Learning from 3D (Point Cloud) Data

– a Tutorial for ACM Multimedia 2019

Winston H. Hsu (徐宏民)
Professor, Communication & Multimedia Lab (CMLab)
Director, NVIDIA AI Lab
National Taiwan University

October 21, 2019

Communication and Multimedia Lab (通訊與多媒體實驗室)
http://winstonhsu.info

tutorial slides

Shot Bio

- **Education**
  - PhD in EE, Columbia University, New York
- **Expertise**
  - Machine learning
  - Large-scale visual retrieval and recognition
- **Experience**
  - Professor, CS Department, National Taiwan University
  - Founding Director, NVIDIA AI Lab (NTU)
  - Technical Committee for National Science Park, Taiwan
  - Co-founder, thingnario (慧景科技), AI for energy & manufacturing
  - Visiting Scientist, IBM TJ Watson Research Center (2016-2017)
  - Visiting Researcher, Microsoft Research Redmond (2014)
  - CyberLink Corp.(訊連科技) Founding Engineer, R&D Manager
  - Editorial Board for IEEE Multimedia Magazine (2010 – 2017)
  - Associate Editor for two premier journals, TCSVT and TMM

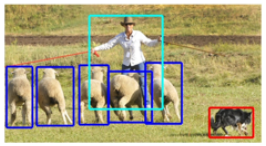
Prior Tutorial Experiences in ACM Multimedia

- Recent Developments in Content-based and Concept-based Image/Video Retrieval
  – Winston Hsu and Rong Yan

- Content-based and Concept-based Analysis for Large-Scale Image/Video Retrieval
  – Rong Yan and Winston Hsu

Super-Human CNN Capabilities over 2D Images

image classification



object localization

pose estimation

face recognition

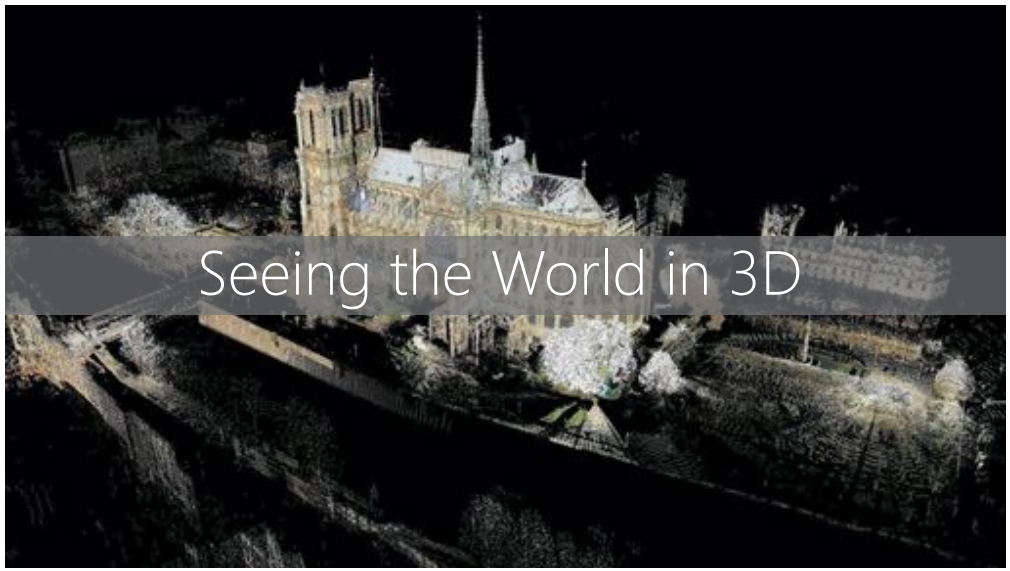
semantic segmentation

instance segmentation

video recognition

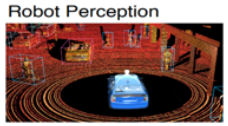
image generation

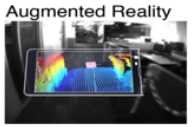
 Lin et al. Microsoft COCO: Common Objects in Context. ECCV 2014

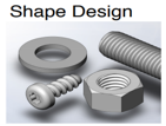


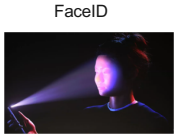
Revolutionizing 3D Data

- Robot perception
- Augmented reality
- Shape design
- Identity recognition





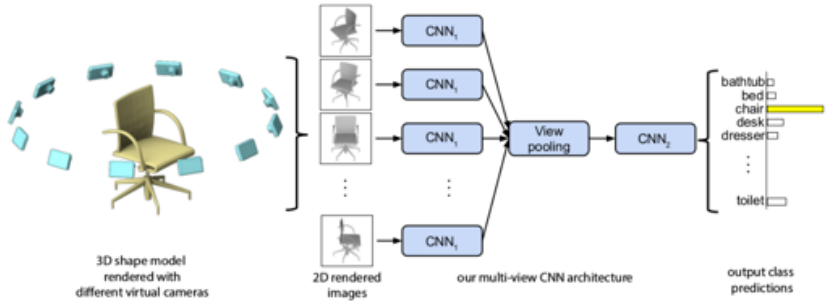




http://stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf

Challenge: 2D Operations for 3D Data (Multiple Views), MVCNN as the Early Solution

- 2D CNN outperforms prior 3D (hand-crafted) shape descriptors.
- Multiple views are helpful

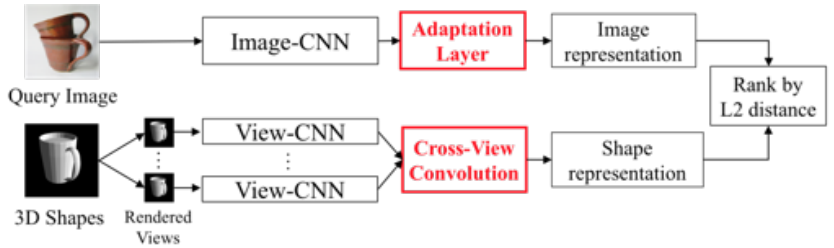


Su, Hang, et al. Multi-view convolutional neural networks for 3d shape recognition. ICCV 2015.

Image-based 3D Shape Retrieval for Productivity

demo

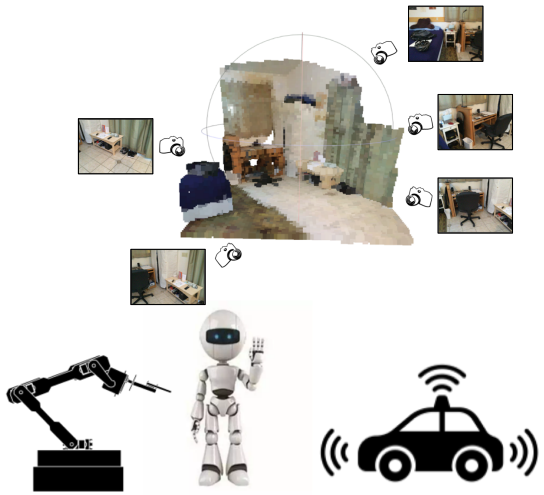
[Lee et al., 3D Vision 2018]



Lee et al. Cross-Domain Image-Based 3D Shape Retrieval by View Sequence Learning. 3DV 2018

Outline

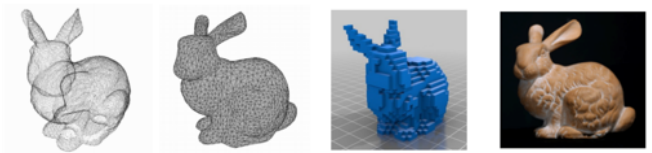
- Introduction
- 3D Sensors & Data Types
- Stereo Vision
- Point Cloud Learning Algorithms
- LiDAR Object Detection
- 3D Face Recognition
- 3D Robotic Grasp Detection
- 3D Quality Enhancement



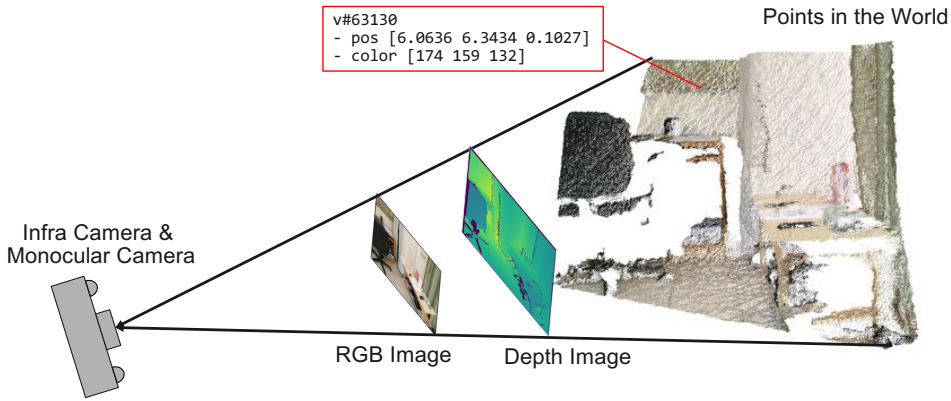
3D Sensors & Data Types

Rich 3D Representations

- Point Cloud
- Mesh
- Volumetric (in voxels)
- Projected View (RGB-D)



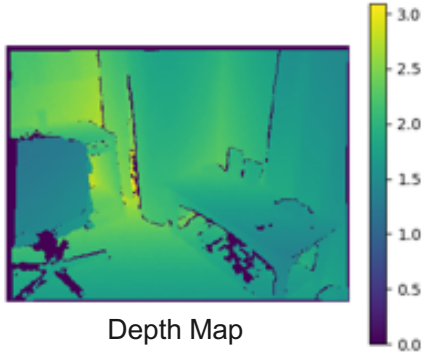
RGB vs. Depth vs. Point Clouds



2.5D RGB-D Depth Map

- Often used for CNN as 4 channel inputs (RGB + Depth) → limited performance

Monocular RGB Image

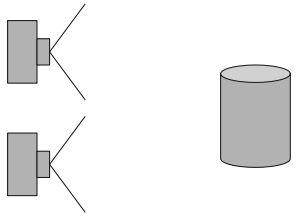


Depth Map

3D Point Clouds (Sampled in an Indoor Scene)

3D Data from Stereo Vision

- Use images of 2 cameras to estimate distance (**more discussions later**)
- Pros
  - Cheap device
  - Easy to install
- Cons
  - **Accuracy drops with distance**
    - disparity error magnified for the far-way regions
    - ambiguous for a big are of the same texture (cloud, sand, sky, wall, etc.)
  - Requiring additional computation
  - **Sensitive to lighting conditions**

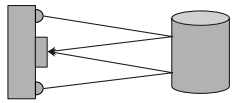


F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles", Sensors, vol. 17, no. 2, pp. 271, 2017.

3D Data from Time-of-Flight (TOF) Camera

- Measures time of flight of infrared light
- Captures the entire scene with one single light pulse
- Low resolution, short range
- **Sensitive to lighting**, color
  - Suitable for indoor environment

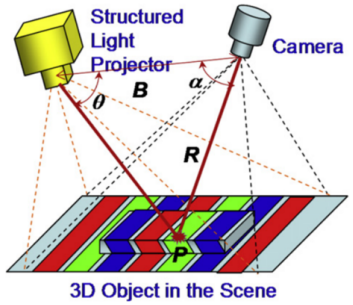
Sensor	Resolution	Range	Azimuth Angle	Accuracy	Cycle
PMD CamBoard	200 × 200	7 m	40°	-, -, -	60 fps
PMD CamCube	200 × 200	-	-	-, -, -	-
SwissRanger SR4000	176 × 144	10 m	40°	1 cm, -, -	50 fps



F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles", Sensors, vol. 17, no. 2, pp. 271, 2017.

3D Data from Structure Light Camera

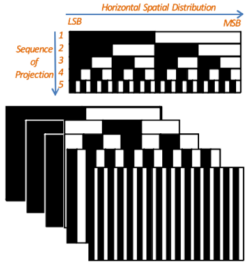
- The key idea of structure light is **triangulation,** used to calculate depth.



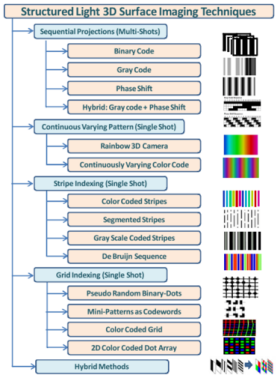
Jason Geng. Structured-light 3D surface imaging: a tutorial. 2011.
Lanman et al. Build Your Own 3D Scanner: 3D Photography for Beginners. ACM SIGGRAPH 2009.

3D Data from Structure Light Camera (cont.)

- Encoding pattern – usually combining with information theory technique



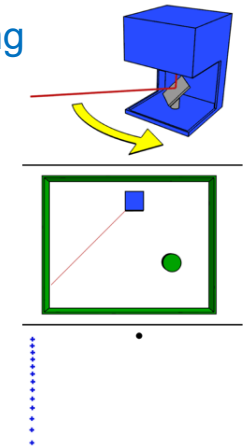
Sequential binary-coded pattern projections for 3D imaging.



Jason Geng. Structured-light 3D surface imaging: a tutorial. 2011.
Lanman et al. Build Your Own 3D Scanner: 3D Photography for Beginners. ACM SIGGRAPH 2009.

3D Data from Light Detection And Ranging (LiDAR)

- Measures time of flight of pulsed laser beam
- Usually with rotation parts
- Long range, high resolution, **high price**
- Sensitive to environment, e.g., weather
  - Commonly used for self-driving car

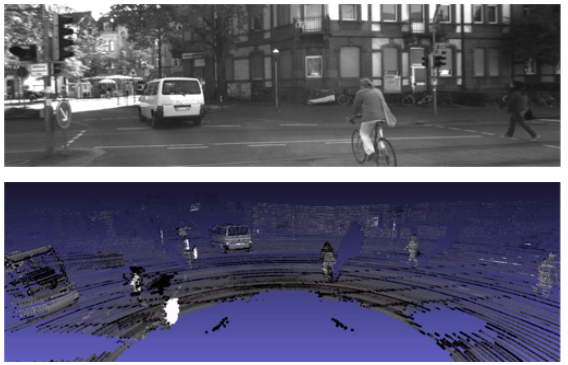


Sensor	Dimensional Resolution	Range	Azimuth Angle	Accuracy	Cycle
Quanergy M8-1	3D	150 m	360°	0.05 m, -, 0.03°	33 ms
Ibeo LUX	2D	200 m	110°	0.1 m, -, 0.125°	20 ms
Continental SRL1	2D	10 m	27°	0.1 m, 0.5 m/s, -	10 ms
Velodyne HDL-64E S2	3D	120 m	360°	0.02 m, -, 0.09°	50 ms

F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles", Sensors, vol. 17, no. 2, pp. 271, 2017.

Viewing the Environment in 3D (LiDAR)

- About 100K points per frame
- Each point with
  - x
  - y
  - z (d)
  - intensity (reflectance)
- **Sparse & occluded**
- Some labels provided in KITTI dataset
  - 3D box label of cars, pedestrians, cyclists, ...



