3D Deep Learning

Hao Su

@Stanford CS231n Guest Lecture
Broad Applications of 3D data

Robotics
Broad Applications of 3D data

Robotics

Augmented Reality
Broad Applications of 3D data

Robotics

Augmented Reality

Autonomous driving
Broad Applications of 3D data

- Robotics
- Autonomous driving
- Augmented Reality
- Medical Image Processing
Traditional 3D Vision

Multi-view Geometry: Physics based
3D Learning: Knowledge Based
Acquire Knowledge of 3D World by **Learning**

A priori knowledge of the 3D world
The Representation Challenge of 3D Deep Learning

Rasterized form (regular grids) vs. Geometric form (irregular)
The Representation Challenge of 3D Deep Learning

Multi-view

Volumetric

Part Assembly

Point Cloud

Mesh (Graph CNN)

Implicit Shape

\[ F(x) = 0 \]
The Richness of 3D Learning Tasks

3D Analysis

- Classification
- Segmentation (object/scene)
- Correspondence

Detection

- 3D box (from PointNet)
- 2D region (from CNN) to 3D frustum
- Depth to point cloud

It is a chair!
The Richness of 3D Learning Tasks

3D Synthesis

Monocular 3D reconstruction

Shape completion

Shape modeling
Agenda

• 3D Classification

• 3D Reconstruction

• Others
Volumetric CNN
Can we use CNNs but avoid projecting the 3D data to views first?

Straight-forward idea: Extend 2D grids 3D grids
Voxelization

Represent the occupancy of regular 3D grids
3D CNN on Volumetric Data

3D convolution uses 4D kernels
Complexity Issue

AlexNet, 2012
Input resolution: 224x224
224x224 = 50176

3DShapeNets, 2015
Input resolution: 30x30x30
224x224 = 27000
Complexity Issue

Polygon Mesh \[\rightarrow\] Occupancy Grid

Information loss in voxelization
Idea 1: Learn to Project

Idea: “X-ray” rendering + Image (2D) CNNs
very low #param, very low computation

Su et al., “Volumetric and Multi-View CNNs for Object Classification on 3D Data”, CVPR 2016

Many other works in autonomous driving that uses *bird’s eye view* for object detection
More Principled: Sparsity of 3D Shapes

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>10.41%</td>
</tr>
<tr>
<td>64</td>
<td>5.09%</td>
</tr>
<tr>
<td>128</td>
<td>2.41%</td>
</tr>
</tbody>
</table>
Store only the Occupied Grids

- Store the sparse surface signals
- Constrain the computation near the surface
Octree: Recursively Partition the Space

Each internal node has exactly eight children

Neighborhood searching: Hash table
Memory Efficiency

![Graph showing memory efficiency for different resolutions with O-CNN and Voxel CNN models.](image)
Implementation

- SparseConvNet
  - [https://github.com/facebookresearch/SparseConvNet](https://github.com/facebookresearch/SparseConvNet)
  - Uses ResNet architecture
  - State-of-the-art for 3D analysis
  - Takes time to train

Graham et al., "Submanifold Sparse Convolutional Networks", arxiv
Point Networks
Point cloud
(The most common 3D sensor data)
Directly Process Point Cloud Data

End-to-end learning for unstructured, unordered point data

Permutation invariance

Point cloud: $N$ orderless points, each represented by a $D$ dim coordinate

2D array representation
Permutation invariance

Point cloud: N orderless points, each represented by a D dim coordinate.

