Affordance Prediction with Vision via Task-Oriented Grasp Quality Metrics

Luca Cavalli
luca3.cavalli@mail.polimi.it
Computer Science and Engineering
Autonomous Robots

- **Sense**: acquire and model data about the environment
- **Plan**: select the course of action
- **Act**: perform each planned action
Affordances
Affordances

Emergent properties embodied in the relations between an animal and its environment directly connected with the possibility of action of the animal with the environment.

Task-Oriented Grasping
Affordance Learning

<table>
<thead>
<tr>
<th>Task</th>
<th>Hammer</th>
<th>Bottle</th>
<th>Mug</th>
</tr>
</thead>
<tbody>
<tr>
<td>T&lt;sub&gt;1&lt;/sub&gt;</td>
<td>hand-over</td>
<td>pouring</td>
<td>tool-use</td>
</tr>
<tr>
<td>T&lt;sub&gt;2&lt;/sub&gt;</td>
<td>good</td>
<td>bad</td>
<td>bad</td>
</tr>
<tr>
<td>T&lt;sub&gt;3&lt;/sub&gt;</td>
<td>unwrap the sphere</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Song et al. Learning task constraints for robot grasping using graphical models. IROS (2010)
Affordance Learning

Detry et al. Task-oriented grasping with semantic and geometric scene understanding. IROS (2017)
State of the Art Limitations
State of the Art Limitations

- Categorical expression of tasks
  - Task definition intrinsic into the dataset associated to its label
  - Not easily extensible number of different tasks
  - Not possible to fine tune the task definition
State of the Art Limitations

- Categorical expression of tasks
  - Task definition intrinsic into the dataset associated to its label
  - Not easily extensible number of different tasks
  - Not possible to fine tune the task definition

- Human labeling
  - Slow process, prevents scaling of dataset size
  - Biased towards human hand affordances
  - No guarantee on optimality
Proposed Approach

- **Affordance function** for $T$ with object $O$, grasp $G$, and use point $U$:

$$F_T : (O, G, U) \mapsto \mathbb{R}$$

The higher the more suited $(O, G, U)$ are for task $T$, e.g., for cutting.
Proposed Approach

• Affordance function for $T$ with object $O$, grasp $G$, and use point $U$:

$$F_T : (O, G, U) \rightarrow \mathbb{R}$$

The higher the more suited $(O, G, U)$ are for task $T$, e.g., for cutting.

• Approximated as a function of base grasp metrics:

$$\tilde{F}_T : \phi \in \mathbb{R}^n \rightarrow \mathbb{R}$$

Base grasp metrics inferred with vision.
Selected Metrics

State of the Art Metrics:
- Grasp Robustness

Geometrical Metrics:
- Rotational Inertia
- Momentum discharge efficiency
- Use local geometry

Optimization Metrics:
- Hand effort on impact
- Hand effort on hold
- Force transmitted to use
Selected Metrics

State of the Art Metrics:
- Grasp Robustness

Geometrical Metrics:
- Rotational Inertia
- Momentum discharge efficiency
- Use local geometry

Optimization Metrics:
- Hand effort on impact
- Hand effort on hold
- Force transmitted to use
Selected Affordance Functions

Beating

1: \textbf{function } \hat{F}_{\text{beat}}(\varepsilon, \delta, I, E_i, E_h) \\
2: \quad \textbf{if } (\varepsilon < \tau_{\varepsilon} \quad \| \quad \delta < \tau_{\delta} \quad \| \quad \sum_{i=1}^{6} E_h[i] = \infty) \textbf{ then} \\
3: \quad \quad \textbf{return } -\infty \\
4: \quad \textbf{else} \\
5: \quad \quad \textbf{return } \frac{1}{E_i} \\
6: \quad \textbf{end if} \\
7: \textbf{end function}

Cutting

1: \textbf{function } \hat{F}_{\text{cut}}(\varepsilon, U_T, U_g) \\
2: \quad \textbf{if } (\varepsilon < \tau_{\varepsilon} \quad || \quad U_g < \tau_{U_g}) \textbf{ then} \\
3: \quad \quad \textbf{return } -\infty \\
4: \quad \textbf{else} \\
5: \quad \quad \textbf{return } U_T \\
6: \quad \textbf{end if} \\
7: \textbf{end function}

Picking

1: \textbf{function } \hat{F}_{\text{pick}}(E_h) \\
2: \quad \textbf{return } -\sum_{i=1}^{6} E_h[i] \\
3: \textbf{end function}
Pregrasps

