ROBOTIC ARM FOR AUTOMATING TESTER CONNECTION

STUDENTS: ALNUR ELBERIER, CORBIN FERRIE, DEAN HOOVER, YIJIE LI

32

INTRODUCTION

- Robotic automation is a growing process that PACCAR wants to add to its Manufacturing Support Team. An automatic solution will free up test operators to use their time more efficiently and reduce the time required to test each dash, saving time and money for the plant and company.
- Our team worked to incorporate a Kawasaki robot arm capable of accurately connecting a plate containing multiple cable connectors to the plugs of the rear of a semi-truck dashboard.

OBJECTIVES & REQUIREMENTS

- **Objectives:**
  - Program a robotic arm capable of automating the connection of a tester cable to semi-truck dash assembly.
  - Reduce errors and damage caused by incorrectly installed cables.
  - Enable the operators to utilize their time more efficiently.
- **High Level Requirements:**
  - Control Kawasaki robot arm to reach target's position and transport the connector plate from an arbitrary position.
  - Use object detection software to locate dashboard from video of a camera.

Fig. 1: Connector Plate Attached to Kawasaki Robot Arm

Fig. 2: Dashboard

- **Milestones:**
  - Operate Moveit, camera sensor, and Kawasaki arm all at once in ROS.
  - Kawasaki arm follows routine designed in ROS.
  - Object Detection: Locate dashboard within a bounding box.

IMPLEMENTATION PIPELINE

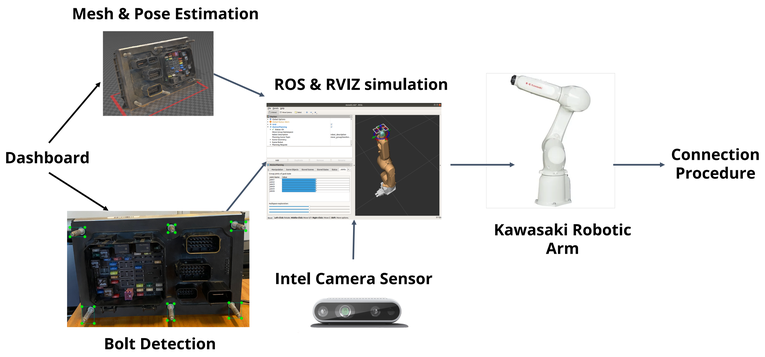

Fig. 3: Implementation of Test Procedure

POSE ESTIMATION FOR INITIAL LINEUP

- DOPE (Deep Object Pose Estimation) node estimates pose of real life objects.
- **Input:** *Rectified Image*
- **Publishes:** *Belief Map*
  - *Hidden layer to detect 9 corners of the object*
- **Data collection:** We used UE4 to collect a synthetic dataset from a 3d scanned mesh of the dashboard.
- **Training process:** The DOPE training script accepts NDDS (Nvidia Deep Learning Data Synthesizer) and we trained the model for 50 epochs.
- **Outcome:** In ROS, we visualized the estimated pose (red arrow) and the arm's end pose (green arrow) with a predetermined offset.

Fig. 5: Belief Map of Real Dashboard

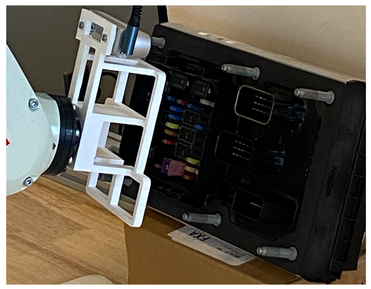

Fig. 6: End Effective Link Position After Dashboard Pose Estimation (Uncalibrated)

BOLT DETECTION FOR CALIBRATION

- Since the pose estimation model only takes rbg images as input, the result is not accurate enough to reach the dashboard. Therefore, we used bolt detection for calibration after the arm reaches the estimated pose.
- **Data collection:** We took hundreds of images of the bolts and manually labelled them using 'labelImg'.
- **Training process:** Tensorflow object detection API was used as backbone and we trained the model for 20,000 steps. Then the model is converted to .tflite format.
- **Outcome:** The model successfully detects all 4 bolts of interest at close distance as shown in Fig. 7.

Fig. 7: Bolt Detection Model Test Visualization

TESTING & RESULTS

- All arm movements/planned trajectories are visualized in simulation.
- DOPE estimates pose of the dashboard.
- End effective link of the arm reaches the estimated pose with a predefined offset
- Camera detects 4 bolts of interest and calibrates itself.
- Plate connects to dash by following the dash orientation vector.

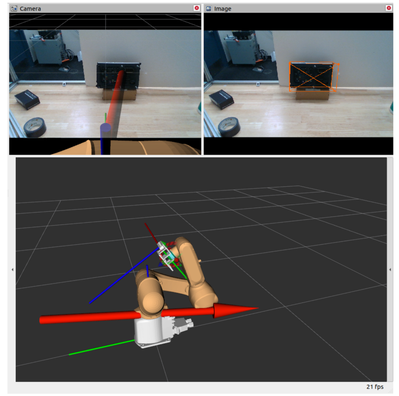

Fig. 8: Pose Estimation w/ Arrow Visualization in RVIZ

CONCLUSION

- Created a ROS workspace directory where Moveit, Intel RealSense Camera sensor, and a pose program can be launched all at once.
- Pose estimation software detects dashboard from within a bounding box so Kawasaki robot can track and align itself to the dashboard.
- Kawasaki arm successfully traveled to dashboard at an almost perfectly aligned position (uncalibrated).

Future Work and References

- Although the models are accurate enough for testing purposes, environmental noise such as lighting can affect their performance. Mechanical parts can be introduced to increase noise tolerance.
- The kawasaki RS007N robotic arm has 6 degrees of freedom, which limits the probability of finding a unique solution when planning trajectory. 7 joint arms or linear actuators can help mitigate the issue.
- References:
  - 3D Pose Estimation - ROS - Realsense d435: https://github.com/yehengchen/DOPE-ROS-D435
  - Bolt Detection - TensorFlow Model - https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md
    - Detection Model: SSD MobileNet V2 FPNLite 320x320
  - Deep Object Pose Estimation Package: https://github.com/NVlabs/Deep_Object_Pose
  - Synthetic Data UE4 DOPE: https://github.com/yehengchen/Synthetic-Data-UE4

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

INDUSTRY ADVISER: CHRIS ROBERTS
FACULTY ADVISER: BRIAN JOHNSON
SPONSOR: PACCAR