Principles of Robot Autonomy II

Learning-based Approaches to Manipulation & Interactive Perception

Jeannette Bohg
Learning Outcome for next four Lectures

Modeling and Evaluating Grasps

Modeling and Executing Manipulation

Apply Learning to Grasping and Manipulation

Use Manipulation to Perceive better
Today’s itinerary

• Recap
  • Learning-based Grasping

• Learning-based Approaches to
  • Planar Pushing
  • Contact-Rich Manipulation Tasks

• Interactive Perception
  • Perception is not passive but active
  • Not seeing, but looking
For a Deeper Dive into Grasping and Manipulation

• CS326 – Topics in Advanced Robotic Manipulation – Fall 2024
Data-Driven Approaches to Grasping


Grasping previously unseen objects using only 2D images \textit{without} 3D meshes.
Data collection

Solution? Use synthetic data!

Realistic rendering using ray tracing.
Enables automatic labeling: random lighting, color, orientation, size...

2500 images
5 object classes
Image preprocessing

Edge filters (Y):

Texture filters (Y):

Average filter (Cb/Cr):

$6 \text{ (edge)} + 9 \text{ (texture)} + 1 \text{ (average)} \times 2 = 17 \text{ features per patch}$
Hardware setup

5 dof arm

Random object location on uncluttered table top
Using more sensing modalities and data to learn features and grasp policies

• DexNet 1.0 – 4.0 – Berkeley – AutoLab
• Google Arm Farm

"Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection" by Levine et al. IRR 2017.

https://berkeleyautomation.github.io/dex-net
Today’s itinerary

• Recap
  • Learning-based Grasping

• Learning-based Approaches to
  • Planar Pushing
  • Contact-Rich Manipulation Tasks

• Interactive Perception
  • Perception is not passive but active
  • Not seeing, but looking
Controlling through contact
Controlling through contact

Better Models
Better Feedback
Predictive Model

Sensory Observations → Model → Predicted Effect

Action

Model–Predictive Control
Modeling Planar Pushing

**Friction limit surface:** describes friction forces occurring when part slides over support.

When pushed with a wrench within the limit surface: **no motion.**

For **quasi-static pushing:** wrench on the limit surface; object twist normal to limit surface.

Relation between wrench cone, limit surface and unit twist sphere. Adopted from Chapter 37, Fig 37.10 in Springer Handbook of Robotics.
Validating Planar Pushing Models

IROS 2016. "More than a Million Ways to Be Pushed: A High-Fidelity Experimental Dataset of Planar Pushing" by Peter Yu and Maria Bauza.
Predicting physical effect
Keep Predicting
Bias-Variance Tradeoff

Error

Bias

Min Error

Variance

Model Complexity
Model-based
Bias-Variance Tradeoff

Error

Bias

Model-based

Min Error

Model Complexity

Variance
Data-Driven
Bias-Variance Tradeoff

Error

Learning

Structure

Bias

Variance

Model-based

Model Complexity

Data-Driven
Hypothesis

Physics Models + Learning = Generalization

Example:

Real motion
Simulated motion

BIG difference!

(Lynch 1992)
Hybrid Model

Sensory Observations

Learned Model

Parameters

Physics-based Model

End-to-End Loss on Effect

Predicted Effect

Extrapolation
Testing Hypothesis on a Case Study

Analytic Model

Relation between wrench cone, limit surface and unit twist sphere. Adopted from Chapter 37, Fig 37.10 in Springer Handbook of Robotics.
Compared Architectures

**Raw Sensory Observations**

**Neural Network only**

- Sensory Observations
- Action
- Learned Model
- Predicted Effect

**Hybrid Model**

- Sensory Observations
- Action
- Parameters
- Physics-based Model
- Learned Model
- Predicted Effect

**Training**: End-to-End

**Loss**: Error between Predicted and Ground Truth Effect
Advantages

Data Efficiency  Extrapolation
Testing Data Efficiency

trans [%]

thousand training examples

physics
Testing Data Efficiency

trans [%]

thousand training examples

hybrid
physics
Testing Data Efficiency

Training = Test Distribution
Testing Generalization

New Pushing Angles & Contact Points

Interpolation

New Push Velocities

Extrapolation

New Object Shapes

Generalization to new push velocities

Extrapolation

\[
\text{trans} \% \\
\text{rot} \%
\]

\[
\text{push velocity} \left[ \frac{\text{mm}}{\text{s}} \right]
\]

