experience what's inside™

MoveIt Workshop 2019 Macau

Build Advanced Industrial Robot Usages with Intel OpenVINO and MoveIt

Yu Yan
Intel SSP Robotics Software Engineering

Agenda

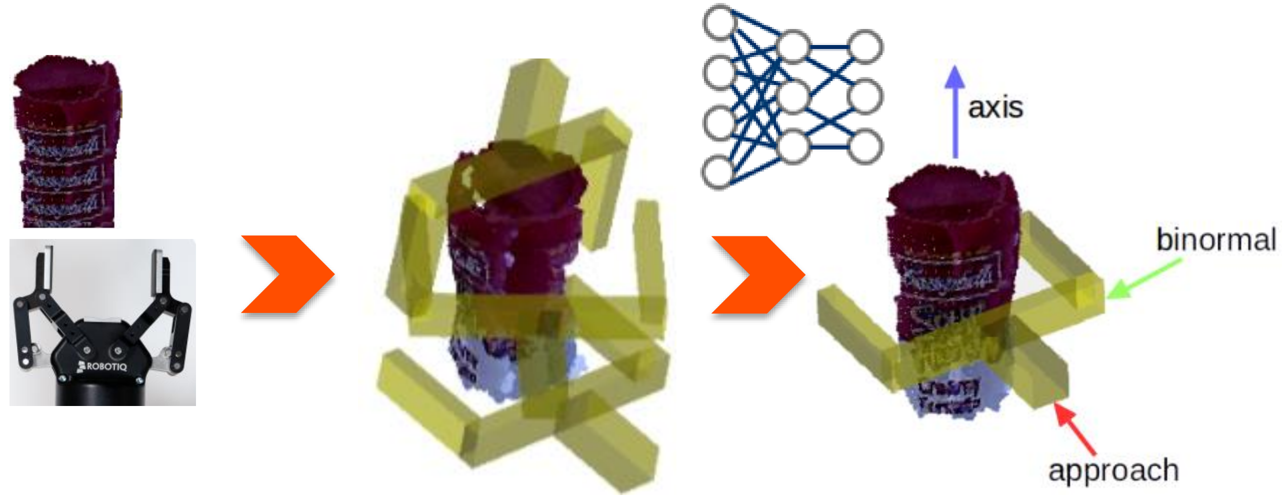
- Who We Are
- Grasp Detection
- OpenVINO™ Toolkit Grasp Detection
- OpenVINO™ Toolkit Grasp Library as MoveIt Plug-in
- MoveIt Hand-Eye Calibration
- MoveIt Example Apps
- Build MoveIt into Advanced Industrial Robot Controllers with ACRN

Who We Are

- Who we are:
  - SSP Robotics Software Engineering Team from Intel Open Source Technology Center (OTC)
- Intel ROS2 projects:
  - ROS2 Realsense Camera: https://github.com/intel/ros2_intel_realsense
  - ROS2 OpenVINO: https://github.com/intel/ros2_openvino_toolkit
  - ROS2 Movidius NCS: https://github.com/intel/ros2_intel_movidius_ncs
  - ROS2 Object Analytics: https://github.com/intel/ros2_object_analytics
  - ROS2 Object Map: https://github.com/intel/ros2_object_map
  - ROS2 Grasp Library: https://github.com/intel/ros2_grasp_library
  - ROS2 Navigation: https://github.com/ros-planning/navigation2
  - Intel® Robot DevKit (RDK): https://github.com/intel/robot_devkit

Grasp Detection

Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp Pose Detection in Point Clouds
The International Journal of Robotics Research, Vol 36, Issue 13-14, pp. 1455 - 1473. October 2017



