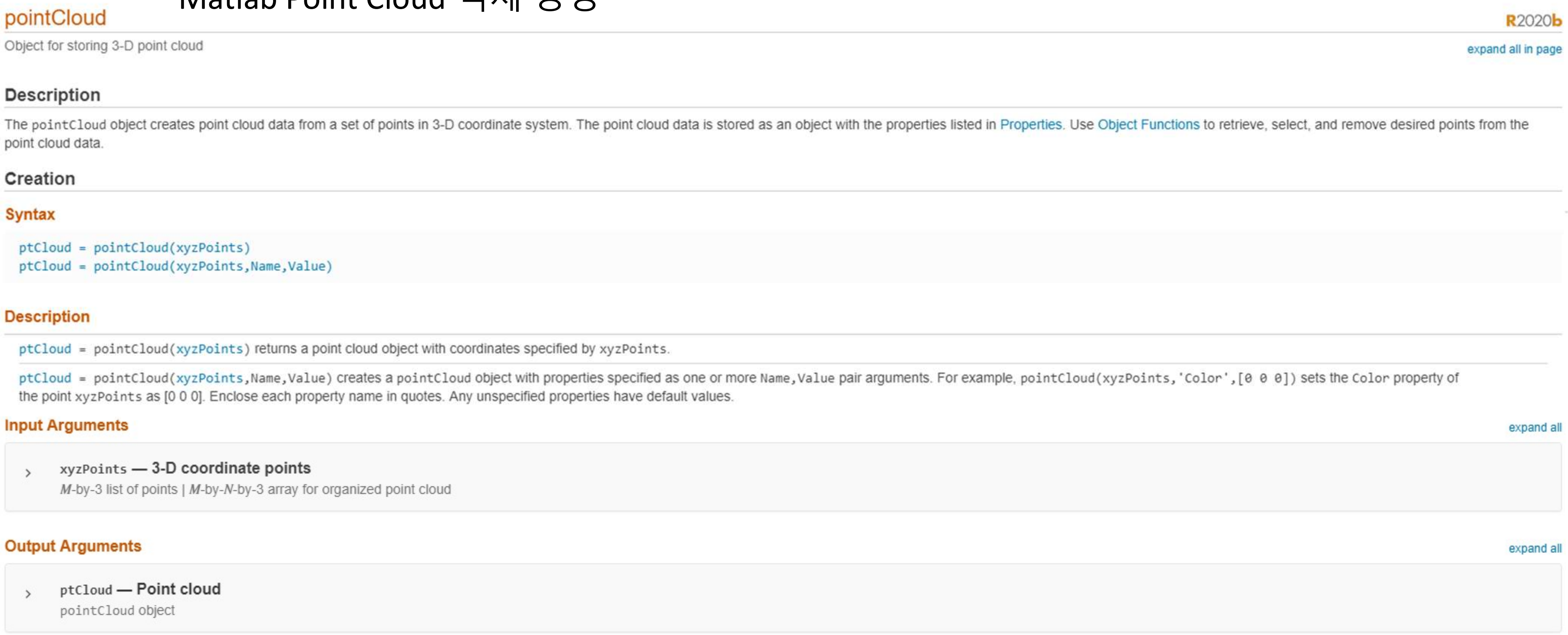
You can use `pcregistericp`, `pcregisterndt`, `pcregistercorr`, and `pcregistercpd` to register a moving point cloud to a fixed point cloud. These registration algorithms are based on the Iterative Closest Point (ICP) algorithm, the Normal-Distributions Transform (NDT) algorithm, the phase correlation algorithm, and the Coherent Point Drift (CPD) algorithm, respectively. You can build a map with the registered point clouds, detect loop closures, optimize the map to correct for drift, and perform localization in the prebuilt map. For more details, see Implement Point Cloud SLAM in MATLAB.







