3D Data Processing

Point Clouds Processing

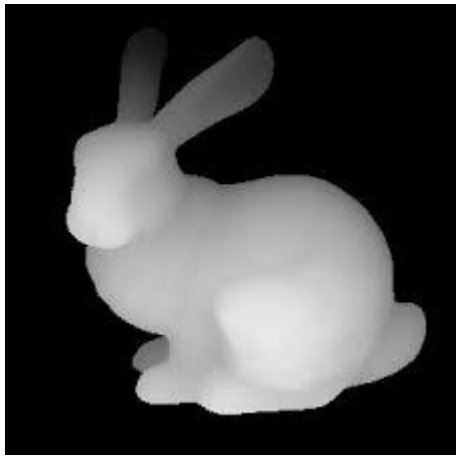
Hyoseok Hwang

Today

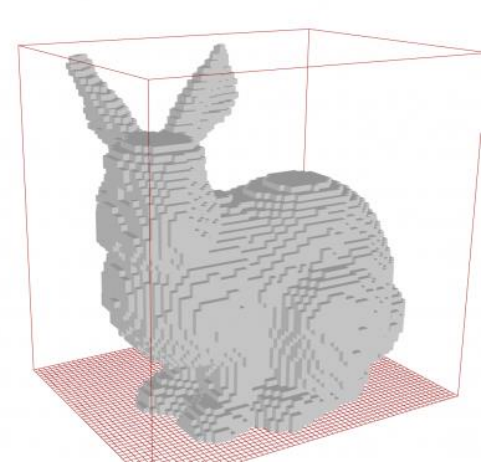
- Representation of 3D data
  - Voxel
  - Point Clouds
  - Mesh
- Point Clouds
  - How to Get?
- Point Clouds processing overview

3D Shape representation

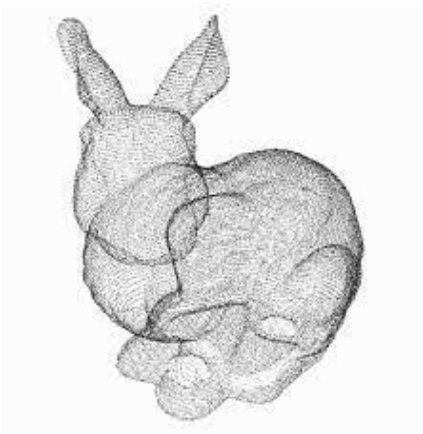
- How can we represent 3D shape?



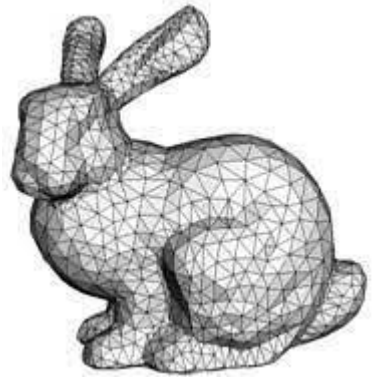
depth map



voxel grid



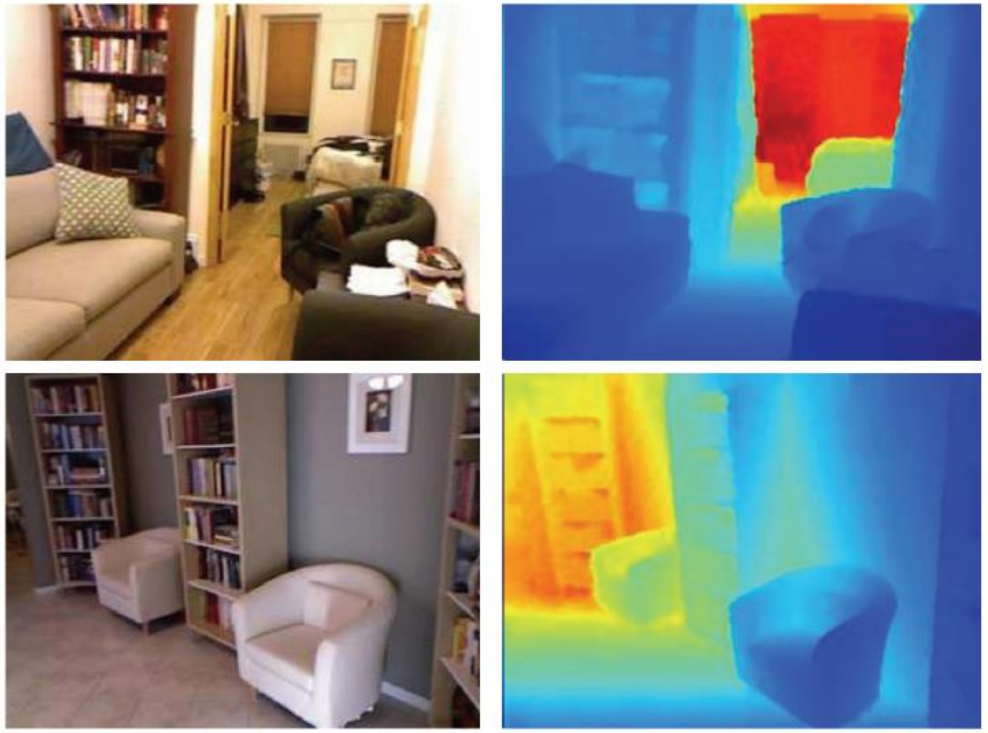
point clouds



mesh

3D Shape representation

- Depth map
  - For each pixel, depth map gives distanc e from the camera to the object in the world at that pixel
- RGB image + Depth image
  - = RGB-D Image (2.5D)
- This type of data can be recorded directly for some types of 3D sensors (e.g. Microsoft Kinect)



RGB Image: 3 x H x W Depth Map: H x W

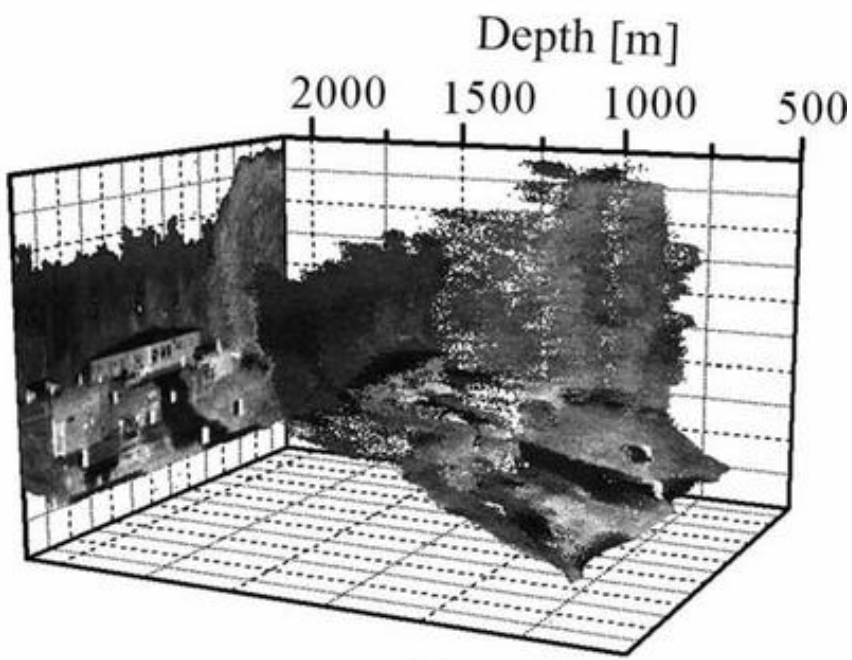
Eigen and Fergus, “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture”, ICCV 2015