Point Cloud Library (PCL) or Depth Hand Geometry

Grasp Candidates

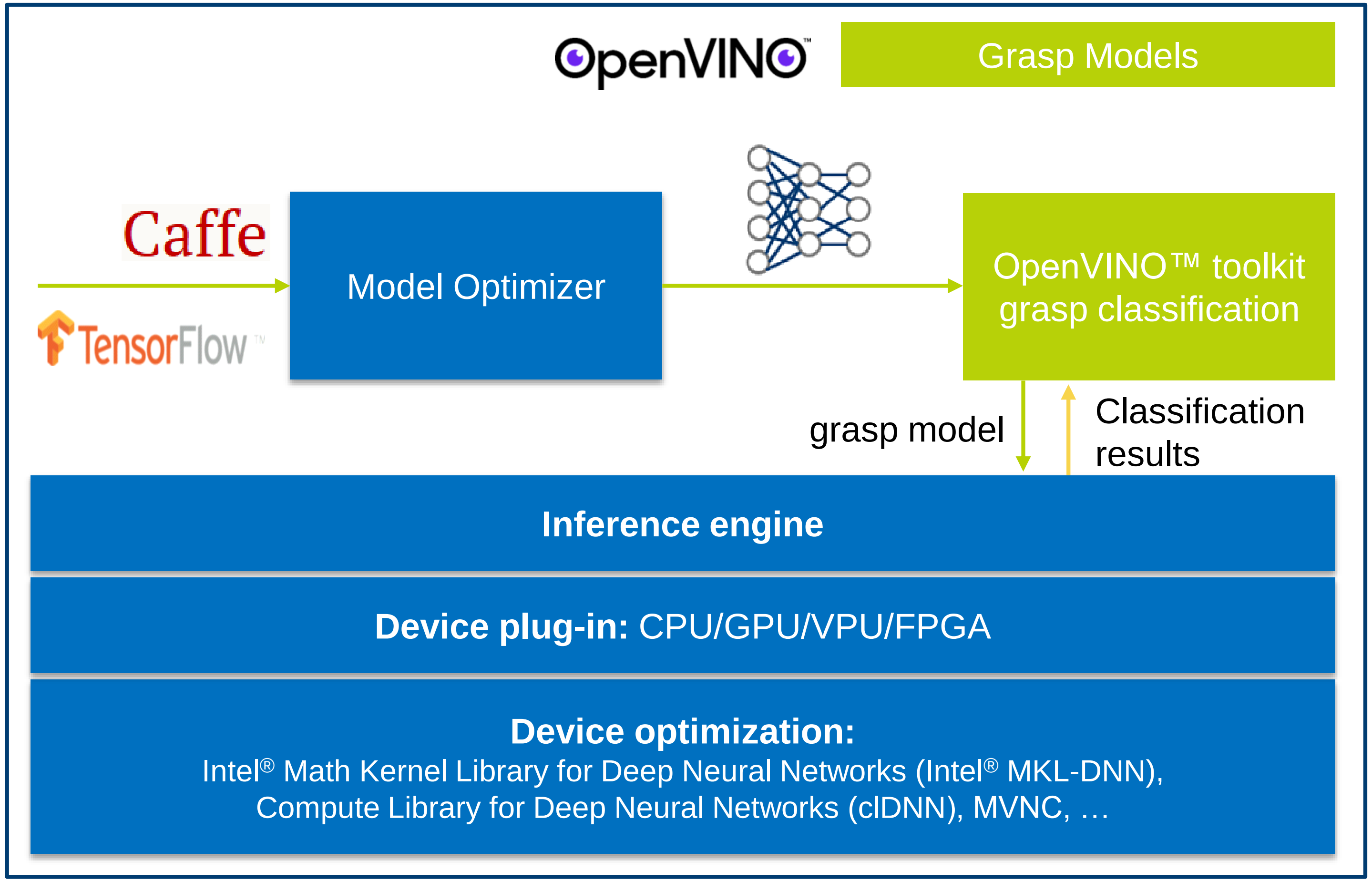
Grasp Classification

Robotiq gripper photo: https://robotiq.com/products/2f85-140-adaptive-robot-gripper

*Other names and brands may be claimed as the property of others.

- Convolutional Neural Networks (CNN)-based grasp detection
  - Dex-Net* TensorFlow
  - Grasp Pose Detection (GPD) Caffe
- Grasp datasets
- Grasp success rate
- Inference time

OpenVINO™ Toolkit Grasp Detection



Grasp Models

Caffe

TensorFlow

Model Optimizer

OpenVINO™ toolkit grasp classification

grasp model

Classification results

Inference engine

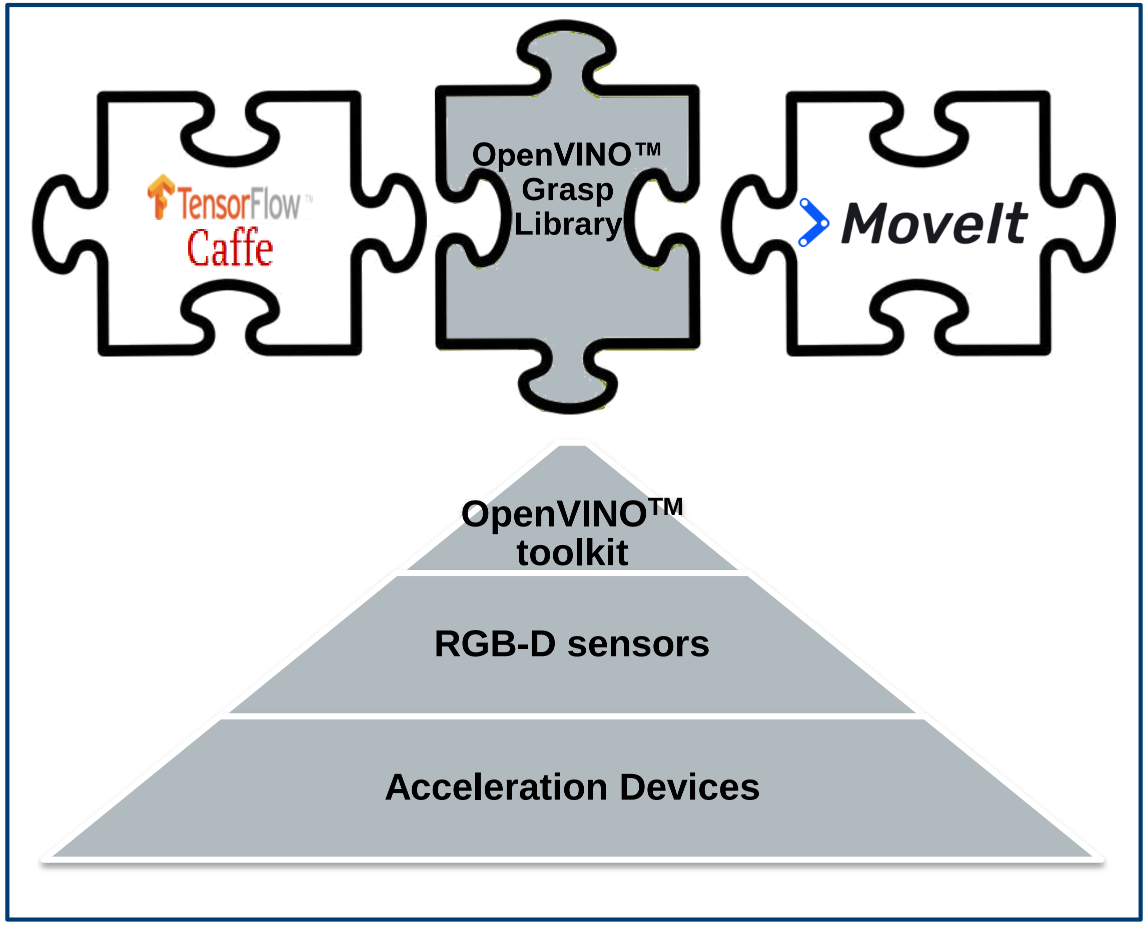
Device plug-in: CPU/GPU/VPU/FPGA

Device optimization:
Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN), Compute Library for Deep Neural Networks (clDNN), MVNC, …

