Electronic Image Stabilization

Tuning Guide

80-ND928-8 Rev D

January 10, 2020

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

Revision history

Revision	Date	Description
A	July 2018	Initial release.
B	July 2018	Added Figure 4-1. Updated Table 4-3.
C	August 2019	Updated Chapter 4.
D	January 2020	Updated Chapter 4 with partial LDC and logging. Updated Chapter 6 with testing steps.

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document is for camera software and testing teams who use Qualcomm's electronic image stabilization (EIS) feature. The document assumes that the intended audience has a firm understanding of Android camera architecture.

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>`.

1.3 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

2 Overview

2.1 Feature description

Electronic image stabilization (EIS) is an image enhancement technique using electronic processing. EIS minimizes blurring and compensates for device shake, often a camera. More technically, this technique is referred to as pan and slant, which is the angular movement corresponding to pitch, yaw, and roll.

2.2 High level architecture

EIS takes in gyroscope sensor data to calculate the camera rotation. The rotation angle is then calculated into a mesh table, the so called dewarp mesh. As an example, the dewarp mesh could have 32 x 32 entries with each entry storing the coordinates of the input image. The dewarp mesh is sent to GPU together with input frame buffers from the ISP. The GPU does a table lookup and, for each output pixel, it looks through the dewarp mesh to get the coordinate of input frame. Then the GPU fetches the pixel from the input frame by the coordinates. The coordinates could be decimal. A decimal coordinate means that the pixel value comes from an interpolation of its neighbor pixels.

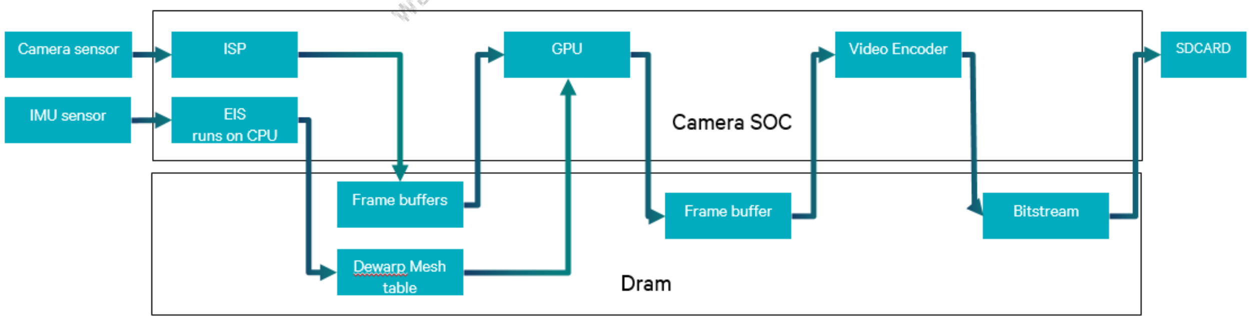

Figure 2-1 System architecture of camera EIS

2.3 EIS functional blocks

The EIS functional blocks comprise the following four parts:

- Frame-based image stabilization (FBIS)– Affine transformation and projection based on the entire frame. FBIS only partially compensates vibration. It compensates the high frequency part and leaves the low frequency vibration, resulting in smooth camera movement.
- User defined effect (UDE)– Image warping effects defined by UI or application.
- Rolling shutter correction (RSC)– Compensates rolling shutter effect caused by different exposure start time. Unlike FBIS, RSC compensate 100% of vibration.
- Lens distortion correction (LDC)– Supports both fisheye lens and pinhole lens.

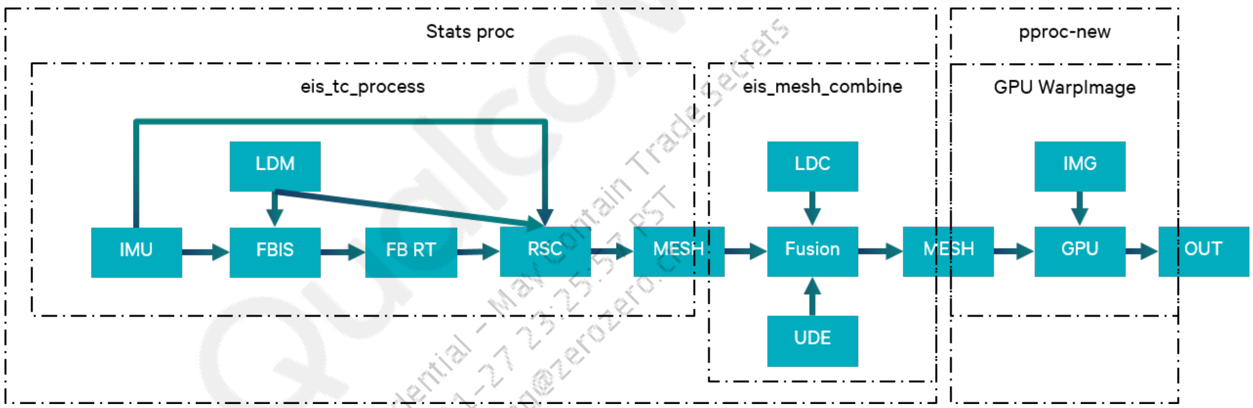

Figure 2-2 EIS function blocks

2.4 EIS integration in mm-camera

The EIS algorithm that calculates the dewarp mesh from the gyroscope data is provided in form of binary, Yocto Linux ABI, and Android native ABI. The EIS algorithm library is integrated to the IS module of the STATS module of the mm-camera framework.

The GPU dewarping functional block is provided as source code. It is integrated into the IMGLIB module of the mm-camera framework.

