Deep Learning in 3D

• We’ll focus on predicting 3D from one or more image
• Supervision: depth, mesh, silhouettes, view supervision
• Representations: Depth, Points, Meshes, Voxels, SDFs
• Neural Scene Representation and Rendering
3D Representation

• Many ways to represent objects in 3D
Learning in 3D
Is a Different Learning Task

Previous Lectures

Whole-image classification

Object detection

<table>
<thead>
<tr>
<th>airplane</th>
<th>automobile</th>
<th>bird</th>
<th>cat</th>
<th>deer</th>
<th>dog</th>
<th>frog</th>
<th>horse</th>
<th>ship</th>
<th>truck</th>
</tr>
</thead>
</table>

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

[Images of various categories and examples of object detection with bounding boxes and confidence scores.]
Pixel Labelling

- Per-Pixel Regression + Classification, Examples, Architectures
- Depth Estimation: direct vs self supervised, pretraining
- Super-Resolution, Colorization, Image Translation
Pixel vs Image Labelling

- Image labelling, e.g., classification (\(N\) class scores per image)

\[
\begin{align*}
\text{H} & \quad \text{W} & \quad 	ext{C} \\
\rightarrow & \quad \text{CNN} & \rightarrow \\
1 & \quad 1 & \quad \text{N}
\end{align*}
\]

- Pixel labelling, e.g., segmentation, depth estimation, superres, (\(N\) class scores, depth, RGB value etc. per pixel)

\[
\begin{align*}
\text{H} & \quad \text{W} & \quad 	ext{C} \\
\rightarrow & \quad \text{CNN} & \rightarrow \\
\text{H} & \quad \text{W} & \quad \text{N}
\end{align*}
\]

[David Fouhey]
Segmentation

- Predict object identity and/or category per pixel

[ Hu et al 2017 ]
Depth + Normals Estimation

• Predict depth or surface normal per pixel, given RGB input

[ Alhashim Wonka 2019 ]

[ Eigen Fergus 2015 ]
Image Colorization

- Predict color per pixel, given grayscale input

[ Zhang et al. 2016 ]
Super-Resolution

- Predict high resolution RGB, given low resolution RGB input

4 x downsampled  |  bicubic upsample  |  4 x superresolution

real size =  |  1 pixel $\rightarrow$ 16 pixels

[ Ledig et al. 2017 ]
Why Not Stack Convolutions?

$n$ 3x3 convs have a receptive field of $2n+1$ pixels

How many convolutions until $\geq 200$ pixels?

100

[David Fouhey]
Why Not Stack Convolutions?

Suppose 200 3x3 filters/layer, H=W=400
Storage/layer/image: $200 \times 400 \times 400 \times 4$ bytes = 122MB

Uh oh!*

*100 layers, batch size of 20 = 238GB of memory!

[David Fouhey]
Encoder-Decoder

Key idea: First **downsample** towards middle of network. Then **upsample** from middle.

**How do we downsample?**

Convolutions, pooling

[ David Fouhey ]
Putting it Together

Convolutions + pooling downsample/compress/encode
Transpose convs./unpoolings upsample/uncompress/decode

[ David Fouhey ]
Putting It Together – Block Sizes

- Often multiple layers at each spatial resolution.
- Often halve spatial resolution and double feature depth every few layers.
Missing Details

Where is the useful information about the high-frequency details of the image?

Result from Long et al. *Fully Convolutional Networks For Semantic Segmentation*. CVPR 2014
How do you send details forward in the network?
You copy the activations forward. Subsequent layers at the same resolution figure out how to fuse things.
U-Net

Extremely popular architecture, was originally used for biomedical image segmentation.

Transpose conv, bilinear upsample etc.

Ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015”
Single-View Depth Estimation
Single-View Depth Estimation

[ T. Zhou, A. Geiger ]
Single-View Depth Estimation

[ T. Zhou, A. Geiger ]
NYU Depth v2 Dataset

- 400K RGBD frames captured using Microsoft Kinect
- ~1500 have segmentation labels (26 classes) as well
- The dataset has depth holes, note offset between RGB and NIR cameras, and NIR dot projector, also raw RGB + D frames are not synchronized
- Synchronized and filled subset of 50K images by [Alhashim Wonka 2018] — see Project 4 description
- Limited to indoor scenes due to active NIR illumination
NYU Depth Estimation

Direct supervision via Kinect RGB+D

multi-scale architecture

Loss, e.g., L2

[ Eigen Fergus 2015 ]
NYU Depth Estimation

U-Net with skip connections

Direct supervision via Kinect RGB+D

Loss, e.g., L2

Image credit: NYU Dataset, Silberman et al. ECCV 2012
Single-View Depth Estimation

U-Net with skip connections

Loss, e.g., L2

Direct supervision via Kinect RGB+D
2-view Stereo

- Form HxWxD=disparity volume and use 3D convolution

Extract features at each pixel using 2D CNN

Form volume by sliding features from 2nd image at D disparities

Perform 3D convolution on feature volume

Treat output as disparity cost volume and perform soft argmax

https://www.youtube.com/watch?v=VtAzDS1NLmo [ Kendall et al. 2017 ]
End-to-end Deep Stereo Regression Architecture

<table>
<thead>
<tr>
<th>Layer Description</th>
<th>Output Tensor Dim.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input image</strong></td>
<td>$H \times W \times C$</td>
</tr>
<tr>
<td><strong>Unary features (section 3.1)</strong></td>
<td></td>
</tr>
<tr>
<td>1 5×5 conv, 32 features, stride 2</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>2 3×3 conv, 32 features</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>3 3×3 conv, 32 features</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>add layer 1 and 3 features (residual connection)</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>4-17 (repeat layers 2,3 and residual connection) × 7</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>18 3×3 conv, 32 features, (no ReLu or BN)</td>
<td>$\frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td><strong>Cost volume (section 3.2)</strong></td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td><strong>Learning regularization (section 3.3)</strong></td>
<td></td>
</tr>
<tr>
<td>19 From Cost Volume: 3-D conv, 3×3×3, 64 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>20 3-D conv, 3×3×3, 32 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>21 3-D conv, 3×3×3, 64 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>22 3-D conv, 3×3×3, 64 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>23 3-D conv, 3×3×3, 64 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>24 3-D conv, 3×3×3, 64 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>25 3-D conv, 3×3×3, 64 features, (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>26 3-D conv, 3×3×3, 64 features, (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>27 3-D conv, 3×3×3, 64 features, (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>28 3-D conv, 3×3×3, 64 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>29 3-D conv, 3×3×3, 64 features</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>30 3-D conv, 3×3×3, 128 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 4F$</td>
</tr>
<tr>
<td>31 3-D conv, 3×3×3, 128 features, (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 4F$</td>
</tr>
<tr>
<td>32 3-D conv, 3×3×3, 128 features, (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 4F$</td>
</tr>
<tr>
<td>33 3×3×3, 3-D transposed conv, 64 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>add layer 33 and 29 features (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>34 3×3×3, 3-D transposed conv, 64 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>add layer 34 and 26 features (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>35 3×3×3, 3-D transposed conv, 64 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>add layer 35 and 23 features (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times 2F$</td>
</tr>
<tr>
<td>36 3×3×3, 3-D transposed conv, 32 features, stride 2</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>add layer 36 and 20 features (residual connection)</td>
<td>$\frac{1}{2}D \times \frac{1}{2}H \times \frac{1}{2}W \times F$</td>
</tr>
<tr>
<td>37 3×3×3, 3-D trans conv, 1 feature (no ReLu or BN)</td>
<td>$D \times H \times W \times 1$</td>
</tr>
<tr>
<td><strong>Soft argmin (section 3.4)</strong></td>
<td>$H \times W$</td>
</tr>
</tbody>
</table>

[ Kendall et al. 2017 ]
Computing Sub-pixel Disparity

(a) Soft ArgMin
(b) Multi-modal distribution
(c) Multi-modal distribution with prescaling

[ Kendall et al. 2017 ]
Plane Sweep Stereo
(reminder from Lecture 5)
Multi-view Stereo

