Qualcomm Technologies, Inc.

Qualcomm Spectra™ 2xx Camera ISP

Tuning Guide

80-NK872-13 Rev. E

February 6, 2018

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

Contents

Tables

Revision history

Revision	Date	Description
A	July 2017	Initial release
B	October 2017	Added new modules to match Chromatix v7.0.7 release
C	November 2017	Updated modules based on new software functionality
D	December 2017	Updated procedure for using the Simulation tab. Added procedure for running AWB simulation in a command line.
E	February 2018	Updated modules based on new software functionality. Added procedure for using the JPG Simulator tab.

Part I Prerequisites

This section defines the prerequisite tasks that are required prior to camera ISP tuning and provides step-by-step procedures for performing each task.

RELATED INFORMATION

"Create a new tuning project" on page 11
"Add scenarios" on page 16
"Add and edit region triggers" on page 15
"Capture images to use for tuning" on page 18

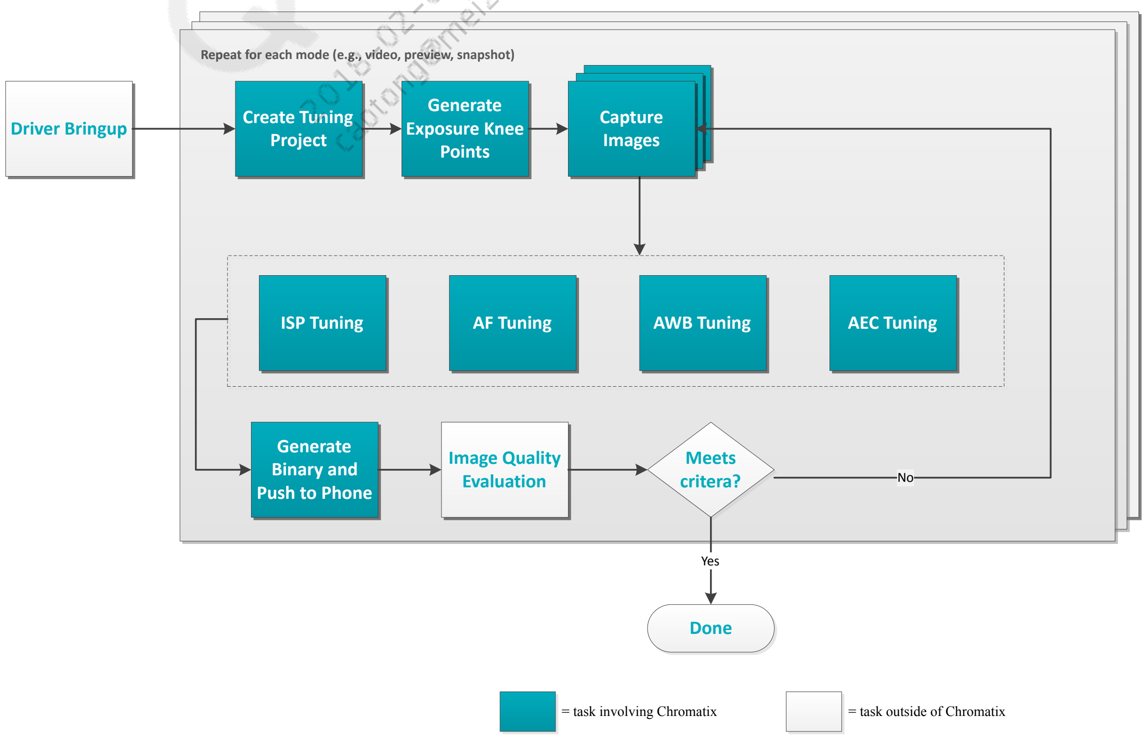
1 Introduction

1.1 Purpose

An important part of integrating the chipset into a device is to tune the camera parameters to ensure that the device captures high-quality images. This document describes the procedures for performing image signal processor (ISP) tuning for camera software.

1.2 ISP tuning process overview

The flow chart below illustrates the overall camera tuning process. The colored boxes represent tasks involving the Chromatix™ and show how they fit in the overall tuning process. Prerequisite tasks for ISP tuning begin after driver bringup is complete and ISP tuning itself occurs mostly in parallel with AF, AWB, and AEC tuning. Changes in these other areas may have an impact on ISP tuning, so maintain best practices in team communication and collaboration to minimize avoidable delays.



Although the process is illustrated as being relatively linear, camera tuning is an iterative process in which you repeat the entire process or sections of the process several times. After completing the initial tuning and saving your Chromatix project, subsequent iterations for the same project usually begin by opening the existing project and refining the parameters with additional basic or advanced tuning.

Task	Description
Prerequisites	Perform tasks that are prerequisite to ISP tuning: create a new project, load the settings on the device, and capture images with the device.
Initial tuning	Perform ISP tuning in multiple iterations. Evaluate the results of tuning at any time using the simulation feature.
Simulate the tuning	Use the simulation feature at any time during the tuning process to see how a specific set of parameters for a specific series of tuning modules affect the raw image. Use the simulator to check the image results as it passes through each tuning module to identify where a specific issue is introduced.
Image quality evaluation	After each tuning session, capture new test images with the tuned device and objectively (numerically) measure image quality.
Load tuned settings	At this stage of the process, generate binary files that contain tuned parameters and load the settings on the device.

1.3 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, #include.