Radio Detection And Ranging (RADAR)

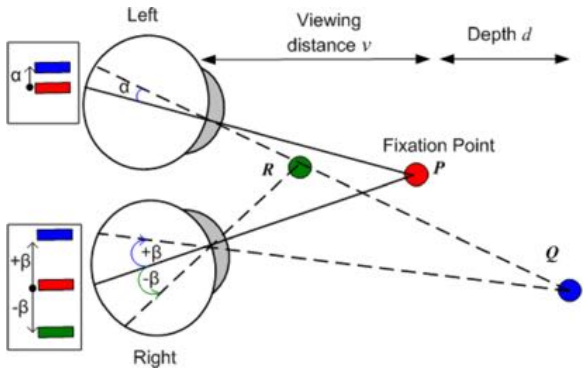
- Using high-frequency electromagnetic waves
- **No lens**
- Robust against environmental conditions
  - Usually used for **obstacle detection** and **speed measurement**

Sensor	Frequency	Bandwidth	Range	Azimuth Angle	Accuracy	Cycle
Bosch LRR3	77 GHz	1 GHz	250 m	±15°	0.1 m, 0.12 m s^{-1}, -	80 ms
Delphi ESR	77 GHz	-	174 m	±10°	1.8 m, 0.12 m s^{-1}, -	50 ms
Continental ARS30x	77 GHz	1 GHz	250 m	±8.5°	1.5 %, 0.14 m s^{-1}, 0.1°	66 ms
SMS UMRR Type 40	24 GHz	250 MHz	250 m	±18°	2.5 %, 0.28 m s^{-1}, -	79 ms
TRW AC100	24 GHz	100 MHz	150 m	±8°	-, -, 0.5°	-

F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles", Sensors, vol. 17, no. 2, pp. 271, 2017.

Stereo Vision

Human Stereopsis (Angular Disparity)



Binocular disparity provided by the two eyes' different positions on the head gives **precise depth perception** !

Silva, et al. Display dependent preprocessing of depth maps based on just noticeable depth difference modeling. IEEE Journal of Selected Topics in Signal Processing 2011

Stereo Camera

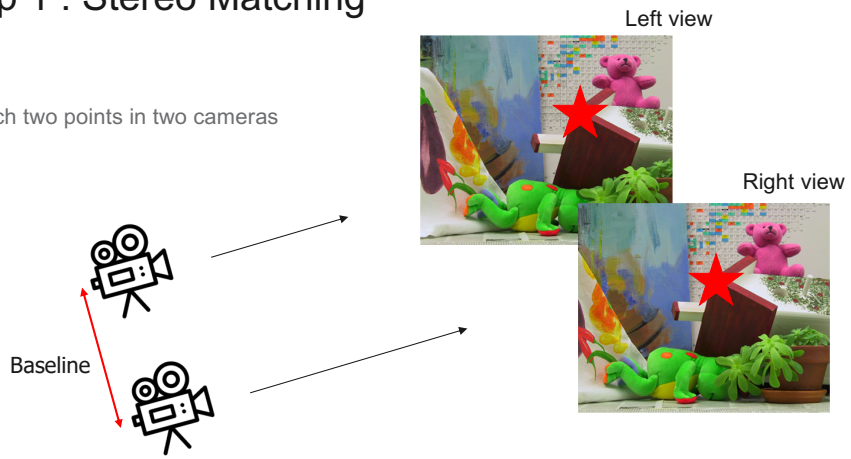
Vérascope 40

ZED

*Figures from https://en.wikipedia.org/wiki/Stereo_camera
https://www.stereolabs.com/zed/

Step 1 : Stereo Matching

- Match two points in two cameras



*Figures from https://www.flaticon.com/free-icon/video-camera_263068#term=camera&page=1&position=7

Step 2 : Disparity Estimation

- Disparity = horizontal displacement of corresponding points in the two images
- Disparity of ★ $D = x_l - x_r$

http://media.ee.ntu.edu.tw/courses/cv/18F/slides/cv2018_lec14.pdf

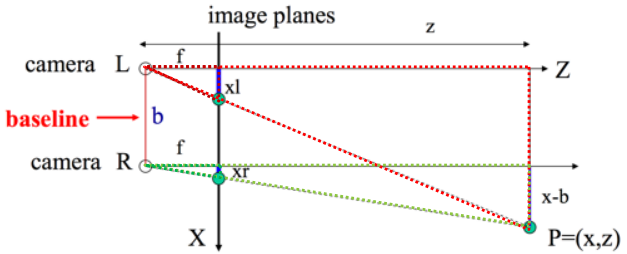
Step 3 : Disparity to Depth

For a single point $P(x, z)$ in the left image : D (disparity) = $x_l - x_r$

$$\frac{f}{z} = \frac{x_l}{x} = \frac{x_r}{x-b}$$

$$x = \frac{bx_l}{x_l - x_r} = \frac{bx_l}{D}$$

$$z = \frac{fx}{x_l} = \frac{f \cdot bx_l}{D \cdot x_l} = \frac{fb}{D}$$



https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16.pdf

Stereo to Disparity Datasets

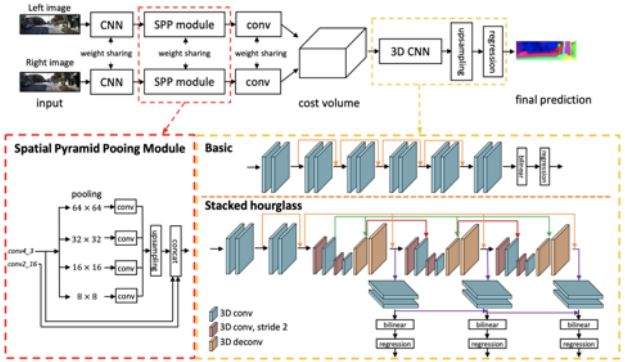
Middlebury Stereo Dataset (2001, 2003, 2005, 2006, 2014)

SceneFlow dataset

KITTI 2012, 2015

PSMNet for Disparity Regression

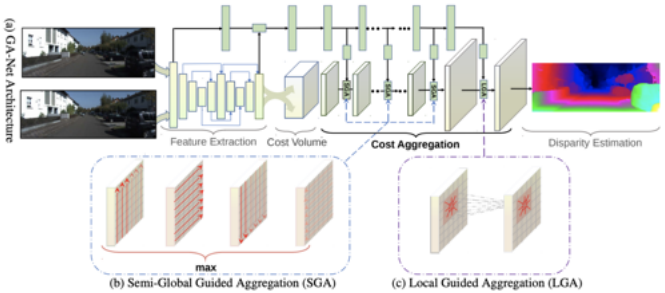
- PSMNet proposed to stack hourglass (encoder-decoder) network architecture that is useful and effective for disparity regression



Chang et al. Pyramid Stereo Matching Network. CVPR 2018

GA-Net: Guided Aggregation Net for End-to-End Stereo Matching

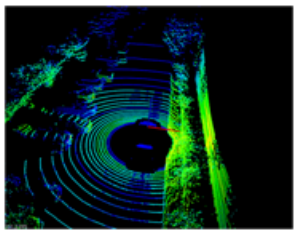
- Realizing traditional semi-global matching as end-to-end backpropagation model and largely reducing computation and memory-consumption while retaining performance gains
- State of the art on KITTI stereo dataset

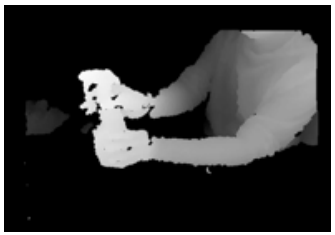


Zhang et al. GA-Net: Guided Aggregation Net for End-to-end Stereo Matching. CVPR 2019

Point Cloud Learning Algorithms

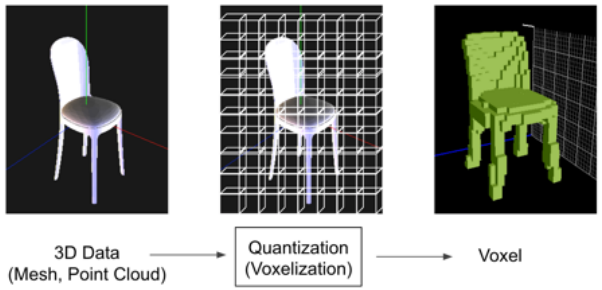
Learning Directly on 3D Data


LiDAR


RGB-D


CAD

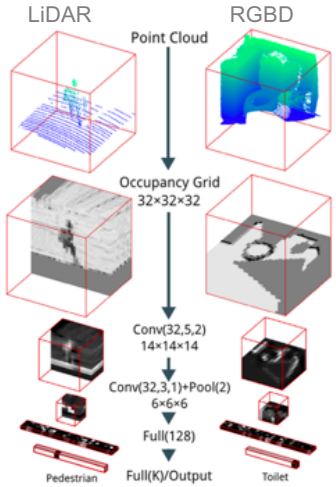
Occupancy Grid (Voxel, Volumetric)



Maturana et al. 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR. ICRA. 2015.
Maturana et al. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IROS. 2015.

VoxNet

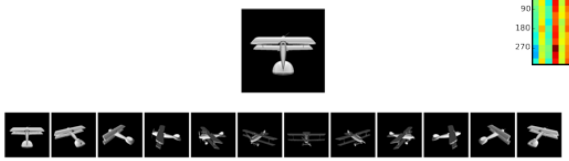
- 3D object recognition via **occupancy grid**
  - Binary grid
  - Density grid
  - Hit grid
- Why?
  - Most prior work utilizes **2.5D** (4 channels) and does not utilize geometric information
  - **Point clouds** (mostly) require time -consuming spatial neighborhood queries
- Problem – small 3D grids (30^3)

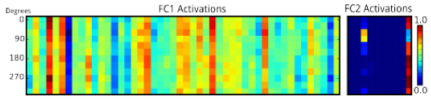


Maturana et al. 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR. ICRA. 2015.
Maturana et al. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IROS. 2015.

VoxNet

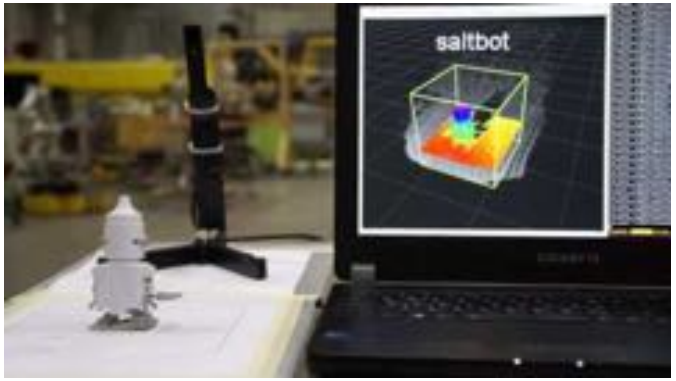
- Machine learns **rotation Invariant**
  - Nontrivial to maintain a consistent orientation of objects
  - Augmenting the dataset by creating copies (12) of each input instance – each rotated around the z axis
    - Training and testing (by pooling)





Maturana et al. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IROS. 2015.

VoxNet (Online Demo)

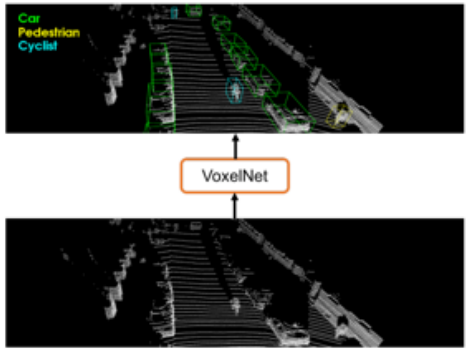


https://www.youtube.com/watch?v=KAB11FrQz_Q

Maturana et al. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IROS. 2015.

VoxelNet

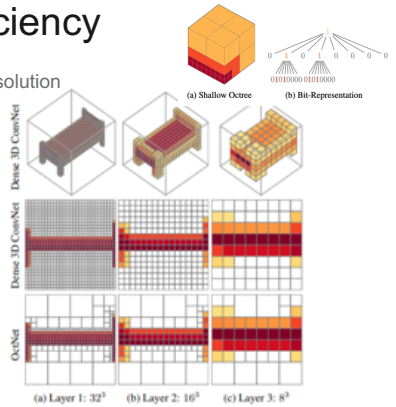
- 3D Object Detection via occupancy grid for LiDAR
- Explained later in LiDAR section



Zhou. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. CVPR 2018.

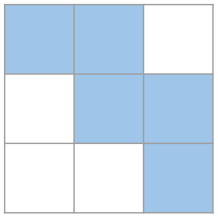
More Volumetric Methods – Memory & Computation Efficiency

multiple resolution



Riegler et al. OctNet, CVPR 2017

- sparse representation
- solving sparsity dialation problem



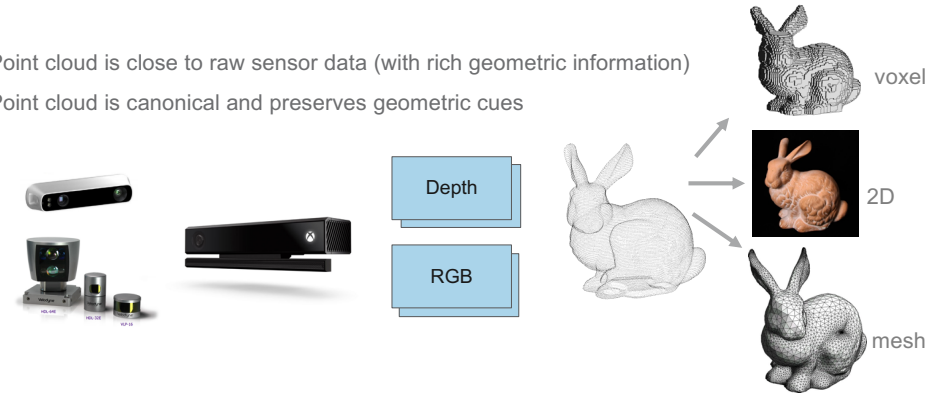
```
[0, 0]
[0, 1]
[1, 1]
[2, 1]
[2, 2]
```

Sparse 3D representation, such as

Graham et al. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks, CVPR 2018

Point Clouds

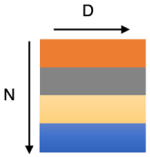
- Point cloud is close to raw sensor data (with rich geometric information)
- Point cloud is canonical and preserves geometric cues

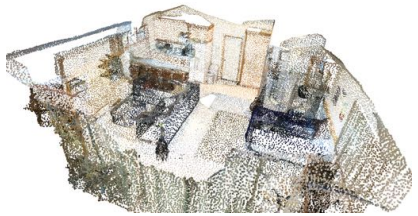


Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 2017

The Challenge of Point Clouds

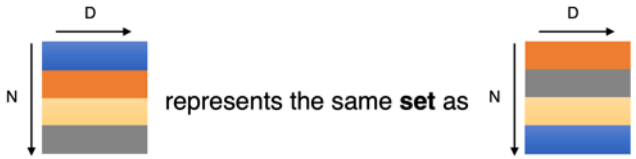
- Unordered point set as the input
  - Model needs to be invariant to ***N!*** Permutations
- Invariance under **geometric** transformations
  - Point cloud rotations should not alter classification results





Permutation Invariant

- Point cloud: N **orderless** points, each represented by a D-dim vector



Model needs to be invariant to $N!$ permutations

Permutation Invariant by Symmetric Functions

- Symmetric functions

$$f(x_1,x_2,\ldots,x_n) \equiv f(x_{\pi_1},x_{\pi_2},\ldots,x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

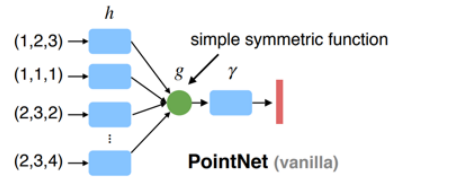
Examples:

$$f(x_1,x_2,\ldots,x_n) = \max\{x_1,x_2,\ldots,x_n\}$$
$$f(x_1,x_2,\ldots,x_n) = x_1 + x_2 + \ldots + x_n$$
$$\ldots$$

Permutation Invariant by Symmetric Functions

- **PointNet** – highly robust to small perturbation of input points as well as to corruption through point insertion (outliers) or deletion (missing data) (via theoretical analysis).