- 3~4X(2018), 6~8X(2019) performance gain on inference time
- 25% CPU offload
- https://github.com/atenpas/gpd

*Other names and brands may be claimed as the property of others.

OpenVINO™ Toolkit Grasp Library as MoveIt Plug-in



TensorFlow

Caffe

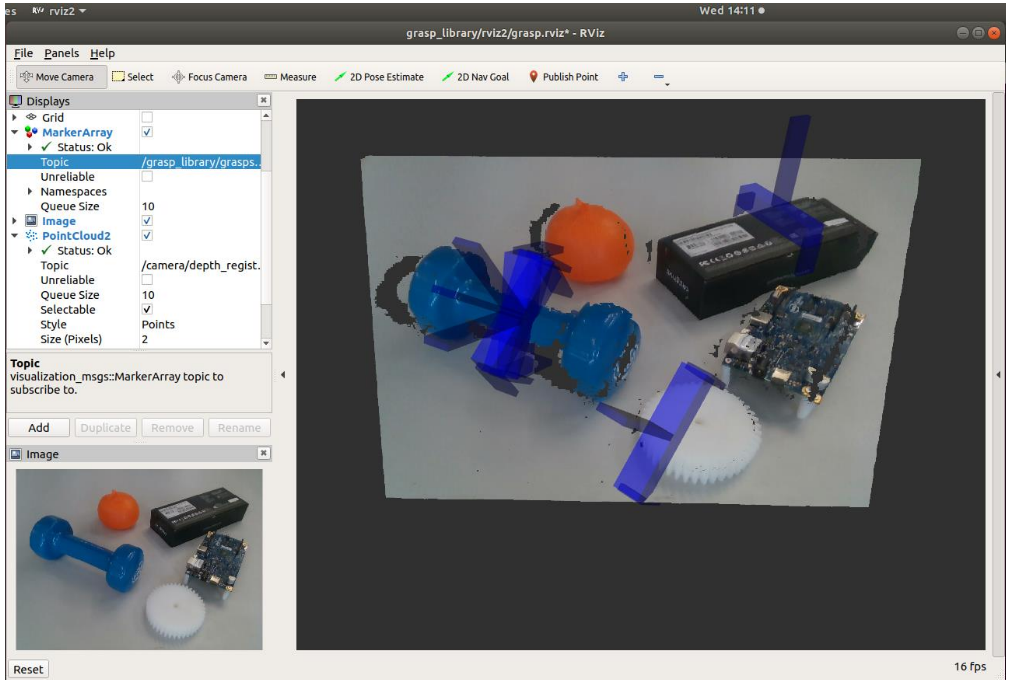
OpenVINO™ Grasp Library

MoveIt

OpenVINO™ toolkit

RGB-D sensors

Acceleration Devices



Grid	□
MarkerArray	☑
✓ Status: Ok	
Topic	/grasp_library/grasps..
Unreliable	□
Namespaces	
Queue Size	10
Image	☑
PointCloud2	☑
✓ Status: Ok	
Topic	/camera/depth_regist.
Unreliable	□
Queue Size	10
Selectable	☑
Style	Points
Size (Pixels)	2

Topic
visualization_msgs::MarkerArray topic to subscribe to.

Visualization of grasp detection results

https://github.com/intel/ros2_grasp_library
Added to website “moveit.ros.org”

*Other names and brands may be claimed as the property of others.

How to use ros2_grasp_library

- Launch grasp planner

```
ros2 run grasp_ros2 grasp_ros2 __params:=src/ros2_grasp_library/grasp_apps/random_pick/cfg/random_pick.yaml
```

```
using GraspPlanning = moveit_msgs::srv::GraspPlanning;
static std::shared_ptr<GraspPlanning::Response> result_ = nullptr;
……

  // create client for grasp planning
  auto client = node_->create_client<GraspPlanning>("plan_grasps");

  // request grasp poses
  auto request = std::make_shared<GraspPlanning::Request>();
  auto result_future = client->async_send_request(request);
  ……

  if (moveit_msgs::msg::MoveItErrorCodes::SUCCESS == result_future.get()->error_code.val) {
        result_ = result_future.get();
        RCLCPP_INFO(node_->get_logger(), "Response received %d", result_->error_code.val);
  } else continue;

  geometry_msgs::msg::PoseStamped p = result_->grasps[0].grasp_pose;
……
```

