Object Detection and Pose Estimation for Robotic Manipulation using Physics Simulation and Monte-Carlo Tree Search

Chaitanya Mitash, Abdeslam Boularias and Kostas E. Bekris

Department of Computer Science, Rutgers, the State University of New Jersey

Motivation for autonomous data generation

Motivation:

- State-of-the-art methods use Convolutional Neural Network (CNN) to perform object segmentation.
- CNNs need access to a large set of labeled data, which requires intensive human labor.
- Current techniques for generating synthetic dataset suffers from dataset bias as they lack realism.

Figure: Rutgers RGBD dataset with manual annotation

Objective:

- Generate a physically-realistic labeled dataset in an autonomous manner to train a CNN for object detection.

Figure: Physically unrealistic synthetic dataset generation

Using Physics Simulation to generate training dataset

- Environmental and geometric constraints are used to generate datasets for setups such as shelf bin and table-top.
- Each scene is generated by randomly sampling object poses from a domain specified for the setup.
- Physics simulation is performed for generating physically realistic scenes so that the training dataset captures appropriate object scales and occlusions.

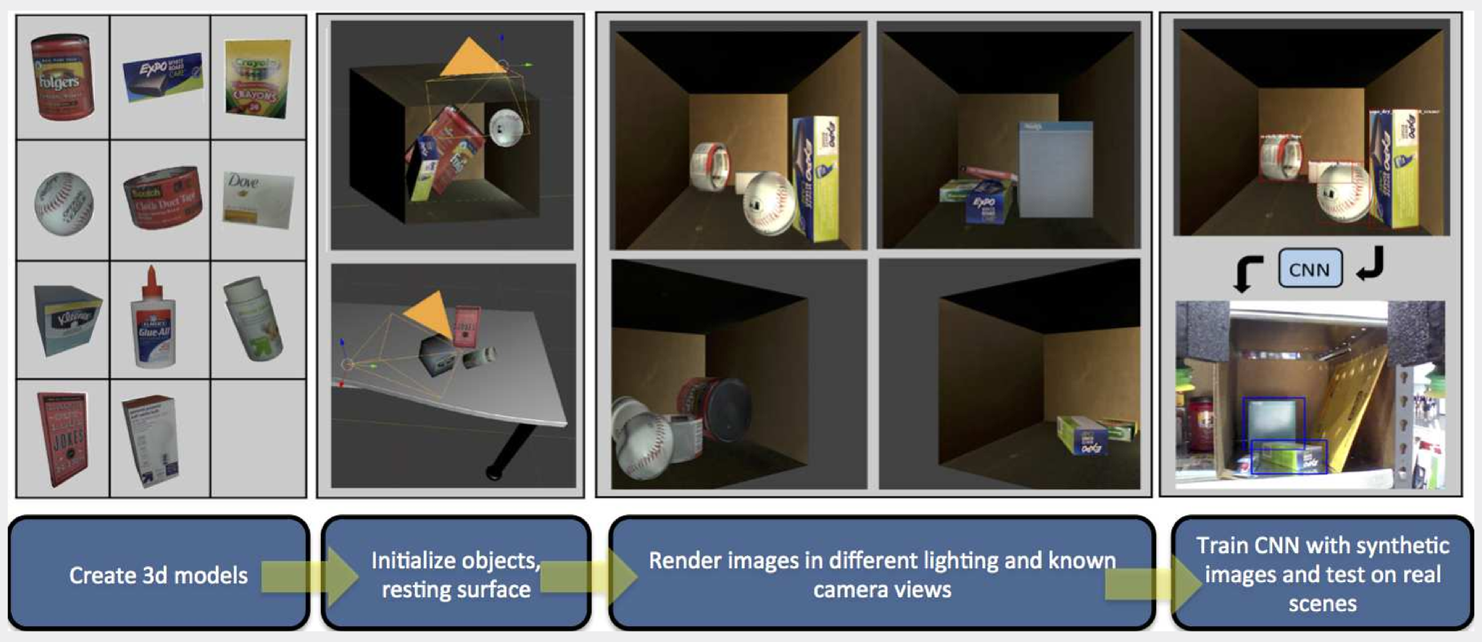


Figure: Pipeline for data generation using physics simulation

Dataset Generation Tool

- The software tool for dataset generation is publicly released and can be found at `https://github.com/cmitash/physim-dataset-generator`
- It uses the `Blender` python API for simulation and rendering.
- The repository also includes CAD models for 16 objects from *Amazon Picking Challenge 2016*.
- The camera parameters, choice of environment, and lighting options are provided as tunable parameters.
- The tool could be used to generate scenes with either bounding-box or pixel-wise class labels for objects.