2D array representation

represents the same set as
Construct a Symmetric Function

Observe:

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

\[ h \]

(1,2,3)
(1,1,1)
(2,3,2)
(2,3,4)
Construct a Symmetric Function

Observe:

\[ f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n)) \]

is symmetric if \( g \) is symmetric

\[ g(1,2,3), (1,1,1), (2,3,2), (2,3,4) \]

simple symmetric function
Observe:

\[ f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n)) \]

is symmetric if \( g \) is symmetric.
Limitations of PointNet

Hierarchical feature learning
Multiple levels of abstraction

Global feature learning
Either one point or all points

- No local context for each point!
- Global feature depends on absolute coordinate. Hard to generalize to unseen scene configurations!
Points in Metric Space

- Learn “kernels” in 3D space and conduct convolution
- Kernels have compact spatial support
- For convolution, we need to find neighboring points
- Possible strategies for range query
  - Ball query (results in more stable features)
  - k-NN query (faster)
PointNet v2.0: Multi-Scale PointNet

Repeat
• Sample anchor points
• Find neighborhood of anchor points
• Apply PointNet in each neighborhood to mimic convolution
Point Convolution As Graph Convolution

- Points -> Nodes
- Neighborhood -> Edges
- Graph CNN for point cloud processing

Wang et al., “Dynamic Graph CNN for Learning on Point Clouds”, Transactions on Graphics, 2019

Liu et al., “Relation-Shape Convolutional Neural Network for Point Cloud Analysis”, CVPR 2019
Agenda

• 3D Classification

• 3D Reconstruction

• Others
Multi-View Stereo (MVS)

Reconstruct the dense 3D shape from a set of images and camera parameters

## Requirements of MVS

<table>
<thead>
<tr>
<th>Applications</th>
<th>Range</th>
<th>Accuracy</th>
<th>Time Efficiency</th>
<th>Computation Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Sensing</td>
<td>★★★★★</td>
<td>★★★</td>
<td>★</td>
<td>★★★</td>
</tr>
<tr>
<td>Autonomous Driving</td>
<td>★★★★★</td>
<td>★★★</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>AR/VR</td>
<td>★★★</td>
<td>★★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Robot Manipulation</td>
<td>★</td>
<td>★★★★★</td>
<td>★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>Inverse Engineering</td>
<td>★</td>
<td>★★★★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
</tbody>
</table>
Reconstruction from Photo-Consistency

NCC (Normalized Cross Correlation)
\[
\sum_{x,y} \frac{(W_1(x, y) - \overline{W_1})(W_2(x, y) - \overline{W_2})}{\sigma_{W_1} \sigma_{W_2}}
\]

SSD (Sum Squared Distance)
\[
\sum_{x,y} |W_1(x, y) - W_2(x, y)|^2
\]

- Requires texture
- Sensitive to Non-lambertian area

Image source: UW CSE455
Cost-Volume-based MVS

Multi-view images and camera parameters
Cost-Volume-based MVS

Build 3D cost volume in reference view frustum
Topdown View of Cost Volume
Cost-Volume-based MVS

Fetch images features for each voxel
- Voxel in ground truth surface shows feature consistency
Cost-Volume-based MVS

Dense 3D CNNs
Differentiable soft-argmin to achieve sub-pixel accuracy.

\[ \text{soft argmin} := \sum_{d=0}^{D_{\text{max}}} d \times \sigma(-c_d) \]
Reconstruction is More Complete

Camp [2]  Ours
Agenda

- 3D Classification
- 3D Reconstruction
- Others
It is possible to generate a **set** (permutation invariant)

From Image to Surface

• Learn to warp a plane to surface

Yang, Yaoqing, et al. “Foldingnet: Point cloud auto-encoder via deep grid deformation”, CVPR 2018
Structured Prediction: Part-based

Recursive Network for Hierarchical Graph AE

Li, Jun et al., “GRASS: Generative Recursive Autoencoders for Shape Structures”, Siggraph 2017

Mo, Kaichun et al., “StructureNet, a hierarchical graph network for learning PartNet shape generation”, Siggraph Asia 2019
Structured Prediction: Part-based

Mo et al., “StructureNet, a hierarchical graph network for learning PartNet shape generation”, Siggraph Asia 2019
Many More to Explore...

Movable Part Segmentation

Motion Parameter Estimation

Long-horizon Planning

Part Manipulation