Matlab Point Cloud 객체 생성



pointCloud

R2020b

Object for storing 3-D point cloud

expand all in page

Description

The `pointCloud` object creates point cloud data from a set of points in 3-D coordinate system. The point cloud data is stored as an object with the properties listed in Properties. Use Object Functions to retrieve, select, and remove desired points from the point cloud data.

Creation

Syntax

```
ptCloud = pointCloud(xyzPoints)
ptCloud = pointCloud(xyzPoints,Name,Value)
```

Description

`ptCloud = pointCloud(xyzPoints)` returns a point cloud object with coordinates specified by `xyzPoints`.

`ptCloud = pointCloud(xyzPoints,Name,Value)` creates a `pointCloud` object with properties specified as one or more `Name,Value` pair arguments. For example, `pointCloud(xyzPoints,'Color',[0 0 0])` sets the `Color` property of the point `xyzPoints` as [0 0 0]. Enclose each property name in quotes. Any unspecified properties have default values.

Input Arguments

expand all

› `xyzPoints` — **3-D coordinate points**
M-by-3 list of points | *M*-by-*N*-by-3 array for organized point cloud

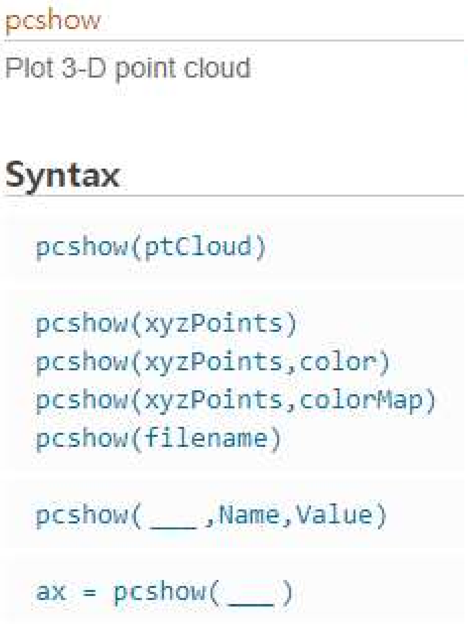
Output Arguments

expand all

› `ptCloud` — **Point cloud**
`pointCloud` object

Matlab Point Cloud Processing

자율주행센서



pcshow

Plot 3-D point cloud

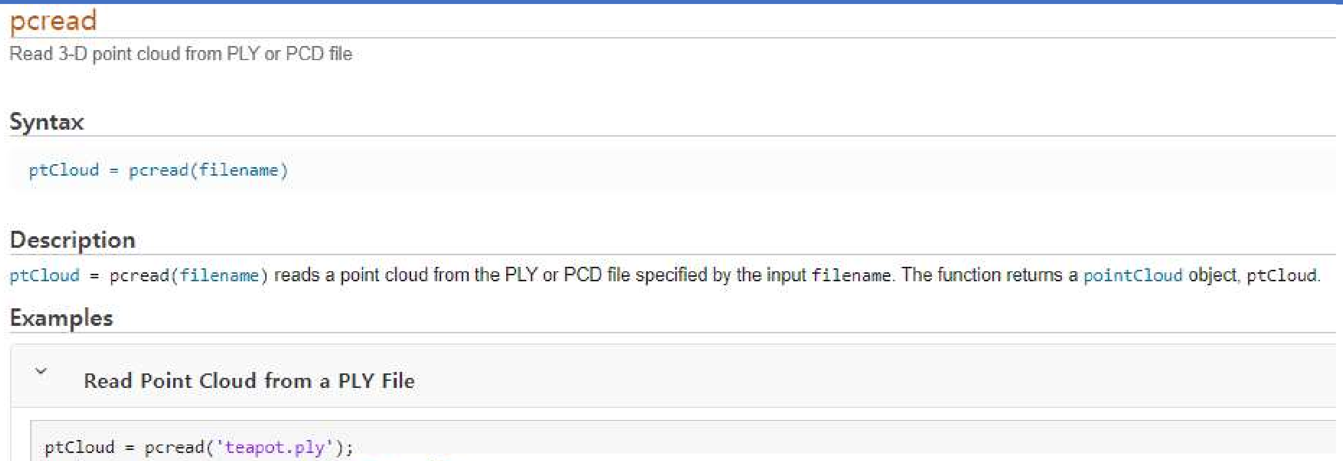
Syntax

```
pcshow(ptCloud)

pcshow(xyzPoints)
pcshow(xyzPoints,color)
pcshow(xyzPoints,colorMap)
pcshow(filename)

pcshow(___,Name,Value)

ax = pcshow(___)
```