Create Grasp Service Client

Request Grasp Pose

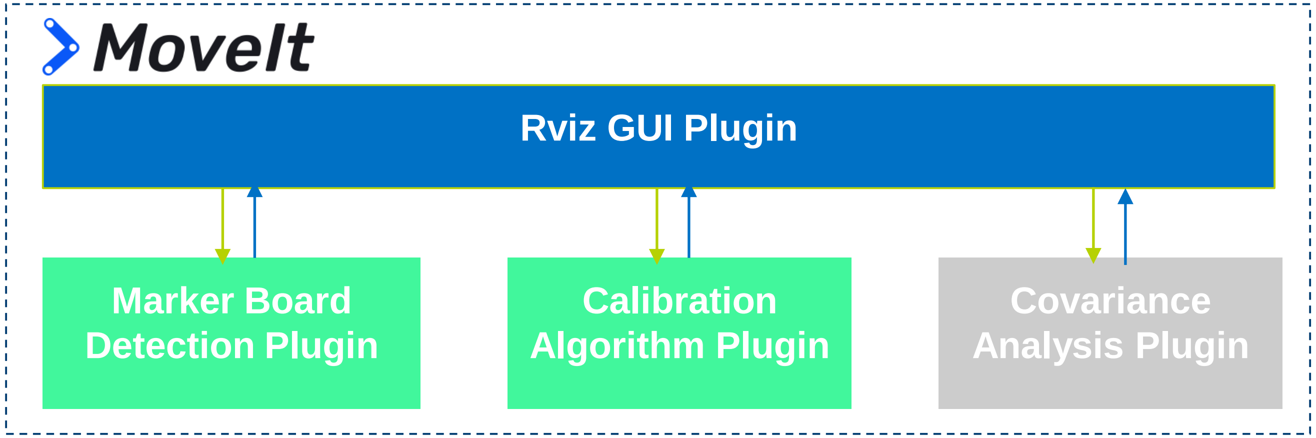
Get Grasp Pose Result

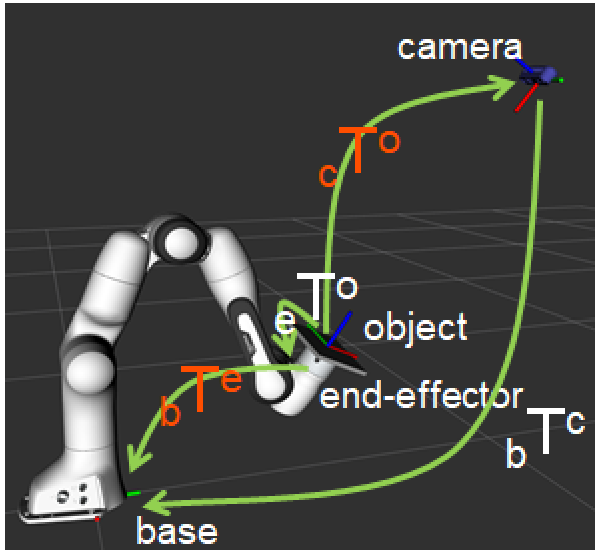
```
GraspDetectorGPD:
  ros__parameters:
    cloud_topic: /camera/pointcloud
    device: 1 # 0:CPU, 1:GPU, 2:VPU
    workspace: [-0.35, 0.35, -0.6, 0.1, 0.0, 1.0]
    finger_width: 0.005
    hand_outer_diameter: 0.100
    hand_depth: 0.038
    hand_height: 0.020
    ……

GraspPlanner:
  ros__parameters:
    grasp_frame_id: "base"
    grasp_approach: [0.0, 0.0, -1.0]
    grasp_approach_angle: 0.523
    grasp_offset: [-0.000, -0.002, 0.000]
    eef_offset: 0.174
    eef_yaw_offset: -0.7854 # M_PI/4
    ……
```

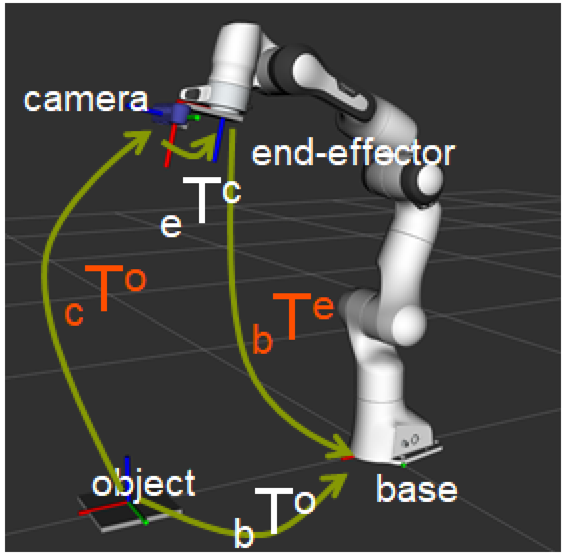
MoveIt Hand-eye Calibration

- Consists of four MoveIt plugins:
  - Rviz GUI Plugin
  - Marker Detection Plugin
  - Calibration Algorithm Plugin
  - Covariance Analysis Plugin (WIP)
- Flexible architecture, each plugin can be replaceable
- Easy to use interface
- Calibration process understandable and visualized in 2/3D
- State-of-the-art algorithms integrated
- Can be used to:
  - Eye-to-hand calibration
  - Eye-in-hand calibration
- PRs under review in MoveIt:
  - MoveIt#1558
  - MoveIt#1559
  - MoveIt#1560


MoveIt
Rviz GUI Plugin
Marker Board Detection Plugin
Calibration Algorithm Plugin
Covariance Analysis Plugin



