Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

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

Intel and the Intel logo Xeon, Core trademarks of Intel Corporation in the U.S. and/or other countries.

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

© 2019 Intel Corporation. All rights reserved.
# Contents

## 1 Introduction
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 ............................................................................. 10

## 2 Overview
2.1 Calibration Parameters ............................................................................. 12
2.2 Types of Dynamic Calibration
  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
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 ............................................................... 18
  3.1.5 Test Installation ...................................................................... 18
3.2 Installation on Windows ............................................................................. 19
  3.2.1 Install 3rd-party dependencies .................................................. 19
  3.2.2 Package installation ...................................................................... 19

## 4 Calibration with Dynamic Calibrator
4.1 Dynamic Calibrator Graphical User Interface (GUI) ..................................... 25
4.2 Dynamic Calibrator Command Line Interface (CLI) .................................... 31
4.3 Connect camera to host system ................................................................ 31
4.4 Multiple camera support ........................................................................... 31
  4.4.1 List all camera devices with Graphical User Interface (GUI) ........... 32
  4.4.2 List all camera devices with Command Line Interface (CLI) ......... 33
  4.4.3 Perform calibration on selected device with Command Line Interface
       (CLI) ...................................................................................... 33
  4.4.4 Stop Calibration ...................................................................... 33
4.5 Targeted Calibration .............................................................................. 35
  4.5.1 Setup Calibration Target ........................................................... 35
  4.5.2 Running Dynamic Calibrator with targeted calibration ............... 35
  4.5.3 Auto Exposure Adjustment during Targeted Calibration ................ 39
  4.5.4 Target calibration phases .......................................................... 42

## 5 Handling Calibration Data with CustomRW Tool ............................................. 47
5.1 Restore Calibration on Device ................................................................. 47
5.2 Read Calibration from Device ................................................................ 48
5.3 Write Calibration to Device .................................................................... 51
5.4 Read and Write RAW Calibration Tables ................................................... 51
5.5 Read and Write Custom Fisheye Data ...................................................... 53
5.6 Read and Write Custom IMU Data ........................................................... 54
5.7 Dynamic Calibrator App Graphical User Interface (GUI) .......................... 56
5.8 Dynamic Calibrator App Command Line Interface (CLI) ........................................ 56
5.9 Connect camera to host system ........................................................................ 56
5.10 Multiple camera support ...................................................................................... 56
5.10.1 List all camera devices with Graphical User Interface (GUI) .................... 56
5.10.2 List all camera devices with Command Line Interface (CLI) ............... 56
5.10.3 Perform calibration on selected device with Command Line Interface (CLI) ........................................................................................................ 56
5.10.4 Stop Calibration .............................................................................................. 56
5.11 Targeted Calibration ......................................................................................... 56
5.11.1 Setup Calibration Target ............................................................................... 56
5.11.2 Running Dynamic Calibrator application with targeted calibration ........ 56
5.11.3 Auto Exposure Adjustment during Targeted Calibration ......................... 56
5.11.4 Targeted calibration phases .......................................................................... 56
5.12 Target-less Calibration ...................................................................................... 56
5.12.1 Running Dynamic Calibrator Application with Target-less Calibration ... 56
5.12.2 Scenes ............................................................................................................. 56
5.12.3 Using Gimbal with target-less calibration .................................................. 56
6 Troubleshooting ...................................................................................................... 57
7 Appendix A - Calibration Check ............................................................................. 59
7.1 Quick Check ........................................................................................................ 59
7.2 Accuracy Check ................................................................................................... 60
8 Appendix B - Calibration Process Highlights .......................................................... 61
8.1 Targeted Calibration .......................................................................................... 61
9 Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup .................... 68
9.1 Introduction ........................................................................................................ 68
9.2 Dynamic Calibration Flow .................................................................................. 68
9.3 Printed Target Setup .......................................................................................... 69
9.4 Phone Target Setup on Apple iPhones Devices .................................................. 72
9.4.1 Locate the app on Apple App Store ............................................................ 73
9.4.2 Install the App on iPhone .............................................................................. 73
9.4.3 Running the app on phone ........................................................................... 74
9.4.4 Perform Dynamic Calibration ....................................................................... 76
9.5 Phone Target Setup on Android Phone Devices ............................................... 78
9.5.1 Locate the app on Google Play Store .......................................................... 78
9.5.2 Install the App on Android Phone ............................................................... 78
9.5.3 Running the app on phone ............................................................................ 74
9.5.4 Perform Dynamic Calibration ....................................................................... 76
10 Appendix D - Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App .......................................................... 83
10.1 Validated IOS Devices ....................................................................................... 83
10.2 Validated AOS devices ...................................................................................... 83
10.3 Devices with accuracy issue in phone target .................................................... 84
Tables