Decouple grasp from object through a fixed **grasping policy** GP:

$$GP(p_0, O) \rightarrow g(O)$$

Initial state, the pregrasp

(a) (b) (c)
Simulating physics: GraspIt!
Data Collection

- Load objects from the Princeton Shape Benchmark
- Extract a *random* pregrasp and use location
- Simulate the random grasp on the target object
- Evaluate the metrics on the simulated grasp
- Log the pregrasp, use location and metric results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>350</td>
<td>400</td>
<td>20</td>
<td>1.25</td>
<td>56.82</td>
</tr>
<tr>
<td>2</td>
<td>280</td>
<td>97</td>
<td>91.3</td>
<td>5.7</td>
<td>259</td>
</tr>
</tbody>
</table>
Dataset Best: Picking

First data collection round

Second data collection round
Dataset Best: Picking

First data collection round

Second data collection round
Dataset Best: Cutting

First data collection round

Second data collection round
Dataset Best: Beating

First data collection round

Second data collection round
Simulating vision
Chollet et al. Xception: Deep learning with depthwise separable convolutions CVPR (2017)
Learning Models: PointNet

Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation CVPR (2017)
Learning Models: Local PointNet

- PointNet captures local geometrical patterns with no explicit notion of locality.

- Bias PointNet towards capturing hierarchical local geometrical patterns like Convolutional Neural Networks do, while preserving point cloud processing.
Learning Models: Local PointNet

- PointNet captures local geometrical patterns with no explicit notion of locality
- Bias PointNet towards capturing hierarchical local geometrical patterns like Convolutional Neural Networks do, while preserving point cloud processing
- Use the neighborhood of pixels in the depth image as a feature for locality of the resulting points
- Use the PointNet to compute point embeddings
Learning Models: Local PointNet

- PointNet captures local geometrical patterns with no explicit notion of locality
- Bias PointNet towards capturing hierarchical local geometrical patterns like Convolutional Neural Networks do, while preserving point cloud processing
- Use the neighborhood of pixels in the depth image as a feature for locality of the resulting points
- Use the PointNet to compute point embeddings
- Preserve the pixel neighborhood of point embeddings from the original depth image
- Use Xception to process point embeddings to the final output
Learning Task Separation

Evaluation model built in two steps:

- First filter stable grasps only with a general classifier
- Then infer the specific metrics only from stable grasps with a specialized regressor

```
1: function $M^\Phi (g, u)$
2:   if $M^\Phi_C (g) < \tau_C$ then
3:     return $v_{fail}$
4:   else
5:     return $M^\Phi_R (g, u)$
6: end if
7: end function
```
Benchmarking: Classification

<table>
<thead>
<tr>
<th>Model code</th>
<th>Cross entropy</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPN</td>
<td>0.3842</td>
<td>0.856</td>
</tr>
<tr>
<td>PNFULL</td>
<td>0.4398</td>
<td>0.829</td>
</tr>
<tr>
<td>PNF0</td>
<td>0.4403</td>
<td>0.820</td>
</tr>
<tr>
<td>PNPO</td>
<td>0.4628</td>
<td>0.819</td>
</tr>
<tr>
<td>CNNE</td>
<td>0.4400</td>
<td>0.818</td>
</tr>
<tr>
<td>PNSLIM</td>
<td>0.4696</td>
<td>0.800</td>
</tr>
<tr>
<td>CNNL</td>
<td>0.5537</td>
<td>0.741</td>
</tr>
</tbody>
</table>
Benchmarking: Regression

<table>
<thead>
<tr>
<th>Model code</th>
<th>MSE</th>
<th>CA</th>
<th>E[GMS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNPO</td>
<td>0.033</td>
<td>0.73</td>
<td>0.818</td>
</tr>
<tr>
<td>PNSLIM</td>
<td>0.034</td>
<td>0.72</td>
<td>0.824</td>
</tr>
<tr>
<td>PNFULL</td>
<td>0.034</td>
<td>0.71</td>
<td>0.771</td>
</tr>
<tr>
<td>PNFO</td>
<td>0.035</td>
<td>0.72</td>
<td>0.770</td>
</tr>
<tr>
<td>LPN</td>
<td>0.037</td>
<td>0.69</td>
<td>0.785</td>
</tr>
<tr>
<td>CNNE</td>
<td>0.038</td>
<td>0.69</td>
<td>0.781</td>
</tr>
<tr>
<td>CNNL</td>
<td>0.050</td>
<td>0.58</td>
<td>0.649</td>
</tr>
</tbody>
</table>
Picking grasps from Vision
Picking grasps from Vision
Presentations

This work produced:

- The substance of my MSc thesis work

- An early (submitted on 1st May) peer-reviewed accepted presentation at the Second International Workshop of Computational Models of Affordance for Robotics (IWCMAR) held in Montreal at ICRA 2019

- An accepted long abstract in the journal Frontiers in Neurorobotics about Computational Models of Affordance for Robotics, planning a submission by the 13th of October
Thank you!