$f(x_1,x_2,\ldots,x_n) = \gamma \circ g(h(x_1),\ldots,h(x_n))$ is symmetric if g is symmetric

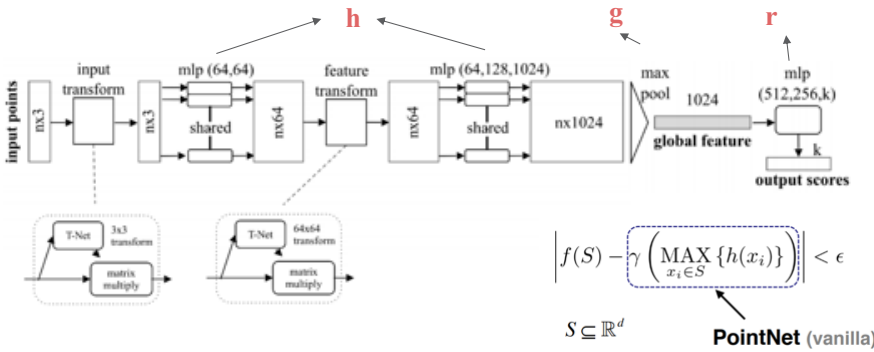


$$\left| f(S) - \gamma \left(\underset{x_i \in S}{\text{MAX}} \{h(x_i)\} \right) \right| < \epsilon$$

$S \subseteq \mathbb{R}^d$

PointNet (vanilla)

PointNet Architecture (Classification)

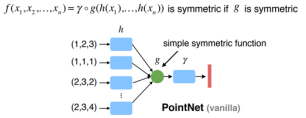


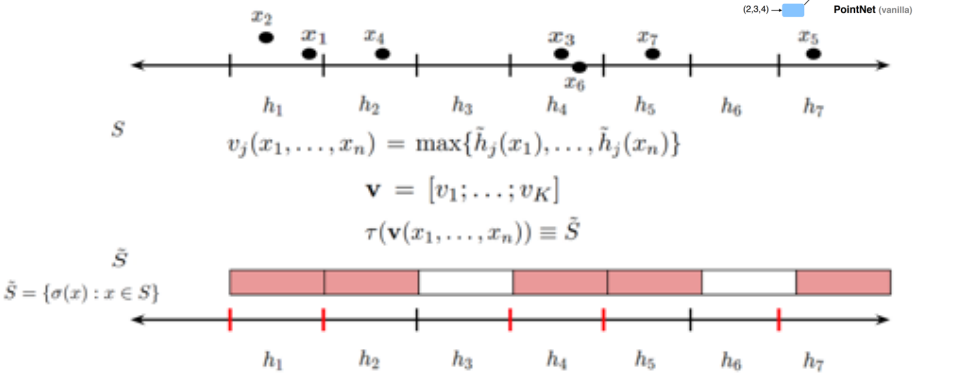
$$\left| f(S) - \gamma \left(\underset{x_i \in S}{\text{MAX}} \{h(x_i)\} \right) \right| < \epsilon$$

$S \subseteq \mathbb{R}^d$

PointNet (vanilla)

Intuition for Proof

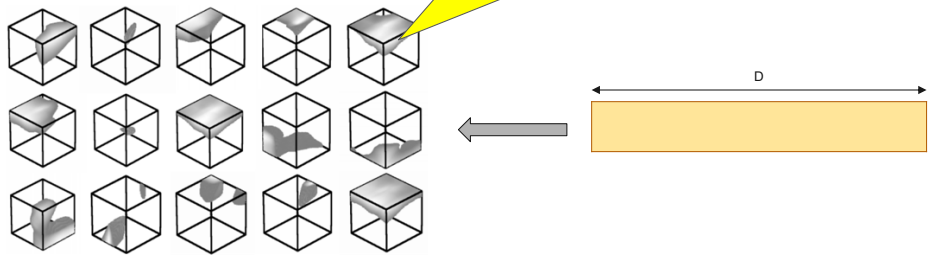




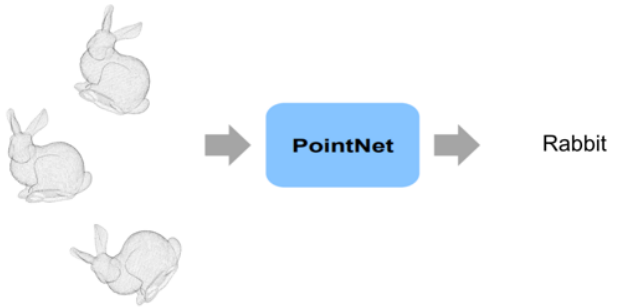
PointNet-Based Methods

- PointNet
  - Randomly pick 15 point functions
  - Visualize the activation regions

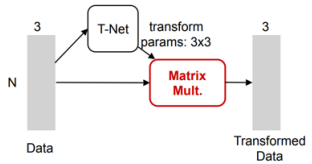
non-uniform voxel

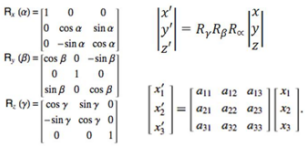


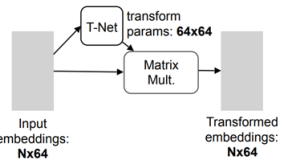
Transformation Invariant



Transformation Invariant





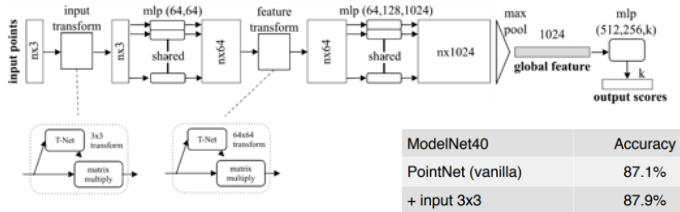


Regularization:

Transform matrix A 64x64 close to orthogonal:

$$L_{reg} = \|I - AA^T\|_F^2$$

PointNet Results (Classification with Transformation)



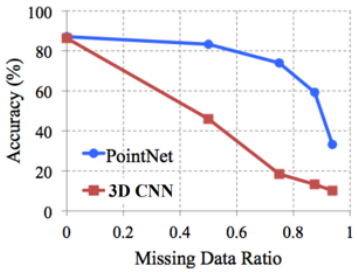
ModelNet40	Accuracy
PointNet (vanilla)	87.1%
+ input 3x3	87.9%
+ feature 64x64	86.9%
+ feature 64x64 + reg	87.4%
+ both	**89.2%**

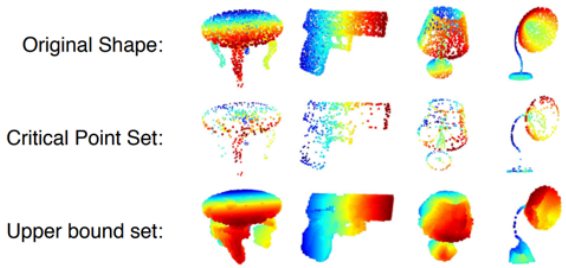
PointNet Results (Classification)

		input	#views	accuracy avg. class	accuracy overall
	SPH [12]	mesh	-	68.2	
3D CNNs	3DShapeNets [29]	volume	1	77.3	84.7
3D CNNs	VoxNet [18]	volume	12	83.0	85.9
3D CNNs	Subvolume [19]	volume	20	86.0	**89.2**
	LFD [29]	image	10	75.5	-
	MVCNN [24]	image	80	**90.1**	-
	Ours baseline	point	-	72.6	77.4
	Ours PointNet	point	1	86.2	**89.2**

dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption

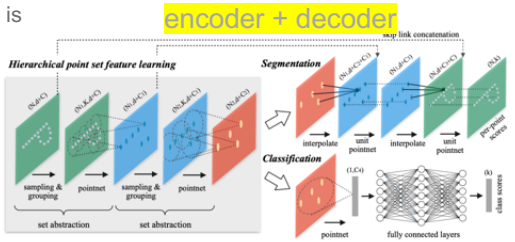




any point cloud that falls between the **critical points set** and the **upper bound shape** gives exactly the same feature.

PointNet++ – An Improvement

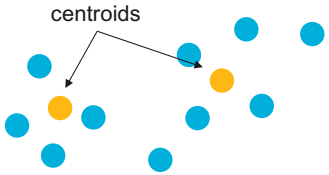
- Original PointNet lacks hierarchical feature learning like modern CNN
  - Does not capture local structures
- Leveraging multi-scales with PointNet
- The basic abstraction layer of PointNet++ is
  - Sample centroids
  - Group points by centroids
  - Apply PointNet on each point group



Qi et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. NIPS 2017.

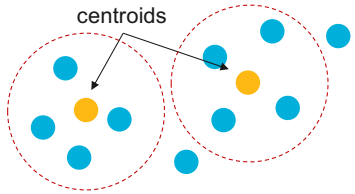
PointNet++ Abstraction layer

- Sampling centroids
  - Uniform sampling
  - Farthest sampling



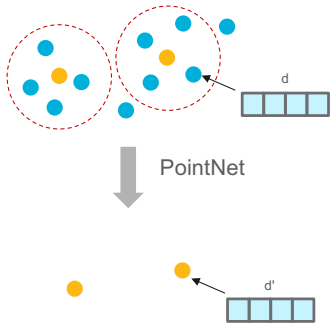
PointNet++ Abstraction layer

- Sampling
  - Uniform sampling
  - Farthest sampling
- Grouping
  - K nearest neighbors
  - Ball query (within range)



PointNet++ Abstraction layer

- Sampling
  - Uniform sampling
  - Farthest sampling
- Grouping
  - K nearest neighbors
  - Ball query (within range)
- Apply PointNet to each group

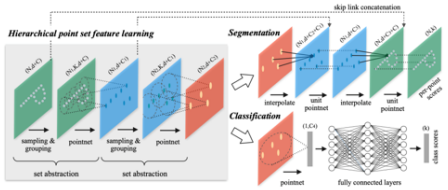


Multi-layer Hierachical Point Cloud Feature Learning

- Better result on ModeNetl40 dataset
- Can apply on large indoor 3D scene semantic segmentation (like ScanNet)

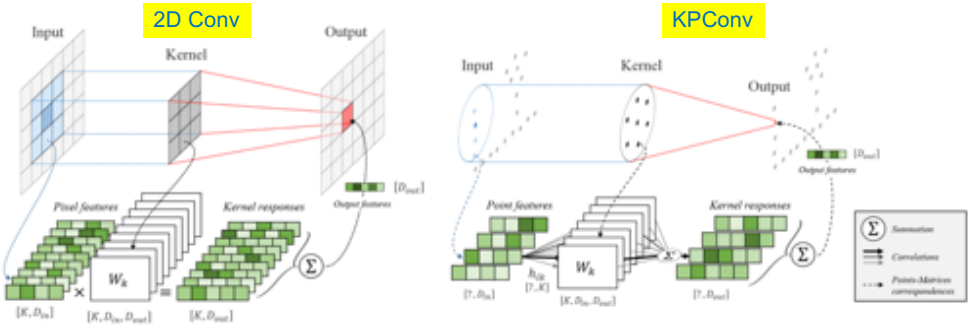
Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Result on ModelNet40. Better than MVCNN now.



KPConv

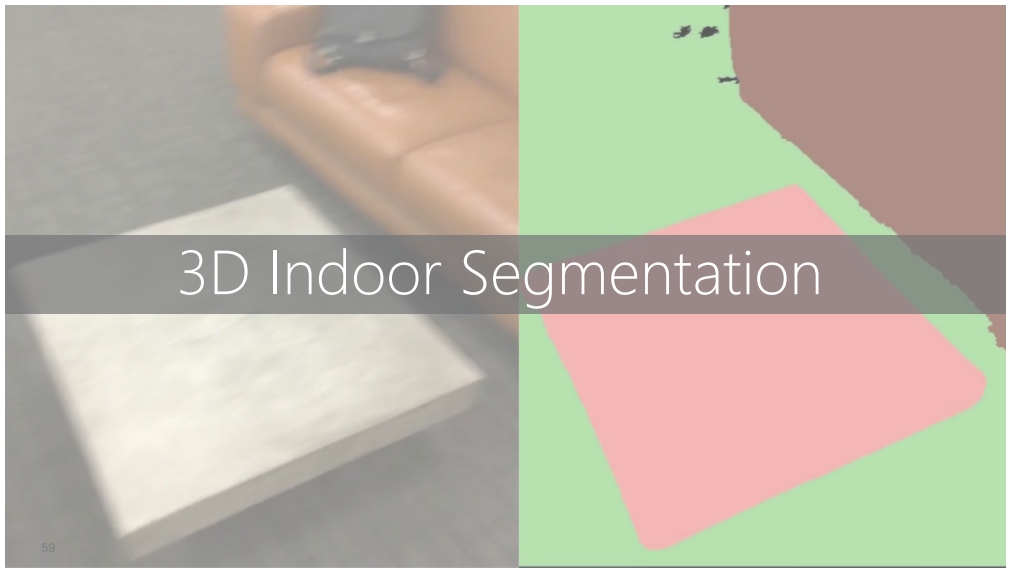
- Put weight on kernel (anchor) points, apply weight on target point feature by the distance between target point and kernel point.
  - Design ***h*** as influence weight for point on kernel points (hinge similarity)

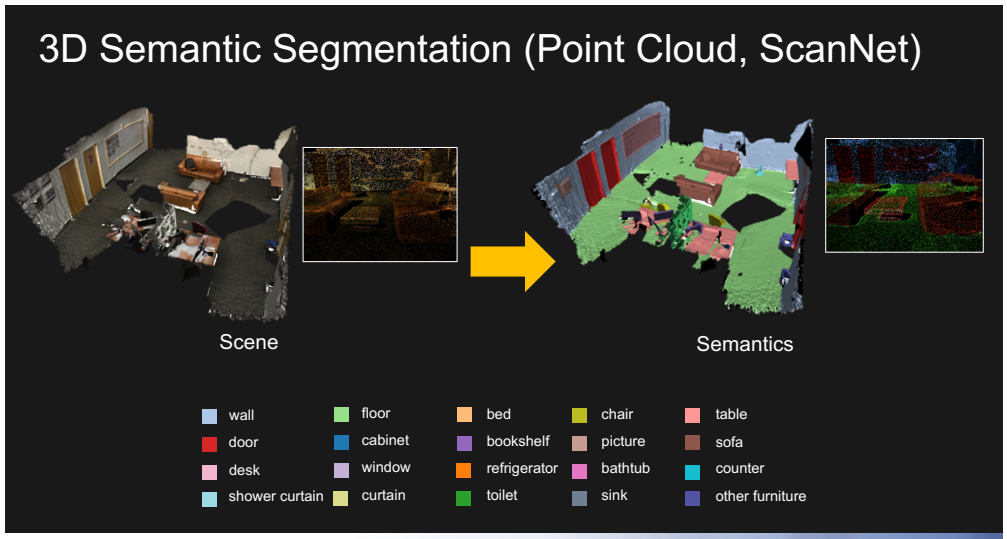


Thomas, et al. KPConv: Flexible and Deformable Convolution for Point Clouds. ICCV 2019

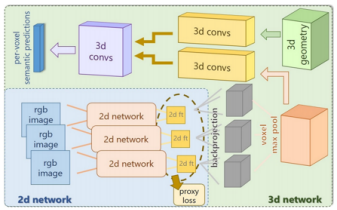
More Point-based Methods

- PCNN
- PointNet++
- DGCNN
- PointWeb
- KPConv
- So-Net
- Deep parametric continuous CNN
- PointCNN

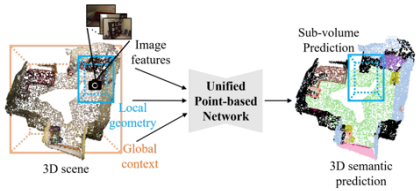




Fusion 2D-3D information



Dai et al. 3DMV, ECCV 2018.