Figure: example image

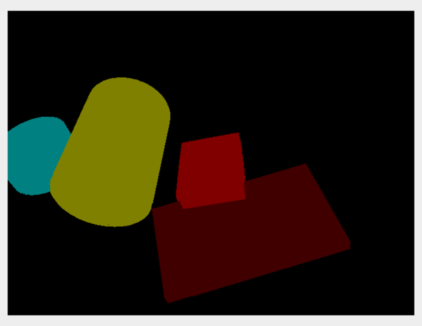


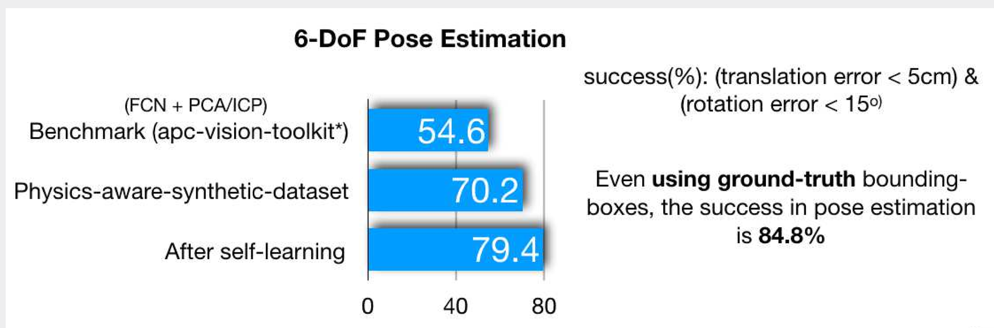
Figure: pixel-wise labeling

Figure: bounding-box labeling

Evaluating Object Detection

- Evaluation is performed on `Shelf & Tote` benchmark dataset.
- `Faster-RCNN` is trained using approximately 2000 physically-realistic scenes. The small number helps avoid overfitting with respect to texture and scene illumination.

	Method	Success(IoU>0.5
	Team MIT-Princeton [5] (Benchmark)	75%
Simulation	Sampled from test data distribution	69%
	Sampled from uniform distribution	31%
	Physics-aware simulation	64%
	Physics-aware simulation + varying light	**70%**



Motivation for MCTS

- Model-matching to individual object segments often result in pose estimates of objects that are physically inconsistent with other objects.
- Efficient search technique is required to search over the combination of individual pose hypotheses which best explains the observed scene.

