Understanding Real World Geometry

Ankur Handa

University of Cambridge and Imperial College London

July 25, 2016
Outline

- Background
- SceneNet: Repository of Labelled Synthetic Indoor Scenes
- Results on Semantic Segmentation on Real World Data
- gvnn: Neural Network Library for Geometric Vision
- Future Ideas
Our initial goal was to segment 3D scenes in real-time. Overlay the per-pixel segmentations onto the 3D map. We are only interested in depth based semantic segmentation here mainly to understand the role of geometry.
Background

How do we get enough training data?

- Real world datasets are limited in size e.g. NYUv2 and SUN RGB-D have 795 and 5K images for training respectively.
- We can leverage computer graphics to generate the desired data.
- There are lots of CAD repositories of objects available but none contains scenes.
- We put together a repository of labelled 3D scenes called SceneNet and train per-pixel segmentation on synthetic data.
Repository of labelled 3D synthetic scenes.

Repository of Labelled Synthetic Indoor Scenes. Make sure you have WebGL enabled in your browser to view the 3D models. We are also looking into generating unlimited data in the form of 3D scenes with our simulated annealing algorithm. We have also automated texturing of these scenes using archivetextures and opensurfaces.

We are increasingly seeing the use of these scenes beyond standard computer vision problems e.g. semantic segmentation, optic flow, 3D scene reconstruction etc. to now physical scene understanding and Deep Reinforcement Learning with agents interacting with their 3D environments.

Work in Progress https://github.com/ankurhanda/SceneNetv1.0

Hosted at http://robotvault.bitbucket.org
SceneNet Basis Scenes

<table>
<thead>
<tr>
<th>Room Type</th>
<th>Number of Scenes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living Room</td>
<td>10</td>
</tr>
<tr>
<td>Bedroom</td>
<td>11</td>
</tr>
<tr>
<td>Office</td>
<td>15</td>
</tr>
<tr>
<td>Kitchens</td>
<td>11</td>
</tr>
<tr>
<td>Bathrooms</td>
<td>10</td>
</tr>
</tbody>
</table>

In total we have 57 very detailed scenes with about 3700 objects. We build upon these scenes to create unlimited number of scenes later for large scale training data.
A very detailed living room. Each object in this scene is labelled directly in 3D.
dimensions: 27×25×2.7 m³
Generating indoor scenes as energy minimisation problem

- **Bounding box constraint.**
  - Each object must maintain a safe distance from the other.

- **Pairwise Constraint.**
  - Objects that co-occur should not be more than a fixed distance from each other, *e.g.* beds and nightstands

- **Visibility Constraint.**
  - Objects with visibility constraint must not have anything joining their line of sight joining their centers, *e.g.* sofa, table and TV must not having anything in between them.

- **Distance to Wall Constraint.**
  - Some objects are more likely to occur next to walls *e.g.* cupboards, sofa etc.

- Use simulated annealing to solve this energy minimisation problem.
Generating Unlimited Number of Synthetic Scenes

Co-occurrence statistics of objects from NYUv2 bedrooms.

Interpretations

- Heat-map of object co-occurrence.
- Training set in NYUv2 is used to obtain these statistics.
- Shows that bed, picture, pillow and nightstand co-occur together.
- We sample new scenes from these object co-occurrence statistics.
Generating Unlimited Number of Synthetic Scenes

Constraints

- Objects that co-occur in real world should be placed together, e.g. beds and nightstands.
- Visibility constraint - nothing in the line of sight between sofa-table and TV.
Generating Unlimited Number of Synthetic Scenes

Very simple scenes with curtains.
Generating Unlimited Number of Synthetic Scenes

Common room obtained from hierarchically optimising chairs and table combination.
Generating Unlimited Number of Synthetic Scenes

Living Room. Note that the inset objects are from Stanford Scenes and have been grouped together already.
Generating Unlimited Number of Synthetic Scenes

Living Room. Grouped objects from Stanford Scenes.
Generating Unlimited Number of Synthetic Scenes

Data Collection

- Collecting trajectories with joy-stick (though we have randomised them now).
- Using OpenGL glReadPixels to obtain the depth as well as ground truth labels.
Generating Unlimited Number of Synthetic Scenes

Assumptions

Here we only study the segmentation from just depth-data in the form of DHA images i.e. depth, height from ground plane and angle with gravity vector.