pcread

Read 3-D point cloud from PLY or PCD file

Syntax

```
ptCloud = pcread(filename)
```

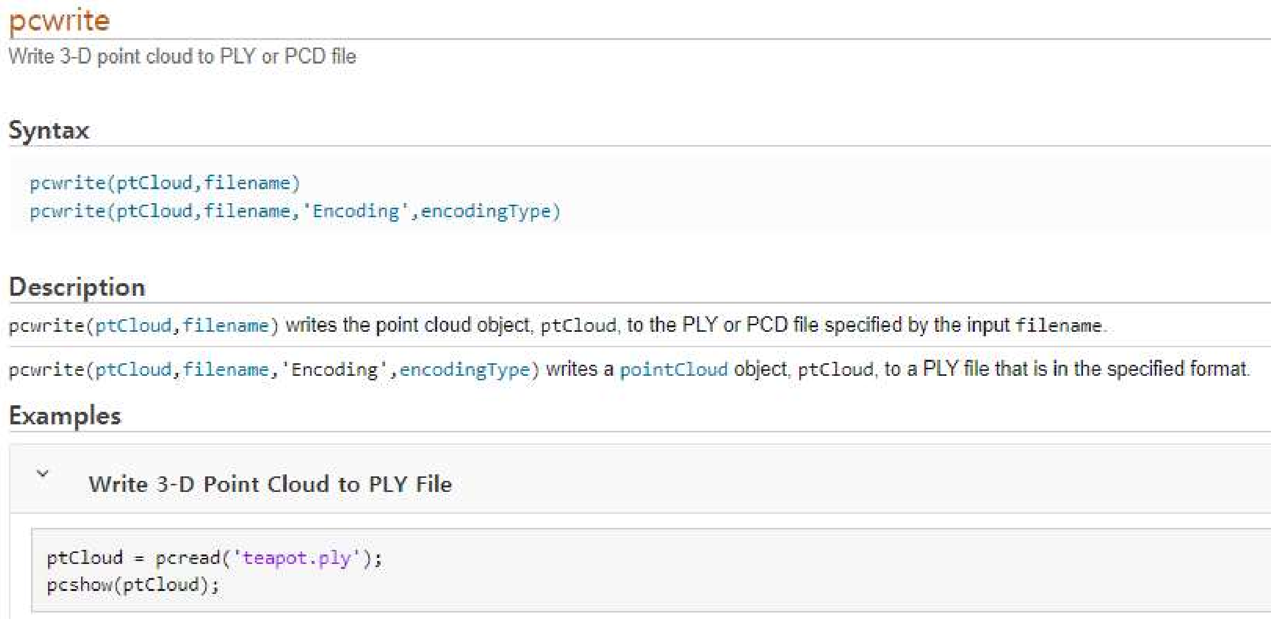
Description

`ptCloud = pcread(filename)` reads a point cloud from the PLY or PCD file specified by the input `filename`. The function returns a `pointCloud` object, `ptCloud`.

Examples

Read Point Cloud from a PLY File

```
ptCloud = pcread('teapot.ply');
pcshow(ptCloud);
```



pcwrite

Write 3-D point cloud to PLY or PCD file

Syntax

```
pcwrite(ptCloud,filename)
pcwrite(ptCloud,filename,'Encoding',encodingType)
```

Description

`pcwrite(ptCloud,filename)` writes the point cloud object, `ptCloud`, to the PLY or PCD file specified by the input `filename`.