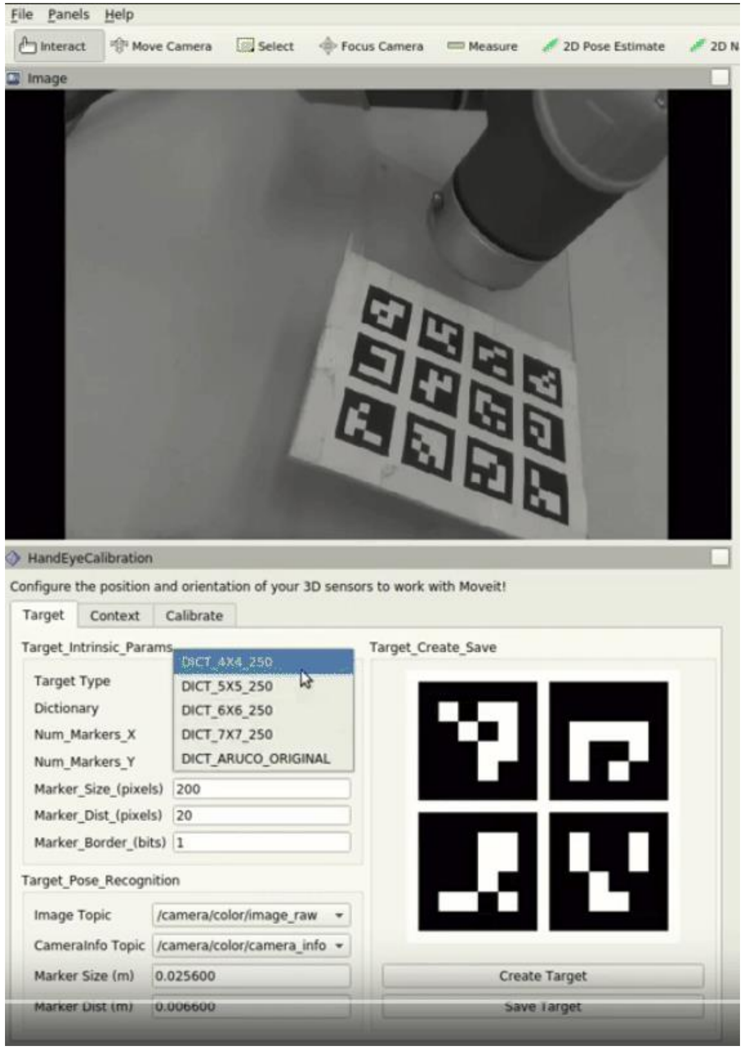
Eye-to-hand



Eye-in-hand

MoveIt Hand-eye Calibration

- Rviz GUI Plugin consists of:
  - Target tab widget (Used for setting and visualizing calibration board detection)
  - Context tab widget (Used for setting calibration context and initial guess)
  - Calibrate tab widget (Used for calibration computing)

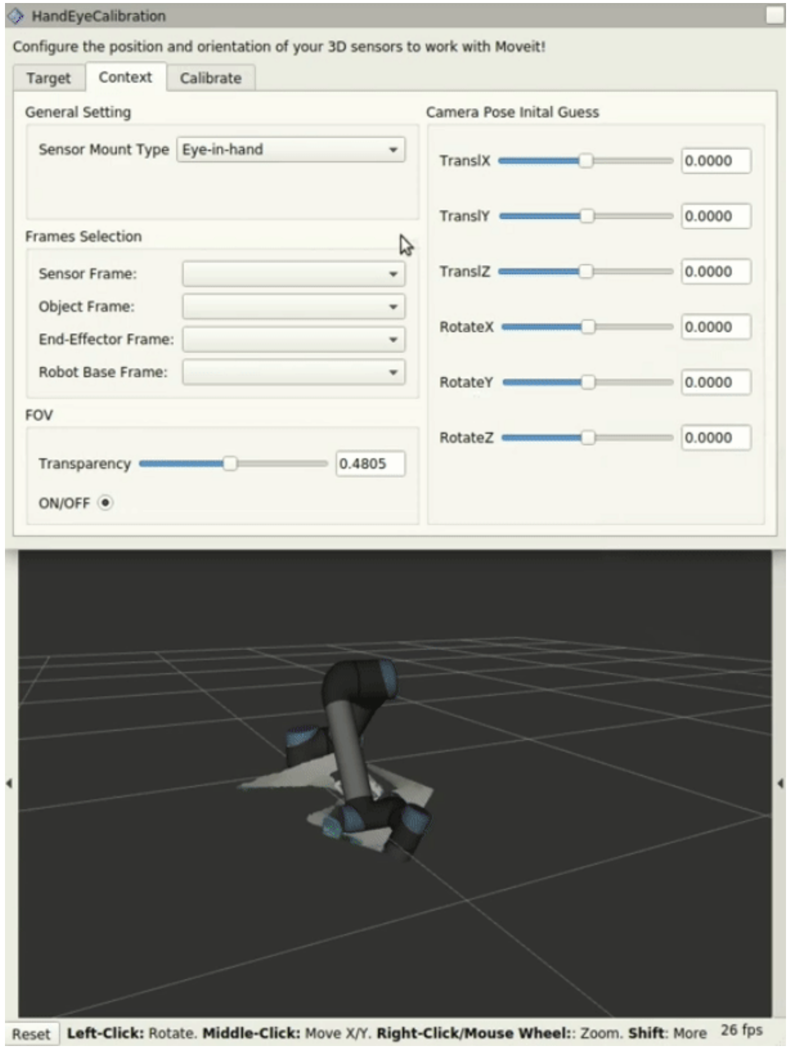


Configure the position and orientation of your 3D sensors to work with Moveit!

Marker_Size_(pixels) 200
Marker_Dist_(pixels) 20
Marker_Border_(bits) 1

Image Topic /camera/color/image_raw
CameraInfo Topic /camera/color/camera_info
Marker Size (m) 0.025600
Marker Dist (m) 0.006600

Target tab widget



Configure the position and orientation of your 3D sensors to work with Moveit!