Table 1-1. Supported Intel® RealSense™ Depth Modules and Depth Cameras ....... 10
Table 1-2. Software Requirements ........................................................................ 10
Table 3-1: Calibration Tools and API Packages ................................................... 15
Table 4-1: Dynamic Calibrator Command Line Options ....................................... 31
Table 5-1: CustomRW Command Line Options .................................................... 47
Table 5-2: Dynamic Calibrator app command line options .................................... 56
Table 6-1 Troubleshooting .................................................................................. 57
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 ........................................... 35
Figure 4-2 Dynamic Calibrator Initial Start Window ........................................... 39
Figure 4-3 Dynamic Calibrator app warm up window ..................................... 40
Figure 4-4 Dynamic Calibrator Targeted Rectification Phase ......................... 42
Figure 4-5 Dynamic Calibrator Targeted Rectification Phase (continued) .......... 43
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 Restore Calibration on Selected Device ......................................... 48
Figure 5-2 Read Calibration on Selected Device ............................................. 49
Figure 5-3 Write Calibration on Selected Device ............................................. 51
Figure 5-4 Read Calibration RAW Data on Selected Device and Display on Screen .............................................................................................................. 52
Figure 5-5 Read and Write Calibration RAW Data on Selected Device .......... 53
Figure 5-6 Read Fisheye Custom Data on Selected Device and Display on Screen ............................................................................................................. 54
Figure 5-7 Read IMU Custom Data on Selected Device and Display on Screen ... 55
Figure 5-8 Read and Write IMU Custom Data on D435i Device........................ 56
Figure 5-9. Calibration target displayed on phone ............................................ 56
Figure 5-10 Dynamic Calibrator app initial starting window ............................ 56
Figure 5-11 Dynamic Calibrator app warm up window .................................. 56
Figure 5-12 Dynamic Calibrator app targeted rectification phase ................ 56
Figure 5-13 Dynamic Calibrator app targeted rectification phase (continued) .... 56
Figure 5-14 Dynamic Calibrator app targeted rectification phase (continue) .... 56
Figure 5-15 Dynamic Calibrator app scale phase .......................................... 56
Figure 5-16 Dynamic Calibrator app scale phase (continue) ........................... 56
Figure 5-17 Dynamic Calibrator Target-less Calibration ............................... 56
Figure 7-1. Calibration Check – Camera in Calibration .................................. 59
Figure 7-2. Calibration Check – Camera Out of Calibration ............................ 60
Figure 9-1 dynamic calibration flow............................................................... 68
Figure 9-2 Target Image ............................................................................... 69
Figure 9-3 Target Print Options ...................................................................... 70
Figure 9-4 Target Verification ......................................................................... 71
Figure 9-5 Printed target setup ....................................................................... 72
## Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Description</th>
<th>Revision Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.2.0</td>
<td>First document for Calibration Tool version 2.5.2.0</td>
<td>Jan 2018</td>
</tr>
<tr>
<td>2.6.4.0</td>
<td>Support RGB calibration/ Updated kernel patching instruction on Linux</td>
<td>August 2018</td>
</tr>
<tr>
<td>2.6.8.0</td>
<td>Support D435i</td>
<td>January 2019</td>
</tr>
<tr>
<td></td>
<td>Custom Calibration Data R/W API</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support hardware sync in RGB calibration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support calibration raw data read/write</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support Windows 10 RS4</td>
<td></td>
</tr>
</tbody>
</table>
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™ D400 Series Calibration Tools and API software release. The usage of calibration Tool API are separately covered in document "Intel® RealSense™ D400 Series Calibration Tools API programmer’s guide”

