Qualcomm Technologies, Inc.

Dual Camera Assembly and Calibration Guide

80-N5126-3 Rev. T

September 22, 2022

For additional information or to submit technical questions, go to: https://createpoint.qti.qualcomm.com

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	June 2015	Initial release
B	June 2015	Updated for qcaldc v1.3.1
C	July 2015	Added printing charts to PDF
D	August 2015	Updated for qcaldc v1.4.1
E	March 2016	Numerous updates have been made to this document; it should be read it its entirety.
F	July 2016	Updated qcaldc v2.1.1 to support single chart calibration; added dual FOV assembly in Section 2.3; numerous changes to stereo calibration procedure in Chapter 6
G	September 2016	This document has been updated to conform with QTI standards; no technical content has been changed
H	November 2016	Added left.tiff, right.tiff, and flatChart.tiff to document attachments
J	February 2017	Chapter 6 is new to this document revision. Numerous changes were made to Section 2.6 and Chapters 5 and 7. Additional changes are shaded in Appendix A.
K	December 2017	Numerous changes have been made to Chapter 3, Chapter 6 and Appendix A.
L	December 2017	Updated the L-chart and F-chart images in the attachment for 60cm calibration distance. Fixed minor inconsistencies in output metric and output parameter definitions in Appendix A.
M	February 2019	Updated AEC sync calibration procedure in Chapter 4.
N	April 2019	Added the relative sync calibration format and absolute sync calibration format to Chapter 4.
P	June 2019	Minor updates to Chapter 4.
R	June 2019	Editorial update.
T	September 2022	Updated Chapters 2, 6, and Appendix A

Contents

1 Introduction 6

1.1 Purpose 6
1.2 Conventions 6
1.3 Technical assistance 6

2 Commence project 7

2.1 Asymmetric assembly 7
2.2 Symmetric Bayer/mono assembly 9
2.3 Dual FOV assembly 10
2.4 Camera alignment 10
2.5 Electrical considerations 12
2.6 Dual camera considerations 12

3 Calibrate Auto Focus sync 14

3.1 Collect initial data 14
3.2 Calibrate module 15
3.3 Validate module 16
3.4 Calculate DoF 19

4 Calibrate AEC sync 21

4.1 Dual-camera relative sync calibration 21
4.1.1 Master and slave cameras 21
4.1.2 Setup 22
4.1.3 Calibrate an individual module 24
4.1.4 AEC sync OTP data format 25
4.1.5 QTI verification 25
4.2 Multi-camera absolute sync calibration 26
4.2.1 Setup 26
4.2.2 Choose proper gain and exposure time 26
4.2.3 Calibrate an individual module 27
4.2.4 AEC sync OTP data format 28
4.2.5 QTI verification 28
4.2.6 Calibrate other camera modules 28

5 Calibrate and verify OIS 29

6 Calibrate stereo 30

6.1 Stereo camera basics 30
6.2 Main and aux cameras 32

6.3 Calibration scene setup 33
6.4 Calibration procedure 36
6.5 Calibration verification 37
6.6 QTI verification........ 38

A qcaldc software package 39
A.1 qcaldc.cfg parameters 40
A.1.1 Calculating focal_length_ratio 42
A.1.2 Setting the cameras intrinsic parameters........ 42
A.1.3 Calculating marker threshold........ 43
A.1.4 Lens distortion correction 43
A.1.5 Running in Debug mode........ 45
A.1.6 Running in Full Calibration mode 46
A.2 qcaldc output metrics 47

B AEC sync calibration sample code........ 50

C References 52
C.1 Related documents 52
C.2 Acronyms and terms 52

Figures

Figure 1-1 Dual camera process........ 6
Figure 2-1 Asymmetric dual camera module mechanical guidelines example 8
Figure 2-2 Symmetric Bayer/mono dual camera module mechanical guidelines 9
Figure 2-3 Mechanical guidelines for dual FOV camera modules........10
Figure 2-4 Ideal symmetric dual camera module alignment for cameras with the same focal length 11
Figure 3-1 AF SYNC validation for AF SYNC Class A........17
Figure 3-2 First-order approximation of camera DoF 19
Figure 4-1 Example of an ideal gray chart 22
Figure 6-1 Pinhole Camera Model 31
Figure 6-2 Dual Camera FOV and Coordinate Systems........31
Figure 6-3 Dual Camera FOV Rectification........32
Figure 6-4 Dual camera calibration charts with green markers........33
Figure 6-5 Dual camera calibration charts with aruco markers........33
Figure 6-6 Calibration scene setup 34
Figure 6-7 Camera height setting 35
Figure 6-8 Camera distance setting........35
Figure A-1 Green enhanced chart image dump 43
Figure A-2 Lens distortion F-Theta model........44
Figure A-3 Lens distortion model example........44
Figure A-4 Rectified images with guidelines 45
Figure A-5 Main Image with detected checkerboard corners........46

Tables

Table 2-1 Asymmetric dual camera module sensors 8
Table 2-2 Symmetric Bayer/mono dual camera module sensor selections 9
Table 3-1 Examples of AF SYNC target distance selection 15
Table 3-2 AF SYNC calibration EEPROM/OTP map 16
Table 3-3 AF sync classes 17
Table A-1 qcaldc configuration parameters 40
Table A-2 Output parameters 47

1 Introduction

1.1 Purpose

This document provides module vendors with procedures to evaluate, calibrate, and verify dual camera modules.

Dual camera modules enable use cases such as depth mapping, instant auto focus (IAF), fusion or image quality enhancement, and optical zoom.

The following flow chart outlines the dual camera process from project commencement to mass production.

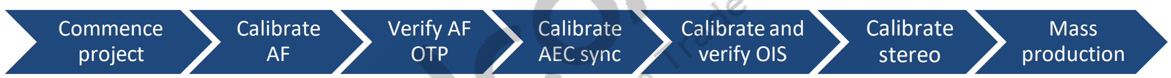


Figure 1-1 Dual camera process

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `cp armcc armcpp`.

Code variables appear in angle brackets, for example, <number>.

Commands to be entered appear in a different font, for example, **copy a:*.* b:**.

Button and key names appear in bold font, for example, click **Save** or press **Enter**.

Shading indicates content that has been added or changed in this revision of the document.

1.3 Technical assistance

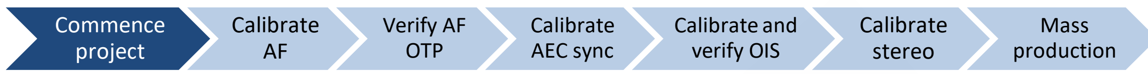
For assistance or clarification on information in this document, open a technical support case at https://support.qualcomm.com/.

You will need to register for a Qualcomm ID account and your company must have support enabled to access our Case system.

Other systems and support resources are listed on https://qualcomm.com/support.

If you need further assistance, you can send an email to qualcomm.support@qti.qualcomm.com.

2 Commence project



Evaluate industry dual camera assemblies during early project stages.

There are multiple types of dual camera assembly used in the industry:

- Asymmetric Bayer/Bayer (or Bayer/YUV) – Uses a high resolution primary sensor and a low resolution auxiliary sensor. Typical use cases for this assembly are high quality depth map and fast auto focus.
- Symmetric Bayer/mono – Uses equivalent sensors and lens optics in Bayer + mono color filter configurations. The typical use case for this assembly is fusion (image quality enhancement). The Qualcomm® Clear Sight™ AOST feature may be used to merge images taken from Bayer and mono cameras.
- Dual FOV (wide/tele) – Uses two camera modules with different FOV to achieve an optical zoom feature.
- Multiple FoV (e.g., TeleZoom/Tele/Wide/UltraWide) – Uses multiple camera modules with different FoV to achieve DSLR like optical range. Dual camera calibration should be applied in this case to every pair of cameras used in concurrency.

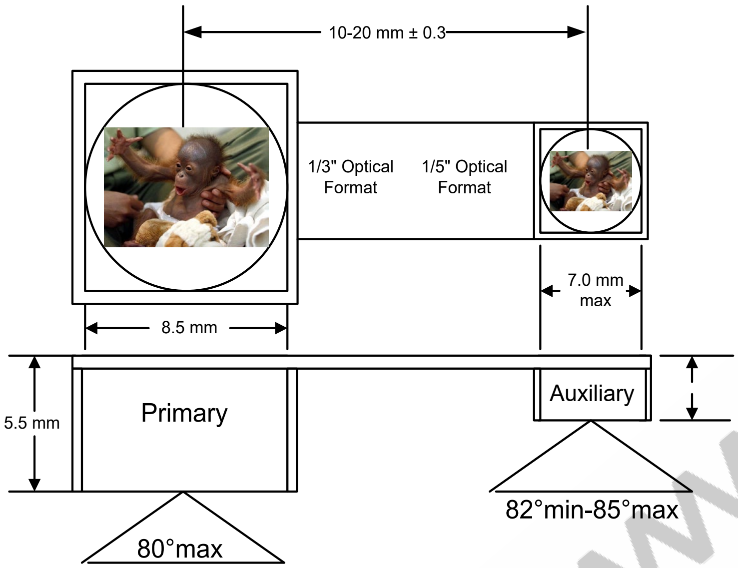
NOTE: Assembly frame, shielding, and connector/flex are not relevant to the compatibility of the assembly with QTI architecture, provided that dimensional stability is maintained.

2.1 Asymmetric assembly

Figure 2-1 illustrates an asymmetric dual camera assembly with a 1/3 in. primary camera. The module is composed of two cameras and nonvolatile memory (NVM) to store calibration coefficients. Other modules may include more cameras with fixed or dynamic focus. Multiple camera assembly may also include usage of several single camera modules, assembled separately on the mobile platform. In this example the primary camera includes auto focus (AF) with a Bayer color filter array (CFA), and the secondary camera has a fixed-focus lens.

Camera separation (baseline) may change depending on the intended primary use case.

- Instant AF (IAF) – 10 mm separation is recommended to allow for compact assembly



Asymmetric Dual Camera Assembly 1/3" Primary

Figure 2-1 Asymmetric dual camera module mechanical guidelines example

The field of view (FOV) for the auxiliary camera needs to be larger than or equal to that of the primary camera, also the following factors should be into considered:

- Manufacturing tolerances of both cameras
- Change in FOV for both cameras over focus range
- Changes in FOV for both cameras over manufacturing and environment
- Overlap at minimum distance
- Maximum resolution loss intended at minimum working distance

An 80° maximum FOV for the primary camera meets required module heights without incurring excessive distortion, logic cell array (LCA), shading, or other aberrations.

The resolution of the auxiliary sensor reduces the resolution of the main sensor by up to one-fourth (in both dimensions) without a visible reduction in quality. For the 13 MP dual camera in Figure 2-1, the auxiliary sensor operates in a 2 x 2 Binning mode with an output of 800 x 600 pixels at 30 fps. For the 21 MP dual camera module, the auxiliary sensor operates at its full resolution of 1600 x 1200 pixels at 24 fps. The following table shows these two example configurations:

Table 2-1 Asymmetric dual camera module sensors

Configuration	Primary	Auxiliary
13 MP + 2 MP	OV13850	OV2685
21 MP + 2 MP	IMX230	OV2685

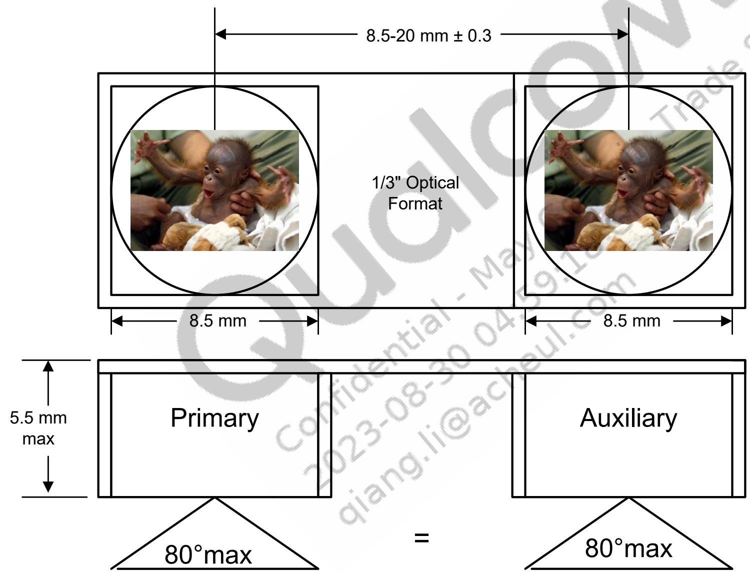
2.2 Symmetric Bayer/mono assembly

Figure 2-2 illustrates a symmetric Bayer/mono dual camera assembly with 1/3” sensors. The module is composed of two cameras and an NVM to store calibration coefficients. The primary camera includes AF with a Bayer CFA. The secondary camera includes AF and a mono sensor.

