Intel® RealSense™ Product Family

D400 Series Calibration Tools

User Guide

July 2020

Version 2.11.0.0

Contents

1 Introduction 9

1.1 Purpose and Scope of this Document 9
1.2 Components 9
1.3 Class 1 Laser and Caution 9
1.4 Hardware Requirements 10
1.5 Software Requirements 11

2 Overview 12

2.1 Calibration Parameters 12
2.2 Types of Dynamic Calibration 12
2.2.1 Target Dynamic Calibration (Depth Scale Calibration) 13
2.2.2 Target-less Dynamic Calibration (Rectification Calibration) 13
2.3 Depth Quality Tool 13

3 Installation 14

3.1 Installation on Linux (Ubuntu 16.04 and Ubuntu 18.04) 14
3.1.1 Install 3rd-party dependencies 14
3.1.2 Calibration Tools and API Package Installation 15
3.1.3 Checking Package Installation 16
3.1.4 Video4Linux backend 17
3.1.5 Test Installation 17
3.1.6 Debian Package Removal 17
3.2 Installation on Windows 18
3.2.1 Install 3rd-party dependencies 18
3.2.2 Package installation 18

4 Calibration with Dynamic Calibrator 22

4.1 Dynamic Calibrator Graphical User Interface 22
4.2 Dynamic Calibrator Command Line Interface 29
4.3 Connect camera to host system 30
4.4 Multiple camera support 30
4.4.1 List all camera devices with Graphical User Interface 30
4.4.2 List all camera devices with Command Line Interface (CLI) 31
4.4.3 Perform calibration on selected device with Command Line Interface (CLI) . 32
4.4.4 Stop Calibration 32
4.5 Targeted Calibration 34
4.5.1 Setup Calibration Target 34
4.5.2 Running Dynamic Calibrator with targeted calibration 34
4.5.3 Auto Exposure Adjustment during Targeted Calibration 40
4.5.4 Target calibration phases 41

5 Advanced Calibration Processes 48

5.1 Target-less Calibration 48
5.2 Hybrid Calibration 50

5.3 Calibration with a Robot (Gimbal) 54
5.3.1 Setup 54
5.3.2 Gimbal 55
5.3.3 Run Calibration with the Robot 55
6 Handling Calibration Data with CustomRW Tool 58
6.1 Restore Calibration on Device 59
6.2 Read Calibration from Device 59
6.3 Write Calibration to Device 62
6.4 Read and Write RAW Calibration Tables 63
6.5 Read and Write Custom Fisheye Data 65
7 Developing Custom Apps with Calibration API 67
7.1 API Package Location 67
7.2 API Package Structure 67
7.3 Compiling Examples 68
7.4 API Documentation 69
8 Troubleshooting 70
9 Appendix A - Calibration Check 72
9.1 Quick Check 72
9.2 Accuracy Check 73
10 Appendix B - Calibration Process Highlights 74
10.1 Targeted Calibration 74
11 Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup 82
11.1 Introduction 82
11.2 Dynamic Calibration Flow 82
11.3 Printed Target Setup 83
11.4 Phone Target Setup on Apple iPhones Devices 87
11.4.1 Locate the app on Apple App Store 87
11.4.2 Install the App on iPhone 87
11.4.3 Running the app on phone 89
11.4.4 Perform Dynamic Calibration 91
11.5 Phone Target Setup on Android Phone Devices 92
11.5.1 Locate the app on Google Play Store 92
11.5.2 Install the App on Android Phone 92
11.5.3 Running the app on phone 94
11.5.4 Perform Dynamic Calibration 96
12 Appendix D - Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App 97
12.1 Validated IOS Devices 97
12.2 Validated AOS devices 97
12.3 Devices with accuracy issue in phone target 98
13 Appendix E – Setting Up and Program Gimbal Robot 99

Tables

Table 1-1. Supported Intel® RealSense™ Depth Modules and Depth Cameras 10
Table 1-2. Software Requirements 11
Table 3-1: Calibration Tools and API Packages 15
Table 4-1: Dynamic Calibrator Command Line Options 29
Table 6-1: CustomRW Command Line Options 58
Table 8-1 Troubleshooting 70

Figures

Figure 1-1. Intel® RealSense™ Depth Camera D435 (Depth Module Integrated) 10
Figure 1-2. Intel® RealSense™ Depth Module D410 with Vision Processor D4 Card (Non-integrated Bare Depth Module) 10
Figure 4-1. Calibration Target Displayed on Phone 34
Figure 4-2 Dynamic Calibrator Initial Start Window 39
Figure 4-3 Dynamic Calibrator app warm up window 39
Figure 4-4 Dynamic Calibrator Targeted Rectification Phase 42
Figure 4-5 Dynamic Calibrator Targeted Rectification Phase (continued) 42
Figure 4-6 Dynamic Calibrator Targeted Rectification Phase (continued) 43
Figure 4-7 Dynamic Calibrator Scale Phase 44
Figure 4-8 Dynamic Calibrator Scale Phase (continued) 44
Figure 5-1 Target-less Dynamic Calibration with Entire FOV 49
Figure 5-2 Target-less Dynamic Calibration with Center FOV 50
Figure 5-3 Hybrid Dynamic Calibration – Target-less Phase 51
Figure 5-4 Hybrid Dynamic Calibration – Targeted Scale Calibration Phase 51
Figure 5-5 Hybrid Dynamic Calibration – Targeted RGB Calibration Phase 52
Figure 5-6 Hybrid Dynamic Calibration – Device and Target Aligned 53
Figure 5-7 Hybrid Dynamic Calibration – Device and Target Aligned 54
Figure 5-8 Dynamic Calibration on Robot – Sample Setup 55
Figure 5-9 Dynamic Calibration on Robot 56
Figure 5-10 Dynamic Calibration on Robot 56
Figure 5-11 Dynamic Calibration on Robot 57
Figure 6-1 Restore Calibration on Selected Device 59
Figure 6-2 Read Calibration on Selected Device 60
Figure 6-3 Write Calibration on Selected Device 62
Figure 6-4 Read Calibration RAW Data on Selected Device and Display on Screen 64
Figure 6-5 Read and Write Calibration RAW Data on Selected Device 65
Figure 6-6 Read Fisheye Custom Data on Selected Device and Display on Screen 66
Figure 9-1. Calibration Check – Camera in Calibration 72
Figure 9-2. Calibration Check – Camera Out of Calibration 73
Figure 11-1 dynamic calibration flow 82
Figure 11-2 Target Image 83
Figure 11-3 Target Print Options 84
Figure 11-4 Target Verification 85
Figure 11-5 Printed target setup 86

Revision History

Revision Number	Description	Revision Date
2.5.2.0	First document for Calibration Tool version 2.5.2.0	Jan 2018
2.6.4.0	Support RGB calibration/ Updated kernel patching instruction on Linux	August 2018
2.6.8.0	Support D435i Custom Calibration Data R/W API Support hardware sync in RGB calibration Support calibration raw data read/write Support Windows 10 RS4	December 2018
2.8.0.0	Improved calibration for robotics UI dynamic adjust to image aspect ratios	March 2019
2.11.0.0	Support Intel® RealSense™ Depth Camera D455 New DC algo compatible with production calibration algo v5.1 Support Linux (Ubuntu 16.04 and Ubuntu 18.04) on ARM platforms (Nvidia Jetson TX2, Xavier, and Firefly RK3399) Fix libpng dependency issue on Ubuntu 18.04 Supports Ubuntu 18.04 kernel 5.x Calibration API is combined into Calibration Tool package	July 2020

1 Introduction

1.1 Purpose and Scope of this Document

This document is a comprehensive user guide for calibration tools released as part of Intel® RealSense™ Product Family D400 Series Calibration Tools and API software release.

The usage of calibration Tool API is separately covered in document "Intel® RealSense™ Product Family D400 Series Calibration Tools API programmer's guide"

1.2 Components

The Intel® RealSense™ Product Family D400 Series Calibration Tools and API software release package includes:

- Calibration Tools including Dynamic Calibrator
- Calibration API libraries, headers, and sample application
- Intel® RealSense™ Dynamic Target phone application

1.3 Class 1 Laser and Caution

The Intel® RealSense™ Product Family D400 Series is classified as a Class 1 Laser Product under the EN/IEC 60825-1, Edition 3 (2014) internationally and IEC60825-1, Edition 2 (2007) in the US.

This product complies with US FDA performance standards under 21 CFR 1040.10 for laser products except for deviations pursuant to Laser Notice No. 50 dated June 24, 2007.

Caution - Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

Do not power on the product if any external damage was observed.

There are no service/maintenance, modification, or disassembly procedures for the stereo module and infrared projector. The system integrator must either notify Intel or return modules before any failure analysis is performed.

- Do not attempt to open any portion of this laser product.
- Invisible laser radiation when opened. Avoid direct exposure to beam.
- There are no user serviceable parts with this laser product.
- Modification or service of the stereo module, specifically the infrared projector, may cause the emissions to exceed Class 1.
- No magnifying optical elements, such as eye loupes and magnifiers, are allowed.

• Do not try to update camera firmware that is not officially released for specific camera module SKU and revision.

1.4 Hardware Requirements

Table 1-1. Supported Intel® RealSense™ Depth Modules and Depth Cameras

No.	Depth Module/Depth Cameras
1	Intel® RealSense™ Depth Module D400
2	Intel® RealSense™ Depth Module D410
3	Intel® RealSense™ Depth Module D415
4	Intel® RealSense™ Depth Camera D415
5	Intel® RealSense™ Depth Module D420
6	Intel® RealSense™ Depth Module D430
7	Intel® RealSense™ Depth Camera D435
8	Intel® RealSense™ Depth Camera D435i
9	Intel® RealSense™ Depth Camera D455

The host processor connection to the camera is through USB 3.1 Gen 1.

Figure 1-1. Intel® RealSense™ Depth Camera D435 (Depth Module Integrated)

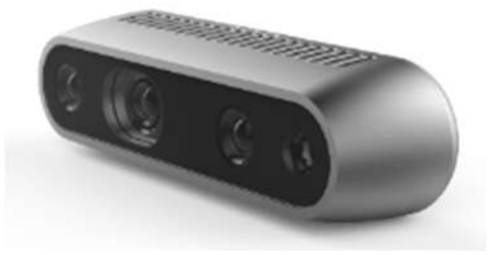
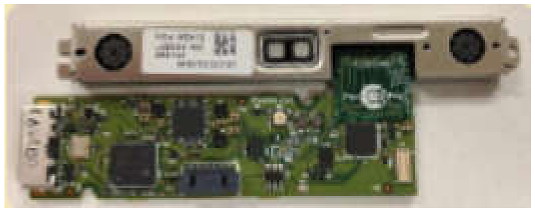


Figure 1-2. Intel® RealSense™ Depth Module D410 with Vision Processor D4 Card (Non-integrated Bare Depth Module)



1.5 Software Requirements

Table 1-2. Software Requirements

Software	Version
Intel® RealSense™ camera firmware	**On Linux/Windows platforms** 5.10.15.0 or higher on D435i 5.10.6.0 or higher on other devices
Linux*	64-bit host system supporting Ubuntu* version 16.04 and Ubuntu 18.04 Supports both Intel and ARM platforms. ARM platforms testing is limited to Nvidia Jetson TX2 and Xavier.
Linux* Kernel	4.4.0 and later
Windows*	Windows* 10 64-bit

§§

2 Overview

This chapter provides an overview of the calibration parameters and dynamic calibration process.

2.1 Calibration Parameters

Dynamic Calibration is optimizing ***extrinsic*** parameters, i.e., they refer to calibration done in the field at the user environment with minimal or no user intervention. They are ONLY extrinsic parameters (translation and rotation) of the camera image with regard to the main axis system (the axis between the left and right). Intrinsic parameters, such as distortion, field of view, principal point, are not dynamically calibrated.

Dynamic calibration is run under the assumption that it is the re-calibration of the depth modules/cameras after factory calibration, or at least that the nominal parameters are known.

The left camera is the reference camera and is located at world origin. RGB parameters only apply to depth modules/cameras with RGB sensor for color, e.g., Intel® RealSense™ Depth Camera D415, D435, and D455.

Intrinsic includes

- Focal length - specified as [fx; fy] in pixels for left, right, and RGB cameras
- Principal point - specified as [px; py] in pixels for left, right, and RGB cameras
- Distortion - specified as Brown's distortion model [k1; k2; p1; p2; k3] for left, right, and RGB cameras

Extrinsic includes

- RotationLeftRight - rotation from right camera coordinate system to left camera coordinate system, specified as a 3x3 rotation matrix
- TranslationLeftRight - translation from right camera coordinate system to left camera coordinate system, specified as a 3x1 vector in millimeters
- RotationLeftRGB - rotation from RGB camera coordinate system to left camera coordinate system, specified as a 3x3 rotation matrix
- TranslationLeftRGB - translation from RGB camera coordinate system to left camera coordinate system, specified as a 3x1 vector in millimeters

2.2 Types of Dynamic Calibration

Different types of dynamic calibrations are supported for Intel® RealSense™ Product Family D400 Series

1. **Rectification calibration:** aligning the epipolar line to enable the depth pipeline to work correctly and reduce the holes in the depth image
2. **Depth scale calibration:** aligning the depth frame due to changes in position of the optical elements

Dynamic Calibration Tool API supports these algorithms in two distinctive operating modes: targeted and target-less. The Dynamic Calibrator supports only targeted calibration.

Targeted calibration is the recommended approach because it supports both rectification and depth scale calibrations and will give more accurate results than rectification only calibration done in target-less calibration.

2.2.1 Target Dynamic Calibration (Depth Scale Calibration)

In targeted mode, Dynamic Calibration API supports depth scale calibration and a target is required. The target is predefined and can be displayed on a smartphone through a phone app. A simplified flow is summarized as following:

1. Take the images from L and R camera, including the depth stream (in real-time)
2. Detect the target on the smartphone in both images
3. Similar to target-less calibration, user moves the phone so that it covered most of the image, repeating steps 1-2
4. Once done, user just keeps taking images by positioning the phone anywhere in the image but must move the phone every time
5. After taking 15 images in step 4, the process is complete
6. The process checks for rectification error (absolute Y difference) but also compares the measured pattern size with ground truth

NOTE: The advantage of targeted calibration is that it's accurate and consistent. It calibrates left/right depth as well as RGB (on devices with RGB). The disadvantage is that it requires calibration target which means it cannot be used in cases where calibrating with a target is not feasible.

2.2.2 Target-less Dynamic Calibration (Rectification Calibration)

In target-less mode, Dynamic Calibration API supports rectification calibration without the need of any target. Its basic flow is summarized as following:

1. Take the images from L and R camera (in real-time)
2. Extract features from the images
3. Match the features between L and R camera
4. The image is binned into 6x8 grid. Check whether each bin contains enough corresponding points.
5. The user sees the panel status and moves the device around (bins without features are blue).
6. Steps 1-6 are iterated until all bins have enough feature points
7. Check rectification error (absolute Y difference). If too large, run a solver to optimize extrinsic parameters to minimize it.

NOTE: Note: Targeted calibration is likely to give more accurate results and is therefore the recommended approach.

NOTE: The advantage of target-less calibration is simplicity, no calibration target is needed. It fits with user cases where calibrating with a target may not be feasible. The disadvantage is that it calibrates left/right depth but not RGB and generally less accurate and less consistent than targeted calibration.

2.3 Depth Quality Tool

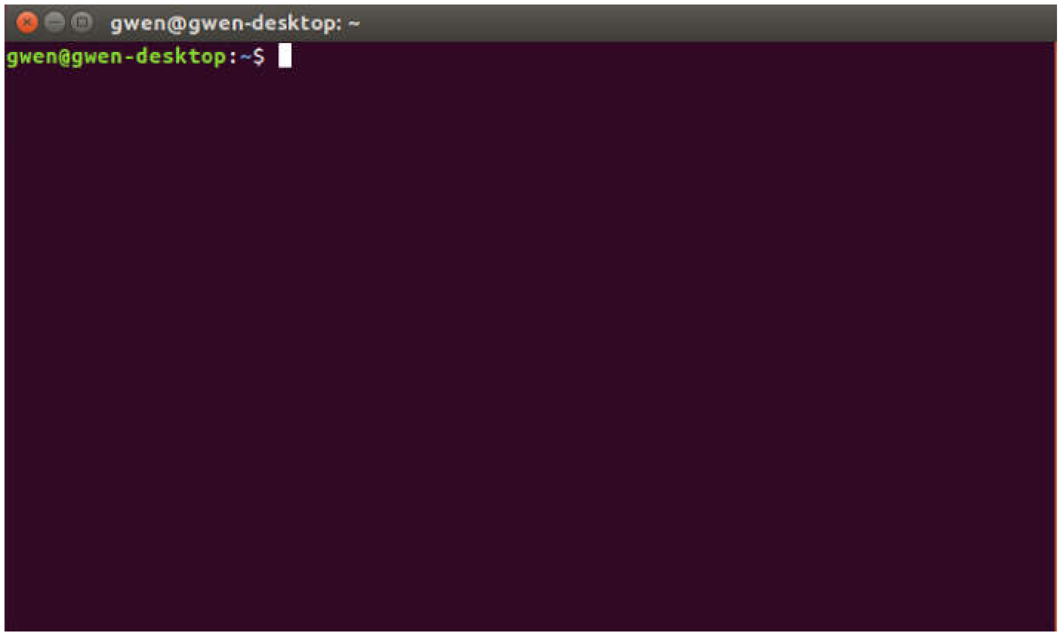
NOTE: The Intel® RealSense™ SDK includes a Depth Quality Tool that can be used to test the quality of the Intel® RealSense™ Product Family D400 Series cameras if it is suspected that the module has gone out of calibration. Refer to the SDK release page at https://github.com/IntelRealSense/librealsense/releases to download Depth Quality Tool

3 Installation

The Intel® RealSense™ Dynamic Calibrator is supported on Linux (Ubuntu 16.04 and Ubuntu 18.04), and Windows 10 64-bit.

3.1 Installation on Linux (Ubuntu 16.04 and Ubuntu 18.04)

To open a terminal window, hit (Ctrl + Alt + T).



```
gwen@gwen-desktop:~$ 
```

A working Internet connection is required for installing the 3rd party dependent libraries and patching the kernel. For systems behind a firewall or proxy server, proxy will also need to be configured. Please consult with your network administrator for details.

3.1.1 Install 3rd-party dependencies

1. Ensure apt-get is up to date:

```
sudo apt-get update
```

2. Install libusb-1.0:

```
sudo apt-get install libusb-dev libusb-1.0-0-dev
```

3. Install libglfw:

```
sudo apt-get install libglfw3 libglfw3-dev
```

4. Install freeglut:

```
sudo apt-get install freeglut3 freeglut3-dev
```

3.1.2 Calibration Tools and API Package Installation

Calibration Tool API provides installation package in dpkg format for Debian OS and its derivatives. The package and its respective content are listed below:

Table 3-1: Calibration Tools and API Packages

Name	Content
librscalibrationtool	Intel® RealSense™ Dynamic Calibrator, Calibration API and example sources and other related tools

The package is generally available for installation through Amazon AWS. If access to AWS is not permitted in specific use case, the Debian package can be obtained and installed locally.

3.1.2.1 Debian Package Installation through Amazon AWS

The Debian packages are available on Amazon AWS and can be installed with the following steps.

- Add Intel server to the list of repositories:

  On Ubuntu 16.04:

```
echo 'deb http://realsense-hw-public.s3.amazonaws.com/Debian/apt-repo xenial main' | sudo tee
/etc/apt/sources.list.d/realsense-public.list
```

  On Ubuntu 18.04:

```
echo 'deb http://realsense-hw-public.s3.amazonaws.com/Debian/apt-repo bionic main' | sudo tee
/etc/apt/sources.list.d/realsense-public.list
```

  It is recommended to backup /etc/apt/sources.list.d/realsense-public.list file in case of an upgrade.

- Register the server's public key:

```
sudo apt-key adv --keyserver keys.gnupg.net --recv-key 6F3EFCDE
```

- Refresh the list of repositories and packages available:

```
sudo apt-get update
```

- Install the librscalibrationtool package which includes **Intel® RealSense™** Dynamic Calibrator:

```
sudo apt-get install librscalibrationtool
```

3.1.2.2 Debian Package Installation through Local Files

The version 2.11.0.0 package is also available in local .deb Debian package file and can be installed with the following steps.

- Install the librscalibrationtool package which includes **Intel® RealSense™** Dynamic Calibrator:

  sudo dpkg -i librscalibrationtool_2.11.0.0_<platform>.deb, for example,

```
sudo dpkg -i librscalibrationtool_2.11.0.0_amd64.deb
```

3.1.3 Checking Package Installation

After the debian package install finishes, check for files installed. Under debian convention, the executables are installed under /usr/bin, library files under /usr/lib, and other files including sample code under /usr/share/doc/librscalibrationapi and /opt/intel/librscalibrationapi/.

For example, for Calibration Tool, librscalibrationtool, the files are installed as below:

```
sudo dpkg -L librscalibrationtool
```

```
/usr
  |.......bin
  |        |..... Intel.Realsense.DynamicCalibrator
  |        |.... Intel.Realsense.CustomRW
  |
  |----- share
  |          |---- doc
  |                  |..... librscalibrationtool
  |                           |---- attributions.txt
  |                           |---- readme.txt
  |                           |---- changelog
  |                           |---- copyright
  |                           |---- License.txt
  |                           |---- README.md
  |                           |
  |                           |---- target
  |                           |        |---- print-target-fixed-width.pdf
  |                           |
  |                           |---- api
  |                                    |---- DynamicCalibrationAPI-Linux-2.11.0.0.tar.gz
```

Notes:

- Intel.Realsense.DynamicCalibrator is the main calibrator app
- Intel.Realsense.CustomRW is the custom calibration data read/write tool
- DynamicCalibrationAPI-Linux-2.11.0.0.tar.gz is the Calibration API and examples package, advanced users can develop custom calibration apps with this API, see chapter ***Developing Custom Apps with Calibration API*** and **Dynamic Calibration Programmer's Guide** for details.

3.1.4 Video4Linux backend

1. Ensure no cameras are presently plugged into the system.
2. Patch and insert modified kernel drivers for Intel® RealSense™ Camera.

   Intel® RealSense™ SDK 2.0 provides dpkg package librealsense2-dkms for the kernel rules and kernel drivers. Please follow instructions in the following link for Ubuntu OS version supported and package installation details:

   https://github.com/IntelRealSense/librealsense/blob/development/doc/distribution_linux.md

   Please note generally on supported platforms, only the Realsense DKMS kernel driver package librealsense2-dkms is required for Dynamic Calibration. The rules are included in librealsense2-udev-rules which librealsense2-dkms depends and will be installed automatically.

```
sudo apt-get install librealsense2-dkms
```

3.1.5 Test Installation

Now check dynamic calibration tool version by running the following command:

The calibration tool executables are located under /usr/bin.

For example,

To print the tool version to screen:

```
/usr/bin/Intel.Realsense.DynamicCalibrator -v
```

To print device information to screen, connect a camera device to the system and execute the following command:

```
/usr/bin/Intel.Realsense.DynamicCalibrator -list
```

To print device calibration data to screen, execute the following command:

```
/usr/bin/Intel.Realsense.CustomRW -r
```

To try out calibrator, execute the following command:

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

If you see the correct version information, device information and calibration data printed on screen, Dynamic Calibrator launched with correct device listed, congratulations you have the right setup and ready to use the calibration tools.

3.1.6 Debian Package Removal

For any reason, if you would like to uninstall the Calibration Tools and API, the Debian packages can be removed by running dpkg commands.

For example, Calibration Tool, librscalibrationtool, can be removed as below:

```
sudo dpkg -r librscalibrationtool
```

3.2 Installation on Windows

The Intel® RealSense™ Dynamic Calibration Tool runs on Windows 10.

3.2.1 Install 3rd-party dependencies

Windows dynamic calibration package requires the following installed on the host system:
Microsoft Visual C++ 2015 Redistributable.

The install package installs this library automatically. So normally user will not need to install this separately.

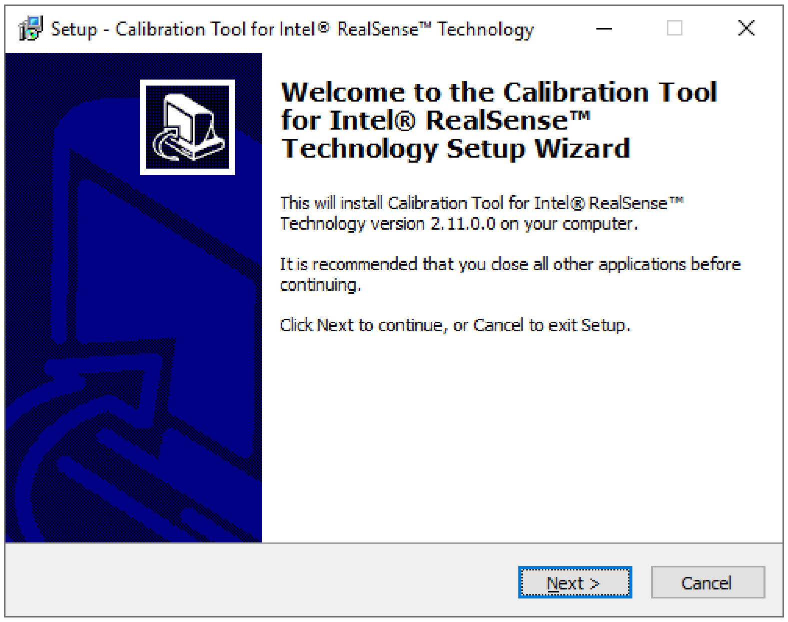
3.2.2 Package installation

The release includes the following install package for Windows:

- Calibration Tool and API
  CalibrationTool-2.11.0.0-Setup.exe

3.2.2.1 Calibration Tool Windows Installer Package

The installer package compresses all contents into a single installer file, for example, CalibrationTool-2.11.0.0-Setup.exe. Simply click it to install on your system. By default, it installs under C:\Program Files\Intel\CalibrationToolAPI\2.11.0.0.



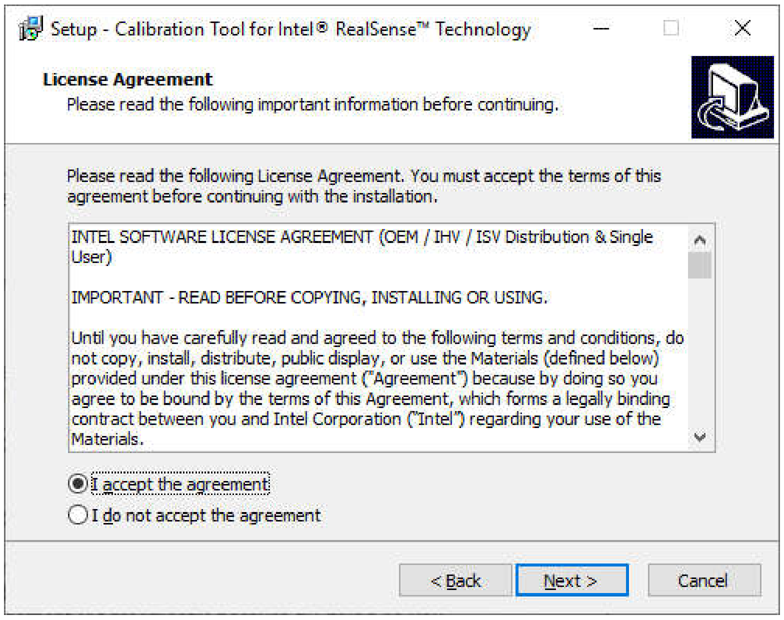
Setup - Calibration Tool for Intel® RealSense™ Technology

Welcome to the Calibration Tool for Intel® RealSense™ Technology Setup Wizard

This will install Calibration Tool for Intel® RealSense™ Technology version 2.11.0.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



Setup - Calibration Tool for Intel® RealSense™ Technology

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

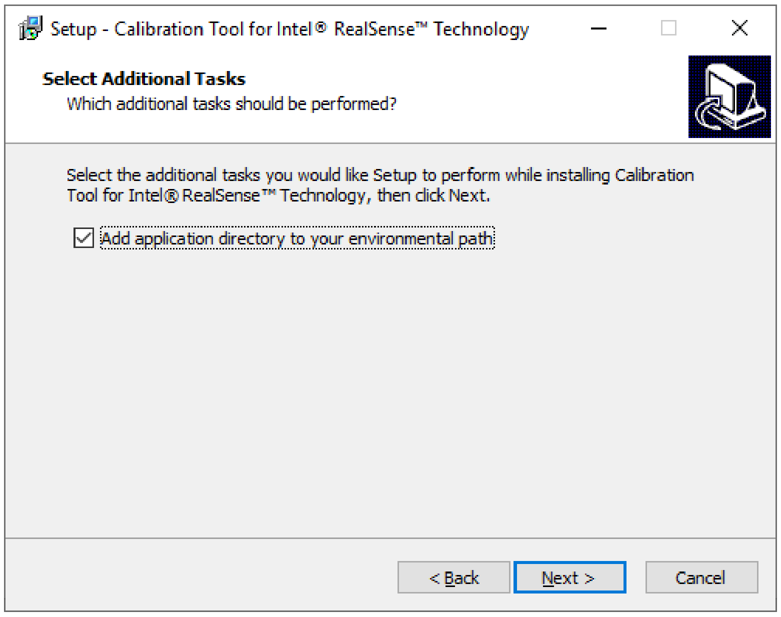
INTEL SOFTWARE LICENSE AGREEMENT (OEM / IHV / ISV Distribution & Single User)

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

Until you have carefully read and agreed to the following terms and conditions, do not copy, install, distribute, public display, or use the Materials (defined below) provided under this license agreement ("Agreement") because by doing so you agree to be bound by the terms of this Agreement, which forms a legally binding contract between you and Intel Corporation ("Intel") regarding your use of the Materials.

● I accept the agreement

○ I do not accept the agreement



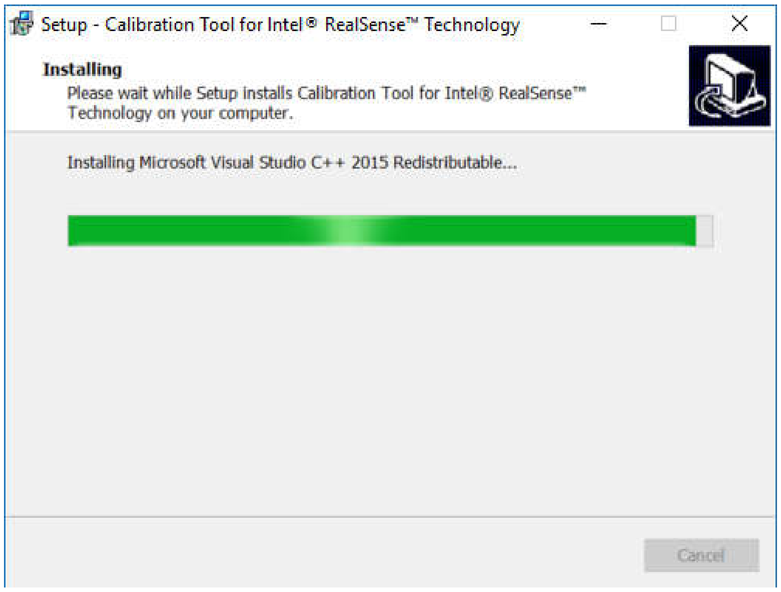
Setup - Calibration Tool for Intel® RealSense™ Technology

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Calibration Tool for Intel® RealSense™ Technology, then click Next.

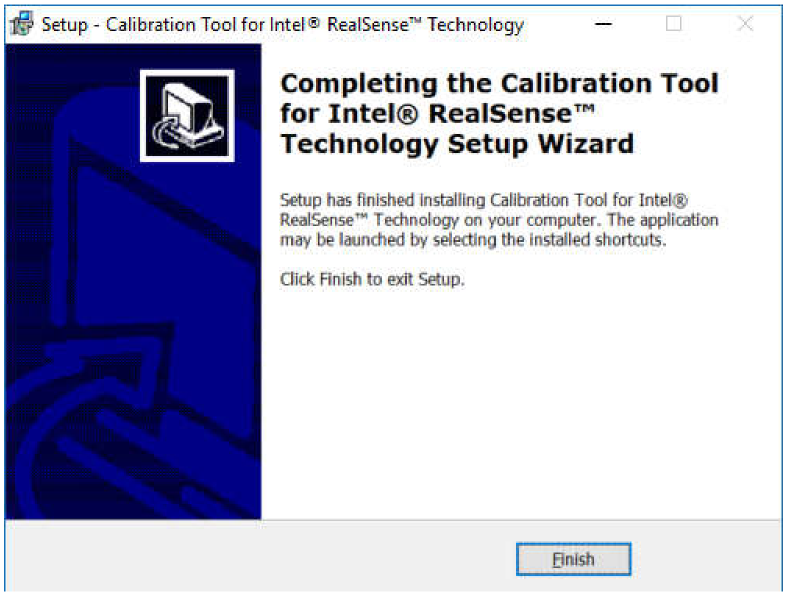
☑ Add application directory to your environmental path


Setup - Calibration Tool for Intel® RealSense™ Technology

Installing

Please wait while Setup installs Calibration Tool for Intel® RealSense™ Technology on your computer.

Installing Microsoft Visual Studio C++ 2015 Redistributable...


