Learning to Place Objects onto Flat Surfaces in Upright Orientations

Rhys Newbury, Kerry He, Akansel Cosgun and Tom Drummond

Introduction

Motivation

- Placement of objects correctly in specific orientations is an important skill for robots
- For example, a robot unpacking the dishwasher should place plates, glasses and bowls on shelves in certain orientations
- However, this is a mostly overlooked field in robotic manipulation research

Introduction

Existing Literature

- Fu [1] explores human preferred object orientations at a theoretical level but doesn't consider this in the context of robotic manipulation.
- Harada [2] uses an analytical approach based on matching planar surfaces of the object and placement surface, but requires access to the 3D object model
- Jiang [3] uses a learning-based approach on hand-chosen features, where possible placements are sampled and scored

Contributions

- Combine concept of human-preferred upright orientations as first explored by Fu [1] in the context of the robotic placement problem
- Propose a novel hardware-in-the-loop iterative approach which continuously applies rotations to the object until it converges to the upright object orientation
- Proof-of-concept implementation on a robotic system to demonstrate sim-to-real transfer feasibility

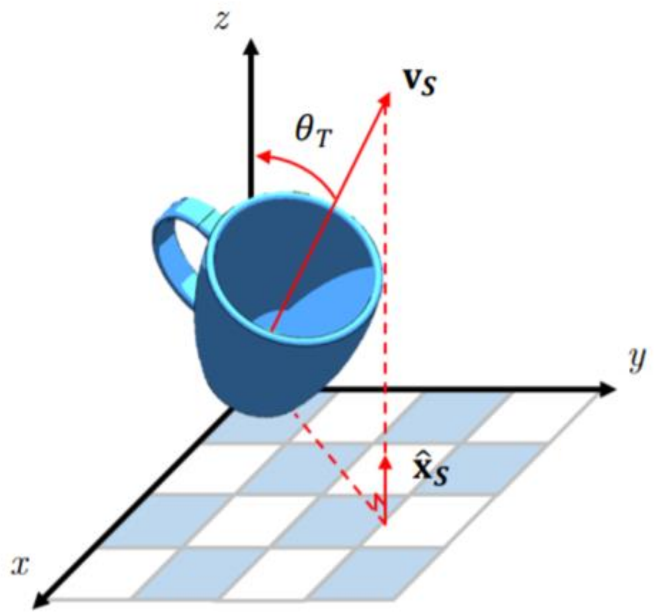
[1] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," in ACM SIGGRAPH, 2008.

[2] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, H. Onda, T. Yoshimi, and Y. Kawai, "Object placement planner for robotic pick and place tasks," in IEEE/RSJ IROS, 2012.

[3] Y. Jiang, C. Zheng, M. Lim, and A. Saxena, "Learning to place new objects," IEEE ICRA, 2011.