For Bayer/mono dual camera sensors, the Bayer sensor must have PDAF to enable fast focus. No PDAF is recommended on the mono camera.

The camera separation (baseline) is based on the primary intended use case:

- Depth mapping performance – 20 mm separation is recommended
- IAF – 10 mm minimum separation recommended
- Fusion (image quality enhancement) – Minimal separation (<10 mm) between the modules for best quality fusion result.



Symmetric Dual Camera Assembly

Figure 2-2 Symmetric Bayer/mono dual camera module mechanical guidelines

An 80° maximum FOV for primary camera meets required module heights without incurring excessive distortion, LCA, shading, or other aberrations. The following table shows two example symmetric dual camera module configurations.

Table 2-2 Symmetric Bayer/mono dual camera module sensor selections

Configuration	Primary Bayer	Auxiliary mono
13 MP + 13 MP	IMX258 Bayer	IMX258 Mono
13 MP + 13 MP	S5K3L8 Bayer	S5K3L8 Mono

2.3 Dual FOV assembly

Figure 2-3 depicts a dual FOV camera assembly. The module is composed of two cameras and an NVM to store calibration coefficients. The utility of this assembly is to provide optical zoom for the customer.

The primary camera has a wide FOV. The secondary camera has a narrow FOV and serves as the telephoto function. Both sensors have a Bayer CFA.

Camera separation is determined by the optical zoom use case. Zoom is optimized with minimum separation between the optical centers of the cameras. The minimum separation is determined by module construction.

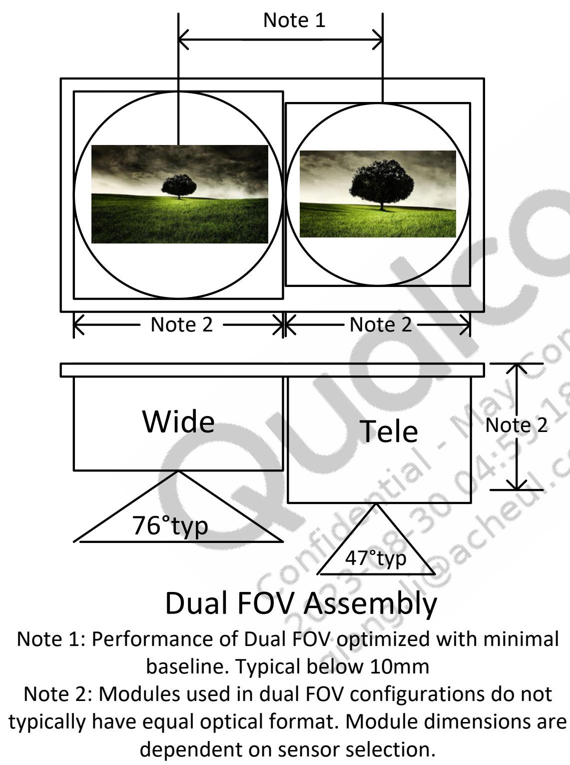


Figure 2-3 Mechanical guidelines for dual FOV camera modules

2.4 Camera alignment

There are three sets of alignment requirements when placing two cameras on one board:

- Static tolerances – Requirements imposed on the maximum allowed translation and rotation between the two cameras, during module production (before stereo calibration).
- Post-calibration static tolerances – Requirements imposed on the sturdiness of module construction (after stereo calibration).
- Dynamic tolerances – Requirements imposed on the amount of dynamic lens tilt allowed (after stereo calibration).

Static tolerances

Static tolerance requirements include translational and rotational tolerances. While the ideal arrangement is not physically feasible, the following constraints should be placed:

- Sensor planes should be aligned with the optical axis of the camera lens.
- Figure 2-4 below shows a simplified illustration of a symmetric dual camera module with two cameras that have the same focal length. The lens planes of the two cameras should be aligned on the same Z-plane, separated by distance B (camera baseline in mm) along the Y-axis, with no vertical shift along the X-axis. Note that if the two cameras have different focal lengths f1 and f2, the ideal image planes will have a small translational shift along Z-axis, equal to the difference of the two focal lengths f1-f2 (assuming f1 > f2).
- Translational tolerances for shifts along X-, Y- and Z-axes shown in Figure 2-4 should not exceed ±0.3 mm with respect to the ideal values. Ideal shifts along X-, Y- and Z-axes are: 0.0 mm, B mm and f1-f2 mm respectively.
- Rotational tolerances around X- , Y- and Z-axes (Yaw, Pitch and Roll respectively) should not exceed 1.0° with respect to ideal values. Ideal rotations along x-, y- and z-axes are 0.0°.

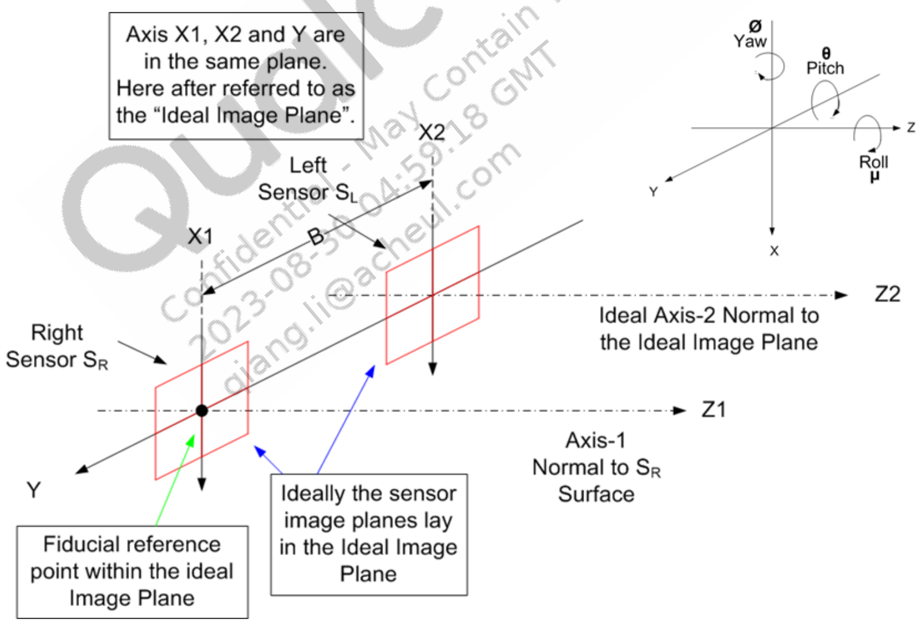


Figure 2-4 Ideal symmetric dual camera module alignment for cameras with the same focal length

Post-calibration static tolerances

Note that all translational and rotational changes **after** Qualcomm Dual Camera calibration must be held to tight tolerances. Some of the steps that can be taken to ensure tight tolerances are:

- Use a sturdy and robust mechanical frame to house the two cameras. The frame should not warp or bend when the module is being mounted on the device or after drop.

- Use high quality glue and curing process to affix each camera to the assembly frame.

Module vendors should conduct extensive drop tests on the final assembled modules to ensure that they remain within static tolerance limits.

Dynamic tolerances

Lens dynamic tilt is defined as movement of the lens along a direction that is not parallel to the optical axis. In AF lenses, the optical axis of the lens barrel can change due to several factors:

- Lens barrel movement during AF operation
- Vibration
- Camera orientation, e.g., pointing up, down, horizontally, or any direction in between

The tolerances for lens dynamic tilt are:

- Maximum allowable lens dynamic tilt around X- and Y-axes (Yaw and Pitch respectively) of each separate camera is ±0.16° about its own optical axis, along the entire range of lens position movement from infinity to macro.

For specific scenario tolerances, performance depends on design factors. Contact QTI for further guidance.

2.5 Electrical considerations

Each camera module has an independent I2C interface with MSM™ chipsets through the MIPI CSI-2 lanes. This allows identical AF controllers and sensors to be used in the two modules.

One camera module is designated the master and the other is designated the slave. The master outputs a vertical frame synchronization signal. The slave has an input pin, which serves as the vertical frame synchronization input. These two lines tie together internally to the dual camera assembly or at the board level.

2.6 Dual camera considerations

Auto-Focus

Dual camera applications, such as depth mapping, instant AF (IAF), fusion and optical zoom require precise control of the AF lens position. This expedites AF functionality and allows for precise alignment of the optical axes of the two cameras during software rectification on the phone.

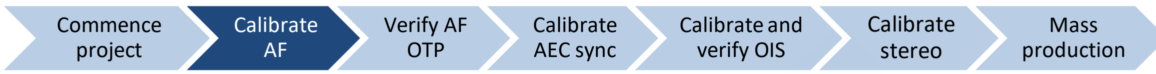
Closed loop AF actuators are strongly recommended for dual camera modules. They achieve precise control of the AF lens position with a lens position feedback mechanism. This decreases lens position inaccuracies from changes in orientation, temperature, and time. Open loop AF actuators may be used to save module cost, but at the expense of reduced lens position accuracy and AF speed.

Optical Image Stabilization

Depth map and related dual camera applications require that the optical axes of the two cameras remain as invariant as possible. OIS operation by definition shifts the optical axis to compensate for user's hand shake. So, it is recommended that the OIS be turned off such that the optical axis remains fixed in the neutral center position during depth map applications. Likewise, OIS should be turned off and the optical axis should be fixed in the same neutral center position during the factory stereo calibration.

Some OIS actuator mechanisms such as VCM spring type offer more precise control of the optical axis compared to others such as VCM ball bearing type. Because it is critical to maintain optical axis at an accurate neutral position during both stereo calibration as well as during depth map application, the use of precise actuator mechanisms is recommended.

3 Calibrate Auto Focus sync



The dual-camera auto focus synchronization (AF SYNC) calibration process characterizes and stores the lens position map between the two cameras in the dual-camera system. This calibration procedure exists in addition to conventional auto focus calibration performed on the individual camera modules. The measured lens position values from the conventional auto focus calibration procedure may be read and re-used during the AF SYNC calibration.

AF SYNC calibration consists of 3 phases:

- Initial data collection from sample modules prior to per module calibration
- Per-module calibration
- Validation of calibration results performed on each module

3.1 Collect initial data

Perform initial data collection from sample modules using DualCameraAFVerification.xlsx from the Attachments portion of this document. The purposes of this data collection is two-fold:

- Ensure the repeatability of lens position measurements based on focus search, and
- Assess whether the conversion of the lens positions between the two cameras can be represented by a linear function. The latter requirement arises from the need to keep per module calibration requirements and its costs as low as possible.

QTI dual-camera auto focus requires the lens actuators to provide a linear actuation of the lens of each of the two cameras to be able to provide a competitive auto focus performance and an accurate manual focus mode.

- In a Wide-Tele dual-camera the auto focus provides synchronized lens movement in the target distance range from 50-cm to infinity.
- In Bayer-Mono dual-camera modules, the auto focus provides synchronized lens movements from 10-cm to infinity.

Nonlinear regions in lens actuation lead to significant module-to-module. Per-module calibration of the actuator response over the nonlinear region using multiple chart distances and rounds of focus search is not standard industry practice. The requirements for the linear actuation of the lens in both cameras enable highly efficient and cost-effective auto focus synchronization calibration.

The data collection prescribed in DualCameraAFVerification.xlsx requires the module vendor to obtain lens positions at predefined target distances at both nominal temperatures and warm temperatures. The warm temperature is decided by the OEM and module vendor. If the dual-camera auto focus is expected to operate without significant performance degradation up to 60 °C, for example, then the data collection must be performed at 60 °C. The suggested number of modules is more than 10, and the number of iteration is more than 3. Refer the DualCameraAFVerification.xlsx file for more detailed information.