Allows us to study the effects of geometry in isolation. Texturing takes time and needs careful photorealistic synthesis. But we have added a custom raytracer to study RGB-D based segmentation now.
SegNet: [Badrinarayanan, Handa, and Cipolla, arXiv 2015]

- Saves pooling indices explicitly in the encoder and passes on to the decoder for upsampling.
Adapting synthetic data to match real world data

- Synthetic depth-maps are injected with appropriate noise models.
- We then denoise these depth-maps to ensure that input images look similar to the NYUv2 dataset.
- Side by side comparison of synthetic bedroom with one of the NYUv2 bedrooms.
Results on Semantic Segmentation on Real World Data

NYUv2 test images
Results on Semantic Segmentation on Real World Data

Ground truth segmentation
Results on Semantic Segmentation on Real World Data

Training on NYUv2 training set only

Ankur Handa
Understanding Real World Geometry
Results on Semantic Segmentation on Real World Data

Training on SceneNet and fine-tuning on NYUv2
Results on Semantic Segmentation on Real World Data

Training on NYUv2 only

Training on SceneNet and fine-tuning on NYUv2

What we learnt? [ICRA 2016, CVPR 2016]

- Computer graphics is a great source for collecting data.
- We need to work on designing or learning new loss functions.
- A loss function that takes local and global context together to avoid any objects mislabelled as others.

Ankur Handa
Understanding Real World Geometry
Results on Semantic Segmentation on Real World Data

What we learnt? [ICRA 2016, CVPR 2016]

- We perform better than state-of-the-art on functional categories of objects
- Suggesting shape is a strong cue for segmentation.
- We fall behind on objects that explicitly need RGB data.

<table>
<thead>
<tr>
<th>11 class semantic segmentation: NYUv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
</tr>
<tr>
<td>SceneNet-DHA</td>
</tr>
<tr>
<td>NYU-DHA</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DHA</td>
</tr>
<tr>
<td>SceneNet-DO-DHA</td>
</tr>
<tr>
<td>NYU-DO-DHA</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DO-DHA</td>
</tr>
<tr>
<td>Eigen et al. (rgbd+normals) [12]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>bed</th>
<th>ceil.</th>
<th>chair</th>
<th>floor</th>
<th>furn</th>
<th>obj.</th>
<th>sofa</th>
<th>table</th>
<th>tv</th>
<th>wall</th>
<th>window</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYU-DHA</td>
<td>64.5</td>
<td>68.2</td>
<td>51.0</td>
<td>95.0</td>
<td>51.0</td>
<td>48.2</td>
<td>49.7</td>
<td>41.9</td>
<td>12.8</td>
<td>84.2</td>
<td>24.5</td>
</tr>
<tr>
<td>SceneNet-DHA</td>
<td>60.8</td>
<td>43.4</td>
<td>68.5</td>
<td>90.0</td>
<td>26.5</td>
<td>24.3</td>
<td>21.2</td>
<td>42.1</td>
<td>0</td>
<td>92.1</td>
<td>0.3</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DHA</td>
<td>70.3</td>
<td>75.9</td>
<td>59.8</td>
<td>96.0</td>
<td><strong>60.7</strong></td>
<td>49.7</td>
<td>59.9</td>
<td><strong>49.7</strong></td>
<td>24.3</td>
<td>84.8</td>
<td>27.9</td>
</tr>
<tr>
<td>NYU-DO-DHA</td>
<td>69.0</td>
<td>74.6</td>
<td>54.0</td>
<td>95.6</td>
<td>57.1</td>
<td>48.7</td>
<td>55.7</td>
<td>42.5</td>
<td>18.5</td>
<td>84.7</td>
<td>25.5</td>
</tr>
<tr>
<td>SceneNet-DO-DHA</td>
<td>67.7</td>
<td>49.9</td>
<td>67.5</td>
<td>87.8</td>
<td>38.6</td>
<td>22.6</td>
<td>15.8</td>
<td>44.2</td>
<td>0</td>
<td>89.0</td>
<td>0.8</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DO-DHA</td>
<td>72.5</td>
<td>74.1</td>
<td>61.0</td>
<td>96.2</td>
<td>60.4</td>
<td>50.0</td>
<td><strong>62.8</strong></td>
<td>43.8</td>
<td>19.4</td>
<td>85.3</td>
<td>30.0</td>
</tr>
<tr>
<td>Eigen et al. (rgbd+normals) [12]</td>
<td>61.1</td>
<td>78.3</td>
<td><strong>72.1</strong></td>
<td><strong>96.5</strong></td>
<td>55.1</td>
<td><strong>52.1</strong></td>
<td>45.8</td>
<td>45.0</td>
<td><strong>41.9</strong></td>
<td>88.7</td>
<td><strong>57.7</strong></td>
</tr>
</tbody>
</table>
What we learnt? [ICRA 2016, CVPR 2016]

- We perform better than state-of-the-art on functional categories of objects.
- Suggesting shape is a strong cue for segmentation.
- We fall behind on objects that explicitly need RGB data.