images from lecture note of university of Michigan

3D Shape representation

- 3D reconstruction using depth map

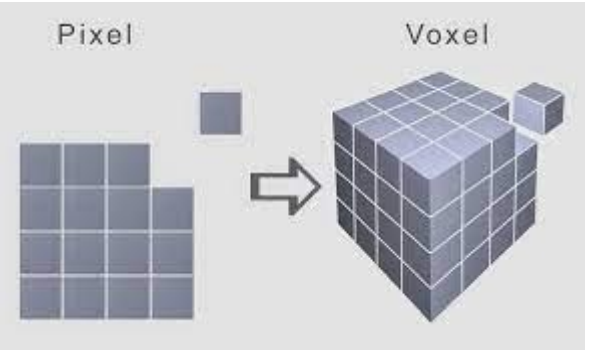
(a)

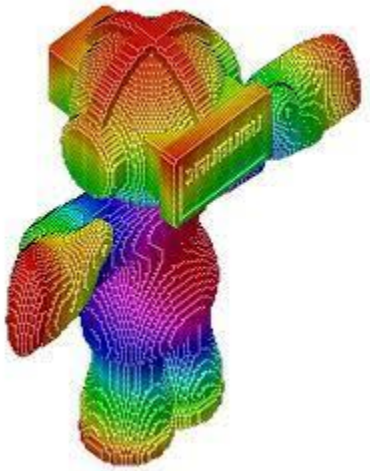


(b)

3D Shape representation

- Voxel grid
  - voxel: volume pixel
  - a voxel represents a value on a regular grid in 3D space.
  - Represent a shape with a V x V x V grid of occupancies
  - Pros
    - Conceptually simple: just a 3D grid
    - Convolution is possible
  - Cons
    - Need high spatial resolution to capture fine structures
      - Most of volume space is empty!
    - Scaling to high resolutions is nontrivial
      - Good for representing object
      - Bad for representing environment (maybe impossible)

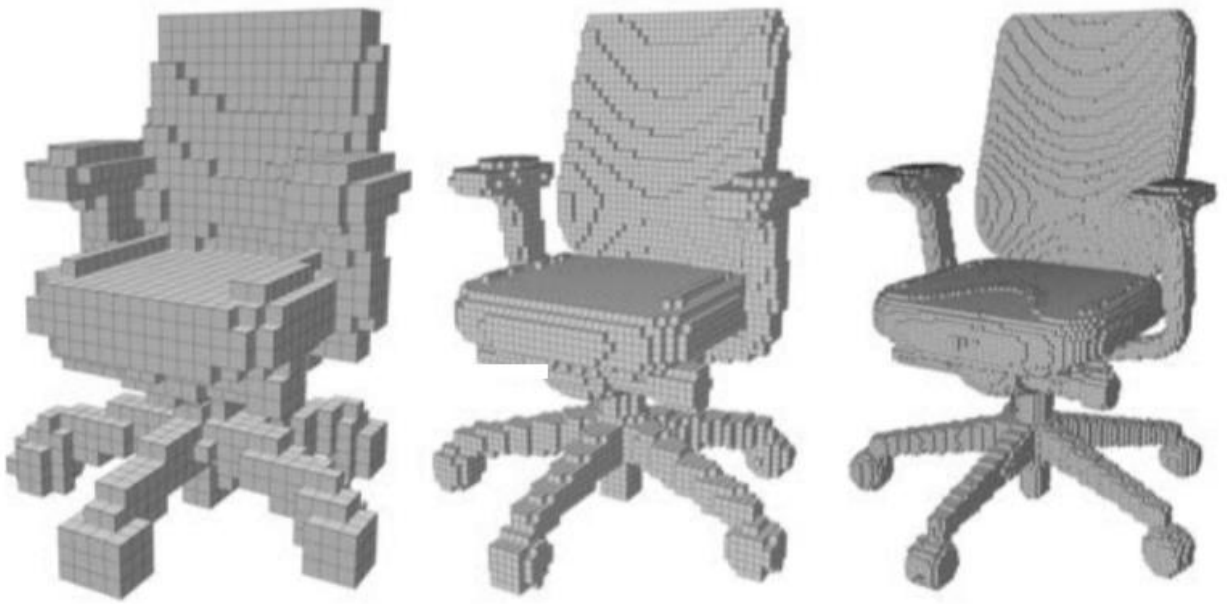




images from lecture note of university of Michigan

3D Shape representation

- Voxel grid
  - Trade-off between density and resolution

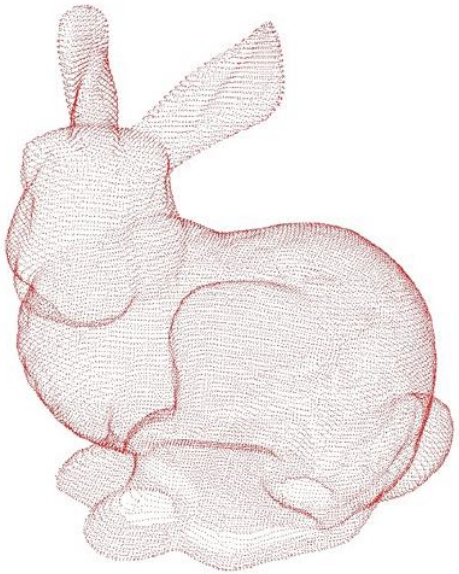


Density	10.41%	5.09%	2.41%
Resolution	32	64	128

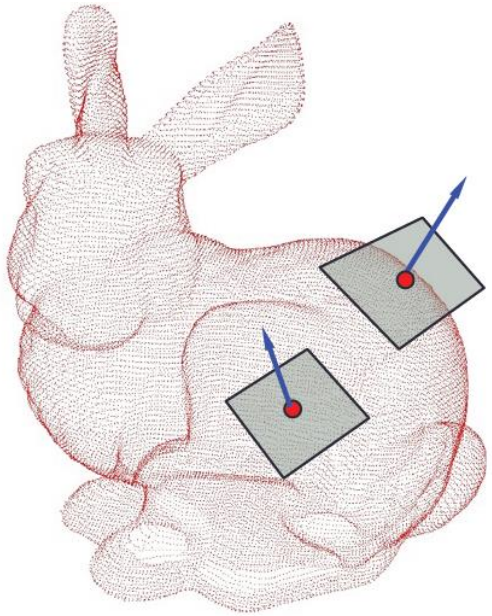
https://medium.com/vitalify-asia/create-3d-model-from-a-single-2d-image-in-pytorch-917aca00bb07

Point Clouds

- Simplest representation: **only points,** no connectivity.
- Type of data: Set, no order.
- Collection of (x,y,z) coordinates, possibly with normal of colors