3.2 Calibrate module

Per-module calibration of AF SYNC calibration entails performing focus searches using a suitable test chart, such as ISO 12233, placed at predefined target distances and storing the corresponding lens positions in the non-volatile memory of dual-camera module. The lens position for each target distance must be measured from each camera. Table 3-1 provides examples of AF SYNC calibration chart distances. The chart distance selection for AF SYNC calibration must adhere to the following requirements:

- AF SYNC chart distance of each camera must include its own macro distance limit
- AF SYNC chart distance of each camera must include a far distance equal to or greater than its hyperfocal distance
- AF SYNC chart distances must be separated by at least 10-cm. Module vendor may increase the separation distance as long as the minimum is 10-cm
- At least 3 chart distances greater than 15-cm must be shared between two cameras

Table 3-1 Examples of AF SYNC target distance selection

Dual camera example	AFS calibration distances
Wide/Tele: Tele macro limit is 50-cm. The higher hyperfocal distance of two is 3.6m	Wide: 10-cm, 50-cm, 60-cm, 3.6m Tele: 50-cm, 60-cm, 3.6m
Wide/Tele: Tele macro limit is 35-cm. The higher hyperfocal distance of two is 4.1m	Wide: 10-cm, 35-cm, 45-cm, 4.1m Tele: 35-cm, 45-cm, 4.1m
Bayer/Mono: Both have macro limit of 10cm. The higher hyperfocal distance of two is 3.2m	Bayer: 10-cm, 20-cm, 30-cm, 3.2m Mono: 10-cm, 20-cm, 30-cm, 3.2m
Bayer/Mono: Both have macro limit of 8cm. The higher hyperfocal distance of two is 3.2m	Bayer: 8-cm, 20-cm, 40-cm, 3.2m Mono: 8-cm, 20-cm, 40-cm, 3.2m

If the measured lens position from auto focus calibration is to be re-used during AF SYNC calibration perform AF SYNC calibration in the same camera module orientation – face-up, face-down, or face-forward – as the conventional auto focus calibration.

Calibrate both cameras in the dual-camera system using the same camera orientation. For example, it is not feasible for Wide camera to be calibrated face-down while Tele camera is calibrated face-up. The requirement to use identical module orientation in dual-camera calibration applies to module designs where closed-loop actuators are used in one or both camera modules.

In cases where the collimating (zoom) optics used by the auto focus test station do not provide a distance that meet or exceed hyperfocal distance of a given camera, estimate the lens position at a hyperfocal distance from auto focus calibration results.

Table 3-2 shows the expected EEPROM/OTP format in which the calibration results are to be stored. The calibration results can be stored in the following ways:

- Each camera stores AF SYNC calibration data in its own EEPROM/OTP. AF SYNC calibration for individual cameras may be performed in parallel and on different test stations.
- AF SYNC calibration results are stored in a single EEPROM/OTP.

Table 3-2 AF SYNC calibration EEPROM/OTP map

Dual Camera AF Calibration Content		
Rel. addr	**Description**	**Endian**
0x0000	Version number	H
0x0001	Version number	L
0x0002	Number of Distances [N]	H
0x0003	Number of Distances [N]	L
0x0004	Chart Distance 0 [cm] (FAR LIMIT)	H
0x0005	Chart Distance 0 [cm] (FAR LIMIT)	L
0x0006	Lens Pos DAC at Distance 0	H
0x0007	Lens Pos DAC at Distance 0	L
0x0008	Chart Distance 1 [cm]	H
0x0009	Chart Distance 1 [cm]	L
0x000A	Lens Pos DAC at Distance 1	H
0x000B	Lens Pos DAC at Distance 1	L
…	Test Chart Distance N [cm] (MACRO LIMIT)	H
…	Test Chart Distance N [cm] (MACRO LIMIT)	L
…	Lens Pos DAC at Distance N	H
	Lens Pos DAC at Distance N	L
…	CHECKSUM %256	

3.3 Validate module

AF SYNC calibration is followed by a validation test to be carried out using Qualcomm provided code. This can be performed on any test station equipped to read AF SYNC calibration results. The AF SYNC validation test must adhere to the following requirements:

- Calibration results from both cameras are read since the test examines the lens position mapping from one camera to another as illustrated in Figure 3-1.
- In the dual-camera system, both auto and manual focus operations require the lens position of two cameras at an identical target distance. This synchronization is achieved with the lens mapping function so that a focus decision may be transferred from the designated camera (master) to the other camera (slave).

- The camera with higher depth-of-focus (DoF) is assigned the role of slave since DoF represents a fundamental noise floor from which the peak focus lens position can be obtained in any focus search.
- For Wide/Tele cameras, the tele camera is designated the slave since a telephoto lens has a higher DoF.
- In Bayer/Mono, Mono is assigned the slave since Mono camera may conditionally power on when low light enhancement or bokeh effects are desired.

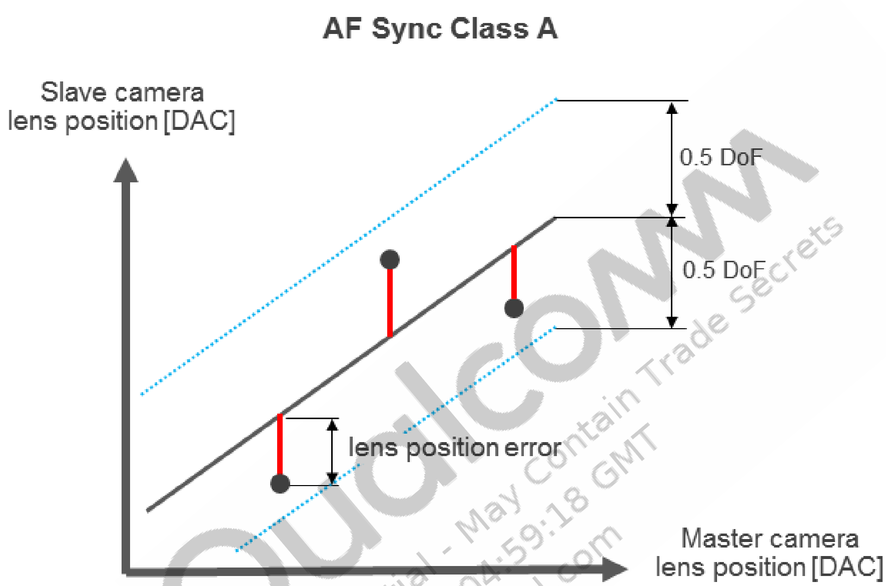


Figure 3-1 AF SYNC validation for AF SYNC Class A

AF SYNC validation code examines fitting errors (or residuals) resulting from a linear regression. It assigns a quality classification to every dual-camera module. There are four quality classes: A, B, C, and D. The following table summarizes the requirements for best focus of each AF sync classes on slave cameras.

Table 3-3 AF sync classes

Auto focus sync class	Requirements	Default limit
A	Successful focus search result on the master camera. No additional focus search needed on the slave camera.	Max abs lens position error < DoF x AF_SYNC_CLASS_A_TOL (default AF_SYNC_CLASS_A_TOL = 0.5)
B	Minimal additional focus search on slave camera	Max abs lens position error < DoF x AF_SYNC_CLASS_B_TOL (default AF_SYNC_CLASS_B_TOL = 3.0)
C	Extended focus search on slave camera	Max abs lens position error < DoF x AF_SYNC_CLASS_C_TOL (default AF_SYNC_CLASS_C_TOL = 5.0)
D	Independent focus search	Max abs lens position error >= DoF x AF_SYNC_CLASS_C_TOL

Class A

Class A describes dual-camera modules for which the lens position for best focus on slave camera is determined from the focus search result of the master camera using the lens position map. The following points illustrate AF SYNC Class A:

- Maintains the default value 0.5 for *AF_SYNC_CLASS_TOL_A* and use without alteration since dual-camera programs assigned AF SYNC Class A will completely bypass fine scan on slave camera auto focus, in order to achieve the highest performance possible.
- Requires the use of closed-loop actuators on each of dual-camera modules to ensure that lens position remains independent of lens sag.

Class B

Class B describes dual-camera modules for which some additional focus search on slave camera is required, using up to 5 additional frames. Class B ensures additional focus search on the slave camera, if the lens position error resulting from linear regression is approximately ±3 DoF of slave camera.

Class C

Class C describes dual-camera modules for which an extended focus search on the slave camera is required. AF SYNC Class C is expected to have lens position error resulting from linear regression of approximately ± 5 DoF. This process can use up to 11 additional frames on focus search performed on slave camera following that of master camera.

Class D

Class D describes dual-camera modules for which an independent focus search on the slave camera is required.

Class application requirements

The following points summarize the recommendations for dual camera:

- Bayer/Mono programs – AF SYNC Class A is required.
- Wide/Tele programs (image fusion not required) – AF SYNC Class C or above is required.

Dual-camera programs in which 100% of the modules are AF SYNC Class A are designated Class A programs. If 99.5% of the modules are Class A with 0.5% Class B, the overall program designation would still be Class B. This follows the observation that the dual-camera auto focus performance is constrained by corner modules rather than median modules. The class designation for the overall program is determined by the worst class designation received. The module vendors and the OEM may decide how to best reach the performance to cost trade-off.

In applications where images from two cameras are fused to provide low light enhancement, improved dynamic range, and/or improved digital zoom, auto focus must be able to synchronize lens position with the highest precision. For such applications, specify manufacturing quality standards such that the AF SYNC Class A program is attained.

In other dual-camera applications where image fusion is not required, uphold manufacturing quality standards such that dual-camera modules with AF SYNC Class C or higher is attained. Depending on the feature requirements of the dual-camera auto focus system, a higher AF SYNC class may be required.

3.4 Calculate DoF

The following figure illustrates the calculation of lens depth-of-focus (DoF) employed by AF SYNC validation code.

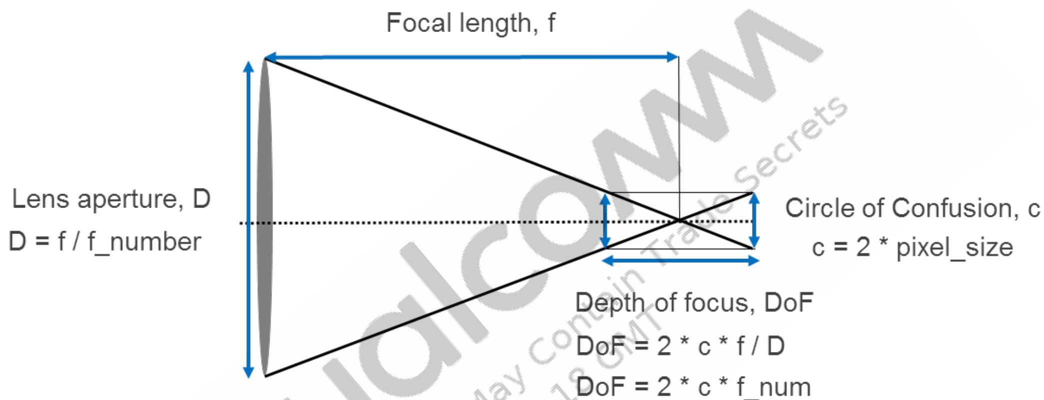


Figure 3-2 First-order approximation of camera DoF

The following points illustrate the DoF calculation:

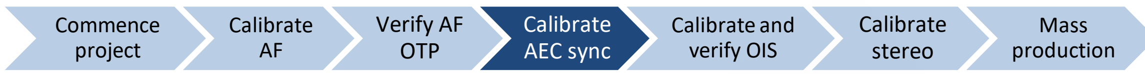
- The first-order approximation of DoF provides a theoretical lower bound on the size of lens position error. For this reason, AF SYNC Class A defines acceptable lens position error tolerance as ±0.5 DoF.
- The lens position resulting from focus search below 15-cm are not used in the linear regression of lens position map and in the validation of lens position error. This is because cameras with F# 2.2 or lower F# have significantly higher effective DoF at these macro distances leading to a substantially reduced precision in focus search results.
- AF SYNC validation code reports the maximum lens position error, estimated slave camera DoF in DAC, and AF SYNC class designation.
- The validation code can output a binary dump file that contains full details of the validation test.
- OEM and module vendors may use validation test results to facilitate a statistical analysis to assess whether focus search used during AF SYNC calibration needs to be improved.