Compare patches in ref image to plane sweep volumes from other images

Perform intra and inter-volume aggregation of features

[DeepMVS, Huang et al. 2018]
DeepMVS: Results
DeepMVS: Ablation Studies

<table>
<thead>
<tr>
<th>Components</th>
<th>Geo. error</th>
<th>Pho. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretraining</td>
<td>0.051</td>
<td>0.242</td>
</tr>
<tr>
<td>+ U-net</td>
<td>0.043</td>
<td>0.230</td>
</tr>
<tr>
<td>+ U-net + VGG</td>
<td>0.040</td>
<td>0.226</td>
</tr>
<tr>
<td>+ U-net + VGG + DenseCRF</td>
<td><strong>0.036</strong></td>
<td><strong>0.224</strong></td>
</tr>
<tr>
<td>+ U-net + VGG + DenseCRF − MVS-SYNTH</td>
<td>0.037</td>
<td>0.225</td>
</tr>
</tbody>
</table>

[Huang et al. 2018]
DeepMVS: Progressive Improvement

[ Huang et al. 2018 ]
3D Shape Representations: Point Cloud

• Represent shape as a set of P points in 3D space
• (+) Can represent fine structures without huge numbers of points
• (  ) Requires new architecture, losses, etc
• (-) Doesn’t explicitly represent the surface of the shape: extracting a mesh for rendering or other applications requires post-processing

Processing Pointcloud Inputs: PointNet

Input pointcloud: $P \times 3$

Point features: $P \times D$

Run MLP on each point

Max-Pool

Pooled vector: $D$

Fully Connected

Class score: $C$

Want to process pointclouds as sets: order should not matter

Qi et al, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”, CVPR 2017
Processing Mesh (and PointCloud): FeaStNet

[ Verma, Boyer, and Verbeek CVPR 2018 ]
FeaStNet: Problem Statement

Vertex-labeling problem:

Reference shape: 6,980 vertices

Let each vertex in the reference shape be its own class (label).

\[ Y = \{0, ..., 6980-1\} \]

For the target shape (on the right), label each vertex using \( Y \)

[ Verma, Boyer, and Verbeek CVPR 2018 ]
Generalized Convolution

← convolution on the image lattice

convolution on an arbitrary graph topology

[ Verma, Boyer, and Verbeek CVPR 2018 ]
\[ y_i = b + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} q_m(x_i, x_j) W_m x_j, \]

\[ q_m(x_i, x_j) \propto \exp \left( u_m^\top x_i + v_m^\top x_j + c_m \right), \]

with \( \sum_{m=1}^{M} q_m(x_i, x_j) = 1, \)

The only additional parameters w.r.t. a conventional CNN are the vectors \( u_m, v_m, \) which contain \( 2MD \) parameters.

[ Verma, Boyer, and Verbeek CVPR 2018 ]
3D Datasets: Object-Centric

ShapeNet

~50 categories, ~50k 3D CAD models
Standard split has 13 categories, ~44k models, 25 rendered images per model
Many papers show results here
(-) Synthetic, isolated objects; no context
(-) Lots of chairs, cars, airplanes


Pix3D

9 categories, 219 3D models of IKEA furniture aligned to ~17k real images
Some papers train on ShapeNet and show qualitative results here, but use ground-truth segmentation masks
(+ ) Real images! Context!
(-) Small, partial annotations – only 1 obj/image

Sun et al, “Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling”, CVPR 2018
3D Shape Prediction: Mesh R-CNN

Mask R-CNN: 2D Image -> 2D shapes

Mesh R-CNN: 2D Image -> Triangle Meshes

Detect objects and extract silhouettes
Estimate 3D mesh

He, Gkioxari, Dollár, and Girshick, “Mask R-CNN”, ICCV 2017
Gkioxari, Malik, and Johnson, “Mesh R-CNN”, ICCV 2019
There Is More To Do in 3D

DeepVoxels

- Embedding vector per voxel

Observations → Neural Scene Representation → Neural Renderer → Re-Rendered Observations

Image Loss

Scene represented as an embedding vector per 3D point

DeepSDF

- CPPN for signed distance function, SDF=f(X)

[ Slides: Jeong Joon Park ]

Neural Radiance Fields

- Another continuous scene representation using a FCN

Predict density at each location, integrate along ray to get color (volume rendering)
We’ve Reached the End of the Class

But there is so much more to computer vision!

Stay in touch: vxa@uw.edu