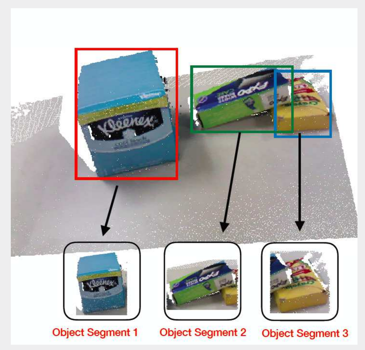


Figure: Object segmentation with Faster-RCNN

Figure: Result of model-matching using Super4PCS

Pose Estimation using Monte Carlo Tree Search

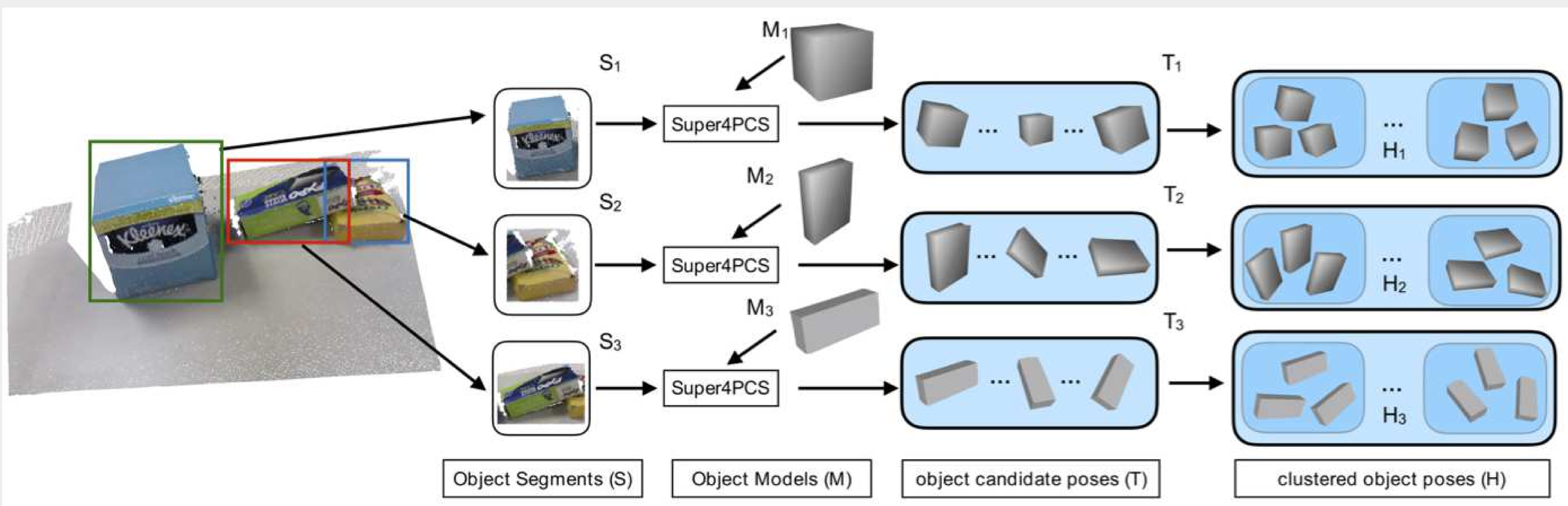


Figure: object candidate pose generation and clustering to reduce set cardiniality

- Pose candidate set is constructed for each object using the extracted object segment and the 3D CAD model.
- For computational efficiency, the set of object hypotheses is clustered to obtain smaller candidate sets while still containing poses close to the true solutions.

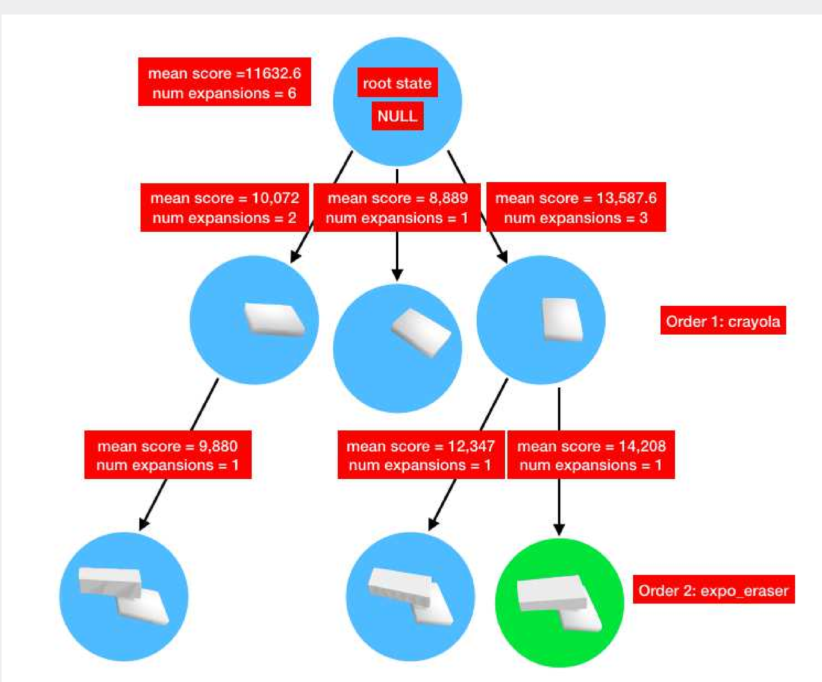


Figure: Monte Carlo Tree Search for pose estimation

- An order of object placement is computed based on a set of rules defined over the object segments.
- Node expansion in the tree corresponds to an object placement, which is constrained (imposed by using physics simulator and point cloud trimming) by the previously placed objects.
- The leaf nodes (complete assignment) are rendered and a score is computed by comparing rendered scene to the observed depth image.
- The search uses Upper Confidence Bound to trade-off exploration and exploitation within the search.