Setup - Calibration Tool for Intel® RealSense™ Technology

Completing the Calibration Tool for Intel® RealSense™ Technology Setup Wizard

Setup has finished installing Calibration Tool for Intel® RealSense™ Technology on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

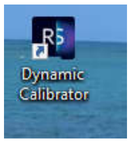
The tools are installed under C:\Program Files\Intel with following directory structure:

```
 CalibrationToolAPI
        |-------- 2.11.0.0
        |            |-------- bin
        |            |            |-------- Intel.Realsense.DynamicCalibrator.exe
        |            |            |-------- Intel.Realsense.DynamicCalibratorCLI.exe
        |            |            |-------- Intel.Realsense.CustomRW.exe
        |            |
        |            |-------- api
        |            |          |-------- DynamicCalibrationAPI.Zip
        |            |
        |            |-------- attributions.txt
        |            |-------- license.txt
        |            |-------- readme.txt
        |            |-------- target
```

The installed directory contains two versions of the Dynamic Calibrator tool:

- **Intel.Realsense.DynamicCalibrator.exe** with an interactive graphical user interface. It supports targeted calibration and is the primary tool regular users should use to calibrate the devices. This is the tool linked to the Dynamic Calibrator icon on the desktop.
- **Intel.Realsense.DynamicCalibratorCLI.exe** with a command line interface with advanced command line options. It supports the targeted calibration process similarly to the GUI version, in addition, it also supports many other features that advanced users may prefer to use to calibrate their devices or experiment new calibration methods.
- **Intel.Realsense.CustomRW.exe** allows user to dump the calibration data from device, write calibration parameters to device, and reset the device calibration to default gold settings.
- **DynamicCalibrationAPI.Zip** is the Calibration API and examples package, advanced users can develop custom calibration apps with this API, see chapter ***Developing Custom Apps with Calibration API*** and **Dynamic Calibration Programmer's Guide** for details.

The installer will create Dynamic Calibrator icon link to Intel.Realsense.DynamicCalibrator.exe at the system desktop.



Double click the icon to start using the tool with its graphical user interface.

Of course, if command line interface preferred, users can execute the programs anywhere since installation location is added to system PATH during installation. User can also always navigate to the install directory using a windows terminal.

For example,

```
cd C:\Program Files\Intel\CalibrationToolAPI\2.11.0.0
cd bin
```

4 Calibration with Dynamic Calibrator

Your camera might be out of calibration if you see the following symptoms:

- Reduced depth density on objects in the operating range (might still get depth on vertical lines).
- Flat surfaces look "wobbly", i.e. there is more deviation from the flatness than usual.
- Measuring the physical distances to objects are not within about 3% of what they should be.

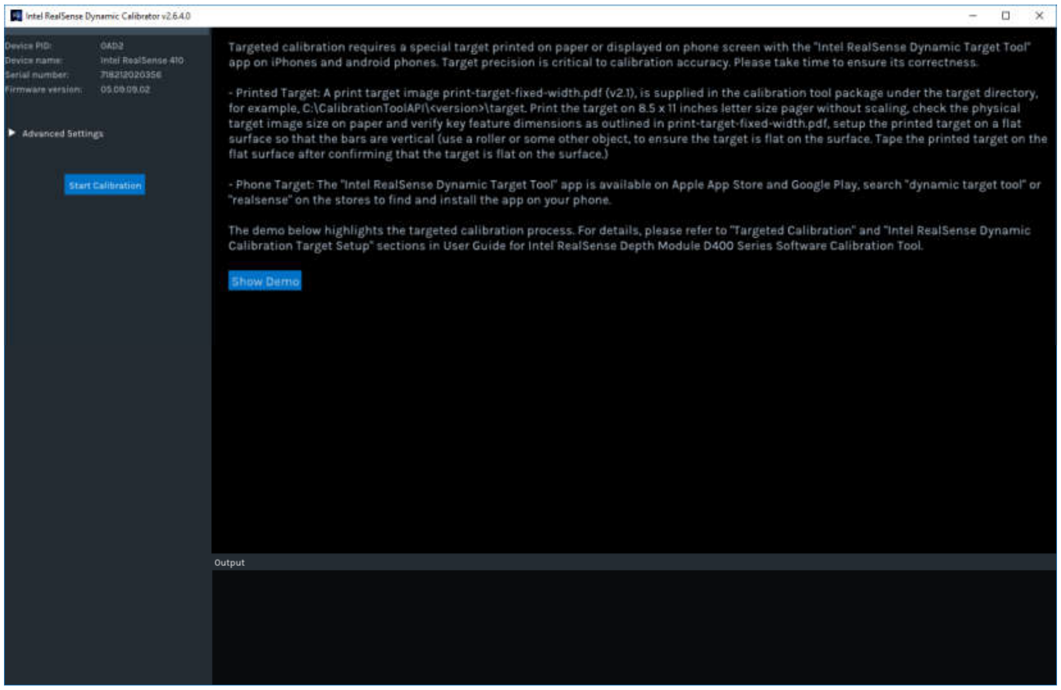
Once you determined the camera is out of calibration, the camera needs to be re-calibrated to correct the changes that might have occurred between the left and right depth sensors.

Dynamic Calibrator is under the bin directory. To support both regular users who prefer graphical user interface (GUI) and OEM/ODMs who prefer command line interface (CLI) for ease of automation, the tool supports both GUI and CLI interfaces.

- On **Linux**, a single executable Intel.Realsense.DynamicCalibrator supports both GUI and CLI interfaces.
- On **Windows**, two separate executables, both share same functionality but different user interfaces, are provided:
    - ✓ Intel.Realsense.DynamicCalibrator.exe with a graphical user interface
    - ✓ Intel.Realsense.DynamicCalibratorCLI.exe with a command line interface with command line options

4.1 Dynamic Calibrator Graphical User Interface

The Graphical User Interface (GUI) is provided for easier use. The app detects all Intel® RealSense™ Product Family D400 Series cameras on the system and lists them in a dropdown box on the top left corner.


Intel RealSense Dynamic Calibrator v2.6.4.0

Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

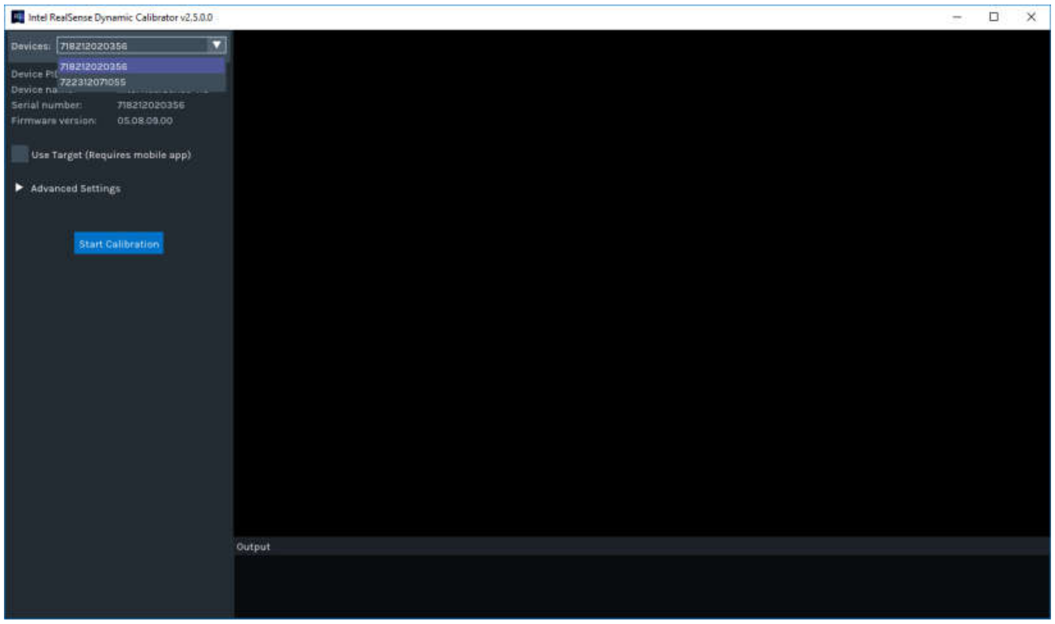
Targeted calibration requires a special target printed on paper or displayed on phone screen with the "Intel RealSense Dynamic Target Tool" app on iPhones and android phones. Target precision is critical to calibration accuracy. Please take time to ensure its correctness.

- Printed Target: A print target image print-target-fixed-width.pdf (v2.1), is supplied in the calibration tool package under the target directory, for example, C:\CalibrationToolAPI\<version>\target. Print the target on 8.5 x 11 inches letter size pager without scaling, check the physical target image size on paper and verify key feature dimensions as outlined in print-target-fixed-width.pdf, setup the printed target on a flat surface so that the bars are vertical (use a roller or some other object, to ensure the target is flat on the surface. Tape the printed target on the flat surface after confirming that the target is flat on the surface.)

- Phone Target: The "Intel RealSense Dynamic Target Tool" app is available on Apple App Store and Google Play, search "dynamic target tool" or "realsense" on the stores to find and install the app on your phone.

The demo below highlights the targeted calibration process. For details, please refer to "Targeted Calibration" and "Intel RealSense Dynamic Calibration Target Setup" sections in User Guide for Intel RealSense Depth Module D400 Series Software Calibration Tool.

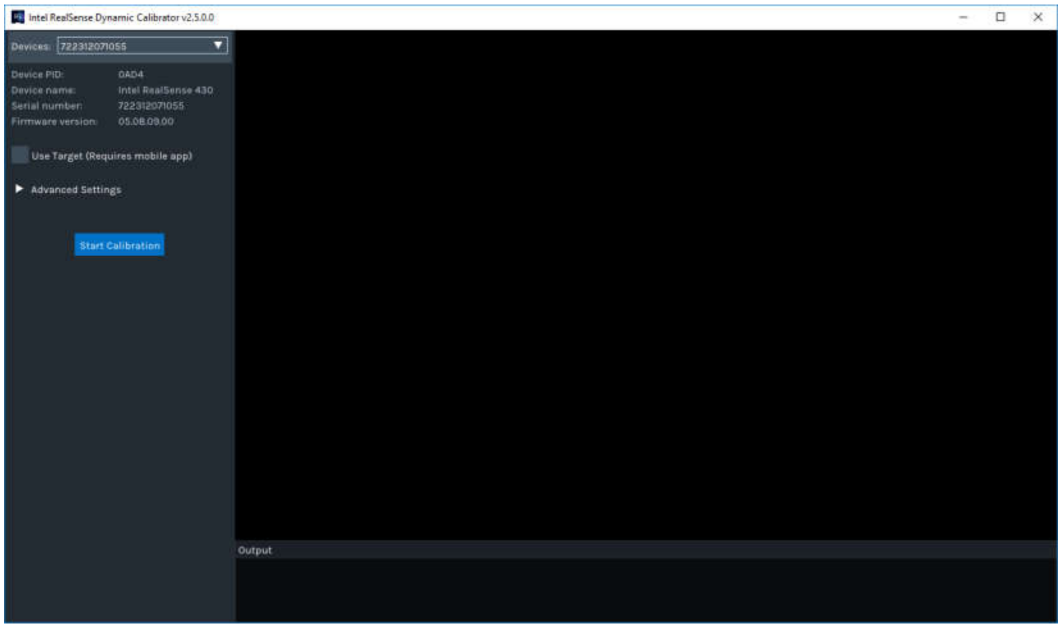
User selects one of the devices from the list to calibrate:


Intel RealSense Dynamic Calibrator v2.5.0.0

Devices: 718212020356
718212020356
722312071055

Device PI
Device na
Serial number: 718212020356
Firmware version: 05.08.09.00

Details on the device including product id, device name, serial number, and FW version displayed:


Intel RealSense Dynamic Calibrator v2.5.0.0
Use Target (Requires mobile app)
Advanced Settings
Start Calibration
Output

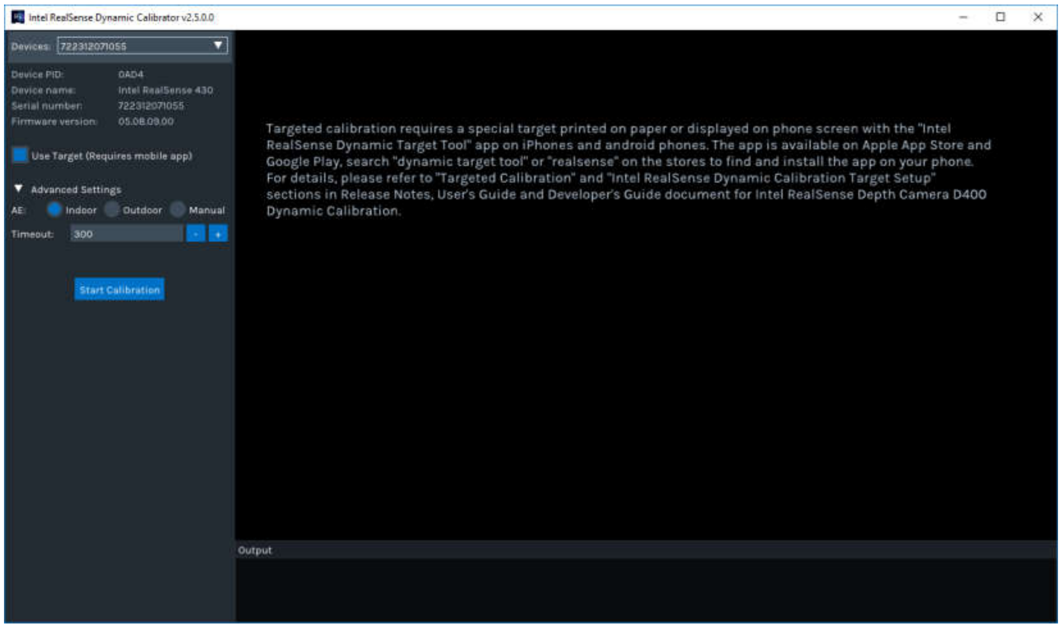
Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

The calibration process requires a special target, either printed or phone target. Please refer to Appendix for target setup details.

By default, the app handles everything automatically. However, a few advanced settings are provided for experienced users who prefer to make custom choices.

For targeted calibration, user can customize AE (auto exposure) setpoint. **Indoor** and **Outdoor** are auto options that let the app to handle it automatically. The app will adjust automatically to the environment light conditions so that calibration target can be found without over exposure. The Manual option allow user to set a fixed setpoint value. This is useful in very complex environment where the app may experience difficulty to sweep through the setpoints automatically. The value set in such cases is completely dependent on the environment lighting, for example, 1000 - 2000 should be sufficient for indoor usage and 3000 – 4000 might work better outdoor. The exact value in user case can be found through trial and error.


Intel RealSense Dynamic Calibrator v2.5.0.0

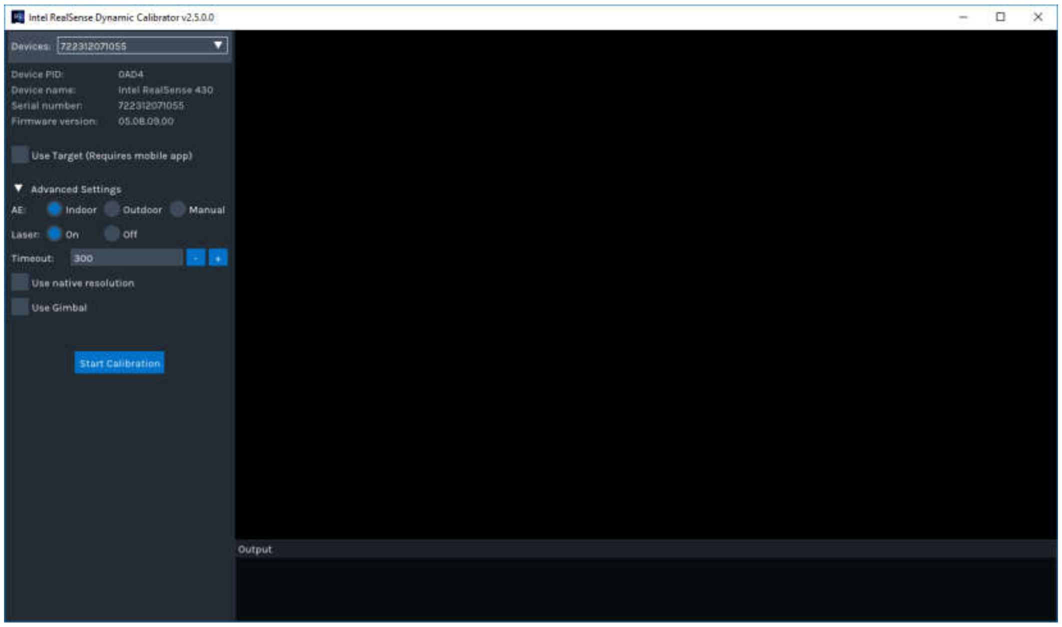
Devices: 722312071055

Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

Use Target (Requires mobile app)

Advanced Settings
AE: Indoor Outdoor Manual
Timeout: 300

Targeted calibration requires a special target printed on paper or displayed on phone screen with the "Intel RealSense Dynamic Target Tool" app on iPhones and android phones. The app is available on Apple App Store and Google Play, search "dynamic target tool" or "realsense" on the stores to find and install the app on your phone. For details, please refer to "Targeted Calibration" and "Intel RealSense Dynamic Calibration Target Setup" sections in Release Notes, User's Guide and Developer's Guide document for Intel RealSense Depth Camera D400 Dynamic Calibration.


Intel RealSense Dynamic Calibrator v2.5.0.0

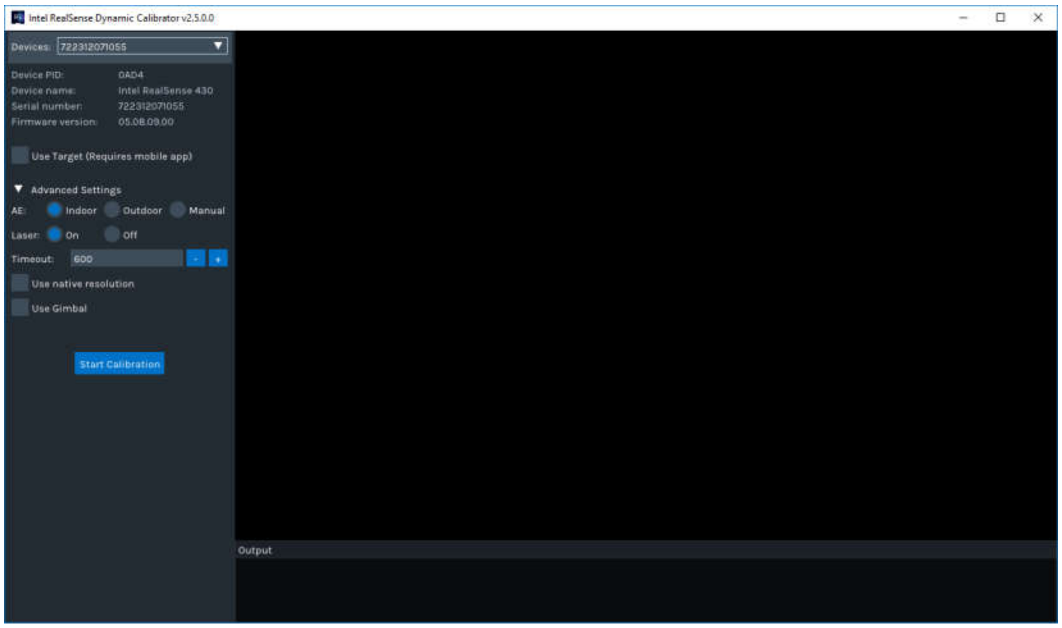
Devices: 722312071055

Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

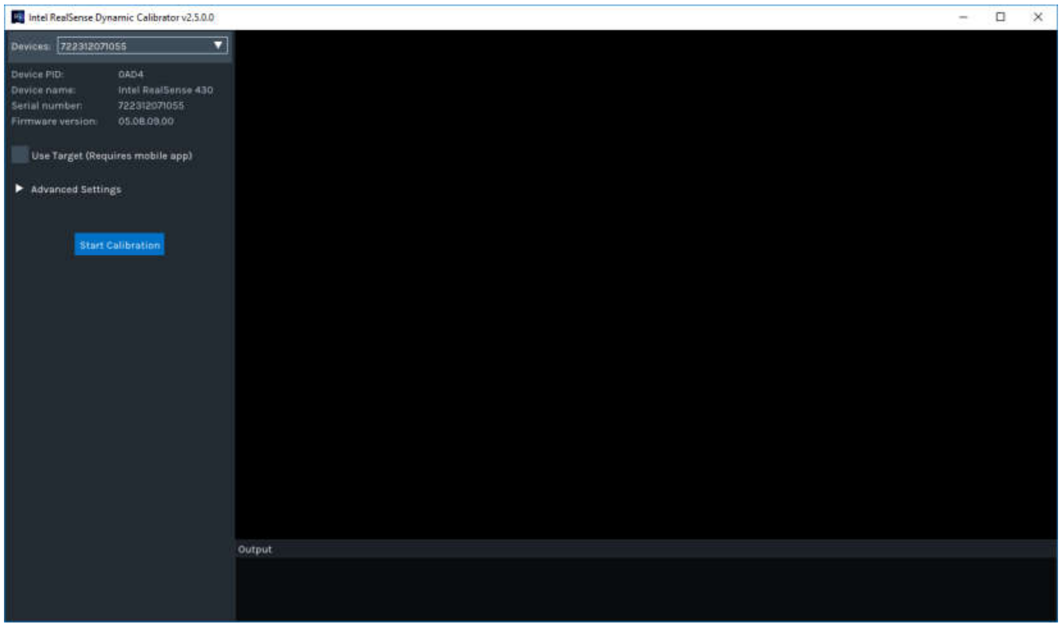
Use Target (Requires mobile app)

Advanced Settings
AE: Indoor Outdoor Manual
Laser: On Off
Timeout: 300
Use native resolution
Use Gimbal

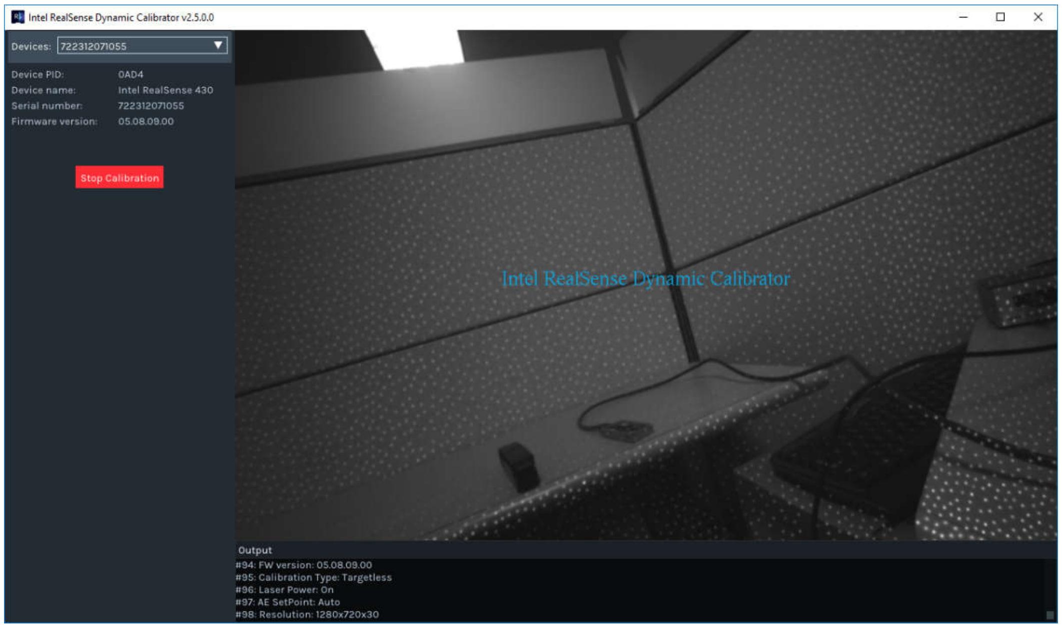
User can also adjust the timeout period. User can adjust it in case an extended period is required in difficult situations:


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00
Use Target (Requires mobile app)
Advanced Settings
AE: Indoor Outdoor Manual
Laser: On Off
Timeout: 600
Use native resolution
Use Gimbal
Start Calibration
Output

The last step is to start calibration by clicking the “Start Calibration” button:


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00
Use Target (Requires mobile app)
Advanced Settings
Start Calibration
Output

User can always stop/cancel calibration and return to main interface at any time by clicking the “Stop Calibration” button”:

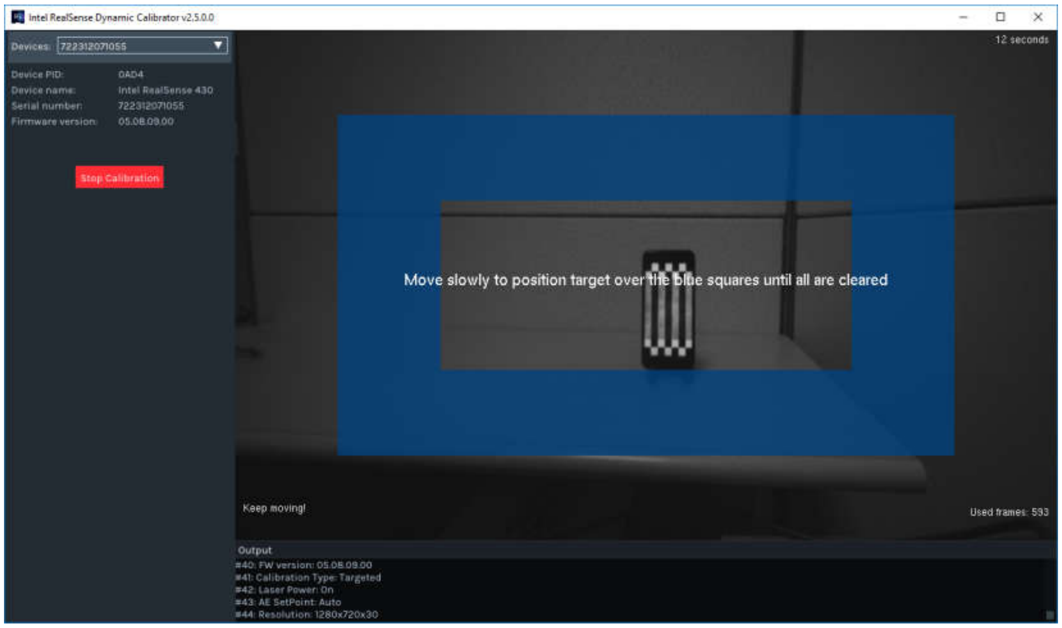

Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Stop Calibration
Intel RealSense Dynamic Calibrator
Output

Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

```
#94: FW version: 05.08.09.00
#95: Calibration Type: Targetless
#96: Laser Power: On
#97: AE SetPoint: Auto
#98: Resolution: 1280x720x30
```

During calibration, the UI displays the elapsed time in the top right corner, and status feedback on the lower left corner, and either the number of frames or target distance on the lower right corner:

Rectification Phase:



Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

12 seconds

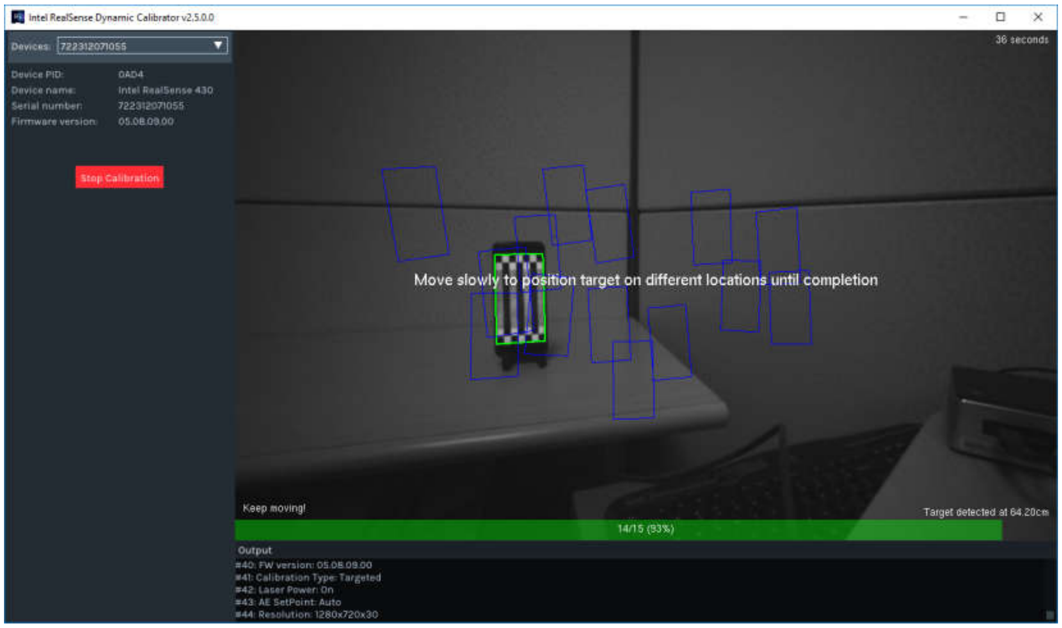
Move slowly to position target over the blue squares until all are cleared

Keep moving!

Used frames: 593

```
#40: FW version: 05.08.09.00
#41: Calibration Type: Targeted
#42: Laser Power: On
#43: AE SetPoint: Auto
#44: Resolution: 1280x720x30
```

Scale Phase:



Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

36 seconds

Move slowly to position target on different locations until completion

Keep moving!

Target detected at 64.20cm

14/15 (93%)

```
#40: FW version: 05.08.09.00
#41: Calibration Type: Targeted
#42: Laser Power: On
#43: AE SetPoint: Auto
#44: Resolution: 1280x720x30
```

4.2 Dynamic Calibrator Command Line Interface

A number of command line options are supported in Dynamic Calibrator Command Line Interface (CLI):

- **Intel.Realsense.DynamicCalibratorCLI.exe** on Windows
- **Intel.Realsense.DynamicCalibrator -cli** on Linux

Table 4-1: Dynamic Calibrator Command Line Options

Option	Description
-help or -?	Display list of command line options.
-version -v	Show Calibration Tool API version info
-cli	Run app in CLI mode
-show or -list	Display list of connected cameras with device name, serial number and firmware version information.
-sn <camera serial number> -serial < camera serial number>	If multiple cameras connected to the current system, choose one of the camera devices to calibrate by specifying its serial number.
-mode <0, 1, 2, 3, or 4> -m <0, 1, 2, 3, or 4>	select calibration mode: 0 for target-less calibration 1 for targeted calibration 2 for hybrid calibration 3 for scale calibration only (for debugging) 4 for rgb calibration only (for debugging)
-dump <0 or 1> -d <0 or 1>	For debugging use. Dump calibration images and other data. 0 to dump only frames used in calibration and results to files. 1 to dump every frame received.
-setpoint <0 to 4095> -s <0 to 4095>	Force AE setpoint (reference point). For dark environment, choose a low setpoint, 400, for example. For normal office lighting, a setpoint around 800 is appropriate. For outdoor bright sunlight environment, a higher setpoint is required, for example, 1200 or higher.
-timeout -t <Time out duration in seconds>	Customize time out duration. By default, calibration session will time out in 180 seconds for target-less and 300 seconds for targeted calibration.
-error -e	Report rectification (vertical alignment) error in pixels periodically until timeout
-a -aligned	device and target aligned in orientation, for example, device is mounted vertically, and target is positioned vertically
-skiprgb	Skip RGB calibration. This is useful if device like D435 and D415 equiped with a RGB camera but user does not use it. Skip RGB phase during calibration process will not impact depth calibration and it makes the process much faster.

-ignore-borders	ignore the border blocks during target-less rectification
-max-images <more than 6>	Specify number of images to capture in scale calibration and rgb calibration phases. This is used when user usage limited by time or space and would like to experiment best settings for their usage. • In targeted calibration, the default is 16 images. • In hybrid calibration, the default is either 6 or 8 images depends on device/target orientation. • For optimal results, recommend at least 6 images.
-force	Force the app to accept specified parameters, even if it's outside of the recommended range. Currently this only applies to -max-images.
-verbose	Display more detailed messages to assist support and debug

4.3 Connect camera to host system

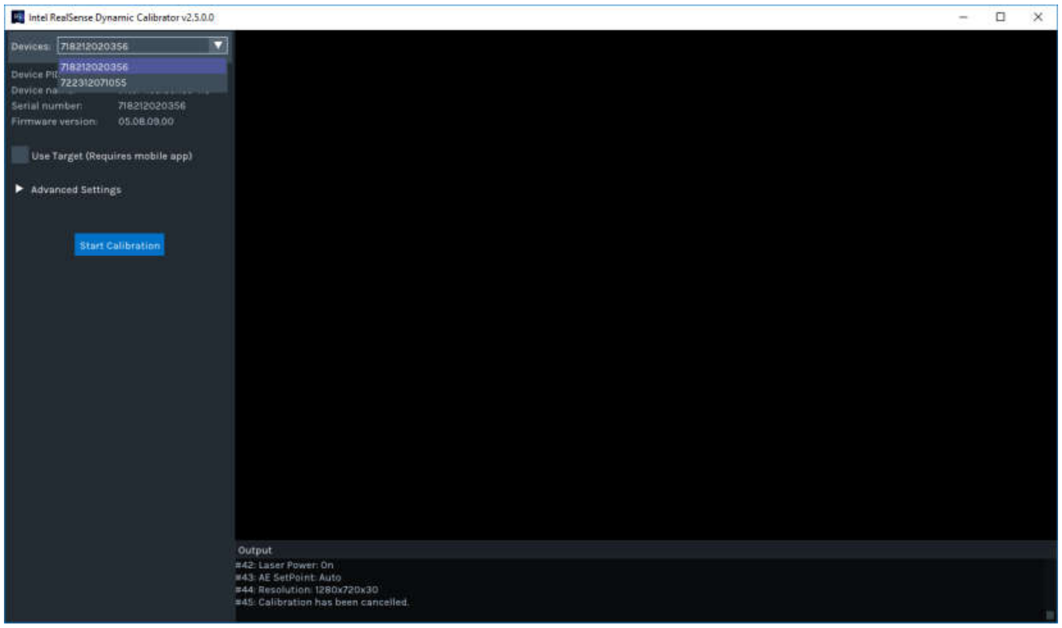
The Intel® RealSense™ camera to be calibrated should be connected to the system where dynamic calibrator software will run. This is usually done through a USB cable (for peripherals). On systems with camera integrated, it's already connected, so no extra step is required.