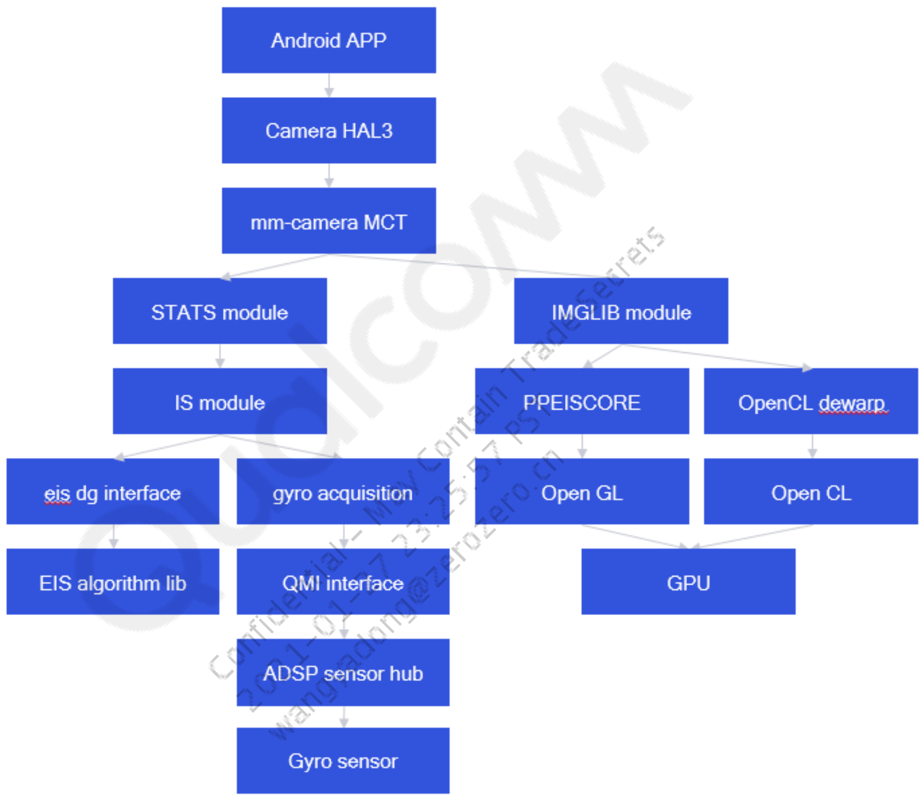

Figure 2-3 Software architecture of EIS

The EIS algorithm library and GPU dewarping run in different threads and are trigged by external events. The IS module, which controls multiple EIS algorithms like EIS2, EIS3, and Digital Gimble, is triggered by VFE stats interrupt, which is nearly the last valid pixel (excluding vertical blanking).

The IS module then generates the gyroscope data query command to QMI followed by the sensor hub ADSP. A period is specified in the query command. Once the senor hub collocates the gyroscope data within the given period before the timeout, it generates a gyroscope data ready event to the IS module. Then the IS module calls the EIS algorithm to calculate the dewarp mesh. The call to the EIS algorithm library is a blocking call. It takes 7ms~20ms depending on system loading and CPU capability. After the EIS library returns with a dewarp mesh, the IS module calls IMGLIB to save the dewarp mesh in IMBLIB space.

The IMGLIB is triggered by frame buffer ready from an upstream module, like VFE. Once triggered, the IMGLIB wraps the image buffer together with the dewarp mesh into several OpenGL or OpenCL commands to the GPU. The graphic command is queued in the GPU driver and is triggered in sequence.

Once the GPU complete a dewarp processing, it generates a callback event to notify the downstream modules.

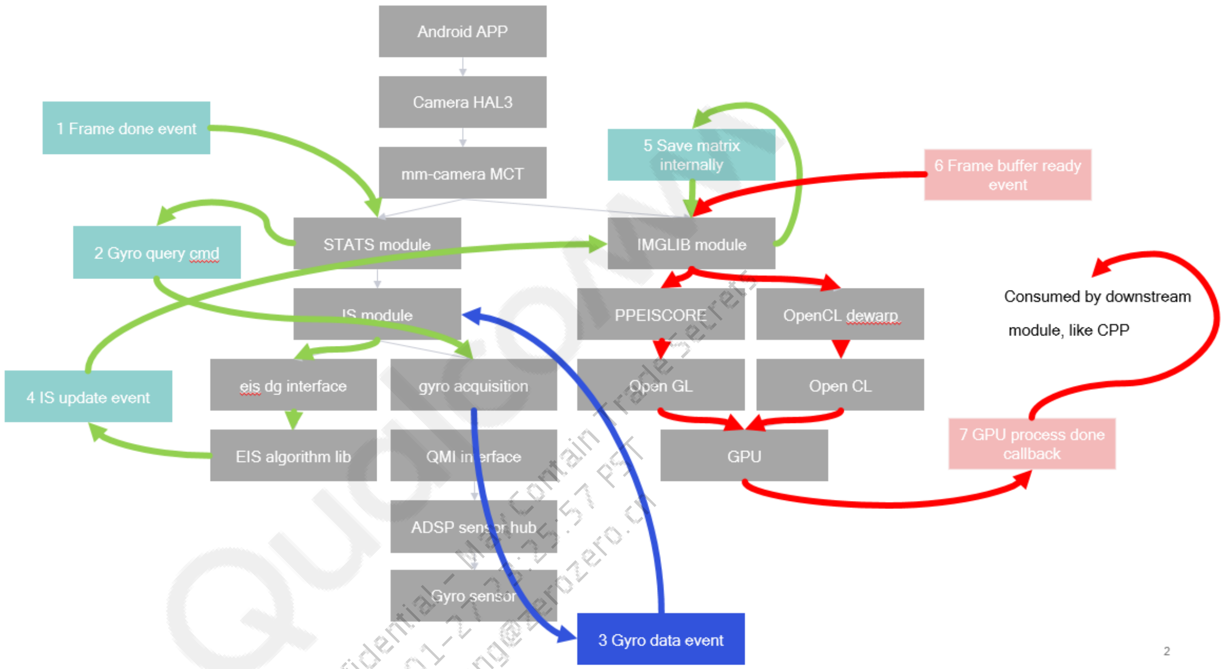

Figure 2-4 Asynchronized call between EIS and image dewarping.

The arrows in Figure 2-4 represent the following:

- Green: IS module event and command flow
- Red: IMGLIB module event and command flow
- Blue: Gyroscope data ready event generated from the sensor hub

2.5 Auto bias correction and auto centering

Given the fact that a MEMS based gyroscope sensor usually gives bias when the gyroscope sensor is stationary, the EIS algorithm internally has preprocessing on the gyroscope raw data. The preprocessing includes two modules. The first module is auto bias correction and the second module is auto centering.

Auto bias correction

The auto bias correction is a simple, first-order infinite impulse response (IIR) low pass filter. This is sometimes called an exponential moving average (EMA) in some places. The raw gyroscope data is feed into the IIR to generate the bias estimation and then the raw gyroscope data is subtracted by the estimated bias.