Sensor Mount Type Eye-in-hand

Transparency 0.4805

ON/OFF ●

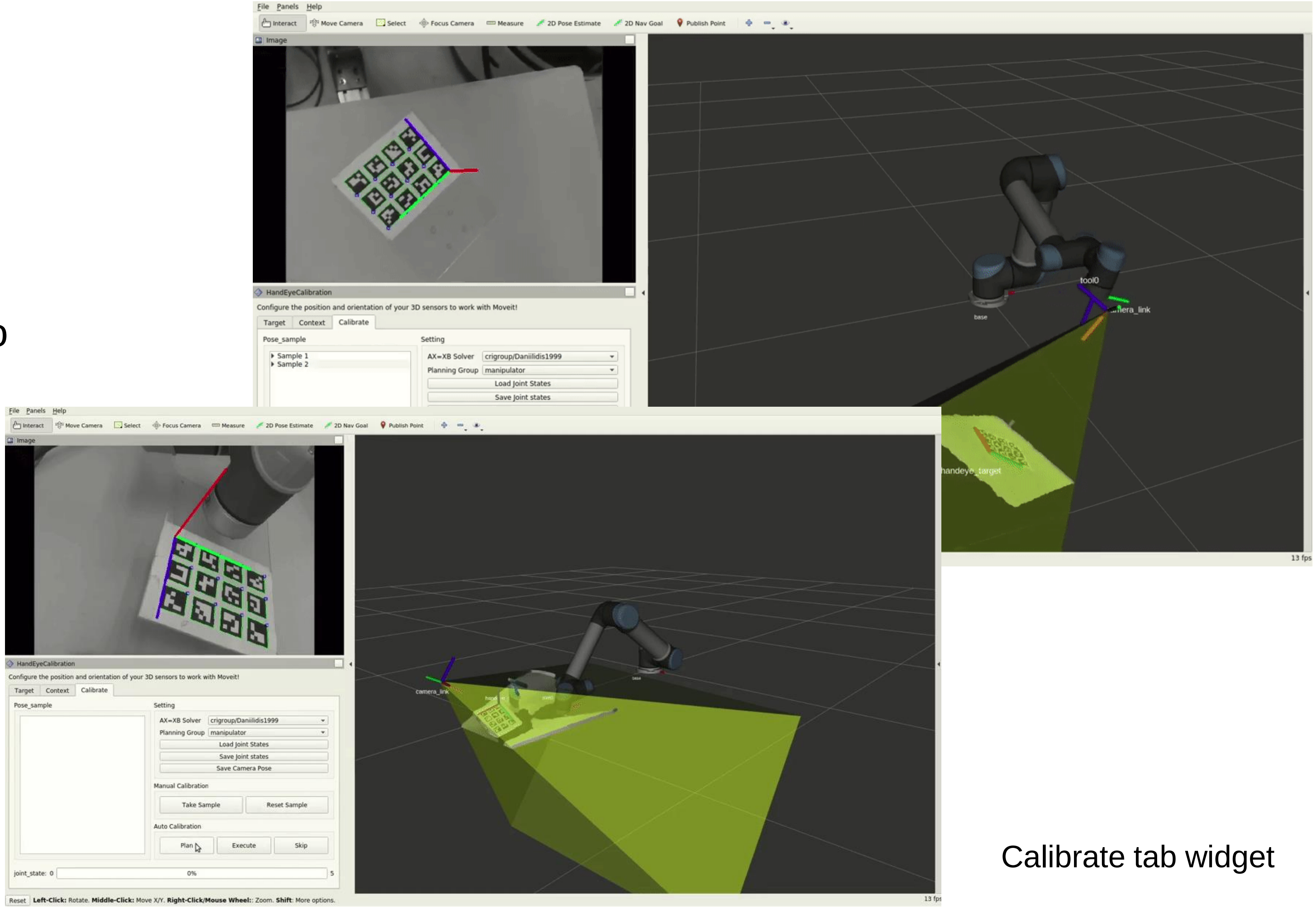
TranslX 0.0000
TranslY 0.0000
TranslZ 0.0000
RotateX 0.0000
RotateY 0.0000
RotateZ 0.0000

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More

Context tab widget

MoveIt Hand-eye Calibration

- Future improvements:
  - Thank @felixvd for reviewing the tool and providing good suggestions
  - Split the calibration tab widget into four tab widgets: (Motion, Collect, Calculate and Test)
  - Add Covariance Analysis Plugin (WIP)



Configure the position and orientation of your 3D sensors to work with MoveIt!