Pose Estimation Evaluation

- Results demonstrate that search is useful in case of a clutter.
- MCTS converges much faster compared to using a heuristic based on registration score.

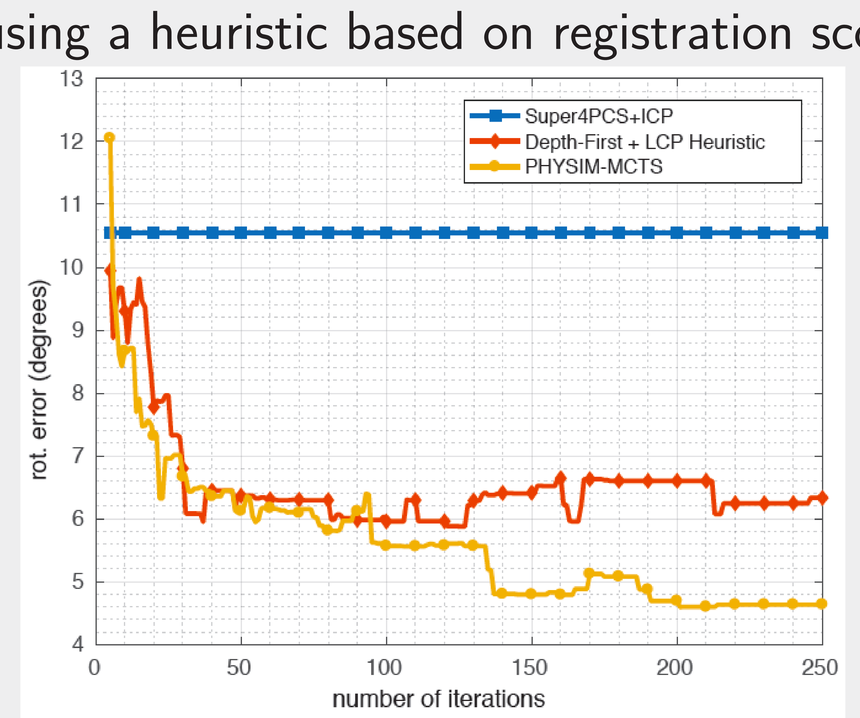


Figure: Rotation error (degrees) vs the number of search expansions

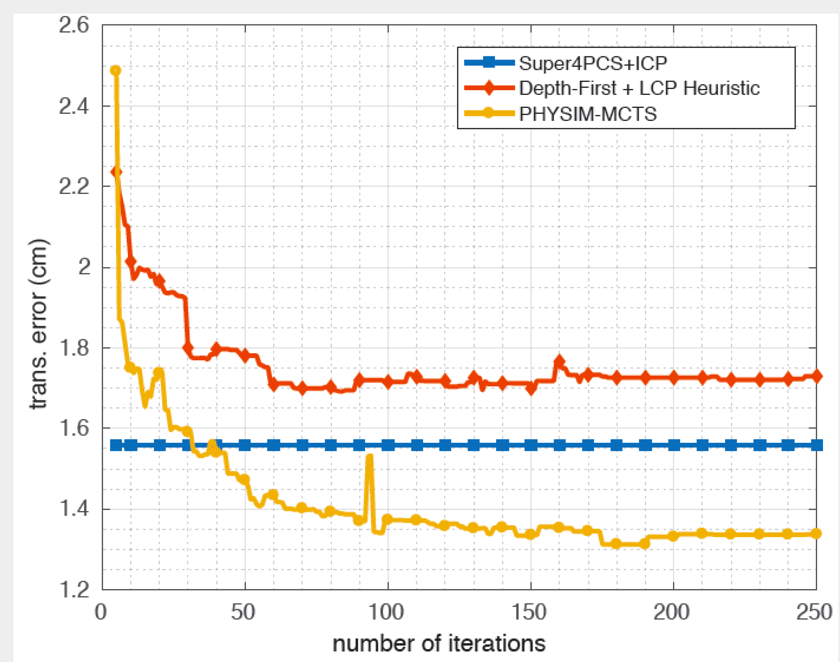


Figure: Translational error (cms) vs the number of search expansions

http://www.physimpose.com

{cm1074, kb572, ab1544}@cs.rutgers.edu