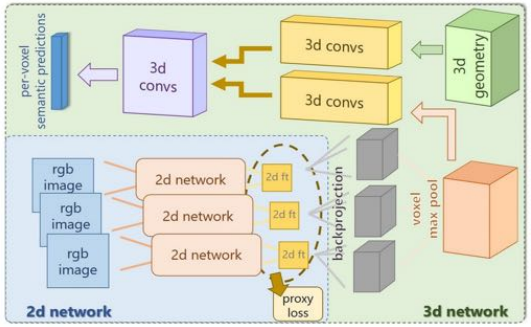


Chiang et al. Unified Point-Based Framework, 3DV 2019

Related Work – 3DMV

- Stanford, TUM at ECCV 2018.
- Volumetric based, jointly with image features.

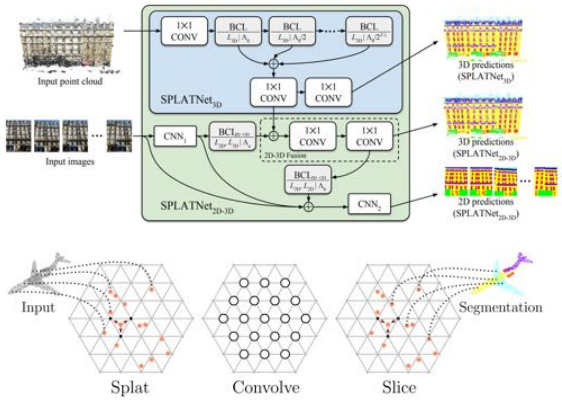
- Issues
  - Pre-processing 3D data into a voxel representation.
  - Voxelization loses input resolution.
  - Spatial redundancy occurs in the voxelized data.
  - Predict only a central column during testing.



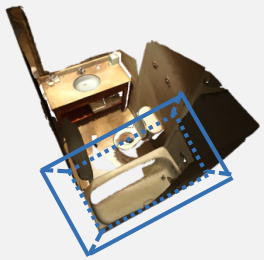
Related Work - SplatNet

- UMass Amherst, NVIDIA at CVPR 2018 (Best Mention Paper)
- Point based, jointly with image features.

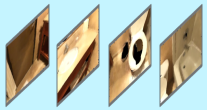
- Issues
  - Permutohedral lattice loses one-dimensional information.
  - Splat-and-Slice introduce the quantization errors.



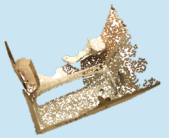
Our Observations



3D Scene



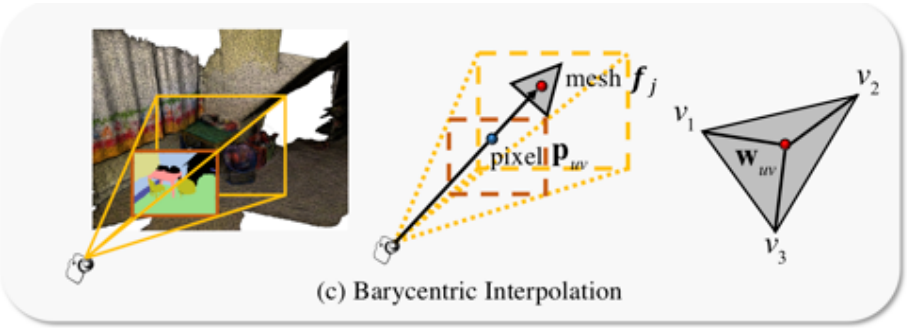
Color Texture Information (Rich 2D Deep Features)



Local Geometry Details

Global Scene Context

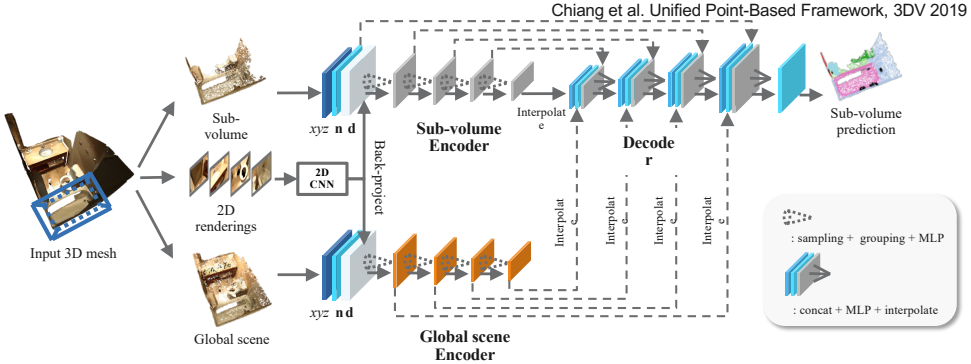
Back-project Deep Feature to 3D Point Clouds



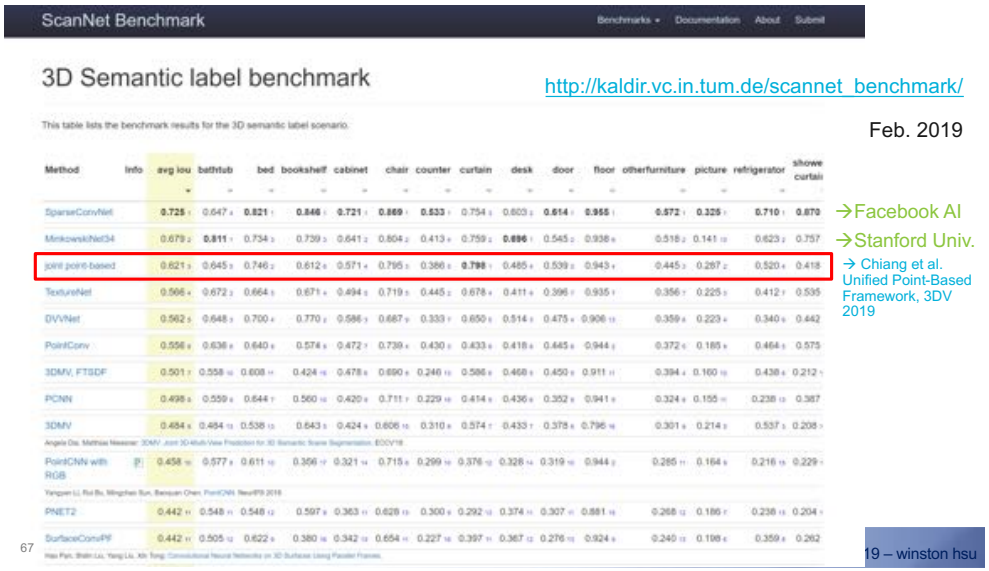
Chiang et al. Unified Point-Based Framework, 3DV 2019

Joint Framework (Global + Local)

Chiang et al. Unified Point-Based Framework, 3DV 2019



Method	Floor	wall	chair	sofa	table	door	cab	bed	desk	toil	sink	wind	pic	bkshf	curt	show	cntr	fridg	bath	other	mIOU
ScanNet [3]	78.6	43.7	52.4	34.8	30	18.9	31.1	36.6	34.2	46	31.8	18.2	10.2	50.1	0.2	15.2	21.1	24.5	20.3	14.5	30.6
PointNet++ [10]	78.6	52.3	36	34.6	23.2	261	25.6	47.8	27.8	54.8	36.4	25.2	11.7	45.8	24.7	14.5	25	21.2	58.4	18.2	33.9
SplatNet [11] CVPR'18	92.7	69.9	65.6	51	38.3	19.7	31.1	51.1	32.8	59.3	27.1	26.7	0	60.6	40.5	24.9	24.5	0.1	47.2	22.7	39.3
3DMV [4] ECCV'18	79.6	53.9	60.6	50.7	41.3	37.8	42.4	53.8	43.3	69.3	47.2	53.9	21.4	**64.3**	57.4	20.8	31	**53.7**	48.4	30.1	48.4
Joint framework (ours)	**94.3**	**79.5**	**79.5**	**74.4**	**57**	**53.9**	**57.1**	**74.6**	**48.5**	**85.9**	**63.5**	**62.8**	**28.7**	61.2	**79.8**	**41.8**	**38.6**	52	**64.5**	**44.5**	**62.1**



Feb. 2019

→Facebook AI

→Stanford Univ.

→ Chiang et al. Unified Point-Based Framework, 3DV 2019

3D Datasets

- 3D models
- Indoor scans
  - Real world
  - Synthetics scene
- Outdoor scans or Lidar

Dai et al. ScanNet, CVPR 2017

Chang et al. ShapeNet, arxiv 2015

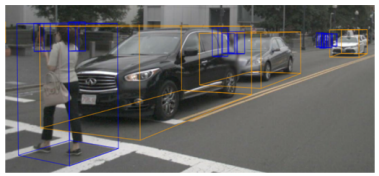
3D Datasets

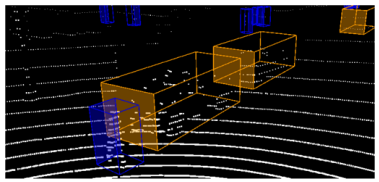
- Indoor
  - **NYUv2**
  - **ScanNet**
  - **MatterPort3D**
  - S3DIS
  - Stanford 3D dataset
  - SceneNet RGB-D
  - SceneNN
  - SUNCG
- 3D models
  - **ShapeNet**
  - ShapeNetPart
  - **ModelNet40**
  - ABC Dataset
- Outdoor
  - Paris-Lille-3D
  - Semantic3D.net
  - Ruemonge2014
  - **KITTI**
  - **nuScene**

LiDAR Object Detection

3D Object Detection for Autonomous Driving

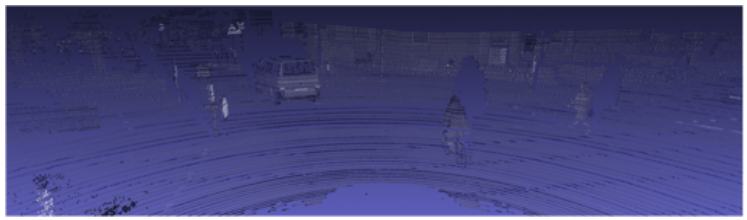
- Given
  - Sensor data
    - image(s)
    - LiDAR point clouds
    - RaDAR point clouds
- **Estimate**
  - Bounding box(es) with its attributes
    - object class
    - position
    - dimensions
    - 3D pose
    - velocity





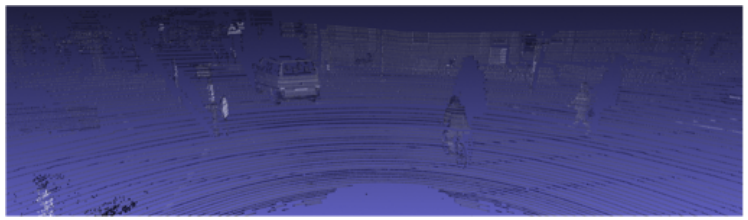
Applications for LiDAR Learning

- Object tracking
- Path planning
- Scene understanding
- …



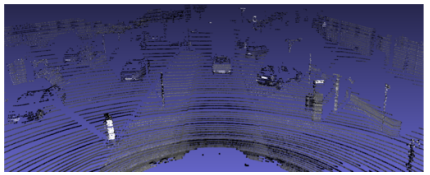
Challenges for Learning on LiDAR Point Clouds

- More dimensions than images
  - More space, time consuming
- Points are unordered, unstructured and sparse
  - Cannot apply (2D) convolution directly



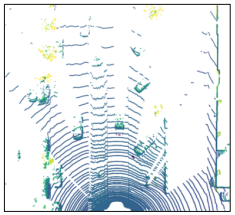
Projection-Based Methods

- Project points into 2D planes
  - Front view (FV)
  - Bird eye view (BEV)

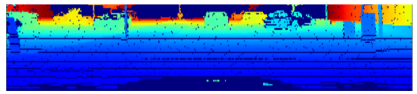


3D Point Cloud

Project

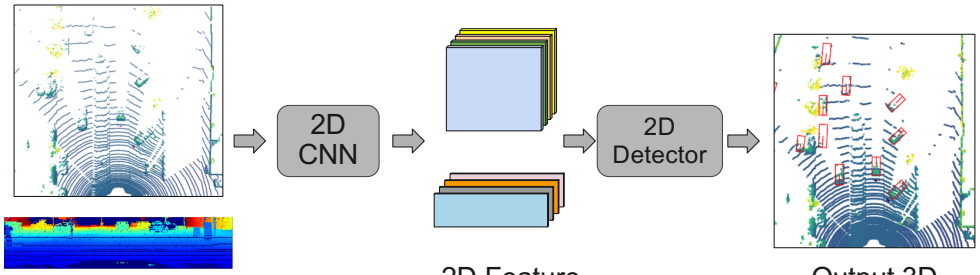


Bird Eye View

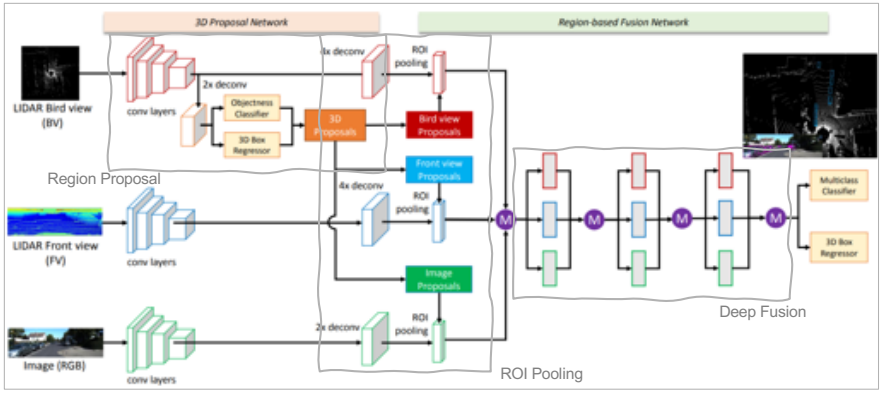


Front View

Projection-Based Methods



Multi-View 3D Object Detection Network (MV3D)



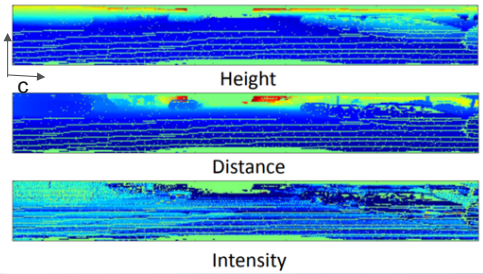
Front View Representation

- For each point $p = (x, y, z)$, calculate:

$$c = \lfloor \text{atan2}(y, x)/\Delta\theta \rfloor$$

$$r = \lfloor \text{atan2}(z, \sqrt{x^2 + y^2})/\Delta\phi \rfloor$$

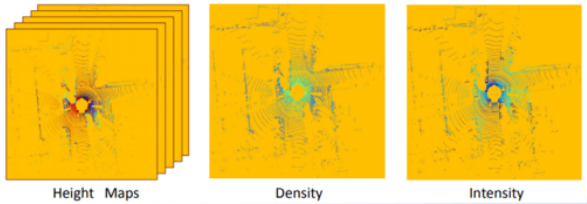
- Form a 64x512 front view representation with 3 channels:
  - Height
  - Distance
  - Intensity



Chen et al. Multi-view 3d object detection network for autonomous driving. CVPR 2017.