(x, y, z)



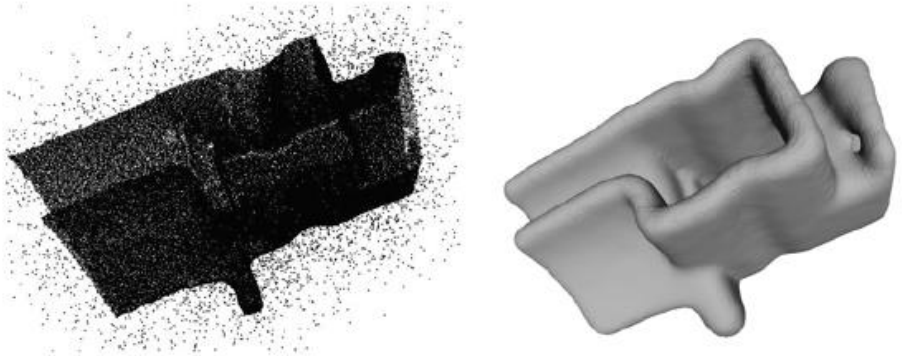
(x, y, z, nx, ny, nz)

(x, y, z, r, g, b)

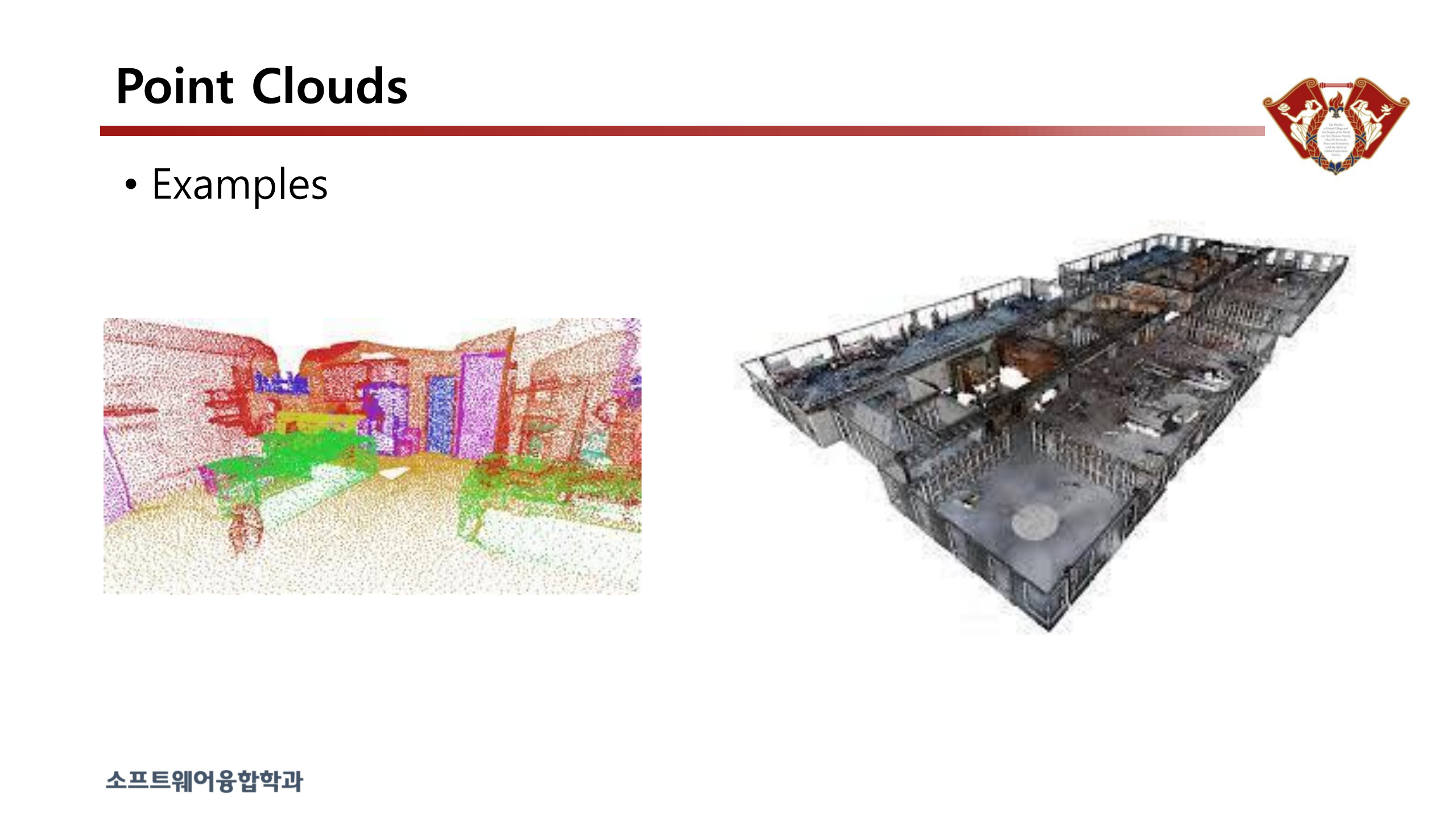
images from lecture note of Princeton university

Point Clouds

- Represent shape as a set of P points in 3D space
- Pros
  - Can represent fine structures
  - The simplest method which can represent 3D data
- Cons
  - Doesn't explicitly represent the surface of the shape
  - Weak approximation power.
  - Noise and outliers



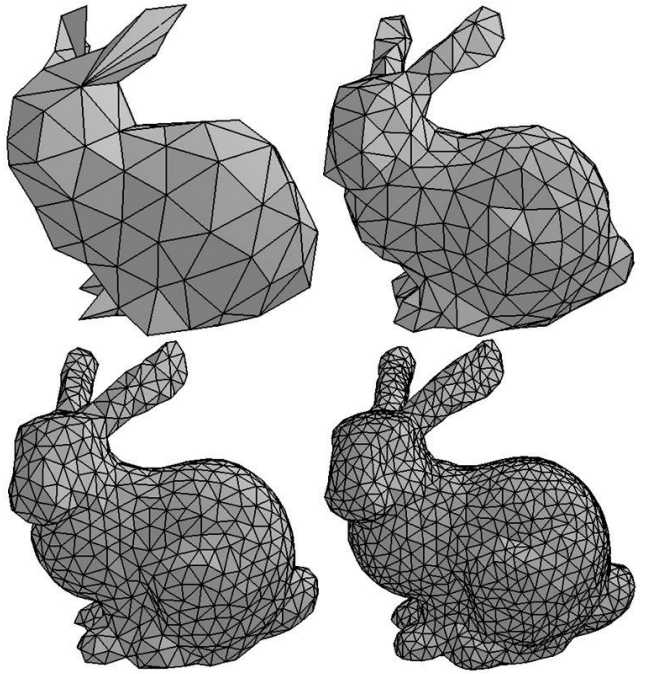
Point Clouds

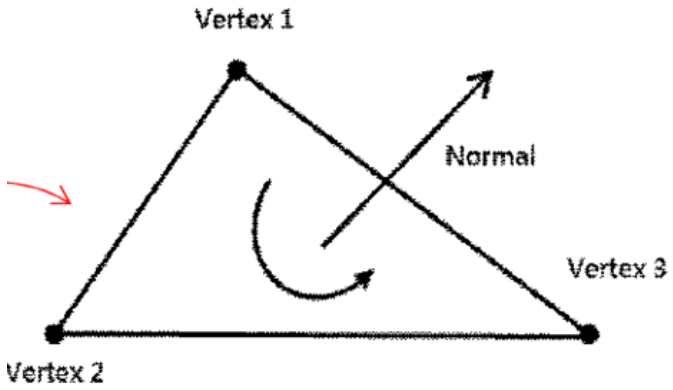


- Examples

Mesh

- Represent each manifold surface as a triangle mesh using graph
- Vectices
  - Location (x,y,z) → Same as point clouds.
- Edge
  - a line between two vertices and connects two faces
- Face
  - a surface created between the edges.
  - Triangle is widely used