---

\text{physics}
Generalization to new push velocities

![Diagram showing training, testing, and extrapolation with graphs representing translation and rotation percentages as functions of push velocity. The graphs compare physics and hybrid models.]
Generalization to new push velocities

Extrapolation

Training ≠ Test Distribution
Don’t throw away structure
Learn to extract given state representation from raw data
A Concrete Suggestion
Interpretability
Today’s itinerary

• Recap
  • Learning-based Grasping

• Learning-based Approaches to
  • Planar Pushing
  • Contact-Rich Manipulation Tasks

• Interactive Perception
  • Perception is not passive but active
  • Not seeing, but looking
Models will never be perfect
Human multimodal sensor–motor coordination

Sensory Inputs

Example of Task–Relevant Information

pose  geo  force  friction
Human multimodal sensor–motor coordination

Sensory Inputs

Example of Task–Relevant Information

pose  geo  force  friction

Action

π
Human multimodal sensor–motor coordination

Sensory Inputs

Example of Task–Relevant Information

pose  geo  force  friction

Action

π
Generalizable representations for multimodal inputs
Generalizable representations for multimodal inputs

\[ \pi(f(o_1, o_2, o_3 \ldots o_n)) = a \]

multimodal sensory inputs

Learn representation

learn policy for new task instances

pose, geo, force, friction, ...
Multimodal sensory inputs

RGB
force/torque
robot states
Experimental setup

Peg geometry

Training

Testing
Learning generalizable representation

Inputs
- RGB image
- Force data
- Robot state

Encoder
Decoder
Representation
Learning generalizable representation

Inputs
- RGB image
- Force data
- Robot state

Fusion

Representation

Decoder
Learning generalizable representation

Inputs
- RGB image
- Force data
- Robot state

Fusion

Representation

Decoder
- pose
- geo
- friction

Task-relevant labels

Expensive
Collect labeled data with self-supervision
Dynamics prediction from self-supervision

Inputs
- RGB image
- Force data
- Robot state

Encoder
- robot action

Representation

Decoders
- Action-conditional optical flow

Outputs
Dynamics prediction from self-supervision

Inputs
- RGB image
- Force data
- Robot state

Encoder
- robot action

Representation

Decoders
- Action-conditional optical flow
- 0 / 1 contact in the next step?
Concurrency prediction from self-supervision

Inputs
- RGB image
- Force data
- Robot state

Encoder
- robot action

Representation
- 0 / 1 contact in the next step?
- 0 / 1 time-aligned?

Decoders
- Action-conditional optical flow

Outputs
Concurrency prediction from self-supervision

Inputs
- RGB image
- Force data
- Robot state

Robot action

Encoder

Representation

Decoders
- Action-conditional optical flow
  - 0 / 1 contact in the next step?
- 0 / 1 time-aligned?

Inputs:
- RGB image
- Force data
- Robot state
Concurrency prediction from self-supervision

Inputs
- RGB image
- Force data
- Robot state

Encoder
- robot action

Representation

Decoders
- Action-conditional optical flow
- 0 / 1 contact in the next step?
  yes, time aligned
Concurrency prediction from self-supervision

Unpaired Inputs
- RGB image
- Force data
- Robot state

Encoder
- robot action

Representation

Decoders
- Action-conditional optical flow
- 0 / 1 contact in the next step?
- 0 / 1 time-aligned?
Concurrency prediction from self-supervision

Unpaired Inputs
- RGB image
- Force data
- Robot state

Decoder
- robot action
- Representation
- Encoder

Decoders
- Action-conditional optical flow
- 0 / 1 contact in the next step?
- no, not time aligned!
Learning sample efficient policies

Inputs

- RGB image
- Force data
- Robot state

Encoder

Representation
How to efficiently learn a policy?

Inputs
- RGB image
- Force data
- Robot state

Encoder
Freeze 500k parameters

Representation

RL Policy
How to efficiently learn a policy?

Inputs
- RGB image
- Force data
- Robot state

Encoder
- Freeze 500k parameters

Representation

Learn 15k parameters

RL Policy
We evaluate our representation with policy learning.

Episode 0
0% success rate

Episode 100
21% success rate
We efficiently learn policies in 5 hours

Episode 300
73% success rate

Episode 300
71% success rate

Episode 300
92% success rate
Our multimodal policy is robust against sensor noise

1. Force Perturbation
2. Camera Occlusion
Force Perturbation

Normalized Force

| Force Perturbation | x1 |

<table>
<thead>
<tr>
<th>Time</th>
<th>Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.5</td>
</tr>
<tr>
<td>-0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>-1.0</td>
<td>-0.5</td>
</tr>
</tbody>
</table>
Camera Occlusion

Agent View

x1
How is each modality used?