Bird Eye View Representation

- Project point cloud into a 2D grid with resolution of 0.1m (704x800)
- For each grid, compute:
  - **Highest** height in 5 height level
  - **Density** = $min(1.0, \frac{log(N+1)}{log(64)})$
  - **Intensity** = reflectance value of the point which has the maximum height



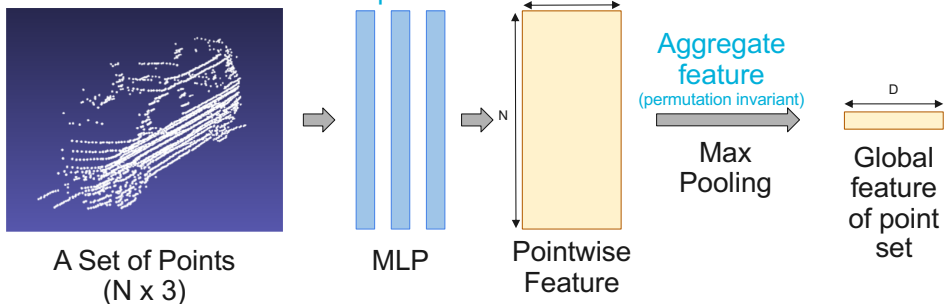
Chen et al. Multi-view 3d object detection network for autonomous driving. CVPR 2017.

Summary for MV3D

- A **multimodal network** to extract and aggregate point cloud and image features
- Use **2D projection** as input to extract point cloud feature
- A deep fusion design to enable more interactions among features
- Cons: using down-sampled bird eye view feature for object proposal → hard to detect small objects, e.g., pedestrians, cyclists, etc.

PointNet-Based Methods

- Second stage classifiers
- Basic feature extractors

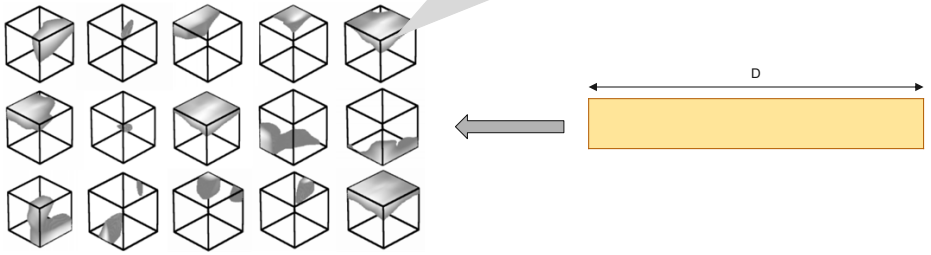


Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 2017.

PointNet-Based Methods

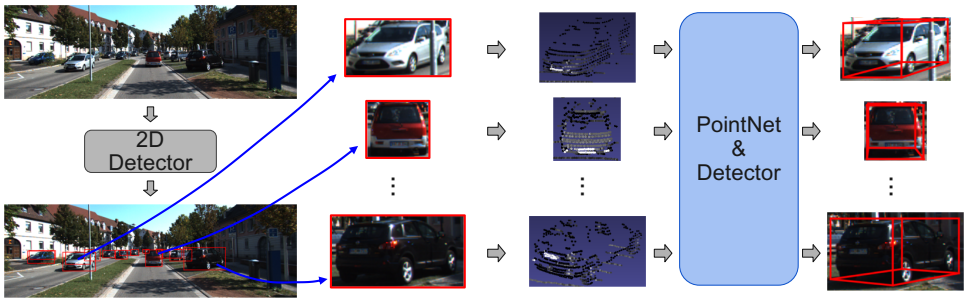
- PointNet
  - Randomly pick 15 point functions
  - Visualize the activation regions

non-uniform voxel



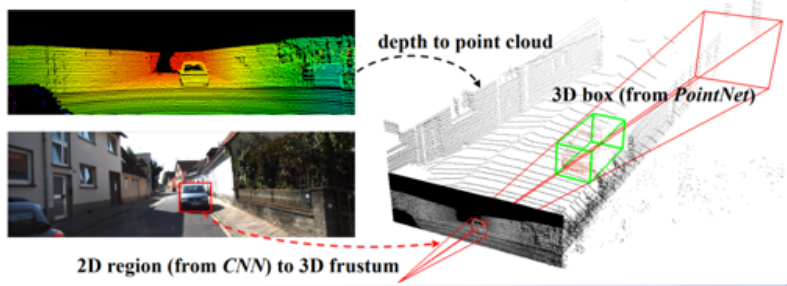
Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 2017.

PointNet-Based Methods – as Second Stage Classifier



Frustum PointNets

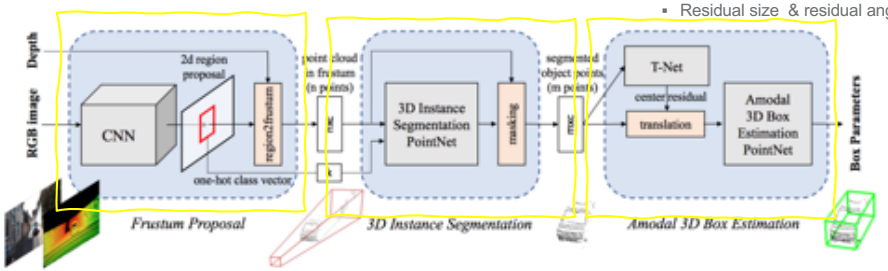
- (1) Generating 2D object region proposals
- (2) 2D region extruded to a 3D viewing for related point clouds
- (3) Predicting a (oriented and amodal) 3D bounding box from the points in frustum



Qi et al.. Frustum pointnets for 3d object detection from rgb-d data. CVPR 2018.

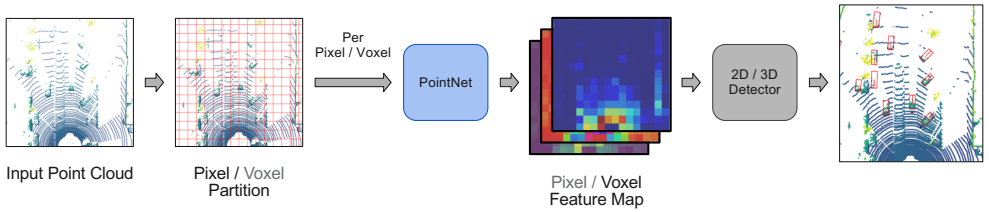
Frustum PointNets

- **Frustum proposal**
  - Start with 2D object detection
    - Higher resolution
    - Reducing search space
  - Using Feature Pyramid Network
  - Adjust coordinate to frustum center
- **3D Instance Segmentation**
  - Point cloud segmentation in frustum
  - Producing a point mask
    - like Mask R-CNN
  - Using PointNet or PointNet++
  - Adjusting coordinate to point mask centroid
- **Amodal 3D Box Estimation**
  - T-Net
    - Predicting object center & orientation
    - Inspired by Spatial Transformer Networks (2D)
  - PointNet or PointNet++ to estimate
    - Residual size & residual angle



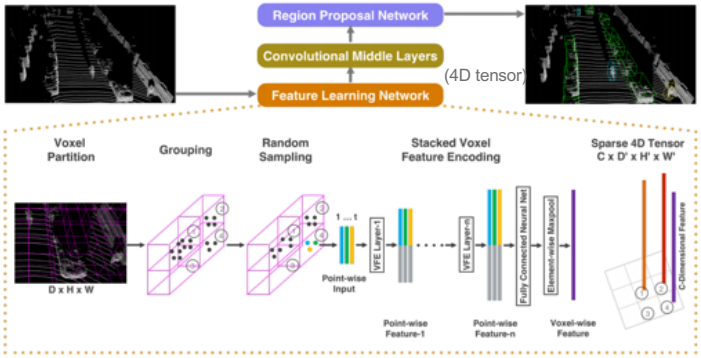
Qi et al.. Frustum pointnets for 3d object detection from rgb-d data. CVPR 2018.

PointNet-Based Methods – as Basic Feature Extractors



VoxelNet

- At fixed (3D) voxel with hugely varying # of point clouds (processing at most T points; others **dropped**)
- Point feature with 3D position, and its delta to the voxel center + reflectance
- (3D) Regional proposal network (RPN) to detect the objects
- 3D conv. is time-consuming
- One of the best.



Zhou et al. VoxelNet: end-to-end learning for point cloud based 3D object detection. CVPR 2018

VoxelNet – Point-wise Feature Learning

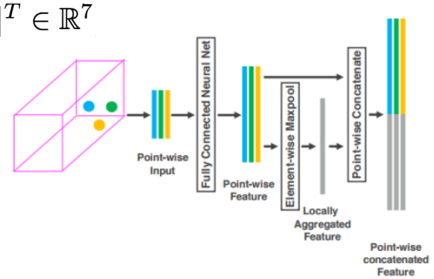
- each point:

$$[x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z]^T \in \mathbb{R}^7$$

- encoded point-wise feature :

$$[\mathbf{f}_i^T, \tilde{\mathbf{f}}^T]^T \in \mathbb{R}^{\hat{2}m}$$

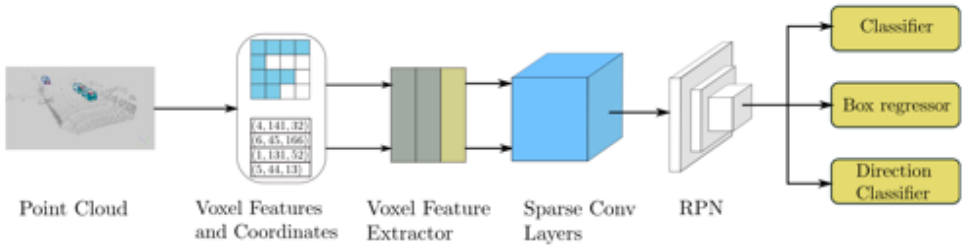
point-wise feature locally aggregated feature



Zhou et al. VoxelNet: end-to-end learning for point cloud based 3D object detection. CVPR 2018

SECOND: Sparsely Embedded Convolutional Detection

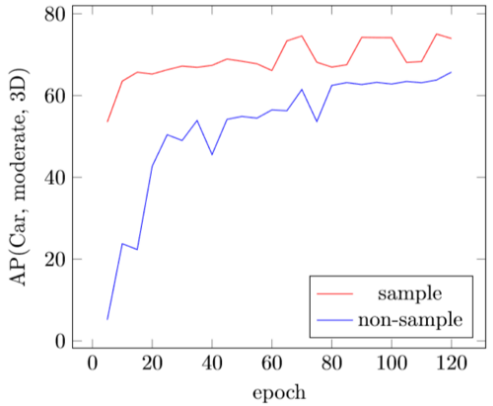
- VoxelNet suffers from time-consuming 3D convolutions
  → replace 3D Convs with 3D **sparse** Convs
- Angle loss regression to improve the orientation estimation
- Achieving **a factor-of-4 speed** enhancement during **training** on the KITTI dataset and a **factor-of-3 improvement** in the speed of inference.



Yan et al. SECOND: Sparsely Embedded Convolutional Detection. Sensors 2018.

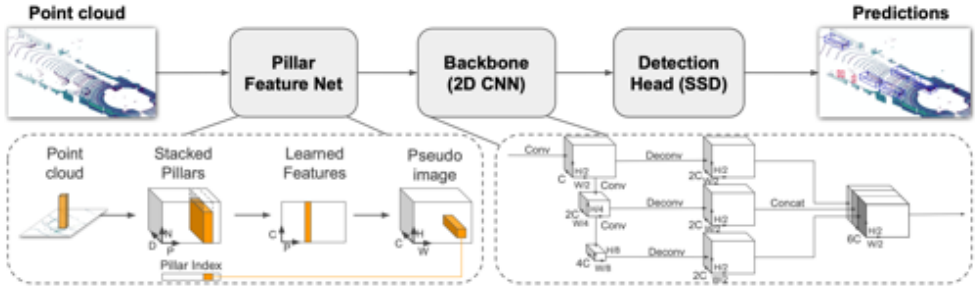
SECOND – Data Augmentation Strategy

- Existence of too few ground truths significantly limited the convergence speed and performance of the network.
- Sampling ground truths from the whole dataset
- Test for collided (sampled) objects
- Random linear transformations & rotation



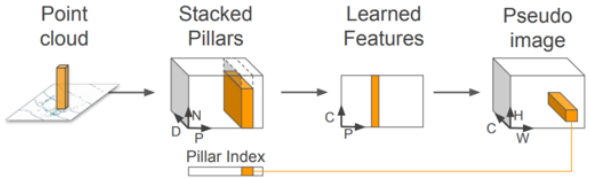
PointPillars

- Utilizing PointNets to learn a representation of point clouds organized in vertical columns (pillars).
- Outperforming STOA in both accuracy and speed

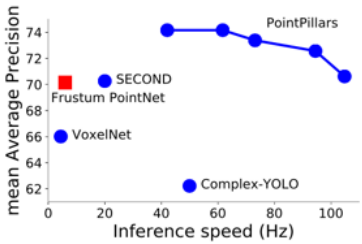


PointPillars vs. SECOND vs. VoxelNet

- VoxelNet & SECOND
  - use VFE to extract voxel-wise feature
  - use 3D convs / sparse convs to merge voxel feature
- PointPillars
  - use PointNets to extract pillar-wise feature and scatter into pseudo image
  - use 2D CNN to merge pillar (pseudo image) feature



PointPillars – Efficient & Effective

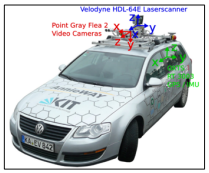


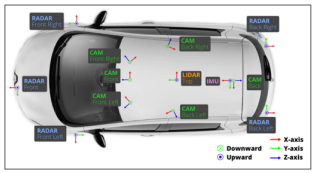
Method	Modality	Speed (Hz)	mAP	Car			Pedestrian			Cyclist		
			Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [2]	Lidar & Img.	2.8	N/A	71.09	62.35	55.12	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [15]	Lidar & Img.	16.7	N/A	82.54	66.22	64.04	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [25]	Lidar & Img.	10	N/A	**83.71**	73.04	59.16	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [11]	Lidar & Img.	10	55.62	81.94	71.88	66.38	50.80	42.81	40.88	64.00	52.18	46.61
F-PointNet [21]	Lidar & Img.	5.9	57.35	81.20	70.39	62.19	51.21	**44.89**	40.23	71.96	56.77	50.39
VoxelNet [33]	Lidar	4.4	49.05	77.47	65.11	57.73	39.48	33.69	31.5	61.22	48.36	44.37
SECOND [30]	Lidar	20	56.69	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90
PointPillars	Lidar	**62**	**59.20**	79.05	**74.99**	**68.30**	**52.08**	43.53	**41.49**	**75.78**	**59.07**	**52.92**

Table 2. Results on the KITTI test 3D detection benchmark.