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

Commands to be entered appear in a different font, e.g., copy a:*.* b:.

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

1.4 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.

Suggestions for improving documentation are greatly appreciated. If you have suggestions regarding the scope, accuracy, or content of this document, send feedback to DocFeedback@qti.qualcomm.com.

2 Create a new tuning project

Prerequisites:

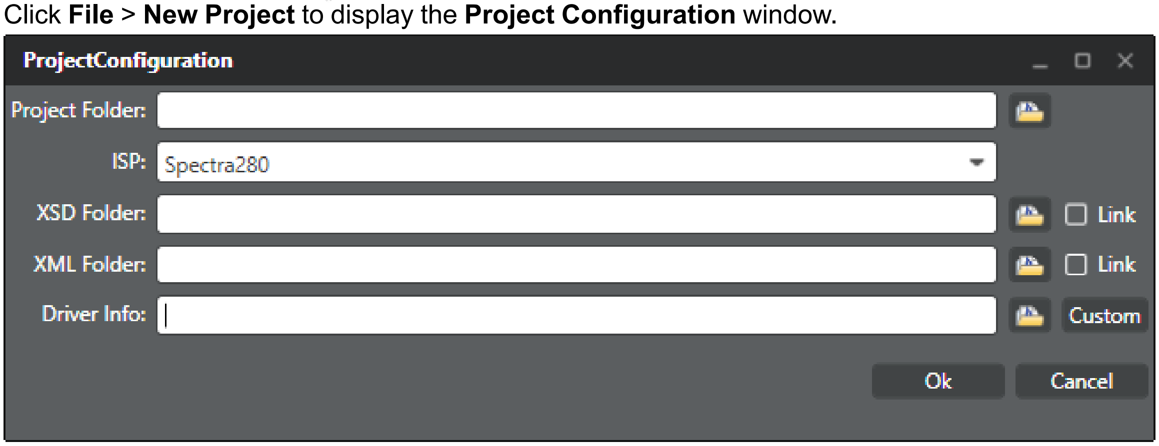
The following are required to create a tuning project:

- XSD folder – A local copy of the default XSD files. The default files are located here: *$BUILD_ROOT\LINUX\android\vendor\qcom\proprietary\chi-cdk\cdk\chromatix\XSD*
- XML folder – A local copy of the default XML files. The default files are located here: *$BUILD_ROOT\LINUX\android\vendor\qcom\proprietary\chi-cdk\cdk\chromatix\XML*
- Driver info – The sensor driver information file is provided by the sensor manufacturer in a text file. If this file is not available, specify the sensor information when you create the project.

Note: It is recommended that you use a similiar directory structure as the build when creating your local XSD and XML folders (e.g., *<someplace>\cdk\chromatix\XSD and <someplace>\chi-cdk \chromatix\XML*).

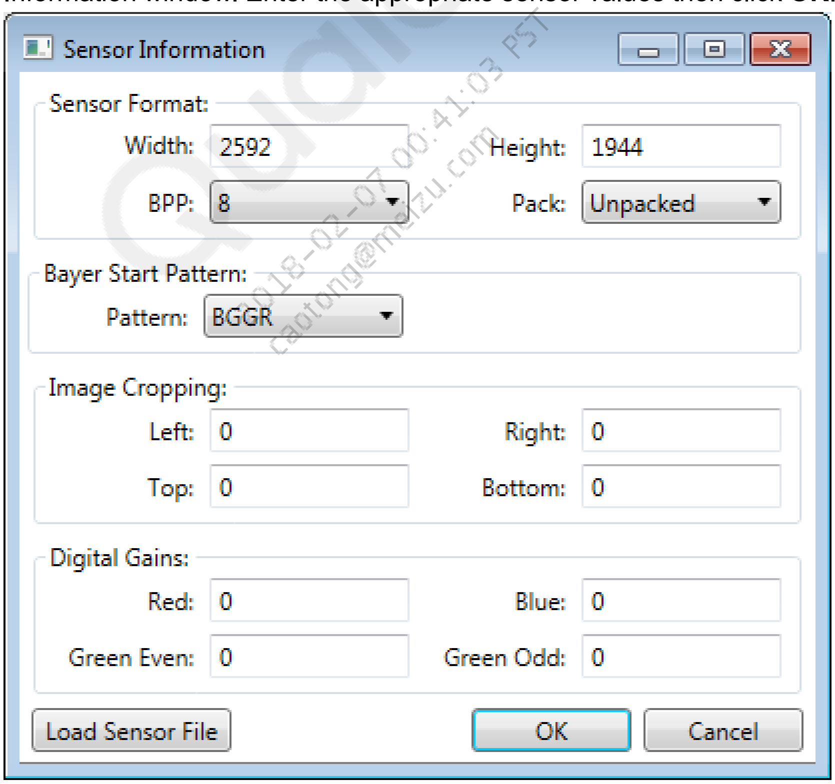
A Chromatix tuning project captures the parameter values set during the tuning process and the camera sensor module information. A tuning project is specific to one ISP version. A single project is used to save the parameters for all use cases (i.e., preview, video, snapshot, etc.) for a single sensor. If your device has multiple sensors (e.g., front camera, rear camera, etc.), then you need a separate project for each sensor and each project must use the same XSD set.