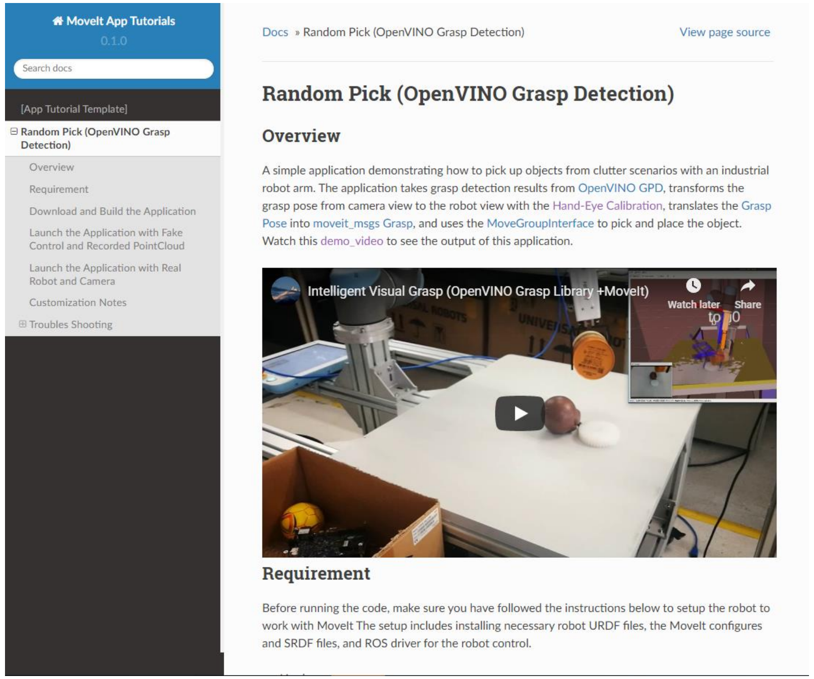
AX=XB Solver: crigroup/Daniilidis1999

Planning Group: manipulator

Calibrate tab widget

MoveIt Example Apps

Screen snapshot of “MoveIt Example Apps Tutorial”


MoveIt App Tutorials
0.1.0
Search docs
[App Tutorial Template]
⊟ Random Pick (OpenVINO Grasp Detection)
Overview
Requirement
Download and Build the Application
Launch the Application with Fake Control and Recorded PointCloud
Launch the Application with Real Robot and Camera
Customization Notes
⊞ Troubles Shooting
Docs » Random Pick (OpenVINO Grasp Detection)
View page source
Intelligent Visual Grasp (OpenVINO Grasp Library +MoveIt)
Watch later
Share

Random Pick (OpenVINO Grasp Detection)

Overview

A simple application demonstrating how to pick up objects from clutter scenarios with an industrial robot arm. The application takes grasp detection results from OpenVINO GPD, transforms the grasp pose from camera view to the robot view with the Hand-Eye Calibration, translates the Grasp Pose into moveit_msgs Grasp, and uses the MoveGroupInterface to pick and place the object. Watch this demo_video to see the output of this application.

Requirement

Before running the code, make sure you have followed the instructions below to setup the robot to work with MoveIt The setup includes installing necessary robot URDF files, the MoveIt configures and SRDF files, and ROS driver for the robot control.

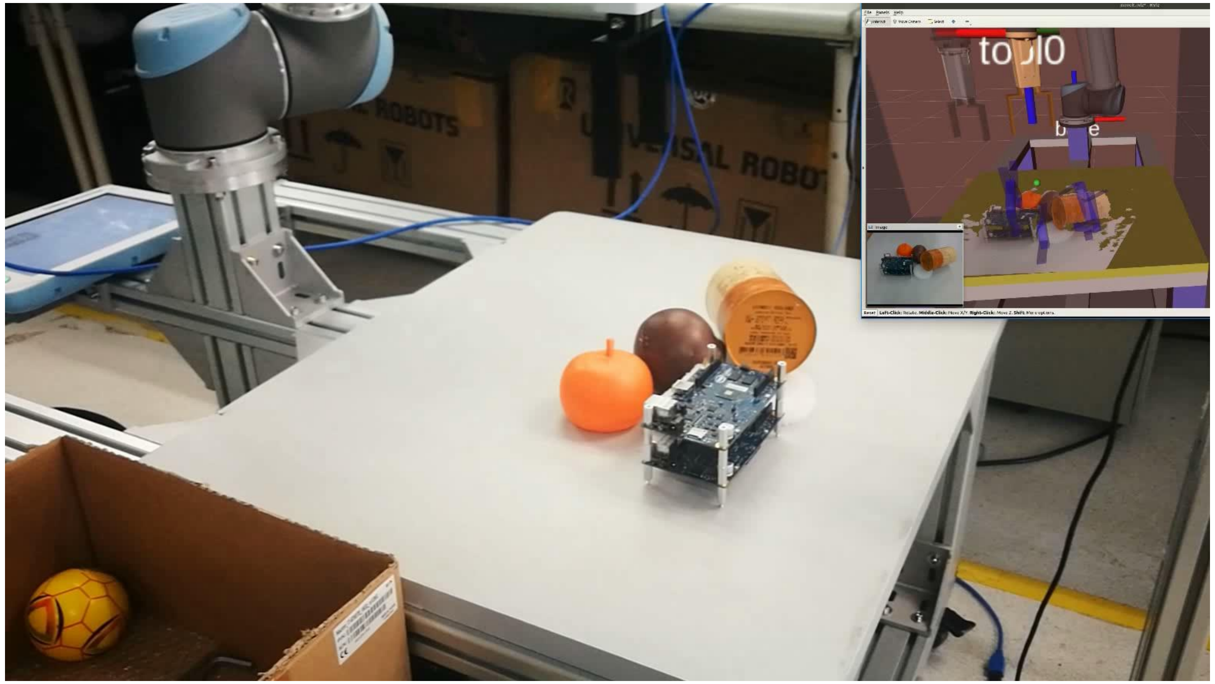
https://github.com/ros-planning/moveit_example_apps

- ○ moveit_example_apps #PR2
- ○ moveit_example_apps #PR3
- ○ moveit_example_apps #PR5

Video: Intelligent Visual Grasp

(OpenVINO™ Grasp Library + MoveIt)

