Pattern-based pose estimation for Tactile Internet

Background

- Tactile internet involves long-distance physical interaction over the internet, while requiring ultra low latency for natural and controllable interaction. This limits the distance over which can be operated.
- To overcome this, the controlled domain is observed and streamed to the master domain, and a simulation of the controlled domain is operated on instead.

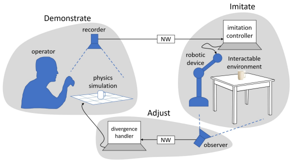

Figure 1: Overview of the Tactile Internet. This research focused on the "observer" part of the system. Illustration by Kees Kroep.

- To accurately stream the workspace to the master domain, objects in it must be precisely tracked. This can be done with several technologies.
- Opting for conventional RGB cameras might allow for accurate and very cost-effective pose estimation.
- Tracking of objects can be done through *pose estimation,* specifically through finding patterns in images using a *Perspective-n-Point* algorithm solver.

Method

- Set up Unity testbed with a programmable camera and checkerboard pattern of known dimensions, in 1:1 real-world scale.
- Find camera parameters using Zhang's method camera calibration.
- Take pictures of the checkerboard in various poses.
- Record ground truth of the poses.
- Send the pictures to a Python Flask server running the pose estimation algorithm.
- Estimate objects' pose by solving PnP with 3D-2D point correspondences.
- Calculate deviation of the estimated poses from ground truth.

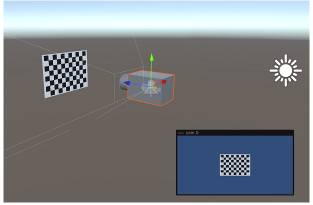

Figure 3: The virtual test bed in Unity, showing the camera, the checkerboard, and a preview of the camera's view.

Parameter	Value
Camera resolution	1920 × 1080
Checkerboard grid	10 × 7
Square size	5 mm
PC Processor	Intel i7-8750H @2.2GHz
PC RAM	24 GB

Figure 4: Test setup parameters used.

Results

- < 0.1 mm accuracy in position, however with some Z-axis error spikes. The origin of these errors is not clear, but might have to do with aliasing in the camera picture.
- Pitch and roll highly accurate between -45° and 45°, no pose detected beyond ±60°.
- Yaw highly accurate between -90° and 90°, high spikes outside that range, when the checkerboard goes "upside down".
- Pose estimation algorithm operates at ~40 Hz.

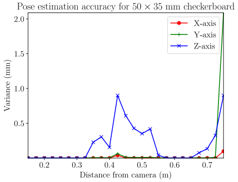

Figure 5: Positional error in 3 axes as distance to the camera increases.

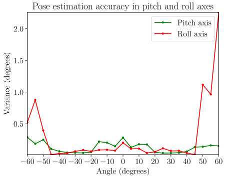

Figure 6: Angular error in pitch and roll axes (green and red respectively in Fig. 2).

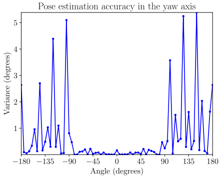

Figure 7: Angular error in the yaw axis (blue in Fig. 2).

Research question

How can RGB cameras be used to do pose estimation on objects in a Tactile Internet workspace?

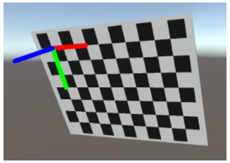

Figure 2: A checkerboard in the virtual test bed whose pose has been estimated and re-projected onto it as a Cartesian coordinate system.
X-axis is shown in red, Y-axis in green and Z-axis in blue.

Conclusion

- Checkerboard pattern recognition-based pose estimation proved to be effective in providing accurate position and rotation estimation.
- < 0.1 mm error attained in position estimation.
- < 0.5° error for most angles, however variance spikes remain when the checkerboard is upside down.
- The virtual test bed allows for rapid prototyping and testing of further developments.

Future work

- ArUCo markers are probably the most practical implementation of pattern pose estimation, instead of checkerboards. This was a stretch goal that couldn't be attained in time.
- A multi-cam setup might be investigated to improve tracking accuracy through sensor fusion.
- Camera resolutions should also be researched, what is the relationship between estimation accuracy and resolution?

Figure 8: An ArUCo marker.

Marijn Craenen
m.l.l.craenen@student.tudelft.nl

CSE3000
Research Project

Professor: Dr. Ranga Rao Venkatesha Prasad
Supervisor: Kees Kroep