1. Click **File** > **New Project** to display the **Project Configuration** window.



2. Click  next to the **Project Folder** field and navigate to or create a folder for Chromatix to store the project files. This file path and folder name should not contain spaces. Make the file path as short as possible to avoid the Windows max path size limitation.
3. Select the appropriate ISP version (e.g., Spectra280) from the ISP list.

4. Click next to the **XSD Folder** field and navigate to the local folder containing the XSD files (ISP, 3A, and common) copied from the default location.
5. Select **Link** next to the **XSD Folder** field. By linking the project to the folder, the tuning project is automatically updated if you update the XSD. Note that the relative path between the root folder and the XSD folder is critical. If this is changed, then the link is broken.
6. Click next to the **XML Folder** field and navigate to the local folder containing the default XML files copied from the default location.
7. (Optional) Select **Link** next to the **XML Folder** field. Unlike the XSD, if you update the XML in the XML folder, then you need to update the XML in the tuning project manually.
8. Do one of the following:
   - If you have a text file with the sensor information (e.g., max line count, bayer pattern, bits per pixel, etc.), then click next to the **Driver Info** field, navigate to the applicable sensor driver text file, and click **Open**.
   - If you do not have a text file with the sensor information, click **Custom** to display the Sensor Information window. Enter the appropriate sensor values then click **OK**.



9. Click **OK** to create the project.

   If the project root folder is not empty, a prompt appears. If you click **OK** at the prompt, Chromatix overwrites any current files within the root folder.
10. To save the project, select **File** > **Save to Project**.

RELATED INFORMATION

"Create a sensor driver information file" on page 13

2.1 Create a sensor driver information file

If the sensor driver information file is not available, create a text file based on the sensor setting information provided by the sensor vendor or enter the sensor information directly into the **Sensor Information** dialog box. A separate file is needed for each mode in the project.

It is important that this information match the intended raw input files, i.e., image dimension, Bayer pattern, etc., or Chromatix will not able to correctly read the input files.

If you choose to manually create a text file, the required format of the sensor driver information text file is as follows:

```
30.000000f,     /* Maximum fps */
1,              /* Minimum line count */
3186,           /* Maximum line count in max fps */
6372,           /* Maximum line count */
1,              /* Minimum gain */
4,              /* Maximum gain fix fps */
50,              /* Maximum gain after linear AFR */
4208,           /* Raw image width */
3120,           /* Raw image height */
BGGR,           /* Bayer pattern */
10,              /* Bits per pixel
1,              /* Packed or not, packed=1, unpacked=0
0,              /* Crop top after CAMIF */
0,              /* Crop bottom after CAMIF */
0,              /* Crop left after CAMIF */
0,              /* Crop right after CAMIF */
256,            /* Black level in 12-bit value based on spec
256,            /* Black level under low light
256,            /* Black level under bright light
```

Table 2-1 Sensor driver information fields

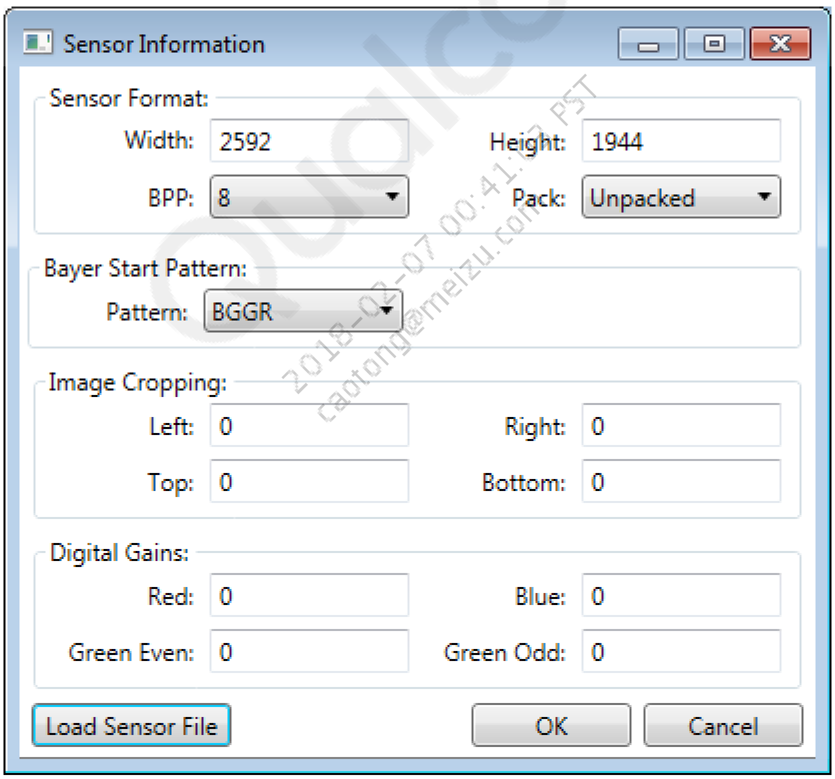
Fields	Sample value
Maximum FPS	30.0
Minimum line count	1
Maximum line count in max fps	3186
Maximum line count	6372
Minimum gain	1.0
Maximum gain for fixed fps	4.0
Maximum gain after linear AFR (Note: Analog sensor gain – not including digital gain)	50.0
Raw image width	4208
Raw image height	3120
Bayer pattern	BGGR
Bits per pixel	10

Table 2-1 Sensor driver information fields (cont.)