There are two tuning parameters for auto bias correction: the clamping threshold and the IIR coefficient.

Auto centering

The purpose of the auto centering is to reset the entire EIS cropping window back to image center when it detects severe motion. The assumption is that when there is severe motion, the EIS is likely to run into margin breach case. Because of this, the EIS will not be able to stabilize the image. Auto centering makes no difference to the current frame. But to the next frames, the centered ROI gives more room to compensate small or moderate amounts of motion.

The severity of motion is defined by a single threshold for each axis. When the frame-to-frame rotation is over the threshold, the auto centering is triggered ***per axis***.

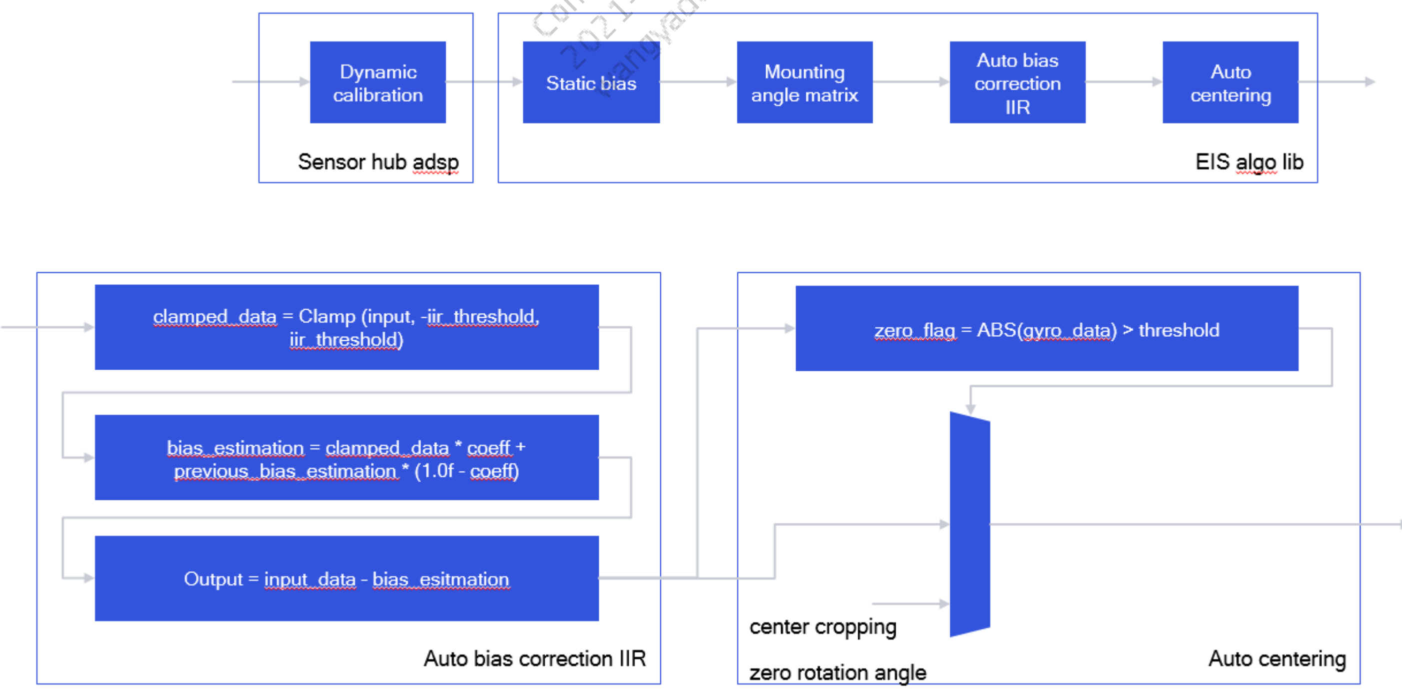

Figure 2-5 Data flow of auto bias correction and auto centering

3 Hardware requirements

The gyroscope sensor must be on the same PCB as the camera CMOS sensor. They do not have to be close. Because the MEMS sensor is sensitive to external pressure, ensure that the industrial design does not contact the MEMS sensor or cause pressure to it.

4 Tuning parameters

Most EIS tuning parameters are stored in the IOT chromatix header. For example:

mm-camera

```
caf\camera\services\mm-camera\mm-camera2\media-
controller\modules\sensors\chromatix\0309\chromatix_imx458\iot\video_4k\
chromatix_imx458_video_4k_iot.h
```

CAMX

```
chi-
cdk\vendor\tuning\imx577_c7project\Scenario.Default\XML\Postproc\ioteis10.x
ml
chi-cdk\cdk\chromatrix\XML\Postproc\ioteis10.xml
```

Lens focal length and sensor mounting angle are stored in the sensor driver XML. For example:

```
caf\camera\services\mm-camera\mm-camera2\media-
controller\modules\sensors\configs\msm8953_camera.xml
```

Sensor pixel size is stored in the sensor driver header. For example:

```
caf\camera\services\mm-camera\mm-camera2\media-
controller\modules\sensors\sensor\libs\imx458\imx458_lib.h
```

See Table 4-1, Table 4-2, and Table 4-3 on the following pages for a list and description of the EIS tuning parameters.

Table 4-1 EIS tuning and calibration parameters

mm-camera

Type	Data field	Sub data field	Meaning	Need per module calibration?	Needs tuning?	Typical value	Range
mvDG_mesh_t	ldc_mesh		Normalized warp map for LDC or partial LDC.	No	Use case dependent	Use case dependent	N/A
float*	ldc_inv_r2_lut		The 1D lens distortion model LUT for partial-LDC.	No	Use case dependent	Use case dependent	N/A
float*	ldc_r2_lut		The 1D lens distortion model LUT.	Better to have	Yes	Lens dependent	N/A
Int	ldc_inv_r2_lut_size		Scale of partial-LDC model LUT.	No	No	Lens dependent	N/A
Int	ldc_r2_lut_size		Scale of lens distortion model LUT.	No	No	Lens dependent	N/A
mvDG_imu_calib_t	imu_calib	pixel_size_in_um	Pixel size in um	No	Yes	Camera sensor dependent; 1.55	1.0~3.0
		focal_length_in_mm	Focal length in mm	Better to have	Yes	Lens dependent; 1.2	1.0~3.0
		Rbc[9]	Rotation between gyro sensor and camera sensor	No	Yes	100,010,001	0.0 ~ 1.0
		acc_unit	Unit of accelerometer sensor	No	No	N/A	0~65535
		gyro_unit	Unit of gyro sensor	No	Gyro Sensor driver dependent	65535	0~65536
		reserved_unit	Reserved				
		Bias	Gyro static bias.	No	No	0	0~65535
		tmpt_bias_comp_en	Obsoleted				
		drift_slope	Obsoleted				
		drift_temp_knee	Obsoleted				