Problem Description

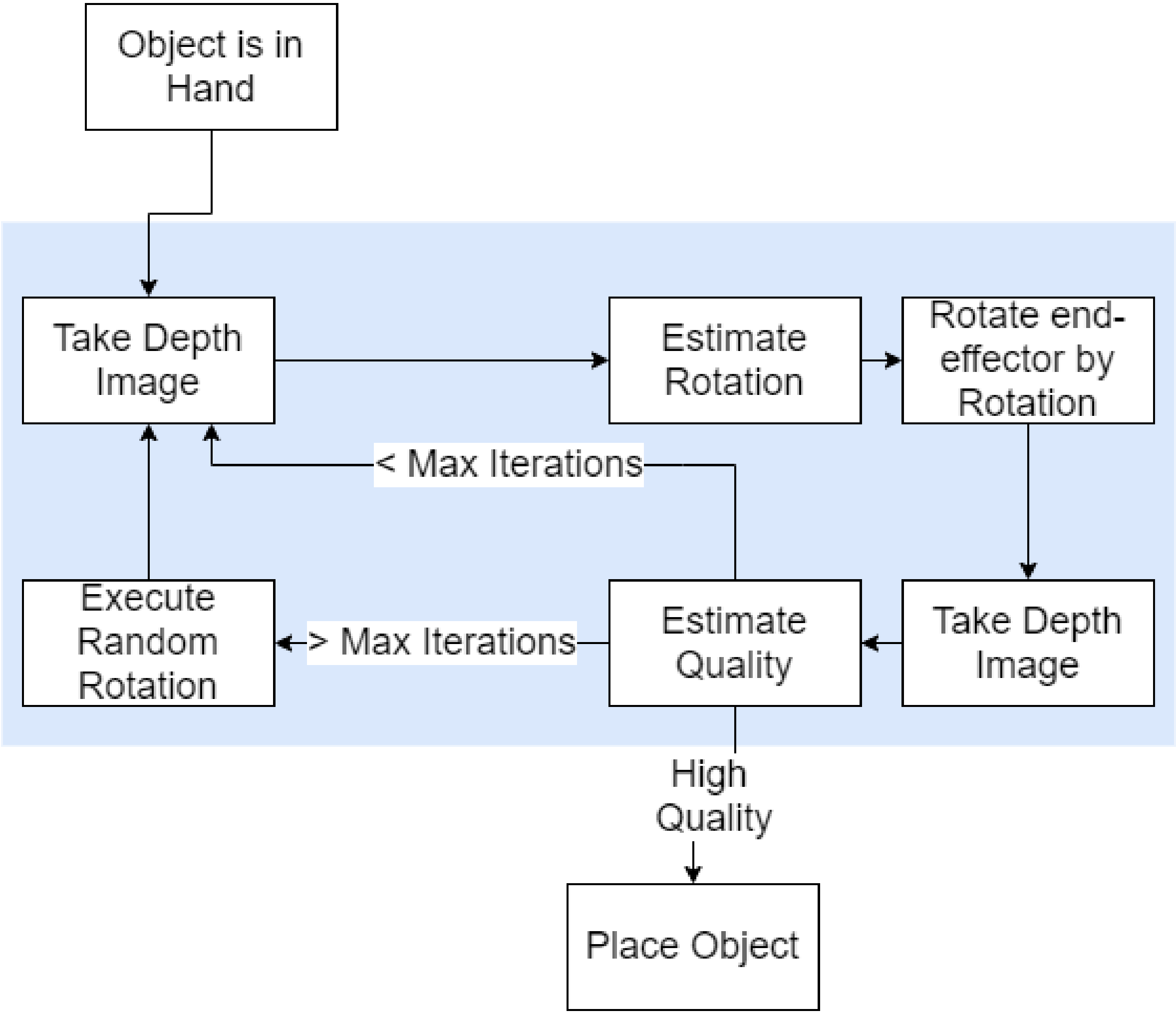
Definitions:
- **Upright orientation**: Human-associated convenient orientations of objects
- **Successful placement**: Object is stable and in the upright orientation under gravitational and contact forces after release



Assumptions:
- No prior knowledge about object class, 3D model or upright orientation
- Has access to depth cameras and force sensing
- Placement surface is flat, infinite and uncluttered
- Each object has one upright orientation

Task:
- Robot starts with object already in hand
- Goal is to place object down successfully by rotating the object to the upright orientation

Methodology: Iterative Algorithm



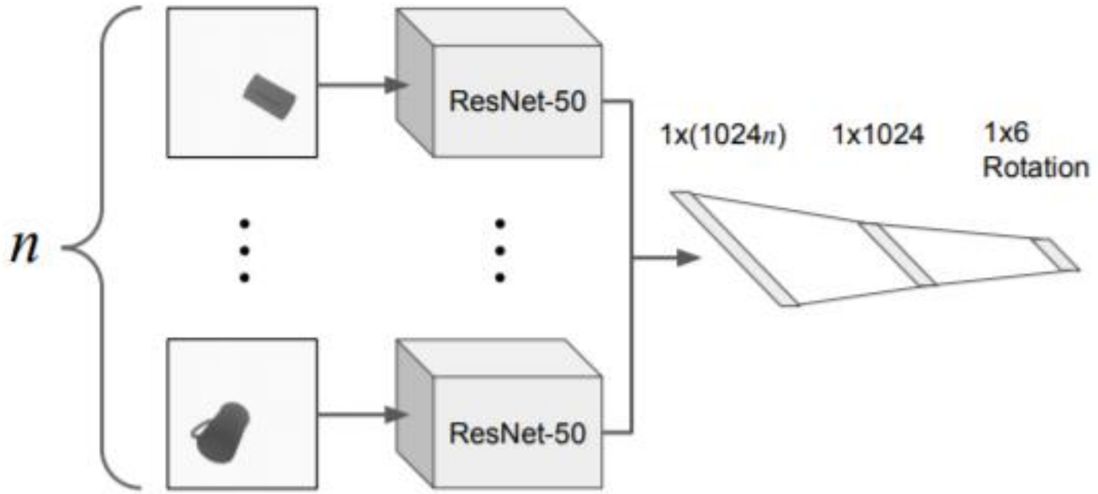
Methodology: Networks

Placement *Rotation* Convolutional Neural Network (PR-CNN)

- Estimates the rotation that transforms the object to the upright orientation

Placement *Quality* Convolutional Neural Network (PQ-CNN)

- Estimates the confidence level that the object would be stable in its upright orientation if it is placed in its current orientation.