4.4 Multiple camera support

If multiple cameras connected to the current system, dynamic calibrator allows user to list all camera devices on the system and choose one of the camera device to calibrate. Only a single camera device can be calibrated at a time.

4.4.1 List all camera devices with Graphical User Interface

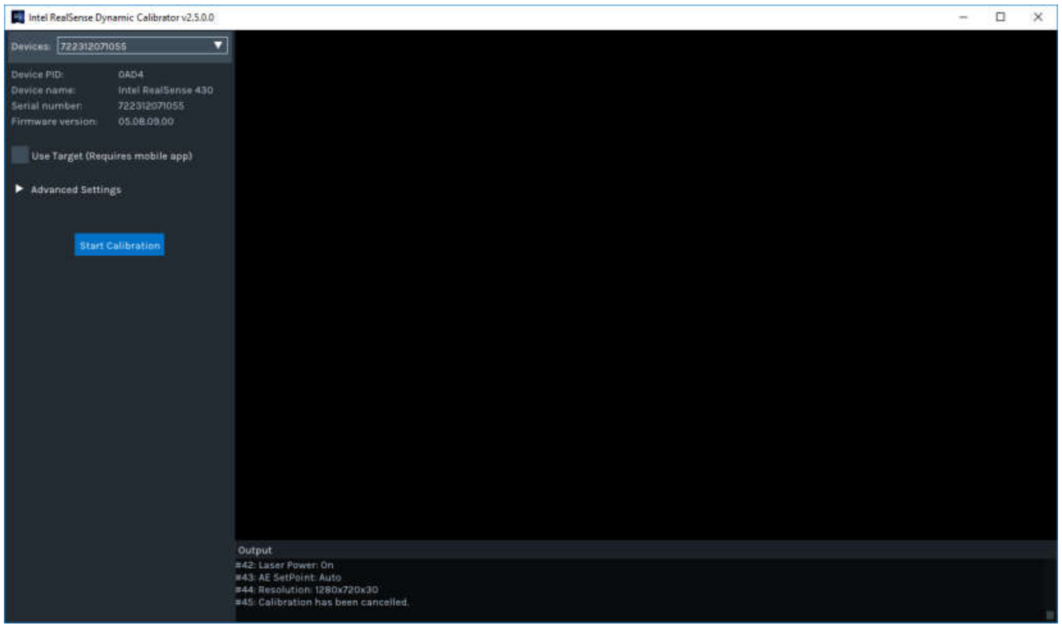
The app detects all Intel® RealSense™ Product Family D400 Series devices on the system and lists them in a dropdown box on the top left corner.



```
Device PI
Device na
Serial number:      718212020356
Firmware version:   05.08.09.00
```

```
Output
#42: Laser Power: On
#43: AE SetPoint: Auto
#44: Resolution: 1280x720x30
#45: Calibration has been cancelled.
```

User selects one of the devices from the list to calibrate and details on the device including product ID, device name, serial number, and Firmware version is displayed:


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Use Target (Requires mobile app)
Advanced Settings
Start Calibration

```
Device PID:         0AD4
Device name:        Intel RealSense 430
Serial number:      722312071055
Firmware version:   05.08.09.00
```

```
Output
#42: Laser Power: On
#43: AE SetPoint: Auto
#44: Resolution: 1280x720x30
#45: Calibration has been cancelled.
```

4.4.2 List all camera devices with Command Line Interface (CLI)

To list all devices, use the –show or –list option in DynamicCalibrator. For example:

On **Linux**:

```
/usr/bin/Intel.Realsense.DynamicCalibrator –list

Device Name              Serial Number     Firmware Version
Intel RealSense 410        718212020356       05.10.06.00
Intel RealSense 430        722312071055       05.10.06.00
```

On **Windows**:

```
cd bin
Intel.Realsense.DynamicCalibrator.exe –list

Device Name              Serial Number     Firmware Version
Intel RealSense 410        718212020356       05.10.06.00
Intel RealSense 430        722312071055       05.10.06.00
```

4.4.3 Perform calibration on selected device with Command Line Interface (CLI)

Then choose one of the camera devices to calibrate by specifying its serial number though the –sn option in DynamicCalibrator app. For example,

On Linux:

```
/usr/bin/Intel.Realsense.DynamicCalibrator –sn 722312071055
```

On Windows:

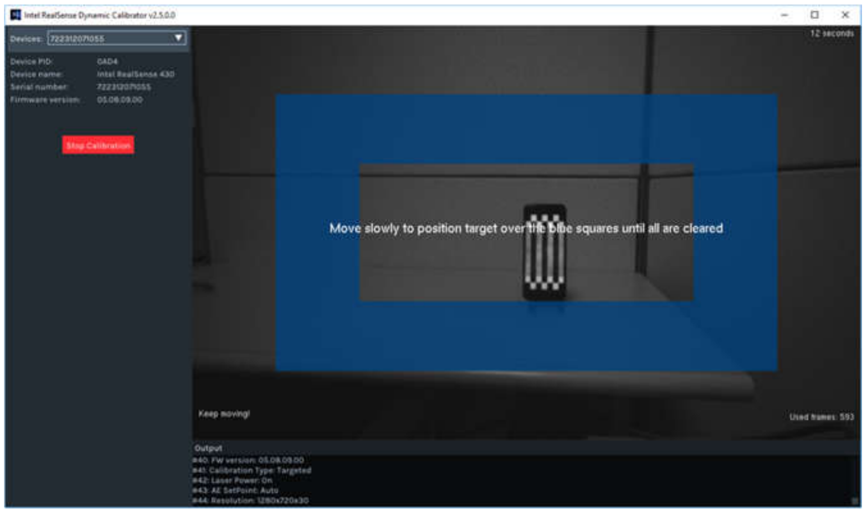
```
Intel.Realsense.DynamicCalibratorCLI.exe –sn 722312071055
```

4.4.4 Stop Calibration

After calibration is started, user can stop the process at any time by doing any **one** of the following:

a) ESC key – press **ESC** key anytime in both GUI and CLI mode will stop the calibration process.

b) Or, in GUI mode, press the "**Stop Calibration**" button at top left corner of the calibration window.


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Stop Calibration
12 seconds
Move slowly to position target over the blue squares until all are cleared
Keep moving!
Used frames: 593
Output

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

```
#40: FW version: 05.08.09.00
#41: Calibration Type: Targeted
#42: Laser Power: On
#43: AE SetPoint: Auto
#44: Resolution: 1280x720x30
```

c) Or, in CLI mode, press the “Q” or “q” key will quit the process

d) Or, simply close the calibration window

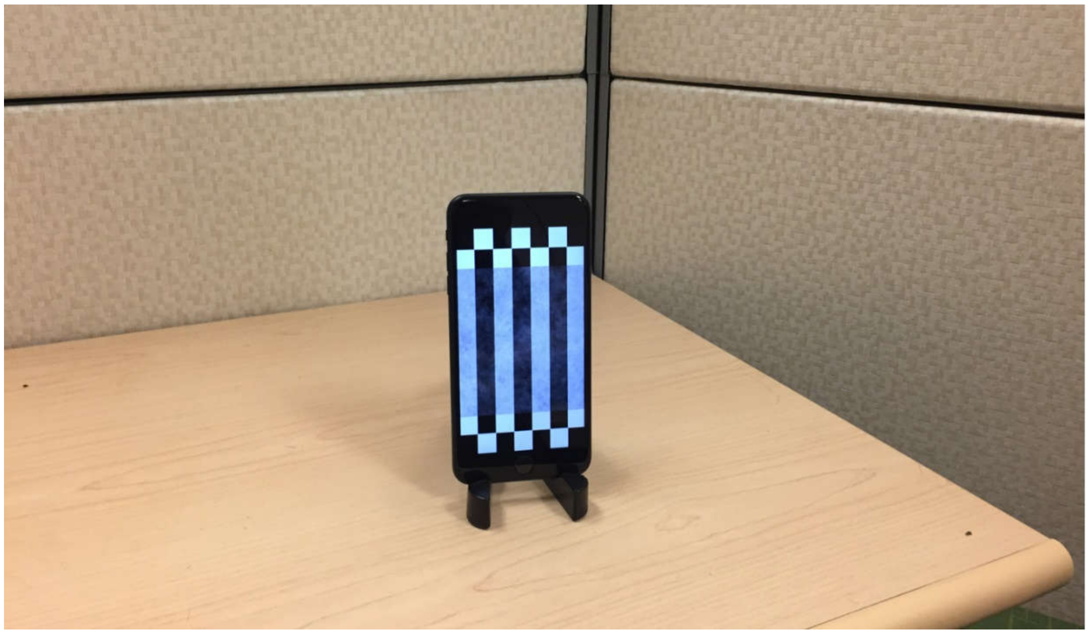
4.5 Targeted Calibration

4.5.1 Setup Calibration Target

Calibration Tools API supports printed target on letter sized paper, target displayed on iPhones and android phones. Please refer to details in Appendix *INTEL® REALSENSE™ DYNAMIC CALIBRATION TARGET SETUP*.

For example, a target displayed on iPhone 7 plus via Intel® RealSense™ Dynamic Target Tool app:

Figure 4-1. Calibration Target Displayed on Phone



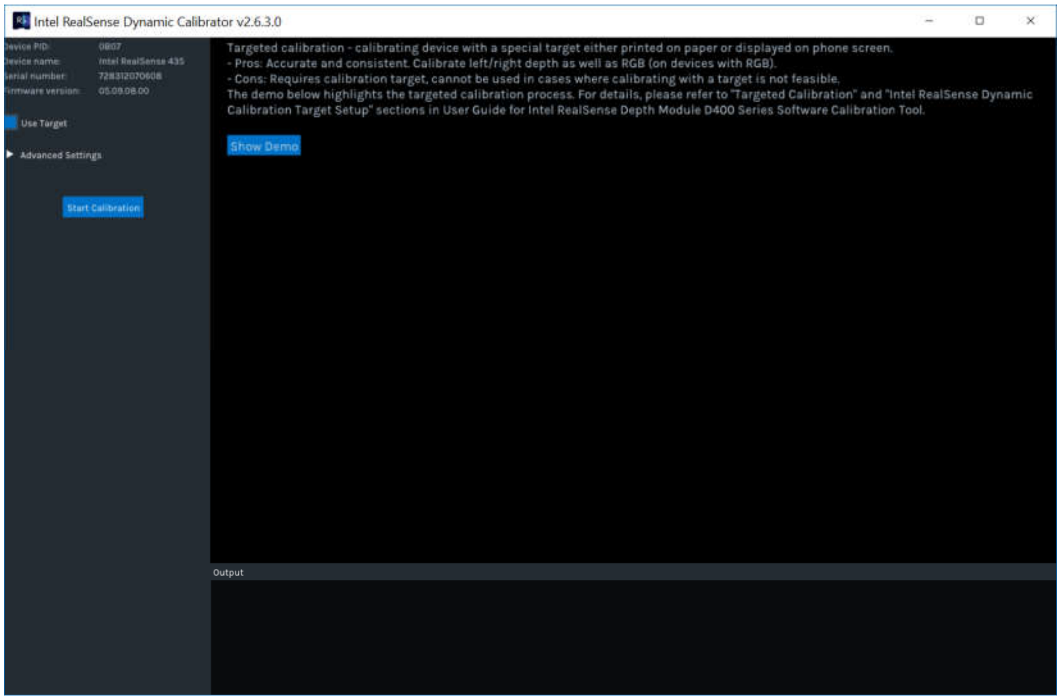
The camera should be pointing to the target at a distance of approximately 65 cm to 100 cm. It should not be too close or too far. Due to various phone sizes, camera device models, and lighting conditions, exact distance cannot be specified. User will need to move the camera in the specified range to find a distance that works best. In most cases, a distance of around 70 cm is sufficient.

4.5.2 Running Dynamic Calibrator with targeted calibration

On **Linux**:

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

Be default, the application displays Graphical User Interface (GUI). It detects all Intel® RealSense™ Product Family D400 Series devices on the system and lists them in a dropdown box on the top left corner. Selects one of the device from the list to calibrate. Details of the device is displayed as well:



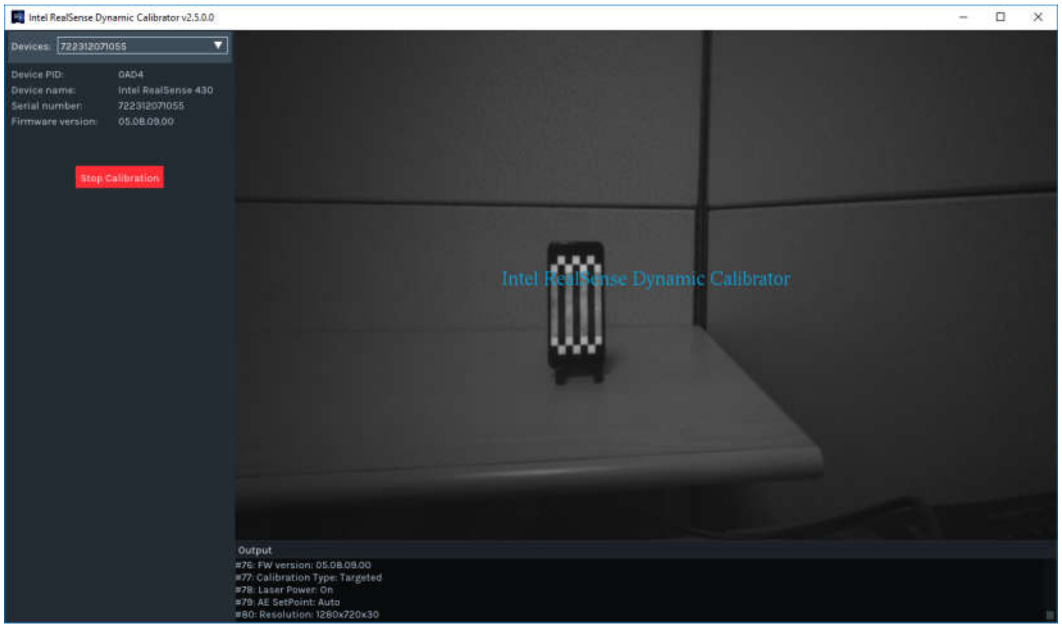
Device PID: 0B07
Device name: Intel RealSense 435
Serial number: 728312070608
Firmware version: 05.09.08.00

Targeted calibration - calibrating device with a special target either printed on paper or displayed on phone screen.
- Pros: Accurate and consistent. Calibrate left/right depth as well as RGB (on devices with RGB).
- Cons: Requires calibration target, cannot be used in cases where calibrating with a target is not feasible.
The demo below highlights the targeted calibration process. For details, please refer to "Targeted Calibration" and "Intel RealSense Dynamic Calibration Target Setup" sections in User Guide for Intel RealSense Depth Module D400 Series Software Calibration Tool.

By default, “Use Target” is selected, so the tool will run in targeted calibration mode. This requires a special target setup described in previous section “Setup Calibration Target”. If you haven’t done that, please follow instructions and complete the target set up either on a printed target or on a phone target.

The “Show Demo” button will highlight the key steps of the calibration process. This is also detailed in Appendix “*Calibration process Highlights*”.

Next, start calibration by clicking the “Start Calibration” button:


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Stop Calibration
Intel RealSense Dynamic Calibrator

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

```
Output
#76: FW version: 05.08.09.00
#77: Calibration Type: Targeted
#78: Laser Power: On
#79: AE SetPoint: Auto
#80: Resolution: 1280x720x30
```

If command line interface is preferred, user can display DynamicCalibrator command line options by running:

```
/usr/bin/ Intel.Realsense.DynamicCalibrator -help
```

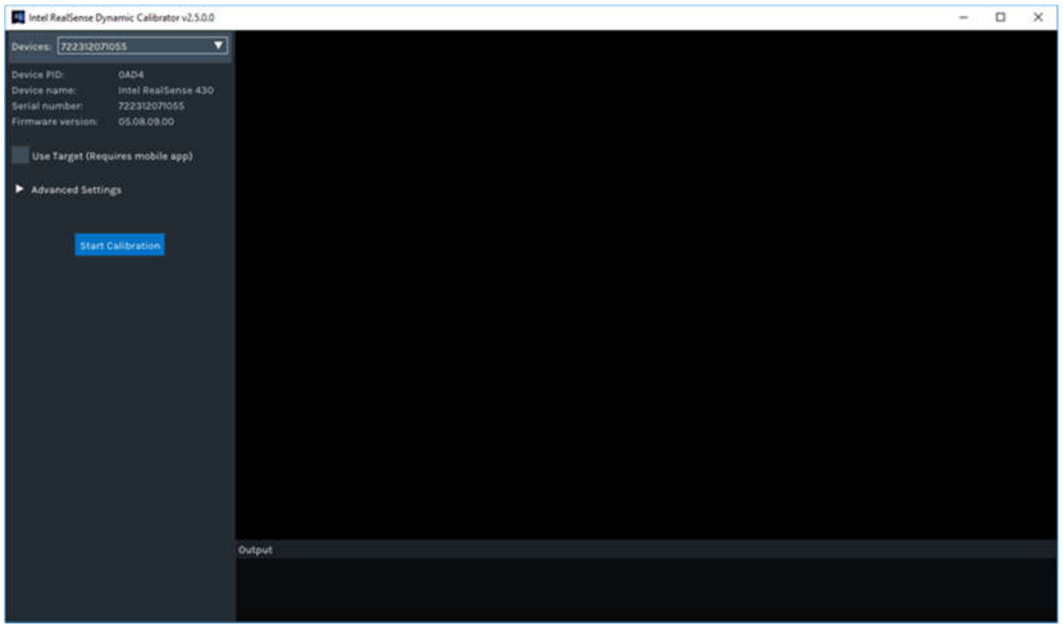
And specify appropriate options on the command line. The GUI control panel is not displayed in command line mode. Calibration process starts immediately by default.

On **Windows**:

If the application is installed through an installer, an app icon “Dynamic Calibrator” should be already created on your desktop.

Double click the icon to start the app.

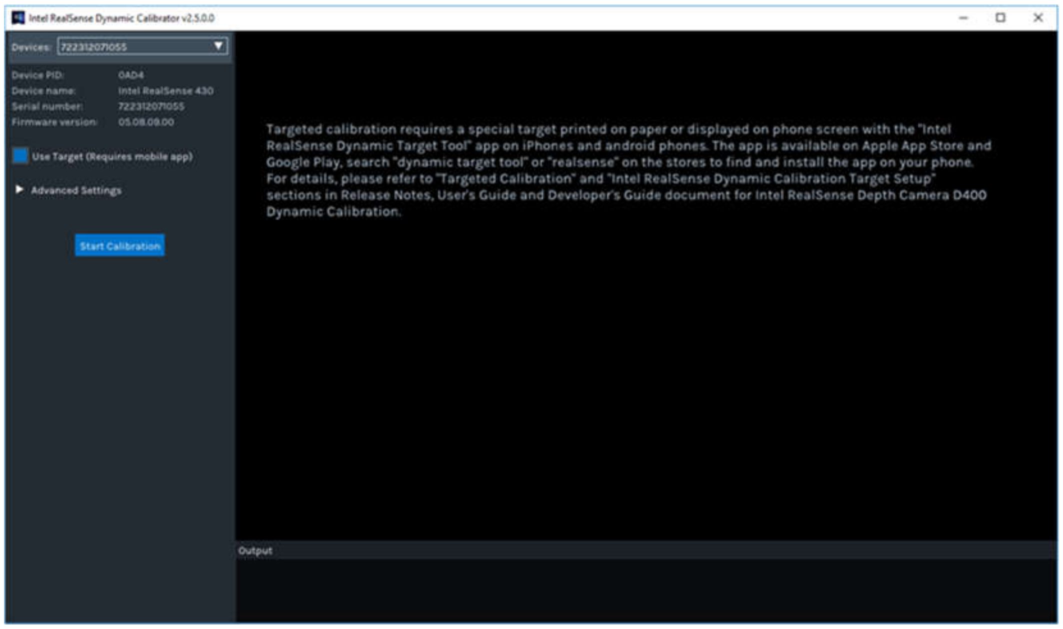
By default, the app starts with Graphical User Interface (GUI). It detects all Intel RealSense D400 devices on the system and lists them in a drop down box on the top left corner. Selects one of the device from the list to calibrate. Details on the device displayed as well:



Intel RealSense Dynamic Calibrator v2.5.0.0

Devices:	722312071055
Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

This calibration process requires a special target setup described in previous section "Setup Calibration Target". If you haven't done that, please follow instructions and complete the target set up either a printed target or phone target.

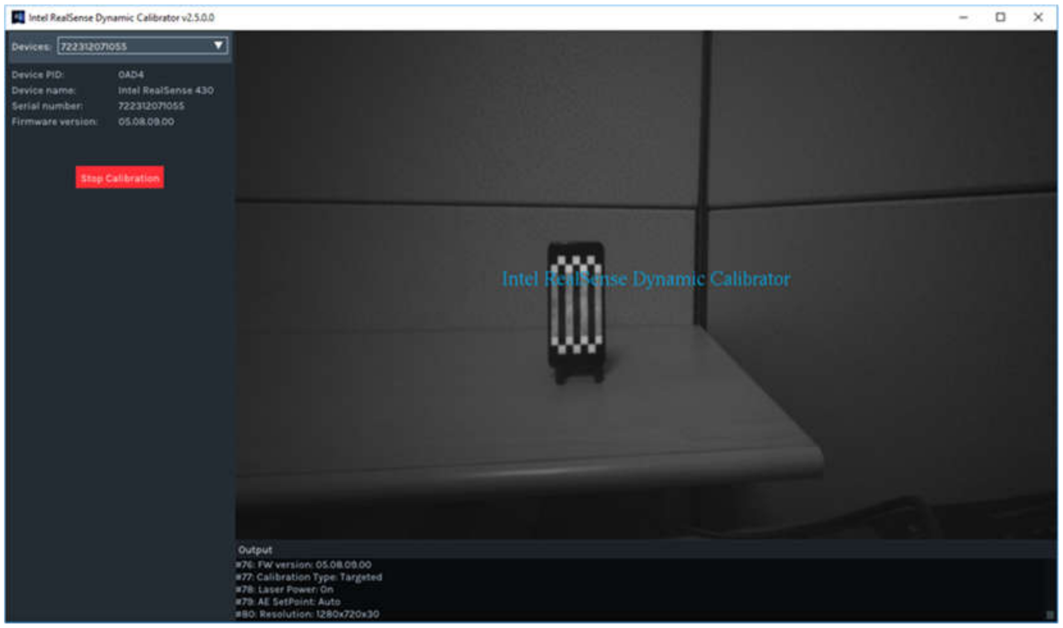


Intel RealSense Dynamic Calibrator v2.5.0.0

Devices:	722312071055
Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

Targeted calibration requires a special target printed on paper or displayed on phone screen with the "Intel RealSense Dynamic Target Tool" app on iPhones and android phones. The app is available on Apple App Store and Google Play, search "dynamic target tool" or "realsense" on the stores to find and install the app on your phone. For details, please refer to "Targeted Calibration" and "Intel RealSense Dynamic Calibration Target Setup" sections in Release Notes, User's Guide and Developer's Guide document for Intel RealSense Depth Camera D400 Dynamic Calibration.

Next to start calibration by clicking the "Start Calibration" button:


Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Stop Calibration
Intel RealSense Dynamic Calibrator
Output

Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

```
#76: FW version: 05.08.09.00
#77: Calibration Type: Targeted
#78: Laser Power: On
#79: AE SetPoint: Auto
#80: Resolution: 1280x720x30
```

If command line interface is preferred, user can display DynamicCalibrator command line options by running:

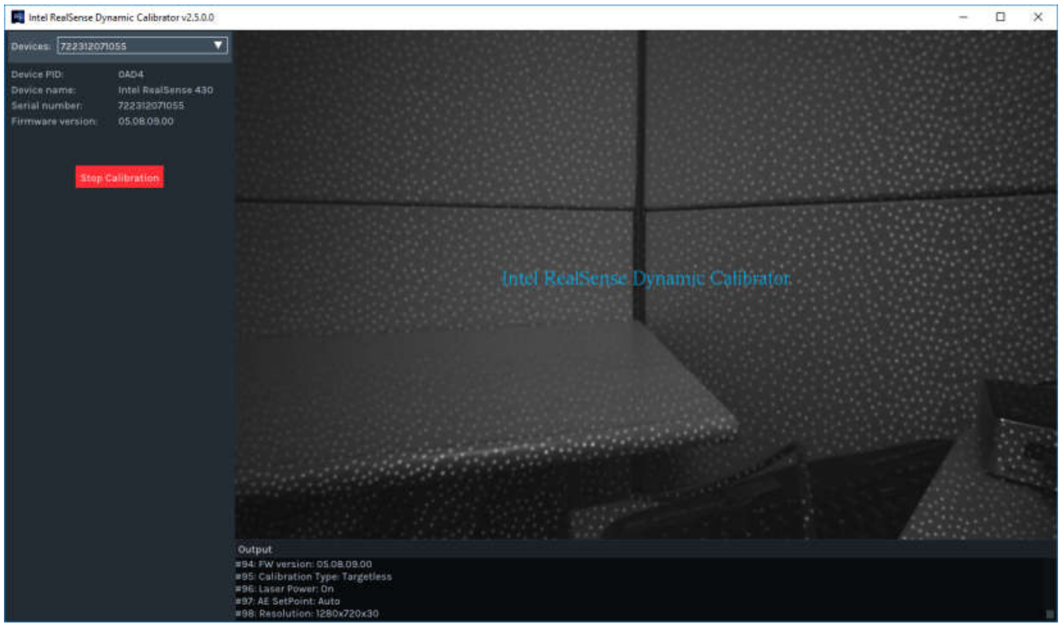
```
Cd bin

Intel.Realsense.DynamicCalibratorCLI.exe -help
```

And specify appropriate options on the command line. The GUI control panel is not displayed in command line mode. Calibration process starts immediately by default.

```
cd bin

Intel.Realsense.DynamicCalibratorCLI.exe
```

When DynamicCalibrator calibration process starts, it briefly displays a logo screen:

Figure 4-2 Dynamic Calibrator Initial Start Window


Intel RealSense Dynamic Calibrator v2.5.0.0
Stop Calibration
Intel RealSense Dynamic Calibrator
Output

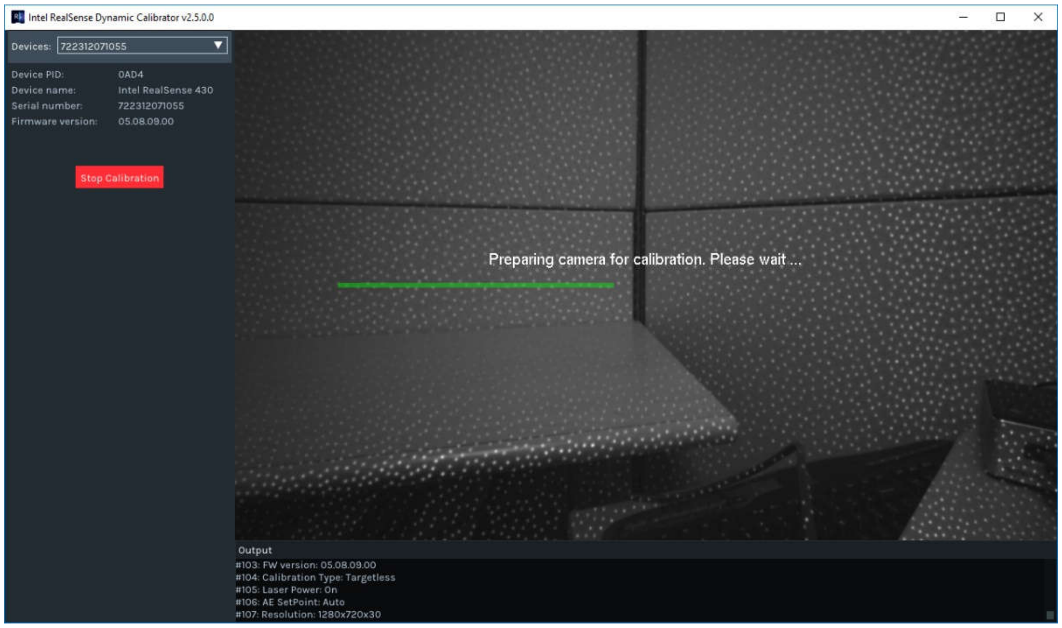
Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

```
#94: FW version: 05.08.09.00
#95: Calibration Type: Targetless
#96: Laser Power: On
#97: AE SetPoint: Auto
#98: Resolution: 1280x720x30
```

Followed by a warm up screen that indicates progress in getting the camera ready for calibration.

Figure 4-3 Dynamic Calibrator app warm up window



Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

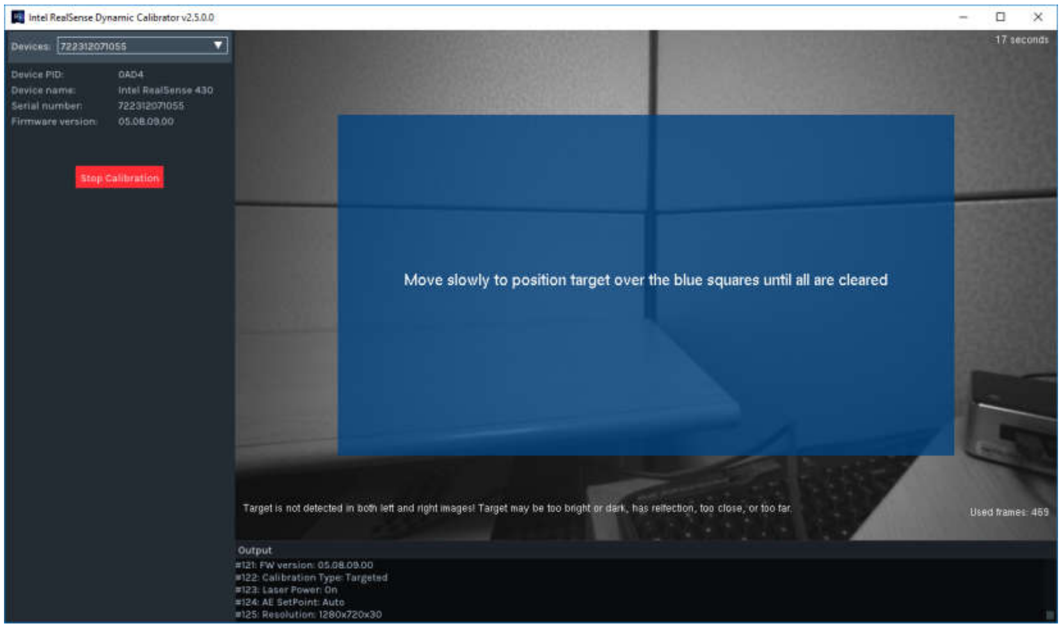
Preparing camera for calibration. Please wait ...

```
#103: FW version: 05.08.09.00
#104: Calibration Type: Targetless
#105: Laser Power: On
#106: AE SetPoint: Auto
#107: Resolution: 1280x720x30
```

4.5.3 Auto Exposure Adjustment during Targeted Calibration

Throughout the targeted calibration process, exposure is adjusted automatically. The tool tries to detect and recognize the calibration target in all lighting conditions. When a target is not found, sometimes because the surrounding environment has a reflection on the phone screen or the target is too dark/bright to detect, it will adjust exposure automatically so that a target is detected. So from time to time, you may see the exposure changes from dark to bright and gradually reduces brightness. This is expected behavior.

In some cases, if the tool adjusted too bright or too dark, and the tool can't make progress, you can point the camera away from the target and wait a couple seconds until the brightness comes to a normal level and then point back to the target.



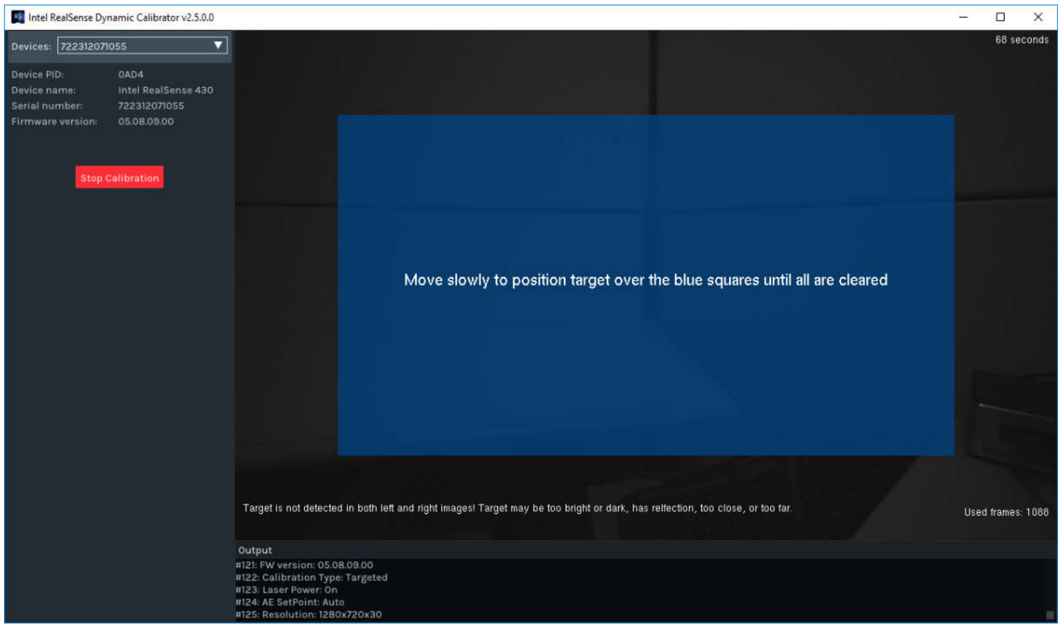
Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

Move slowly to position target over the blue squares until all are cleared

Target is not detected in both left and right images! Target may be too bright or dark, has reflection, too close, or too far.