1.2 Components

The Intel® RealSense™ 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


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

<table>
<thead>
<tr>
<th>No.</th>
<th>Depth Module/Depth Cameras</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel® RealSense™ Depth Module D400</td>
</tr>
<tr>
<td>2</td>
<td>Intel® RealSense™ Depth Module D410</td>
</tr>
<tr>
<td>3</td>
<td>Intel® RealSense™ Depth Module D415</td>
</tr>
<tr>
<td>4</td>
<td>Intel® RealSense™ Depth Camera D415</td>
</tr>
<tr>
<td>5</td>
<td>Intel® RealSense™ Depth Module D420</td>
</tr>
<tr>
<td>6</td>
<td>Intel® RealSense™ Depth Module D430</td>
</tr>
<tr>
<td>7</td>
<td>Intel® RealSense™ Depth Camera D435</td>
</tr>
<tr>
<td>8</td>
<td>Intel® RealSense™ Depth Camera D435i</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® RealSense™ camera firmware</td>
<td>5.10.13.0 or higher on D435i, 5.10.6.0 or higher on other devices</td>
</tr>
<tr>
<td>Platform</td>
<td>Requirements</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Linux*</td>
<td>64-bit host system supporting Ubuntu* version 16.04 and Ubuntu 18.04</td>
</tr>
<tr>
<td>Linux* Kernel</td>
<td>4.4.0 and later</td>
</tr>
<tr>
<td>Windows*</td>
<td>Windows* 8.1 64-bit, Windows* 10 64-bit</td>
</tr>
</tbody>
</table>
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., D415 and D435.

Intrinsic includes
- Focal length - specified as \([f_x; f_y]\) in pixels for left, right, and RGB cameras
- Principal point - specified as \([p_x; p_y]\) in pixels for left, right, and RGB cameras
- Distortion - specified as Brown’s distortion model \([k_1; k_2; p_1; p_2; k_3]\) 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™ 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. Similarly 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

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

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

The Intel® RealSense™ SDK includes a Depth Quality Tool that can be used to test the quality of the Intel® RealSense™ D400 series depth module/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), Windows 8.1 64-bit 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).

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:

   ```bash
   sudo apt-get update
   ```

2. Install libusb-1.0:

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

3. Install libglfw:

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

4. Install freeglut:

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

5. Install libpng12-dev:

On Ubuntu 16.04, run the following command:

```
sudo apt-get install libpng12-dev
```

On Ubuntu 18.04, run the commands below:

```
wget -q -O /tmp/libpng12.deb
http://security.ubuntu.com/ubuntu/pool/main/libp/libpng/libpng12-0_1.2.54-1ubuntu1.1_amd64.deb
sudo dpkg -i /tmp/libpng12.deb
```

3.1.2 Calibration Tools and API Package Installation

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>librscalibrationtool</td>
<td>Intel® RealSense™ Dynamic Calibrator and other related tools</td>
</tr>
<tr>
<td>librscalibrationapi</td>
<td>Developer package including Calibration API and example sources</td>
</tr>
</tbody>
</table>

The packages are available for installation through Amazon AWS.

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.

- Register the server’s public key:

```
sudo apt-key adv --keyserver keys.gnupg.net --recv-key C8B3A55A6F3EFCDE || sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key C8B3A55A6F3EFCDE
```

- Add Intel server to the list of repositories:

```
sudo add-apt-repository "deb http://realsense-hw-public.s3.amazonaws.com/Debian/apt-repo xenial main"
```

- Remove the old records:

```
sudo rm -f /etc/apt/sources.list.d/realsense-public.list
```
• 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
```

• (Optional) Developers shall also install librscalibrationapi package:

```
sudo apt-get install librscalibrationapi
```

This is optional. Regular user does not require this package. Only users who would like to develop third party calibration apps or customize the calibration apps for their specific usages need to install this package.

### 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.

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
 | | | |...... changelog
 | | | |...... copyright
 | | | |...... License.txt
 | | | |...... README.md
 | | | |...... target
```

Using a terminal, navigate to the directory where the dynamic calibration tools release files are installed. For example,

```
cd /usr/share/doc/librscalibrationtool
```
Installation