### 13 class semantic segmentation: NYUv2

<table>
<thead>
<tr>
<th>Training</th>
<th>global acc.</th>
<th>class acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hermans et al. (rgbd+crf) [22]</td>
<td>54.2</td>
<td>48.0</td>
</tr>
<tr>
<td>SceneNet-DHA</td>
<td>54.4</td>
<td>37.1</td>
</tr>
<tr>
<td>NYU-DHA</td>
<td>63.6</td>
<td>47.2</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DHA</td>
<td>66.5</td>
<td>51.7</td>
</tr>
<tr>
<td>SceneNet-DO-DHA</td>
<td>55.3</td>
<td>37.6</td>
</tr>
<tr>
<td>NYU-DO-DHA</td>
<td>65.0</td>
<td>48.6</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DO-DHA</td>
<td>67.2</td>
<td>52.5</td>
</tr>
<tr>
<td>Eigen et al. (rgbd+normals) [12]</td>
<td><strong>68.0</strong></td>
<td><strong>60.8</strong></td>
</tr>
</tbody>
</table>

### NYU-DHA

<table>
<thead>
<tr>
<th></th>
<th>bed</th>
<th>books</th>
<th>ceil.</th>
<th>chair</th>
<th>floor</th>
<th>form</th>
<th>obj.</th>
<th>paint</th>
<th>sofa</th>
<th>table</th>
<th>tv</th>
<th>wall</th>
<th>window</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYU-DHA</td>
<td>67.7</td>
<td>6.5</td>
<td>69.9</td>
<td>47.9</td>
<td><strong>96.2</strong></td>
<td>53.8</td>
<td>46.5</td>
<td>11.3</td>
<td>50.7</td>
<td>41.6</td>
<td>10.8</td>
<td>85.0</td>
<td>25.8</td>
</tr>
<tr>
<td>SceneNet-DHA</td>
<td>60.8</td>
<td>2.0</td>
<td>44.2</td>
<td>68.3</td>
<td>90.2</td>
<td>26.4</td>
<td>27.6</td>
<td>6.3</td>
<td>21.1</td>
<td>42.2</td>
<td>0</td>
<td><strong>92.0</strong></td>
<td>0.0</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DHA</td>
<td>70.8</td>
<td>5.3</td>
<td>75.0</td>
<td>58.9</td>
<td>95.9</td>
<td>63.3</td>
<td>48.4</td>
<td>15.2</td>
<td>58.0</td>
<td>43.6</td>
<td>22.3</td>
<td>85.1</td>
<td>29.9</td>
</tr>
<tr>
<td>NYU-DO-DHA</td>
<td>69.6</td>
<td>3.1</td>
<td>69.3</td>
<td>53.2</td>
<td>95.9</td>
<td>60.0</td>
<td>49.0</td>
<td>11.6</td>
<td>52.7</td>
<td>40.2</td>
<td>17.3</td>
<td>85.0</td>
<td>27.1</td>
</tr>
<tr>
<td>SceneNet-DO-DHA</td>
<td>67.9</td>
<td>4.7</td>
<td>41.2</td>
<td>67.7</td>
<td>87.9</td>
<td>38.4</td>
<td>25.6</td>
<td>6.3</td>
<td>16.3</td>
<td>43.8</td>
<td>0</td>
<td>88.6</td>
<td>1.0</td>
</tr>
<tr>
<td>SceneNet-FT-NYU-DO-DHA</td>
<td><strong>70.8</strong></td>
<td>5.5</td>
<td>76.2</td>
<td>59.6</td>
<td>95.9</td>
<td><strong>62.3</strong></td>
<td><strong>50.0</strong></td>
<td>18.0</td>
<td><strong>61.3</strong></td>
<td>42.2</td>
<td>22.2</td>
<td>86.1</td>
<td>32.1</td>
</tr>
<tr>
<td>Eigen et al. (rgbd+normals) [12]</td>
<td>61.1</td>
<td><strong>49.7</strong></td>
<td>78.3</td>
<td><strong>72.1</strong></td>
<td>96.0</td>
<td>55.1</td>
<td>40.7</td>
<td><strong>58.7</strong></td>
<td>45.8</td>
<td><strong>44.9</strong></td>
<td><strong>41.9</strong></td>
<td>88.7</td>
<td>57.7</td>
</tr>
<tr>
<td>Hermans et al. (rgbd+crf) [22]</td>
<td>68.4</td>
<td>N/A</td>
<td><strong>83.4</strong></td>
<td>41.9</td>
<td>91.5</td>
<td>37.1</td>
<td>8.6</td>
<td>N/A</td>
<td>28.5</td>
<td>27.7</td>
<td>38.4</td>
<td>71.8</td>
<td>46.1</td>
</tr>
</tbody>
</table>
Results on Semantic Segmentation on Real World Data