Fields	Sample value
Packed or not	1
Crop top after CAMIF	0
Crop bottom after CAMIF	0
Crop left after CAMIF	0
Crop right after CAMIF	0
Black level in 12-bit based on specification	256
Black level under low light	256
Black level under bright light	256

The other option is to enter sensor information directly into Chromatix when creating a project.

1. From the new project window, click **Custom** to open the **Sensor Information** dialog box.



2. Enter the **Image Width**, **Image Height**, and **Bits per Pixel** of the raw image.
3. Choose whether the raw image is in **Unpacked**, **Packed**, or **MIPI** format.
4. Select the appropriate raw image **Bayer start pattern**.
5. Set the **Image Cropping** fields, if needed.
6. Enter the **Digital Gains**, if needed.
7. Click **OK**.

3 Add and edit region triggers

A region is a set of triggers that are indicative of light conditions, lens positions or other conditions as applicable to a given module. A project which is created with the build XSD and XML files has only one region for each module. Chromatix 7 gives the flexibility to create multiple regions for an ISP module. Adding regions to the module enables to optimize camera performance based on the established triggers for each region. Add regions to the pipeline before tuning the ISP and before adding modes to the project.

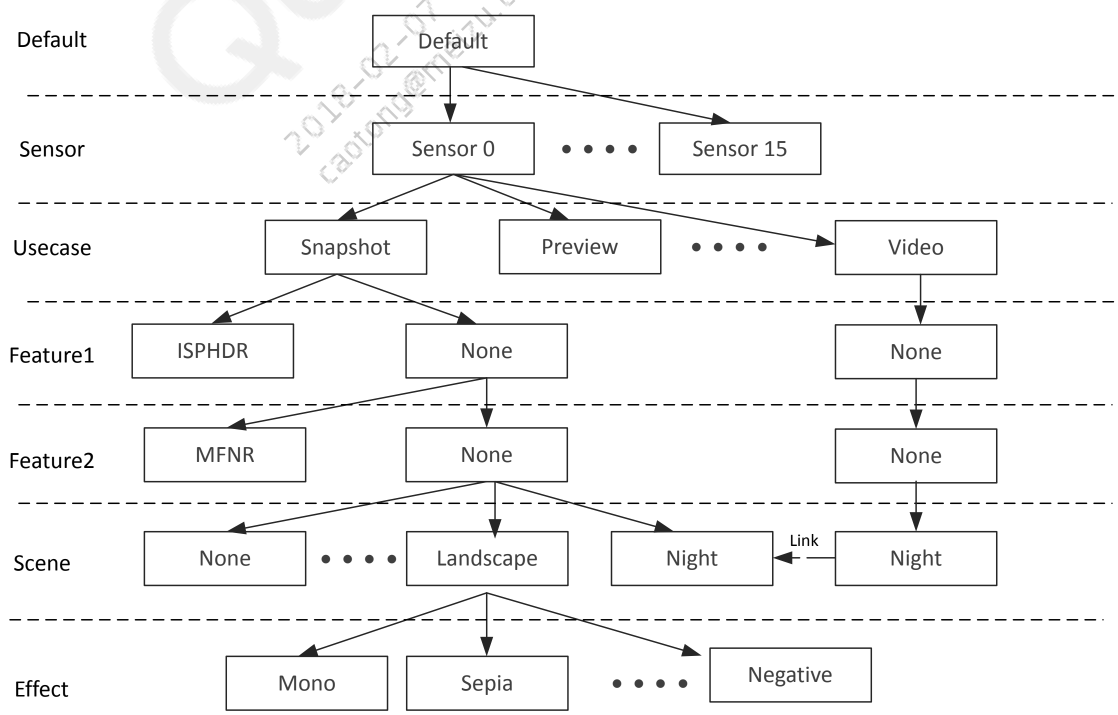
To add regions:

1. Open the **Parameter Editor**. In the left frame, expand the applicable pipelines (BPS, IFE, or IPE).
2. Expand the applicable module (for example, Linearization 34 BPS).
3. Click **Chromatix [module name] Core** to show the list of editable triggers. The number of editable triggers vary between modules. For example, linearization has five triggers but ABF has only three.
4. Hover the cursor over a trigger value and use the up and down arrows onscreen to add triggers above or below the current trigger.
5. Repeat the previous step until all the regions are added to the module. Regions are added from left to right. If a region is added from the left-most column, an entire row of triggers is added. If a region is added from any other column, the region is nested to the triggers to the left from where the region has been added.
6. To view the range of values in a trigger, click the trigger in the main frame.
7. Edit the values in the bottom frame as needed to specify the appropriate range for a trigger.
8. Repeat the previous step until the range values have been specified for all the triggers of all the regions.
9. Delete triggers by hovering the cursor over a trigger value and clicking the X onscreen.
10. Save the project before closing Chromatix 7.

4 Add scenarios

The Chromatix 7 data structure begins with a default scenario created in the root folder of your project. Add child scenarios to your project so that you can tune for various use cases, features, scenes, and effects that diverge from the parent of the child. A directory structure for the scenario is added to its parent in the project folder whenever you add a scenario. When you tune the parameters for a child scenario, you only need to tune the deltas. XML representing the deltas is created and stored in the child scenario directory structure. When you generate a binary, the child inherits the settings of its parent except for the parameters that are tuned explicitly for the child. Refer to *Chromatix 7 Mode Tree: Inheritance, Use Cases, Scenes, and Other Scenarios (Part 1)* and *Chromatix 7 Mode Tree: Inheritance, Use Cases, Scenes, and Other Scenarios (Part 2)*for more information.