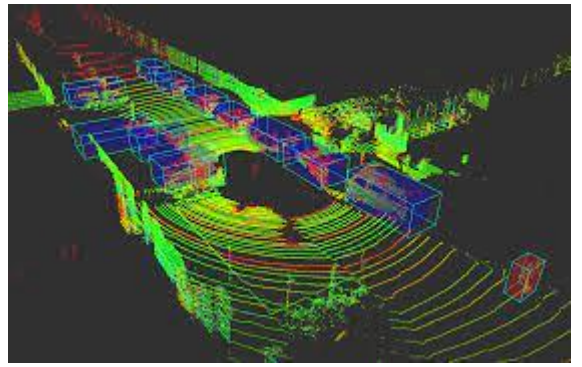


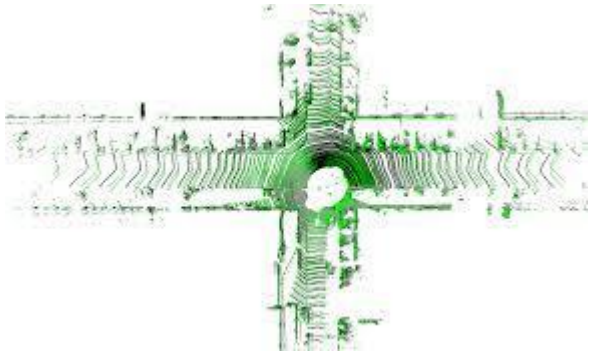


images from lecture note of Stanford university

Point Clouds generation

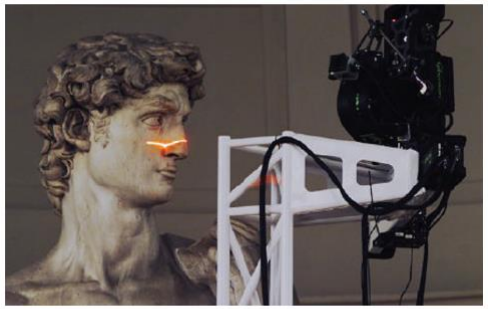
- Bird-eye View (BEW)
  - Represents data at a very far distance on the axis corresponding to height in the WORLD coordinate system.
  - 2-dimensional data
  - Images and 3D data can be represented as BEW





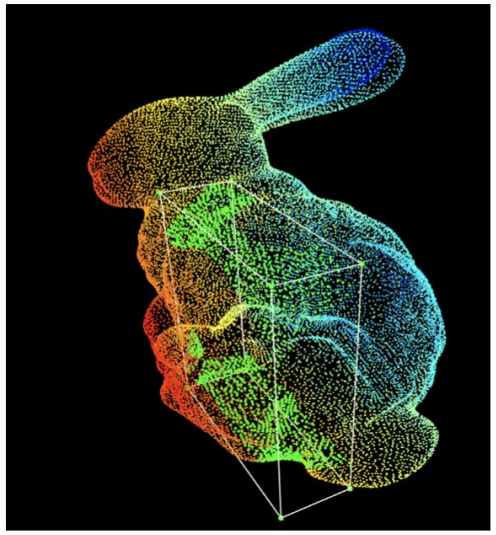
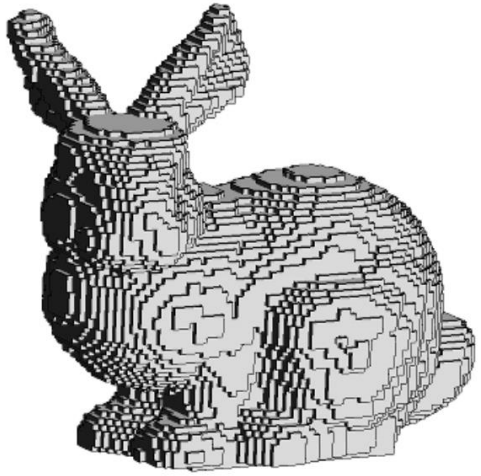
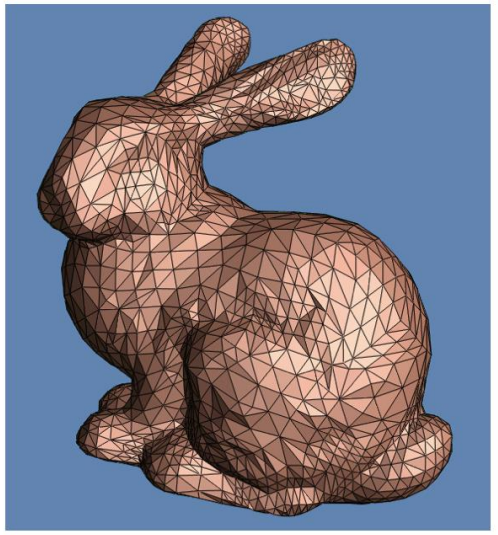
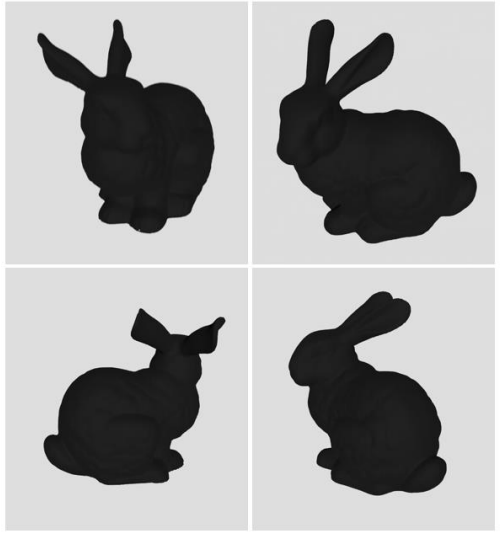
Why Point Clouds?

- Point clouds are easy to edit, filter and non-intrusive
  - There are many point cloud formats
  - There are many point cloud library (PCL, open3D, etc.)
- Typically, PC are a kind of standard
  - Nearly all 3d scanning devices produce point clouds
- Locality: sometimes, easier to handle, especially in hardware.



