Embodied Visual Recognition

Katerina Fragkiadaki

Carnegie Mellon University
Internet Vision

Internet pictures:
- Regular sizes, viewpoints, centered objects
- Every image tells a story
- An intelligent agent has taken the picture
Mobile Computer Vision

Drone/ground robot videos:
- Dramatic size, viewpoint variations, lots of occlusions
- An image does not suffice for scene understanding
- An untrained agent has taken the video
2D CNNs do not have common sense

- No object permanence: objects disappear at occlusions
- Objects “move” with camera motion
- Objects change size during camera zoom in / zoom out motion
- Objects appear as 2D surfaces as opposed to having 3D extent
3D representations have some common sense

- Object permanence: objects do not disappear at occlusions
- Scene and camera motion are disentangled
- Objects do not change size during camera zoom in / zoom out motion
- Objects have 3D extent
but...

- they are hard to obtain in detail
- they discard semantic information
- they need multiple views
- they cannot handle dynamic scenes
``Internal world models which are complete representations of the external environment, besides being impossible to obtain, are not at all necessary for agents to act in a competent manner.”

*Intelligence without reason*, IJCAI, Rodney Brooks (1991)
To 3D or not to 3D?
3D feature maps

$H \times W \times D \times C$

3 spatial dimensions, multiple feature dimensions
3D feature maps

$H \times W \times D \times C$

3 spatial dimensions, multiple feature dimensions
3D feature maps

$H \times W \times D \times C$

3 spatial dimensions, multiple feature dimensions
3D feature maps

$H \times W \times D \times C$

3 spatial dimensions, multiple feature dimensions
This talk

- Visual recognition under arbitrary camera motion
  (what we can do for embodied vision)

- Learning to see by moving and watching objects move
  (what embodied vision can do for unsupervised visual feature learning)

- Grounding language to visual representations learnt by embodiment
  (what embodied vision can do for language understanding)
1. Hidden state: geometrically consistent 3D feature maps
2. Egomotion-stabilized hidden state updates
Unprojection (2D to 3D)
Unprojection (2D to 3D)
Unprojection (2D to 3D)
Unprojection (2D to 3D)
Unprojection (2D to 3D)
Rotation

azimuth
elevation
Egomotion-stabilized memory update

3D feature memory

Relative Rotation $R$

cross convolution

Unprojection

Rotation
Egomotion-stabilized memory update

\[ h_t \xrightarrow{\text{Hidden state update}} h_{t+1} \]

Unprojection

Rotation $-R$
Projection (3D to 2D)
Projection (3D to 2D)
Projection (3D to 2D)
Projection (3D to 2D)
Projection (3D to 2D)
Geometry-Aware Recurrent Networks (GRNNs)

$H \times W \times D \times C$
Geometry-Aware Recurrent Networks (GRNNs)

$H \times W \times D \times C$
Geometry-Aware Recurrent Networks (GRNNs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)
2D RNNs (conv-LSTMs/GRUs)

\[ h_t \quad \rightarrow \quad h_{t+1} \quad \rightarrow \quad h_{t+2} \]

CNN

\[ t \]
2D RNNs (conv-LSTMs/GRUs)
Geometry-Aware Recurrent Networks
Geometry-Aware Recurrent Networks

CNN

CNN

CNN
Geometry-Aware Recurrent Networks

$h_t$

CNN

CNN

CNN

$\text{CNN}$

$\text{CNN}$

$\text{CNN}$

$t$
Geometry-Aware Recurrent Networks
Geometry-Aware Recurrent Networks

$h_t$

$R_t, T_t$

egomotion

CNN

CNN

CNN

$t$
Geometry-Aware Recurrent Networks

\[ h_t \xrightarrow{R_t, T_t} h_{t+1} \]

egomotion

\[ \text{CNN} \]
Geometry-Aware Recurrent Networks

$h_t$ → $h_{t+1}$

$R_t, T_t$ → $R_{t+1}, T_{t+1}$

CNN → CNN → CNN

t
Geometry-Aware Recurrent Networks

$h_t$, $h_{t+1}$, $h_{t+2}$

$R_t, T_t$, $R_{t+1}, T_{t+1}$

egomotion

CNN

$t$
Geometry-Aware Recurrent Networks

$h_t$ → $h_{t+1}$ → $h_{t+2}$

$R_t, T_t$ → egomotion → CNN → $h_t$

$R_{t+1}, T_{t+1}$ → egomotion → CNN → $h_{t+1}$

$R_{t+2}, T_{t+2}$ → egomotion → CNN → $h_{t+2}$
Geometry-Aware Recurrent Networks