The following image is for example purposes and is meant to highlight the hierarchy of modes (e.g., default, sensor, usecase, etc.). It does not include all available options for each mode. Notice that it is possible to link modes at the same level. In doing so, the source mode (Video|None|None|Night in the example) is configured with the XML of its target.



Note: If you want a child scenario to have the same regions and triggers as its parent, add regions to the default scenario before adding child scenarios.

Prerequisites: A loaded project is required before you can add or edit a scenario.

To add a scenario to a project, do the following:

1. Click **Edit** > **Edit Scenario** to display the Tuning Scenario Configuration window.
2. Select the parent scenario in the **Scenario Hierarchy View** panel on the left-side of the window. The scenario you add becomes a child of the selected scenario.
3. Do one of the following:
   - Select the appropriate sub mode from the **Sub Mode** list.
   - If applicable, select **Add all sub modes** from the **Sub Mode** list.
4. To link the scenario to a scenario that already in the hierarchy, select the target scenario from the **Link** to list. This field is disabled if you selected Add all sub modes in the previous step, but links can be established after. See step ?
5. If you are adding scenarios to Snapshot, select **BPS+IPE** in the **Pipeline** list. Otherwise, accept the default.
6. (Optional) If the driver information differs from the parent scenario, load the additional driver information files or click **Custom** to create new information.
7. Click **Add**. The new scenario is added to the **Scenario Hierarchy View** and a directory structure for the scenario is added to parent in the project root folder. If you selected Add all sub modes, a prompt appears asking you to confirm that you want to add all sub modes. Click **OK** to confirm.
8. Repeat Steps 2-7 as necessary.
9. The active scenario is the last scenario added. To change the current scenario, click another scenario in the Scenario Hierarchy view and click Set Current.
10. Click Close.

5 Capture images to use for tuning

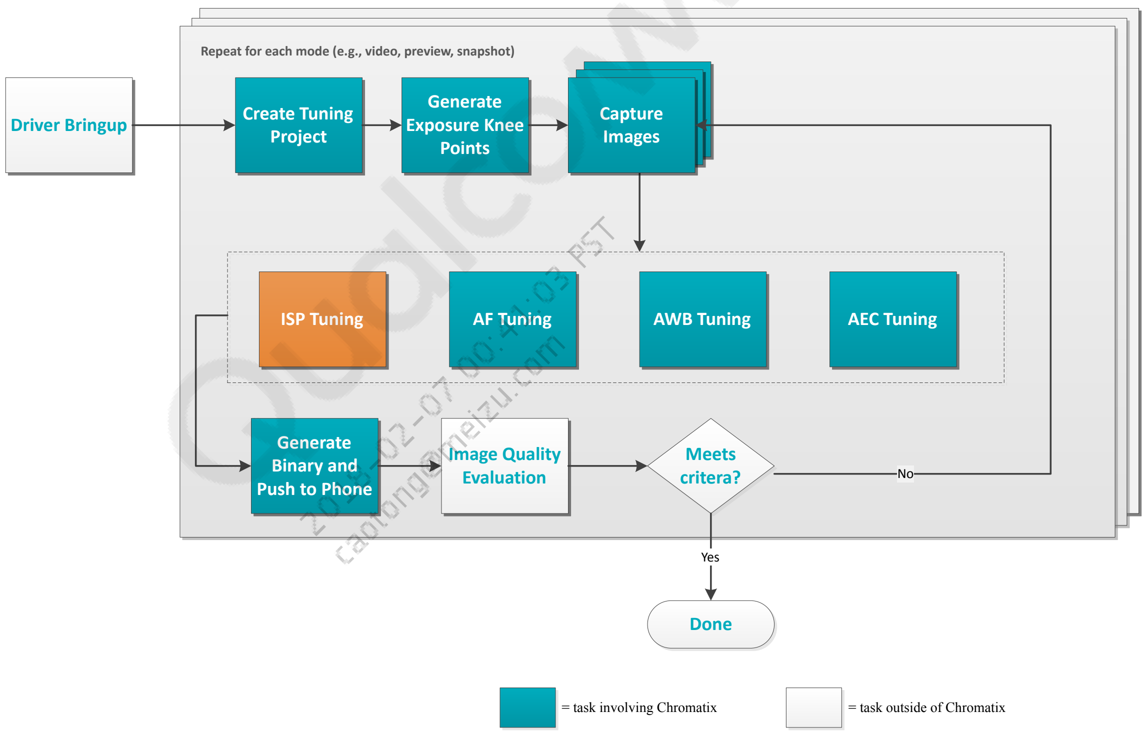
See the *Capturing Images for Camera Tuning* appendix for guidance on how to capture images to use for tuning. A spreadsheet attached to the tuning guide PDF lists the characteristics of the images required for initial tuning. The **Info** column lists special characteristics for select images.

RELATED INFORMATION

"Capturing Images for Camera Tuning" on page 173

Part II Spectra 2xx ISP Tuning

This section provides step-by-step procedures for tuning the ISP modules.