Notes:
- Intel.Realsense.DynamicCalibrator is the main calibrator app
- Intel.Realsense.CustomRW is the custom calibration data read/write tool

For Calibration API, librscalibrationapi, if installed, the files are installed as below:

```bash
sudo dpkg -L librscalibrationapi
```

```
/usr
 | .......lib
 |     | ....... libDSDynamicCalibrationAPI.so
 |     | ....... librs2-crw-mm.so
 |     | ....... librealsense2.so
 |     | ....... librealsense2.so.2
 |     | ....... librealsense2.so.2.16.5
 | ----include
 |     | librscalibrationapi
 |         | ---- DSDynamicCalibration.h
 |         | ---- DSOSUtils.h
 |         | ---- DSShared.h
 |         | ---- DSCalData.h
 |         | ---- rs2-custom-calibration-mm.h
 | ----share
 |     | ---- doc
 |         | ...... librscalibrationapi
 |             | ...... attributions.txt
 |             | ...... changelog
 |             | ...... copyright
 |             |     | ...... include
 |             | ...... License.txt
 |             | ...... README.md
 |             |     | ...... target
 |             |     | ...... examples
 |             |         | ---- DynamicCalibrator
 |             |         | ---- CustomRW
 |             |         | ---- simple-example
```

Using a terminal, navigate to the directory with the dynamic calibration API release files are installed. For example,

```bash
cd /usr/share/doc/librscalibrationapi
```

Notes:
Installation

- libDSDynamicCalibrationAPI.so is the calibration API and DSDynamicCalibration.h is its main header file
- librs2-crw-mm.so is the IMU custom calibration data read/write API and rs2-custom-calibration-mm.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, and simple-example contains an example for simple IMU data read/write.

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.

```bash
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:

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

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

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

If you see the correct version information and device data printed on screen, congratulations you have the right setup and ready to use the calibration tools.
3.2 **Installation on Windows**

The Intel® RealSense™ Dynamic Calibration Tool runs on Windows 8.1 and 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 two install packages for Windows:

- Calibration Tool
  
  CalibrationTool-2.6.8.0-Setup.exe

- Calibration API
  
  CalibrationAPI-2.6.8.0-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-2.6.8.0-Setup.exe. Simply click it to install on your system. By default, it installs under C:\CalibrationToolAPI\2.6.8.0.
Installation
The final installation should look like following directory structure:

C:\CalibrationToolAPI

|-------- 2.6.8.0
|       |-------- bin
|       |       |-------- Intel.Realsense.DynamicCalibrator.exe
|       |       |-------- Intel.Realsense.DynamicCalibratorCLI.exe
|       |       |-------- Intel.Realsense.CustomRW.exe
|       |       |-------- attributions.txt
|       |       |-------- license.txt
|       |-------- target

The installed directory contains two versions of the Dynamic Calibrator tool, both share same functionality but different user interfaces:

- Intel.Realsense.DynamicCalibrator.exe with a graphical user interface
- Intel.Realsense.DynamicCalibratorCLI.exe with a command line interface with command line options

It also includes Intel.Realsense.CustomRW.exe tool which allows user to dump the calibration data from device, write calibration parameters to device, and reset the device calibration to default gold settings.

The installer will create Dynamic Calibrator icon link to 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 always navigate to the install directory using a windows terminal.

For example,