Type	Data field	Sub data field	Meaning	Need per module calibration?	Needs tuning?	Typical value	Range
		constant_delay_off_pts_in_us	Value will be added to gyro timestamps to solve timestamps mismatch	No	Yes	0	-2000 ~2000
		reserved	Reserved field				
mvDGTC_tuning_t	fbis_margin		margin for frame base IS.	No	Yes	1.1	1.0~2.0
	rsc_margin		margin for rolling shutter correction.	No	Yes	1.1	1.0~2.0
	tuning_param	filter_type	Obsoleted				
		ldc_en	Event flag to inform EIS about different LDC mode.0 – disable;1 – full LDC;2 – partial LDC;	No	Use case dependent	0	0~2
		filter_param[16]	Auto bias correction and auto centering tuning parameters. See another table				
		debug_mode	1: disable auto centering; disable auto bias correction; disable margin breach control. Repeated pixel could be seen at image boundary.	No	No	0	0 or 1
		rsc_debug	0 - RSC enabled 1- RSC disabled	No	No	0	0 or 1
		ldc_debug	0 - LDM enabled 1- LDM disabled, EIS takes camera as ideal pinhole camera	No	No	0	0 or 1
		horizon_level_correction	Reserved.	No	No	0	0
		mvDGTC_eis_strength_t eis_strength	EIS lowlight strength tuning parameters. See another table				
int	prior_ldc_present		If Lens distortion correction is enabled, this flag should be set. Otherwise this flag should be clear.	No	No	0/1	0 or 1

CAMX

DGTC							
Data field	**Sub data field**	**Type**	**Meaning**	**Needs per module calibration?**	**Needs tuning?**	**Typical value**	**Typical range**
<rotation>	X	float	Rotation between gyro sensor and camera sensor	X	O	$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$	-1.0 ~ 1.0
<imu_calib>	<pixel_size_in_um>	float	Pixel size in um	X	O	Camera sensor dependent	1.0 ~ 3.0
	<focal_length_in_mm>	float	Focal length in mm	Better to have	O	Camera lens dependent	1.0 ~ 3.0
	<acc_unit>	uint	Unit of accelerometer sensor	X	X	X	
	<gyro_unit>	uint	Unit of gyro sensor	X	Gyro sensor driver dependent	1	1 ~ 65536
	<constant_delay_off_pts_in_us>	int	Value will be added to gyro timestamps to solve timestamps error	X	O	0	-2000 ~ 2000
<imu_data_sampling_rate>	X	uint	IMU sampling rate in Hz	X	IMU sensor dependent	1600	500 ~ 3200
<fbis_margin>	X	float	Margin for frame-based image stabilization	X	O	1.1	1.0 ~ 2.0
<rsc_margin>	X	float	Margin for rolling shutter correction	X	O	1.1	1.0 ~ 2.0
<tuning_param>	<filter_param>	N/A	Tuning parameters for auto bias correction and auto centering.	X	O	N/A	N/A
	<eis_strength>	N/A	EIS lowlight strength tuning parameters.	X	O	N/A	N/A

LDC							
Data field	**Sub data field**	**Type**	**Meaning**	**Needs per module calibration?**	**Needs tuning?**	**Typical value**	**Typical range**
<num_ver_x>	X	uint	Number of vertex along x direction for mapping table	X	X	65	1 ~ 129
<num_ver_y>	X	unit	Number of vertex along y direction for mapping table	X	X	97	1 ~ 97
<input_width>	X	unit	The image width from which LDC mesh were generated	X	O	Camera sensor dependent	1 ~ 65535
<input_height>	X	uint	The image height from which LDC mesh were generated	X	O	Camera sensor dependent	1 ~ 65535
<output_width>	X	uint	The image width from which LDC mesh were generated. It should be the same as input.	X	O	Camera sensor dependent	1 ~ 65535
<output_height>	X	unit	The image height from which LDC mesh were generated. It should be the same as input.	X	O	Camera sensor dependent	1 ~ 65535
<mapping>	X	float	Normalized warp map for LDC.	X	Use case dependent	Camera lens dependent	0.0 ~ 1.0
<ldc_r2_lut_size>	X	float	original size of <ldc_r2_lut>	X	O	Camera lens dependent	0.0 ~ 65535.0
<ldc_r2_lut>	X	float	1D lens distortion model that maps pinhole to distorted image	X	O	Camera lens dependent	0.0 ~ 65535.0
<reserved_lut_size>	X	float	Fill in the last entry of <ldc_r2_lut>	X	O	Camera lens dependent	0.0 ~ 65535.0

Table 4-2 EIS auto bias correction and auto centering tuning parameters

mm-camera: filter_param						
Type	**Data field**	**Meaning**	**Needs per module calibration?**	**Needs tuning?**	**Typical value**	**Range**
float	filter_param[0]	Auto centering threshold, pitch	No	Yes	0.9	0.0~100.0
float	filter_param[1]	Auto centering threshold, yaw	No	Yes	0.9	0.0~100.0
float	filter_param[2]	Auto centering threshold, roll	No	Yes	0.9	0.0~100.0
float	filter_param[3]	IIR clamp threshold	No	Yes	0.001	0.0~100.0
float	filter_param[4]	IIR coefficients	No	Yes	0.95	0.0~1.0