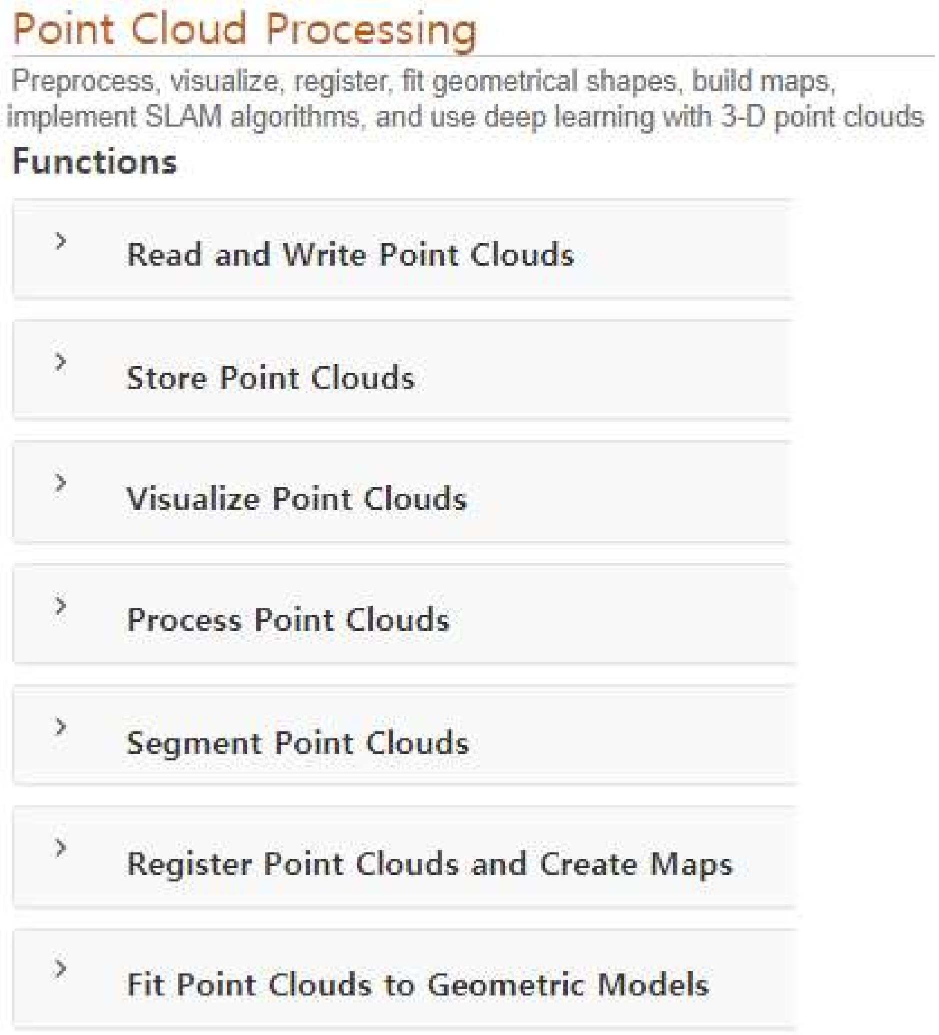
`pcwrite(ptCloud,filename,'Encoding',encodingType)` writes a `pointCloud` object, `ptCloud`, to a PLY file that is in the specified format.