Customers should:

1. Convert the calculated DoF in physical distance (for example, in µm) to DAC scale based on the estimated actuator sensitivity (µm/DAC) of each individual module.

2. Provide AF SYNC validation test results including the optional binary dump files from at least 25 dual-camera modules.

4 Calibrate AEC sync



AEC sync factory calibration is the process of estimating the sensor response difference between camera modules. There are two calibration methods:

- Dual-camera relative sync calibration
  - Applicable on both non-CamX and CamX platforms with AEC 6.0/7.0 and later.
- Multi-camera absolute sync calibration
  - Only applicable on SM8250 and later chipsets.

Follow the proper calibration format according to the platform of the project. Confirm the platform of project before AEC sync calibration.

The steps for the AEC sync calibration procedure are below:

- Choose the master and slave cameras (relative sync calibration only)
- Setup
- Enable sensors in un-binned mode
  - Because the goal is to find the average luma at the original sensor response, enable sensors in un-binned mode.
- Individual module calibration
  - The goal of this stage is to calibrate each individual camera module and write the corresponding data into OTP.

4.1 Dual-camera relative sync calibration

4.1.1 Master and slave cameras

For the purpose of AEC sync, one of the cameras is designated as the master and the other as the slave. The master and slave designation for AEC sync is independent from the main and aux designation for stereo calibration.

The following are the master and slave designations (for AEC sync purposes) for the various dual-camera module configurations:

- Bayer + mono dual camera
  - Bayer is the master and mono is the slave.
- Wide + tele dual camera
  - Wide is the master and tele is the slave.
- Wide + ultra-wide dual camera
  - Wide is the master and ultra-wide is the slave.
- Bayer + Bayer dual camera
  - The camera that is considered as the default or the primary camera is the master. This typically refers to the camera with the higher resolution. The other camera is considered the slave.

4.1.2 Setup

The goal of this stage is to find the proper reference gain and exposure time (in ms) for master and slave sensors (i.e., the slave_gain, slave_exposure_time_ms, master_gain, and master_exposure_time_ms). The setup stage is only needed to be run for one module unit during setup. The flat gray chart is used for master slave dual camera AEC sync calibration.

Figure 4-1 Example of an ideal gray chart

Setup capture environment

1. Use a flat gray chart or uniform target
    a. Make sure the chart is flat
    b. Make sure the gray is uniform
2. Target the dual camera module to the gray chart in the light box. The color temperature can be between 5000k to 6500k. The chart can be transmission type as well.
3. Move the camera to make sure the gray chart occupies 60-80% of the FOV in the center.
4. Adjust the camera angle to let optical axis perpendicular to the gray chart.

Choose proper gain and exposure time

1. Choose an initial set of slave_gain and slave_exposure_time_ms for the slave sensor. Since the exposure time is recorded only as an integer quantity, it is recommended that the exposure time is no less than 15 ms.
2. Collect slave raw data.
3. Calculate slave_g:
    a. Subtract the slave_black_level.
    b. Average the pixel values in the center region (1/8 width and 1/8 height).
4. Repeat steps 2 and 3 above and keep updating slave_gain and slave_exposure_time_ms until slave_y is 20% of the maximum brightness value.
    a. For example, target brightness value = 50 for pixel value range from 0 to 255 (8-bit)
    b. The 20% brightness does not need to be exactly 20% brightness (3% tolerance is acceptable). This means the brightness value can be in the range of 42 to 58 for pixel value range from 0-255 (8-bit).
    c. The given gain and exposure time should operate in the linear operation range of the sensor.
5. Save the slave_gain and slave_exposure_time_ms for future use.
6. Choose an initial set of master_gain and master_exposure_time_ms for the master sensor.
7. Collect master raw data.
8. Calculate master_g:
    a. Subtract the master_black_level.
    b. Average the pixel values at GR channel in the center region (1/8 width and 1/8 height).
9. Repeat steps 7 and 8 above and keep updating master_gain and master_exposure_time_ms until master_g is 20% of the maximum brightness value. The given gain and exposure time should operate in the linear operation range of the sensor.

10. Save the master_gain and master_exposure_time_ms for future use.

The results from this setup stage are:

- slave_gain
- slave_exposure_time_ms
- master_gain
- master_exposure_time_ms

4.1.3 Calibrate an individual module

The goal of this stage is to calculate the brightness ratio information of each individual dual camera module, and write this brightness_ratio, along with the reference slave_gain, slave_exposure_time_ms, master_gain, and master_exposure_time_ms obtained during the previously described setup stage into OTP (EEPROM).

Setup capture environment

Use the same setup as in Section 4.1.2.

Calculate the brightness ratio for each dual-camera module

1. Use the slave_gain and slave_exposure_time_ms obtained during the setup stage for the slave sensor.
2. Collect slave raw data.
3. Calculate slave_g:
    a. Subtract the slave_black_level.
    b. Average the pixel values in the center region (1/8 width and 1/8 height).
4. Use master_gain and master_exposure_time_ms obtained during Setup stage for Master sensor
5. Collect master raw data.
6. Calculate master_g:
    a. Subtract the master_black_level.
    b. Average the pixel values at GR channel in the center region (1/8 width and 1/8 height).
7. Calculate the brightness_ratio = slave_g/master_g.
8. Write the brightness_ratio into OTP data, format is 16-bit Q10 (see Appendix B for sample code).

Q Number conversion

To covert from floating point to fixed point Q10 16-bit integer conversion (see Appendix B for sample code):

1. Multiply the float value by 2^{10}

2. Round this float value to 16-bit integer

To convert from fixed point Q10 16-bit integer to floating point:

1. Convert the 16-bit integer to float value
2. Divide this float value by 2^{10}

4.1.4 AEC sync OTP data format

The result generated during Section 4.1.3 has to be programed into the OTP (EEPROM) for the camera driver to use. Format of the OTP data is shown below. Total size of the AEC sync OTP data is 10 bytes.

Relative byte offset	Data	Type	Bits	Description
0	brightness_ratio	Fixed point Q10	16	AEC Sync brightness ratio
2	ref_slave_gain	Fixed point Q10	16	Reference slave gain value obtained from setup stage and used during calibration stage
4	ref_slave_exposure_time_ms	16 bit unsigned Integer	16	Reference slave exposure time in milliseconds (truncated to the nearest integer) obtained from the setup stage and used during calibration
6	ref_master_gain	Fixed point Q10	16	Reference Master gain value obtained from setup stage and used during calibration stage
8	ref_master_exposure_time_ms	16 bit unsigned Integer	16	Reference Master exposure time in milliseconds (truncated to the nearest integer) obtained from the setup stage and used during calibration

4.1.5 QTI verification

After the AEC sync setup stage (on 1 module unit) and AEC sync individual module calibration stage (on at least 15 different module units), the module vendor should provide the following data to QTI for further verification:

1. The AEC sync OTP data of each dual camera module.
2. The captured (gray chart) raw images, from both the master and slave, of each dual camera module.
3. The black level used of each dual camera module (i.e., master_black_level and slave_black_level).

After QTI receives this data, QTI will validate the calibration correctness and provide feedback and guidance for the mass production stage.

4.2 Multi-camera absolute sync calibration

Multi-camera absolute sync calibration is only applicable on SM8250 and later chipsets.

4.2.1 Setup

The goal of this stage is to find the proper reference gain and exposure time (in us) for a golden module (i.e., the gain and exposure_time_us). The reference gain and exposure time (in us) will later be used for individual module calibration. The setup stage only needs to be run for one module unit (the golden module) during setup. Use a flat gray chart for multi camera AEC sync calibration (see Figure 4-1).

Setup capture environment

1. Use a flat gray chart or uniform target.
    a. Make sure the chart is flat.
    b. Make sure the gray is uniform.
2. Target the camera module to the gray chart in the light box. The color temperature can be between 5000k to 6500k. The chart can be transmission type.
3. Move the camera to make sure the gray chart occupies 60-80% of the FOV in the center.
4. Adjust the camera angle to make the optical axis perpendicular to the gray chart.

4.2.2 Choose proper gain and exposure time

1. Choose an initial set of gain and exposure_time_us for the golden module. Since the exposure time is recorded only as an integer quantity, it is recommended that the exposure time is no less than 15000 us (15 ms).
2. Collect the raw data.
3. Calculate average_luma:
    a. Extract the green channel.
    b. Subtract the black_level.
    c. Average the pixel values in the center region (1/8 width and 1/8 height).
4. Repeat steps 2 and 3 above and keep updating gain and exposure_time_us until the average_luma is ~20% of the maximum luma value.
    a. For example, target luma value = 50 for pixel value range from 0 to 255(8-bit).
    b. The 20% luma does not need to be exactly 20% brightness (3% tolerance is acceptable). This means the average_luma can be in the range of 42 to 58 for pixel value range from 0-255(8-bit).
    c. The given gain and exposure time should operate in the linear operation range of the sensor.
5. Save the gain and exposure_time_us for future use

After these steps, we will have the following results from this setup stage:

- gain
- exposure_time_us

4.2.3 Calibrate an individual module

The goal of this stage is to calculate the average_luma of each individual camera module by using the reference gain and exposure time. The calculated average_luma, along with the reference gain and exposure_time_us will be written into OTP (EEPROM).

Setup capture environment

Use the same setup as in Section 4.2.1 . Make sure to keep the same lighting condition, and position each individual module at the same location as the golden module during calibration.

Calculate the average luma for each individual module

1. Use the gain and exposure_time_us obtained during the setup stage for the current sensor.
2. Collect raw data.
3. Calculate average_luma:
   a. Extract the green channel.
   b. Subtract the black_level.
   c. Average the pixel values in the center region (1/8 width and 1/8 height).
4. Write the gain, exposure_time_us, and average_luma into OTP data. The gain has a format of 16-bit Q10. The exposure_time_us has the type of 16-bit unsigned integer. The average_luma also has the type of 16-bit unsigned integer. Assume 16 bits used for pixels. If not, scale the average_luma accordingly (see Appendix B for sample code).

Q Number conversion

To covert from floating point to fixed point Q10 16-bit integer conversion (see Appendix B for sample code):

1. Multiply the float value by 2^{10}
2. Round this float value to 16-bit integer

To convert from Fixed point Q10 16-bit integer to floating point:

1. Convert the 16-bit integer to float value
2. Divide this float value by 2^{10}

4.2.4 AEC sync OTP data format

The result generated during Section 4.2.3 has to be programed into the OTP (EEPROM) for the camera driver to use. Format of the OTP data is shown below. Total size of the AEC Sync OTP data is 8 bytes. Use 2.0 as the version number for this calibration guide.

Relative byte offset	Data	Type	Bits	Description
0	version	16-bit unsigned Integer	16	Aec sync OTP data format version number.
2	average_luma	16-bit unsigned Integer	16	Average of green channel pixels (16 bits) in the center of the frame.
4	gain	Fixed point Q10	16	Reference gain value obtained from setup stage and used during calibration stage
6	exposure_time_us	16-bit unsigned Integer	16	Reference exposure time in microseconds (truncated to the nearest integer) obtained from the setup stage and used during calibration

4.2.5 QTI verification

After AEC sync setup stage (on 1 module unit) and AEC sync individual module calibration stage (on at least 15 different module units), the module vendor should provide the following data to QTI for further verification:

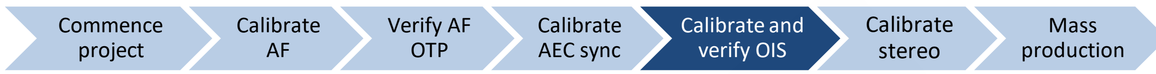
1. The AEC sync OTP data of each dual camera module.
2. The captured (gray chart) raw image of each camera module.
3. The black level used of each camera module.

After QTI receives this data, QTI will validate the calibration correctness and provide feedback and guidance for the mass production stage.

4.2.6 Calibrate other camera modules

The calibration process described earlier is only for one module (e.g., wide). For other modules (tele, ultra-wide, etc.), use the same procedure to find the proper gain and exposure time followed by per module calibration. It is important to note that the lighting condition and location should always stay the same during per module calibration and when switching the calibration from one type of module to another.

