Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

Nikolay Pashov

Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

Sergey Levine SLEVINE@GOOGLE.COM
Peter Pastor PETERPASTOR@GOOGLE.COM
Alex Krizhevsky AKRIZHEVSKY@GOOGLE.COM
Deirdre Quillen DEQUILLEN@GOOGLE.COM
Google

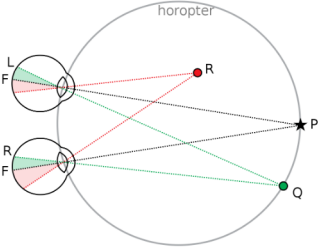
Abstract

We describe a learning-based approach to hand-eye coordination for robotic grasping from monocular images. To learn hand-eye coordination for grasping, we trained a large convolutional neural network to predict the probability that task-space motion of the gripper will result in successful grasps, using only monocular camera images and independently of camera calibration or the current robot pose. This requires

Image from paper


Learning hand-eye coordination for robotic grasping
Watch later
Share

Binocular vision



You cannot collect depth information with only one eye!

Comparison to similar works

- Monocular visual input
- No calibration
- No communication between robot and imaging module

Algorithm

Component 1: Convolutional neural network $g(I_t, \vec{v}_t)$
outputs the probability that,
given an input image I_t
a motion command $\vec{v}_t$ will lead to a successful grasp

Algorithm

Component 2: Servoing mechanism $f(I_t)$
uses the image input I_t
to choose an action for the robotic hand.
It calls $g(I_t, \vec{v}_t)$ internally!

Algorithm

Algorithm 1 Servoing mechanism $f(\mathbf{I}_t)$

Given current image $\mathbf{I}_t$ and network g.
Infer $\mathbf{v}_t^\star$ using g and CEM.
Evaluate $p = g(\mathbf{I}_t, \emptyset)/g(\mathbf{I}_t, \mathbf{v}_t^\star)$.
if $p > 0.9$ **then**
    Output $\emptyset$, close gripper.
else if $p \leq 0.5$ **then**
    Modify $\mathbf{v}_t^\star$ to raise gripper height and execute $\mathbf{v}_t^\star$.
else
    Execute $\mathbf{v}_t^\star$.
end if