Examples

Write 3-D Point Cloud to PLY File

```
ptCloud = pcread('teapot.ply');
pcshow(ptCloud);
```

Matlab Point Cloud Processing



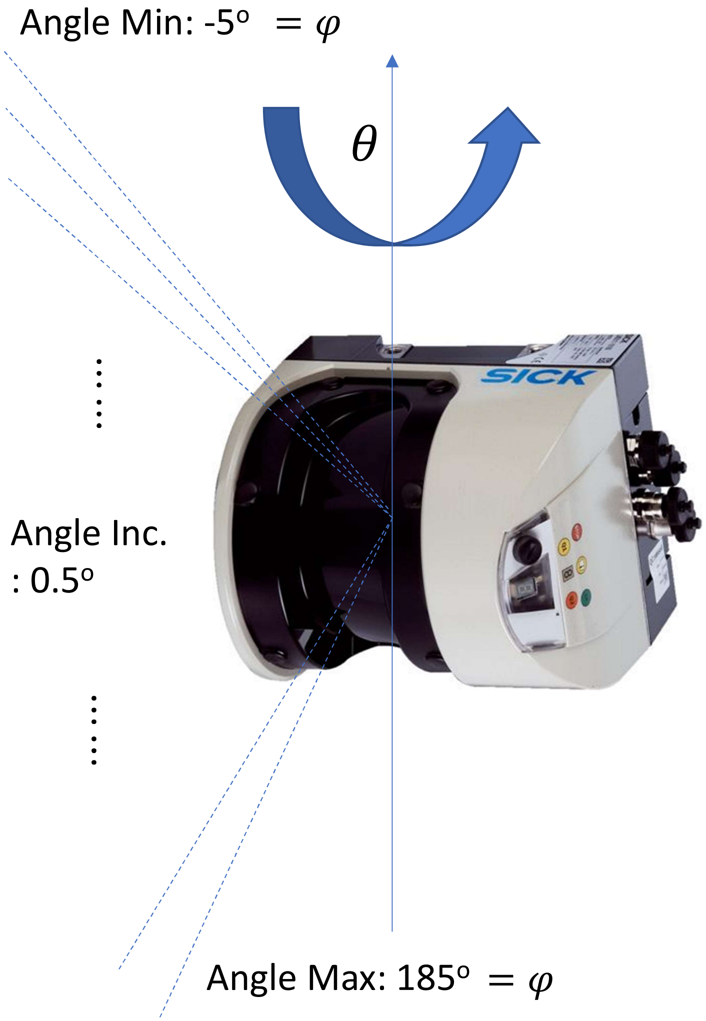
Point Cloud Processing

Preprocess, visualize, register, fit geometrical shapes, build maps, implement SLAM algorithms, and use deep learning with 3-D point clouds

Functions

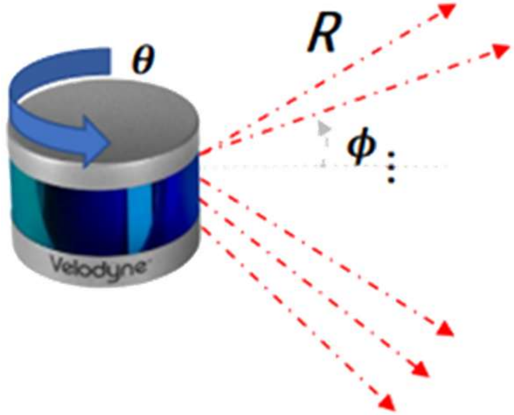
- › Read and Write Point Clouds
- › Store Point Clouds
- › Visualize Point Clouds
- › Process Point Clouds
- › Segment Point Clouds
- › Register Point Clouds and Create Maps
- › Fit Point Clouds to Geometric Models