```
  cd CalibrationToolAPI
  cd 2.6.7.0
```

### 3.2.2.2 Calibration API Windows Installer Package

This is optional. Regular user does not require this package. Only users who would like to develop third party calibration apps for the D400 devices or customize the calibration apps for their specific usages need to install this package.

The installer package compresses all contents into a single installer file, for example, `CalibrationAPI-2.6.7.0-Setup.exe`. Simply click it to install on your system. By default, it installs under `C:\CalibrationToolAPI\2.6.7.0`. 
By default, it installs to the same folder as Calibration Tool package. The following additional directories/files are added on top of Calibration Tool package:

C:\CalibrationToolAPI

<table>
<thead>
<tr>
<th>2.6.8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>DSDynamicCalibrationAPI.dll</td>
</tr>
<tr>
<td>rs2-crw-mm.dll</td>
</tr>
<tr>
<td>freeglut.dll</td>
</tr>
<tr>
<td>realsense2.dll</td>
</tr>
</tbody>
</table>
Notes:
- DSDynamicCalibrationAPI.DLL is the calibration API and DSDynamicCalibration.h is its main header file
- rs2-crw-mm.dll is the IMU custom calibration data read/write API and rs2-custom-calibration-mm.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
  - simple-example contains an example for simple IMU data read/write, and
  - 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 Depth Module D400 Series Custom Calibration* white paper
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 (GUI)

The Graphical User Interface (GUI) is provided for easier use. The app detects all Intel RealSense D400 series devices on the system and lists them in a dropdown box on the top left corner.
Calibration with Dynamic Calibrator

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

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

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.

Targeted calibration requires a special target printed on paper or displayed on phone screen with the “Intel RealSense Dynamic Target Tool” app on Windows, Linux and Android phones. This app is available on Apple App Store and Google Play, search “Dynamic Target Tool” or “realssense” 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 Developers Guide document for Intel RealSense Depth Camera D400 Dynamic Calibration.
Calibration with Dynamic Calibrator

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

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

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

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:
Calibration with Dynamic Calibrator

Rectification Phase:

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

Scale Phase:

Move slowly to position target on different locations until completion.
4.2 Dynamic Calibrator Command Line Interface (CLI)

A number of command line options are supported in Dynamic Calibrator command line interface.

**Table 4-1: Dynamic Calibrator Command Line Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help or -?</td>
<td>Display list of command line options.</td>
</tr>
<tr>
<td>-version -v</td>
<td>Show Calibration Tool API version info</td>
</tr>
<tr>
<td>-show or -list</td>
<td>Display list of connected cameras with device name, serial number and firmware version information.</td>
</tr>
<tr>
<td>-dump &lt;0 or 1&gt; -d &lt;0 or 1&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>-setpoint &lt;0 to 4095&gt; -s &lt;0 to 4095&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>-sn &lt;camera serial number&gt; -serial &lt; camera serial number&gt;</td>
<td>If multiple cameras connected to the current system, choose one of the camera devices to calibrate by specifying its serial number.</td>
</tr>
<tr>
<td>-timeout -t &lt;Time out duration in seconds&gt;</td>
<td>Customize time out duration. By default, calibration session will time out in 180 seconds for target-less and 300 seconds for targeted calibration.</td>
</tr>
<tr>
<td>-cli</td>
<td>Run app in CLI mode</td>
</tr>
<tr>
<td>-error -e</td>
<td>Report rectification (vertical alignment) error in pixels periodically until timeout</td>
</tr>
<tr>
<td>-rgbOnly -rgb</td>
<td>Calibrate RGB only, for debug purpose</td>
</tr>
</tbody>
</table>

The app source code is contained in the Sample folder.

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 (GUI)

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

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

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Serial Number</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel RealSense 410</td>
<td>718212020356</td>
<td>05.10.06.00</td>
</tr>
<tr>
<td>Intel RealSense 430</td>
<td>722312071055</td>
<td>05.10.06.00</td>
</tr>
</tbody>
</table>

On **Windows**:

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

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Serial Number</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel RealSense 410</td>
<td>718212020356</td>
<td>05.10.06.00</td>
</tr>
<tr>
<td>Intel RealSense 430</td>
<td>722312071055</td>
<td>05.10.06.00</td>
</tr>
</tbody>
</table>

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.
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 50 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 D400 Series 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 of the device is displayed as well:
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:
Calibration with Dynamic Calibrator

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:

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.
Calibration with Dynamic Calibrator

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

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.

```bash
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**

Followed by a warm up screen that indicates progress in getting the camera ready for calibration.
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.
Calibration with Dynamic Calibrator
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 blue squares are present in middle part of the window:

**Figure 4-4 Dynamic Calibrator Targeted Rectification Phase**
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)**

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
Calibration with Dynamic Calibrator

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

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

Figure 4-8 Dynamic Calibrator Scale Phase (continued)
Calibration with Dynamic Calibrator

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.
Calibration with Dynamic Calibrator

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:

![GUI Calibration Successful](image1)

Results update in CLI mode:

![CLI Calibration Results](image2)

