CIRCU LI-ION

Computer Vision Coding Challenge

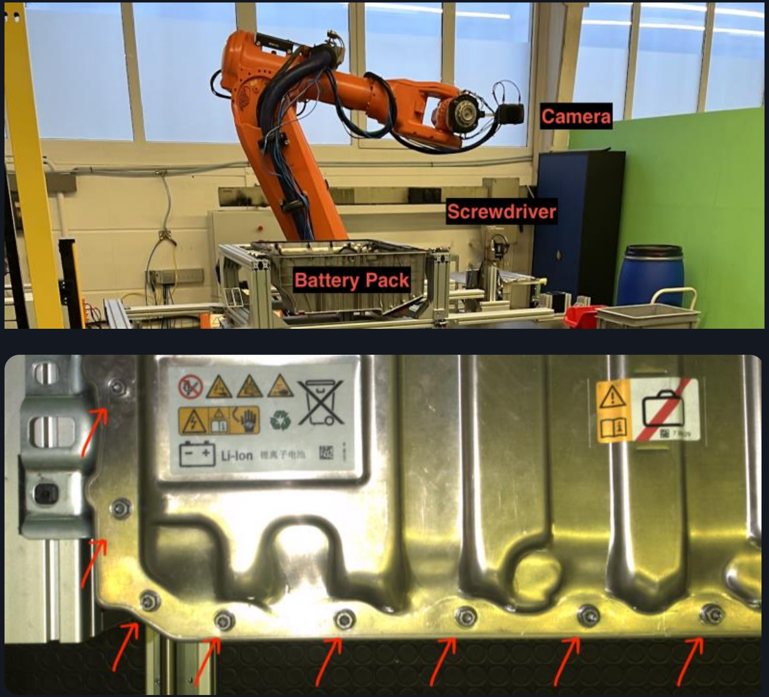
www.circuli-ion.com

The task

Your task is to implement the system to detect screws on an EV battery pack to unscrew them with a robot as a first step of disassembly.

You are provided a set of RGB images and pointclouds from the camera mounted on the robot. You will need to process RGB and / or 3D points to recover 3D pose (position and orientation) of every screw relative to the camera's optical center.

Bonus points: you are also given camera poses in robot coordinate system for every camera snapshot. Can you return all unique screw 3D poses in robot coordinate system?



An overview of the setup, with an example of the image from the camera, with screws marked with red arrows.

Details

Inside the zip archive you will find two sets of images taken for two different battery packs. Every folder contains data for a single camera snapshot.

Inside every folder there are 3 files – an image, a pointcloud in .ply format, and a camera pose as a 4x4 transformation matrix stored as a JSON file.

Pointcloud coordinates and translations are in mm.

Images and pointclouds are produced by a Zivid 2+ camera.

You can choose how to define the screw origin and axes.

If missing any other input, please make some reasonable assumptions.

What we expect

We would like you to showcase your coding skills but also reasoning to choose the right method for the task, given limited resources – in your case time (and maybe data, depending on which method you choose).

You are expected to deliver:

- A script that runs on a set of files and returns screw poses in a JSON file*. Please use either Python, C++ or Rust, and make sure you're happy with the quality of your code and documentation. Although Python is preferred, we definitely don't want to see a Jupyter Notebook.
- A short write-up of your approach, explaining the algorithm and any trade-offs you've made (<1 page) and a visualization of the results on the sample dataset.

Good luck and looking forward to seeing your ideas!

*Bonus points would be given for the implementation of an HTTP server hosting an API for detecting screws.