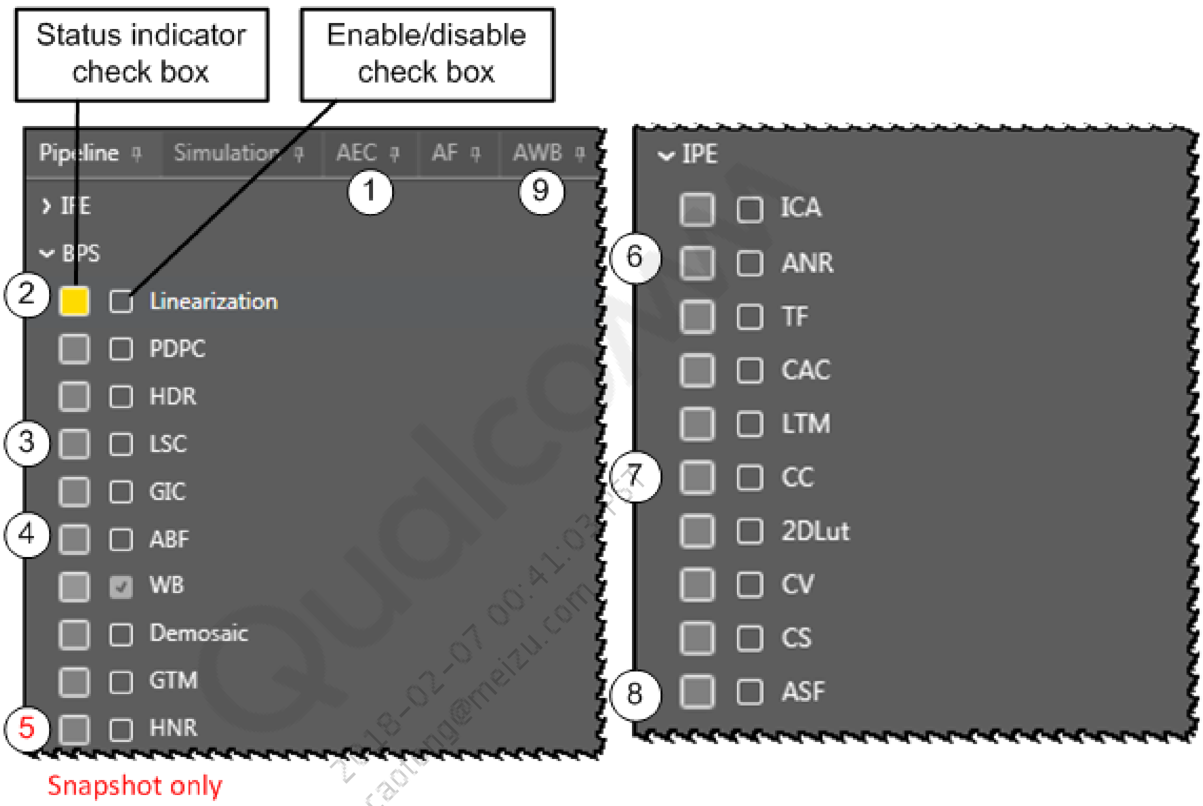


NOTE Motion compensated temporal filter (MCTF), multi-frame noise reduction (MFNR), and multi-frame super resolution (MFSR) are not supported for Windows on Snapdragon (WoS).

For initial tuning, use the numbers in the pipeline figure below as a guide. The numbers indicate the tasks required for manual initial tuning for the ISP 2xx and the order in which the tasks must be performed. Refer to procedures named for each ISP module for detailed instructions.

1. Tune AEC trigger points for ISP modules
2. Tune linearization
3. Tune LSC
4. Tune ABF
5. Tune HNR (snapshot mode only)

6. Tune ANR
7. Tune Color Correction/Gamma
8. Tune ASF
9. Tune AWB reference points



The non-numbered modules are optional and can be tuned after initial tuning. In the pipeline drawing, the tuning status of each module is indicated by the box on the right (gray = untuned, yellow = partially tuned, green = fully tuned).

Use the simulation feature at any time during the tuning process to evaluate the progress of the tuning. After each tuning session, capture new test images with the tuned device and objectively measure image quality.

RELATED INFORMATION

"Simulation Tool" on page 191

6 BPS tuning

6.1 Linearization tuning concepts

Because of image photosensor characteristics, sensor output response to scene lightness may not necessarily be linear and may be different for various lighting conditions. For example, the sensor photonic response curve would be nearly flattened on bright light saturation. For dark environments, sensor output would not be totally zero due to dark current.

The black level offset value depends on the specific sensor being used and is also a function of the integration time, overall gain settings corresponding to the exposure settings, and temperature. The black level of the sensor response must be corrected for each channel on every pixel for different lighting conditions.

The black level offset uses a 2- or 4-channel black level correction. The black level offset values are then subtracted from the pixel output to get the corrected value.