Used frames: 469

```
#121: FW version: 05.08.09.00
#122: Calibration Type: Targeted
#123: Laser Power: On
#124: AE SetPoint: Auto
#125: Resolution: 1280x720x30
```



Intel RealSense Dynamic Calibrator v2.5.0.0

Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

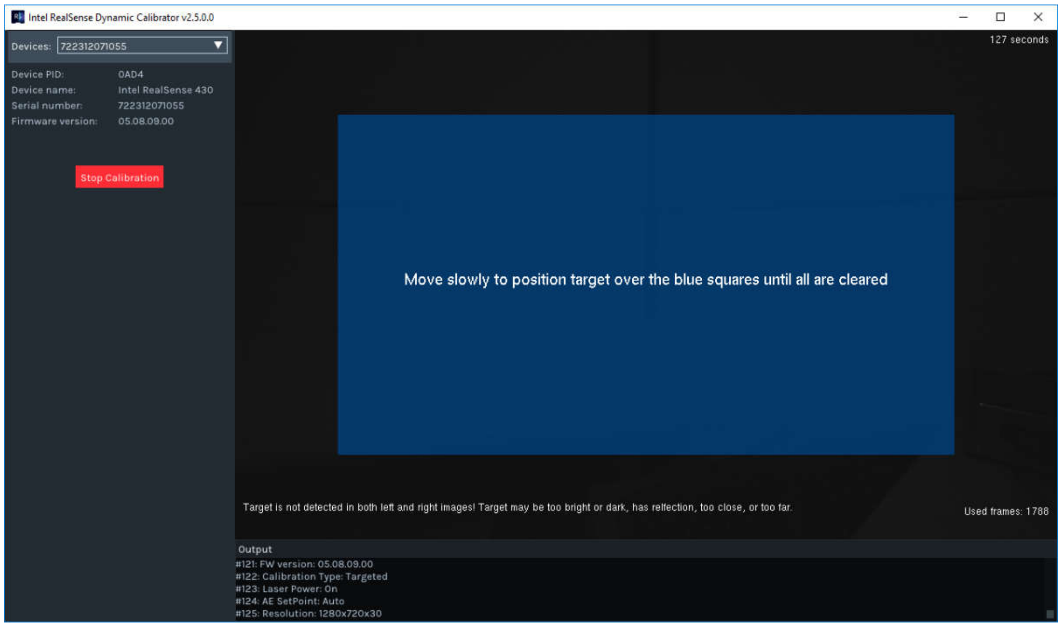
68 seconds

Move slowly to position target over the blue squares until all are cleared

Target is not detected in both left and right images! Target may be too bright or dark, has relfection, too close, or too far.

Used frames: 1088

```
#121: FW version: 05.08.09.00
#122: Calibration Type: Targeted
#123: Laser Power: On
#124: AE SetPoint: Auto
#125: Resolution: 1280x720x30
```



Intel RealSense Dynamic Calibrator v2.5.0.0

Devices: 722312071055

Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

127 seconds

Move slowly to position target over the blue squares until all are cleared

Target is not detected in both left and right images! Target may be too bright or dark, has relfection, too close, or too far.

Used frames: 1788

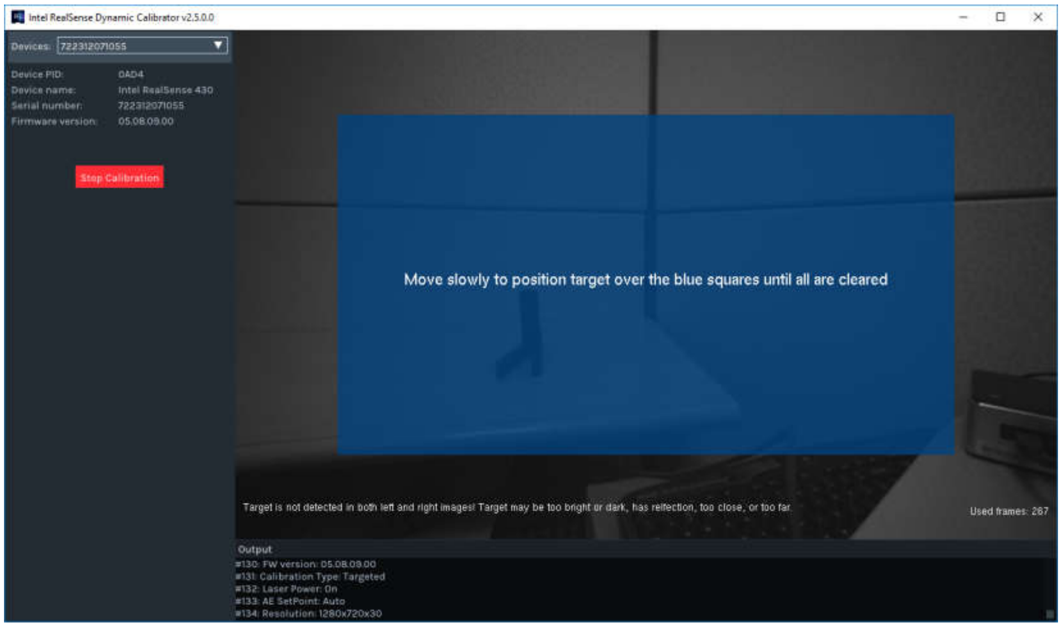
```
#121: FW version: 05.08.09.00
#122: Calibration Type: Targeted
#123: Laser Power: On
#124: AE SetPoint: Auto
#125: Resolution: 1280x720x30
```

4.5.4 Target calibration phases

Target calibration process consists of two phases – **rectification** and **scale** calibration in sequential order.

For the first phase, rectification, a block of be squares are present in middle part of the window:

Figure 4-4 Dynamic Calibrator Targeted Rectification Phase



```
Devices: 722312071055
Device PID:        0AD4
Device name:       Intel RealSense 430
Serial number:     722312071055
Firmware version:  05.08.09.00
```

Move slowly to position target over the blue squares until all are cleared

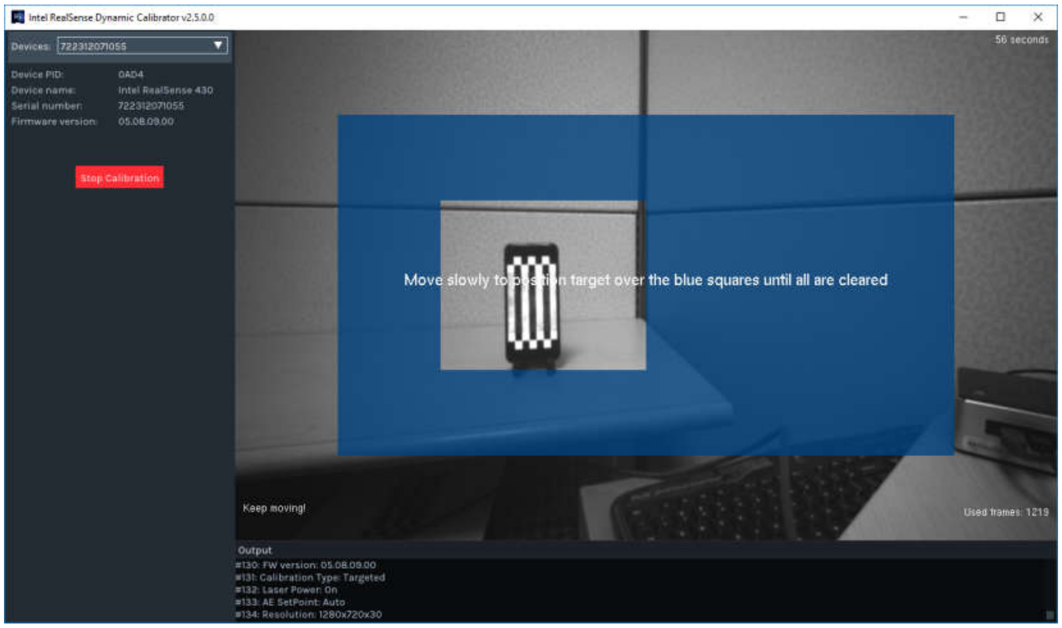
Target is not detected in both left and right images! Target may be too bright or dark, has reflection, too close, or too far.

Used frames: 267

```
Output
#130: FW version: 05.08.09.00
#131: Calibration Type: Targeted
#132: Laser Power: On
#133: AE SetPoint: Auto
#134: Resolution: 1280x720x30
```

Point the camera to the calibration target and move camera slowly to position target over the blue squares until all are cleared:

Figure 4-5 Dynamic Calibrator Targeted Rectification Phase (continued)



```
Devices: 722312071055
Device PID:        0AD4
Device name:       Intel RealSense 430
Serial number:     722312071055
Firmware version:  05.08.09.00
```

56 seconds

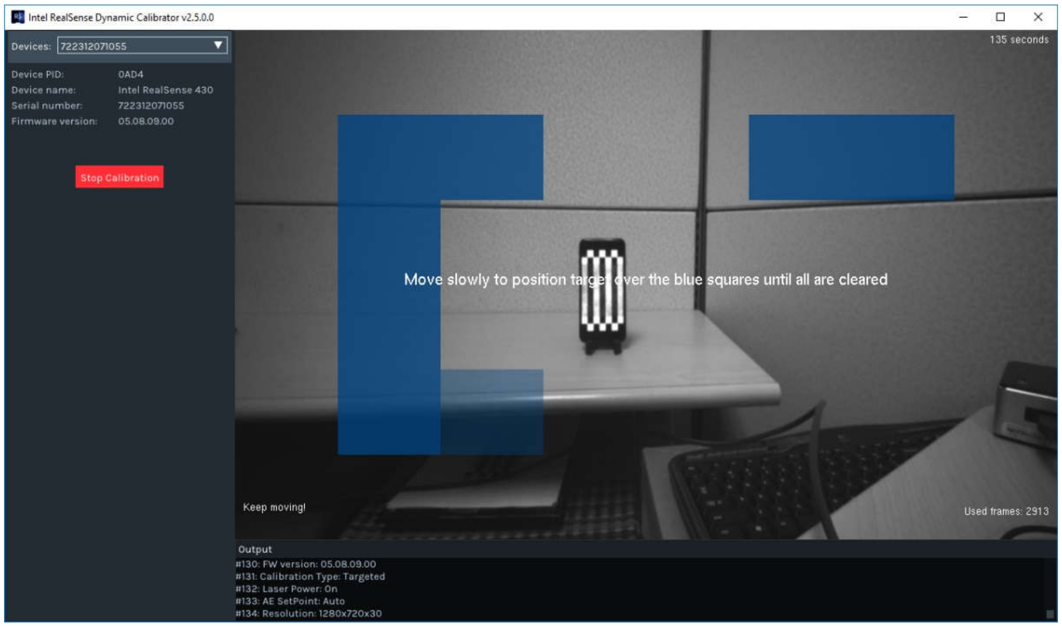
Move slowly to position target over the blue squares until all are cleared

Keep moving!

Used frames: 1219

```
Output
#130: FW version: 05.08.09.00
#131: Calibration Type: Targeted
#132: Laser Power: On
#133: AE SetPoint: Auto
#134: Resolution: 1280x720x30
```

Figure 4-6 Dynamic Calibrator Targeted Rectification Phase (continued)


Intel RealSense Dynamic Calibrator v2.5.0.0
Stop Calibration
135 seconds
Move slowly to position target over the blue squares until all are cleared
Keep moving!
Used frames: 2913

Devices: 722312071055

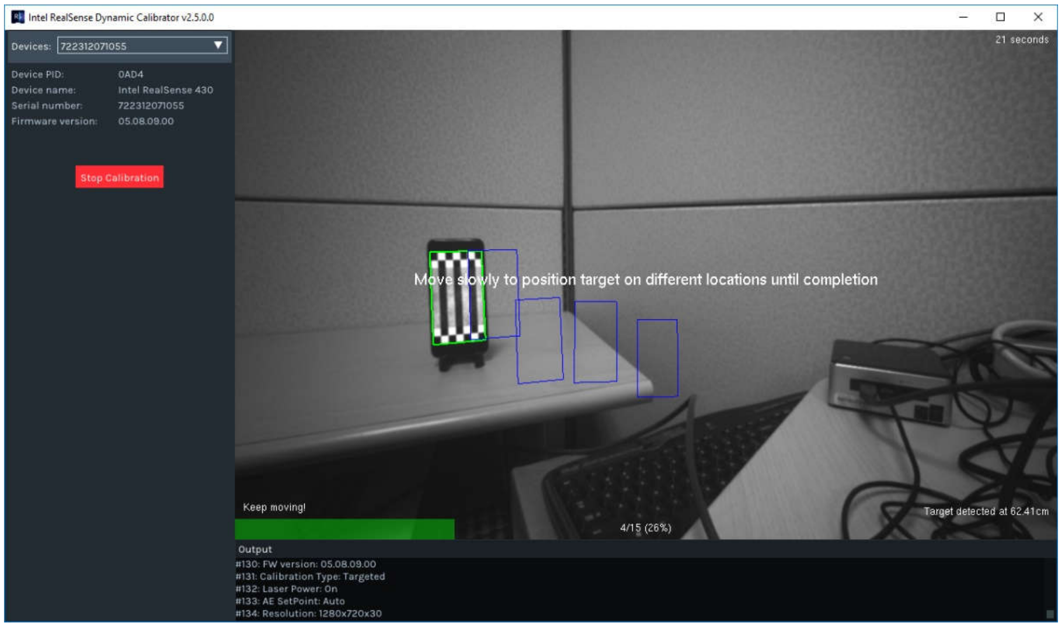
Device PID:	0AD4
Device name:	Intel RealSense 430
Serial number:	722312071055
Firmware version:	05.08.09.00

```
Output
#130: FW version: 05.08.09.00
#131: Calibration Type: Targeted
#132: Laser Power: On
#133: AE SetPoint: Auto
#134: Resolution: 1280x720x30
```

Once all blue squares are cleared, rectification phase is completed. Intermediate result is applied to the current stream.

Scale calibration phase will start immediately afterwards. Point the camera to the target and move slowly. A green progress bar will appear as it makes progress in scale calibration. Target images will be captured and analyzed and accepted target positions highlighted on screen. Total 15 successful target images will be accepted.

Figure 4-7 Dynamic Calibrator Scale Phase



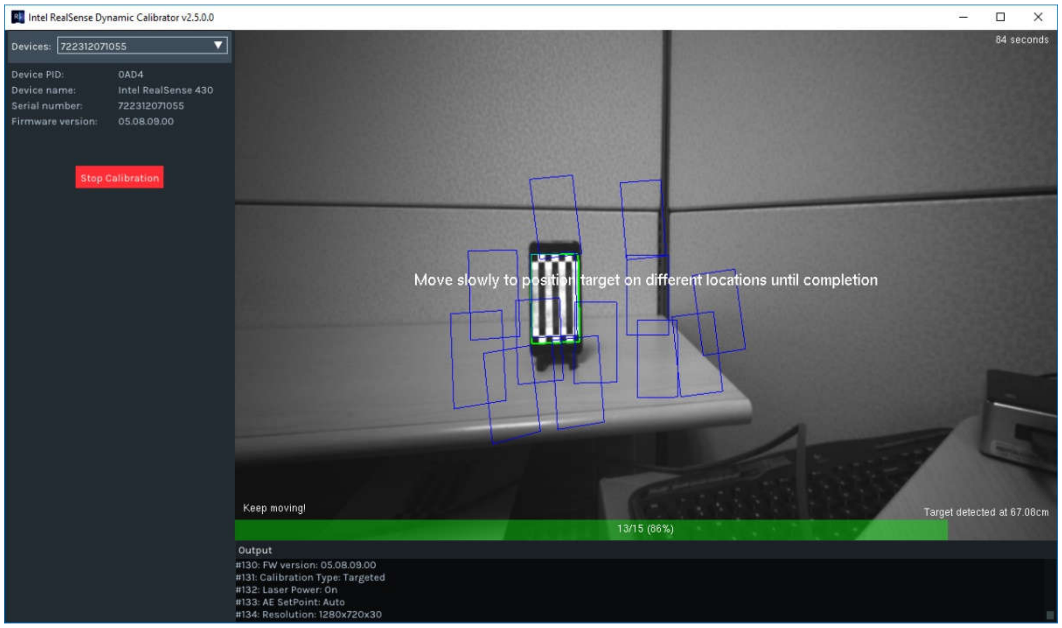
Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

Move slowly to position target on different locations until completion

Target detected at 62.41cm

```
#130: FW version: 05.08.09.00
#131: Calibration Type: Targeted
#132: Laser Power: On
#133: AE SetPoint: Auto
#134: Resolution: 1280x720x30
```

Figure 4-8 Dynamic Calibrator Scale Phase (continued)



Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

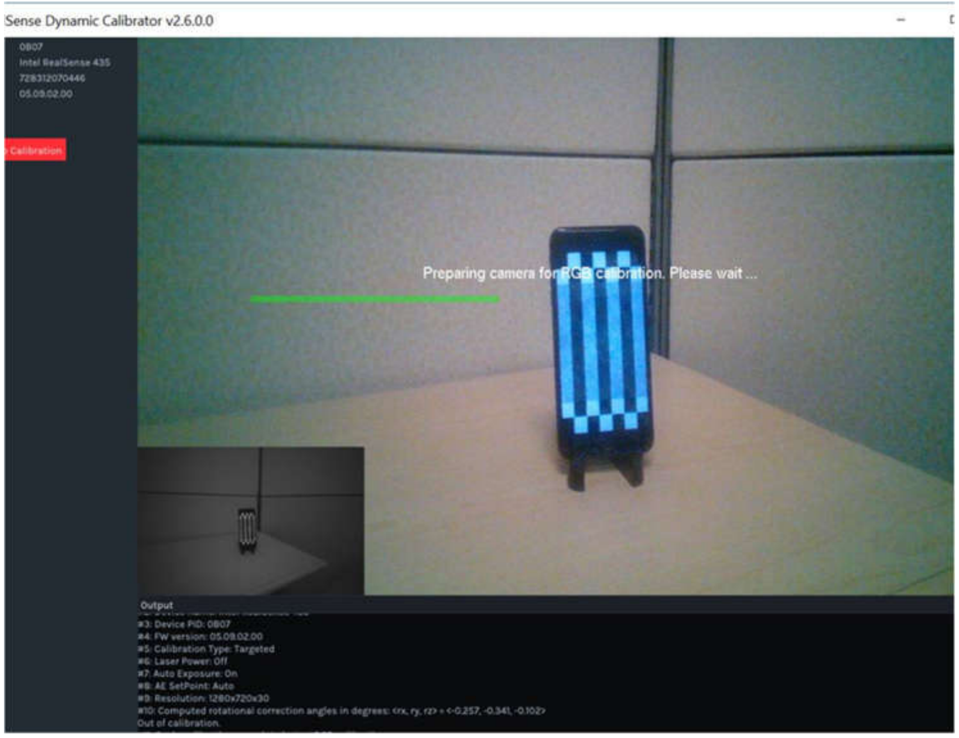
Move slowly to position target on different locations until completion

Target detected at 67.08cm

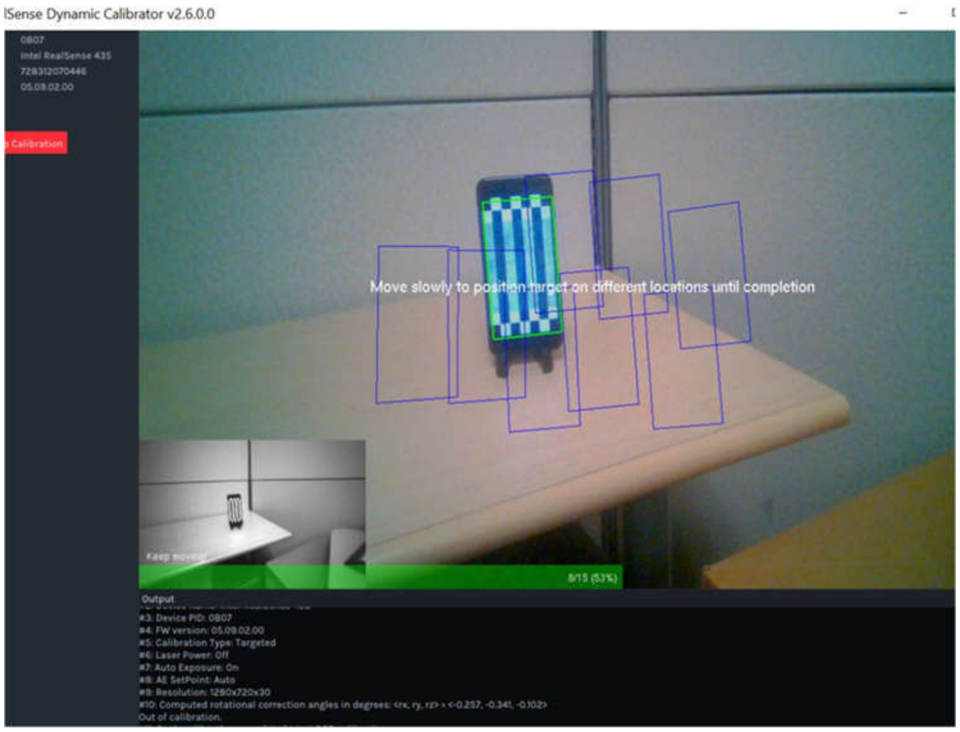
```
#130: FW version: 05.08.09.00
#131: Calibration Type: Targeted
#132: Laser Power: On
#133: AE SetPoint: Auto
#134: Resolution: 1280x720x30
```

After the green bar extends to full, scale calibration completes.

On D435 and D415 devices, an extra step will be run to calibrate the RGB camera. The usage is similar as the depth scale phase.


Sense Dynamic Calibrator v2.6.0.0
0807
Intel RealSense 435
728312070446
05.09.02.00
Calibration
Preparing camera for RGB calibration. Please wait ...
Output

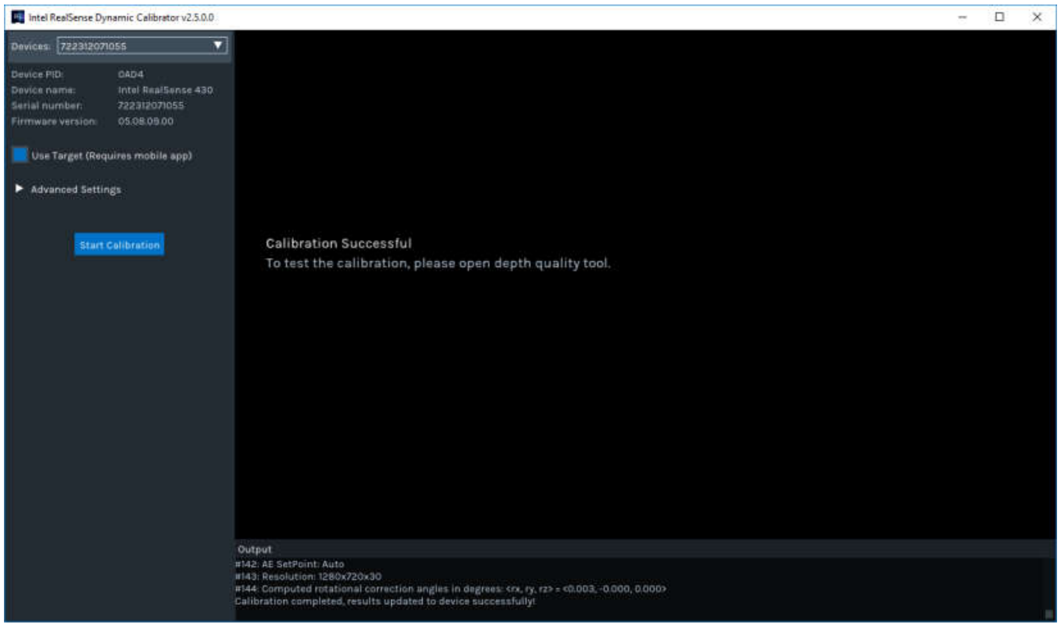
```
#3: Device PID: 0807
#4: FW version: 05.09.02.00
#5: Calibration Type: Targeted
#6: Laser Power: Off
#7: Auto Exposure: On
#8: AE SetPoint: Auto
#9: Resolution: 1280x720x30
#10: Computed rotational correction angles in degrees: <rx, ry, rz> = <-0.257, -0.341, -0.102>
Out of calibration.
```


lSense Dynamic Calibrator v2.6.0.0
0807
Intel RealSense 435
728312070446
05.08.02.00
Calibration
Move slowly to position target on different locations until completion
Keep moving
8/15 (53%)
Output

```
#3: Device PID: 0807
#4: FW version: 05.09.02.00
#5: Calibration Type: Targeted
#6: Laser Power: Off
#7: Auto Exposure: On
#8: AE SetPoint: Auto
#9: Resolution: 1280x720x30
#10: Computed rotational correction angles in degrees: <rx, ry, rz> = <-0.257, -0.341, -0.102>
Out of calibration.
```

Depends on what user interface (GUI or CLI) used, result will be either on the GUI status box or displayed in the command line window. Results will be updated to device automatically. Success or fail status will also be displayed along with the results.

Results update in GUI mode:

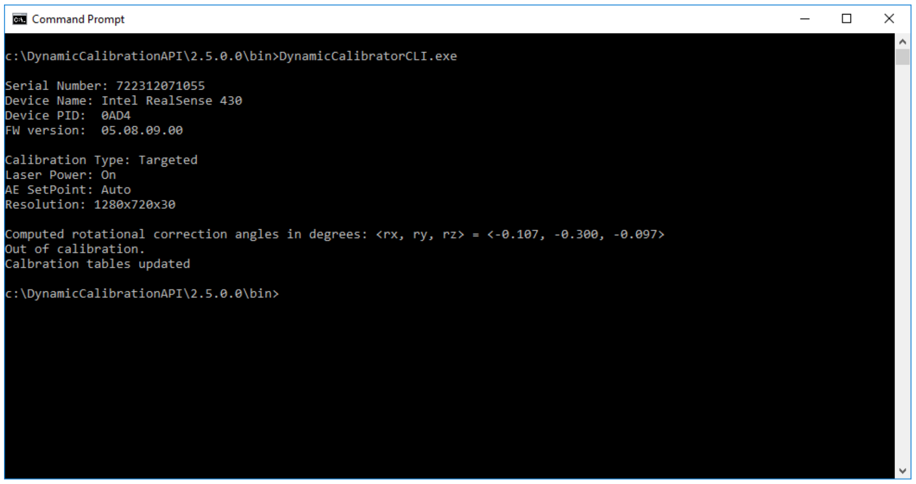

Intel RealSense Dynamic Calibrator v2.5.0.0
Devices: 722312071055
Use Target (Requires mobile app)
Advanced Settings
Start Calibration
Output

Device PID: 0AD4
Device name: Intel RealSense 430
Serial number: 722312071055
Firmware version: 05.08.09.00

Calibration Successful
To test the calibration, please open depth quality tool.

```
#142: AE SetPoint: Auto
#143: Resolution: 1280x720x30
#144: Computed rotational correction angles in degrees: <rx, ry, rz> = <0.003, -0.000, 0.000>
Calibration completed, results updated to device successfully!
```

Results update in CLI mode:



```
c:\DynamicCalibrationAPI\2.5.0.0\bin>DynamicCalibratorCLI.exe

Serial Number: 722312071055
Device Name: Intel RealSense 430
Device PID:  0AD4
FW version:  05.08.09.00

Calibration Type: Targeted
Laser Power: On
AE SetPoint: Auto
Resolution: 1280x720x30

Computed rotational correction angles in degrees: <rx, ry, rz> = <-0.107, -0.300, -0.097>
Out of calibration.
Calbration tables updated

c:\DynamicCalibrationAPI\2.5.0.0\bin>
```

This completes the targeted calibration process.

5 Advanced Calibration Processes

As described in previous chapter, Dynamic Calibrator Command Line Interface (CLI) supports advanced options for user to explore.

- **Intel.Realsense.DynamicCalibratorCLI.exe** on Windows
- **Intel.Realsense.DynamicCalibrator -cli** on Linux

In addition to Targeted Calibration, it also supports a few advanced calibration processes for automation like calibrating devices in robotics.

The sample app source code is included as part of the Calibration API package, so users can experiment and develop own calibration app for their usage.

5.1 Target-less Calibration

This calibration mode does not require a target. User simply points the device to a scene with textures and move slowly until the calibration process finishes. The advantage is no target is required, so it's convenient in integration and automation where a target is not available. The disadvantage is that it's less accurate as Targeted Calibration. It rectifies the left and right cameras (improves fill rate) but does not improve scaling error as well as Targeted Calibration (less accurate and consistent Z-accuracy).