$h_t$ $\rightarrow$ $h_{t+1}$ $\rightarrow$ $h_{t+2}$

$R_t, T_t$ $\rightarrow$ $R_{t+1}, T_{t+1}$

egomotion

CNN

$R_{t+1}, T_{t+1}$

egomotion

CNN

$R_{t+1}, T_{t+1}$

egomotion

CNN

$t$
Geometry-Aware Recurrent Networks

$h_t \xrightarrow{R_t, T_t} h_{t+1} \xrightarrow{R_{t+1}, T_{t+1}} h_{t+2}$

egomotion

CNN

CNN

CNN
Geometry-Aware Recurrent Networks

\[ h_t \rightarrow R_t, T_t \rightarrow h_{t+1} \rightarrow R_{t+1}, T_{t+1} \rightarrow h_{t+2} \]

\[ \text{egomotion} \]

CNN

\[ t \]
Training GRNNs

1. **Self-supervised** for view prediction
2. **Supervised** for 3D object detection
Training GRNNs

1. Self-supervised for view prediction
2. Supervised for 3D object detection
View prediction

rotate to query view

project
Results - view prediction

1. Neural scene representation and rendering DeepMind, Science, 2018
Results - view prediction

1. Neural scene representation and rendering DeepMind, Science, 2018
<table>
<thead>
<tr>
<th>Input views</th>
<th>GRNNs</th>
<th>GQN [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Input views" /></td>
<td><img src="image2" alt="GRNNs" /></td>
<td><img src="image3" alt="GQN" /></td>
</tr>
</tbody>
</table>

1. *Neural scene representation and rendering* DeepMind, Science, 2018
Results - view prediction

Geometry-aware RNN

GQN [1]

1. Neural scene representation and rendering DeepMind, Science, 2018
Results - view prediction

# of input views
Results - view prediction

# of input views
Results - view prediction

# of input views
Training GRNNs

1. Self-supervised for view prediction
2. Supervised for 3D object detection
3D Object Detection

**Input:** the 3D latent feature map

**Output:** 3D boxes and segmentations for the objects
Results - 3D object detection

# of input views
Results - 3D object detection

# of input views
Results - 3D object detection

# of input views
Common sense emerges

- Objects persist over time, objects have 3D extent, camera motion is disentangled from scene appearance
Embodied visual recognition

- Can view prediction work beyond the toy simulation worlds we have just showed?
- Can view prediction learn features useful for object detection?

Yes, with a change in the loss function...
GRNNs in CARLA

View prediction

3D feature memory

Unprojection

3D object detection

Estimated egomotion

R, T
View-contrastive prediction

Views 1…K

Egomotion estimation

3D ML loss

2D ML loss

View K+1

Embodied View-Contrastive 3D Feature Learning, Harley et al., arxiv
View-contrastive prediction

Target view

RGB estimates
View-contrastive prediction

Target view

Embeddings
View-contrastive prediction

*Embodied View-Contrastive 3D Feature Learning*, Harley et al., arxiv
View-contrastive GRNN training helps 3D object detection
3D object detection in the CARLA simulator

![Graph showing mean average precision vs dataset size]

- View-contrastive pretraining
- View regression pretraining
- Random weight initialization

Embodied View-Contrastive 3D Feature Learning, Harley et al., arxiv
CARLA-to-KITTI transfer

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP@IOU</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.33</td>
<td>0.50</td>
<td>0.75</td>
</tr>
<tr>
<td>No pre-training</td>
<td>.59</td>
<td>.52</td>
<td>.17</td>
</tr>
<tr>
<td>Gen. pret., frozen</td>
<td>.64</td>
<td>.54</td>
<td>.15</td>
</tr>
<tr>
<td>Gen. pret.</td>
<td>.65</td>
<td>.55</td>
<td>.18</td>
</tr>
<tr>
<td>Con. pret., frozen</td>
<td>.67</td>
<td>.58</td>
<td>.15</td>
</tr>
<tr>
<td>Con. pret.</td>
<td><strong>.70</strong></td>
<td><strong>.60</strong></td>
<td><strong>.19</strong></td>
</tr>
</tbody>
</table>

Table 1: 3D object detection on KITTI.
3D objects emerge without any annotations
Static scenes
Dynamic scenes

R, T
3D imagination flow captures motion of the dynamic part of the scene only, since the maps have been transformed to cancel ego-motion.
3D object discovery

Common fate: Center-surround score based on 3D motion content

Embodied View-Contrastive 3D Feature Learning, Harley et al., arxiv
3D imagination flow

R, T
3D feature representations for language grounding
People can infer affordability of utterances.

- “He used the newspaper to protect his face from the wind.”
- “He used the matchbox to protect his face from the wind.”

Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning, Glenberg and Robertson, 2000
People can answer million questions regarding the described situation.

“He used the newspaper to protect his face from the wind.”