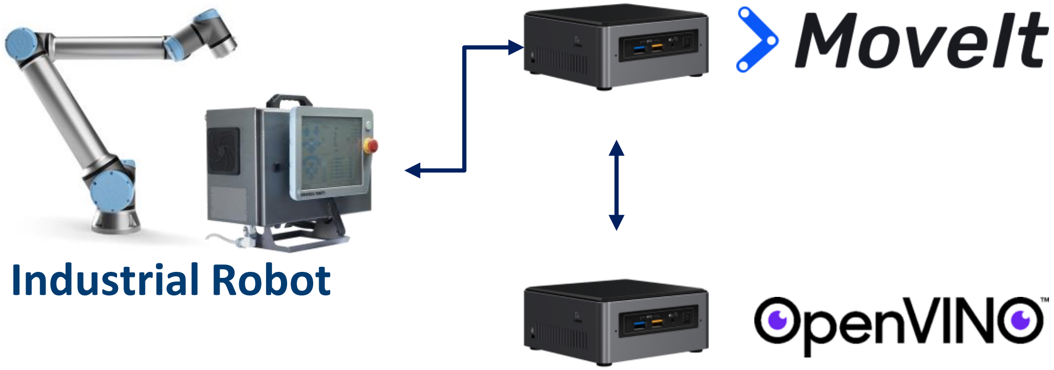
Youtube https://www.youtube.com/watch?v=b4EPvHdidOA&list=PLxCmGJeiLgoxq3uqcCVSYnSJ9iQk1L9yP



Build MoveIt into Advanced Industrial Robot Controllers with ACRN

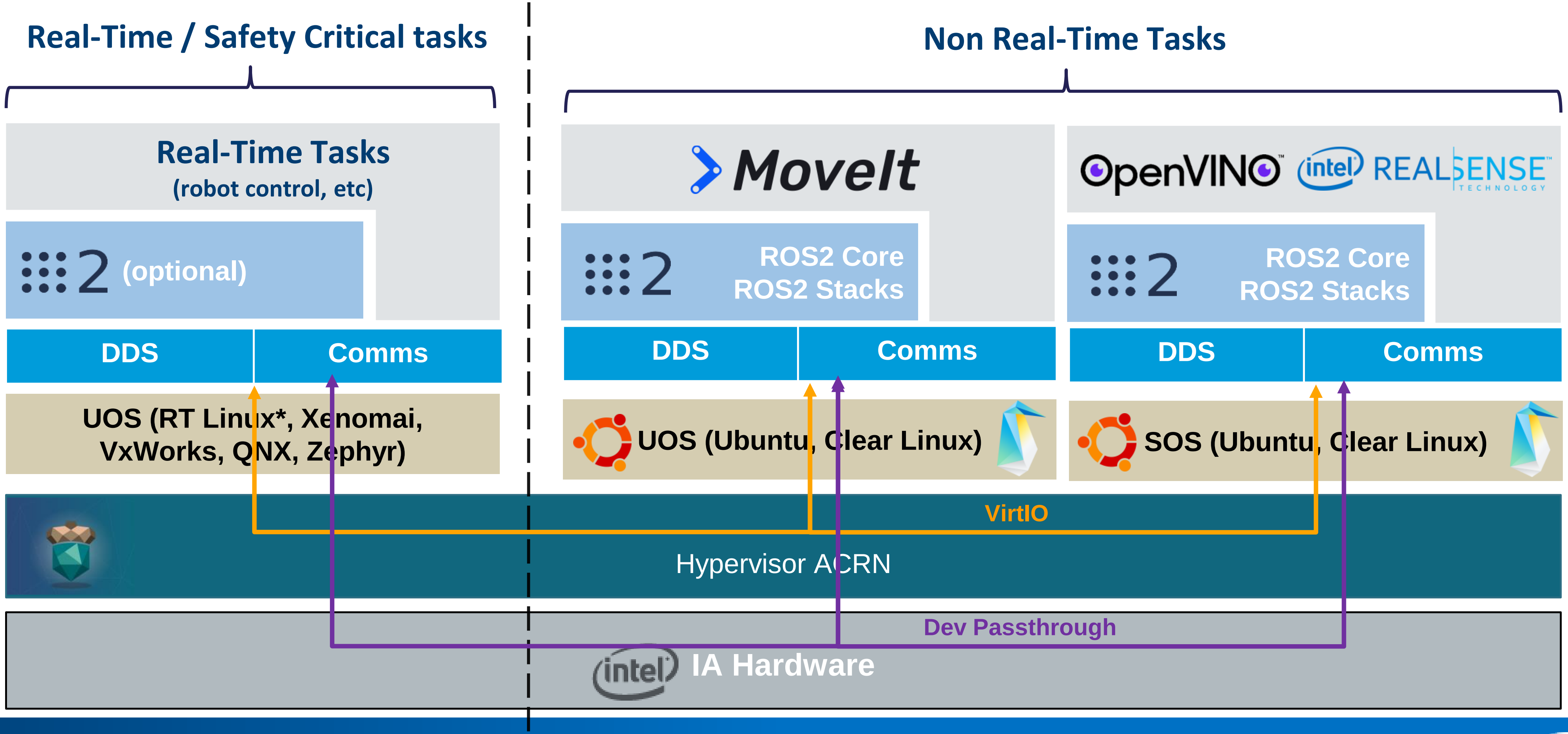
- Usual way to implement MoveIt on industrial robots:
  - Machine 1 (Hard real-time OS, motor level control)
  - Machine 2 (Non real-time OS, MoveIt motion planning)
  - Machine 3 (Non real-time OS, Perception)

- It's useful to support real-time and non real-time development safely and effectively at a single machine
- Using multiple machines is not good at:
  - System safety and reliability
  - Space possession
  - Adaptation challenges
  - Power consumption



*Other names and brands may be claimed as the property of others.

Build MoveIt into Advanced Industrial Robot Controllers with ACRN



Real-Time / Safety Critical tasks

Non Real-Time Tasks

Legal Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel RealSense, OpenVINO, and the OpenVINO logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.