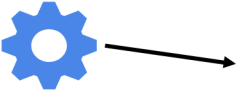
CEM = vector sampling mechanism

$g(I_t, \emptyset)$ is the probability of a successful grasp with no motion

Image from paper

Algorithm

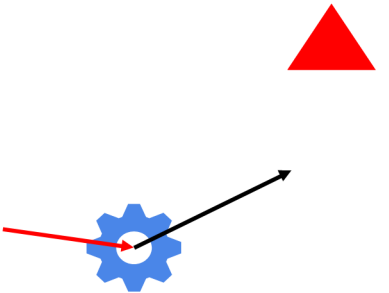
Propose a motion command



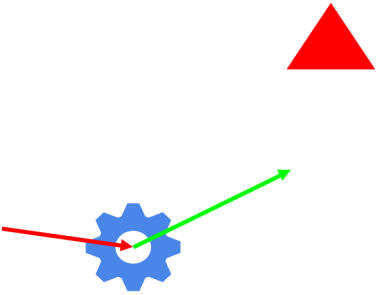
$$p \leq 0.5$$

Lift gripper up

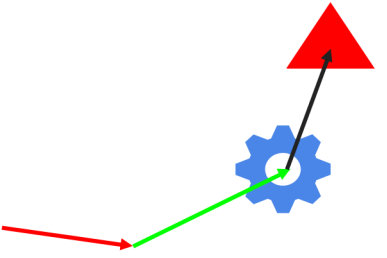
Propose new motion command



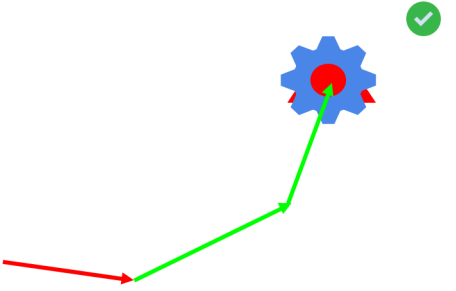
$$0.5 \leq p \leq 0.9$$



Propose new motion command



$$p \geq 0.9$$



Neural network

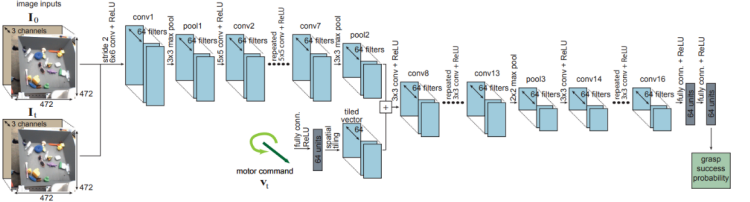


Image from paper

A traditional layer sequence

- Conv + Pooling + ReLU
- Following a well established pattern

Input

- Image with gripper 472x472
- Image without gripper 472x472
- 5D Vector encoding translation and rotation

Input

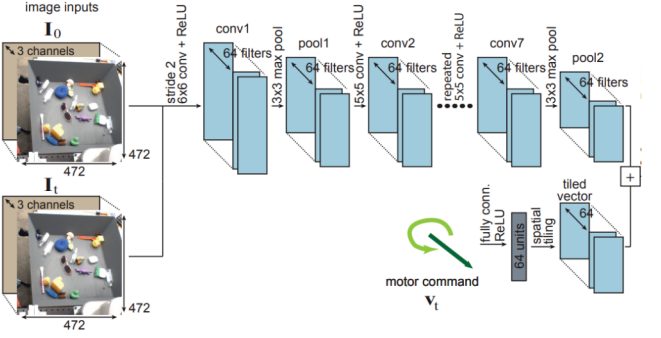


Image from paper

Training

Generating training data

1. The robot executes a motion command T times
   - $2 \leq T \leq 10$
   - T-th command is always a grasp attempt
2. After grasp attempt, the grasp success is evaluated
3. A data point is generated: $(I_t, p_T - p_t, l)$
   - p_x is the gripper position at step x

Training

Generating training data

- First 50% of samples are generated with random motions
- The rest are generated using the last trained model with previous data
- Model is re-trained 4 times during data generation

Observing the algorithm as RL

- The CNN $g(I_t, \vec{v}_t)$ is approximating the Q-function defined by the policy of the servoing mechanism (when T=2)
  - Note: Q-function = action-value function

Evaluating grasp success

1. Measure distance between fingers. If more than 1cm, a successful grip is considered
2. Make an image without the gripper and subtract from the initial image. After a successful grip, the difference should be 0

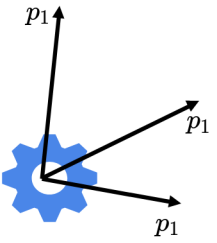
$$\text{Data point: } (I_t, p_T - p_t, l)$$

$$l = 1 \text{ given a successful grip}$$

$$l = 0 \text{ given an unsuccessful grip}$$

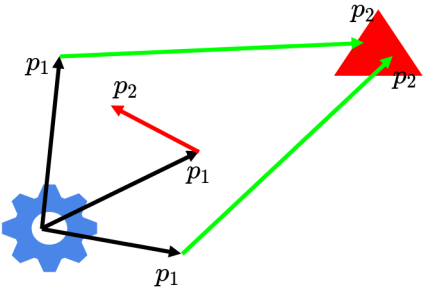
Visualize data generation

T = 3



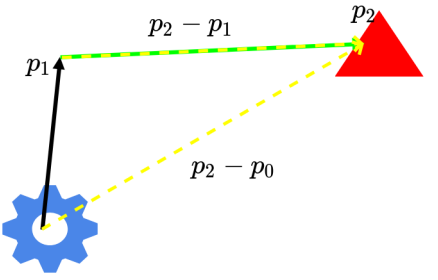
Visualize data generation

T = 3



Positive data point

T = 3

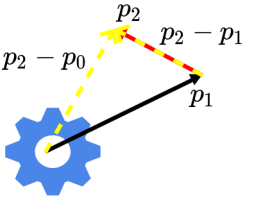


$l = 1$

Note that p3 is the grasp itself!