Datasets

- **KITTI** (2017)
  - Samples: 3769 train / 3712 val / 7518 test
  - Sensors: 2 * RGB Cameras, 1 * **LiDAR (64 channels)**
- **NUSCENES** (2019)
  - Samples: ~28000 train / ~6000 val/ ~6000 test
  - Sensors: 6 * RGB Cameras, 1 * **LiDAR (32 channels),** 5 * RaDAR

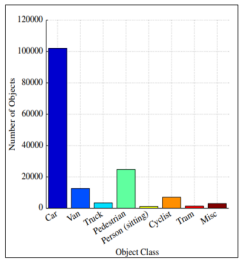




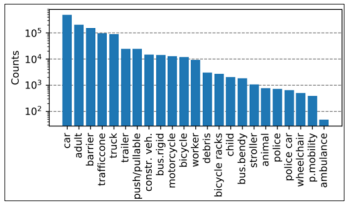
Geiger et al. Are we ready for autonomous driving? the kitti vision benchmark suite. CVPR 2012.
Caesar, et al. nuScenes: A multimodal dataset for autonomous driving. arXiv 2019.

Datasets

- KITTI
  - 80,256 labeled objects of 8 object classes
  - **90 degree** annotation range



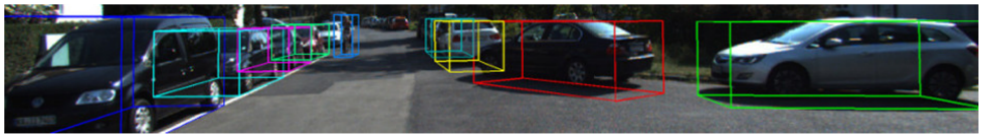
- NUSCENES
  - 1,166,187 labeled objects of 23 object classes
  - **360 degree** annotation range
  - objects labeled with velocity



Geiger et al. Are we ready for autonomous driving? the kitti vision benchmark suite. CVPR 2012.
Caesar, et al. nuScenes: A multimodal dataset for autonomous driving. arXiv 2019.

Evaluation - KITTI

- Only evaluating 3 classes: Car / Pedestrian / Cyclist
- Measuring precision & recall with **3D IOU threshold**
  - 0.7 / 0.5 / 0.5 for Car / Pedestrian / Cyclist
- Boxes are divided into easy / moderate / hard difficulties
  - according to its size on image and occlusion level



Geiger et al. Are we ready for autonomous driving? the kitti vision benchmark suite. CVPR 2012.

Evaluation - NUSCENES

- Only evaluating 10 of the 23 classes
- Measures precision & recall with distance threshold
  - 0.5m / 1m / 2m / 4m
- Box error is measured by true positive metrics
- average translation / scale / velocity / orientation / attribute error

.Caesar, et al. nuScenes: A multimodal dataset for autonomous driving. arXiv 2019.

3D Face Recognition

Face Recognition – Verification vs. Identification

- Face verification (open-set)

1 : 1

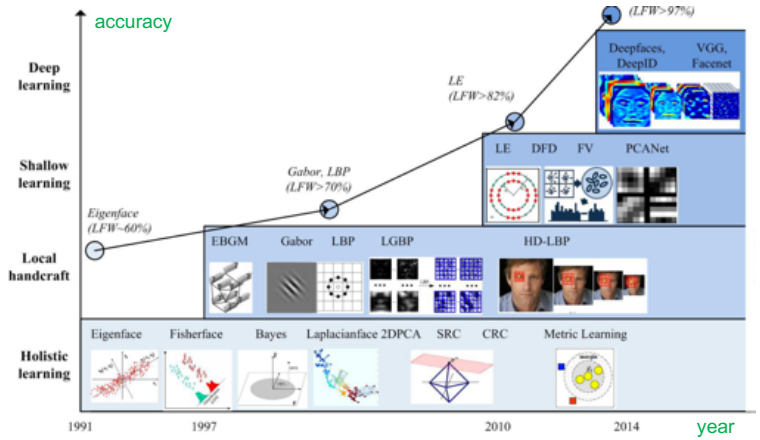
- Face identification (open-set)

Probe set

Gallery set

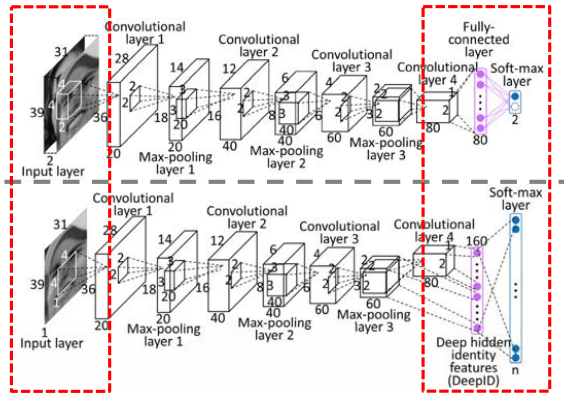
1 : N

Evolvement of Face Recognition (Extending to 3D)



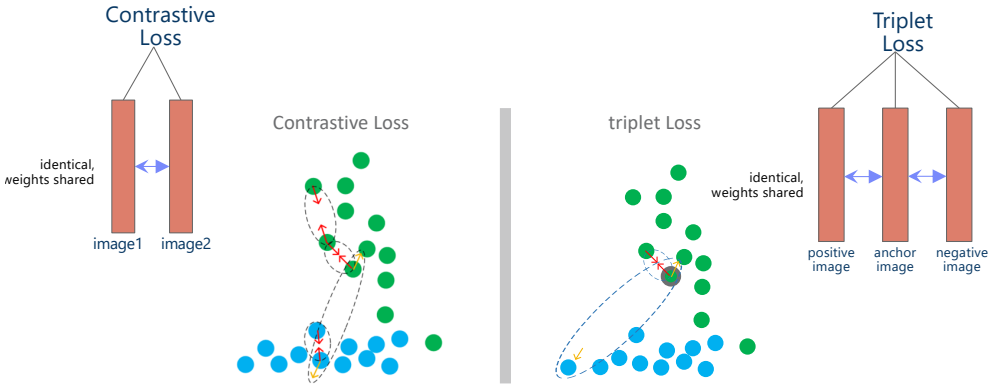
Wang et al. Deep Face Recognition: A Survey. arXiv 2019

From Classification to Metric Learning (Verification)

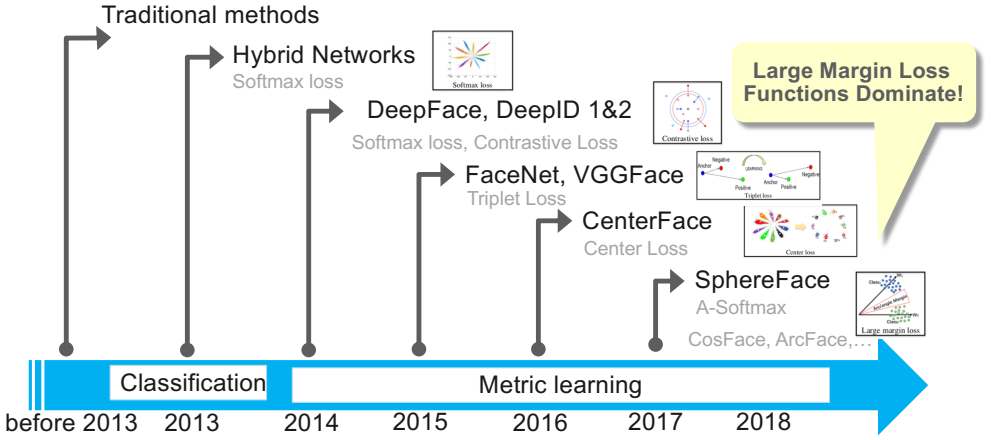


- **DeepID**: Sun Y, et al. Deep learning face representation from predicting 10,000 classes. CVPR 2014.
- **DeepFace**: Taigman Y, et al. Deepface: Closing the gap to human-level performance in face verification. CVPR 2014

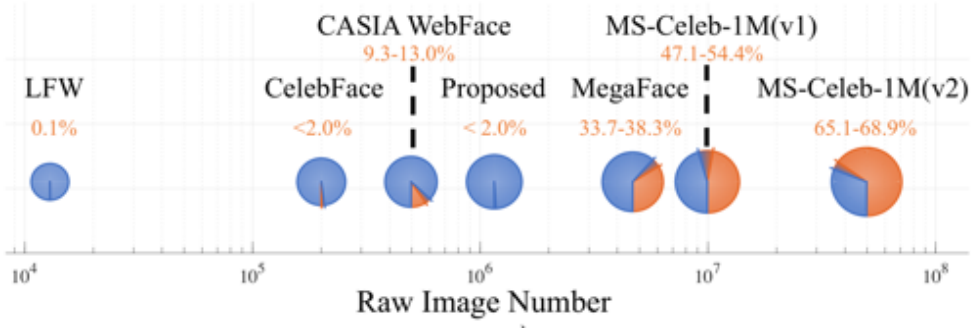
Distance Metric Learning: Siamese vs. Triplet



Evolvement for the Face Recognition Methods



Large (2D) Face Datasets (but Biased & Noisy)

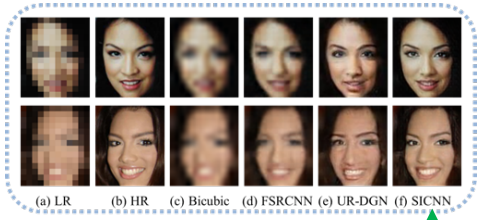


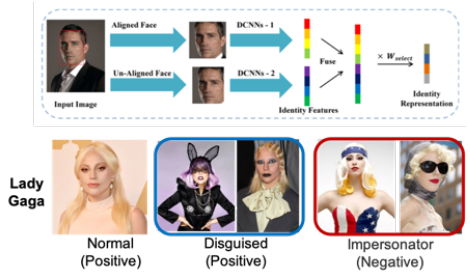
- Wang et al. profiled the noise distribution in existing datasets and showed that the noise percentage increases dramatically along the scale of data.
- Some datasets are further cleaned and shared in the community

Wang et al. The devil of face recognition is in the noise. ECCV 2018.

Challenges for Low-Resolution and Disguised Face Recognition

- Identity-preserving super-resolution from very low-resolution facial images (e.g., 12x14 pixels)
- Disguised face recognition by informative and principal deep representations
- First Place in IARPA Disguised Faces in the Wild Competition (CVPR 2018)

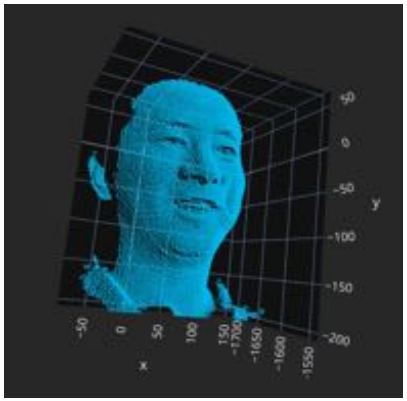




Zhang et al. Super-Identity Convolutional Neural Network for Face Hallucination. ECCV 2018
Zhang et al. Deep Disguised Faces Recognition. CVPRW 2018

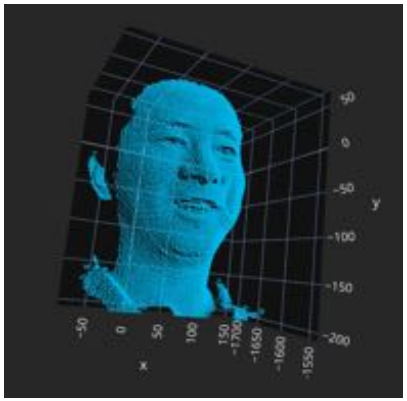
Human Faces as 3D Point Clouds

- Requirement – invariant and efficient 3D face recognition models considering
  - Minimal point cloud
  - Efficient computation
  - Invariance (translation, rotation, etc.)
  - 2D vs. 2.5D vs. 3D
  - Normalization, landmarks
  - Loss functions?
  - Voxel vs. Point Cloud
- Most existing ones are over 2.5D
- Not so many discussions yet → limited by (3D face) dataset availability



3D Face Recognition Pipeline (Traditional)

- Preprocessing
  - spike/hole removal
  - landmark detection
  - face normalization
- Transformation
  - point cloud to depth map
  - surface normal estimation
- Recognition
- Others, yet to be discovered!!!



HK Classification (Curvature-based Landmark Detection)

	K < 0	K = 0	K > 0
H < 0	Hyperbolic	Cyl. convex	Ellip. convex
H = 0	Hyperbolic	Planar	Impossible
H > 0	Hyperbolic	Cyl. concave	Ellip. concave

TABLE I
HK-CLASSIFICATION [16].

$$H(x,y) = \frac{(1+f_y^2)f_{xx} - 2f_x f_y f_{xy} + (1+f_x^2)f_{yy}}{2(1+f_x^2+f_y^2)^{\frac{3}{2}}},$$

$$K(x,y) = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1+f_x^2+f_y^2)^2},$$

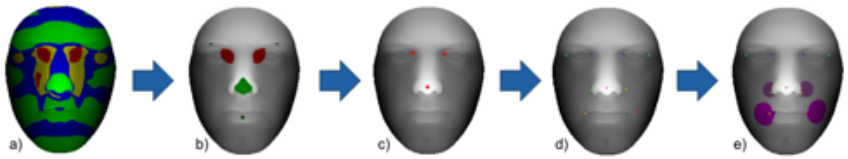
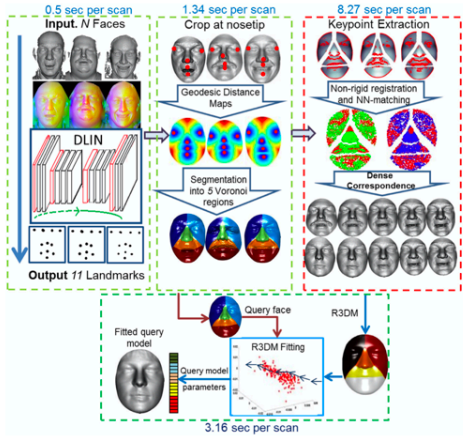


Fig. 2. Main points localization algorithm: a) HK-Classification, b) nose and eyes regions, c) (coarse localization) the nose tip point and the inner corners of eyes points, d) generic model aligment, e) fine adjusting of points

P Szeptycki, A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking, ICB 2009.