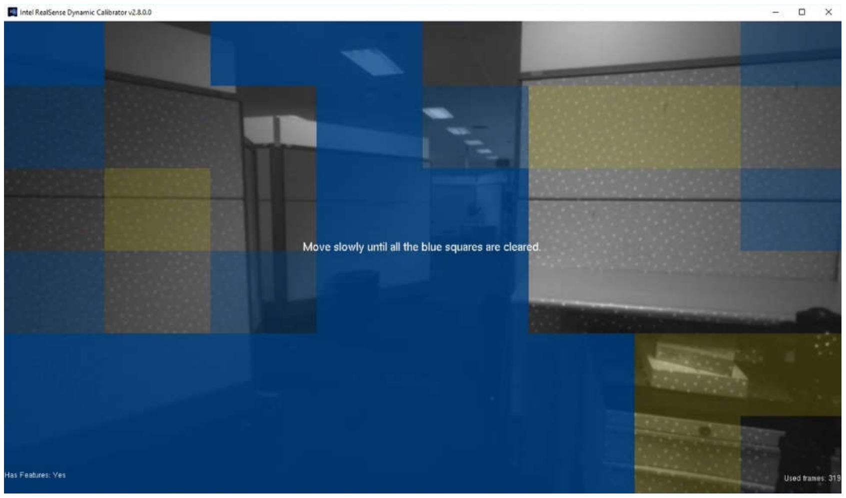
(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 0
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 0
```

Figure 5-1 Target-less Dynamic Calibration with Entire FOV


Intel RealSense Dynamic Calibrator v2.8.0.0
Move slowly until all the blue squares are cleared.
Has Features: Yes
Used frames: 319

By default, Target-less considers features in the entire FOV. In some cases, this can be troublesome, for example, device mounted on a robot and very close to the ground floor, or cases where features not available in the edges or corners of the FOV. Since the target-less calibration algorithm uses features in the entire FOV, it will take much longer than usual to complete the calibration process or in worst cases may not complete at all.

The -ignore-borders option is provided to handle such difficult situations. Accuracy may be slightly impacted.

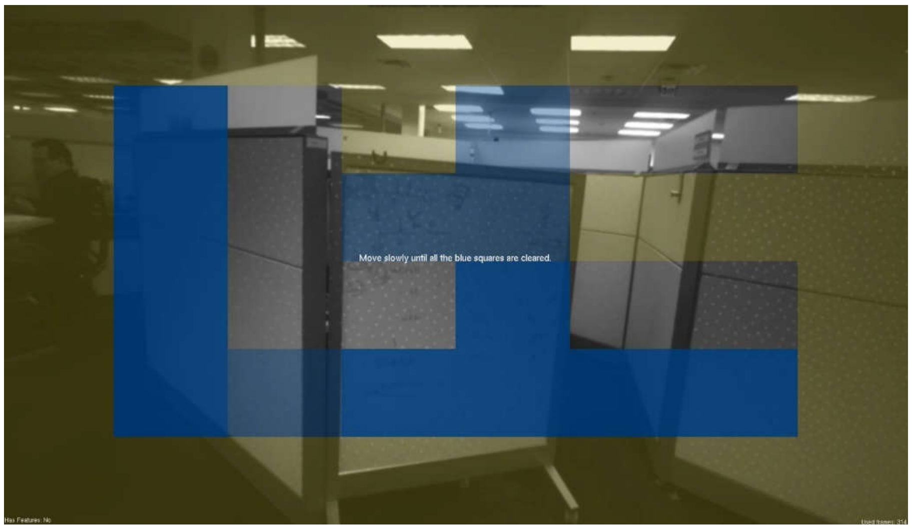
(Windows)

Intel.Realsense.DynamicCalibratorCLI.exe -m 0 -ignore-borders

(Linux)

Intel.Realsense.DynamicCalibrator -cli -m 0 -ignore-borders

Figure 5-2 Target-less Dynamic Calibration with Center FOV



5.2 Hybrid Calibration

Hybrid Calibration combines best of Target-less Calibration (convenience) and Targeted Calibration (accuracy) into a single process. It starts with a target-less calibration to rectifies the left/right images and then followed with a targeted scale calibration. It provides flexibility as well as accuracy.

(Windows)

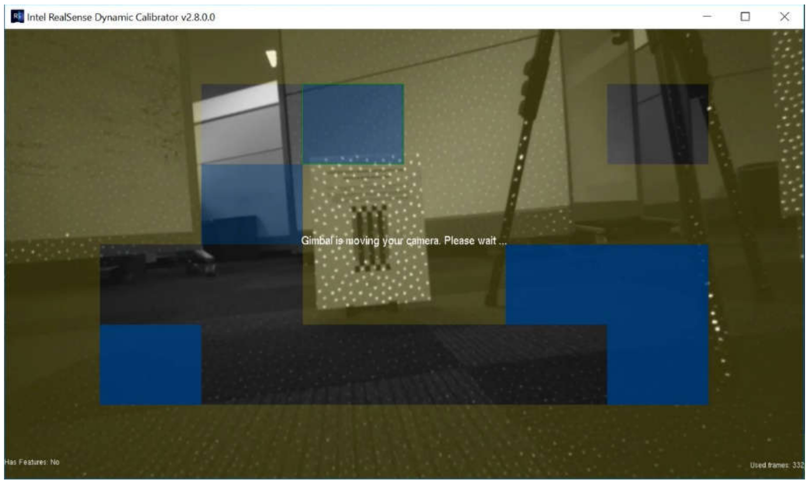
```
Intel.Realsense.DynamicCalibratorCLI.exe -m 2
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 2
```

Hybrid Calibration starts the initial phase in target-less calibration:

Figure 5-3 Hybrid Dynamic Calibration – Target-less Phase



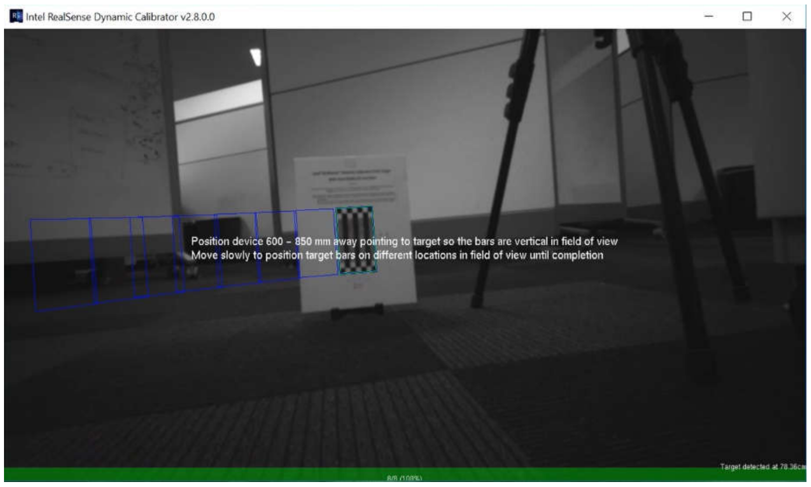
Gimbal is moving your camera. Please wait ...

Has Features: No

Used frames: 332

Once the target-less rectification is completed, it continues to a scale calibration with the same target used in Targeted Calibration.

Figure 5-4 Hybrid Dynamic Calibration – Targeted Scale Calibration Phase



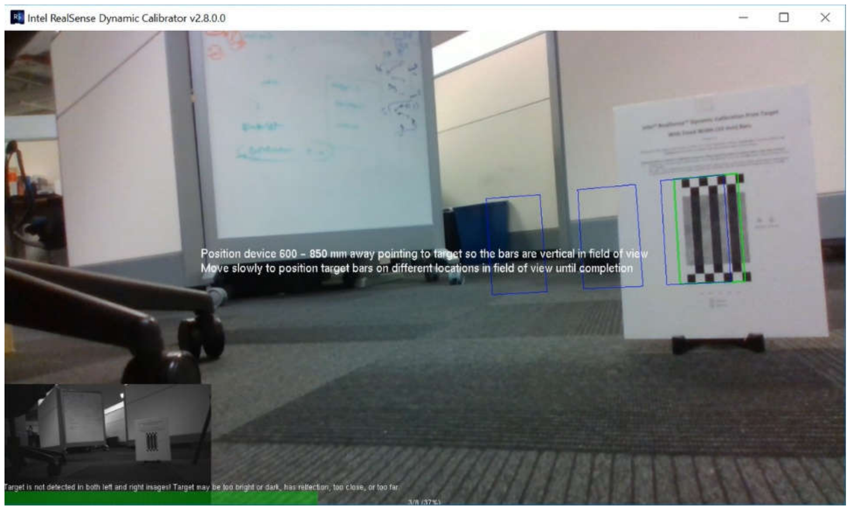
Position device 600 – 850 mm away pointing to target so the bars are vertical in field of view
Move slowly to position target bars on different locations in field of view until completion

Target detected at 78.36cm

8/8 (100%)

On devices with RGB, like D435 and D415, the calibration process will continue to a third phase to calibrate the RGB. However, this phase can be skipped with the -skiprgb option if user does not use the RGB in their usage.

Figure 5-5 Hybrid Dynamic Calibration – Targeted RGB Calibration Phase



Position device 600 – 850 mm away pointing to target so the bars are vertical in field of view
Move slowly to position target bars on different locations in field of view until completion

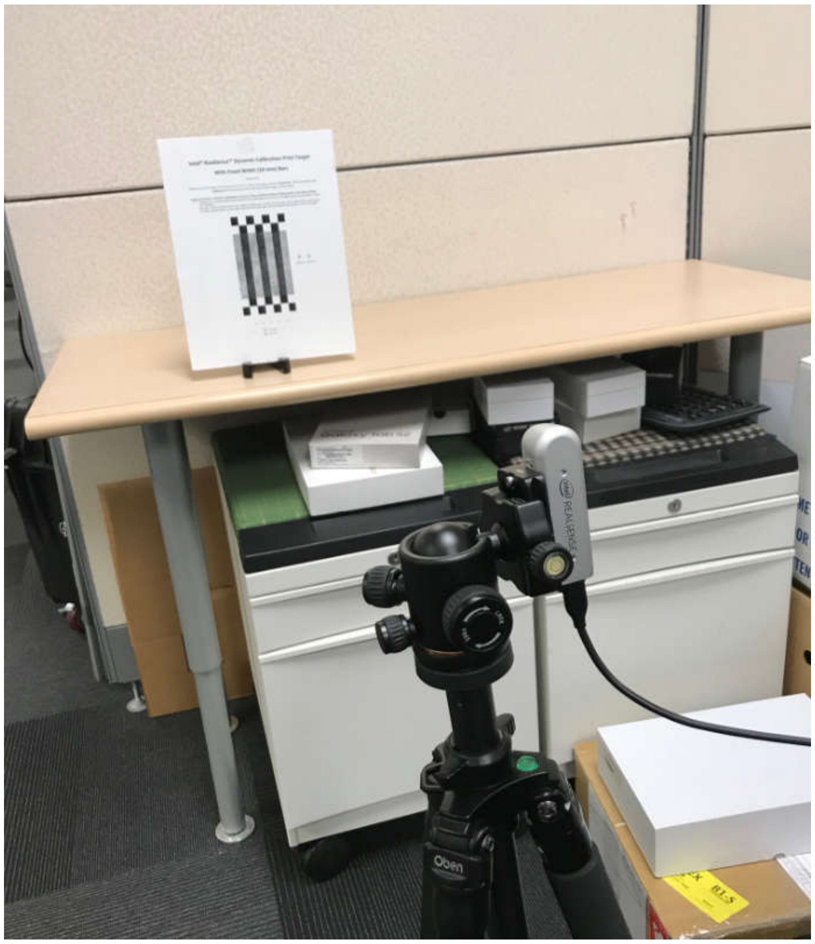
Target is not detected in both left and right images! Target may be too bright or dark, has relfection, too close, or too far.

A few additional command line options are provided to fit into user usage:

Option	Description
-a -aligned	device and target aligned in orientation, for example, device is mounted vertically, and target is positioned vertically
-skiprgb	Skip RGB calibration. This is useful if device like D435 and D415 equiped with a RGB camera but user does not use it. Skip RGB phase during calibration process will not impact depth calibration and it makes the process much faster.
-ignore-borders	ignore the border blocks during target-less rectification
-max-images <more than 6>	Specify number of images to capture in scale calibration and rgb calibration phases. This is used when user usage limited by time or space and would like to experiment best settings for their usage. • In targeted calibration, the default is 16 images. • In hybrid calibration, the default is either 6 or 8 images depends on device/target orientation. • For optimal results, recommend at least 6 images.
-force	Force the app to accept specified parameters, even if it's outside of the recommended range. Currently this only applies to -max-images.

When the device is mounted vertically on a robot and the robot is limited in any vertical movement. A very limited number of target images can be fit into the FOV due to the movement limitation.

Figure 5-6 Hybrid Dynamic Calibration – Device and Target Aligned



In this case place the target with the stripes vertically as well and use -a option to align the target and the device. More target images can be captured in the scale calibration and RGB calibration phases and therefore better accuracy in the calibration result.

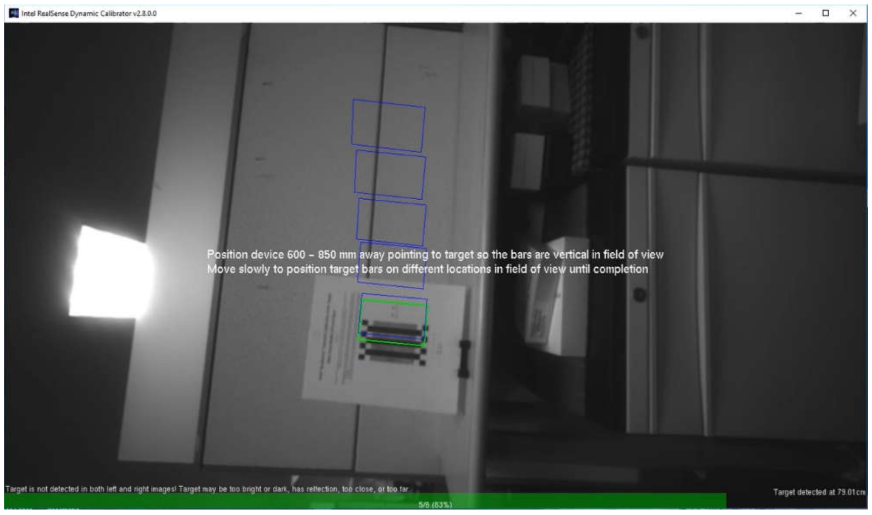
(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 2 -a
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 2 -a
```

Figure 5-7 Hybrid Dynamic Calibration – Device and Target Aligned



Position device 600 - 850 mm away pointing to target so the bars are vertical in field of view
Move slowly to position target bars on different locations in field of view until completion

Target is not detected in both left and right images! Target may be too bright or dark, has reflection, too close, or too far

Target detected at 79.01cm

5/6 (83%)

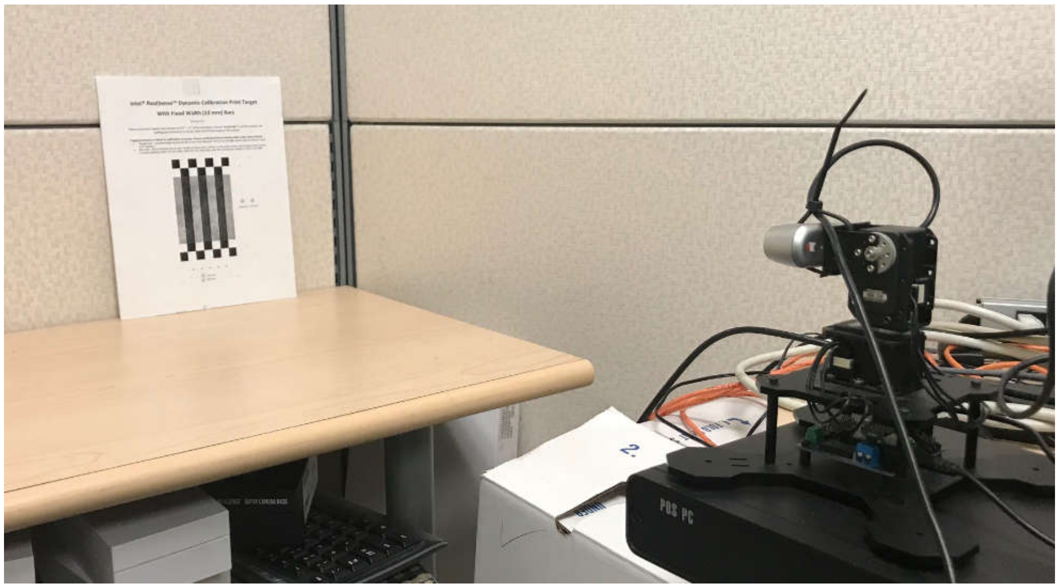
5.3 Calibration with a Robot (Gimbal)

Both Target-less Calibration and Hybrid Calibration can be integrated and automated on a robot. A complete set of programmable interfaces is provided in Calibration API. The Calibrator app demonstrates this ability through the -gimbal option.

5.3.1 Setup

The sample setup is shown below including a Dynamixel MX-28 robot actuator on Ardino board. The Intel® RealSense™ Product Family D400 Series camera is mounted horizontally on the robot.

Figure 5-8 Dynamic Calibration on Robot – Sample Setup



5.3.2 Gimbal

The robot is connected to the PC through a USB cable. For information in setting up and programming the gimbal robot, please refer to ***Appendix E – Setting Up and Program Gimbal Robot***.

5.3.3 Run Calibration with the Robot

Target-less Calibration with a robot:

(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 0 -gimbal <COM port on your system>
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 0 -gimbal /dev/ttyUSB0
```

Hybrid Calibration with a robot:

(Windows)

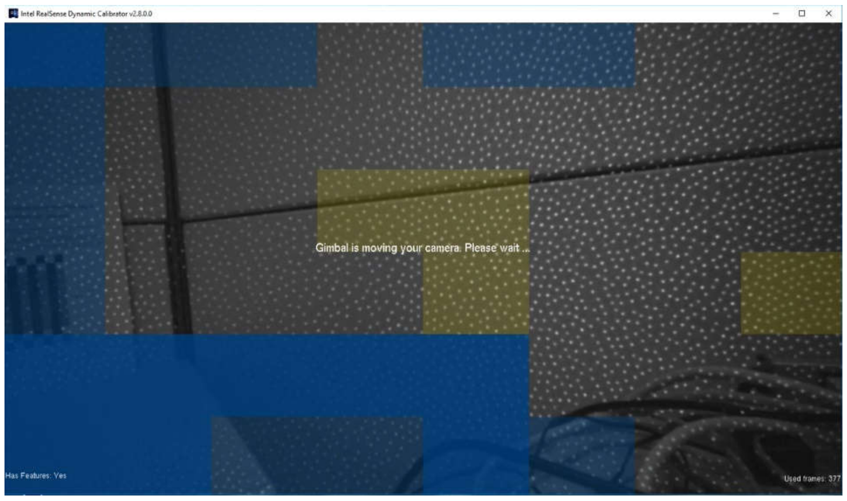
```
Intel.Realsense.DynamicCalibratorCLI.exe -m 2 -gimbal <COM port on your system>
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 2 -gimbal /dev/ttyUSB0
```

The calibration process only rotates the device through the simple robot used in this sample. When user develops their own calibration app on their robot, it can take advantage of horizontal or other movement if available to improve the process.

Figure 5-9 Dynamic Calibration on Robot

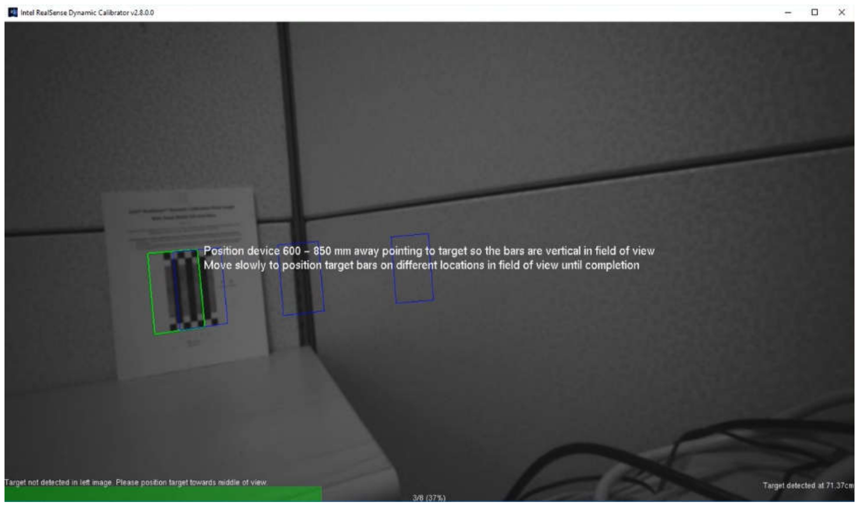


Gimbal is moving your camera. Please wait ...

Has Features: Yes

Used frames: 377

Figure 5-10 Dynamic Calibration on Robot



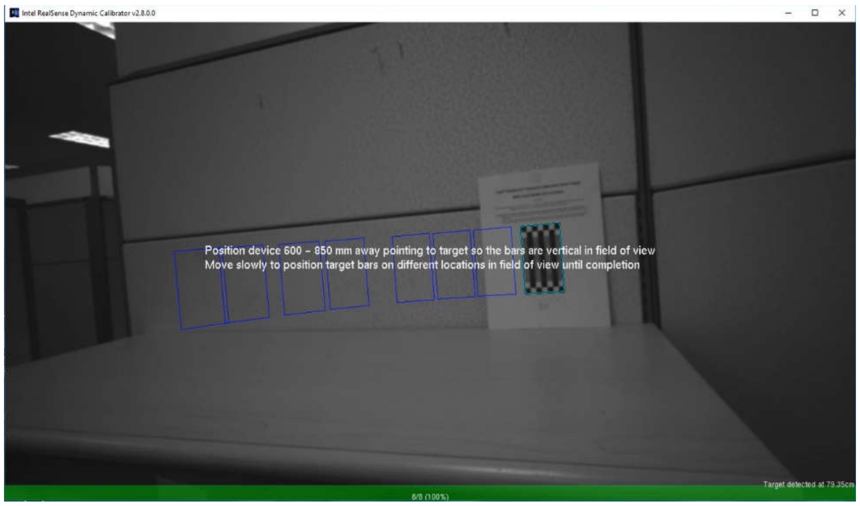
Position device 600 – 850 mm away pointing to target so the bars are vertical in field of view
Move slowly to position target bars on different locations in field of view until completion

Target not detected in left image. Please position target towards middle of view.

3/8 (37%)

Target detected at 71.37cm

Figure 5-11 Dynamic Calibration on Robot



Intel RealSense Dynamic Calibrator v2.8.0.0

Position device 600 – 850 mm away pointing to target so the bars are vertical in field of view
Move slowly to position target bars on different locations in field of view until completion

Target detected at 79.35cm

8/8 (100%)

6 Handling Calibration Data with CustomRW Tool

While users calibrate devices with dynamic calibrator or through their own developed calibration tools based on Calibration API, there is a need to read, write, and restore calibration data on the device. CustomRW Tool provides this ability to users. CustomRW Tool is included in the Calibration Tool package and is installed by default under C:\CalibrationToolAPI\2.6.7.0\bin on Windows and /usr/bin on Linux.

The tool supports options to restore and read/write calibration data:

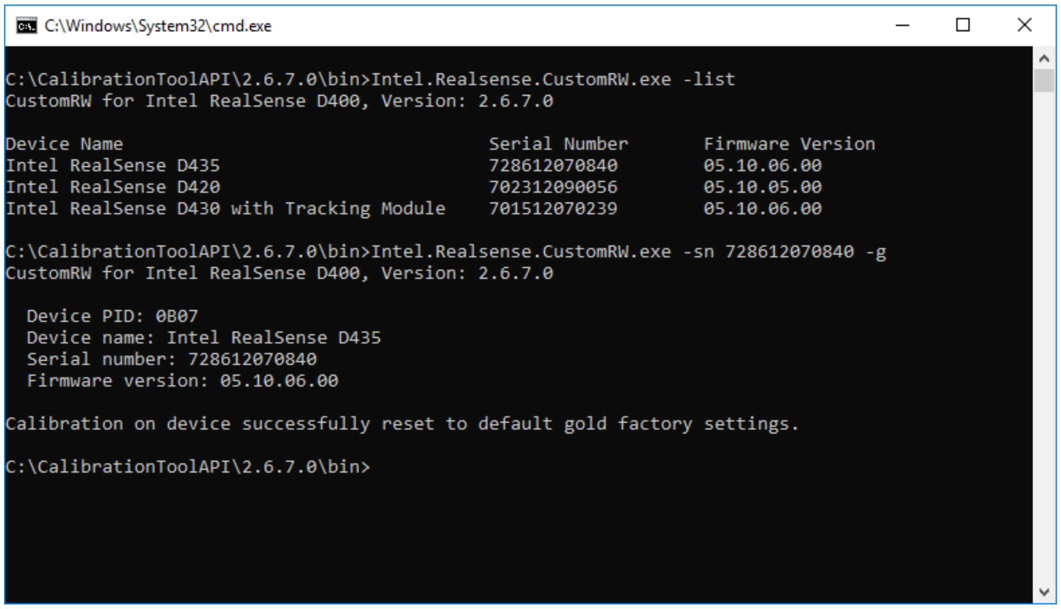
Table 6-1: CustomRW Command Line Options

Option	Description
-help or -?	Display list of command line options.
-version -v	Show version info
-list	Display list of connected devices with device name, serial number and firmware version information.
-sn <camera serial number> -serial < camera serial number>	If multiple cameras connected to the current system, choose one of the camera devices to calibrate by specifying its serial number. When a serial number is not specified, the tool will detect devices on the system and choose the first one to operate.
-g	Reset calibration on selected device to default gold factory settings
-r	Read calibration data
-w	Write calibration data
-raw <19, 1f, or 20>	Specify calibration table raw binary content to read/write. Valid table id 19, 1f, and 20
-fe	Specify custom fisheye data to read/write. Available ONLY on selected device SKU.
-mm	Specify custom IMU data to read/write. Available ONLY on D435i
-file <file name> -f <file name>	Specify file source/destination. When read, by default, without specifying a file destination, the data will be display on screen in the command console.

6.1 Restore Calibration on Device

The –g option restores the calibration on device to default gold settings. Depending up on the type of calibration done on the device, the gold settings represents the original factory setting or the calibration setting from OEM Calibration if such a calibration is done on the device after factory calibration.

Figure 6-1 Restore Calibration on Selected Device

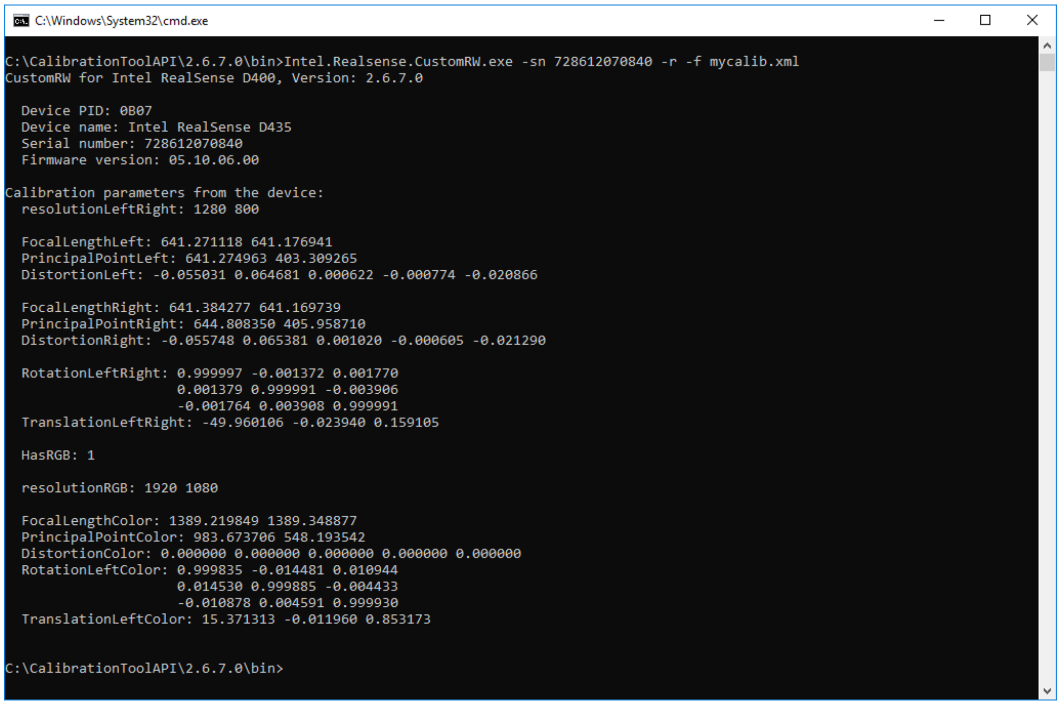


```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -list
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device Name                                  Serial Number       Firmware Version
Intel RealSense D435                         728612070840        05.10.06.00
Intel RealSense D420                         702312090056        05.10.05.00
Intel RealSense D430 with Tracking Module    701512070239        05.10.06.00

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -g
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

Calibration on device successfully reset to default gold factory settings.

C:\CalibrationToolAPI\2.6.7.0\bin>
```

6.2 Read Calibration from Device

The tool supports –r option to read the calibration parameters from the device. By default, the parameters are printed on screen. If a file name is provided, the parameters can also be dumped into a file in XML format.

Figure 6-2 Read Calibration on Selected Device



C:\Windows\System32\cmd.exe

```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -r -f mycalib.xml
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

Calibration parameters from the device:
  resolutionLeftRight: 1280 800

  FocalLengthLeft: 641.271118 641.176941
  PrincipalPointLeft: 641.274963 403.309265
  DistortionLeft: -0.055031 0.064681 0.000622 -0.000774 -0.020866

  FocalLengthRight: 641.384277 641.169739
  PrincipalPointRight: 644.808350 405.958710
  DistortionRight: -0.055748 0.065381 0.001020 -0.000605 -0.021290

  RotationLeftRight: 0.999997 -0.001372 0.001770
                     0.001379 0.999991 -0.003906
                     -0.001764 0.003908 0.999991
  TranslationLeftRight: -49.960106 -0.023940 0.159105

  HasRGB: 1

  resolutionRGB: 1920 1080

  FocalLengthColor: 1389.219849 1389.348877
  PrincipalPointColor: 983.673706 548.193542
  DistortionColor: 0.000000 0.000000 0.000000 0.000000 0.000000
  RotationLeftColor: 0.999835 -0.014481 0.010944
                     0.014530 0.999885 -0.004433
                     -0.010878 0.004591 0.999930
  TranslationLeftColor: 15.371313 -0.011960 0.853173


C:\CalibrationToolAPI\2.6.7.0\bin>
```

The calibration parameters are defined in section 2.1 "**Calibration Parameters**". The XML file format is as below:

```
<?xml version="1.0"?>
<Config>
  <param name = "ResolutionLeftRight">
    <value>1280</value>
    <value>800</value>
  </param>
  <param name = "FocalLengthLeft">
    <value>641.271</value>
    <value>641.177</value>
  </param>
  <param name = "PrincipalPointLeft">
    <value>641.275</value>
    <value>403.309</value>
  </param>
  <param name = "DistortionLeft">
    <value>-0.0550314</value>
    <value>0.0646809</value>
    <value>0.00062232</value>
    <value>-0.000774438</value>
    <value>-0.0208662</value>
  </param>
  <param name = "FocalLengthRight">
    <value>641.384</value>
    <value>641.17</value>
  </param>
```

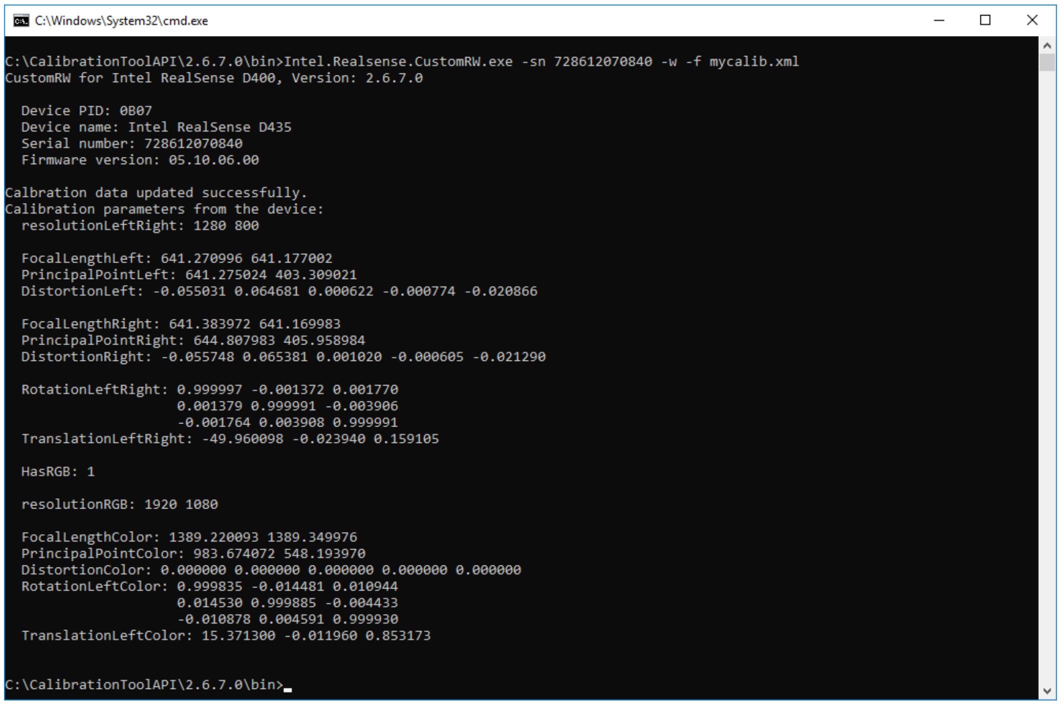
```
<param name = "PrincipalPointRight">
   <value>644.808</value>
   <value>405.959</value>
</param>
<param name = "DistortionRight">
   <value>-0.055748</value>
   <value>0.0653811</value>
   <value>0.00102014</value>
   <value>-0.000605032</value>
   <value>-0.0212901</value>
</param>
<param name = "RotationLeftRight">
   <value>0.999997</value>
   <value>-0.00137212</value>
   <value>0.00176977</value>
   <value>0.00137902</value>
   <value>0.999991</value>
   <value>-0.00390553</value>
   <value>-0.0017644</value>
   <value>0.00390796</value>
   <value>0.999991</value>
</param>
<param name = "TranslationLeftRight">
   <value>-49.9601</value>
   <value>-0.0239399</value>
   <value>0.159105</value>
</param>
<param name = "HasRGB">
   <value>1</value>
</param>
<param name = "ResolutionRGB">
   <value>1920</value>
   <value>1080</value>
</param>
<param name = "FocalLengthRGB">
   <value>1389.22</value>
   <value>1389.35</value>
</param>
<param name = "PrincipalPointRGB">
   <value>983.674</value>
   <value>548.194</value>
</param>
<param name = "DistortionRGB">
   <value>0</value>
   <value>0</value>
   <value>0</value>
   <value>0</value>
   <value>0</value>
</param>
<param name = "RotationLeftRGB">
   <value>0.999835</value>
   <value>-0.0144807</value>
   <value>0.0109438</value>
   <value>0.0145299</value>
   <value>0.999885</value>
```

```
      <value>-0.00443285</value>
      <value>-0.0108784</value>
      <value>0.00459113</value>
      <value>0.99993</value>
    </param>
    <param name = "TranslationLeftRGB">
      <value>15.3713</value>
      <value>-0.0119596</value>
      <value>0.853173</value>
    </param>
  </Config>
```

6.3 Write Calibration to Device

User can come up with the calibration parameters from their own calibration algorithm or from previous backups of the calibration data in XML format and write it to the device through the –w option.

Figure 6-3 Write Calibration on Selected Device



C:\Windows\System32\cmd.exe

```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -w -f mycalib.xml
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

Calbration data updated successfully.
Calibration parameters from the device:
  resolutionLeftRight: 1280 800

  FocalLengthLeft: 641.270996 641.177002
  PrincipalPointLeft: 641.275024 403.309021
  DistortionLeft: -0.055031 0.064681 0.000622 -0.000774 -0.020866

  FocalLengthRight: 641.383972 641.169983
  PrincipalPointRight: 644.807983 405.958984
  DistortionRight: -0.055748 0.065381 0.001020 -0.000605 -0.021290

  RotationLeftRight: 0.999997 -0.001372 0.001770
                     0.001379 0.999991 -0.003906
                     -0.001764 0.003908 0.999991
  TranslationLeftRight: -49.960098 -0.023940 0.159105

  HasRGB: 1

  resolutionRGB: 1920 1080

  FocalLengthColor: 1389.220093 1389.349976
  PrincipalPointColor: 983.674072 548.193970
  DistortionColor: 0.000000 0.000000 0.000000 0.000000 0.000000
  RotationLeftColor: 0.999835 -0.014481 0.010944
                     0.014530 0.999885 -0.004433
                     -0.010878 0.004591 0.999930
  TranslationLeftColor: 15.371300 -0.011960 0.853173