This completes the targeted calibration process.
5 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 5-1: CustomRW Command Line Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help or -?</td>
<td>Display list of command line options.</td>
</tr>
<tr>
<td>-version</td>
<td>Show version info</td>
</tr>
<tr>
<td>-list</td>
<td>Display list of connected devices with device name, serial number and firmware version information.</td>
</tr>
<tr>
<td>-sn &lt;camera serial number&gt; -serial &lt; camera serial number&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>-g</td>
<td>Reset calibration on selected device to default gold factory settings</td>
</tr>
<tr>
<td>-r</td>
<td>Read calibration data</td>
</tr>
<tr>
<td>-w</td>
<td>Write calibration data</td>
</tr>
<tr>
<td>-raw &lt;19, 1f, or 20&gt;</td>
<td>Specify calibration table raw binary content to read/write. Valid table id 19, 1f, and 20</td>
</tr>
<tr>
<td>-fe</td>
<td>Specify custom fisheye data to read/write. Available ONLY on selected device SKU.</td>
</tr>
<tr>
<td>-mm</td>
<td>Specify custom IMU data to read/write. Available ONLY on D435i</td>
</tr>
<tr>
<td>-file &lt;file name&gt; -f &lt;file name&gt;</td>
<td>Specify file source/destination. When read, by default, without specifying a file destination, the data will be display on screen in the command console.</td>
</tr>
</tbody>
</table>

5.1 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 5-1 Restore Calibration on Selected Device**

```
C:\CalibrationTool\API\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -s 728612070848
CustomRW for Intel RealSense D400, Version: 2.6.7.0
Device Name Serial Number Firmware Version
Intel RealSense D435 728612070848 05.10.06.00
Intel RealSense D420 70312090056 05.10.05.00
Intel RealSense D430 with Tracking Module 781512070239 05.10.06.00

C:\CalibrationTool\API\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070848 -g
CustomRW for Intel RealSense D400, Version: 2.6.7.0
Device PID: 8807
Device name: Intel RealSense D435
Serial number: 728612070848
Firmware version: 05.10.06.00
Calibration on device successfully reset to default gold factory settings.
C:\CalibrationTool\API\2.6.7.0\bin>
```

### 5.2 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.
The calibration parameters are defined in section 2.1 "Calibration Parameters". The XML file format is as below:

```xml
<?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>641.171</value>
  </param>
  <param name = "PrincipalPointRight">
    <value>644.808</value>
    <value>405.959</value>
  </param>
  <param name = "DistortionRight">
    <value>-0.0550314</value>
    <value>0.0646809</value>
    <value>0.00062232</value>
    <value>-0.000774438</value>
    <value>-0.0208662</value>
  </param>
</Config>
```
<param name = "RotationLeftRight">  
  <value>-0.055748</value>  
  <value>0.0653811</value>  
  <value>-0.00102014</value>  
  <value>-0.000605032</value>  
  <value>-0.0212901</value>  
</param>  
<param name = "RotationLeftRGB">  
  <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>
5.3 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 5-3 Write Calibration on Selected Device**

![Write Calibration on Selected Device](image)

5.4 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
```
Handling Calibration Data with CustomRW Tool

Figure 5-4 Read Calibration RAW Data on Selected Device and Display on Screen

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