• How many free hands the man has?
• Is the newspaper in front or behind his eyes?
• Can the newspaper be a single page?
• Is he holding the newspaper?
• Is he lying on top of the newspaper?
• Is the newspaper protecting also his neck from the wind? His feet?
People can follow natural language instructions: "put the pen in front of the book"
Computational models of language and vision

...cannot answer *basic* questions

Where is the child sitting?
- fridge
- arms

Where are the arms sitting? Can the fridge door close? Can a baby hold two bottles? Can a baby hold three bottles? Does a baby disappear when mom walks in front? Is mom or baby taller?
Learn to associate natural language utterances with 3D feature representations of the scene described.
1. We consider an embodied agent that can see a scene from multiple viewpoints

“The green rubber cylinder is on the right of the blue bowl”
1. We consider an embodied agent that can see a scene from multiple viewpoints.

“The green rubber cylinder is on the right of the blue bowl”
2. Our agent learns to map an RGB image to a set of 3D feature maps by training GRNNs to predict views.

“The green rubber cylinder is on the right of the blue bowl”
3. Our agent maps noun phrases to object-centric 3D feature maps (we assume 3D object boxes available at training time)

"The green rubber cylinder is on the right of the blue bowl"
3. Our agent maps noun phrases to object-centric 3D feature maps (we assume 3D object boxes available at training time)

“The green rubber cylinder is on the right of the blue bowl”
“The green rubber cylinder is on the right of the blue bowl”

4. Our agent maps spatial expressions to relative 3D offsets
5. Our agent populates a 3D canvas with the predicted object tensors and their relative offsets.

“The green rubber cylinder is on the right of the blue bowl”
“The green rubber cylinder is on the right of the blue bowl”

5. Our agent populates a 3D canvas with the predicted object tensors and their relative offsets.
5. Our agent populates a 3D canvas with the predicted object tensors and their relative offsets.

“The green rubber cylinder is on the right of the blue bowl”
6. The generated canvas when projected should match the RGB image views

"The green rubber cylinder is on the right of the blue bowl"
Scene imagination

“Red Rubber Cylinder to the left front of Blue Rubber Cube to the left front of Green Rubber Cylinder to right front of Blue Rubber Cube”

“Red Rubber Cube to the left front of the Blue Rubber Sphere to the right front of Cyan Metal Cylinder”

- **Neural rendering**: project the 3D feature maps using our learned project+RGB decoder neural module
- **Blender rendering**: use the object-centric 3D feature maps to retrieve nearest 3D mesh neighbors from a training set, then arrange the retrieved meshes based on predicted 3D spatial offsets
Scene imagination

“Purple Cylinder to the left behind of Brown Cube to the left front of Purple Sphere”

“Purple Cylinder to the left behind of Cyan Cube to the left front of Cyan Cube”

- **Neural rendering**: project the 3D feature maps using our learned project+RGB decoder neural module
- **Blender rendering**: use the object-centric 3D feature maps to retrieve nearest 3D mesh neighbors from a training set, then arrange the retrieved meshes based on predicted 3D spatial offsets
Scene imagination

“cyan sphere to the left of red cube”

“red cylinder to the front of red sphere to the left-front of blue sphere”

“cyan cylinder to the left of red sphere to the front of green sphere”

“blue sphere to the left front of green cube”

“cyan cylinder to the front of yellow cube”

“cyan cylinder to the front of yellow sphere to the behind of green sphere to the front of blue sphere to the front of gray cylinder to the behind of red sphere”
Grounding arbitrarily long utterances

“yellow sphere to the left front of green sphere to the left behind of blue sphere to the left front of blue cylinder to the left behind of red cube to the left front of gray cube”

IOU > 0.1

Object Out of Camera View

IOU = 0

Top View

“gray sphere to the left front of blue sphere to the left front of red sphere to the left behind of cyan sphere to the left behind of green sphere”

IOU > 0.1

Object Out of Camera View

IOU = 0

Top View
``put the cube inside the bowl”
``put the cube on the right of the bowl”
``put the cube on the left of the bowl”
Ongoing work

- Use 3D feature learning for robot control, imitation, RL
- Train geometry-aware tactile feature representations
- Extend language grounding to more complex language domains, e.g., actions
- Dynamic scenes: self-supervised multi object 3D tracking,
- Implicit 3D feature representations instead of feature voxels
- Affordable (cheap) memory-based video processing
Conclusion

Embodiment is the problem and the solution to visual recognition and common sense learning
Conclusion

```
We must perceive in order to move, but we must also move in order to perceive"
```

James J. Gibson
“If we figure out the right way to do 3D perception, no one will use 2D again, the same way when color TV was invented no one used black and white”

Yaser Sheikh
Thank you!

- Learning spatial common sense with geometry-aware recurrent networks, Tung et al., CVPR 2019,
- Embodied View-Contrastive 3D Feature Learning, Harley et al., arxiv
- Embodied language grounding, Prabhudesai et al., to be arxived soon