3D Landmark Detection for Dense 3D Face Correspondence

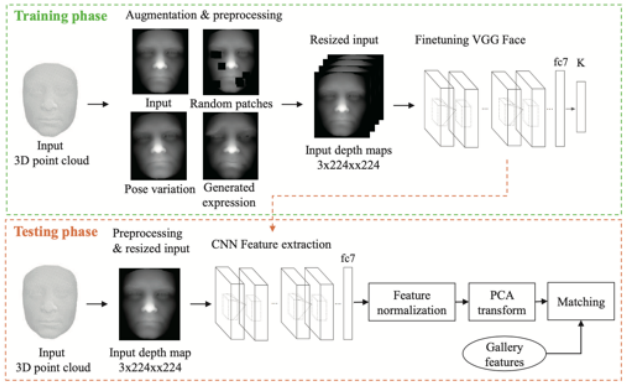
- **Dense 3D shape correspondence** – mapping between a large number points on one surface to topologically similar points on other surfaces.
- Deep Landmark Identification Network (DLIN) for 11 landmarks over the 3 channel images
- Correspondences (with keypoints within regions)
  - Also use for facial similarity



Gilani et al. Deep, dense and accurate 3D face correspondence for generating population specific deformable models. Pattern Recognition 2017

Augmentation with 3D Expression and Pose Variations

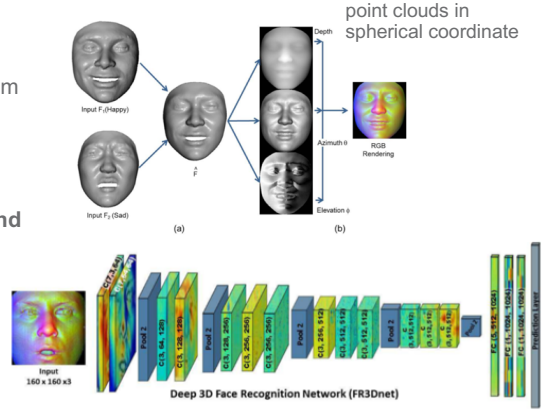
- Augmented 3D faces with **expression** and **pose** variations
  - By 3D **morphable model** (3DMM), after fitting with point clouds
- Random patches are removed from the depth maps
- (2D) 2.5D face recognition



Kim et al. Deep 3D Face Identification. IJCB 2017

Learning from Millions of 3D Scans for Large-scale 3D Face Recognition

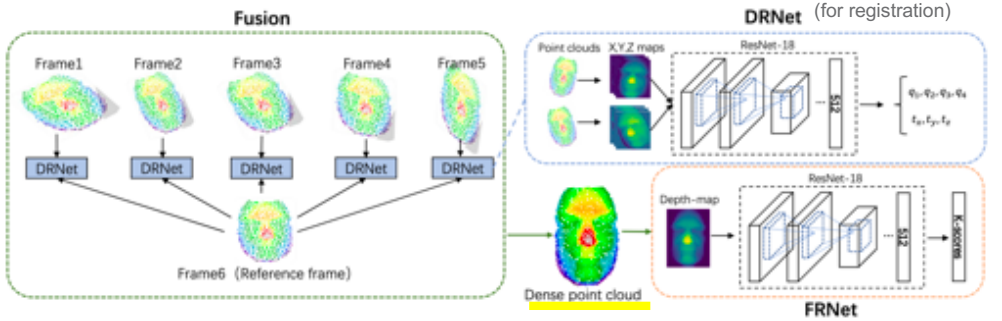
- **3D face generation**
  - **Generating** 90,100 distinct faces from existing 3D face datasets
  - Dense correspondence over 3D faces using the **keypoints based** algorithm (software)
  - **Interpolating between identities and expressions**, they generate new identities not in the linear space of the original identities
- Network Architecture for face recognition (2D or 2.5D)



Gilani et al. Learning from Millions of 3D Scans for Large-scale 3D Face Recognition. CVPR 2018

Face Recognition from **Sequential Sparse** 3D Data via Deep Registration

- Robust to sparse point cloud data by **aggregating** frames of 1000+ points (low-cost cam)
- Projected maps for multiple-channel inputs for estimating registration numbers (not ICP)



Tan et al. Face Recognition from Sequential Sparse 3D Data via Deep Registration. ICB 2019

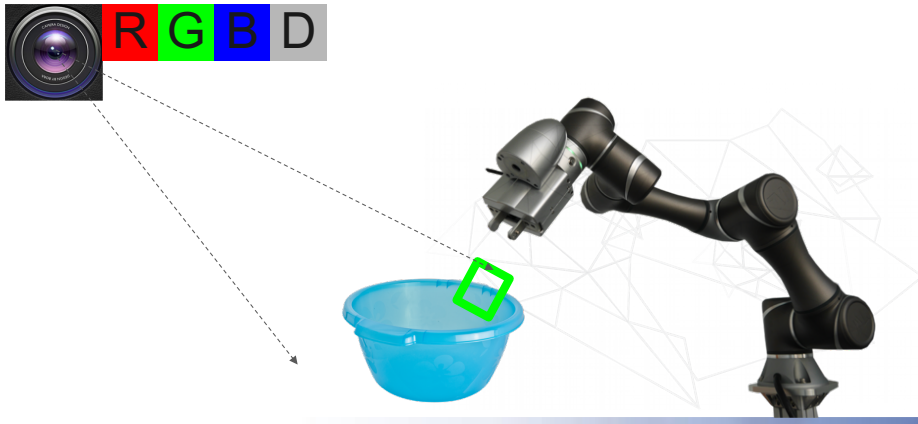
3D Face Datasets

Name	IDs	Scans	Expressions	Pose	Occlusion	Scanner
FRGCv2 [45]	466	4,007	Multiple	±15°	None	Laser
BU3DFE [59]	100	2,500	6 ×4	Frontal	None	Stereo
Bosphorus [47]	105	4,666	7	±90°	4 types	Stereo
GavabDB [39]	61	488	Multiple	±30°	None	Laser
Texas FRD [22]	118	1,151	Multiple	Frontal	None	Stereo
BU4DFE [58]	101	3,030	6 ×5	Frontal	None	Stereo
CASIA [57]	123	4674	6	±90°	None	Laser
UMB DB [11]	143	1,473	4	Frontal	7 types	Laser
3D-TEC [56]	214	428	2	Frontal	None	Laser
ND-2006 [15]	422	9,443	Multiple	±15°	None	Laser

Gilani et al. Learning from Millions of 3D Scans for Large-scale 3D Face Recognition. CVPR 2018

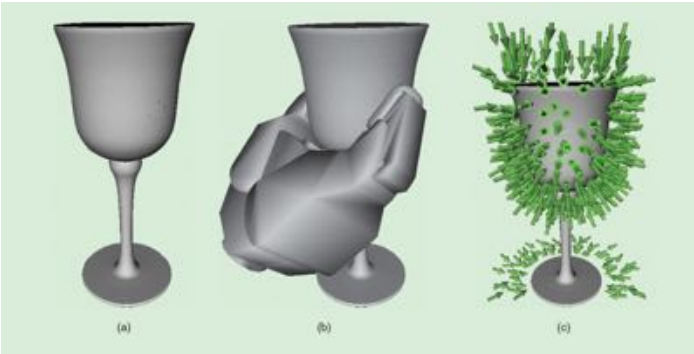
3D Robotic Grasp Detection

Problem Description



Grasp Planning for Fitting Known Objects (CAD)

- Prior methods replying on known CAD model to fit the observed point cloud data



 Chitta et al. Mobile Manipulation in Unstructured Environments: Perception, Planning, and Execution. IEEE Robotics & Automation Magazine 2012.

Learning for Grasp Rectangle

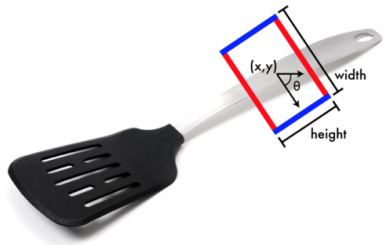
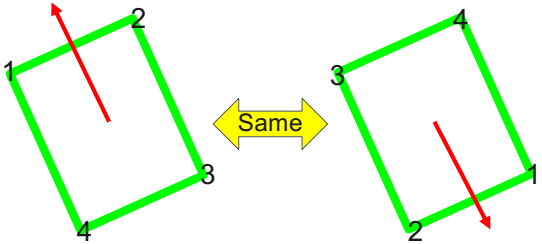


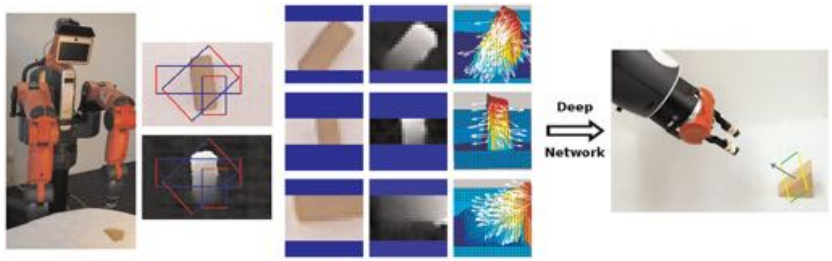
Fig. 2. A five-dimensional grasp representation, with terms for location, size, and orientation. The blue lines demark the size and orientation of the gripper plates. The red lines show the approximate distance between the plates before the grasp is executed.



 Redmon & Angelova. Real-Time Grasp Detection Using Convolutional Neural Networks. ICRA 2015.

Ranking Grasp Candidates

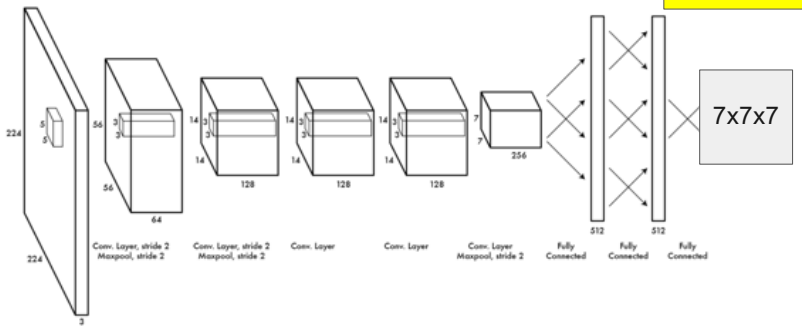
- Ranking grasp candidates by color, depth images and surface normal
- **Cornell Grasp Dataset**



Lenz et al. Deep learning for detecting robotic grasps. IJRR 2015.

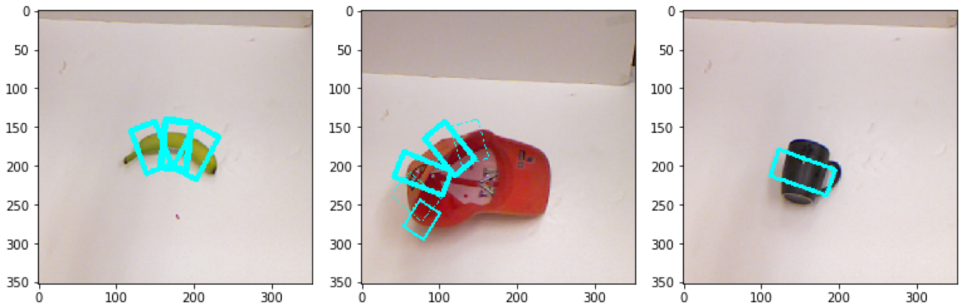
Real-Time Grasp Detection Using Convolutional Neural Networks (YOLO-Like)

7x7 grid:
confidence, x, y, w, h, sin, cos
(very similar to YOLOv1)



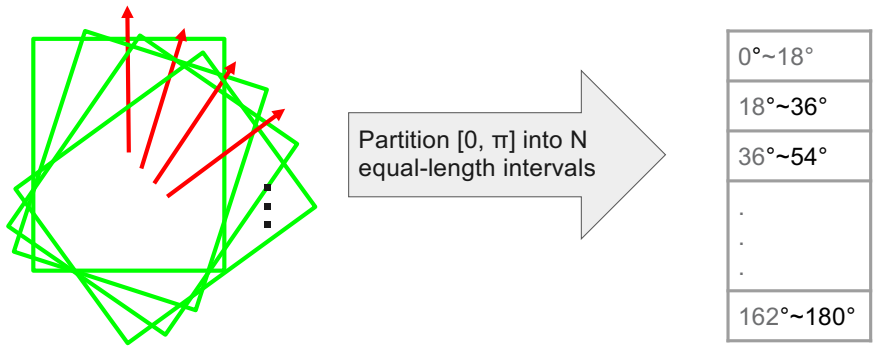
Redmon & Angelova. Real-Time Grasp Detection Using Convolutional Neural Networks. ICRA 2015.

Real-Time Grasp Detection Using Convolutional Neural Networks (YOLO-Like)



Redmon & Angelova. Real-Time Grasp Detection Using Convolutional Neural Networks. ICRA 2015.

Improvement – from Regression to Categorization



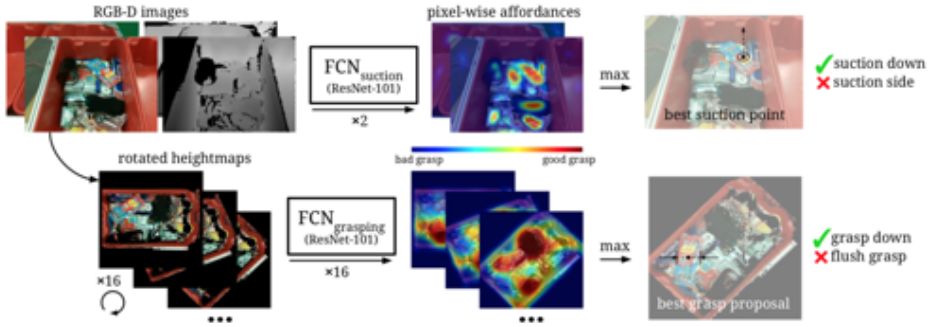
Improvement – from Regression to Categorization

TABLE I
SINGLE-OBJECT SINGLE-GRASP EVALUATION

approach	image-wise	object-wise	speed
	Prediction Accuracy (%)		fps
Jiang et al. [15]	60.5	58.3	0.02
Lenz et al. [18]	73.9	75.6	0.07
Redmon et al. [20]	88.0	87.1	3.31
Wang et al. [19]	81.8	N/A	7.10
Asif et al. [11]	88.2	87.5	–
Kumra et al. [21]	89.2	88.9	16.03
Mahler et al. [25]	93.0	N/A	~1.25
Guo et al. [23]	93.2	89.1	–
Ours: VGG-16 (RGB-D)	**95.5**	**91.7**	17.24
Ours: Res-50 (RGB)	**94.4**	**95.5**	8.33
Ours: Res-50 (RGB-D)	**96.0**	**96.1**	8.33

Chu et al. Real-World Multiobject, Multigrasp Detection, IEEE Robotics and Automation Letters, 2018

Suction and Grasp Affordance prediction from Multi-view **RGB-D** Images



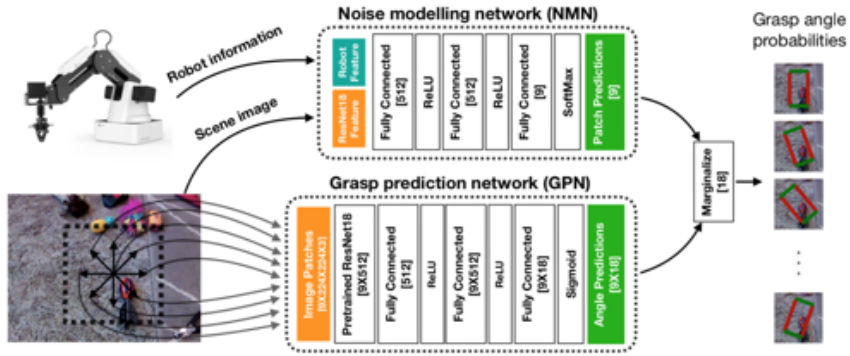
Zeng et al., Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching, ICRA 2018

Robot Learning in Homes: Improving Generalization and Reducing Dataset Bias

Figure 3: Homes used for collecting training data and environments where models were tested

Gupta et al. Robot Learning in Homes: Improving Generalization and Reducing Dataset Bias, NIPS 2018

Robot Learning in Homes (via Airbnb)

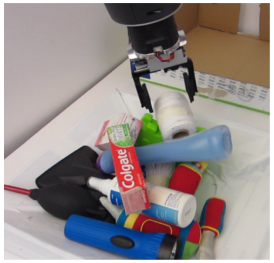


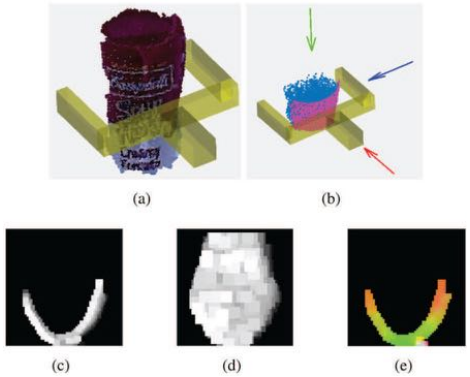
Gupta et al. Robot Learning in Homes: Improving Generalization and Reducing Dataset Bias, NIPS 2018