Negative data point

T = 3



$l = 0$

Note that p3 is the grasp itself!

Results

1. Presented algorithm vs random gripper motion

Results

2. Presented algorithm vs hand-designed system

a.k.a. using sensors to gather depth information objects are localized and the calibrated gripper is sent to that location

Results

2. Presented algorithm vs open loop

a.k.a. using the CNN to predict successful grasp probability and then directly attempt grasping

a.k.a the same as the presented algorithm, without the continuous servoing

Results

without replacement	first 10 ($N = 40$)	first 20 ($N = 80$)	first 30 ($N = 120$)
random	67.5%	70.0%	72.5%
hand-designed	32.5%	35.0%	50.8%
open loop	27.5%	38.7%	33.7%
our method	**10.0%**	**17.5%**	**17.5%**
with replacement	failure rate ($N = 100$)		
random	69%		
hand-designed	35%		
open loop	43%		
our method	**20%**		

Table 1. Failure rates of each method for each evaluation condition. When evaluating without replacement, we report the failure rate on the first 10, 20, and 30 grasp attempts, averaged over 4 repetitions of the experiment.

Image from paper

Results

without replacement	first 10 ($N = 40$)	first 20 ($N = 80$)	first 30 ($N = 120$)
random	67.5%	70.0%	72.5%
hand-designed	32.5%	35.0%	50.8%
open loop	27.5%	38.7%	33.7%
our method	**10.0%**	**17.5%**	**17.5%**
with replacement	failure rate ($N = 100$)		
random	69%		
hand-designed	35%		
open loop	43%		
our method	**20%**		

Table 1. Failure rates of each method for each evaluation condition. When evaluating without replacement, we report the failure rate on the first 10, 20, and 30 grasp attempts, averaged over 4 repetitions of the experiment.

Image from paper

Evaluating results improvement with the increase of trainging data

without replacement	first 10 $N=40$	first 20 $N=80$	first 30 $N=120$
12%: $M = 182{,}249$	52.5%	45.0%	47.5%
25%: $M = 407{,}729$	30.0%	32.5%	36.7%
50%: $M = 900{,}162$	25.0%	22.5%	25.0%
100%: $M = 2{,}898{,}410$	**10.0%**	**17.5%**	**17.5%**

Table 2. Failure rates of our method for varying dataset sizes, where M specifies the number of images in the training set, and the datasets correspond roughly to the first eighth, quarter, and half of the full dataset used by our method. Note that performance continues to improve as the amount of data increases.

Image from paper

Non-trivial strategies for grasping

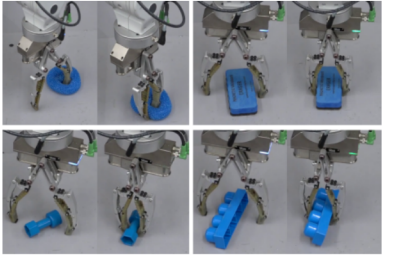


Figure 9. Grasps chosen for objects with similar appearance but different material properties. Note that the soft sponge was grasped with a very different strategy from the hard objects.

Image from paper

Non-trivial strategies for grasping

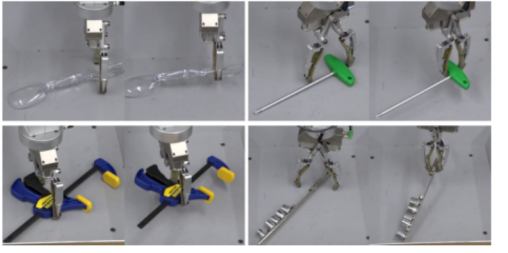


Figure 10. Examples of difficult objects grasped by our algorithm, including objects that are translucent, awkardly shaped, and heavy.

Image from paper

Future work

- Diversifying the training set for increased generalization
- Integration of reinforcement learning to allow the system more "creativity"
- Adapting the method to a real-life scenario with varying environments and tasks for the robots

Sampling motion vectors

- Perform 3 times:
  - sample 64 vectors
  - pick the 6 with highest grasp probability (top 10%)
  - fit a Gaussian distribution to the picked ones
- The first 64 samples are from a zero-mean Gaussian centered on the gripper position

Thanks for the attention