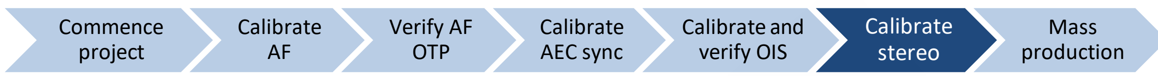
5 Calibrate and verify OIS



Depth map and related dual camera applications require that the optical axes of the two cameras remain as invariant as possible. OIS operation by definition shifts the optical axis to compensate for user's hand shake. It is therefore recommended that the OIS be turned off such that the optical axis remains fixed in the neutral center position during depth map applications. Likewise, the OIS should be turned off and the optical axis should fixed in the same neutral center position during factory stereo calibration.

Some OIS actuator mechanisms such as VCM spring type offer more precise control of the optical axis compared to others such as VCM ball bearing type. Because it is critical to maintain optical axis at an accurate neutral position during both stereo calibration and depth map application, using precise actuator mechanisms is recommended.

6 Calibrate stereo



Stereo Calibration (also called Dual camera calibration) is the process of estimating the relative spatial transformation between two cameras, or between each camera and a different plane, depending on the use case.

For example:

- Reprojection – Projecting the FOV of one camera onto the FOV of the other camera. This is needed when we want to combine the data from two sensor, e.g., an active depth sensor and a camera RGB sensor. Since the two sensors have a different FOV we would need to reproject the depth sensor image onto the RGB sensor image.
- SAT – Sensor Alignment Transformation
  In this use case we want to achieve a smooth transition between the FOV of two cameras, such as needed during zoom. For this use case we would need to project the FOV of each camera onto the FOV of the other camera, and onto various planes in between.
- DFS – Depth From Stereo
  In this use case we want to project the FOV of the two cameras onto a virtual plane, on which we can search for the disparity between the two images

6.1 Stereo camera basics

Some stereo camera calibration processes are based on a simplified camera model called the pinhole camera model. The pinhole camera model is an ideal model that neglects the effect of a lens and the aperture size, but it allows to describe the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane (as is the case with a camera and sensor).

Figure 6-1 depicts the pinhole camera model.

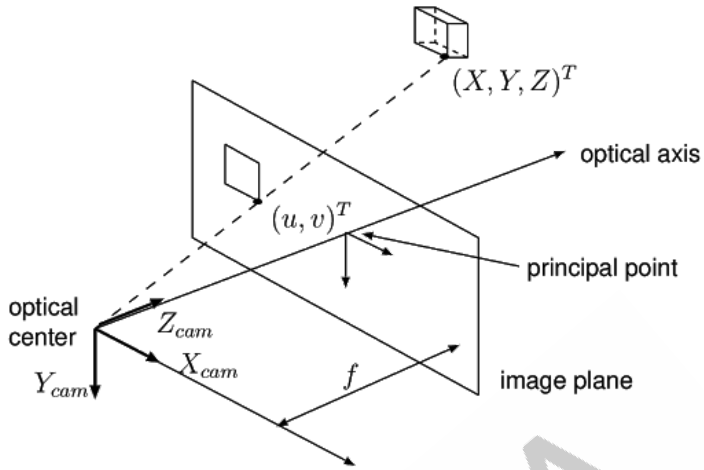


Figure 6-1 Pinhole Camera Model

A general calibration process is used for creating a projection transform for mapping world coordinates to pixel coordinate, including the following:

- Intrinsic camera parameters – Describing individual camera characteristics such as focal length and principal point
- Extrinsic parameters describing the transform between 3D world coordinates to 3D camera coordinate, or equivalently, the camera position and heading in the world coordinate
- For multiple cameras, the process would create a projective transform from one camera to another or to a mutual plane

NOTE: Nonlinear intrinsic parameters such as lens distortion are also important, although they cannot be included in the linear camera model described by the camera matrix. A preceding lens distortion correction calibration, involving 10-20 images, is needed for generating the lens distortion parameters, which has to be corrected before the stereo calibration, or be given as an input to the stereo camera calibration tool.

Figure 6-2 depicts two cameras coordinate system.

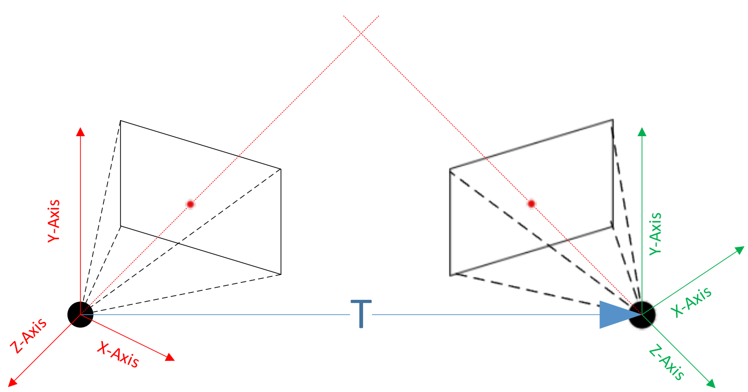


Figure 6-2 Dual Camera FOV and Coordinate Systems

T indicates the baseline, which is the distance between the two cameras optical centers. Each camera has its own coordinate system and FOV.

The stereo camera calibration calculates a rotation matrix and translation vector per camera, which enables the warping of the images between the two cameras coordinate system, or to another coordinate plane.

Specifically, for the use case of calculating depth from stereo, the stereo camera calibration tool calculates the transforms for each camera to a virtual (rectified) domain, in which the epipolar lines of the two cameras become parallel, and this makes the search for disparity between the two FOVs much more efficient.

Figure 6-3 shows the warping of the two cameras FOV to the rectified domain.

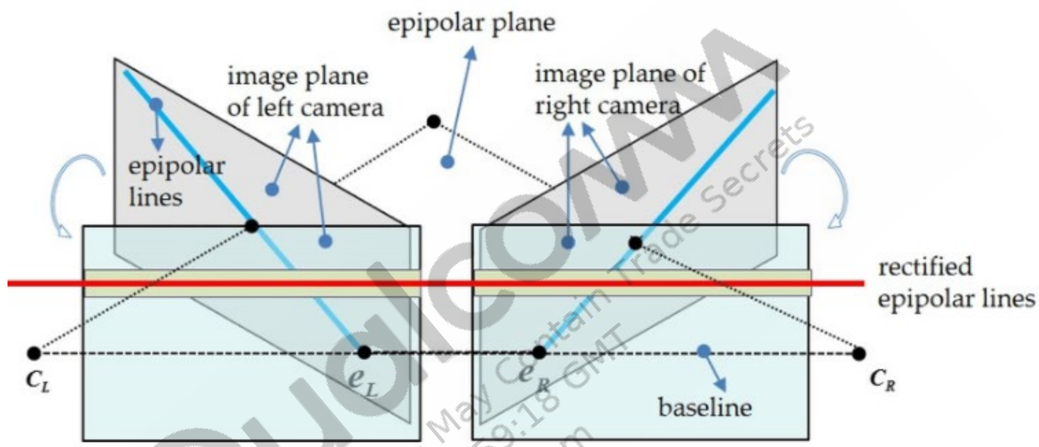


Figure 6-3 Dual Camera FOV Rectification

The two cameras are referred to equivalently as left and right, reference and auxiliary, or main and auxiliary.

6.2 Main and aux cameras

For the purpose of stereo calibration, one of the cameras is designated as the main and the other as the aux. The main and aux designation for stereo calibration is independent of the master and slave designation for AEC Sync. The main camera represents the viewpoint to which the auxiliary camera will be transformed. Designation of the main camera may change per use case, or even per zoom position.

The following are some examples:

- Bayer + Mono – Bayer is the main and Mono is the aux.
- Wide + Tele – Tele is the main and wide is the aux.
- Wide + Ultra-wide – Wide is the main and ultra-wide is the aux.
- SAT – During zoom which includes a transition between two cameras, one of the camera will act as the main and the other as aux during zoom-in, and vice versa during zoom-out

6.3 Calibration scene setup

Stereo calibration is required for every multi-camera module during mass production. Separate camera modules require stereo calibration for every device after the assembly on the device.

The calibration uses three identical checkerboard charts. The middle chart has four additional green circles (Figure 6-4) or Aruco markers (Figure 6-5).

Figure 6-4 Dual camera calibration charts with green markers

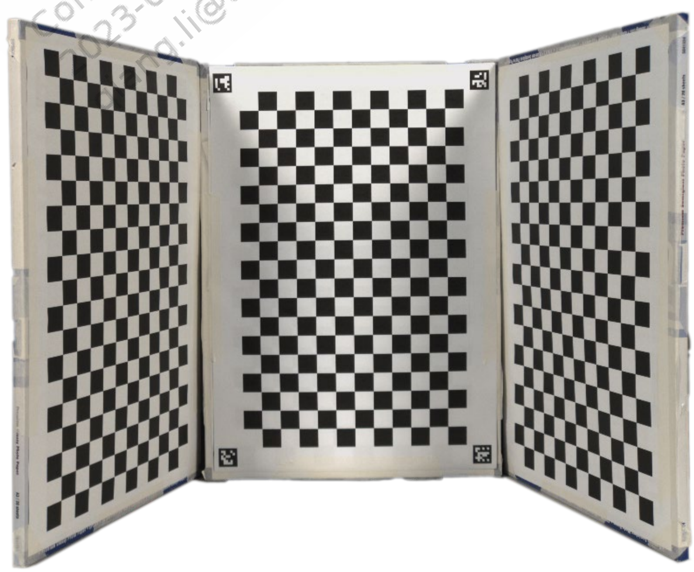


Figure 6-5 Dual camera calibration charts with aruco markers

The charts can be printed at any size (as long as they all have the same size), and should be attached to flat rigid surfaces. It is recommended to use the chart images which are provided with the qcaldc package (see Appendix A) and print them on a size A3 paper.

Figure 6-6 shows a top view of the charts and cameras setup.

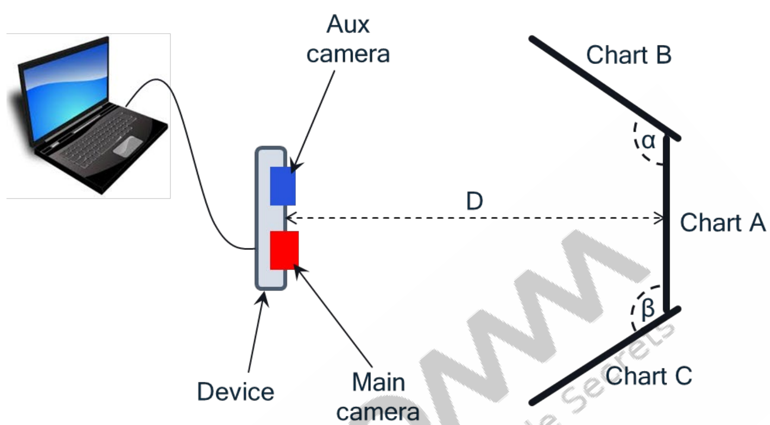
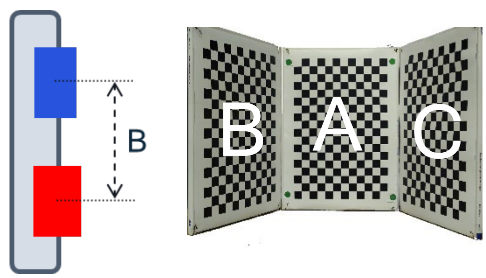


Figure 6-6 Calibration scene setup

The following designation are used in the figure above.

- Chart A – Middle chart
- Charts B, C – Side charts
- D – Distance between the device/module and chart
- B – Distance between camera pair centers
- α, β – Angles between main and side charts



A computer connected to the device is required for controlling the device for the image capture, and for download and running of the calibration tool. The calibration tool itself can be downloaded from https://createpoint.qti.qualcomm.com/.

The lighting conditions during capture are not strict, but it is recommended to have a light intensity of above 150 Lux on the chart surface. It is also important to avoid glare and reflections from the charts, which might impact the calibration tool.

Charts A, B, and C should be placed at roughly the same height, ideally on a flat surface, as shown above, where chart A is placed in the middle. The angles between the side charts and the main chart are not strict, but should be around 135° to chart A.

For best results, the device should be placed in front of chart A at a height and position directly against the center of the chart as shown in Figure 6-7.