C:\CalibrationToolAPI\2.6.7.0\bin>_
```

6.4 Read and Write RAW Calibration Tables

The device keeps the calibration data in its internal raw tables. Intel maintains the tables in its proprietary formats. The read/write options above convert the internal formats to XML so users can understand the actual calibration parameters. There are cases where read/write the raw tables is necessary. For example, these raw tables can be generated by other Intel tools like OEM Calibration Tool with offline pre-captured images, user can then write the raw tables directly to the device through the –w option. Similarly, user can also choose to dump the raw tables for records and restore the device to this calibration data later.

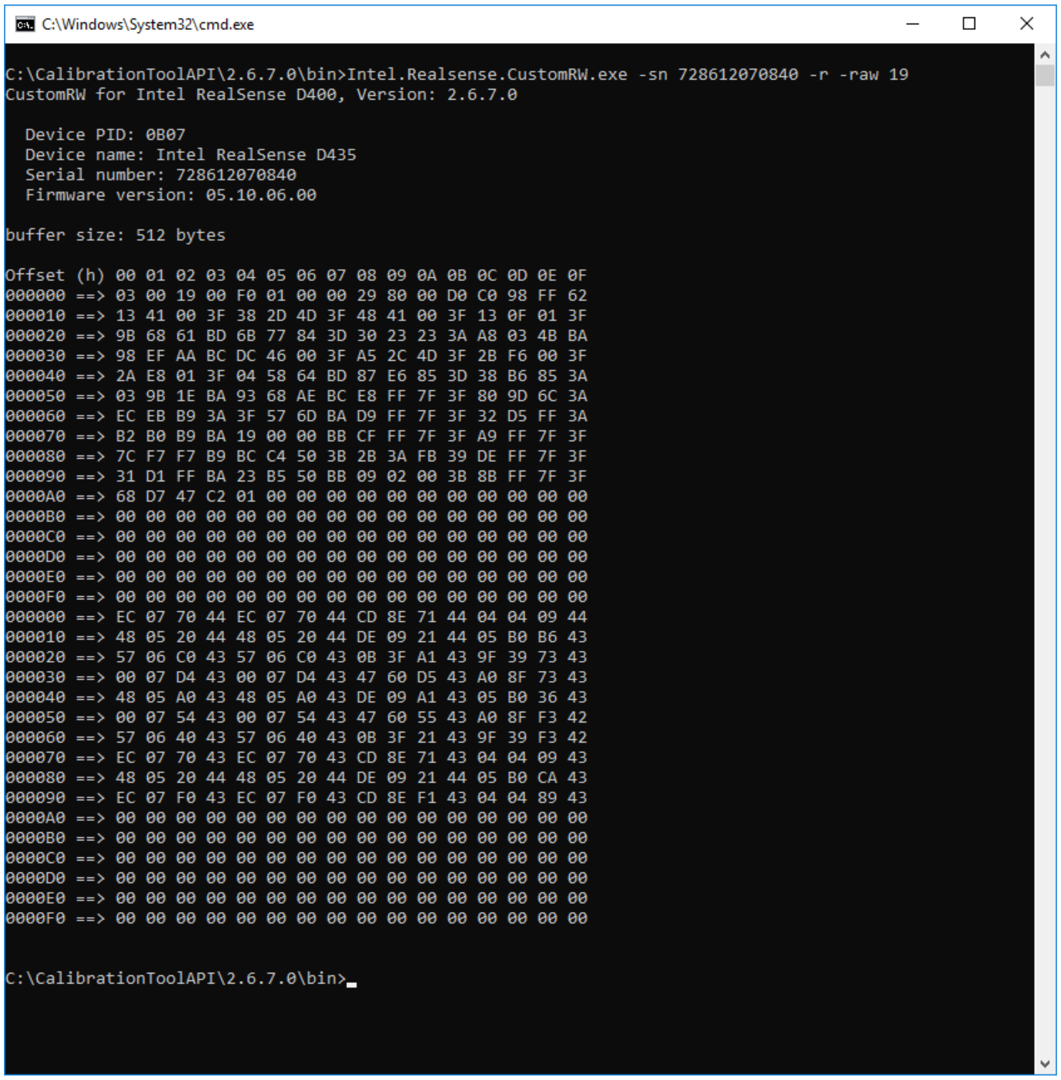
The -raw <table id> option supports raw tables 19 (coefficient table), 1f (depth table) and 20 (RGB table). Coefficient table and depth table are available on all devices. RGB table only exists on devices with RGB, like D415 and D435.

For example,

Example: read raw calibration table from device and display to the terminal, supported table id 19, 1F, and 20, for example,

```
Intel.Realsense.CustomRW.exe -r -raw 19
```

Figure 6-4 Read Calibration RAW Data on Selected Device and Display on Screen



C:\Windows\System32\cmd.exe

```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -r -raw 19
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

buffer size: 512 bytes

Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000 ==> 03 00 19 00 F0 01 00 00 29 80 00 D0 C0 98 FF 62
000010 ==> 13 41 00 3F 38 2D 4D 3F 48 41 00 3F 13 0F 01 3F
000020 ==> 9B 68 61 BD 6B 77 84 3D 30 23 23 3A A8 03 4B BA
000030 ==> 98 EF AA BC DC 46 00 3F A5 2C 4D 3F 2B F6 00 3F
000040 ==> 2A E8 01 3F 04 58 64 BD 87 E6 85 3D 38 B6 85 3A
000050 ==> 03 9B 1E BA 93 68 AE BC E8 FF 7F 3F 80 9D 6C 3A
000060 ==> EC EB B9 3A 3F 57 6D BA D9 FF 7F 3F 32 D5 FF 3A
000070 ==> B2 B0 B9 BA 19 00 00 BB CF FF 7F 3F A9 FF 7F 3F
000080 ==> 7C F7 F7 B9 BC C4 50 3B 2B 3A FB 39 DE FF 7F 3F
000090 ==> 31 D1 FF BA 23 B5 50 BB 09 02 00 3B 8B FF 7F 3F
0000A0 ==> 68 D7 47 C2 01 00 00 00 00 00 00 00 00 00 00 00
0000B0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000C0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000D0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000E0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000F0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000 ==> EC 07 70 44 EC 07 70 44 CD 8E 71 44 04 04 09 44
000010 ==> 48 05 20 44 48 05 20 44 DE 09 21 44 05 B0 B6 43
000020 ==> 57 06 C0 43 57 06 C0 43 0B 3F A1 43 9F 39 73 43
000030 ==> 00 07 D4 43 00 07 D4 43 47 60 D5 43 A0 8F 73 43
000040 ==> 48 05 A0 43 48 05 A0 43 DE 09 A1 43 05 B0 36 43
000050 ==> 00 07 54 43 00 07 54 43 47 60 55 43 A0 8F F3 42
000060 ==> 57 06 40 43 57 06 40 43 0B 3F 21 43 9F 39 F3 42
000070 ==> EC 07 70 43 EC 07 70 43 CD 8E 71 43 04 04 09 43
000080 ==> 48 05 20 44 48 05 20 44 DE 09 21 44 05 B0 CA 43
000090 ==> EC 07 F0 43 EC 07 F0 43 CD 8E F1 43 04 04 89 43
0000A0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000B0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000C0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000D0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000E0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000F0 ==> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00


C:\CalibrationToolAPI\2.6.7.0\bin>_
```

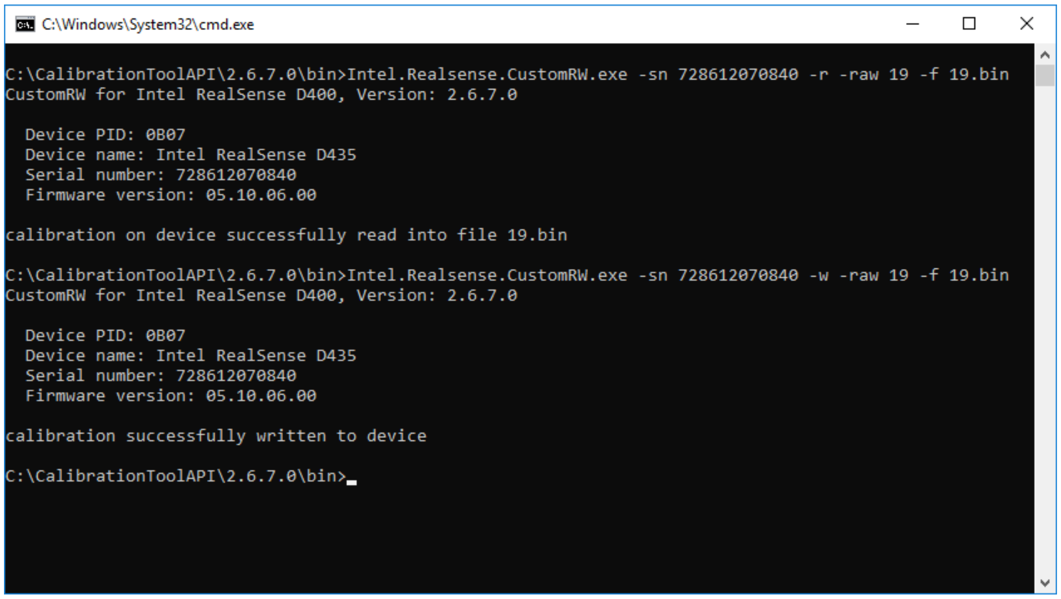
Example: read raw calibration table from device and save into a binary file, supported table id 19, 1F, and 20, for example,

```
Intel.Realsense.CustomRW.exe -r -raw 19 -f mytable-19.bin
```

Example: write raw calibration table from binary file into the device, supported table id 19, 1F, and 20, for example,

```
Intel.Realsense.CustomRW.exe -w -raw 19 -f mytable-19.bin
```

Figure 6-5 Read and Write Calibration RAW Data on Selected Device



```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -r -raw 19 -f 19.bin
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

calibration on device successfully read into file 19.bin

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -w -raw 19 -f 19.bin
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0B07
  Device name: Intel RealSense D435
  Serial number: 728612070840
  Firmware version: 05.10.06.00

calibration successfully written to device

C:\CalibrationToolAPI\2.6.7.0\bin>
```

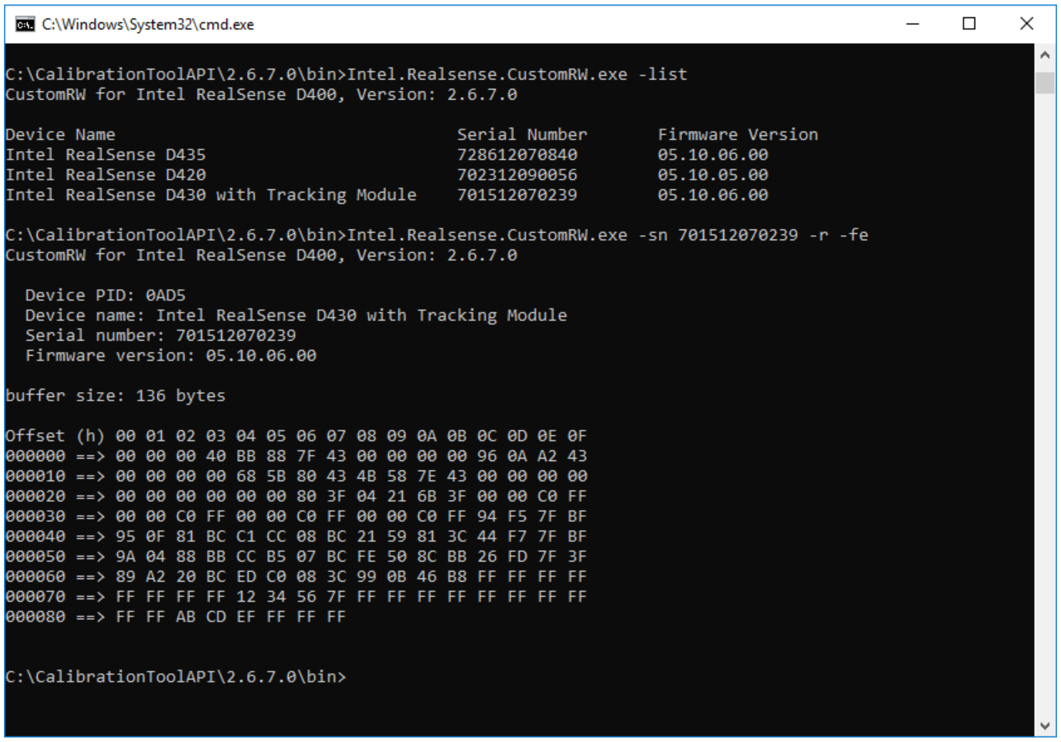
6.5 Read and Write Custom Fisheye Data

The -fe option supports read/write the custom fisheye data on selected device SKU. It only works on very limited range of devices. Use this option ONLY when the device is supported.

Example #8: read custom fisheye data from device and display to the terminal

```
Intel.Realsense.CustomRW.exe -r -fe
```

Figure 6-6 Read Fisheye Custom Data on Selected Device and Display on Screen



C:\Windows\System32\cmd.exe

```
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -list
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device Name                                 Serial Number       Firmware Version
Intel RealSense D435                        728612070840        05.10.06.00
Intel RealSense D420                        702312090056        05.10.05.00
Intel RealSense D430 with Tracking Module   701512070239        05.10.06.00

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 701512070239 -r -fe
CustomRW for Intel RealSense D400, Version: 2.6.7.0

  Device PID: 0AD5
  Device name: Intel RealSense D430 with Tracking Module
  Serial number: 701512070239
  Firmware version: 05.10.06.00

buffer size: 136 bytes

Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000 ==> 00 00 00 40 BB 88 7F 43 00 00 00 00 96 0A A2 43
000010 ==> 00 00 00 00 68 5B 80 43 4B 58 7E 43 00 00 00 00
000020 ==> 00 00 00 00 00 00 80 3F 04 21 6B 3F 00 00 C0 FF
000030 ==> 00 00 C0 FF 00 00 C0 FF 00 00 C0 FF 94 F5 7F BF
000040 ==> 95 0F 81 BC C1 CC 08 BC 21 59 81 3C 44 F7 7F BF
000050 ==> 9A 04 88 BB CC B5 07 BC FE 50 8C BB 26 FD 7F 3F
000060 ==> 89 A2 20 BC ED C0 08 3C 99 0B 46 B8 FF FF FF FF
000070 ==> FF FF FF FF 12 34 56 7F FF FF FF FF FF FF FF FF
000080 ==> FF FF AB CD EF FF FF FF


C:\CalibrationToolAPI\2.6.7.0\bin>
```

Example: read custom fisheye data from device and save into a binary file
Intel.Realsense.CustomRW.exe -r -fe -f myfe.bin

Example: write custom fisheye data from binary file into the device
Intel.Realsense.CustomRW.exe -w -fe -f myfe.bin

7 Developing Custom Apps with Calibration API

A programmable interface is provided for developing custom calibration apps. A compressed package of the interface and examples is included in the tool installation.

7.1 API Package Location

On **Linux**:

```
/usr
  |
  |----- share
  |         |---- doc
  |                 |..... librscalibrationtool
  |                          |
  |                          |---- api
  |                                 |---- DynamicCalibrationAPI-Linux-2.11.0.0.tar.gz
```

On **Windows**:

```
C:\Program Files\Intel\CalibrationToolAPI
                            |-------- 2.11.0.0
                            |            |
                            |            |-------- api
                            |            |            |-------- DynamicCalibrationAPI.Zip
```

7.2 API Package Structure

Copy DynamicCalibrationAPI-Linux-2.11.0.0.tar.gz or DynamicCalibrationAPI.Zip to your working folder and unzip it into similar directory structure as below:

```
DynamicCalibrationAPI
          |
          |---- 2.11.0.0
          |         |---- CMakeLists.txt
          |         |
          |         |---- lib
          |         |       |---- libDSDynamicCalibrationAPI.so or DSDynamicCalibrationAPI.dll
          |         |
          |         |...... examples
          |         |           |---- DynamicCalibrator
          |         |           |---- CustomRW
          |         |           |---- CustomCalibration
          |         |           |---- rs-calibration-converter
          |         |           |---- simple-rw
          |         |           |---- simple-fe
          |         |
          |         |----Include
          |         |      |---- DSDynamicCalibration.h
          |         |      |---- DSOSUtils.h
          |         |      |---- DSShared.h
          |         |      |---- DSCalData.h
          |         |
          |         |...... attributions.txt
          |         |...... License.txt
          |         |...... README.md
          |         |...... target
          |         |---- librealsense
          |         |---- 3rdparty
```

Notes:

- API - libDSDynamicCalibrationAPI.so or DSDynamicCalibrationAPI.DLL is the calibration API library and DSDynamicCalibration.h is its main header file
- Under examples folder
  - ✓ DynamicCalibrator contains the source code for the Dynamic Calibrator
  - ✓ CustomRW contains the source code for the CustomRW tool
  - ✓ CustomCalibration contains sources for an example custom calibration of depth and RGB sensors with non-Intel algorithms, the example is described in detail in the *Intel® RealSense™ Product Family D400 Series Custom Calibration* white paper

7.3 Compiling Examples

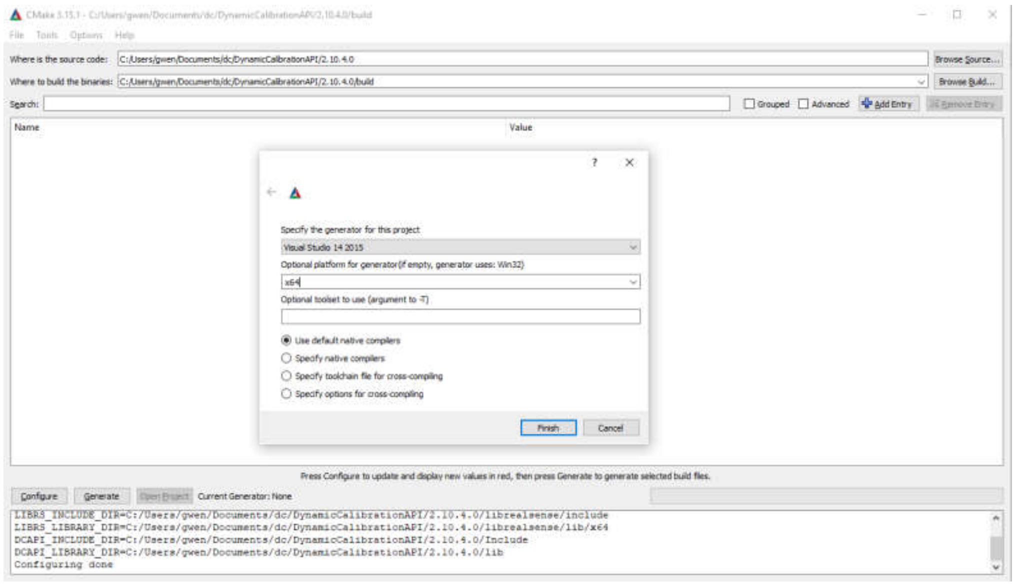
The API package support cmake.

On **Linux**:

```
cd DynamicCalibrationAPI/2.11.0.0
mkdir build
cd build
cmake ..
        -- The C compiler identification is GNU 5.4.0
        -- The CXX compiler identification is GNU 5.4.0
        -- Check for working C compiler: /usr/bin/cc
        -- Check for working C compiler: /usr/bin/cc -- works
        -- Detecting C compiler ABI info
        -- Detecting C compiler ABI info - done
        -- Detecting C compile features
        -- Detecting C compile features - done
        -- Check for working CXX compiler: /usr/bin/c++
        -- Check for working CXX compiler: /usr/bin/c++ -- works
        -- Detecting CXX compiler ABI info
        -- Detecting CXX compiler ABI info - done
        -- Detecting CXX compile features
        -- Detecting CXX compile features - done
        -- CMAKE_SYSTEM_PROCCESSOR=x86_64
        -- BUILD_SHARED_LIBS=ON
        --
        LIBRS_INCLUDE_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.11.0.0/librealsense/i
        nclude
        --
        LIBRS_LIBRARY_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.11.0.0/librealsense/li
        b/linux/x86_64
        -- DCAPI_INCLUDE_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.11.0.0/Include
        -- DCAPI_LIBRARY_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.11.0.0/lib
        -- Configuring done
        -- Generating done
        -- Build files have been written to:
        /home/gwen/dc/test/DynamicCalibrationAPI/2.10.4.0/build
make -j8
```

On **Windows**:

Config cmake for compilaing CalibrationToolAPI\2.11.0.0 and generate project files for Visual Studio 2015 and x64 platform.



Where is the source code: C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0

Where to build the binaries: C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0/build

Specify the generator for this project: Visual Studio 14 2015

Optional platform for generator(if empty, generator uses: Win32): x64

Optional toolset to use (argument to -T):

[✓ Use default native compilers]
Specify native compilers
Specify toolchain file for cross-compiling
Specify options for cross-compiling

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