What we learnt? [ICRA 2016, CVPR 2016]

- We perform better than state-of-the-art on functional categories of objects
- Suggesting shape is a strong cue for segmentation.
- We fall behind on objects that explicitly need RGB data.
Adding Physics

- Allows us to create scenes with arbitrary clutter.
- Placing pens and other small objects on tables is relatively easy with physics.
- We don’t want chairs always in upright positions. Physics allows us to explore the space of poses of objects relatively easy.
SceneNet and Raytracing

Low-res on laptop | High quality

Adding Raytracing

- Sufficient Photorealism is desirable.
- Depth sensors have limited range.
- RGB is universal and allows us to model various different cameras.
- Need good approximation of shadows, lighting, and various other global lighting artefacts.
What is gvnn?

- A new library inspired by Spatial Transformer Networks (STN).
- Includes various different transformations often used in geometric computer vision.
- These transformations are implemented as new layers that allow backpropagation as in STN.
- Brings together the domain knowledge in geometry within the neural network.
Different new layers in gvnn

What is gvnn?

- Original Spatial Transformer (NIPS 2015) had 2D transformations mainly.
- Added SO3, SE3 and Sim3 layers for global transformation on the image.
- Optical flow, over-parameterised Optical flow, Slanted plane disparity.
- Pin-Hole Camera Projection layer.
- Per-pixel SO3, SE3 and Sim3 for non-rigid registration.
- Different robust M-estimators.
- Very useful for 3D alignment, unsupervised warping with optic flow or disparity and geometric invariance for place recognition.
Global Transformations in gvn

SO3 Layer (SE3 and Sim3 follow easily from here)

\[
\frac{\partial C}{\partial v} = \frac{\partial C}{\partial R(v)} \cdot \frac{\partial R(v)}{\partial v} \tag{1}
\]

\[
\frac{\partial R(v)}{\partial v_i} = \frac{v_i[v]_x + [v \times (I - R)e_i]_x}{||v||^2} R \tag{2}
\]

Notes

- \( C \) is the cost function being minimised.
- \( v = (v_1, v_2, v_3) \) is the SO3 vector. Note the derivative is not at \( v_i \approx 0 \) and \( e_i \) is the \( i^{th} \) column of the Identity matrix.
Per-pixel 2D Transformations in gvnn

Optical flow and Over-parameterised Optical Flow

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix}
= 
\begin{pmatrix}
  x + t_x \\
  y + t_y
\end{pmatrix}
\tag{3}
\]

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix}
= 
\begin{pmatrix}
  a_0 & a_1 & a_2 \\
  a_3 & a_4 & a_5
\end{pmatrix}
\begin{pmatrix}
  x \\
  y \\
  1
\end{pmatrix}
\tag{4}
\]

Notes

- Over-parameterised optical flow also needs extra regularisation.
Per-pixel 2D Transformations in gvnn

Disparity and Slated Plane disparity

\[
\begin{pmatrix}
  x' \\
y'
\end{pmatrix}
= \begin{pmatrix}
x + d \\
y
\end{pmatrix}
\]

\[d = ax + by + c\]  

Notes

- Fitting slanted planes at each pixel.
- Again, very useful for warping images (and unsupervised learning), Garg et al. ECCV2016.
Camera Projection Layer in gvnn