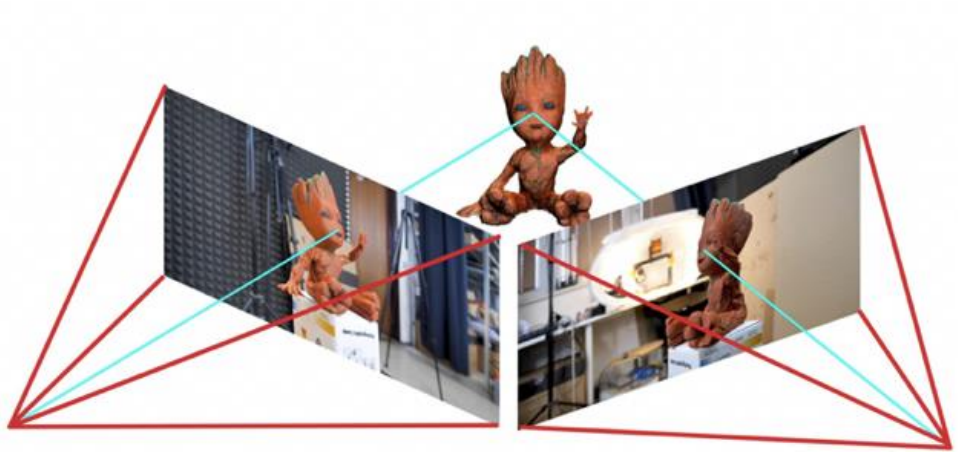
Why Point Clouds?

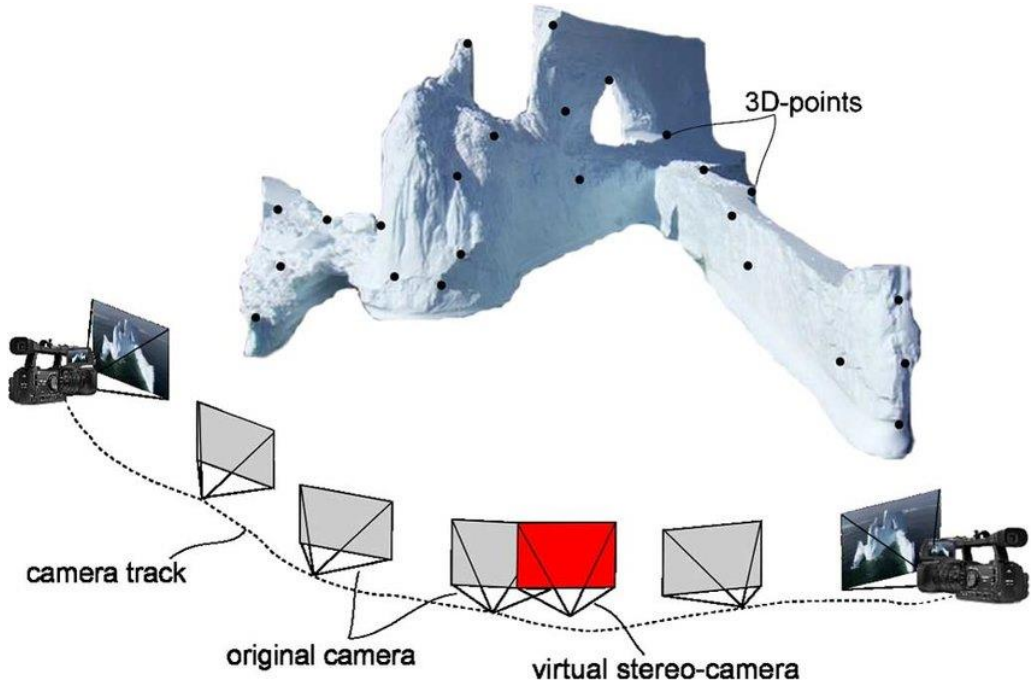
- Point clouds can be easily converted to other formats.
  - Convert To voxel by dividing sections in 3D
  - Convert To BEW by accumulate points to Z-axis
  - Convert to 3D Mesh by connecting edges using Delaunay triangulation

Point Clouds generation

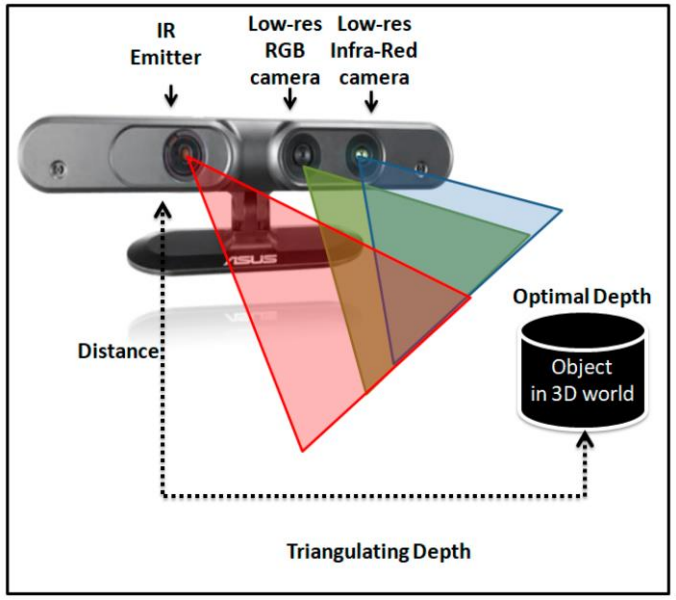
- Stereo or Multiview images
  - Images from two cameras
  - Very cheap
  - 3D Quality is lower (especially for texture-less scenes)

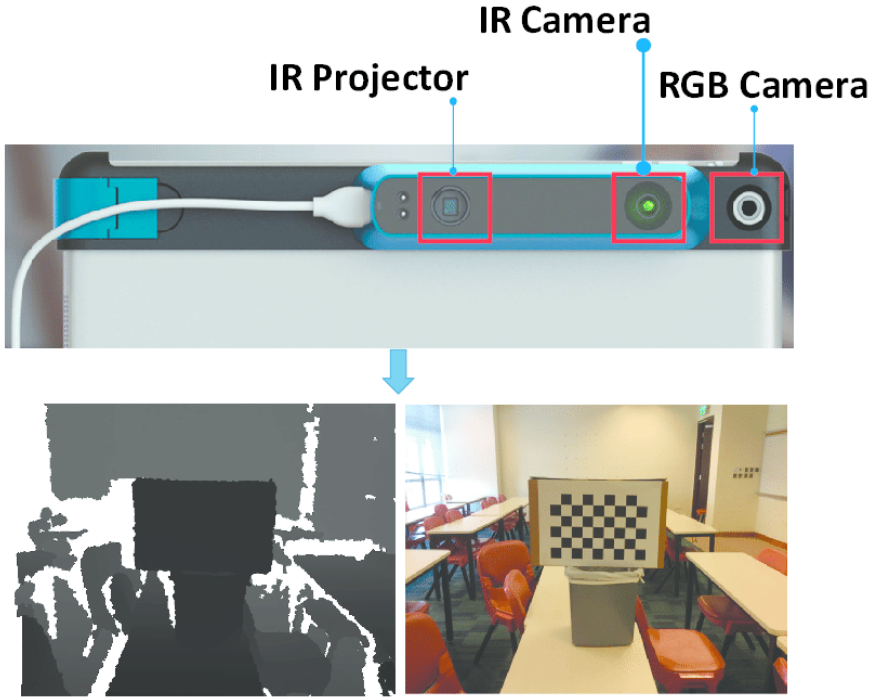




Point Clouds generation

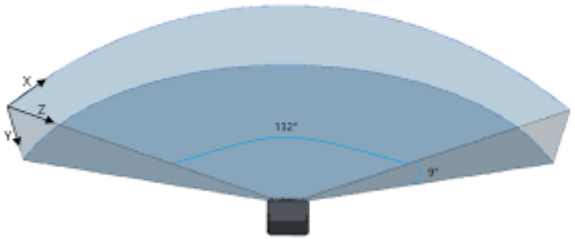
- RGB-D camera
  - RGB camera + depth camera
  - Depth camera produces dense depth corresponding RGB image
  - Triangulation method is the same with stereo camera
  - Difficult to use outdoors.