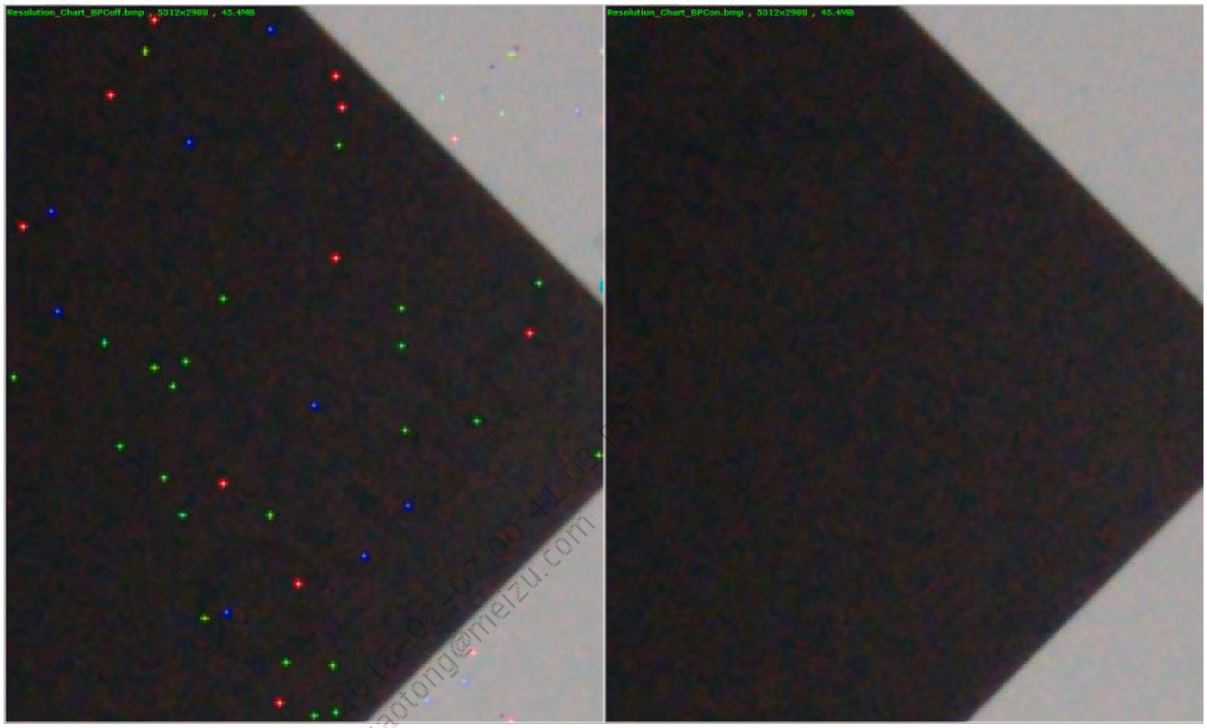
6.1.1 Tune linearization

Linearization tuning adjusts the level of brightness in the darkest part of the image. Channel values from three images are averaged to calculate the black level offset values. These offset values are then subtracted from the pixel output to obtain the corrected value.

1. On the **Pipeline** tab, click **Linearization**, and then click a region in the Regions group.
2. (Optional) Deselect the **Bls Enable** check box you would like to use default black level values for black level subtraction (BLS).
3. Calculate linearization values:
   a. Click a region in the Regions group.
   b. Click **Load Image**, select the applicable black level reference image, and click **Open**.
      The image values are listed in the table fields with the average values listed underneath.
   c. Click **Recalculate** to recalculate linearization values for the R/G and G/ B values.
   d. Click **OK**.
4. Repeat process for each region.
5. Click **File** > **Save to Project** to save any changes made.

6.2 Pixel correction tuning concepts

A defective pixel has a response that is noticeably different from that of the other pixels in the array under dark or uniformly illuminated conditions. Defective pixels may show up on an image as bright points (hot pixels) or as dark points (cold pixels). The pixel correction module includes processing for sensors with or without PD pixels and for both Bayer and zzHDR patterns.



(L) Bad pixel correction off; (R) Bad pixel correction on, defective pixels corrected

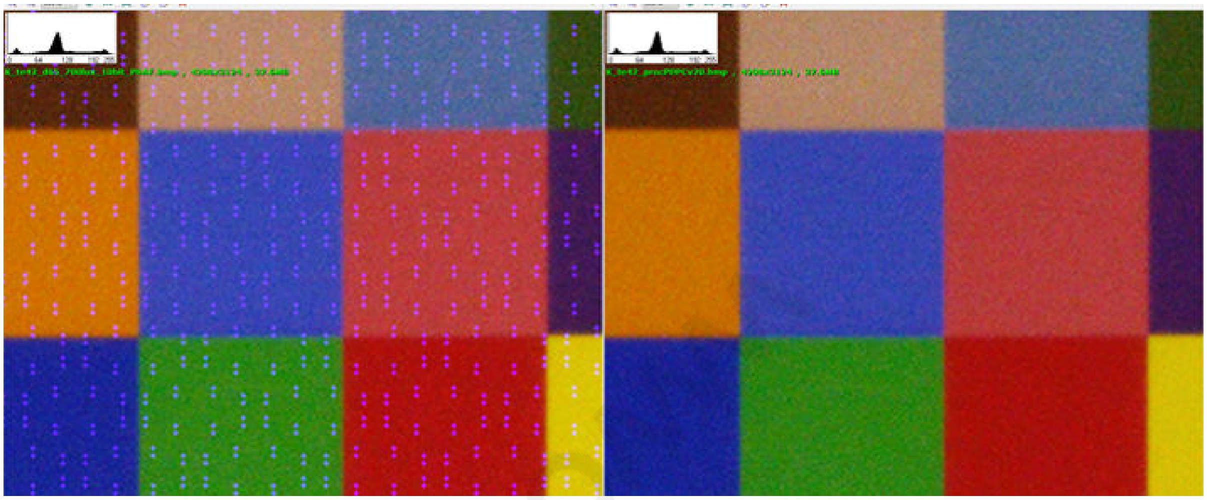
Defective pixels can be caused by the following:

- Imperfections in the semiconductor processing of the sensor module (CMOS or CCD) – During manufacturing, bad pixels or clusters can be caused by leakage to or from the well, abnormal pixel sensitivity, or resistance losses.
- Dust introduced when lens is mounted – A dust particle can fall on top of the sensor when the lens is mounted.
- Age of the sensor module – Over time, a sensor module accumulates more bad pixels.