Simulation Results
(Randomized box location)

**Force Only**: Can’t find box

**Image Only**: Struggles with peg alignment

**Force & Image**: Can learn full task completion
Does our representation generalize to new geometries?
Does our representation generalize to new geometries?

92% Success Rate

Tested on

Representation

Policy
Does our representation generalize to new policies?

92% Success Rate

62% Success Rate

Policy does not transfer

Tested on

Representation

Policy
Does our representation generalize?

92% Success Rate

Policy does not transfer

Tested on  
Representation ▲
Policy ▲

62% Success Rate

Representation transfers

Tested on  
Representation ▲
Policy ▲

92% Success Rate

Tested on  
Representation ▲
Policy ▲

Tested on  
Representation ▲
Policy ▲
Overview of method

Self-supervised data collection

$0_{\text{RGB}}, 0_{\text{force}}, 0_{\text{robot}}$

100k data points
90 minutes

Representation learning

$f(0_{\text{RGB}}, 0_{\text{force}}, 0_{\text{robot}})$

20 epochs on GPU
24 hours

Policy learning

$\pi(f(\cdot)) = a$

Deep RL
5 hours

Lessons Learned

1. Self-supervision gives us rich learning objectives

2. Representation that captures concurrency and dynamics can generalize across task instances

3. Our experiments show that multimodal representation leads to learning efficiency and policy robustness
State Representation - Physically Meaningful or Learned?

Explicit Representation

Sensory Observations → Learned Model → Parameters → Physics-based Model → Predicted Effect

Learned Representation

...
Today’s itinerary

- Recap
  - Learning-based Grasping

- Learning-based Approaches to
  - Planar Pushing
  - Contact-Rich Manipulation Tasks

- Interactive Perception
  - Perception is not passive but active
  - Not seeing, but looking
Interactive Perception

\[ S \times A \times t \]

Sensory Data \quad Actions \quad Time

Exploiting Multi-Modality

J. J. Gibson (1966) – The Senses considered as a Perceptual System.
Concurrency of Motion and Sensing

Held and Hein (1963). Movement-Produced Stimulation in the Development of Visually-Guided Behaviour
Accumulation over Time

Thanks to Octavia Camps at Northeastern University, Boston
Interactive Perception

Selfsupervised Learning of a Multimodal Representation

Concurrency prediction from self-supervision

Interactive Perception

Exploiting RGB, Depth and Motion for Instance Segmentation

Self-supervised Learning of a Multimodal Representation

Interactive Perception – Leveraging Action in Perception and Perception in Action.
Bohg, Hausman, Sanakaran, Brock, Kragic, Schaal and Sukhatme. TRO ’17.
Instance segmentation from a robotics viewpoint.
Optical Flow - Scene Flow

A Database and Evaluation Methodology for Optical Flow. Baker et al. IJCV. 2011

Pixel-wise Prediction

Computing Scene Flow

Scene Flow = Displacement of one 3D point $P_i^t$ over time

$S_i = P_i^{t-1} - P_i^t$

Motion-based Object Segmentation

Pixels that move rigidly belong to one object.

Trajectory feature \( \xi_k = [X_k, X_k + T_k] \)

Clustering

Data Set of Frame Pairs

Qualitative Results - Synthetic

Qualitative Results - Real

Interactive Perception

Exploiting RGB, Depth and Motion for Instance Segmentation

Interactive Perception

S × A × t

Sensory Data Actions Time

Homework 3 – Problem 3 Learning Intuitive Physics

Interactive Perception – Leveraging Action in Perception and Perception in Action.
Bohg, Hausman, Sanakaran, Brock, Kragic, Schaal and Sukhatme. TRO ’17.
Conclusions

• Interaction generates a rich sensory signal that eases perception.
• Action-conditional, multi-modal representation helps contact-rich manipulation and generalizes.
Suggested Reading

- "Data-Driven Grasp Synthesis – A survey" by Bohg et al. TRO 2014
- "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics" by Mahler et al.. RSS 2017. [https://berkeleyautomation.github.io/dex-net](https://berkeleyautomation.github.io/dex-net)
- "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection" by Levine et al. IJRR 2017.
Next time

Wednesday – Imitation Learning I – Dorsa Sadigh

Feb 21 - Guest Lecture by Quan Vuong
Software engineer at Google DeepMind Robotics.