CAMX: filter_param							
Data field	**Sub data field**	**Type**	**Meaning**	**Needs per module calibration?**	**Needs tuning?**	**Typical value**	**Typical range**
<auto_center_th_pitch>	X	float	Auto-centering threshold, for pitch	X	X	100	0.0 ~ 200.0
<auto_center_th_yaw>	X	float	Auto-centering threshold, for yaw	X	X	100	0.0 ~ 200.0
<auto_center_th_roll>	X	float	Auto-centering threshold, for roll	X	X	100	0.0 ~ 200.0
<iir_clamp_threshold>	X	float	IIR clamp threshold, in radian	X	O	10.0	0.0 ~ 100.0
<iir_coefficients>	X	float	IIR coefficients	X	O	0.95	0.0 ~ 100.0
<jerkness_control_en>	X	float	Jerkiness control flag. 0.0 – disable; 1.0 - enable IIR coefficients will be dynamically updated per frame. jerkness_index_pitch, yaw and roll should be tuned. This function is especially useful for hand-held device which include large camera motion. See **Figure 4-1**.	X	Use case dependent	0.0	0.0 ~ 100.0
<jerkness_index_pitch>	X	float	Larger value: more signal will be considered as low-frequency signal while updating IIR coefficients for pitch.	X	Use case dependent	1.0	0.0 ~ 16.0
<jerkness_index_yaw>	X	float	Larger value: more signal will be considered as low-frequency signal for yaw.	X	Use case dependent	1.0	0.0 ~ 16.0
<jerkness_index_roll>	X	float	Larger value: more signal will be considered as low-frequency signal for roll.	X	Use case dependent	1.0	0.0 ~ 16.0

CAMX: filter_param							
<additional_k>	X	float	Will be updated in future release	X	X	1.0	0.0 ~ 100.0
<jerkness_lut_mid_point>	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0
<jerkness_lut_1st_order_start>	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0
<jerkness_lut_2nd_order_start >	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0
<jerkness_lut_slope>	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0
<jerkness_lut_1st_order_end>	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0
<jerkness_lut_2nd_order_end>	X	float	Will be updated in future release	X	X	0.0	0.0 ~ 100.0

When there is large motion on the camera, the EIS may not be able to fully compensate the motion, which will induce video jerkiness. To avoid this phenomenon, turn on jerkiness control and tune for smoother EIS compensation. Jerkiness control will dynamically update IIR coefficient according to the gyro data, which means <iir_coefficient> will be overwritten when jerkiness control is enabled. However, enabling jerkiness control will leave residual motion on the video.

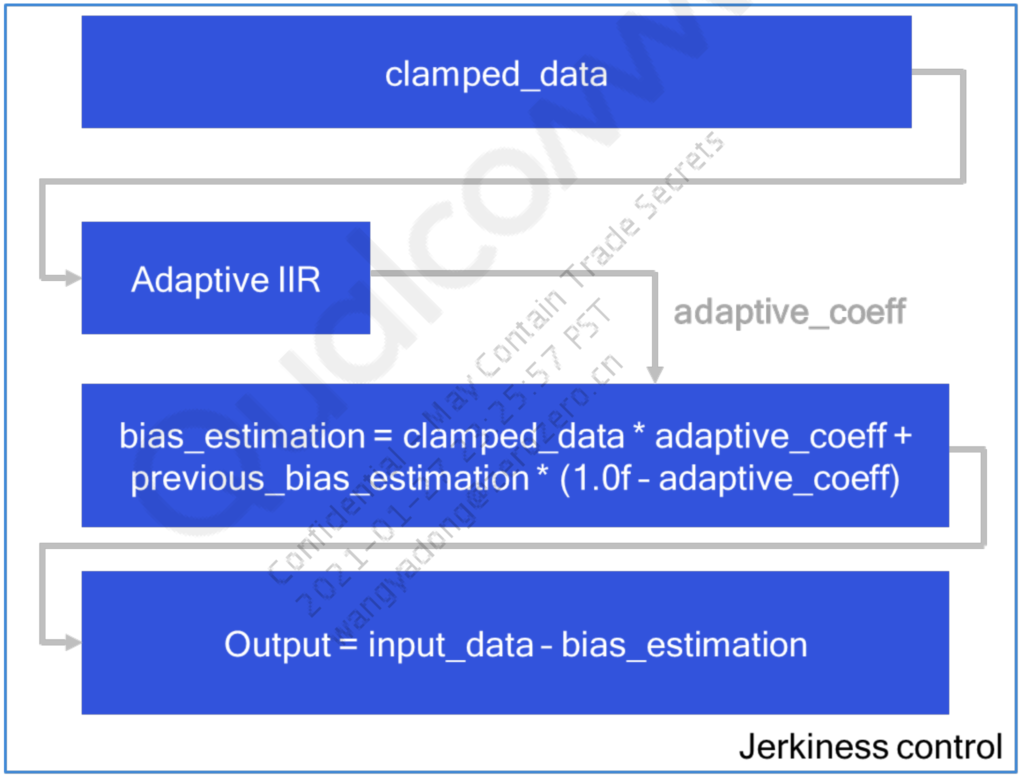

Figure 4-1 EIS jerkiness control: adaptive IIR.

Table 4-3 EIS lowlight strength tuning parameters

mm-camera

Type	Data field	Sub data field	Meaning	Needs per module calibration?	Needs tuning?	Typical value	Range
float	temporal_filter		Exposure time is applied by this IIR filter to avoid sudden change of eis strength	No	Yes	0.9	0.1~0.99
float	number_of_entries		Number of LUT entries	No	No	6	0~16
mvDGTC_eis_strength_one_entry_t	eis_strength_one_entry	trigger_start	Exposure time of trigging the entry of the LUT	No	Yes	0.03	0.0~0.33
		trigger_end		No	Yes	0.10	0.0~0.33
		strength	EIS strength	No	Yes	1.0	0.0~1.0
		reserved	Reserved				

eis_strength CamX							
Data field	**Sub data field**	**Type**	**Meaning**	**Needs per module calibration?**	**Needs tuning?**	**Typical value**	**Typical range**
<temporal_filter>	X	float	Exposure time is applied by this IIR filter to avoid sudden change of EIS strength	X	O	0.9	0.1 ~ 0.99
<number_of_entries>	X	uint	Number of LUT for tuning EIS strength	X	O	6	0.0 ~ 200.0
<eis_strength_entries>	<trigger_start>	float	Exposure time of trigging the entry of the LUT Start point	X	O	0.03	0.0 ~ 0.33
	<trigger_end>	float	Exposure time of trigging the entry of the LUT End point	X	O	0.1	0.0 ~ 0.33
	<strength>	float	EIS strength	X	O	1.0	0.0 ~ 1.0

When the exposure time gets longer for lowlight, the motion blur gets more obvious. To avoid motion blur, there is a mechanism to reduce the EIS stabilization strength. The reduction of EIS strength is decided by exposure time. By the exposure time as index, it looks up the tuning table to get the EIS strength.