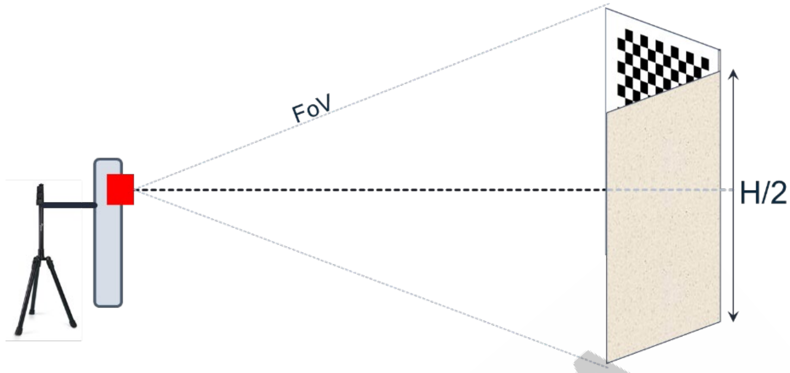


Figure 6-7 Camera height setting

The distance from the chart should be according to the FOV of the camera with the narrower FOV, such that the entire checkerboards and the margins around them are visible, as demonstrated in Figure 6-8.

Figure 6-8 Camera distance setting

The following data is also required for use either by the calibration tool or for calibration verification.

- The baseline distance B should be provided by the module/device maker, and would be used for yaw correction and calibration verification.
- The distance D should be measured as accurately as possible, and could also be used for the calibration verification.
- The checkerboard box size in mm, would also be used for yaw correction and calibration verification
- The lens distortion models that would be used for lens distortion correction
- The cameras intrinsic matrices (would also be available from the LDC calibration)
- Focal length ratio (FLR) per camera (should be calculated or recorded. See Appendix A.1.1.

NOTE: The same lens distortion model and intrinsic parameters can be applied to an entire camera module batch, and do not have to be calculated per device/module.

NOTE: If the intrinsic parameters are not available, the tool can estimate them from the input images, but this method is expected to yield less accurate results

NOTE: FLR is the ratio between the focal length at the current focus point and the focal length when focus is set to infinity. The focal length may change depending on the cameras FOV and the distance from the chart, so it is important to record it during the image capture.

NOTE: As mentioned in 2.6 Section 2.6, OIS must be disabled during capture.

6.4 Calibration procedure

1. Setup the calibration scene as described in Section 6.3.
2. Start the camera application on the device.
3. Focus the main and aux cameras to the center of the middle chart.
4. Capture the main and auxiliary images simultaneously and record the FLR for each camera.
5. Download the captured images.
6. Update the configuration file as needed and run the calibration tool.

Running the calibration tool

Use the command line from the folder where the tool executable is located:

```
qcaldc.exe <calibration configuration file>
```

qcaldc prints status messages to the console during operation, for example:

```
===========================
qcaldc version: 3.0
One Chart calibration
===========================
Parsing data
===========================
Splitting images
===========================
Calibrating images
===========================
Calibration verification results ...

         Metric                      Value   Threshold Result
Residual vertical disparity [%]      0.10       0.20   PASS
Scale difference [%]                 0.24      10.00   PASS
STDEV of Horizontal Disparity        1.79      10.00   PASS
Maximum Vertical Disparity [pixels]  6.63      15.00   PASS
Distance error [%]                   0.87      20.00   PASS
Writing calibration OTP data to new_otp.bin ...
```

```
=========================
Calibration successful
```

The tool returns the values of selected metrics, like residual disparity after rectification, scale difference between the rectified images, etc., and compares them to the predefined threshold, to produce a PASS/FAIL indication.

For additional information on the tool options type

```
qcaldc -help
```

For analysis use the -debug option, and redirect qcaldc output to a log file:

```
qcaldc -d qcaldc.cfg > qcaldc.log
```

The tool also supports running the OpenCV Stereo camera Calibration procedure, which can be used as reference. This procedure requires multiple (20-30) captures to run properly.

See Appendix A for more information on qcaldc.

6.5 Calibration verification

The calibration tool can also be operated in depth verification mode. In this mode, the tool receives another pair of captures of the charts, assuming the same device is used for capture.

The same configuration file should be used with the following updates:

- The updated paths for the input main and auxiliary image captures
- The distance between the camera and the middle chart during the updated capture
- The focal length ratio used during the updated capture

The tool applies the calibration data to rectify the images, and evaluates the quality of alignment, using the same metrics used by qcaldc during the calibration.

If any of the validation metrics exceed the pre-defined thresholds, depth verification fails. The failed module should be pulled from the production line for inspection. Logs should be sent to QTI for inspection.

Use the following command line to run the tool in verification mode:

```
qcaldc.exe -v <calibration configuration file>
```

qcaldc prints the following message with some statistics:

```
===========================
qcaldc version: 3.0
Verification mode
===========================
Parsing data
===========================
Splitting images
===========================
Calibration verification results ...
         Metric                      Value  Threshold Result
Residual vertical disparity [%]      0.08       0.20   PASS
Scale difference [%]                 0.32      10.00   PASS
STDEV of Horizontal Disparity        1.58      10.00   PASS
Maximum Vertical Disparity [pixels]  3.54      15.00   PASS
Distance error [%]                   0.87      20.00   PASS
===========================
Saving projected images
```

Running in verification mode (as well as in debug mode) will result in additional output files. See further details in A.1.5.

For analysis or debugging, redirect qcaldc output to a log file:

```
qcaldc -v qcaldc.cfg > qcaldc.log
```

6.6 QTI verification

Send calibration and verification data along with the corresponding images for at least 15 modules to QTI for validation. This is required when calibrating modules for a new project or making changes to an existing QTI-validated calibration procedure setup.

Provide the following information for each module:

- Calibration configuration file
- Main and auxiliary 3D chart calibration images
- Calibration output files (some may require running in debug mode)
  - OTP file
  - Corner detection images (main/aux_img_w_corners.jpg)
  - Green marker detection images (marker_main/aux.jpg)
  - Rectified images side-by-side (image_comparison.jpg)
- Execution log (qcaldc.log) from the qcaldc tool

Based on analysis of this data, QTI will provide feedback and guidance on mass production stage.

A qcaldc software package

Qualcomm Calibration for Dual Camera (qcaldc) is a command-line Windows program that calibrates dual camera sensor modules and generates the binary calibration data necessary for aligning the images from the two sensors. This mode of operation is called calibration mode. qcaldc can also operate in depth verification mode, wherein it verifies the quality of calibration for a given module.

The images of the 3D calibration chart from each sensor and the configuration file serve as qcaldc inputs. A binary file containing calibration data is generated as output. qcaldc checks the dual camera sensor module for vertical misalignment, scale, and predicted distance accuracy allowances. Based on the allowances, qcaldc qualifies a module as Pass or Fail.

The software package includes:

- qcaldc.exe – Frontend executable
- qcaldc.dll – Backend QTI proprietary calibration software library
- qcaldc.cfg – Sample calibration configuration file
- qcaldc_chart_w_markers.png – Recommended example for chart A
- qcaldc_chart.png – Recommended example for charts B and C
- ref_.png – Sample main sensor calibration image
- aux_.png – Sample auxiliary sensor calibration image

See A.1 for a detailed description of the configuration file parameters.

Running the tool (e.g. `qcaldc.exe qcaldc.cfg`) will produce the following outputs:

- OTP data – a binary file with all the calibration results, as defined in Table A-2. The file name is determined by `calib_otp_data_file` configuration parameter
- otp.yaml – A yaml file containing the intrinsic and extrinsic matrices and distortion parameters of the main and auxiliary cameras
- Yield statistics – The aggregated calibration statistics since this file was last deleted. The file name is determined by `calib_statistics_file` configuration parameter

The tool can also be run in debug mode (see Appendix A.1.3) and Full calibration mode (see Appendix A.1.4).

A.1 qcaldc.cfg parameters

The qcaldc configuration file requires the parameters described in Table A-1.

Table A-1 qcaldc configuration parameters

Parameter name	Description
`main_image_file`	Main image file name (PNG, JPG or BMP format)
`aux_image_file`	Aux image file name (PNG, JPG or BMP format)
`main_image_crop_0..3`	Main image crop with respect to the full sensor image **without scaling** (in pixels) [star_x,start_y,width,height]
`main_image_scale`	Downscaling factor from sensor resolution to the resolution of the image processed during calibration
`estimate_intrinsic_params`	Estimate the main/aux intrinsic parameters automatically from the input images ▪ 0 – Use the main/aux_camera_matrix params ▪ 1 – Estimate the main/aux intrinsic matrices
`main_camera_matrix_0..8`	Main camera matrix coefficients
`main_focal_length_mm`	Focal length at infinity in mm
`main_focal_length_ratio`	Ratio of the focal length during capture and the focal length at infinity of the main sensor. Set to 1 if camera has a fixed focus, or if capture is taken when focus is at infinity
`main_distortion_correction_method`	Select the main camera lens distortion correction method ▪ 0 – General distortion model (Radial + Tangential) ▪ 1 – Fisheye distortion model ▪ 2 – Use a specific lens distortion model table
`main_distortion_coeff_0..5`	Main camera distortion coefficients (k1, k2, p1, p2, k4, k5)
`main_ldc_lut_file`	The file name holding the main camera lens distortion model
`main_ldc_grid_file`	The file name holding the main camera distortion correction grid
`main_sensor_mono_flag`	Indicates whether main sensor is color or monochrome: ▪ 0 – color sensor ▪ 1 – Monochrome sensor
`main_flip_setting`	Main sensor settings during calibration: ▪ 0 – No flip ▪ 1 – Horizontal flip ▪ 2 – Vertical flip ▪ 3 – both horizontal and vertical flip
`main_aux_position_flag`	Indicates which sensor is left and which is right ▪ 0 – Main is left of the auxiliary ▪ 1 – Main is right of the auxiliary
`main_aux_rotate_setting`	Indicates how the image is rotated by *qcaldc* before calibration: ▪ 0 – No rotation or flip ▪ 1 – 90° rotation ▪ 2 – 180° rotation ▪ 3 – -90° rotation

Parameter name	Description
`aux_image_crop_0..3`	Aux image crop with respect to the full sensor image **without scaling** (in pixels) [star_x,start_y,width,height]
`aux_image_scale`	Scale ratio between aux image size used in calibration and the aux sensor image size
`aux_camera_matrix_0..8`	Aux camera matrix coefficients
`aux_focal_length_mm`	Focal length at infinity in mm
`aux_focal_length_ratio`	Ratio of the focal length during capture and the focal length at infinity of the aux sensor. Set to 1 if camera has a fixed focus, or if capture is taken when focus is at infinity
`aux_distortion_correction_method`	Select the aux camera lens distortion correction method ▪ 0 – General distortion model (Radial + Tangential) ▪ 1 – Fisheye distortion model ▪ 2 – Use a specific lens distortion model table
`aux_distortion_coeff_0..5`	Aux camera distortion coefficients (k1, k2, p1, p2, k4, k5)
`aux_ldc_lut_file`	The file name holding the aux camera lens distortion model
`aux_ldc_grid_file`	The file name holding the aux camera distortion correction grid
`aux_sensor_mono_flag`	Indicates whether aux sensor is color or monochrome: ▪ 0 – color sensor ▪ 1 – Monochrome sensor
`aux_flip_setting`	Auxiliary sensor settings during calibration: ▪ 0 – No flip ▪ 1 – Horizontal flip ▪ 2 – Vertical flip ▪ 3 – both horizontal and vertical flip
`chart_width`	Number of squares in the checker board horizontally
`chart_height`	Number of squares in the checker board vertically
`chart_square_size_mm`	Length in mm of the checkerboard square side
`marker_detection_method`	Type of marker used for detecting the line between the charts ▪ 0 – Green markers ▪ 1 – Aruco markers
`main_marker_threshold`	The minimal value to match the green marker for the Main image
`aux_marker_threshold`	The minimal value to match the green marker for the Aux image
`cam_baseline_mm`	Baseline distance in mm between the cameras optical centers
`chart_distance_mm`	Physical distance in mm between module and the middle chart during calibration
`calib_otp_data_file`	File name of binary one-time programmable (OTP) data
`calib_statistics_file`	File name of aggregated statistics of previous calibrations
`residual_vertical_disparity_ threshold_percent`	Threshold for residual vertical disparity between main and aux images after applying projective transform, in percent of the calibration image height (after scaling)
`main_fl_ratio_tolerance_percent`	Max Main FLR variation from theoretical value [default is 0.5].
`aux_fl_ratio_tolerance_percent`	Max Aux FLR variation from theoretical value [default is 0.5].
`scale_difference_threshold_percent`	Threshold for scale difference of vertical lines in main and auxiliary images, in percentage
`chart_distance_threshold_percent`	Threshold for chart distance error, in percentage