Experiments

Simulated

- 50 daily object models: 45 training objects, 5 test objects
- 5-fold cross validation
- 250 placement trials for each cross validation set
- 3 RGB-D cameras from different viewpoints
- Two sets of experiments
  - Without robot – object is rotated in free space
  - With robot – kinematics are enforced

Real world

- Only green bottle was in the training set, all other objects are novel
- 10 trials per object
- Single RGB-D camera
- Simple force-feedback used to place objects
- Direct sim-to-real transfer

Experiments

Methods

- Baseline: Place on largest flat plane
- Single pass (SP): One use of PR-CNN
- Iterative (ITR): Iterative use of PR-CNN
- Iterative with Quality (ITR-Q): Our full approach combining PR-CNN and PQ-CNN.

Metrics

- Success Rate: Object is in upright orientation
- Stability Rate: Object does not fall over.
- Angular Error: The average angle difference.

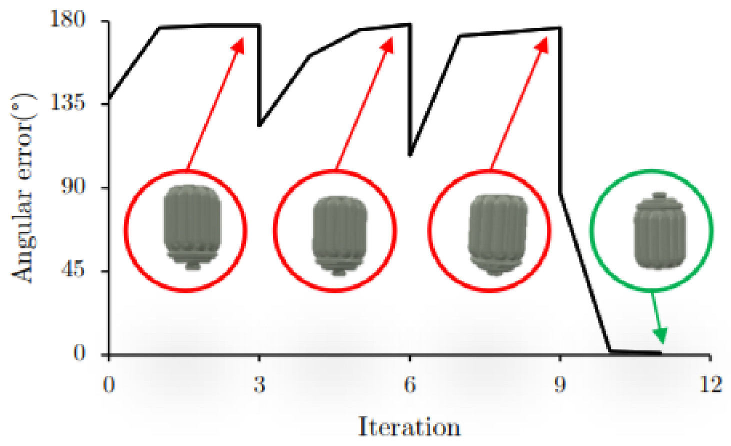
Experiments: Simulated + No Robot

Results

	Success Rate (%)	Stability Rate (%)	Angular Error (°)
Baseline	54.0 ± 10.0	96.7 ± 4.2	47.7 ± 11.5
SP	84.4 ± 8.9	89.9 ± 5.8	22.8 ± 13.1
ITR	96.1 ± 4.5	98.3 ± 1.6	8.0 ± 7.1
ITR-Q	**98.1** ± 1.9	**99.3** ± 1.1	**5.2** ± 2.5

Failure Modes

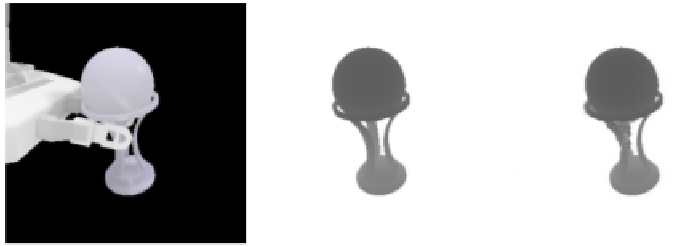
- PR-CNN could converge to an incorrect orientation.



Experiments: Simulated + Robot

Object Segmentation

- Train a GAN to remove the gripper from the image.
- The GAN had to 'hallucinate' occluded by the gripper



Results

- Decrease in performance, especially with small objects where the GAN may struggle to segment the object.

	Success Rate (%)	Stability Rate (%)	Angular Error (°)
ITR-Q w/ Gripper	90.3 ± 2.6	95.3 ± 1.9	16.7 ± 7.2

Experiments: Real World

Modifications

- Depth images were preprocessed
- Moving-window of depth images to calculate average rotation to remove outliers
- Positions and rotations about z-axis were sampled to find kinematically feasible paths and goal poses for the robot

Results

- Achieved 88% success rate in the real world

	Success Rate (%)	Avg. Num. Iterations
Bowl	100.0	1.8
Sunscreen	90.0	1.8
Spray Bottle	80.0	1.6
Green Pitcher	80.0	1.7
Green Bottle	90.0	1.3

Conclusion

Limitations and future work
- More detailed analysis to be completed into sim-to-real transfer.
- Closed-loop reactive approach could be more useful.
- Extend representation to include multiple ground truth rotations.