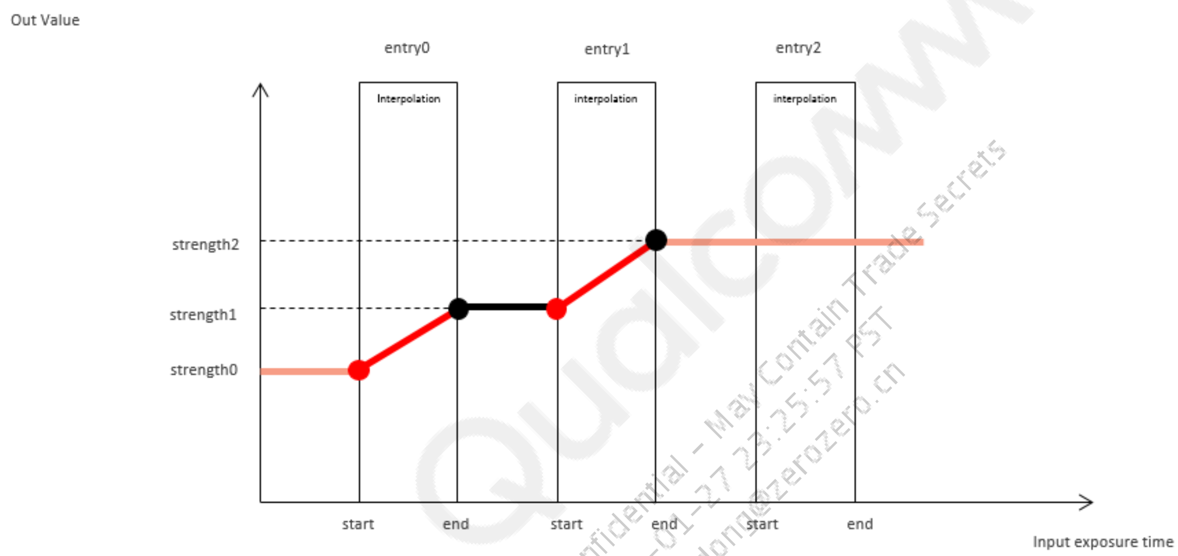

Figure 4-2 Trigger interpolation between adjacent entries of LUT

4.1 adb commands for live tuning

Adb commands	Description	XML
adb shell setprop persist.camera.eis_dg_log	Set to one to enable EIS logs	X
adb shell setprop persist.eis4.online_tuning_en	Set to one to enable EIS online tuning. This flag will only be applied during start session of video streaming. If this flag is disabled, none of the below flags will work.	X
adb shell setprop persist.eis4.focal_length	Focal length in mm	<focal_length_in_mm>
adb shell setprop persist.eis4.delay	Constant delay between camera time stamp and gyro sensor time stamp, in us	<constant_delay_off_pts_in_us>
adb shell setprop persist.eis4.scale	Gyro data extra scale	X
adb shell setprop persist.eis4.dbg_mode	Debug mode flag 1 - disable auto centering; disable auto bias correction; disable margin breach control. Repeated pixel could be seen at image boundary	X
adb shell setprop persist.eis4.lp_dbg	Low-pass filtering debug flag 0 - LP enabled 1 - LP disabled Auto centering and auto bias correction debug flag	X
adb shell setprop persist.eis4.rsc_dbg	Rolling-shutter correction debug flag 0 - RSC enabled 1 - RSC disabled	X
adb shell setprop persist.eis4.ldc_dbg	Lens distortion model debug flag 0 - LDM enabled 1 - LDM disabled EIS takes camera as ideal pinhole camera	X
adb shell setprop persist.eis4.margin	margin breach control debug flag 0 - margin control enabled 1 - margin control disabled	X

4.2 Tuning procedure

There are four main steps in the EIS tuning procedure:

1. Preparation
2. FBIS tuning
3. LDM tuning
4. RSC tuning

The following sections describe each of the EIS tuning steps.

4.2.1 Step 1: preparation

To prepare for EIS tuning, do the following:

1. Wait until the camera sensor driver is done and verified by PLD gating. EIS is sensitive to sensor driver change, especially those that control window setting and exposure setting.
2. Get the gyroscope sensor driver ready. Verify drifting calibration with the gyroscope sensor driver engineer.
3. Prepare LDC and LDM 1D table using LdcTool.exe. (This tool is distributed separately. Contact Qualcomm Customer Engineering for access to the tool).
4. Complete the following fields with the correct value from the camera sensor spec and the lens spec:
   - pixel_size_in_um
   - focal_length_in_mm
5. Complete the following fields according to the hardware layout and gyroscope sensor driver:
   - Rbc[9]
   - gyro_unit
6. Fill the data below in .xml with the default values:
   - <constant_delay_off_pts_in_us> set to 0
   - <fbis_margin> set to 1.1
   - <rsc_margin> set to 1.1
7. Set the flags below:
   - adb shell setprop persist.eis4.online_tuning_en 1.0
   - adb shell setprop persist.eis4.dbg_mode 1.0
   - adb shell setprop persist.eis4.lp_dbg 1.0
   - adb shell setprop persist.eis4.rsc_dbg 1.0
   - adb shell setprop persist.eis4.ldc_dbg 1.0
   - adb shell setprop persist.eis4.margin 1.0

4.2.2 Step 2: FBIS tuning

This step assumes that the lens distortion model (LDM) is not verified. Thus, we mainly use roll rotation whose correction is not impacted by lens distortion.