Matlab 실습과제

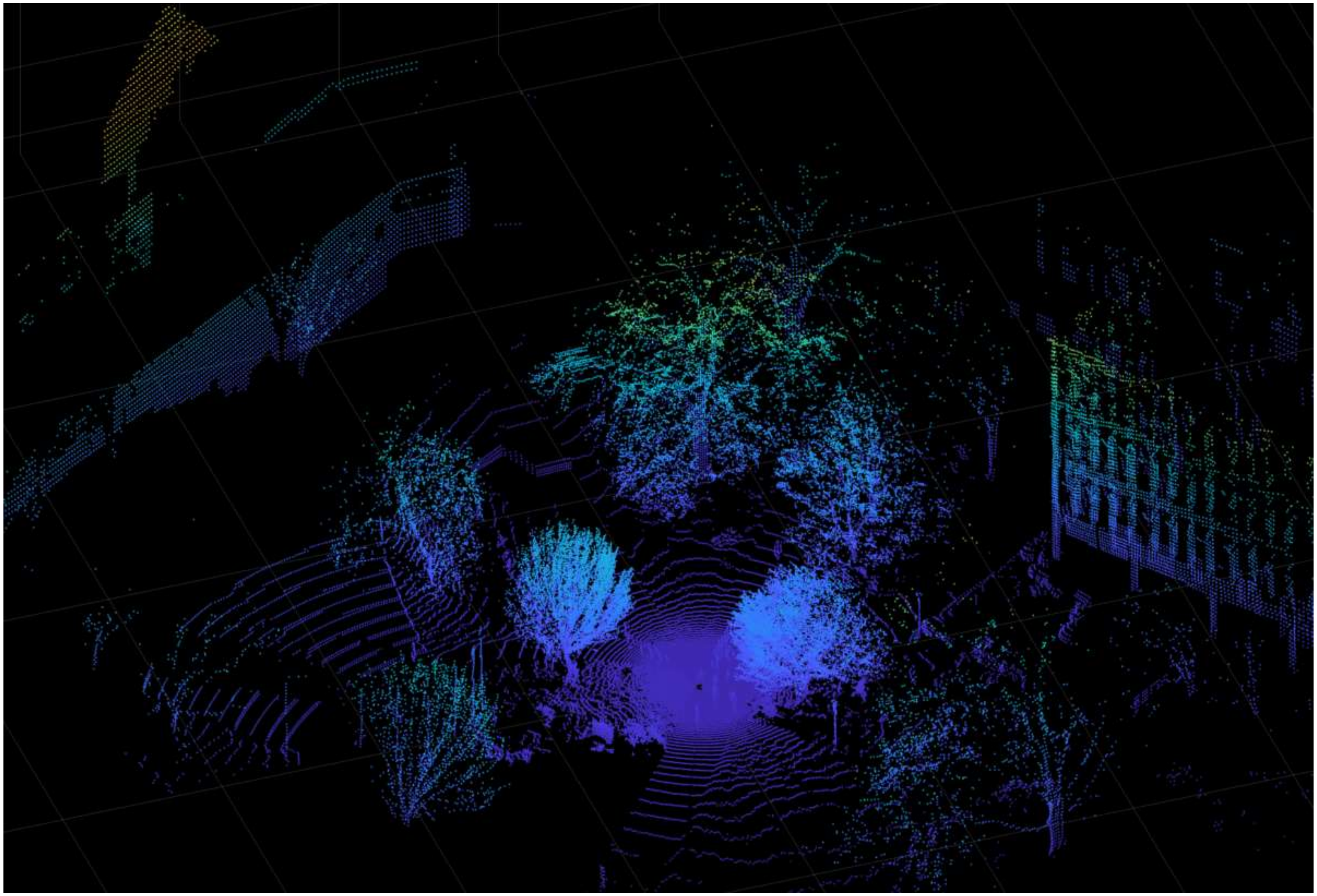


- Phi_angle = -5° ~ 185° (0.5° 간격, 381개)
- Theta_angle = 0.36° ~ 360° (0.36° 간격, 1000개)
- Range data 제공 -> `load('LidarRangeData.mat')`
- 위의 세개 데이터를 이용하여 x, y, z 포인트클라우드 데이터를 생성하라.
- 아래 식을 사용할 수 있으나 각도의 정의와 좌표계 관계를 잘 생각해야 함.

$$x = R\cos\varphi\cos\theta$$
$$y = R\cos\varphi\sin\theta$$
$$z = R\sin\varphi$$