```
LIBRS_INCLUDE_DIR=C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0/librealsense/include
LIBRS_LIBRARY_DIR=C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0/librealsense/lib/x64
DCAPI_INCLUDE_DIR=C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0/Include
DCAPI_LIBRARY_DIR=C:/Users/gwen/Documents/dc/DynamicCalibrationAPI/2.10.4.0/lib
Configuring done
```

Open rs-dynamic-calibration-api-samples.sln under build directory to compile.

7.4 API Documentation

For details of the calibration API and examples, please refer to the ***Intel® RealSense™ Product Family D400 Series Software Calibration Tool Programmer Guide***.

8 Troubleshooting

Table 8-1 Troubleshooting

Issue	Resolution

With targeted calibration on Linux, after rectification phase is successfully completed, sample app windows may become black	The most like cause is the kernel not improperly patched or os booted into the non-patched kernel. Please follow instructions to redo the patching in installation section Video4Linux backend.
	
	By default, Ubuntu auto kernel update feature is enabled. So you may experience the same screen blacking issue again if your Ubuntu kernel is automatically updated.
	
	To avoid this, you can choose to do one of the following:
	
	Method 1 - **Pin/hold your kernel**
	Examples:
	Discover the current latest kernel currently installed. It's the one with the biggest version number.
	
	$ dpkg -l | grep linux-image
	$ uname -r
	4.4.0-79-generic
	$ sudo apt-mark hold linux-image-4.4.0-79-generic
	linux-image-4.4.0-79-generic set on hold.
	$ sudo apt-mark hold linux-headers-4.4.0-79-generic
	$ sudo apt-mark hold linux-image-extra-4.4.0-79-generic
	linux-image-extra-4.4.0-79-generic set on hold.
	$ dpkg --get-selections | grep linux-image-4.4.0-79-generic
	linux-image-4.4.0-79-generic hold
	
	Method 2 - **Disable automatic updates on Ubuntu**:
	
	Open the Unity Dash
	Search for 'Software & Updates'
	Select the 'Updates' tab.
	Change 'Automatically check for updates' from 'Daily' to 'Never'.
	This setting will stop the system from checking for ANY updates

In certain cases, when the device is badly out of calibration or negative effect of lighting conditions, dynamic calibration may not be able to find the right correction and complete the process.	Try to find a different scene or environment to run dynamic calibration again. If multiple try do not resolve the issue, the device may need to be recalibrated with technician calibration or factory calibration.

Devices calibrated through targeted dynamic calibration shows greater depth error than expected (i.e., more than 2%)	Follow target setup instruction to check the target physical sizes. Target size is critical to calibration accuracy.
With modules with old calibration tables, target calibration does not change calibration on device	A simple workaround is to perform a target-less dynamic calibration on the device first and then targeted calibration would have no issue. Additional calibration would have no such limitation.

9 Appendix A - Calibration Check

This section describes two methods to check the camera calibration. The first method is a quick and easy way to check the alignment of left and right cameras. The second method is to test the accuracy of depth measurement.

The two methods are explained below:

9.1 Quick Check

With the quick check, the user should view the depth image from the camera with a tool that can display depth, Intel® RealSense™ Viewer or Depth Quality Tool in Intel® RealSense™ SDK 2.0.
https://github.com/IntelRealSense/librealsense/releases/latest

Point the camera to a flat surface such as a wall about 1 to 2 meters away (3 to 6 feet). Avoid black surfaces.

Visually inspect the depth image display of the wall. A lot of black dots or holes on the image is an indication of the camera being out of calibration.

Figure 9-1. Calibration Check – Camera in Calibration

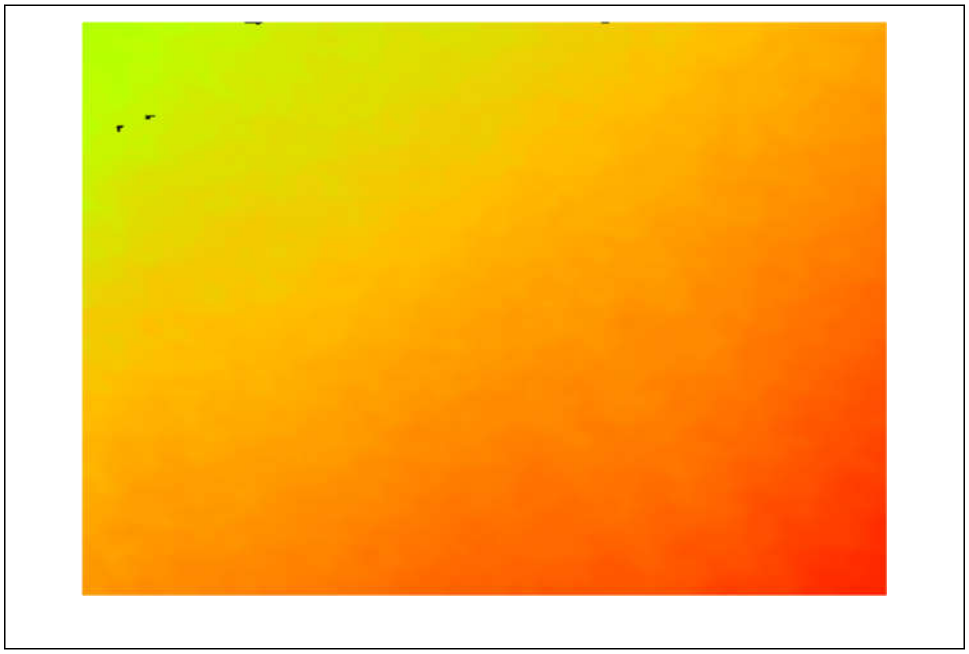
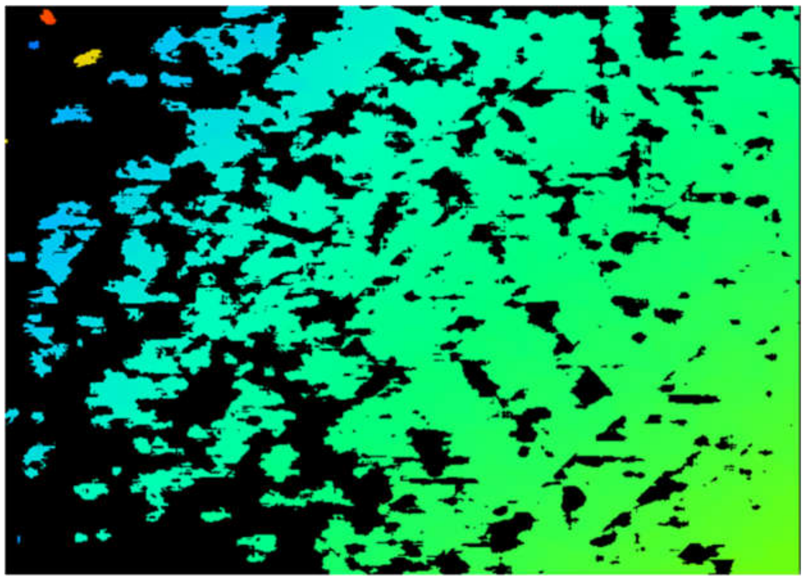


Figure 9-2. Calibration Check – Camera Out of Calibration



9.2 Accuracy Check

This procedure should be used to check the accuracy of the camera. Place the camera in parallel to a flat wall and exactly two meter (2000 mm) away.

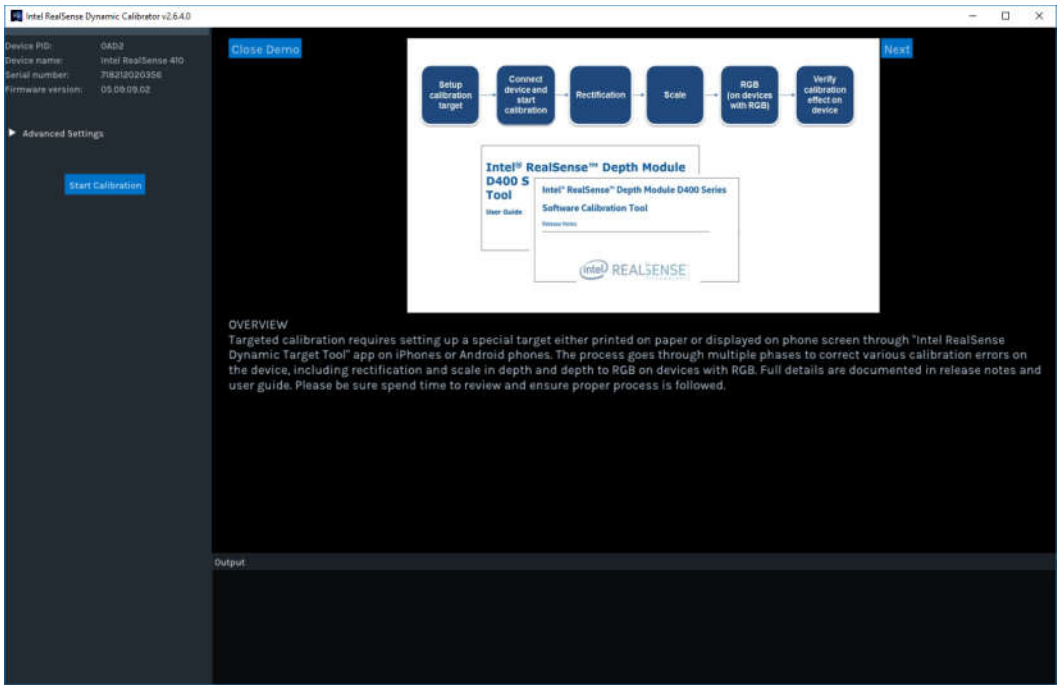
Once the camera is placed in its position, Use Intel® RealSense™Viewer or Depth Quality Tool to measure the absolute distance. For a flat surface at a distance of 2 meter the absolute distance should be within 2% or better at 2 meter (2000mm).

If the distance is not within the defined range, then the camera needs to be calibrated.

10 Appendix B - Calibration Process Highlights

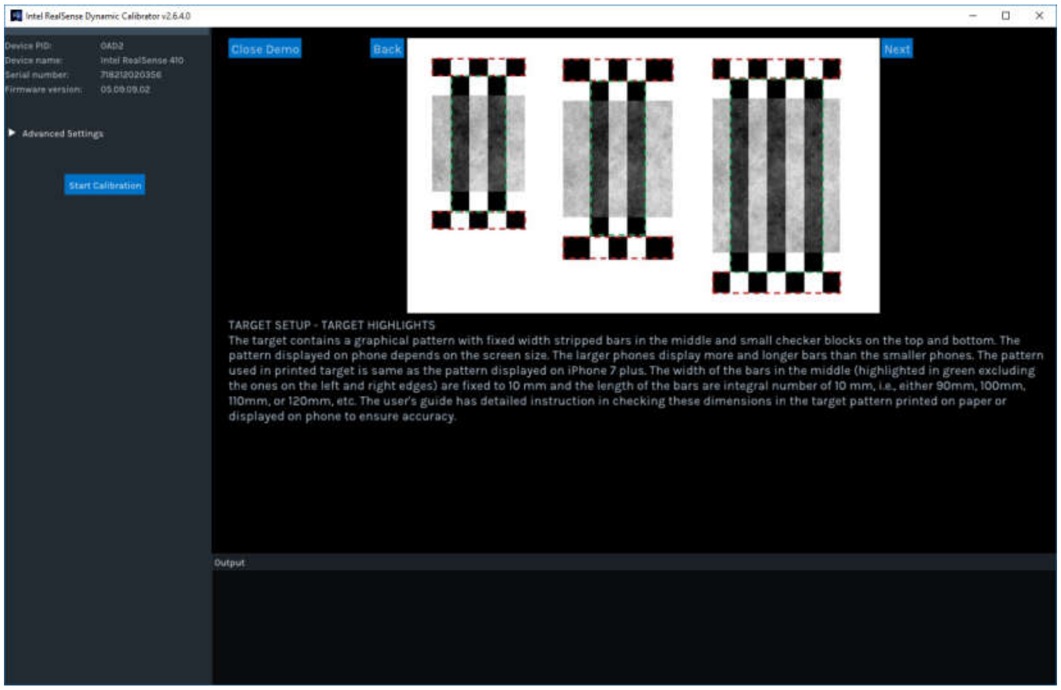
10.1 Targeted Calibration

Targeted calibration requires setting up a special target either printed on paper or displayed on phone screen through "Intel RealSense Dynamic Target Tool" app on iPhones or Android phones. The process goes through multiple phases to correct various calibration errors on the device, including rectification and scale in depth and depth to RGB on devices supporting RGB. Targeted calibration should be used to calibrate all D400 Series devices if a target setup is permitted.


Intel RealSense Dynamic Calibrator v2.6.4.0
Close Demo
Next
Setup calibration target
Connect device and start calibration
Rectification
Scale
RGB (on devices with RGB)
Verify calibration effect on device
Intel® RealSense™ Depth Module D400 S Tool
User Guide
Intel® RealSense™ Depth Module D400 Series Software Calibration Tool
Release Notes
intel REALSENSE
Advanced Settings
Start Calibration
Output

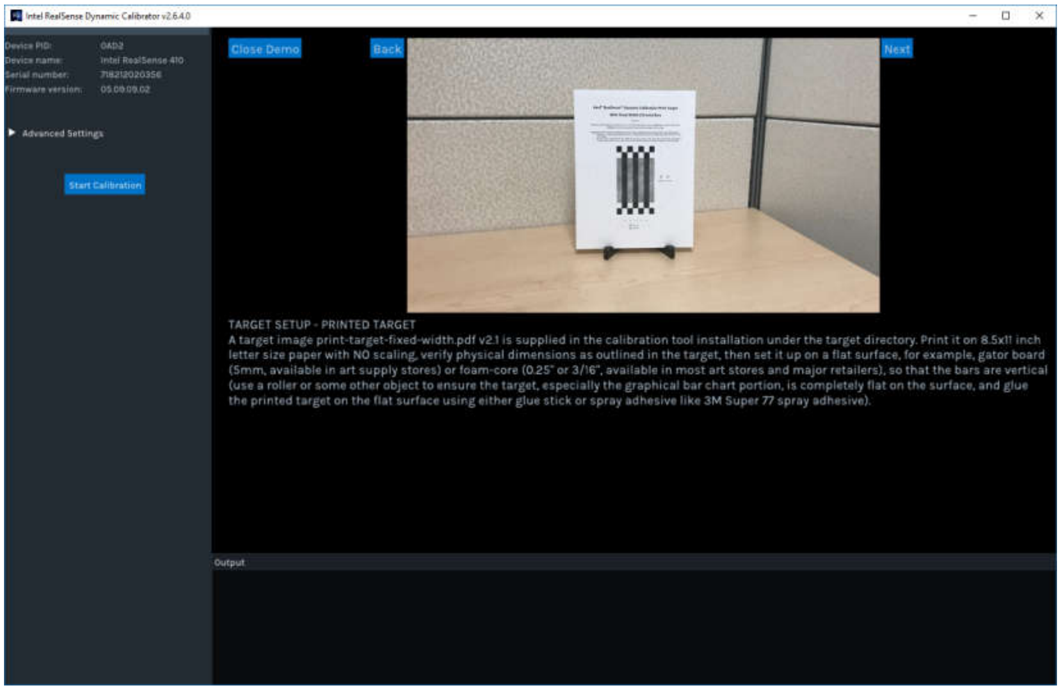
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

OVERVIEW
Targeted calibration requires setting up a special target either printed on paper or displayed on phone screen through "Intel RealSense Dynamic Target Tool" app on iPhones or Android phones. The process goes through multiple phases to correct various calibration errors on the device, including rectification and scale in depth and depth to RGB on devices with RGB. Full details are documented in release notes and user guide. Please be sure spend time to review and ensure proper process is followed.



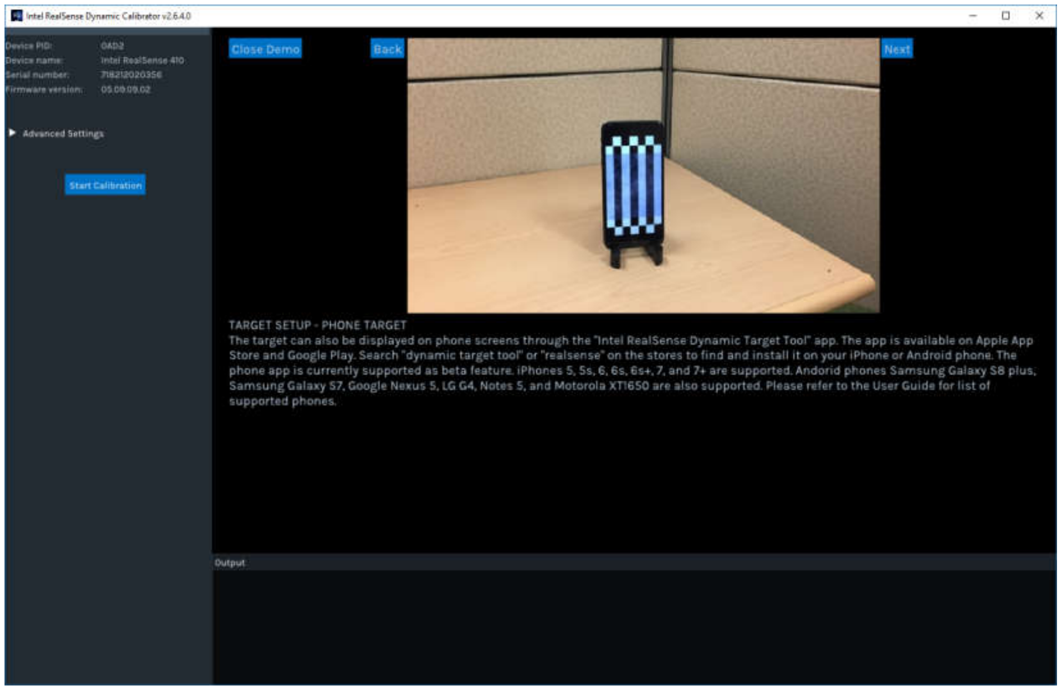
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

TARGET SETUP - TARGET HIGHLIGHTS
The target contains a graphical pattern with fixed width stripped bars in the middle and small checker blocks on the top and bottom. The pattern displayed on phone depends on the screen size. The larger phones display more and longer bars than the smaller phones. The pattern used in printed target is same as the pattern displayed on iPhone 7 plus. The width of the bars in the middle (highlighted in green excluding the ones on the left and right edges) are fixed to 10 mm and the length of the bars are integral number of 10 mm, i.e., either 90mm, 100mm, 110mm, or 120mm, etc. The user's guide has detailed instruction in checking these dimensions in the target pattern printed on paper or displayed on phone to ensure accuracy.



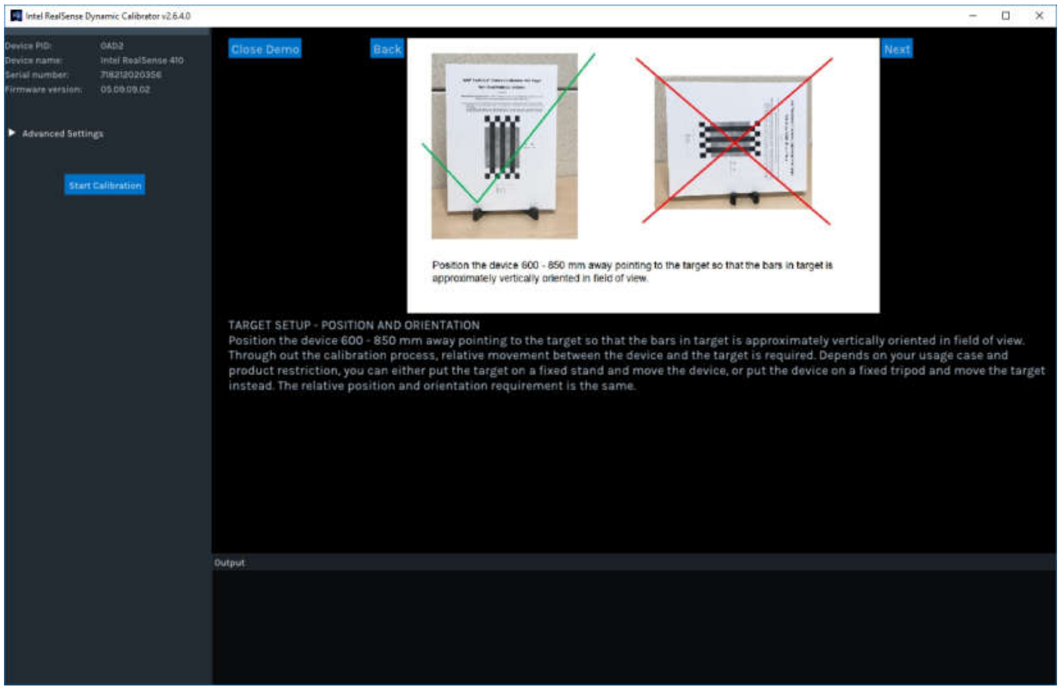
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

TARGET SETUP - PRINTED TARGET
A target image print-target-fixed-width.pdf v2.1 is supplied in the calibration tool installation under the target directory. Print it on 8.5x11 inch letter size paper with NO scaling, verify physical dimensions as outlined in the target, then set it up on a flat surface, for example, gator board (5mm, available in art supply stores) or foam-core (0.25" or 3/16", available in most art stores and major retailers), so that the bars are vertical (use a roller or some other object to ensure the target, especially the graphical bar chart portion, is completely flat on the surface, and glue the printed target on the flat surface using either glue stick or spray adhesive like 3M Super 77 spray adhesive).



Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

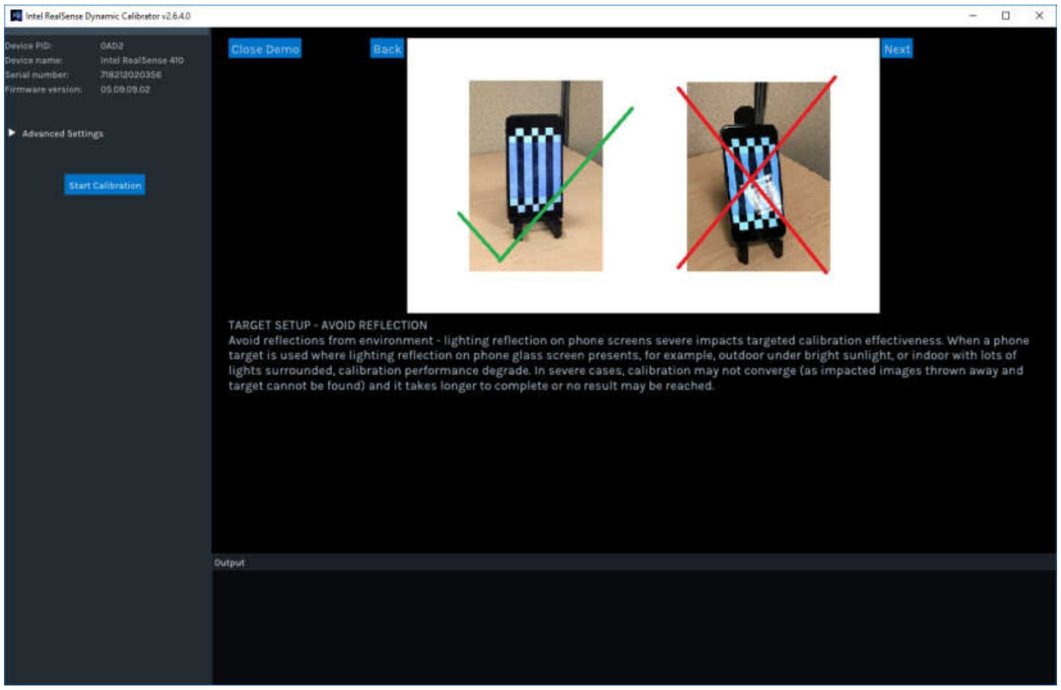
TARGET SETUP - PHONE TARGET
The target can also be displayed on phone screens through the "Intel RealSense Dynamic Target Tool" app. The app is available on Apple App Store and Google Play. Search "dynamic target tool" or "realsense" on the stores to find and install it on your iPhone or Android phone. The phone app is currently supported as beta feature. iPhones 5, 5s, 6, 6s, 6s+, 7, and 7+ are supported. Andorid phones Samsung Galaxy S8 plus, Samsung Galaxy S7, Google Nexus 5, LG G4, Notes 5, and Motorola XT1650 are also supported. Please refer to the User Guide for list of supported phones.



Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

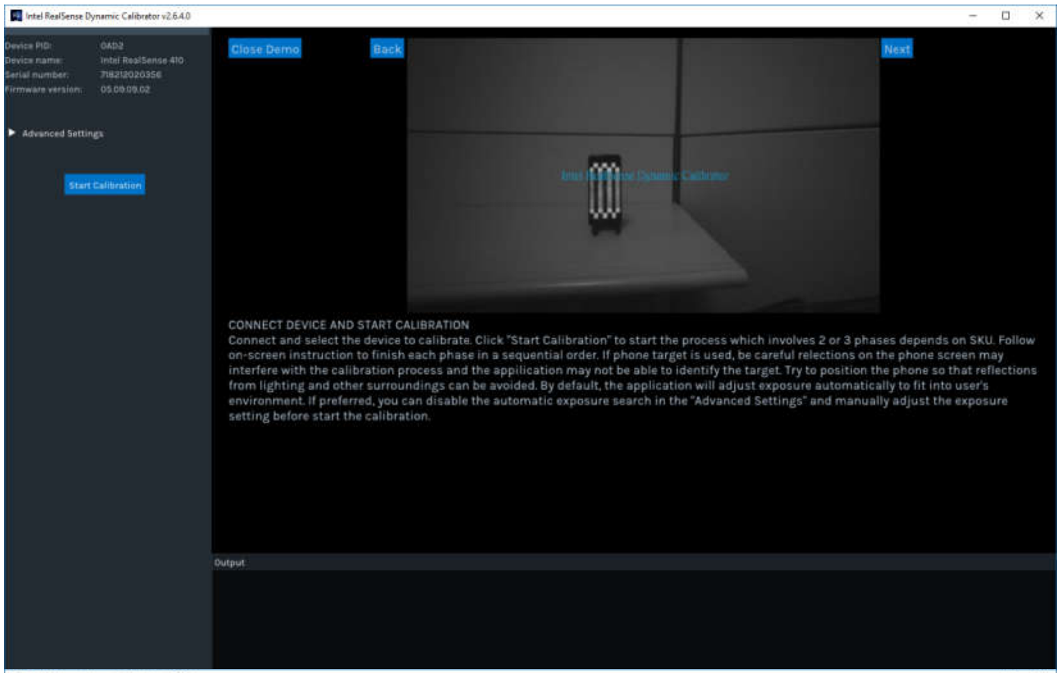
Position the device 600 - 850 mm away pointing to the target so that the bars in target is approximately vertically oriented in field of view.

TARGET SETUP - POSITION AND ORIENTATION
Position the device 600 - 850 mm away pointing to the target so that the bars in target is approximately vertically oriented in field of view. Through out the calibration process, relative movement between the device and the target is required. Depends on your usage case and product restriction, you can either put the target on a fixed stand and move the device, or put the device on a fixed tripod and move the target instead. The relative position and orientation requirement is the same.



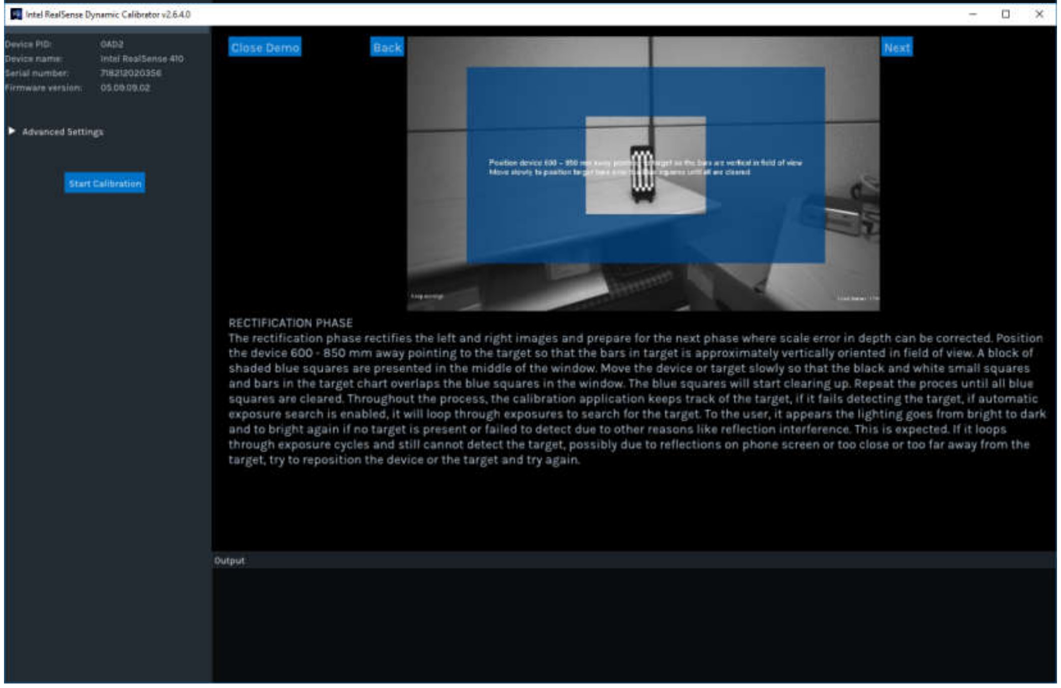
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

TARGET SETUP - AVOID REFLECTION
Avoid reflections from environment - lighting reflection on phone screens severe impacts targeted calibration effectiveness. When a phone target is used where lighting reflection on phone glass screen presents, for example, outdoor under bright sunlight, or indoor with lots of lights surrounded, calibration performance degrade. In severe cases, calibration may not converge (as impacted images thrown away and target cannot be found) and it takes longer to complete or no result may be reached.


Intel RealSense Dynamic Calibrator v2.6.4.0
Close Demo
Back
Next
Advanced Settings
Start Calibration
Intel RealSense Dynamic Calibrator
Output

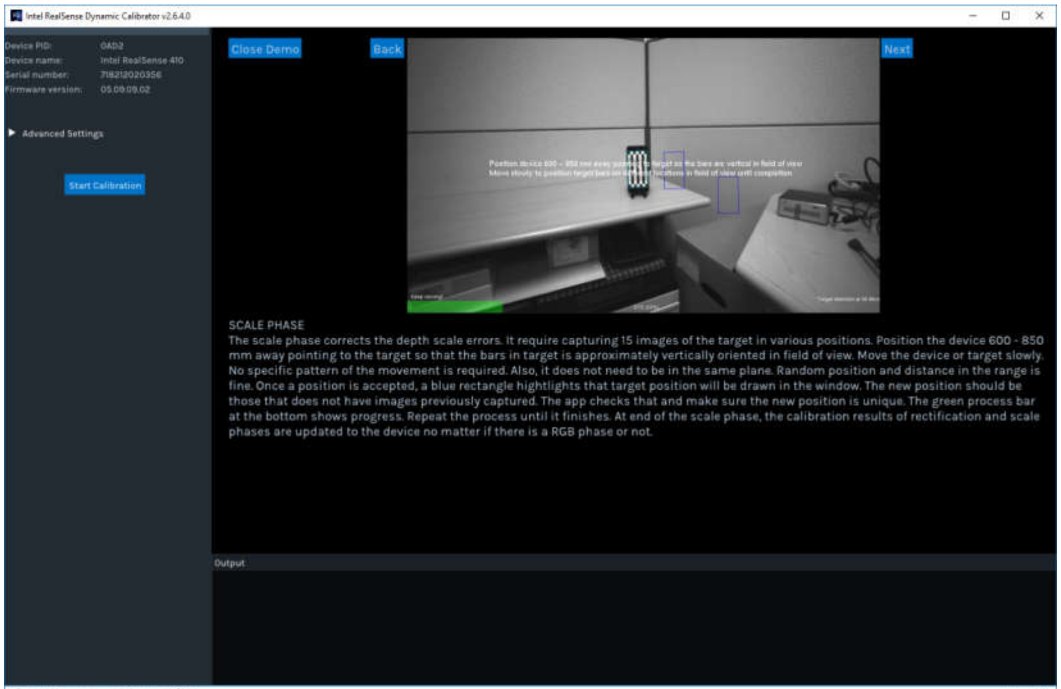
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

CONNECT DEVICE AND START CALIBRATION
Connect and select the device to calibrate. Click "Start Calibration" to start the process which involves 2 or 3 phases depends on SKU. Follow on-screen instruction to finish each phase in a sequential order. If phone target is used, be careful relections on the phone screen may interfere with the calibration process and the appilication may not be able to identify the target. Try to position the phone so that reflections from lighting and other surroundings can be avoided. By default, the application will adjust exposure automatically to fit into user's environment. If preferred, you can disable the automatic exposure search in the "Advanced Settings" and manually adjust the exposure setting before start the calibration.



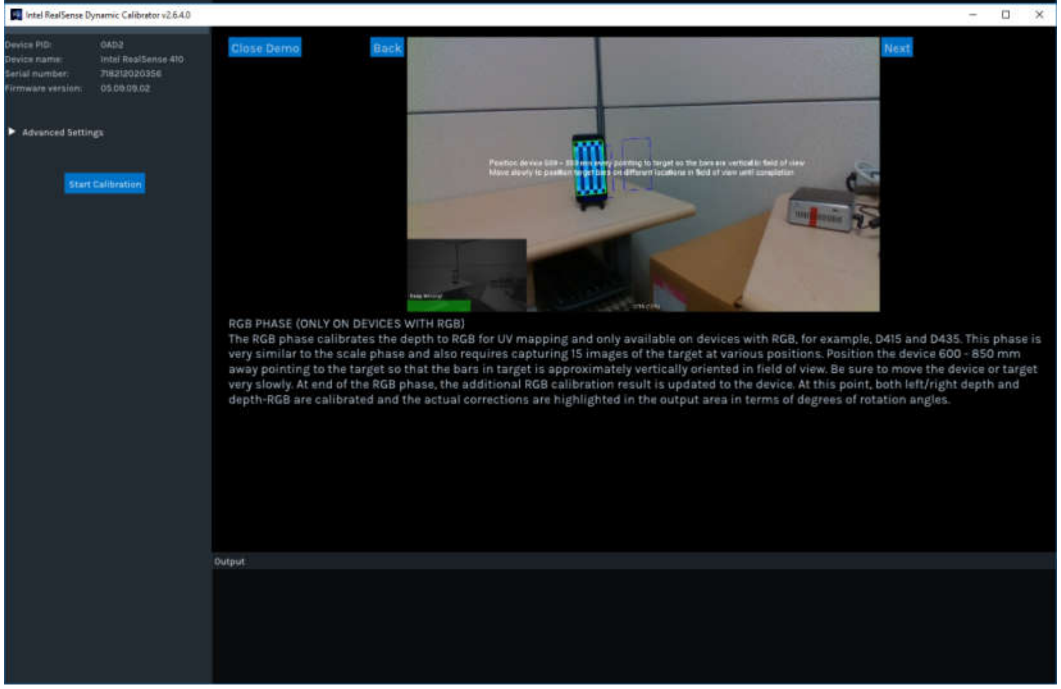
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

RECTIFICATION PHASE
The rectification phase rectifies the left and right images and prepare for the next phase where scale error in depth can be corrected. Position the device 600 - 850 mm away pointing to the target so that the bars in target is approximately vertically oriented in field of view. A block of shaded blue squares are presented in the middle of the window. Move the device or target slowly so that the black and white small squares and bars in the target chart overlaps the blue squares in the window. The blue squares will start clearing up. Repeat the proces until all blue squares are cleared. Throughout the process, the calibration application keeps track of the target, if it fails detecting the target, if automatic exposure search is enabled, it will loop through exposures to search for the target. To the user, it appears the lighting goes from bright to dark and to bright again if no target is present or failed to detect due to other reasons like reflection interference. This is expected. If it loops through exposure cycles and still cannot detect the target, possibly due to reflections on phone screen or too close or too far away from the target, try to reposition the device or the target and try again.



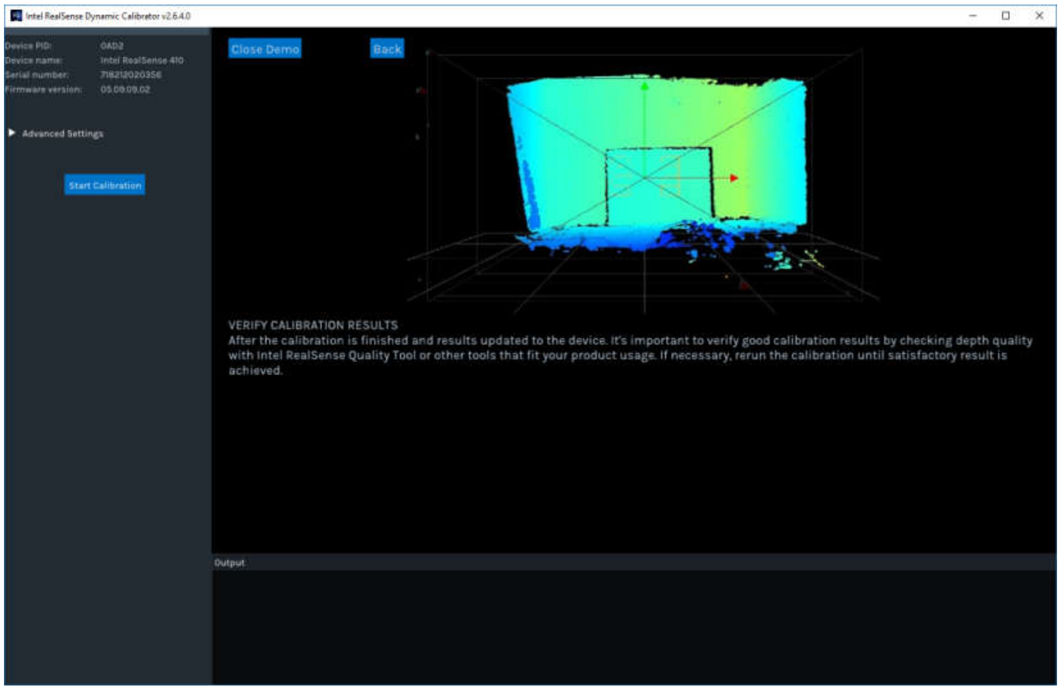
Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

SCALE PHASE
The scale phase corrects the depth scale errors. It require capturing 15 images of the target in various positions. Position the device 600 - 850 mm away pointing to the target so that the bars in target is approximately vertically oriented in field of view. Move the device or target slowly. No specific pattern of the movement is required. Also, it does not need to be in the same plane. Random position and distance in the range is fine. Once a position is accepted, a blue rectangle hightlights that target position will be drawn in the window. The new position should be those that does not have images previously captured. The app checks that and make sure the new position is unique. The green process bar at the bottom shows progress. Repeat the process until it finishes. At end of the scale phase, the calibration results of rectification and scale phases are updated to the device no matter if there is a RGB phase or not.



Device PID: 0AD2
Device name: Intel RealSense 410
Serial number: 718212020356
Firmware version: 05.09.09.02

RGB PHASE (ONLY ON DEVICES WITH RGB)
The RGB phase calibrates the depth to RGB for UV mapping and only available on devices with RGB, for example, D415 and D435. This phase is very similar to the scale phase and also requires capturing 15 images of the target at various positions. Position the device 600 - 850 mm away pointing to the target so that the bars in target is approximately vertically oriented in field of view. Be sure to move the device or target very slowly. At end of the RGB phase, the additional RGB calibration result is updated to the device. At this point, both left/right depth and depth-RGB are calibrated and the actual corrections are highlighted in the output area in terms of degrees of rotation angles.



Device PID:	0AD2
Device name:	Intel RealSense 410
Serial number:	718212020356
Firmware version:	05.09.09.02

VERIFY CALIBRATION RESULTS
After the calibration is finished and results updated to the device. It's important to verify good calibration results by checking depth quality with Intel RealSense Quality Tool or other tools that fit your product usage. If necessary, rerun the calibration until satisfactory result is achieved.

11 Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup

11.1 Introduction

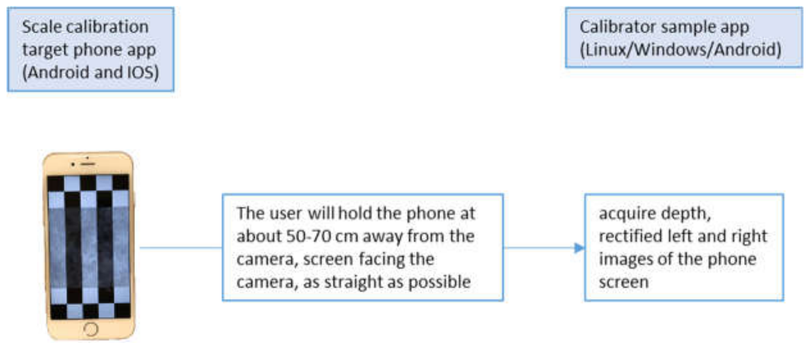
Dynamic calibration is used to re-calibrate the Intel® RealSense™ Product Family D400 Series camera. As part of the process, the algorithm requires a vertical stripped bar image as calibration target. The target image can be printed on 8.5x11 letter size paper or displayed on an Apple or Android phones through the Intel® RealSense™ Dynamic Target Tool phone apps.

This section provides instructions to setup the target for dynamic calibration.

11.2 Dynamic Calibration Flow

The figuration below shows the overall dynamic calibration process flow.

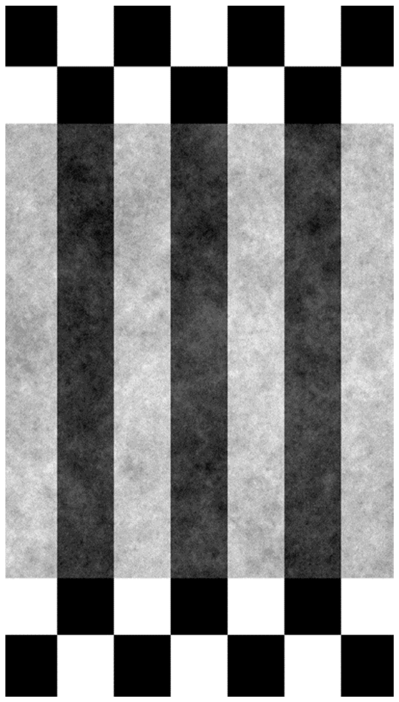
Figure 11-1 dynamic calibration flow


Scale calibration target phone app (Android and IOS)
Calibrator sample app (Linux/Windows/Android)
The user will hold the phone at about 50-70 cm away from the camera, screen facing the camera, as straight as possible
acquire depth, rectified left and right images of the phone screen

1) A target image with multiple vertical bars and black and white blocks is presented on a mobile phone via the Intel® RealSense™ Dynamic Target Tool app or printed target.
    a. Phone target - the target image is dynamically generated on the fly at runtime. The image will be exact match to screen resolution and its content rendered in such that the bar widths are exactly 10 mm (except the most left and right bars on the edges) and the vertical distance between the top black blocks and bottom black blocks also meet certain requirement.
    b. Printed target - The target can be printed on standard 8.5"x11" paper and taped to a flat surface.
2) The target (either a printed, or displayed on phone) is placed about 60-85 cm away in front of the Intel Realsense Camera as straight as possible.
3) The calibration app acquires the target images and perform scale calibration on the Intel Realsense Camera.

An example of the fixed width bar target image is shown below. The actual image depends on scenario – printed, iphone or android phone, and screen sizes.

Figure 11-2 Target Image

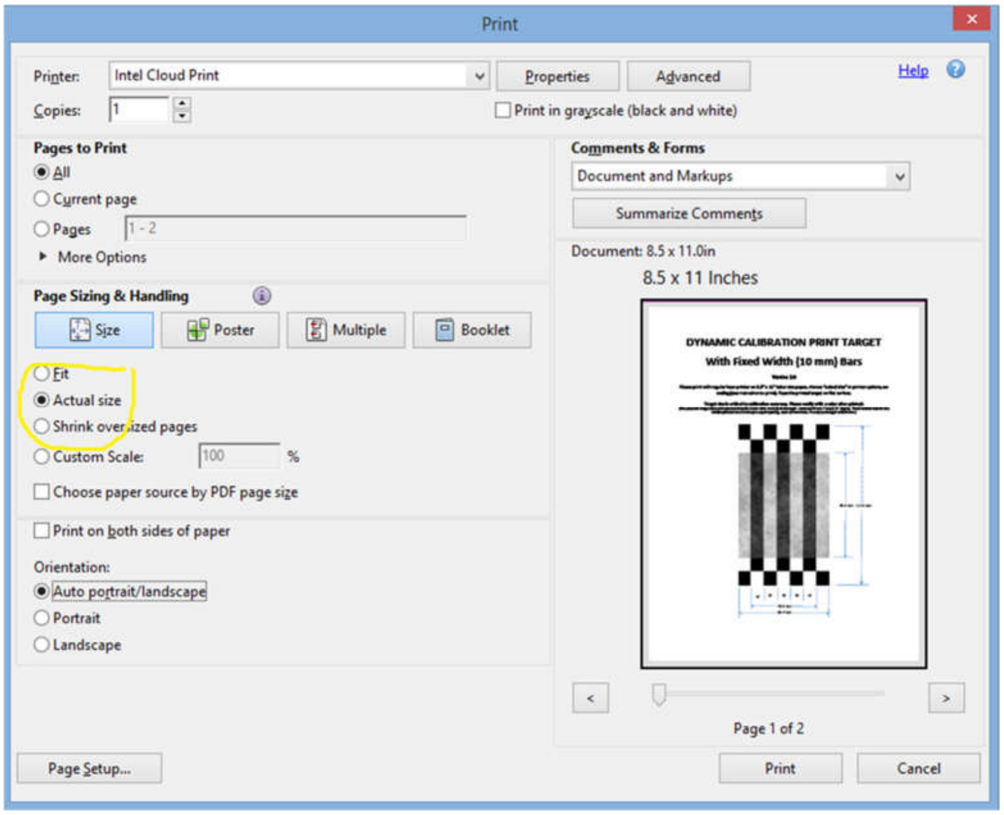


Targets printed on paper or displayed on phone/tablet screens with the Intel® RealSense™ Dynamic Target Tool app on Android and IOS phone devices are supported.

11.3 Printed Target Setup

1. Print target on letter size pager
   a) A print target image v2.1, print-target-fixed-width.pdf, is supplied in the dynamic calibration package under the target directory.
   b) Please print with regular laser printer on 8.5" x 11" letter size paper, choose "**actual size**" in printer options, **no scaling**.

Figure 11-3 Target Print Options



Print

Printer: Intel Cloud Print | Properties | Advanced | Help

Copies: 1 | ☐ Print in grayscale (black and white)

Pages to Print
- [✓ All]
- Current page
- Pages 1 - 2
- More Options

Page Sizing & Handling

[✓ Size] | Poster | Multiple | Booklet

- Fit
- [✓ Actual size]
- Shrink oversized pages
- Custom Scale: 100 %

☐ Choose paper source by PDF page size

☐ Print on both sides of paper

Orientation:
- [✓ Auto portrait/landscape]
- Portrait
- Landscape

Comments & Forms

Document and Markups

Summarize Comments

Document: 8.5 x 11.0in

8.5 x 11 Inches

DYNAMIC CALIBRATION PRINT TARGET

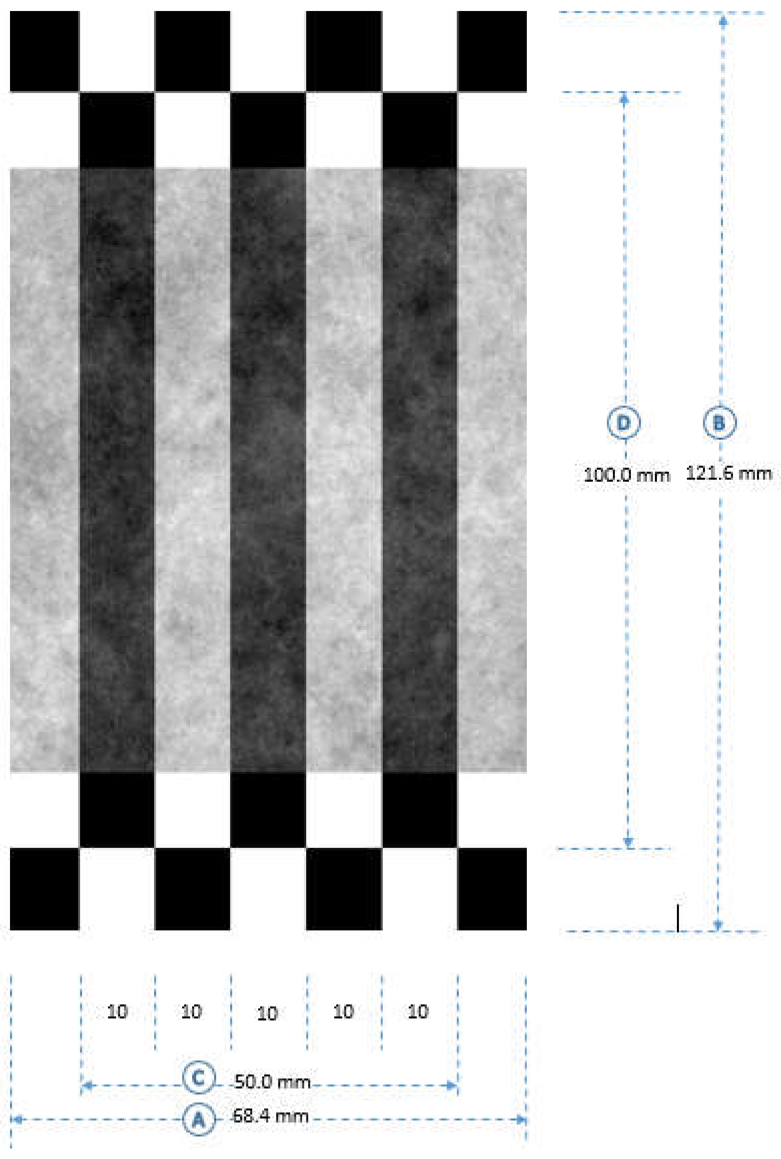
With Fixed Width (10 mm) Bars

Page 1 of 2

Page Setup... | Print | Cancel

c) After the target is printed, please check the physical target image on the paper. The specific dimensions are marked around the image.

Figure 11-4 Target Verification

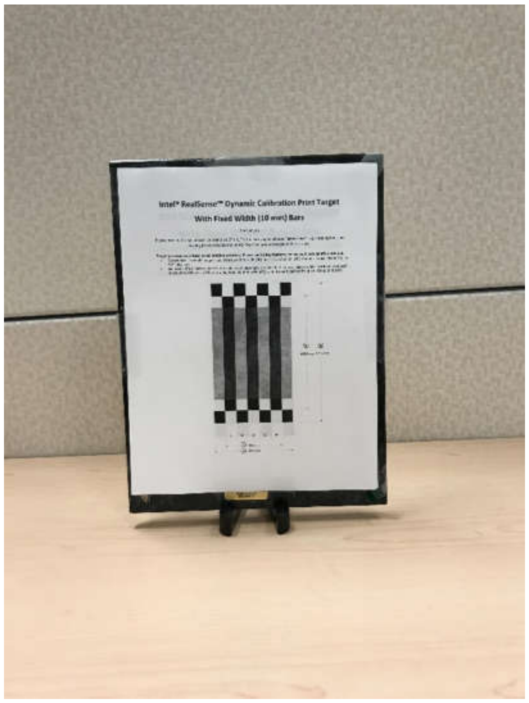


d) **Target precision is critical to calibration accuracy. Please verify key features below with a ruler after printed**.

- Target size - overall image should be 68.4 mm wide (A) and 121.6 mm tall (B) (same size as iPhone 7 plus 5.5" display)
- Bar size - the 5 vertical bars in the middle (3 black and 2 white, in the order, black-white-black-white-black) in equal spacing each 10 mm wide, total 50 mm wide (C), and the vertical bar length is 100.0 mm (D)

2. Setup the printed target on a flat surface so that the **bars are vertical** (use a roller or some other object, to ensure the target is flat on the surface. Tape the printed target on the flat surface after confirming that **the target is flat**).

Figure 11-5 Printed target setup



3. Place the camera about 60 – 85 cm away and point to the target. It should not be too close or too far. Due to various phone sizes, camera device models, and lighting conditions, exact distance cannot be specified. User will need to move the camera in the specified range to find a distance that works best. In most cases, a distance of around 70 cm is sufficient.

4. On the host, start Intel RealSense Dynamic Calibrator app. Refer to section **Calibration with Dynamic Calibrator App** in this document for details**.**

   On Linux
   /usr/bin/Intel.Realsense.DynamicCalibrator

   On Windows
   Double click the "Dynamic Calibrator" icon on Windows desktop to start the calibrator app.

11.4 Phone Target Setup on Apple iPhones Devices

For Apple iPhone devices, Intel® RealSense™ Dynamic Target Tool app is available on Apple App Store for iPhones with IOS 10.0 and later. The latest IOS version tested is 11.2.

- App full name: **Intel® RealSense™ Dynamic Target Tool**
- App short name: **Dynamic Target Tool**

Supported devices include iPhone 5, 5s, 6, 6s, 6s+, 7, 7+, 8, 8+, and iPhone X. Please refer to appendix **Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App** for details.

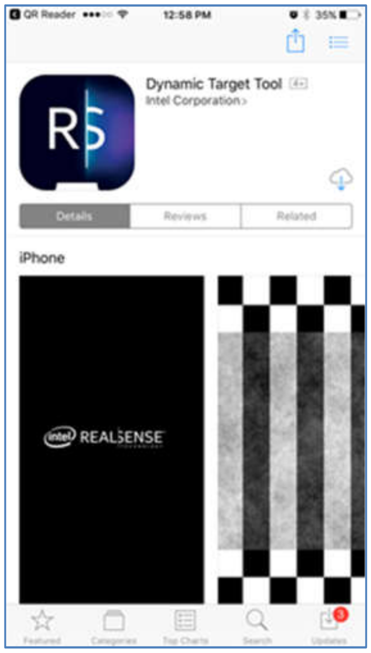
11.4.1 Locate the app on Apple App Store

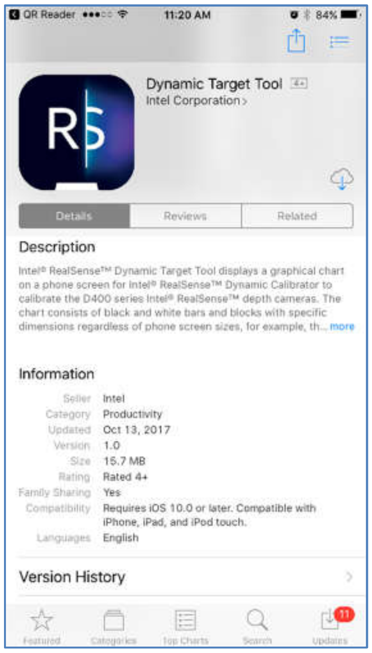
A few simple ways to find the app on Apple App Store:

- ✓ Search App Store with keywords: "**dynamic target tool**" or "**realsense**"
- ✓ Direct download links (for users who want to download/install with a browser): App Store – https://itunes.apple.com/app/dynamic-target-tool/id1291448596
- ✓ Scan QR code for downloading from App Store:

11.4.2 Install the App on iPhone

Once the app is found on the app store, the following informational pages display





Description

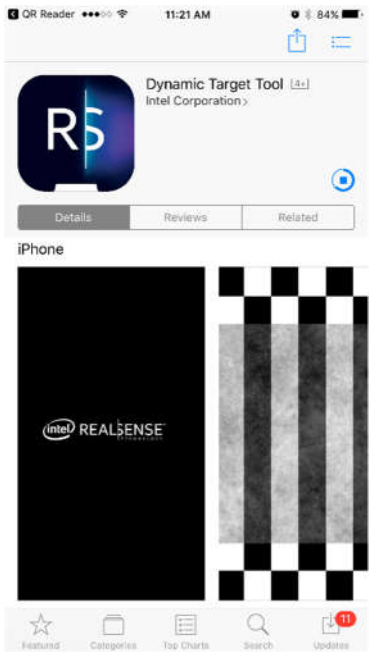
Intel® RealSense™ Dynamic Target Tool displays a graphical chart on a phone screen for Intel® RealSense™ Dynamic Calibrator to calibrate the D400 series Intel® RealSense™ depth cameras. The chart consists of black and white bars and blocks with specific dimensions regardless of phone screen sizes, for example, th... more

Information

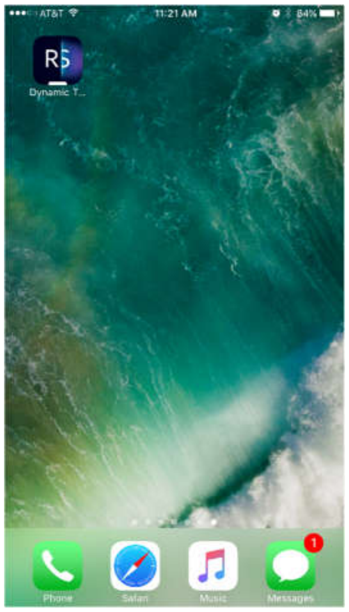
Seller	Intel
Category	Productivity
Updated	Oct 13, 2017
Version	1.0
Size	15.7 MB
Rating	Rated 4+
Family Sharing	Yes
Compatibility	Requires iOS 10.0 or later. Compatible with iPhone, iPad, and iPod touch.
Languages	English

Version History

Press the download/install icon to install the app on your phone.

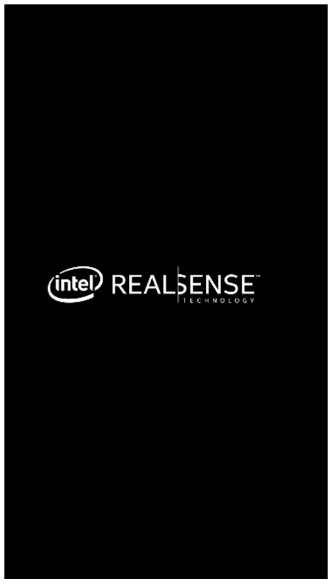


App installed and the Dynamic Target Tool icon appears on phone screen.



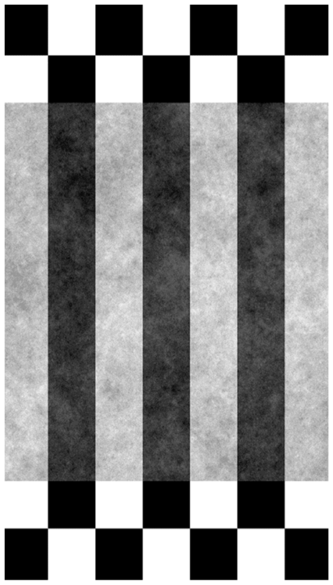
11.4.3 Running the app on phone

Press the app icon to start Intel® RealSense™ Dynamic Target Tool, a brief splash screen with Intel® RealSense™ logo should appear:

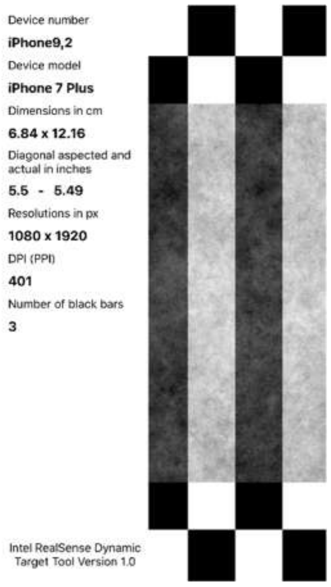


After a brief moment (~ 1 second), stripped bar target should display. This is the target chart for dynamic calibration. The bars are showed vertically. The width of the black bars should be equal and always 10 mm.

The number of bars can be varying depends on size of the device screen. Please refer to section “Target Image Accuracy Check” in appendix “Intel® RealSense™ Dynamic Target Tool Phone App Technical Specification” for details.



There is hidden information page that can show the device display details. This is not normally needed, but a good information in case user need to verify. While in the Intel® RealSense™ Dynamic Target Tool app, swipe from left to right, or touch on the left edge will display this information page. Swipe again from right to left or touch again will dismiss the information page and return the full screen to target.

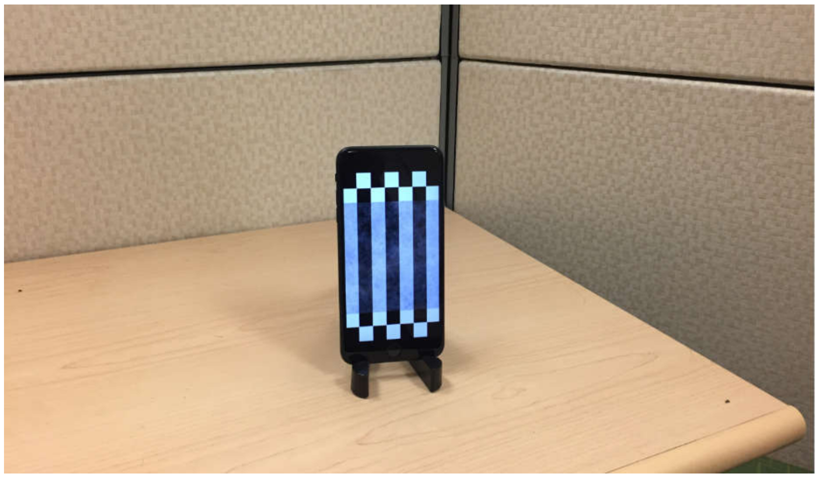


Device number
iPhone9,2
Device model
iPhone 7 Plus
Dimensions in cm
6.84 x 12.16
Diagonal aspected and actual in inches
5.5 - 5.49
Resolutions in px
1080 x 1920
DPI (PPI)
401
Number of black bars
3

Intel RealSense Dynamic Target Tool Version 1.0

11.4.4 Perform Dynamic Calibration

Place the camera about 60 – 85 cm away and point to the phone target. It should not be too close or too far. Due to various phone sizes, camera device models, and lighting conditions, exact distance cannot be specified. User will need to move the camera in the specified range to find a distance that works best. In most cases, a distance of around 70 cm is sufficient.



On the host, start Intel RealSense Dynamic Calibrator app. Refer to section **Calibration with Dynamic Calibrator App** in this document for details**.**

On Linux

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

On Windows

Double click the “Dynamic Calibrator” icon on Windows desktop to start the calibrator app.

11.5 Phone Target Setup on Android Phone Devices

For Android phone devices, Intel® RealSense™ Dynamic Target Tool app is also available on Google Play Store for phones with Android 4.4 and later.

- App full name: **Intel® RealSense™ Dynamic Target Tool**
- App short name: **Dynamic Target Tool**

Tested devices include Samsung Galaxy S8 plus, Samsung Galaxy S7, Google Nexus 5, LG G4, Notes 5, and Motorola XT1650. Please refer to appendix **Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App** for details.

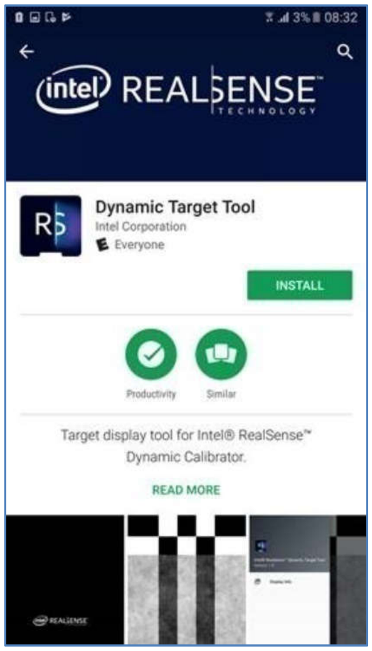
11.5.1 Locate the app on Google Play Store

A few simple ways to find the app on Google Play Store:

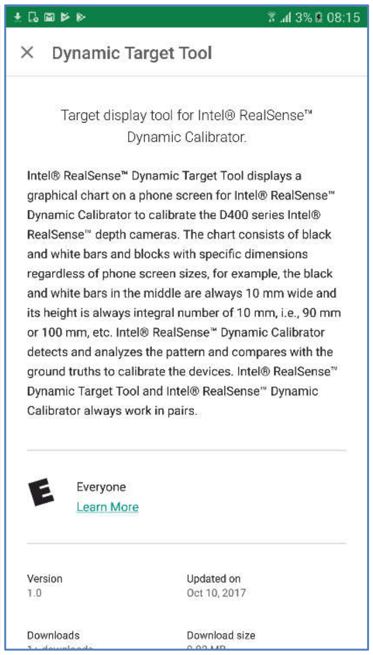
- ✓ Search Google Play with keywords: "**dynamic target tool**" or "**realsense**"
- ✓ Direct download links (for users who want to download/install with a browser): Google Play – https://play.google.com/store/apps/details?id=com.intel.realsenseviewer17613
- ✓ Scan QR code for downloading from Google Play:

11.5.2 Install the App on Android Phone

Once the app is found on the app store, the following informational pages display



Target display tool for Intel® RealSense™ Dynamic Calibrator.

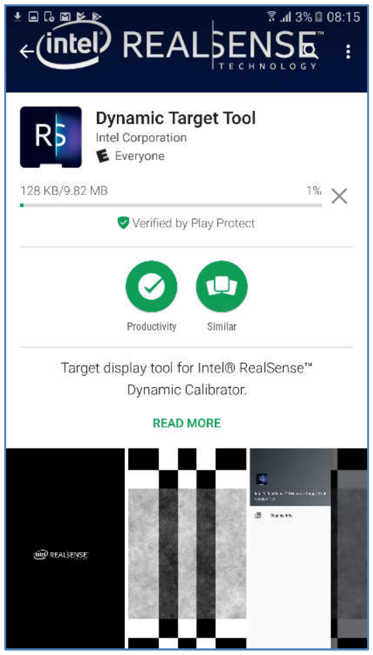


Target display tool for Intel® RealSense™ Dynamic Calibrator.

Intel® RealSense™ Dynamic Target Tool displays a graphical chart on a phone screen for Intel® RealSense™ Dynamic Calibrator to calibrate the D400 series Intel® RealSense™ depth cameras. The chart consists of black and white bars and blocks with specific dimensions regardless of phone screen sizes, for example, the black and white bars in the middle are always 10 mm wide and its height is always integral number of 10 mm, i.e., 90 mm or 100 mm, etc. Intel® RealSense™ Dynamic Calibrator detects and analyzes the pattern and compares with the ground truths to calibrate the devices. Intel® RealSense™ Dynamic Target Tool and Intel® RealSense™ Dynamic Calibrator always work in pairs.

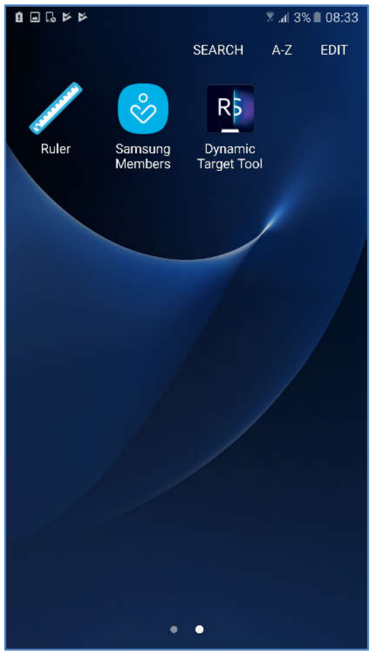
Version	Updated on
1.0	Oct 10, 2017
Downloads	Download size

Press the INSTALL button to install the app on your phone.



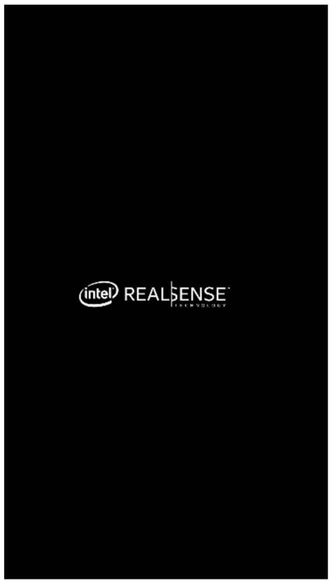
Target display tool for Intel® RealSense™ Dynamic Calibrator.

App installed and the Dynamic Target Tool icon appears on phone screen:

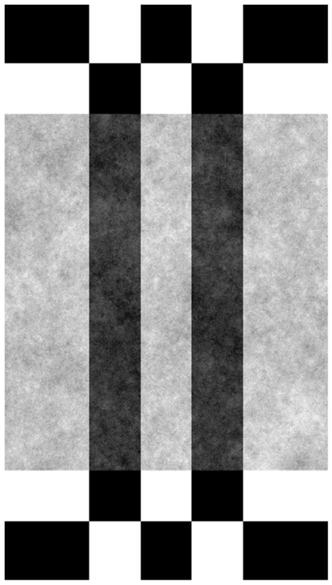


11.5.3 Running the app on phone

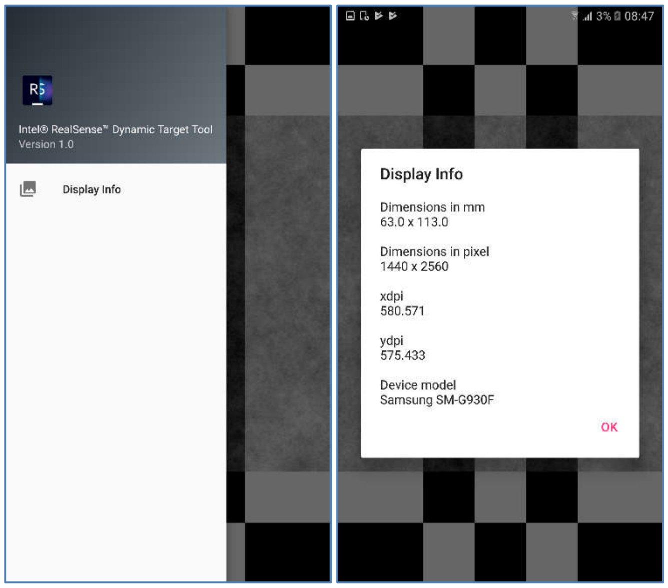
Press the app icon to start Intel® RealSense™ Dynamic Target Tool, a brief splash screen with Intel® RealSense™ logo should appear:



After a brief moment (~ 1 second), stripped bar target should display. This is the target chart for dynamic calibration. The bars are showed vertically. The width of the black bars should be equal and always 10 mm. ***The number of bars can be varying depends on size of the device screen***. Please refer to section "**Target Image Accuracy Check**" in appendix "**Intel® RealSense™ Dynamic Target Tool Phone App Technical Specification**" for details.



There is hidden information page that can show the device display details. This is not normally needed, but a good information in case user need to verify. While in the Intel® RealSense™ Dynamic Target Tool app, swipe from left to right, or touch on the left edge will display this information page. Swipe again from right to left or touch again will dismiss the information page and return the full screen to target.



Display Info

Dimensions in mm
63.0 x 113.0

Dimensions in pixel
1440 x 2560

xdpi
580.571

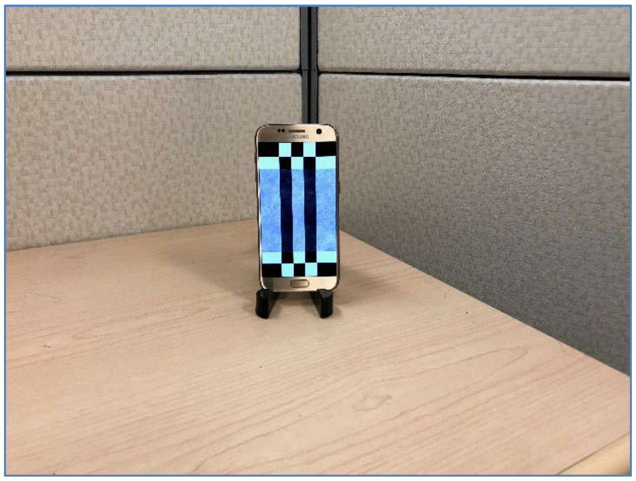
ydpi
575.433

Device model
Samsung SM-G930F

OK

11.5.4 Perform Dynamic Calibration

Place the camera about 60 – 85 cm away and point to the phone target. It should not be too close or too far. Due to various phone sizes, camera device models, and lighting conditions, exact distance cannot be specified. User will need to move the camera in the specified range to find a distance that works best. In most cases, a distance of around 70 cm is sufficient.



On the host, start Intel RealSense Dynamic Calibrator app. Refer to section **Calibration with Dynamic Calibrator App** in this document for details**.**

On Linux

/usr/bin/Intel.Realsense.DynamicCalibrator

On Windows

Double click the "Dynamic Calibrator" icon on Windows desktop to start the calibrator app.

12 Appendix D - Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App

The Intel® RealSense™ Dynamic Target Tool phone app is validated on a limited number of mobile devices.

The phone calibration portion of this software is currently at Beta and may contain a number of known or unknown issues. It should be used for early preview or demonstration of the calibration method only. Please report issues found to the RealSense development team via https://realsense.intel.com/community/

12.1 Validated IOS Devices

On Apple iPhones, display resolution and pixel density is limited to a few combinations, so although validation is done only on limited number of devices, it covers all display combinations in existing Apple iPhones.

Apple Device Displays	Resolution (pixel)	DPI (PPI)	Screen Physical Dimension (cm) Width x Height	Devices with this display	Devices tested formally in dev and QA	Devices tested ad hoc by users
4"	640 x 1136	326	4.99 x 8.85	iPhone 5, 5s, 5c, SE	iPhone 5s	iPhone 5
4.7"	750 x 1334	326	5.88 x 10.46	iPhone 6, 6s, 7, 8	iPhone 6	iPhone 6s, 7
5.5"	1080 x 1920	401	6.84 x 12.16	iPhone 6 plus, 6s plus, 7 plus, 8 plus	iPhone 6 plus, 7 plus	iPhone 6s plus
5.8"	1125 x 2436	458	6.24 x 13.51	iPhone X	iPhone X	

12.2 Validated AOS devices

Display resolution and pixel density varies from device and vendors on Android devices. The target image is dynamically generated to fit to the display on each device. The image generation algorithm is precise in case the display resolution and pixel density is accurate. However, there are known cases where the OS reports wrong pixel density that would impact target accuracy. It's important to physically check the target sizes in order to identify such issue early on.

The following android devices were validated during development and proved to work with Intel® RealSense™ Dynamic Calibrator.

Device tested formally in dev and QA	OS
Samsung Galaxy S7	6.0.1 (Marshmallow)

Google Nexus 5	4.4.2 (KitKat)
LG G4	6.0 (Marshmallow)
Galaxy Note 5	6.0 and 7.0

The following android devices were tested ad hoc by users and reported to work with Intel® RealSense™ Dynamic Calibrator.

Devices tested ad hoc by users
Google Nexus 6
Galaxy S7 Edge
Samsung Galaxy S8 plus

12.3 Devices with accuracy issue in phone target

The following table lists devices that are known to generate inaccurate targets. Please avoid to use these devices:

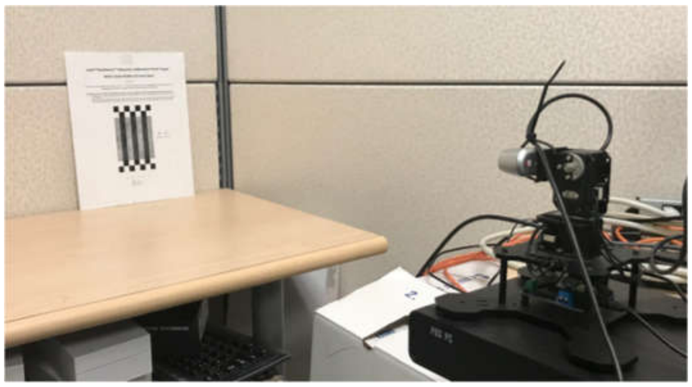
Device with target accuracy issue	OS	Notes
Samsung Galaxy S5	5.1.1	Target on galaxy S5 phone not accurate

13 Appendix E – Setting Up and Program Gimbal Robot

For Target-less Calibration or Hybrid Calibration, Gimbal can be used to automatically move the camera based on scene detection results. Before using Gimbal, first setup and program it (load the custom firmware WidowX-MX28.ino).

Firstly, mount camera to Gimbal, connect the FTDI-USB cable to an USB port on your host PC, then connect the camera to host using the USB 3.0 cable or USB Type C cable. And finally, connect the Gimbal power cable and power it up. If the FTDI-USB cable is connected properly, a serial port will be shown on the host PC, which is used by the sample app to send commands to Gimbal and control its movement.

Figure 13-1 Sample app calibration using Gimbal



13.1.1.1 Gimbal setup and usage on Ubuntu

Setup reference:
http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide

1. Install the Arduino on Ubuntu:

1a) Install Java runtime (needed by Arduino):

$ sudo apt-get install openjdk-8-jre

1b)
Download arduino-1.0.6-linux64.tgz from https://www.arduino.cc/download_handler.php?f=/arduino-1.0.6-linux64.tgz and unzip it to your home directory (~/Downloads/arduino-1.0.6, for example).

2. Install the ArbotiX-M Hardware and Library Files:

Download and unzip https://github.com/trossenrobotics/arbotix/archive/master.zip to a folder of your choice, then move these 3 folders into your 'Arduino' user folder (~/sketchbook).

To find your 'Arduino' user folder:
Open the Arduino 1.0.6 IDE and open the 'Preferences' panel (File->Preferences). Here you will find a file path under 'Sketchbook location:'. This is the path to your 'Arduino' user folder (~/sketchbook):