1. Enable EIS Logging:

   ```
   adb shell setprop persist.camera.eis_dg_log 1
   ```

2. Verify the mappings of <rotation>. Keyword of logs: FBIS: filter(in): pitch: %3.2f, yaw: %3.2f, roll: %3.2f
   a. Set shake table to yaw only.

   b. Read logs and check if pitch and roll are close to zero while yaw has meaningful values.

   c. Redo above step for pitch only and roll only

3. Verify the sign of <rotation>

   a. Disable EIS by adb command:

   ```
   adb shell setprop persist.eis4.scale 0.0
   ```

   b. Set shake table to Yaw only.

   c. Enable live preview to PC.

   d. Enable EIS by below command while closely watching video:

   ```
   adb shell setprop persist.eis4.scale 1.0
   ```

   e. Change sign by below command while closely watching video:

   ```
   adb shell setprop persist.eis4.scale -1.0
   ```

   If negative scale gives more stabilized video, it means the sign of Rbc[9] is incorrect.

   f. Redo above steps for pitch only and roll only.

4. Tune the focal length.

   a. Set the shake table to yaw only.

   b. Enable live preview to PC.

   c. Tune the focal length using the following command and change the value until you see the most stabilized video at the center of the image. Since LDM is not tuned, do not judge the motion compensation result other than image center.

   ```
   adb shell setprop persist.eis4.focal_length 1.20
   ```

   The number takes effect immediately without stopping and restarting streaming.

4.2.3 Step 3: LDM tuning

Fully undistorted LDC

LDM can be generated from two different ways: offline tuning from lens profile or offline tuning from image calibration. The following figures show the procedure to do LDM tuning by the Ldc_Tool. Contact customer engineering to get the external release of the Ldc_Tool and detailed instructions.

- From the lens model: it is ideal that the lens profile is provided by the lens vendor from the optical model.

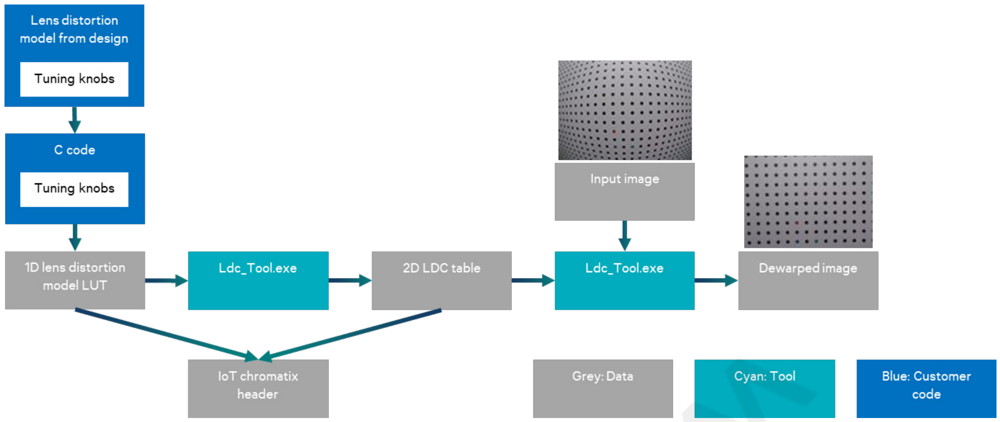

Figure 4-3 LDM offline tuning (from lens model)

- From image calibration:

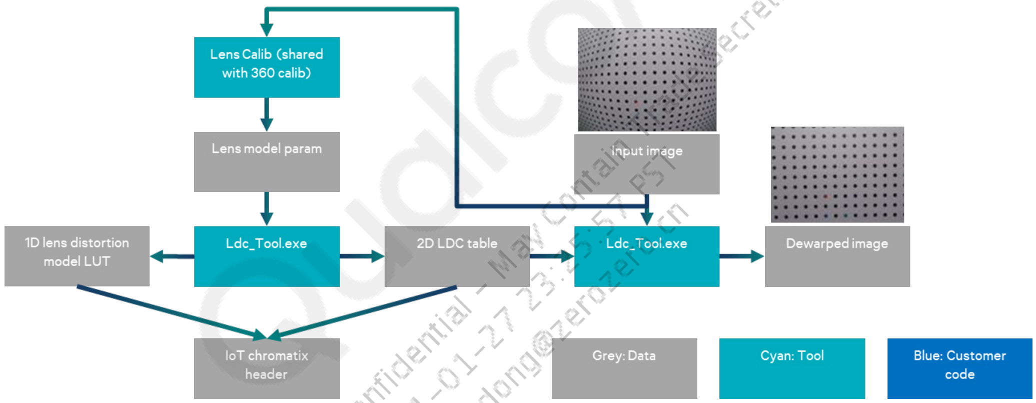

Figure 4-4 LDM offline tuning (from image calibration)

During LDM tuning, one of the kernel ideas is to visually check the position of the image center before LDC and after LDC. We should make sure the image center should have no upscaling and no downscaling, and the output image should be straight lines instead of curved lines.