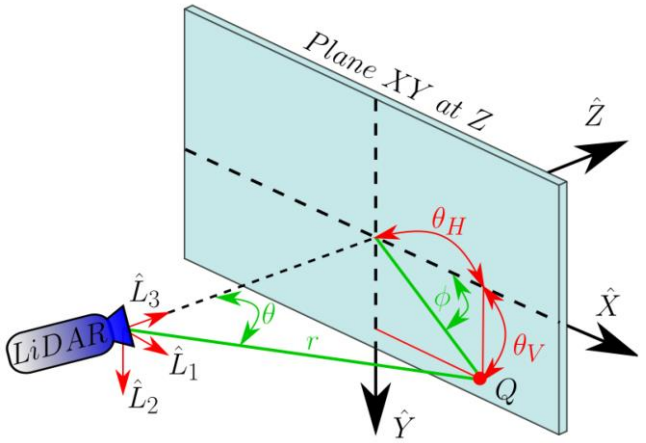


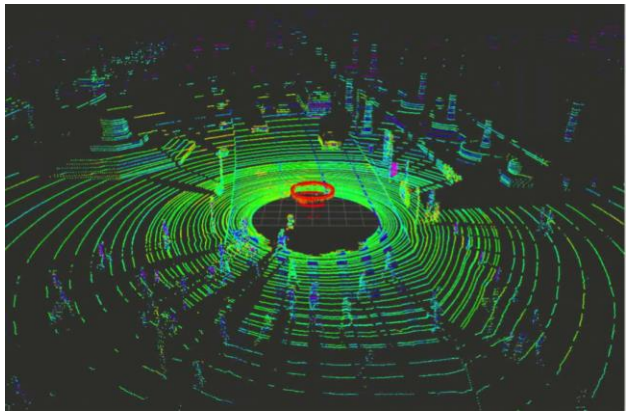


Point Clouds generation

- 3D Sensor
  - 3D Lidar
    - Horizontal range: 360 deg
    - Vertical range: up to 30 deg
  - Solid-state Lidar
    - FOV is similar with cameras or RGBD

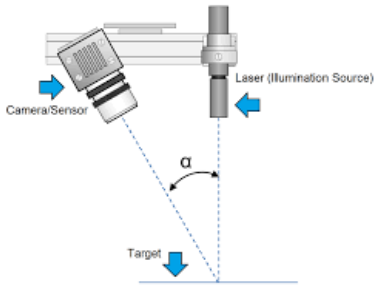




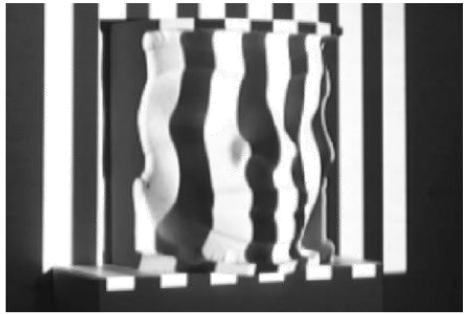


3D sensors

- Triangulation
  - Laser line sweep
  - Structured light
- Stereo / computer vision
  - Passive stereo
  - Active stereo
- Photometric 3D