Pin Hole Camera Projection Layer

\[
\pi \left( \begin{array}{c} u \\ v \\ w \end{array} \right) = \left( \begin{array}{c} f_x \frac{u}{w} + p_x \\ f_y \frac{v}{w} + p_y \end{array} \right)
\]  \hspace{1cm} (7)

\[
\frac{\partial C}{\partial p} = \frac{\partial C}{\partial \pi(p)} \cdot \frac{\partial \pi(p)}{\partial p}
\]  \hspace{1cm} (8)

\[
\frac{\partial \pi}{\partial \left( \begin{array}{c} u \\ v \\ w \end{array} \right)} = \left( \begin{array}{ccc} f_x \frac{1}{w} & 0 & -f_x \frac{u}{w^2} \\ 0 & f_y \frac{1}{w} & -f_y \frac{v}{w^2} \end{array} \right)
\]  \hspace{1cm} (9)
Non-rigid per-pixel transformation

Non-rigid registration for point clouds: 6DoF and 10DoF

Per-pixel 6DoF and 10DoF transformations.

\[
\begin{pmatrix}
x'_i \\
y'_i \\
z'_i \\
\end{pmatrix} = T_i \begin{pmatrix}
x_i \\
y_i \\
z_i \\
1 \\
\end{pmatrix} \tag{10}
\]

\[x'_i = s_i(R_i(x_i - p_i) + p_i) + t_i \tag{11}\]

Also useful for volumetric spatial transformers when using voxel grid representation.
M-estimators

Notes

- Different M-estimators for robust outlier rejection.
- Very useful when doing regression.
- Implemented as layers in gvnn.
Sanity Checks on End-to-End Visual Odometry (SO3)

Notes

- General dense image registration with global transformation.
- Can use optical flow/disparity for dense per-pixel image registration either with a CNN or RNN.
- Initial experiments with supervised learning.
Notes

- General dense image registration with global transformation.
- Can use optical flow/disparity for dense per-pixel image registration either with a CNN or RNN.
Sanity Checks on End-to-End Visual Odometry (SO3)

Notes

- General dense image registration with global transformation.
- Can use optical flow/disparity for dense per-pixel image registration either with a CNN or RNN.
General dense image registration with global transformation.

Can use optical flow/disparity for dense per-pixel image registration either with a CNN or RNN.
Sanity Checks on End-to-End Visual Odometry (SO3)

Notes

- General dense image registration with global transformation.
- Can use optical flow/disparity for dense per-pixel image registration either with a CNN or RNN.
- Aim to train on large scale data from SceneNet and RGB videos for unsupervised learning.
\[ \delta_{\text{update}} = f_{\text{siamese}}(I^k_t, I_{t+1}) \] (12)

\[ \delta_k = \delta_{k-1} + \delta_{\text{update}} \] (13)

\[ \hat{I}^k_t = f_{\text{warping}}(I_t, \delta_k) \] (14)

\[ \hat{I}^0_t = I_t \] (15)

\[ \delta_0 = 0 \] (16)
Notes

- Needs explicitly the depth to do warping.
- This can either come from a sensor or an extra CNN/RNN module that learns depth.
Sanity Checks on End-to-End Visual Odometry (SE3)

(a) Prediction  
(b) Ground Truth  
(c) Residual (difference)
Where can deep learning help geometry?

- CNNs provide relatively stable features for images of same scene taken across different lighting conditions.
- These images cannot be aligned with geometry based per-pixel dense image alignment methods.
- We can put this together with change detection segmentation to also reason about dynamic scenes.
- Easy to collect data with SceneNet.
Future Ideas

- Physical scene understanding with SceneNet to reason about how the dynamics of the scene is going to evolve over time.
- Understanding dynamic scenes - change detection for persistent 4D mapping.
  - Imagine two images of the scene taken at different times of the day from relatively similar positions.
  - STN-SE3 could be used to align the images.
  - Aligned images can be fed into a segmentation network that gives per-pixel change mask.
  - Joint end-to-end training.
- Attend-Infer-Repeat style 3D scene understanding in the form of instances of objects and their 6DoF poses even for cluttered scenes, with recurrent neural networks.
References

- **SceneNet: an Annotated Model Generator for Indoor Scene Understanding**, Ankur Handa, Viorica Patraucean, Simon Stent, Roberto Cipolla, ICRA 2016
- **Understanding Real World Indoor Scenes with Synthetic Data**, Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, Roberto Cipolla, CVPR 2016
Thank you

- Vijay Badrinarayanan, Viorica Patraucean, Simon Stent, and Roberto Cipolla, University of Cambridge
- Zoubin Ghahramani, University of Cambridge
- John McCormac and Andrew Davison, Imperial College London
- Daniel Cremers, TUM Munich

Thank you for listening.