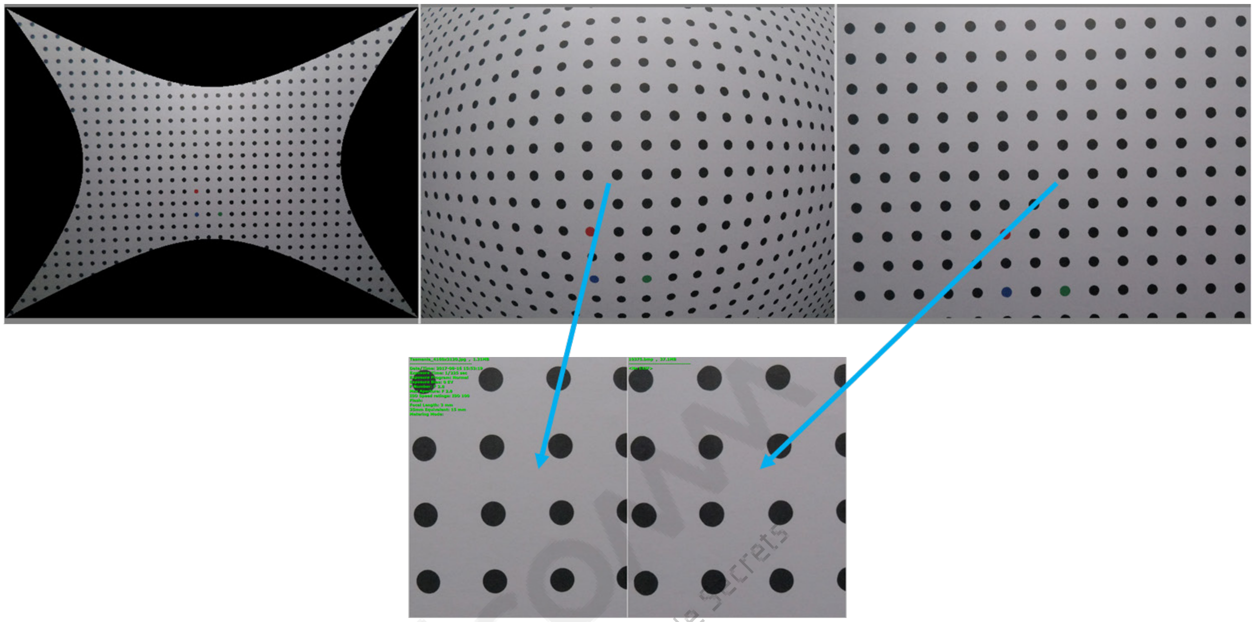

Figure 4-5 LDM offline tuning from image calibration. (Left) Reference LDC image of full FOV. (Middle) Input image for LDM. (Right) Output image from Ldc_Tool.

Partial LDC

The camera lens generally suffers from image distortion. LDC and EIS used to be supported concurrently. However, fully correct lens distortion will greatly reduce FOV. By enabling partial LDC, lens distortion will not be fully corrected. We can keep more FOV while undistorting some area of the image.

To enable partial LDC update ldc_en in headers or .xml according to the following use case:

LDC enabled: 1

Partial LDC enabled: 2

```
// DGTC
//==============================
{
    /*filter_type*/
    0,
    /*ldc_en*/
    0,
    /*rotation_relative_coord*/
    0,
    /*horizon_level_correction*/
    0,
    /*filter_param*/
    {
      0.9f,
      0.9f,
      0.9f,
      0.001f,
      0.95f,
```

4.2.4 Step 4: RSC tuning

In this step, tune <constant_delay_off_pts_in_us> to compensate the timestamp error between the gyro data and the start-of-frame from the sensor driver.

1. Enable RSC and LDM by adb command.

```
adb shell setprop persist.eis4.rsc_dbg 0.0
adb shell setprop persist.eis4.ldc_dbg 0.0
```

2. Tune timestamp mismatch.
   a. Set shake table to yaw only. Higher frequency and larger degree of rotation on the shake table will make it easier to observe the rolling shutter phenomenon.
   b. Tune timestamp error compensation with the command below. Change the value until you see the most stabilized video (vertical lines in the captured image should not be slanted).

```
adb shell setprop persist.eis4.delay 800
```

   Typically, a reasonable value should be in the range of -2000~2000. If the tuning result shows a larger value, the driver for the image sensor and gyro sensor should be further verified. After finishing tuning, fill in the field <constant_delay_off_pts_in_us> in the .xml file.

4.3 Logging

Set the below setprop to get the EIS related logs.

adb shell setprop persist.camera.eis_dg_log 1

adb shell setprop persist.camera.stats.is.debug 5

Look for the logs below to confirm that EIS is enabled and all parameters are set properly.

```
mvDG    : int mvDG_StartSession(mvDG *, mvDGConfiguration *)
eis_config_wrapper dump: input size: (1984x1116),output size(1920x1080),
ldc_mesh 0xeecc7b88, final_mesh[0] 0xe32a4a00, final_mesh[1] 0x0, ldc_inv:
0xeecd50b4, 3240.000000 , prior_ldc_present is 0 self: 0x0

mvDG    : int mvDG_StartSession(mvDG *, mvDGConfiguration *)
eis_config_wrapper dump: calib: pixel_size 1.400000, focal_length 2.500000,
unit 65536

mvDG    : int mvDG_StartSession(mvDG *, mvDGConfiguration *)
eis_config_wrapper dump: calib: Rbc: 1.000000, 0.000000, 0.000000,
0.000000, -1.000000, 0.000000, 0.000000, 0.000000, -1.000000

mvDG    : int mvDG_StartSession(mvDG *, mvDGConfiguration *)
eis_config_wrapper dump: tuning: fbis_margin 1.100000, rsc_margin 1.100000

mvDG    : int mvDG_StartSession(mvDG *, mvDGConfiguration *)
eis_config_wrapper dump: misc: sample rate 1000

mvDG    : eis_tc_init: total margin is 1.210000

mvDG    : eis_tc_init: success: initialized prime RT 0.826446, 0.000000,
172.165329, 0.000000, 0.826446, 96.843002, 0.000000, 0.000000, 1.000000
```

5 Limitations

5.1 Operational modes

- EIS only applies to encoded streams. It does not apply to preview.
- Intra-frame (RSC only) EIS for snapshot mode is not implemented or supported.

5.2 Performance

- PPEISCORE/ iWarp GPU dewarp engine: 4K30 fps
- Maximum supported gyroscope sampling rate: 3000 Hz.
- Maximum verified gyroscope sampling rate: 1600 Hz.

5.3 Concurrency requirements and feature interaction

Feature	Interaction Description
TNR	Concurrency supported
YUV CAC	Concurrency supported
EIS 3.0	Conflict
LDC	Concurrency supported
CDS	Conflict.
Video HDR	Not supported (because of time stamp problem)
Binning correction	Not supported (because of time stamp problem)
Digital zoom	Concurrency supported
Dual camera	Not supported
360 camera	Not supported yet
Face detection	Concurrency supported
Auto focus	Limited support (will not work with higher frequencies)

6 Testing

6.1 Functional tests

Refer to the logging section above (look for MVDG logs).

6.2 Stability tests

The following stability tests need to be thoroughly executed for this feature. Check for frame drops, EIS compensation, and drifting.

- 4K30 video recording w/ 2nd stream recording
- 1920x1440@60 video recording w/ 2nd stream recording

6.3 Testing equipment

The following test equipment is required:

- Shake table
  - Minimum required: 10hz 0.5 degree
  - Recommended: 20hz 1degree
- Test chart
  - Minimum: Checker board, 700lux, 3 feet away from shake table
  - Recommended: Checker board, 1500 lux, 10 feet away from shake table

A References

A.1 Acronyms and terms

Acronym or term	Definition
EIS	Electronic image stabilization
FBIS	Frame based image stabilization
RSC	Rolling shutter compensation
LDC	Lens distortion correction
UDE	User defined effect
LDM	Lens distortion model
DIS	Digital image stabilization
OIS	Optical image stabilization