Task-oriented Visual Object Pose Estimation for Robot Manipulation: A Modular Approach

Ahmed Abdelrahman[1], Peter So[1], Hoan Quang Le[1], Abdalla Swikir[2], and Sami Haddadin[2]

[1] Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), Munich, Germany
[2] Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates

March 25, 2025

Background: The euROBIN Project

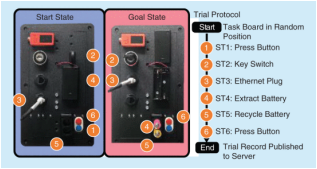
AI NoE euROBIN
THE EUROPEAN EXCELLENCE NETWORK ON AI-POWERED ROBOTICS

- Aim: effective transfer of robot skills across platforms, applications, research groups, etc.
- Objectives:
  - developing transferable robot skills
  - demonstration and evaluation on tasks through collaborative competitions
  - sharing software and knowledge on a central platform
- euroCore: ”a centralized repository for sharing software modules, data, and expertise”

Source: https://www.eurobin-project.eu/

Background: The Electronic Task Board [1] Benchmark

- Automated benchmarking of manipulation task performance in robot competitions
- Examples of onboard tasks:
  - pushing buttons
  - plugging cables
  - using a multimeter probe
  - replacing batteries
- Tracks task progress and collects metrics in a cloud database
- To solve the task board and similar problems ⇒ **a transferable, vision-based object recognition and pose estimation method**

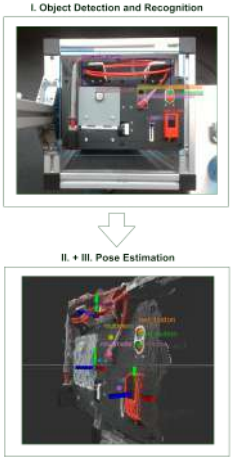


Example of a task board trial protocol [1]

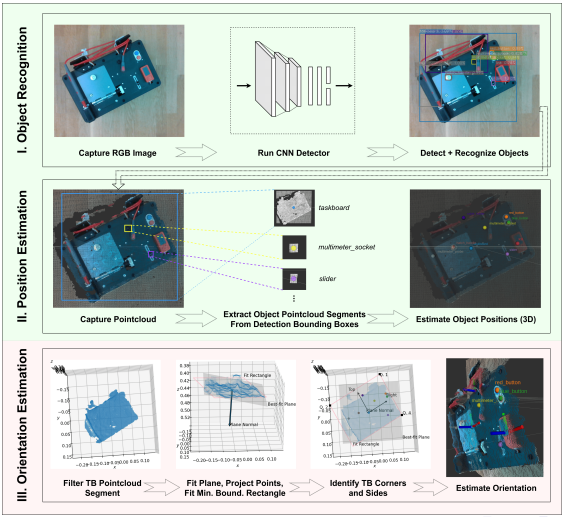
[1] So, Peter, et al. "Digital robot judge: Building a task-centric performance database of real-world manipulation with electronic task boards." IEEE Robotics & Automation Magazine 31.4 (2024): 32-44.

Solution: RGB-D-based Pose Estimation Approach

- An open-source, reusable, modular object pose estimator
- Requires an RGB-D sensor, such as the Intel Realsense D435
- Validated on the electronic task board: from color and depth images, locate the board and its components (position and orientation)
- Involves three stages:
  I. Object Recognition
  II. Position Estimation
  III. Orientation Estimation

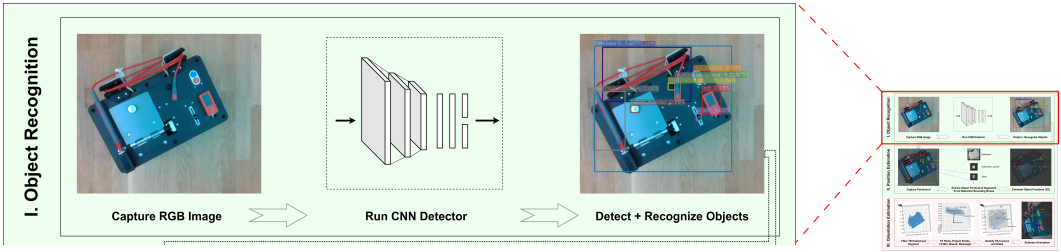


Overview of Workflow:



Stage I: Object Recognition

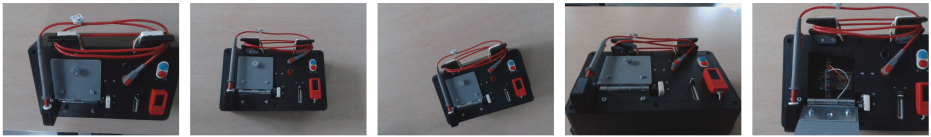
- Conventional CNN [2] for detecting known objects from RGB images
- A pre-trained Faster R-CNN model, fine-tuned using *transfer learning*
- Training data: a small set of images in three lighting conditions with varying board configurations and cluttered environments



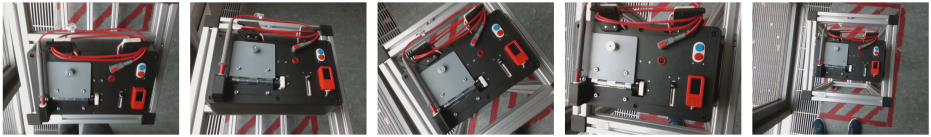
[2] CNN: Convolutional Neural Network

Object Recognition Training Samples

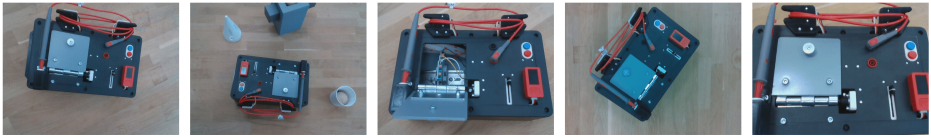
Low Light



Medium Light

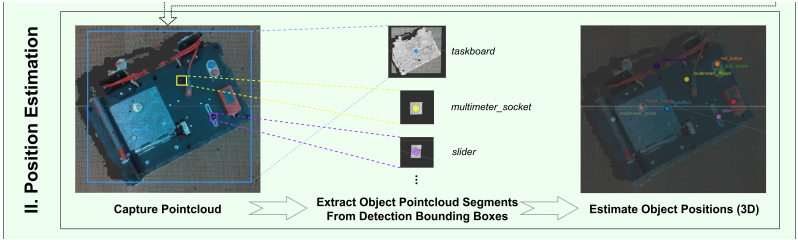


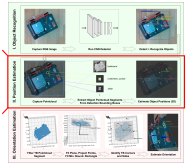
High Light



Stage II: Position Estimation

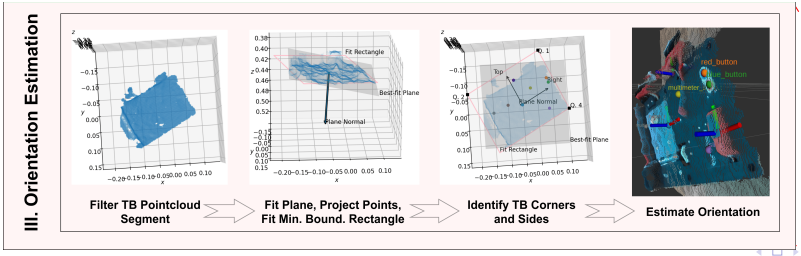
- **Strategy**: estimate object positions from their extracted segments of the aligned depth image (point cloud)
- Project all 3D depth points, P, onto the 2D RGB image plane using *central projection*
- For each object's bounding box (from stage I.), $\mathcal{B}$:
  - determine all 2D projected points, (u, v), that fall within $\mathcal{B}$
  - from all corresponding 3D depth points, (x, y, z), form the object's "cropped" point cloud segment: $\mathbf{P}^{object}$
- Object position estimate: mean of $\mathbf{P}^{object}$