Conclusion
- Placement is just as important as grasping
- Once the object is grasped, the robot must do something with the object, and the simplest thing is to place it back down
- Object orientations have meaning, and for placements to be useful these need to be understood by the robot

Appendix A – Loss Function [4]

Angular representation

6D continuous rotational representation. SO(3) to 6D on left, 6D to SO(3) on right

$$g_{GS}\left(\begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix}\right) = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}$$

$$f_{GS}\left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix}$$

$$b_i = \begin{bmatrix} \begin{cases} N(a_1) & \text{if } i = 1 \\ N(a_2 - (b_1 \cdot a_2)b_1) & \text{if } i = 2 \\ b_1 \times b_2 & \text{if } i = 3 \end{cases} \end{bmatrix}^T$$

Geodesic Error

Ground truth rotation: R_s

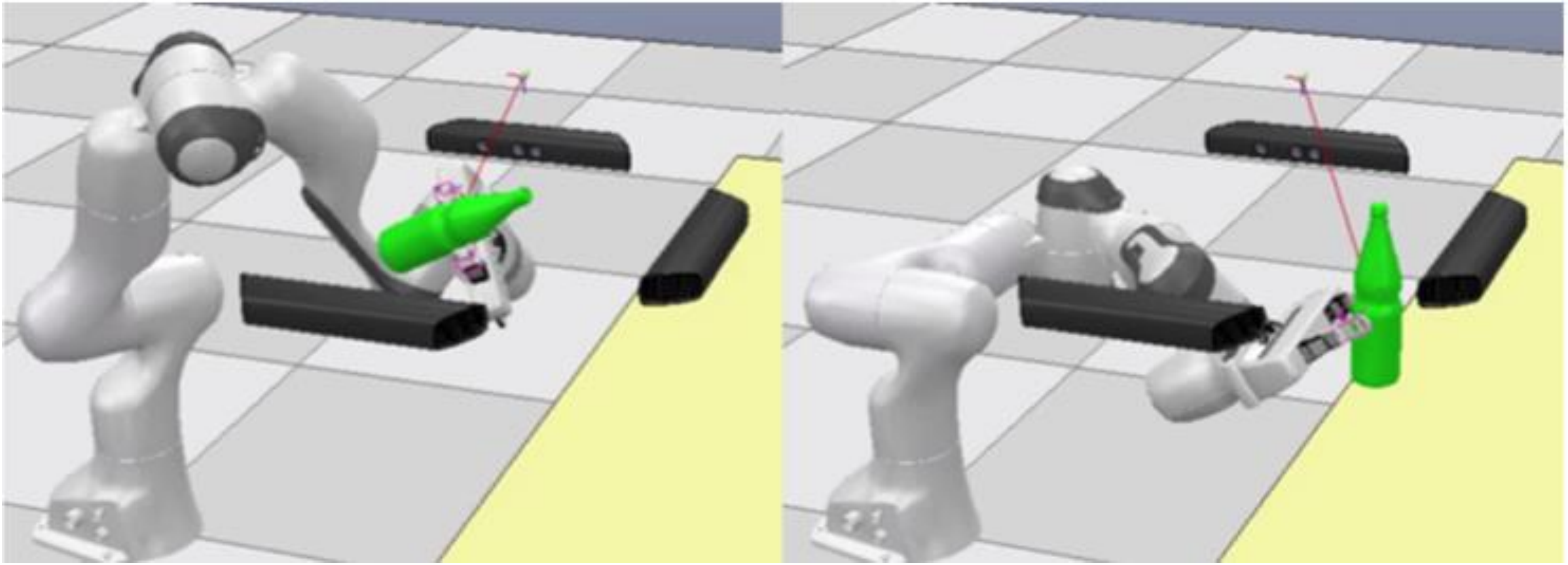
Output rotation: R_s

$$\mathcal{L}_{geodesic} = \arccos\left(\frac{tr(R_s R_T^{-1}) - 1}{2}\right)$$

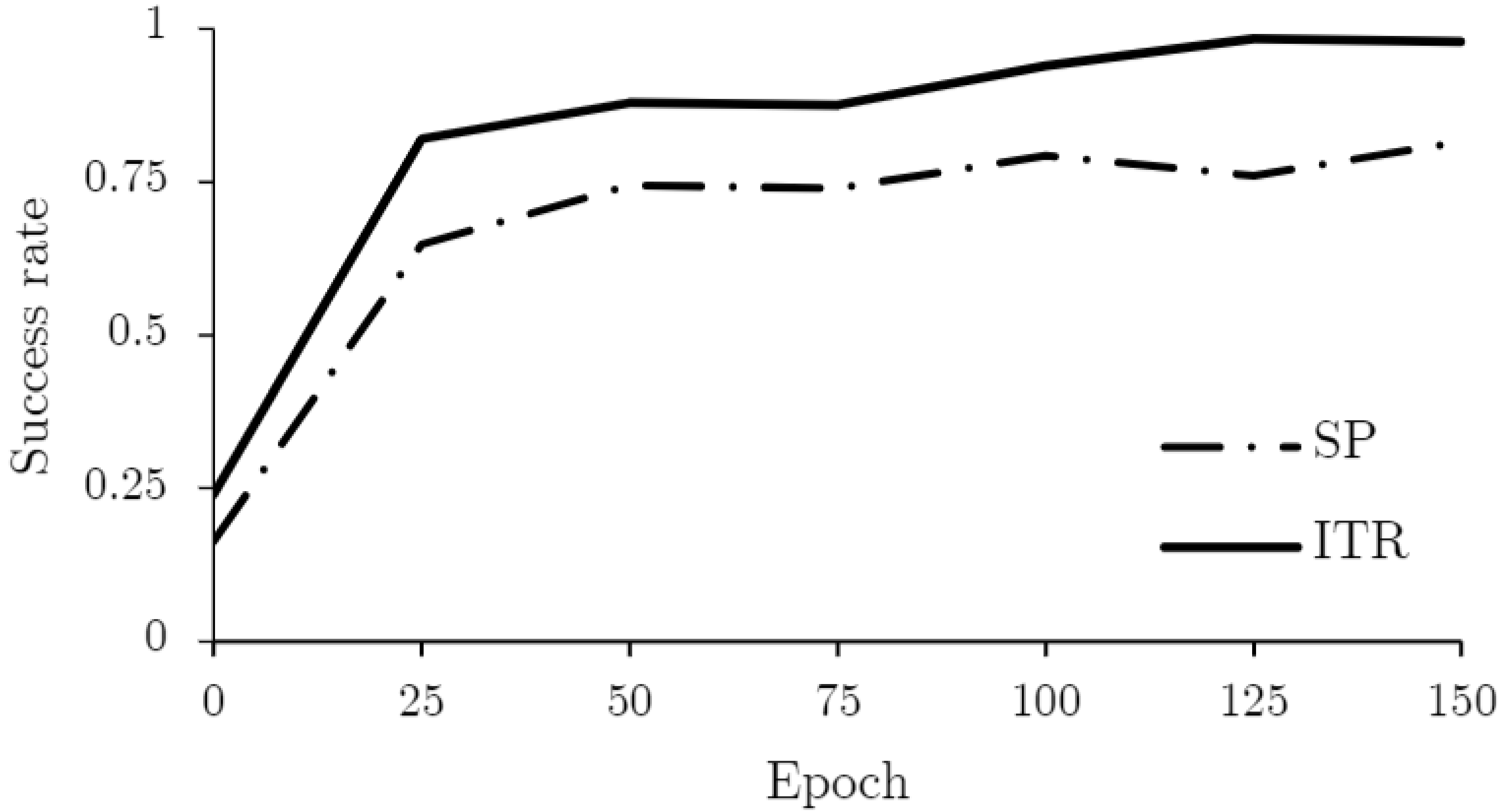
$$tr(R) = R_{00} + R_{11} + R_{22}$$

[4] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, “On the continuity of rotation representations in neural networks.” in CVPR, 2019.

Appendix B – Simulation Environment



Appendix C – ITR v SP Training



Appendix D – Design Parameters

Network architecture

	ITR (Success Rate %)
ResNet-50 SW	54.8
ResNet-50 PT	89.6
ResNet-50 PT SW	**98.4**

Number of cameras

	ITR (Success rate %)
1 Camera	85.2
2 Cameras	96.8
3 Cameras	**98.4**
4 Cameras	96.4

Angular representation

	ITR (Success rate %)
Euler	32.4
Quaternion	96.4
6D [25]	**98.4**