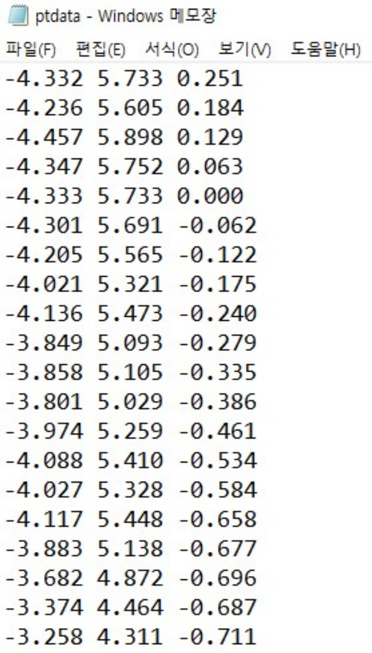


Matlab 결과



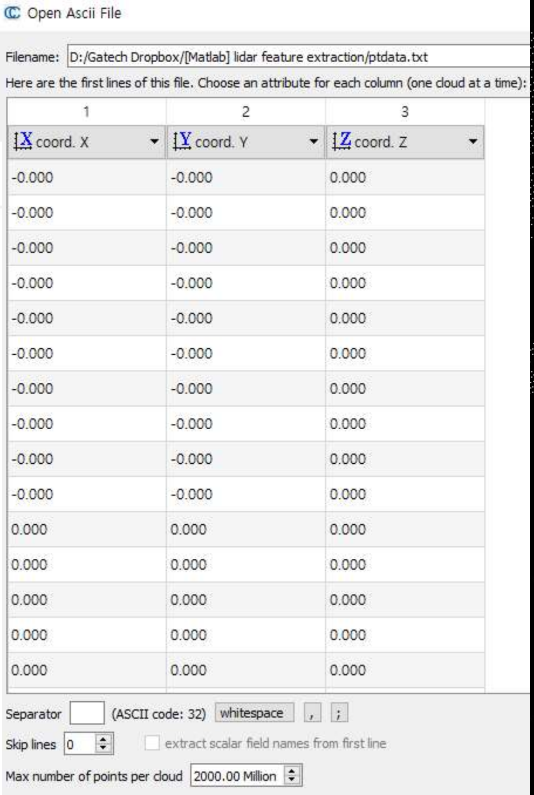
CloudCompare로 열기

```
fid = fopen('ptdata.txt','wt');
for i = 1:length(points)
    fprintf(fid,'%.3f %.3f %.3f ', points(i,:));
    fprintf(fid,'\n');
end
fclose(fid);
```



ptdata - Windows 메모장

```
-4.332 5.733 0.251
-4.236 5.605 0.184
-4.457 5.898 0.129
-4.347 5.752 0.063
-4.333 5.733 0.000
-4.301 5.691 -0.062
-4.205 5.565 -0.122
-4.021 5.321 -0.175
-4.136 5.473 -0.240
-3.849 5.093 -0.279
-3.858 5.105 -0.335
-3.801 5.029 -0.386
-3.974 5.259 -0.461
-4.088 5.410 -0.534
-4.027 5.328 -0.584
-4.117 5.448 -0.658
-3.883 5.138 -0.677
-3.682 4.872 -0.696
-3.374 4.464 -0.687
-3.258 4.311 -0.711
```



Open Ascii File

Filename: D:/Gatech Dropbox/[Matlab] lidar feature extraction/ptdata.txt

Here are the first lines of this file. Choose an attribute for each column (one cloud at a time):

1	2	3
coord. X	coord. Y	coord. Z
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
-0.000	-0.000	0.000
0.000	0.000	0.000
0.000	0.000	0.000
0.000	0.000	0.000
0.000	0.000	0.000
0.000	0.000	0.000

Separator (ASCII code: 32)

Skip lines 0

□ extract scalar field names from first line

Max number of points per cloud 2000.00 Million