Grasp Pose Detection in Point Clouds

New grasp representation
(The regions closed by the gripper)

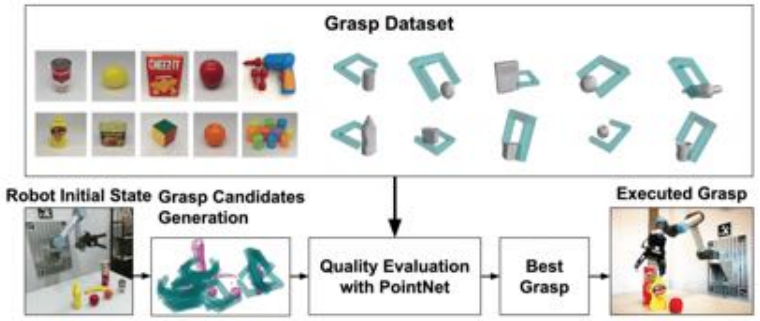
Use CNN to classify “good grasp”





Pas et al. Grasp Pose Detection in Point Clouds. IJRR 2017

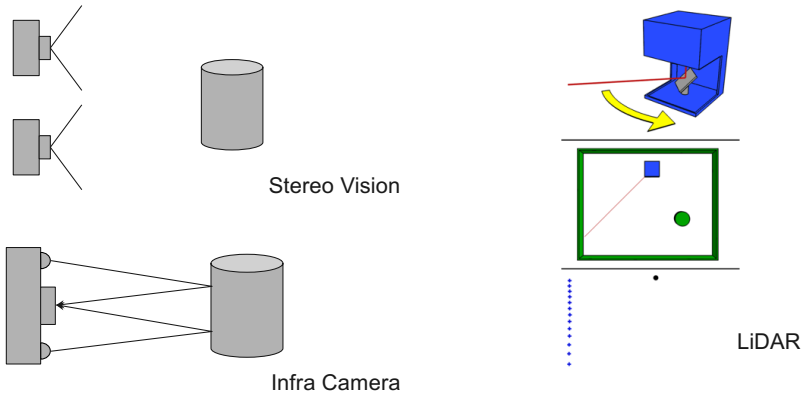
Grasp Quality Ranking with PointNet



Liang et al. PointNetGPD: Detecting grasp configurations from point sets. ICRA 2019

3D Quality Enhancement

Recap : Different Depth Sensors

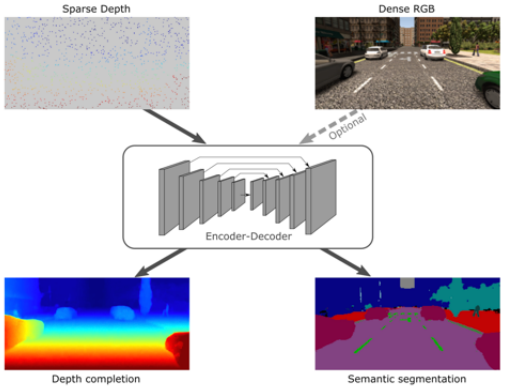


Limitations of Structured Light vs Stereo Vision

Stereo
Not precise but complete

Structured Light
Precise but Incomplete

Depth Quality Enhancement – Outdoor Data



Jaritz et al. Sparse and Dense Data with CNNs : Depth Completion and Sementic Segmentation. 3DV 2018

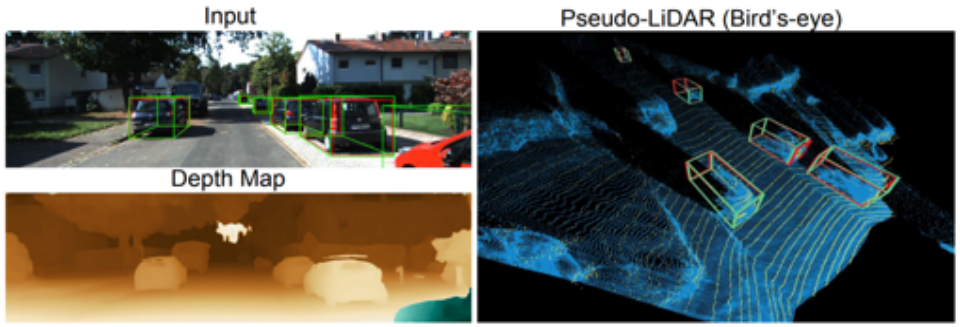
(Pseudo) LiDAR Outperforming 2D Camera - Vehicle Detection

Detection algorithm	Input signal	IoU = 0.5			IoU = 0.7		
		Easy	Moderate	Hard	Easy	Moderate	Hard
MONO3D [4]	Mono	30.5 / 25.2	22.4 / 18.2	19.2 / 15.5	5.2 / 2.5	5.2 / 2.3	4.1 / 2.3
MLF-MONO [33]	Mono	55.0 / 47.9	36.7 / 29.5	31.3 / 26.4	22.0 / 10.5	13.6 / 5.7	11.6 / 5.4
3DOP [5]	Stereo	55.0 / 46.0	41.3 / 34.6	34.6 / 30.1	12.6 / 6.6	9.5 / 5.1	7.6 / 4.1
MLF-STEREO [33]	Stereo	-	53.7 / 47.4	-	-	19.5 / 9.8	-
AVOD [17]	LiDAR + Mono	90.5 / 90.5	89.4 / 89.2	88.5 / 88.2	89.4 / 82.8	86.5 / 73.5	79.3 / 67.1
F-POINTNET [25]	LiDAR + Mono	96.2 / 96.1	89.7 / 89.3	86.8 / 86.2	88.1 / 82.6	82.2 / 68.8	74.0 / 62.0

Wang et al. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. CVPR 2019

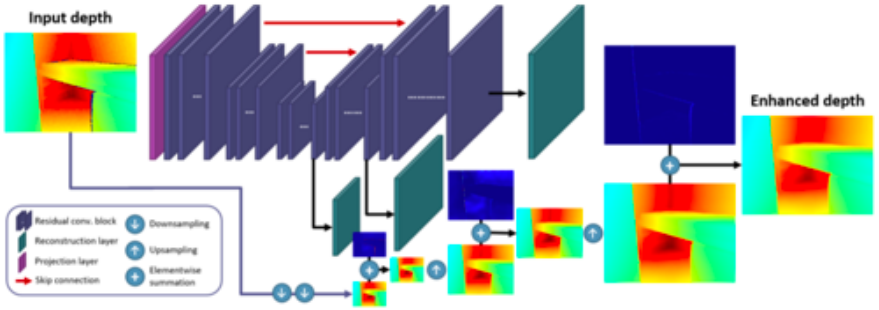
Pseudo-LiDAR from Two Cameras (Cost Effective)

- Approximating the state-of-the-art LiDAR performance with the (two) stereo cams



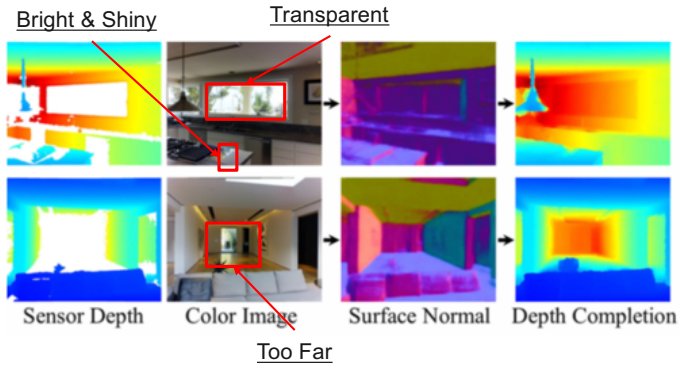
Wang et al. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. CVPR 2019

Depth Quality Enhancement – Indoor Data



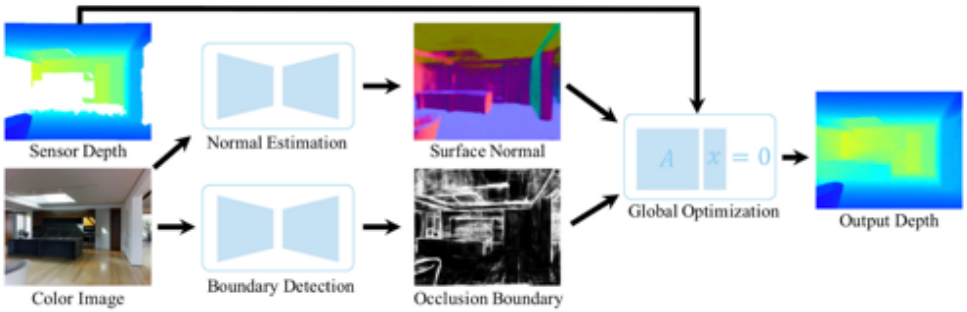
Jeon et al. Reconstruction-based Pairwise Depth Dataset for Depth Image Enhancement Using CNN. ECCV 2018

Depth Completion – Infrared structured light depth sensor results in large hole



Zhang et al. Deep Depth Completion of a Single RGB-D Image. CVPR 2018

Previous Work – Global Optimization



Zhang et al. Deep Depth Completion of a Single RGB-D Image. CVPR 2018

Motivation – Improve NN-based Depth Completion

- Naïve FCN leads to bad results
  - Copy and Interpolation
  - Blurred Boundaries

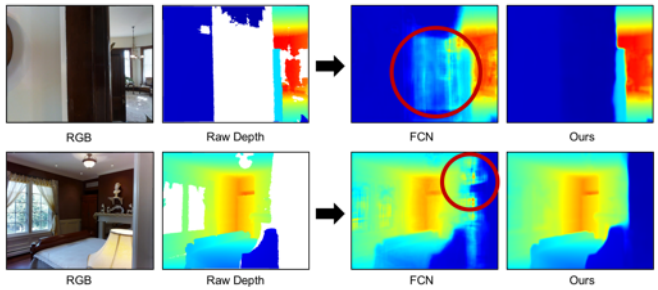
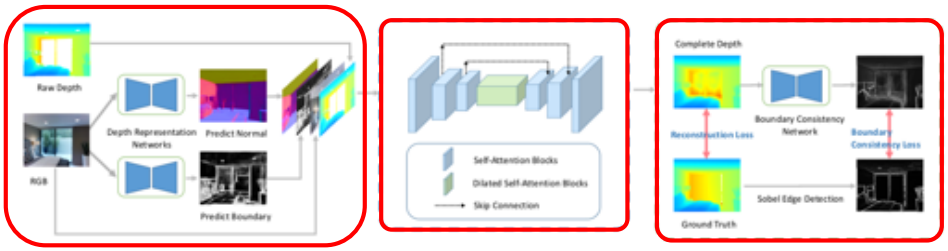


*Figure https://www.sdu.dk/en/om_sdu/institutter_centre/idk/projekter/human-robot+interaction/robot+zoo/kinect+for+xbox+360

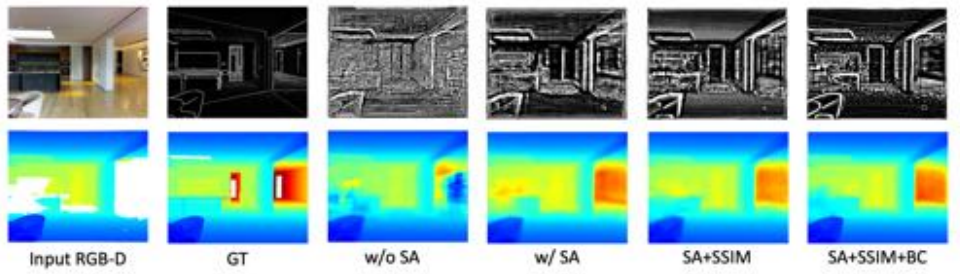
System Pipeline & Network Architecture

- **Feature Extraction**
  - Use two neural networks to extract surface normal and occlusion boundary
  - Stack the feature together and propagate to the main network
- **Attention Module**
  - Attention only on useful information controlled by soft gating layer.
  - Network can jointly fuse these input faetures
- **Boundary Consistency**
  - Ensure completed depth preserves boundary information
  - SSIM loss on complete depth and L1 loss on depth's boundary map



Huang et al. Indoor Depth Completion with Boundary Consistency and Self-Attention. ICCVW 2019.

Boundary Consistency Improves Depth Map Quality



Huang et al. Indoor Depth Completion with Boundary Consistency and Self-Attention. ICCVW 2019.

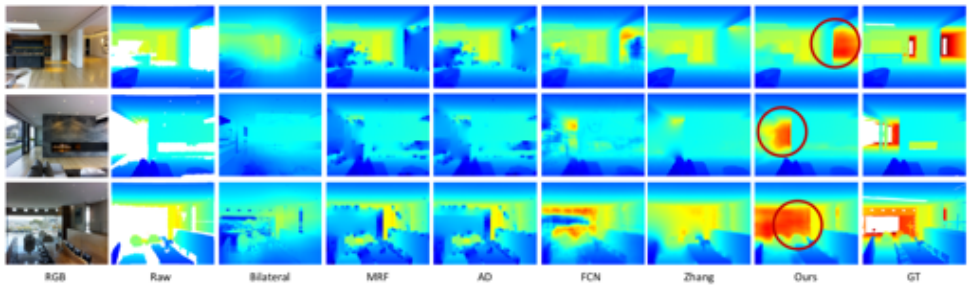
Experiment Results

- Dataset
  - Matterport3D indoor RGB-D dataset
  - 1M training data / about 500 testing data

Model	RMSE↓	Mean↓	SSIM↑	1.05 ↑	1.10 ↑	1.25 ↑	1.25^2 ↑	1.25^3 ↑
Bilateral	1.978	0.774	0.507	0.385	0.497	0.613	0.689	0.730
MRF	1.675	0.618	0.692	0.506	0.556	0.651	0.780	0.856
AD	1.653	0.610	0.696	0.503	0.560	0.663	0.792	0.861
FCN	1.262	0.517	0.605	0.397	0.527	0.681	0.808	0.868
Zhang	1.316	0.461	0.762	0.657	0.708	0.781	0.851	0.888
Ours	**1.092**	**0.342**	**0.799**	**0.661**	**0.750**	**0.850**	**0.911**	**0.936**

Huang et al. Indoor Depth Completion with Boundary Consistency and Self-Attention. ICCVW 2019.

Visualization – Learn Better on Geometric Meaning



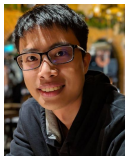
Huang et al. Indoor Depth Completion with Boundary Consistency and Self-Attention. ICCVW 2019.

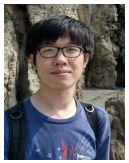
Summary

tutorial slides

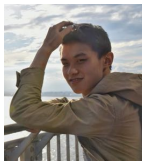
- 3D sensors are getting cheap and using in many important scenarios
  - Enabling numerous and exciting applications
- Point clouds retain rich geometric information
- Open researches in adopting (or fusing) 3D sensors for numerous applications
- Multimodal fusion requires more investigations
- Rich advancements for point cloud learning algorithms recently
  - Balancing efficiency and effectiveness
- 3D datasets are expensive to collect (& annotate)
  - How to leverage 2D or other augmentation methods?
- More challenging as deploying in the fields

Acknowledgement (Team & Industry Partners)


HungYueh Chiang


Steven Wang


Yueh-Cheng Liu


Tsung-Han Wu


Yu-Kai Huang


Kuang-Yu Jeng

MOST 科技部 Ministry of Science and Technology

AInfu 科技部臺灣大學人工智慧研究中心

NVIDIA

FIH 富智康 FIH Mobile Limited

QUALCOMM®