```
```

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

```
```
5.5 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

```bash
Intel.Realsense.CustomRW.exe -r -fe
```
5.6 Read and Write Custom IMU Data

The -mm option supports read/write the custom IMU data on selected device SKU. It only works on very limited range of devices, currently D435i. Use this option ONLY when the device is supported.

**Important:** On a new D435i devices, IMU data is empty. The device IMU should be calibrated and IMU data is stored to the device first. Reading empty IMU data will fail.

Example #8: read custom IMU data from device and display to the terminal
```
Intel.Realsense.CustomRW.exe -r -mm
```
Handling Calibration Data with CustomRW Tool

Figure 5-7 Read IMU Custom Data on Selected Device and Display on Screen

Example: read custom IMU data from device and save into a binary file
Intel.Realsense.CustomRW.exe -r -mm -f myimu.bin

Example: write custom IMU data from binary file into the device
Intel.Realsense.CustomRW.exe -w -mm -f myimu.bin
Figure 5-8 Read and Write IMU Custom Data on D435i Device

C:\Windows\System32\cmd.exe

C:\CalibrationToolAPI\2.6.8.0\bin>Intel.RealSense.CustomRW.exe -r -mm -f myimu.bin
CustomRW for Intel RealSense D400, Version: 1.6.8.0

Device PID: 883A
Device name: Intel RealSense D435i
Serial number: 831612073544
Firmware version: 05.10.13.00

imu custom data on device successfully read into file myimu.bin

C:\CalibrationToolAPI\2.6.8.0\bin>Intel.RealSense.CustomRW.exe -w -mm -f myimu.bin
CustomRW for Intel RealSense D400, Version: 2.0.8.0

Device PID: 883A
Device name: Intel RealSense D435i
Serial number: 831612073544
Firmware version: 05.10.13.00

imu custom data successfully write to device
# 6 Troubleshooting

## Table 6-1 Troubleshooting

<table>
<thead>
<tr>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>With targeted calibration on Linux, after rectification phase is successfully completed, sample app windows may become black</td>
<td>The most likely 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: <strong>Method 1 - Pin/hold your kernel</strong> Examples: Discover the current latest kernel currently installed. It's the one with the biggest version number. $ dpkg -l</td>
</tr>
<tr>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>Devices calibrated through targeted dynamic calibration shows greater depth error than expected (i.e., more than 2%)</td>
<td>Follow target setup instruction to check the target physical sizes. Target size is critical to calibration accuracy.</td>
</tr>
</tbody>
</table>
### Troubleshooting

| 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. |
7 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:

7.1 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 7-1. Calibration Check – Camera in Calibration
7.2 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.
Appendix B - Calibration Process Highlights

8 Appendix B - Calibration Process Highlights

8.1 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.
Appendix B - Calibration Process Highlights

TARGET SETUP - TARGET HIGHLIGHTS
The target contains a graphical pattern with fixed-width striped bars in the middle and small checker blocks on the top and bottom. This pattern displayed on a phone depends on the screen size. The larger phones display more and larger bars than the smaller screens. The pattern used in printed targets is not the same as the pattern displayed on different phones. The width of the bars in the middle highlighted in green (except for 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., 30mm, 60mm, 100mm, 150mm, etc. The pattern guide has detailed instructions in checking these dimensions in the target pattern printed on paper or displayed on phones to ensure accuracy.

TARGET SETUP - PRINTED TARGET
A target image print target fixed width, 4.25 x 7.5 inch is supplied in the calibration tool installation under the target directory. Print it on 8.5 x 11 inch letter size paper with 1/2 inch top and bottom border. This target is ideal to verify the size and position of the target when using different devices or positions. The target is printed on a flat surface, for example, a poster board (5mm, available in art supply stores) or foam-core 4.25" x 7.5", available in most art stores and major retailers, so that the bars are vertical. Make a reference mark on either side of 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.
Appendix B - Calibration Process Highlights

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. Scans "dynamic target test" or "Windrose" on the apps to find and install it on your iPhone or Android phone. The phone app is currently supported on iOS on iPhones 5, 6, 6+, 6S, 6S+, 7, and 7r, and on Android on Samsung Galaxy 3, 4, 5, 6, 7. Android 8.0 and higher are also supported. Please refer to the User Guide for list of supported phones.

TARGET SETUP - POSITION AND ORIENTATION
Position the device 800 - 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, minimal movement between the device and the target is required. Depending 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.
Appendix B - Calibration Process Highlights

TARGET SETUP - AVOID REFLECTION

Avoid reflections from environment - Lighting reflections on phone screen severe impacts targeted calibration effectiveness. When a phone target is used where lighting reflections on phone's front screens presents, for example, reflection under bright sunlight, or indoor with lots of lights surrounding, calibration performance is degraded because of false targets. Calibration may not converge (los impacted images blown away and target cannot be found) and it takes longer to template or no result may be reached.

Default
Appendix B - Calibration Process Highlights

**CONTRAST DEVICE AND START CALIBRATION**

Connect both devices to the device to calibrate. Click "Start Calibration" to start the process which involves two A photos depending on Wiki. Follow all instructions to finish each phase in a sequential order. If phase target is used, be careful with the target. The phase target may interfere with the calibration process and the application may not be able to identify the target. Try to position the target at a location from lighting and other surroundings can be avoided. If necessary, you can disable the automatic exposure search in the "Threshold Setting" and manually adjust the exposure setting before starting the calibration.