```
~/sketchbook$ tree -L 1

+-- ArbotiX Sketches
+-- hardware
+-- libraries
+-- README.md
```

3. Select Tools settings:

Tools->Board->:
Select "ArbotiX"

Tools->Programmer->:
Select "AVRISP mkII"

Tools->Serial Port->:
Pick the serial port (/dev/ttyUSB0, for example) for the FTDI device.

4. Upload flash image:
Open WidowX-MX28.ino in Arduino and select File->Upload.

5.

Also make sure /dev/ttyUSB0 exists and has the correct permissions so your account/app can access it. You can add your user to the 'dialout' group in order to have write permissions on that port:

```
$ sudo usermod -a -G dialout yourUserName
```

Log off and log on again for the changes to take effect!

Or simply

```
$ sudo chmod 777 /dev/ttyUSB0
```

6. Run, example:

```
$ /usr/bin/Intel.Realsense.DynamicCalibrator -cli -m 2 -gimbal /dev/ttyUSB0
```

13.1.1.2 Gimbal setup and usage on Windows

Setup reference:
http://learn.trossenrobotics.com/arbotix/7-arbotix-quick-start-guide

1. Install the Arduino IDE v1.0.6 (arduino-1.0.6-windows.exe)

2. Install the FTDI drivers (2.12.26). Unzip and then:
Right click on ftdibus.inf, select Install.
Right click on ftdiport.inf, select Install.

Find serial port (com3, for example) for the FTDI device from the Device Manager.

3. Install the ArbotiX-M Hardware and Library Files

To install the ArbotiX-M files you will move these 3 folders into your 'Arduino' user folder. To find your 'Arduino' user folder:
Open the Arduino 1.0.6 IDE and open the 'Preferences' panel (File->Preferences). Here you will find a file path under 'Sketchbook location:'. This is the path to your 'Arduino' user folder. For Windows 8 and above, this folder is: C:\Users\yourusername\Documents\Arduino\

4. Select Tools settings:
Tools->Board->:
Select "ArbotiX"

Tools->Programmer->:
Select "AVRISP mkII"

Tools->Serial Port->:
Pick the serial port for the FTDI device.

5. Upload flash image:
Open WidowX-MX28.ino in Arduino 1.0.6 and select File->Upload.

6. Run, example:
```
/usr/bin/Intel.Realsense.DynamicCalibratorCLI.exe -m 2 -gimbal com3
```