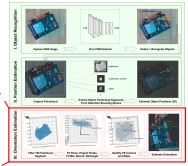




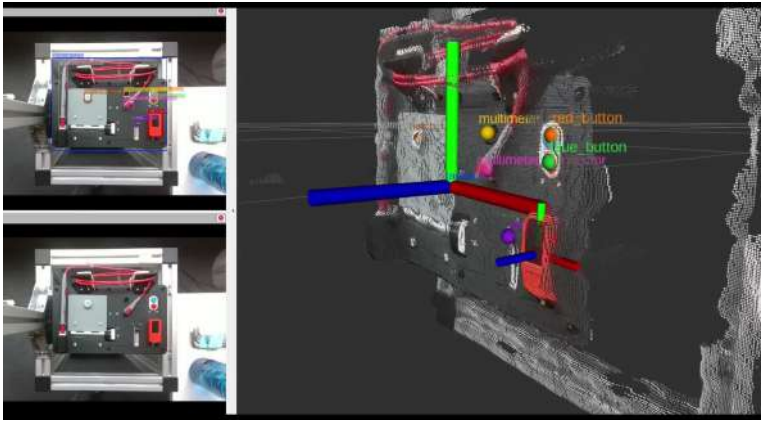
Stage III: Orientation Estimation

- Filter task board point cloud, $\mathbf{P}^{TB}$, to remove outliers using a clustering algorithm
- Find normal to best-fit plane (eigenvector of cov. matrix corresponding to smallest eigenvalue): $\vec{R}_3$
- Project points onto plane and fit minimum bounding rectangle, whose sides align with $\vec{R}_1$ and $\vec{R}_2$
- Use estimated 3D positions of components to correctly orient the two remaining axes
- Orientation estimate: orthonormal matrix, $\mathbf{R} = [\vec{R}_1, \vec{R}_2, \vec{R}_3]$





Video: Example Output

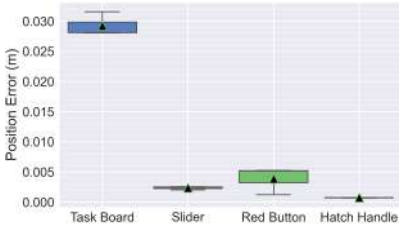


Quantitative Performance Analysis

- Object recognition: detection statistics determined by IoUs [3] between output and ground-truth bounding boxes (in image test set)
- Position estimation: absolute error between output and pre-defined, ground-truth 3D object positions (in point cloud test set)
- Future: orientation estimation error

Metric	Result
True positives (TPs)	262
False positives (FPs)	7
False negatives (FNs)	4
Precision	0.974
Recall	0.985
F1 score	0.979

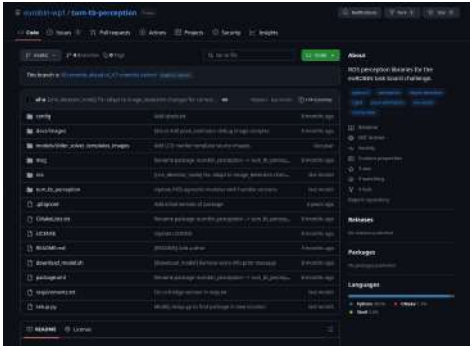
Table: Object Recognition (I) Performance Statistics

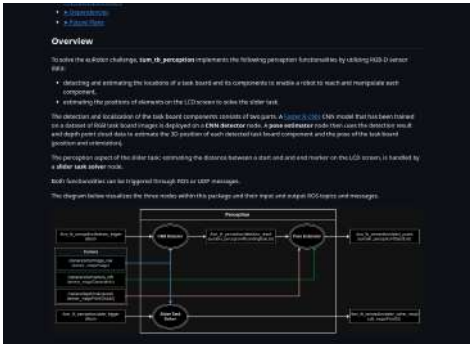


Plot: Position Estimation (II) Errors (four sample objects)

[3] IoU: Intersection over Union

Open-Source Software Package





- Available for ROS 1 (*noetic*) and ROS 2 (*humble*) [4]
- Utilized by multiple teams in the 1st euROBIN Coopetition (Nancy, France) ⇒ transferability and reusability

[4] https://github.com/eurobin-wp1/tum-tb-perception

Conclusions

Summary

- A reusable, modular object pose estimation robot skill
- Using *transfer learning* and direct depth data processing ⇒ data-efficient object recognition + 3D position estimation, easily adaptable to any object(s) for similar manipulation tasks
- Using prior information about the task ⇒ task-specific orientation estimation
- Software package: easy to integrate in perception stacks, adapt and extend
- Validated on the electronic task board and demonstrated as a reusable skill in a collaborative competition

Limitations and Future Work

- Orientation estimation is specific to the task board ⇒ need more general, task-agnostic strategy
- A quantitative analysis of orientation estimation performance (errors, accuracy, etc.)
- Continued development into a general-purpose robot vision skill

Special thanks to our team and the Konrad Zuse School of Excellence in Reliable AI (relAI)!

Konrad Zuse
School of Excellence
in Reliable AI

Questions?