Geometry-Aware Neural Rendering

Josh Tobin, OpenAI Robotics, Pieter Abbeel
How to model complex robots scenes?

From... ...To
Model the state of all objects?

- Scales with scene complexity
- How to deal with complex internal state?
- How to deal with out-of-distribution?
Only use the state implicitly?

- Can be data inefficient
- May require learning from scratch (which can be dangerous)
- Often lacks reusability
Model the 3D structure of the scene?

- High-dimensional representations - scale poorly with scene detail
- No notion of semantics
Model the 3D structure implicitly?

The Neural Rendering problem

Random viewpoints

Model

Arbitrary “query” viewpoint
Motivation: Generative Query Nets

Motivation: Generative Query Nets

Key questions

• Can it scale to high(er)-dimensional images (GQN is 64x64)?
• Does it work for objects with complex state?
• Does it work for a wide range of realistic objects?
• Is it useful for downstream robotics tasks?
Potential limitations

- Scene representation is a sum — each feature contains only local information
- Rendering process cannot interact with the full representations (except through backprop)
Background: Epipolar Geometry
Our Approach

Our Model

Representation network

Generation network

Attention mechanism

Previous Method

Ground Truth

Josh Tobin

Geometry-Aware Neural Rendering
Epipolar extraction
Attention mechanism

![Diagram of attention mechanism]

$Q_i, K^k, V^k$ are the query, key, and value tensors, respectively. $h' \times w' \times d_k$ is the dimensionality of these tensors. $e^k$ and $h_{l-1}$ are intermediate representations. The diagram shows the attention mechanism applied to these representations.
Does it help?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Mean Absolute Error (pixels)</th>
<th>Root Mean Squared Error (pixels)</th>
<th>ELBO (nats / dim)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GQN</td>
<td>E-GQN</td>
<td>GQN</td>
</tr>
<tr>
<td>rrc</td>
<td>7.40 ± 6.22</td>
<td>3.59 ± 2.10</td>
<td>0.5637 ± 0.0013</td>
</tr>
<tr>
<td>rfc</td>
<td>12.44 ± 12.89</td>
<td>12.05 ± 12.79</td>
<td>0.5637 ± 0.0011</td>
</tr>
<tr>
<td>jaco</td>
<td>4.30 ± 1.12</td>
<td>4.00 ± 0.90</td>
<td>0.5634 ± 0.0007</td>
</tr>
<tr>
<td>sm7</td>
<td>3.13 ± 1.30</td>
<td>2.14 ± 0.53</td>
<td>0.5637 ± 0.0009</td>
</tr>
<tr>
<td>oab</td>
<td>10.99 ± 5.13</td>
<td>5.47 ± 2.54</td>
<td>1.2587 ± 0.0018</td>
</tr>
<tr>
<td>disco</td>
<td>18.86 ± 7.16</td>
<td>12.46 ± 9.27</td>
<td>1.2635 ± 0.0055</td>
</tr>
<tr>
<td>rro</td>
<td>10.12 ± 5.15</td>
<td>6.59 ± 3.23</td>
<td>1.2573 ± 0.0011</td>
</tr>
</tbody>
</table>
Examples
Examples

Context

GQN

E-GQN

Target

Josh Tobin

Geometry-Aware Neural Rendering
Examples

Context

GQN

E-GQN

Target
Conclusion

• Geometrically-inspired neural network primitives improve implicit 3D understanding

• Forcing the model to understand geometry can improve downstream robotic tasks

• How to go from this to general 3D understanding?