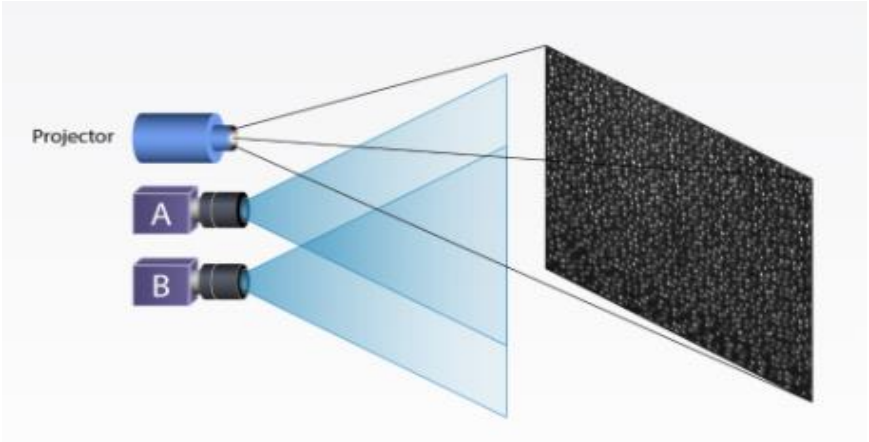


Line-sweep



Structured light

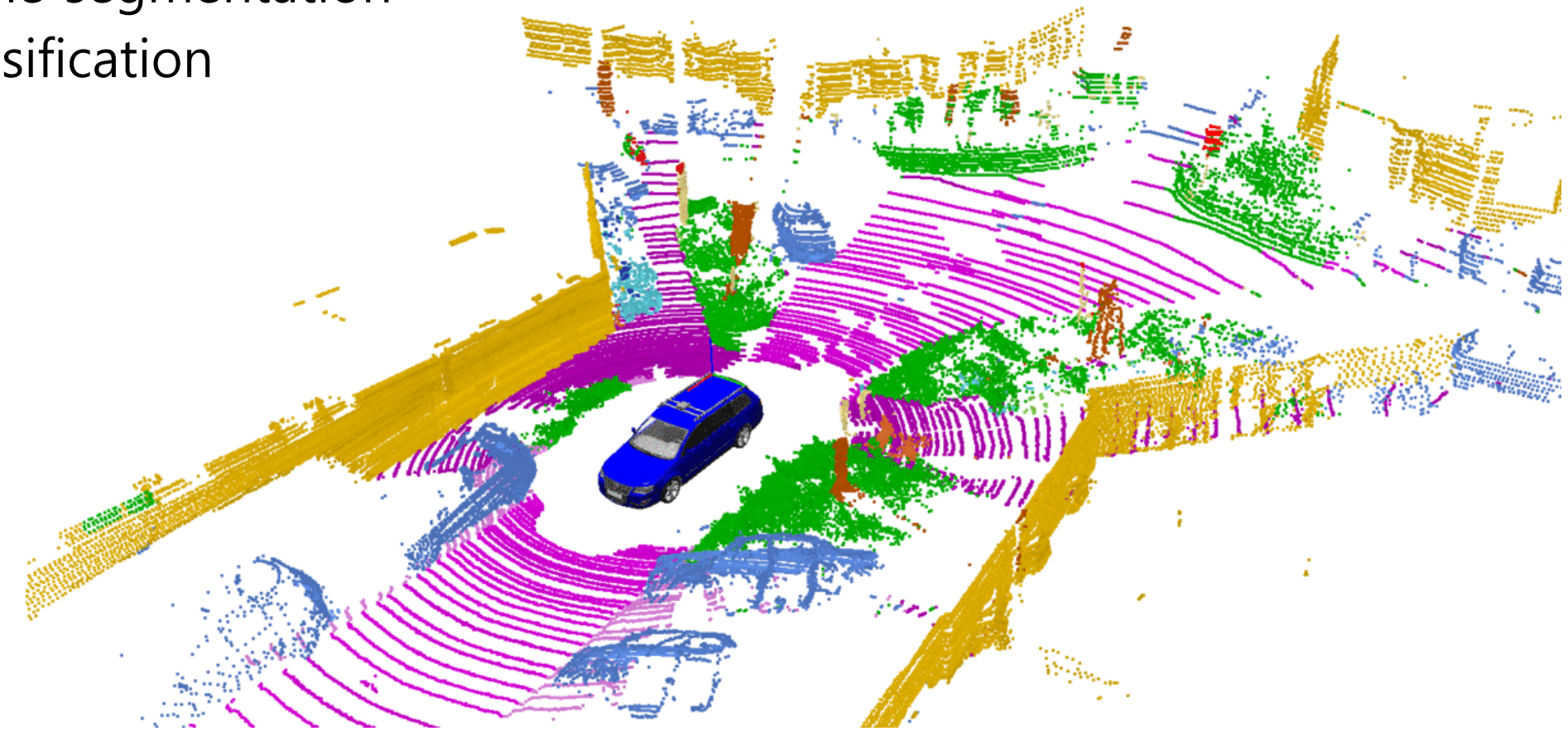
Photometry 3D



active stereo

Point Clouds application

- Autonomous driving
  - Scene segmentation
  - Classification

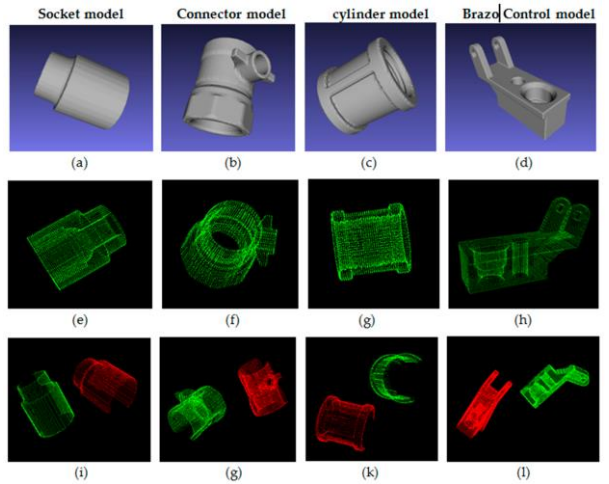


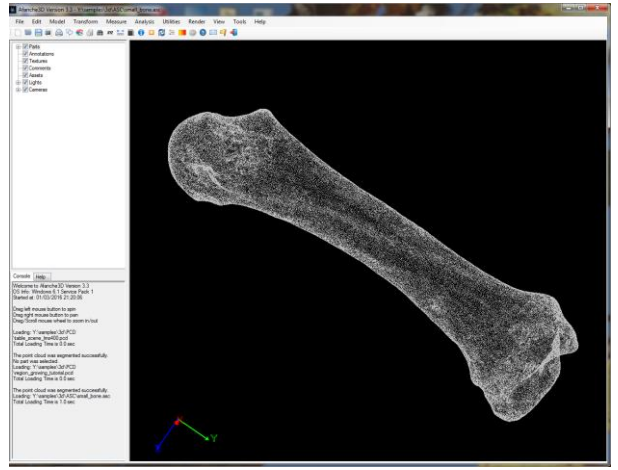
Point Clouds application

- 3D reconstruction
  - Registration

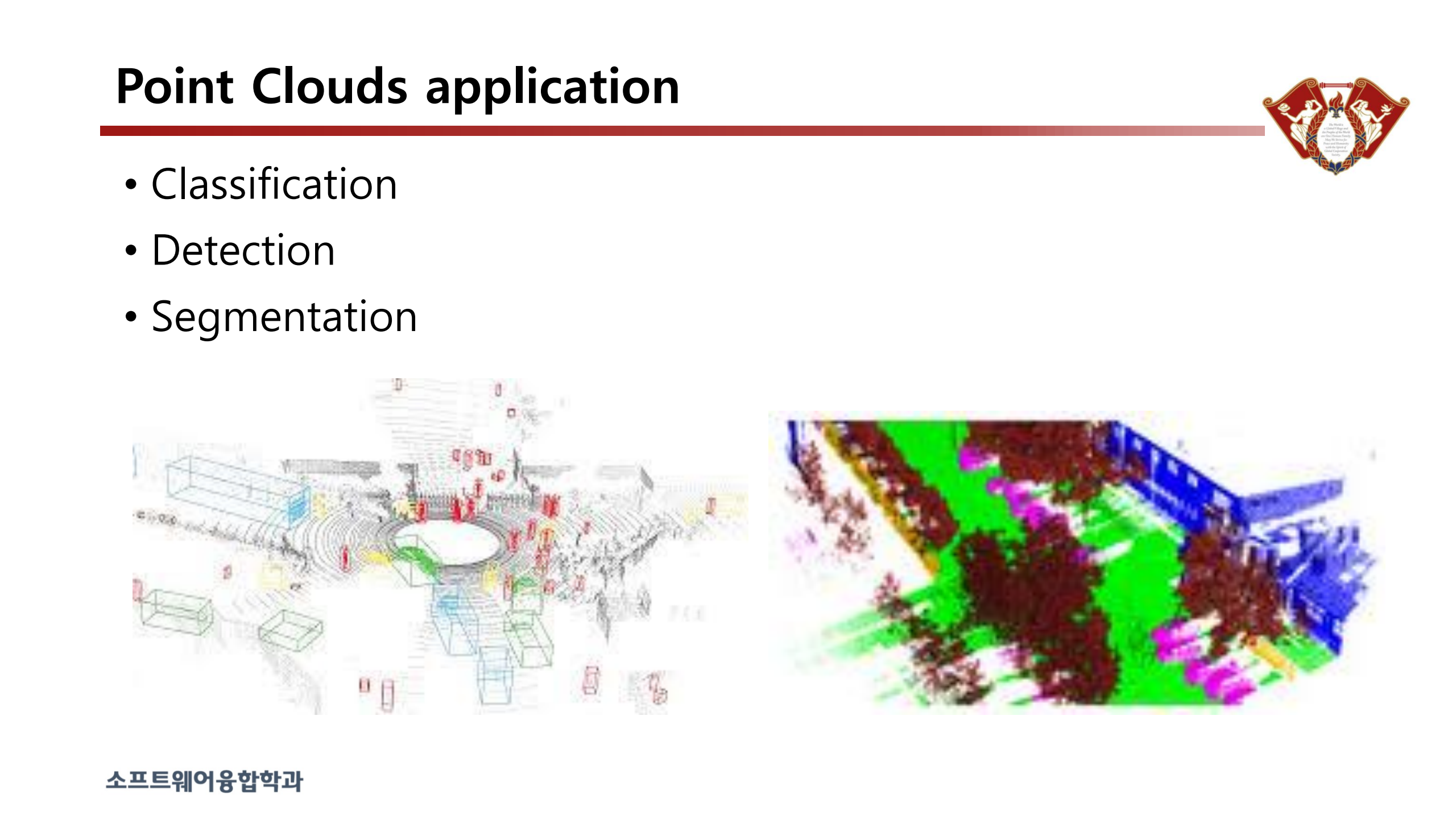
Point Clouds application

- Manufacturing:
  - One shape is a model and the other is a scan of a product. Finding defects.
- Medicine:
  - Finding correspondences between 3D MRI scans of the same person or different people.
- Animation Reconstruction & 3D Video.
- Statistical Shape Analysis:
  - Building models for a collection of shapes.





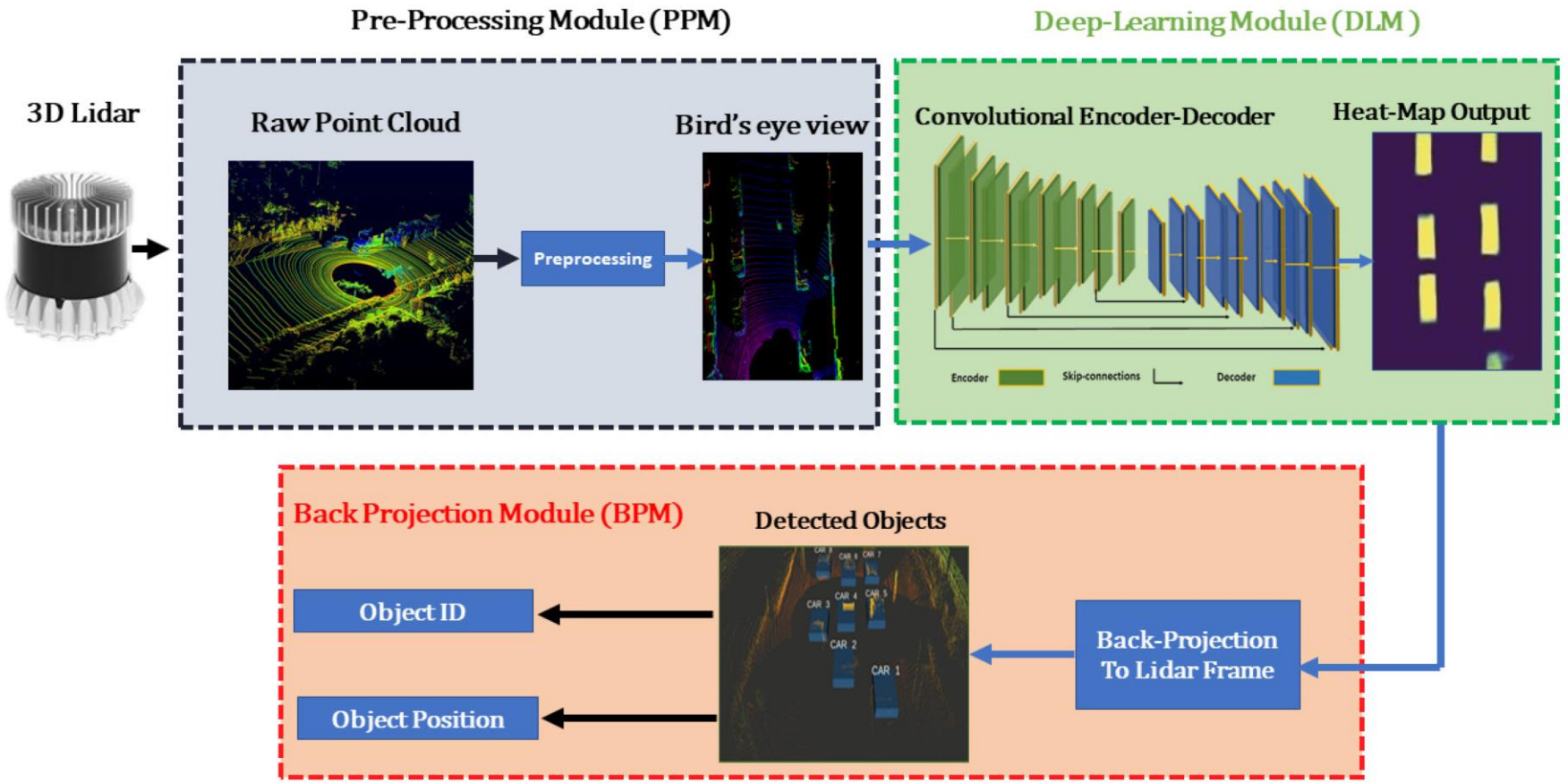
Point Clouds application



- Classification
- Detection
- Segmentation

Point Clouds application

- Deep neural Networks



Point Clouds Processing

- PC Generation
- Preprocessing
  - Downsampling
  - Outlier removal
  - Cropping
- Segmentation
  - Plane segmentation
  - Object segmentation
  - Clustering
- Features
  - 3D Keypoint
- Registration

point_cloud.txt - Notepad

File Edit Format View Help

% x	y	z