**RECTIFICATION PHASE**

The rectification phase rectifies the left and right images and prepares for the next phase where scene error in depth can be corrected. Position the device 600-1000mm away pointing to the target so that the target is approximately vertically oriented on the 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 blocks and white squares and bars in the target chart overlaps the blue squares in the window. The blue squares will start blinking. Upon detecting the process until all blue squares are closed. 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 exposure to search for the target). If this fails, it implies that the lighting faces 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 keeps through exposure system and still cannot detect the target, possibly due to reflections on phone screen or too close objects far away from the target, try to position the device or the target and try again.
Appendix B - Calibration Process Highlights

Scale Phase

The scale phase corrects the depth scale errors. It requires capturing 15 images of the target at various positions. Position the device 800 - 600 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 rigid motion of this movement is required. Also, it does not need to be in the same plane. Capture position and distance in the range is fine. Once a position is accepted, a blue rectangle highlights that target position will be drawn in the window. The new position should be the one that does not have images previously captured. The app chooses that and marks the new position if unique. The green precision bar at the bottom shows progress. Repeat this process until it finishes. At end of the scale phase, the calibration results of rectification and scale phase are updated to the device no matter if there is a good phase or not.

RGB Phase (Only on devices with RGB)

The RGB phase defines the depth to RGB for UV mapping and only available on devices with RGB, for example, D445 and D455. This phase is very similar to the scale phase and also requires capturing 15 images of the target at various positions. Position the device 800 - 650 mm away pointing at 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 IR/depth depths and depth-UV are calibrated and the actual conversions are highlighted in the output error in terms of degrees of rotation angles.
Appendix B - Calibration Process Highlights

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

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

9.1 Introduction

Dynamic calibration is used to re-calibrate the Intel® RealSense™ 3D 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.

9.2 Dynamic Calibration Flow

The figuration below shows the overall dynamic calibration process flow.

![Figure 9-1 dynamic calibration flow](image)

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

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 9-2 Target Image

![Target Image](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.

9.3 Printed Target Setup

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

9.4 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.

9.4.1 **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"
- Scan QR code for downloading from App Store:

![QR Code](image)

9.4.2 **Install the App on iPhone**

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

![App Store screen](image)

Press the download/install icon to install the app on your phone.
Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup

9.4.3 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:
Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup

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.
**9.4.4 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.
/usr/bin/Intel.Realsense.DynamicCalibrator

On Windows
Double click the "Dynamic Calibrator" icon on Windows desktop to start the calibrator app.
9.5 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.

9.5.1 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):
  - Scan QR code for downloading from Google Play:

9.5.2 Install the App on Android Phone

Once the app is found on the app store, the following informational pages display
Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup

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

App installed and the Dynamic Target Tool icon appears on phone screen:
9.5.3 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**
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.
9.5.4 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.
Appendix D - Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App

10 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/

10.1 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.

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

10.2 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.
### Appendix D - Validated Devices for Intel® RealSense™ Dynamic Target Tool Phone App

<table>
<thead>
<tr>
<th>Device tested formally in dev and QA</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung Galaxy S7</td>
<td>6.0.1 (Marshmallow)</td>
</tr>
<tr>
<td>Google Nexus 5</td>
<td>4.4.2 (KitKat)</td>
</tr>
<tr>
<td>LG G4</td>
<td>6.0 (Marshmallow)</td>
</tr>
<tr>
<td>Galaxy Note 5</td>
<td>6.0 and 7.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Devices tested ad hoc by users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Nexus 6</td>
</tr>
<tr>
<td>Galaxy S7 Edge</td>
</tr>
<tr>
<td>Samsung Galaxy S8 plus</td>
</tr>
</tbody>
</table>

### 10.3 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:

<table>
<thead>
<tr>
<th>Device with target accuracy issue</th>
<th>OS</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung Galaxy S5</td>
<td>5.1.1</td>
<td>Target on galaxy S5 phone not accurate</td>
</tr>
</tbody>
</table>