Parameter name	Description
`chart_horizontal_STD_threshold`	Threshold for standard deviation of horizontal disparity
`vertical_disparity_max_threshold`	Threshold for maximum vertical disparity
`horizontal_disparity_max_threshold`	Threshold for maximum horizontal disparity

A.1.1 Calculating focal_length_ratio

Calculate `main_focal_length_ratio` and `aux_focal_length_ratio` using the following steps per camera (unless the camera has a fixed focus):

1. Record the DAC values at infinity (DAC_inf) and macro (DAC_macro) for the module being calibrated. Also, record the total lens shift range between macro and infinity lens positions:

$$lens\_shift\_range\_in\_mm = \mathrm{abs}(macro\_lens\_shift\_mm - infinity\_lens\_shift\_mm)$$

2. Follow instructions in Section 6.4, and set the focus to the center of the middle chart.
3. Record the DAC value (*DAC_curr*) after the focus is complete and the lens has settled.
4. Compute the lens shift in mm between the infinity and the current lens position:

$$lens\_shift\_in\_mm = \frac{lens\_shift\_range\_in\_mm * (DAC\_curr - DAC\_\mathrm{inf})}{(DAC\_macro - DAC\_\mathrm{inf})}$$

5. Compute the focal length ratio:

$$\mathrm{focal\_length\_ratio} = \frac{focal\_length\_at\_infinity\_mm + lens\_shift\_in\_mm}{focal\_length\_at\_infinity\_mm}$$

If the focal length ratio does not change linearly between Macro and Infinity focus points, it is recommended to capture several points for intermediate lens positions, and recalculate the matching focal length ratios.

A.1.2 Setting the cameras intrinsic parameters

If the cameras intrinsic parameters are available, either from the lens vendor, or through a previous camara calibration, those parameters can be input to the tool rather than letting the tool estimate them.

The intrinsic parameters are set by the following parameters in the configuration file:

- `main/aux_camera_matrix_11` – The focal length in pixels widths
- `main/aux_camera_matrix_22` – The focal length in pixels heights
- `main/aux_camera_matrix_13` – The principal point horizontal location in pixel widths
- `main/aux_camera_matrix_23` – The principal point vertical location in pixel heights
- `main/aux _camera_matrix_33` – Set to 1
- `main_camera_matrix_12/21/31/32` – Set to 0

The most cases the module/lens vendor can provide the lens focal length, the sensor pixels pitch, and the principal point location on the sensor.

If we designated the Intrinsic camera matrix as IM, then:

$$IM[11] = \frac{focal\_length\_in\_mm}{pixel\_width\_in\_mm} \quad ; \quad IM[22] = \frac{focal\_length\_in\_mm}{pixel\_height\_in\_mm}$$

A.1.3 Calculating marker threshold

The green markers are used for helping the tools identify the margins between the main and side charts. Any green shade should work, but in case the tool misses the markers, the exact color perceived by the cameras can be extracted using the `-debug` switch, which will cause the tool to output the image after being filtered by a green enhancing filter, as shown in Figure A-1. The user should set the `chart_marker_threshold` configuration parameter to a value slightly below the extracted marker value.

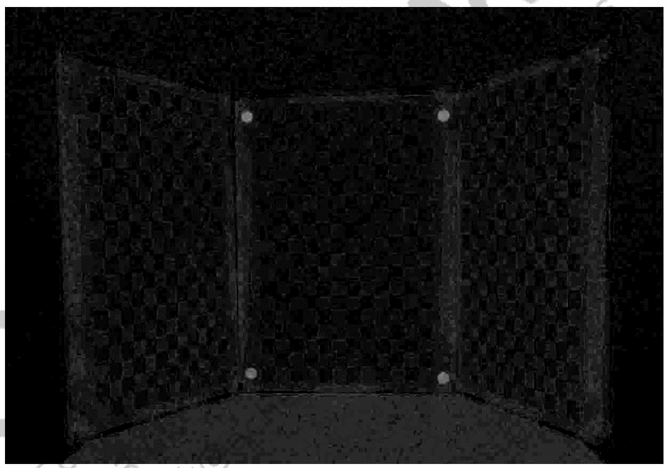


Figure A-1 Green enhanced chart image dump

A.1.4 Lens distortion correction

In order to optimize the calibration accuracy, qcaldc supports several methods for dealing with lens distortion. The tool can consume a table (provided by the camera module vendor) which defines the lens distortion per radial distances between the center and the edge of the field of view. Alternatively, it can consume the distortion coefficients based on OpenCV lens distortion model. If the distortion information is not available, the tool will try to estimate the distortion based on the captured image of the 3D chart.

The following configuration parameters control the lens distortion estimation and correction:

- `estimate_intrinsic_params` – Selects whether to use the camera intrinsic parameters as given in the `main/aux_camera_matrix_0..8` parameters, or try to optimize them further. This parameter affects both the main and aux camera's intrinsic parameters.
- `main/aux_distortion_correction_method` – Selects whether to use OpenCV models for estimating the main/aux camera lens distortion, or to use the lens distortion table provided by the camera module vendor

- `main/aux_ldc_lut_file` – Pointing to the file holding the lens distortion information

The lens distortion table should be a text file with one or more columns. Only the last column is used, and each value should indicate the percentage of distortion of the radial distance from the principal point per a specific angle. It is assumed that the ray angles cover the entire field of view, and that they are evenly spaced from principal ray to the maximal field of view.

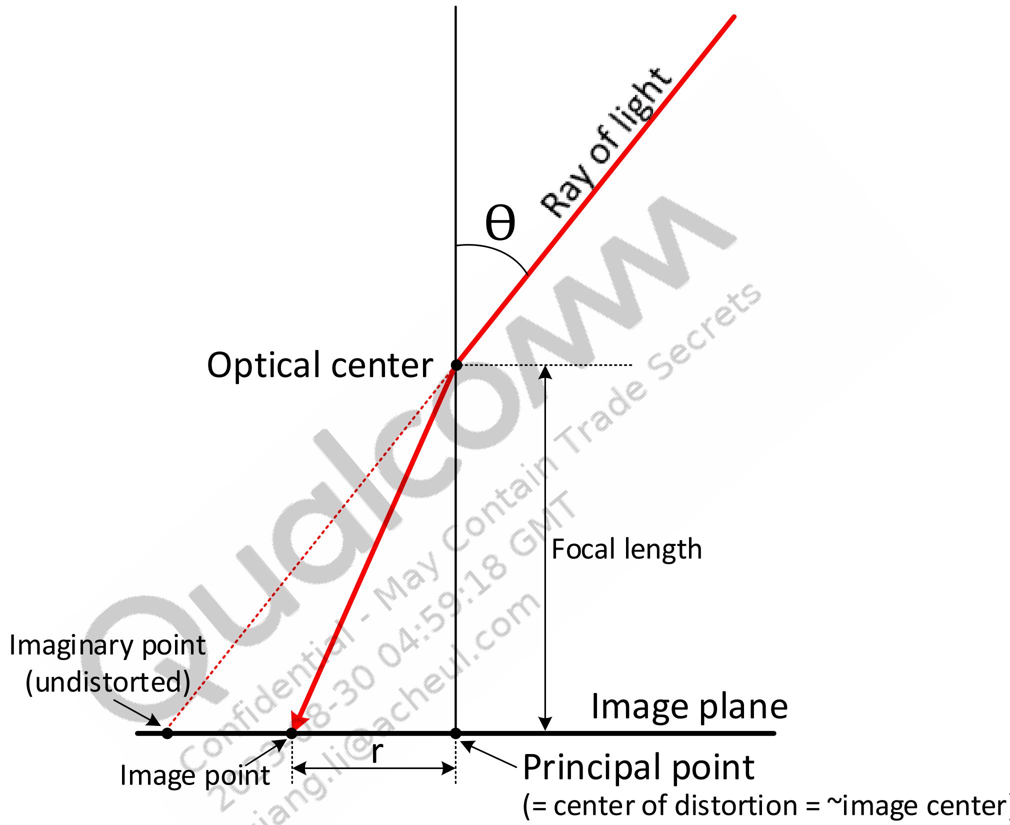


Figure A-2 Lens distortion F-Theta model

The following table shows an example of how the lens model can be represented.

FOV	Angle (°)	Real Height	Distortion (%)
0	0	0.0000	0.0000
0.1	3.25	0.4474	-0.1028
0.2	6.5	0.8949	-0.4213
0.3	9.75	1.3421	-0.9760
0.4	13	1.7881	-1.7999
0.5	16.25	2.2314	-2.9390
0.6	19.5	2.6690	-4.4414
0.7	22.75	3.0979	-6.3344
0.8	26	3.5157	-8.6082
0.9	29.25	3.9217	-11.2148
1	32.5	4.3163	-14.0992

Figure A-3 Lens distortion model example

It is recommended to use the lens distortion model if it can be provided by the camera module vendor. It is assumed to be the most accurate method and, if selected, the distortion coefficients indicated in the configuration file are ignored.

If the selected distortion correction method was to use the distortion coefficients, the tool would try to optimize them during the calibration process, and output the updated distortion coefficient in the output parameters file.

If the distortion coefficients are not available as well, it is required to set them all to zeros, and the tool will try to estimate them.

The following output parameters are affected by the selected distortion correction method:

- `ica_grid_file` – If the lens distortion model is used, qcaldc will output an XML file which can later be used by the camera software to apply the lens distortion. This parameter specifies the XML file name.
- `main/aux_distortion_model` – Indicates which model was used for lens distortion correction: regular, fisheye, or lens specific model

A.1.5 Running in Debug mode

In order to run the tool in debug mode, enter the following command:

```
qcaldc.exe -d <calibration configuration file>
```

Running in Debug mode (as well as in Verification mode) will result in the following additional output files:

- Rectified main and auxiliary images
- A side-by-side comparison of the rectified main and auxiliary images with horizontal guide lines. See example in Figure A-4.
- Main and auxiliary images after green markers filter. See example in Figure A-4.
- Main and auxiliary images with identified checkerboard corners. See example in Figure A-5.
- Output parameters text file. Contains the data generated for OTP in text format

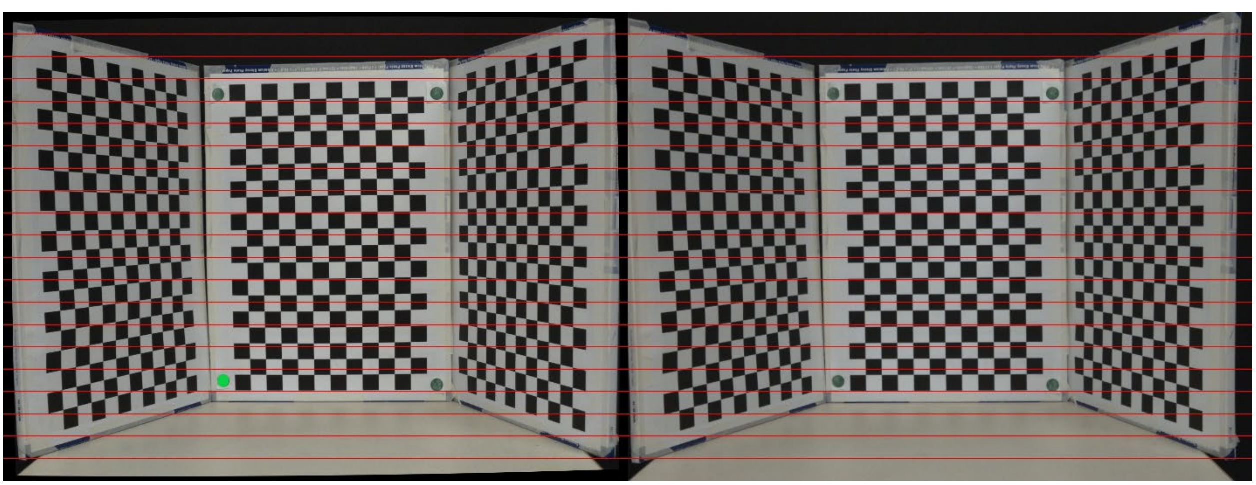


Figure A-4 Rectified images with guidelines

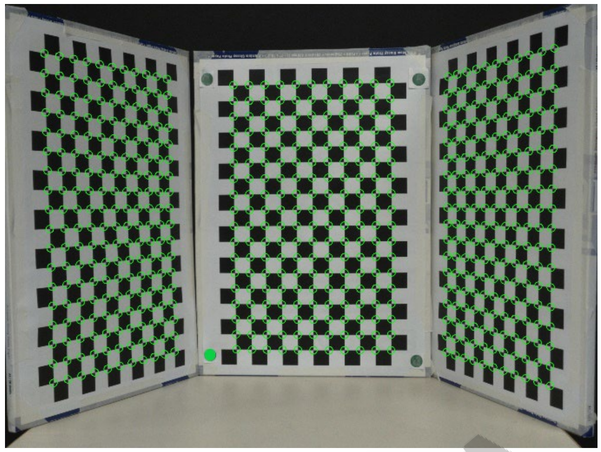


Figure A-5 Main Image with detected checkerboard corners

Debug mode run will also produce the OTP data in binary and text format and the otp.yaml file with the camera matrices.