9.117608087	6.815297175	0.378845387
8.985479962	7.089485484	0.451241969
8.781697931	6.784964094	0.407603478
8.836222603	7.35231719	0.53203334
8.660284039	7.06351441	0.489312327
8.670406076	7.603482155	0.617530684
8.518808049	7.289897821	0.570696796
8.488567953	7.842515578	0.705210656
8.353948192	7.532774085	0.665161915
8.291240988	8.068832509	0.793146159
8.179953082	7.747238083	0.755795569
9.231862679	6.530143116	0.319705851
8.904464051	6.545443525	0.346901894
9.327240674	6.234904095	0.279987613
8.999700451	6.262175803	0.299785836
8.59752683	6.537127276	0.390210886
8.50244736	6.768957169	0.451631462
8.37291744	7.024273623	0.535931277
8.239235691	7.225828938	0.615570611
8.074018623	7.468059581	0.716101201
8.078870134	8.28185297	0.879226414
7.990868345	7.933684133	0.842761364
7.907365874	7.625395145	0.796326771
7.851811739	8.480972752	0.960971605
7.78398438	8.153045012	0.936346766
7.610374237	8.665507914	1.035379618
7.568719244	8.318889106	1.015671964
7.354943818	8.834609545	1.098883228
7.333090803	8.490274928	1.090455056
9.402489043	5.931713268	0.263948885
9.080546334	5.996126139	0.277108328
9.456583565	5.623587512	0.265265737
9.130251655	5.689444728	0.272770097
9.489131691	5.312563785	0.27468382
9.1778898	5.401368057	0.278315971

Thank you