A.1.6 Running in Full Calibration mode

In order to run the tool in full calibration mode, enter the following command:

```
qcaldc.exe -f <calibration configuration file>
```

As opposed to the regular calibration mode, in full mode the tool will run a full OpenCV calibration procedure that requires 20-30 captures of both cameras. The tool will calculate both cameras intrinsic and extrinsic parameters, so they do not have to be set in the configuration file.

Only a minimal configuration file is needed:

- main_image_path = Folder name of main camera captures
- aux_image_path = Folder name of auxiliary camera captures

- main_image_prefix = Main file names prefix (e.g., *_main)
- aux_image_prefix = Aux file names prefix (e.g., *_aux)

- main_image_format = jpg, png, bmp, etc.
- aux_image_format = jpg, png, bmp, etc.

- calib_otp_data_file = Cameras matrices output file name
- chart_square_size = Checkerboard square size in mm

- chart_size_width = Number of internal vertices on the checkerboard horizontal lines
- chart_size_height = Number of internal vertices on the checkerboard vertical lines

- image_to_project = the number of the input frame pair which is rectified and saved

The output file (in YAML format) includes the following information:

- M1, M2 – The camera matrices with the cameras intrinsic parameters
- Distcoeff1/2 – The undistortion coefficients of a 2D polynomial describing the lens distortion per camera
- R1,R2 – 3D rotation matrix per camera for rectification
- P1, P2 – The new camera matrices of the virtual cameras after rectification

A.2 qcaldc output metrics

The following validation metrics are evaluated and recorder during qcaldc run:

- Residual vertical disparity [%] – A*verage* residual vertical disparity between the rectified main and auxiliary images.
- Scale difference [%] – Vertical scale difference between the rectified main and auxiliary images
- Standard deviation of horizontal disparity [pixels] – Standard deviation of the horizontal disparity across the image between the rectified main and auxiliary images
- Maximum vertical disparity [pixels] – Maximal residual vertical disparity between the rectified main and auxiliary images
- Distance error [%] – Error in the predicted distance to the center of the chart
- Minimal horizontal disparity [pixels] – TBD
- Horizontal & vertical disparity maps – TBD
- FL adjustment knee – TBD
- FLR distortion adjustment points - TBD

If any of the validation metrics fails, calibration fails and qcaldc will not produce a binary output. The failed module should be pulled from the production line for inspection. Logs should be sent to QTI for inspection.

If all validation metrics pass, calibration succeeds and qcaldc produces a binary file containing the calibration data. Data fields are written in little-endian byte order and the name of the output file is determined by the `calib_otp_data_file` parameter in the qcaldc configuration file.

Table A-2 Output parameters

Param name	Type	Start byte	Description
`stereo_calib_version`	char	0	Tool version
`stereo_calib_sub_version`	char	1	Tool sub-version
`main_image_crop_0..3`	float[4]	2	Main image crop with respect to the full sensor image **without scaling** [star_x,start_y,width,height]

Param name	Type	Start byte	Description
`main_image_scale`	float	18	Scale ratio between main image size used in calibration and the cropped sensor image size
`main_focal_length_mm`	float	22	Main focal length when focus is set to infinity
`main_camera_matrix_0..8`	float[9]	26	Main original camera matrix
`main_new_camera_matrix_0..8`	float[9]	62	Virtual main camera matrix in the rectified domain
`main_to_rect_rotation_0..8`	float[9]	98	3D rotation matrix from the original main camera coordinate to the rectified domain coordinates
`main_to_aux_rotation_0..8`	float[9]	134	3D rotation matrix from the original main camera coordinate to the auxiliary camera coordinates
`main_to_aux_translation_0..2`	float[3]	170	[T_x,T_y,T_z] translation vector from main camera optical center to auxiliary camera optical center
`main_dist_coeffs_0..7`	float[8]	182	Main camera distortion coefficients (k1, k2, p1, p2, k3, k4, k5, k6)
`main_distortion_model`	char	214	The distortion model used for representing the lens distortion 0 – General (Radial + Tangential) 1 – Fisheye 2 – Lens specific model
`main_flip_setting`	char	215	Main sensor settings during calibration: ▪ 0 – No flip ▪ 1 – Horizontal flip ▪ 2 – Vertical flip ▪ 3 – both horizontal and vertical flip
`main_aux_position_flag`	char	216	Indicates which sensor is left and which is right ▪ 0 – Main is left of the auxiliary ▪ 1 – Main is right of the auxiliary
`Main_aux_rotate_setting`	char	217	Indicates how the image is rotated by *qcaldc* before calibration: ▪ 0 – No rotation or flip ▪ 1 – 90° rotation ▪ 2 – 180° rotation ▪ 3 – -90° rotation
`aux_image_crop_0..3`	float[4]	218	Aux image crop with respect to the full frame **without scaling** [star_x,start_y,width,height]
`aux_image_scale`	float	234	Scale ratio between aux image size used in calibration and the aux sensor image size
`aux_focal_length_mm`	float	238	Aux focal length when focus is set to infinity
`aux_camera_matrix_0..8`	float[9]	242	Aux original camera matrix
`aux_new_camera_matrix_0..8`	float[9]	278	Virtual aux camera matrix in the rectified domain
`aux_to_rect_rotation_0..8`	float[9]	314	3D rotation matrix from the original aux camera coordinate to the rectified domain coordinates
`aux_dist_coeffs_0..7`	float[8]	350	k1, k2, p1, p2, k3, k4, k5, k6 in the stereo params struct
`aux_distortion_model`	char	382	The distortion model used for representing the lens distortion 0 – General (Radial + Tangential)

Param name	Type	Start byte	Description
			1 – Fisheye 2 – Lens specific model
`aux_flip_setting`	char	383	Aux sensor settings during calibration: ▪ 0 – No flip ▪ 1 – Horizontal flip ▪ 2 – Vertical flip ▪ 3 – both horizontal and vertical flip
`cam_baseline_mm`	float	384	Distance between the cameras focal points
`cam_distance_to_chart`	float	388	Distance to chart during capture
`mean_vertical_disparity`	float	392	Mean vertical disparity after rectification
`scale_error_at_calibration`	float	396	Scale error at calibration
`distance_error_percent`	float	400	Distance error percentage
`std_horiz_disparity`	float	404	Standard deviation of horizontal disparity on the middle chart
`max_vertical_disparity`	float	408	Maximum vertical disparity after rectification
`rms_vert_disparity_map_0..8`	float[9]	412	RMS map of vertical disparity on 3x3 grid zones in the image
`rms_horiz_disparity_map_0..8`	float[9]	448	RMS map of the horizontal disparity on 3x3 grid zones on the image
`95th_vert_disparity_percentile`	float	484	95th percentile of vertical disparity
`90th_vert_disparity_percentile`	float	488	90th percentile of vertical disparity
`80th_vert_disparity_percentile`	float	492	80th percentile of vertical disparity
`95th_horiz_disparity_percentile`	float	496	95th percentile std of horizontal disparity
`90th_horiz_disparity_percentile`	float	500	90th percentile std of horizontal disparity
`80th_horiz_disparity_percentile`	float	504	80th percentile std of horizontal disparity
`main_fl_adjust_knee_0..4`	float[5]	508	Focal length ratio anchor points for scale adjustment
`main_flr_distortion_adjust_0..4`	float[5]	528	Scale adjustment factor due to distortion at specified focal length ratio anchor point
`aux_fl_adjust_knee_0..4`	float[5]	548	Focal length ratio anchor points for scale adjustment
`aux_flr_distortion_adjust_0..4`	float[5]	568	Scale adjustment factor due to distortion at specified focal length ratio anchor point
Reserved	float[16]	588	Reserved parameters

B AEC sync calibration sample code

The sample code provided here is to demonstrate how to calculate average luma and Q number conversion. Notice that, the raw data format is assumed to be 16-bit unpacked, and there's no read from file in this sample code. Vendor should take care of correct raw file read and data format conversion. Another assumption is the black_level is provided by vendor since anyway this value will be obtained during other sensor calibration.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <math.h>
float CalculateRGAverage(uint16_t* master_raw, int width, int height, int
black_level)
{
        long int sum = 0;
        // Set up height start, range, and end
        int h_start = height * 7 / 16;
        int h_range = height / 8;
        int h_end = h_start + h_range;

        // Set up width start, range, and end
        int w_start = width * 7 / 16;
        int w_range = width / 8;
        int w_end = w_start + w_range;

        for (int j = h_start; j<h_end; j++)
        {
                for (int i = w_start; i<w_end; i++)
                {
                        // Pick only RG pixel
                        if (j % 2 == 0 && i % 2 == 0)
                        {
                                int v = master_raw[j*width + i];
                                if (v>black_level)
                                        sum += (v - black_level);
                        }
                }
        }

        // average area is 1/4 of the original (RG pixel only)
        float area = w_range*h_range * 1 / 4;

        return (float)sum / area;
}
uint16_t* ReadUint16RawData(const size_t raw_width, const size_t raw_height)
{
        uint16_t* p = (uint16_t*)malloc(sizeof(uint16_t)*raw_width*raw_height);
```

```
	uint16_t* pp = p;

	for (size_t i = 0; i<raw_width*raw_height; i++)
	{
		*pp = 820; //20% of 12-bit, just an example. Vendor should read real
raw data from file
		pp++;
	}

	return p;
}

uint16_t FloatToQ10(const float v)
{
	float t = v * 1024; //2^10
	return (uint16_t)round(t);
}

float Q10ToFloat(const uint16_t v)
{
	float t = v;
	return t / 1024;
}

int main(int argc, char* argv[])
{
	const size_t raw_width = 4208;
	const size_t raw_height = 3120;

	//The ReadUint16RawData should return unpacked raw data
	// where every 2 bytes represents 1 pixel

	// Notice the average calculation doesn't take care of endian,
	// if the endian is different, please modify the pixel access accordingly

	uint16_t* raw = ReadUint16RawData(raw_width, raw_height);
	//  Assume the raw pixels are in 12 bits.

	int black_level = 15;
	float gr_avg = CalculateRGAverage(raw, raw_width, raw_height, black_level);

	// Since the pixel is in 12 bits, scale the luma to 16 bits
	uint16_t average_luma = (uint16_t) (gr_avg / (pow(2, 12) - 1) * (pow(2, 16)
- 1));
	printf("average_luma is: %d\n", average_luma);

	return 0;
}
```

C References

C.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Camera Module Selection and Calibration Data	80-N5126-1

C.2 Acronyms and terms

Acronym or term	Definition
AF	Auto focus
BPP	Bits per pixel
CFA	Color filter array
DAC	Digital to analog converter
FOV	Field of view
FLR	Focal length ratio
IAF	Instant AF
LCA	Logic cell array
NVM	Nonvolatile memory
OTP	One-time programmable
VCM	Voice coil motor