PD pixel correction concepts

A CMOS sensor can have phase pixels to speed up autofocus. One difference between phase pixels and the normal pixels in a sensor is that the location of the phase pixels is known. The PDPC module takes advantage of this location information to correct defective PD pixels that occur in those locations. Camera software detects phase pixel information in the sensor or driver information file and passes it to the simulator. Currently, the PDPC module only corrects bad PD pixels if they are on a red or blue pixel. Green pixels are not corrected at this time.

Because the PDPC module knows the location of a defective phase pixel, it has all the information it needs to make a correction based on the neighboring pixels. Thus, there is no parameter tuning required for PD pixel correction. The PD pixel correction supports sensors that use the Bayer and zzHDR pattern.

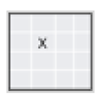
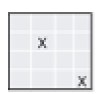
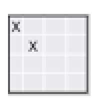
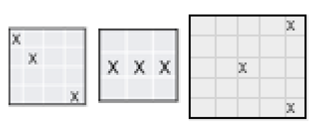


(L) Original image, PDPC off, defective PD pixels on blue locations; (R) PDPC on, defective PD pixels corrected

Non-PD pixel correction concepts

The pixel correction module includes processing to correct bad pixels for a non-PD pixel sensor project that uses either the Bayer or zzHDR pattern. The following table identifies the types of single and couplet pixel defects that can be corrrected in the flat region.

Table 6-1 Flat region defects supported by bad pixel correction

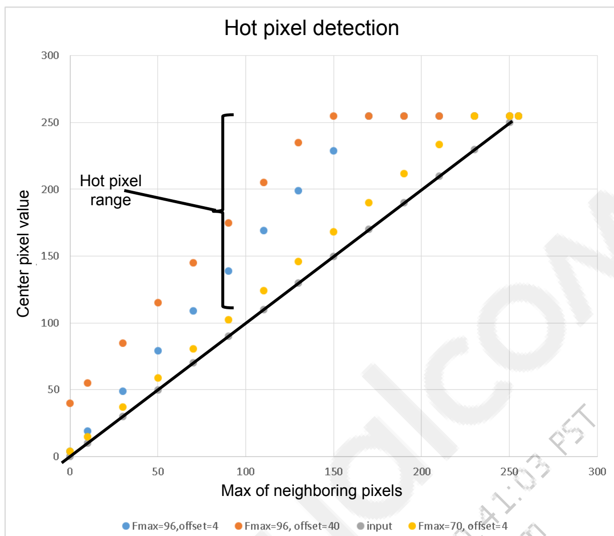
Type of defect	Example	Supported
Single		Yes
Same channel couplet		Yes
Cross channel couplet		Yes
Cluster ≥ 3		No

To correct hot pixels, the module evaluates the value of the bad pixel with the formula: (Fmax * MP) + Offset

- MP is the maximum pixel value from among the neighboring pixels of the same channel
- Fmax is a user-defined threshold factor that specifies which pixels to correct

- Offset is a user-defined adjustment value that further refines the threshold for a bad pixel or pixel couplet
- If the value of the bad pixel is greater than the result, replace the bad pixel with the value of MP. If the bad pixel value is less than or equal to the result, do not change the bad pixel value.

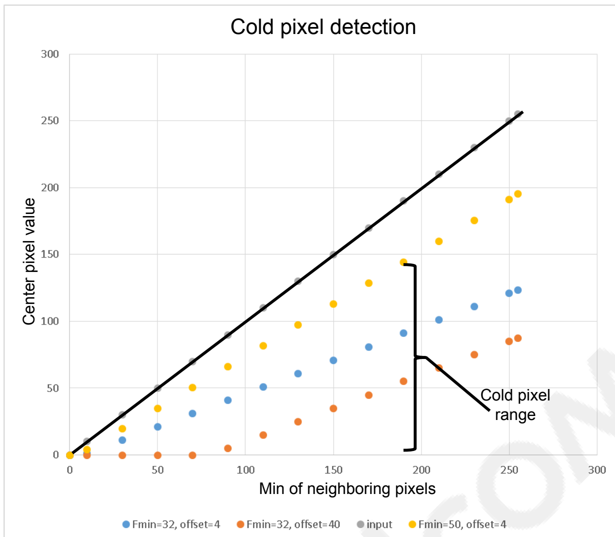
The following figure is a sample plot of the hot pixel equation.



Similarly, to correct cold pixels, the module evaluates the value of the bad pixel with the formula: (Fmax * mP) + Offset

- mP is the minimum pixel value from among the neighboring pixels of the same channel
- Fmin is a user-defined threshold factor that specifies which pixels to correct
- Offset is a user-defined adjustment value that further refines the threshold for a bad pixel
- If the value of the bad pixel is less than the result, replace the bad pixel with the value of mP. If the bad pixel value is greater than or equal to the result, do not change the bad pixel value.

The following figure is a sample plot of the cold pixel equation.



Coordinating pixel correction with other BPS modules

Keep the following considerations in mind when tuning BPS modules:

- If the sensor used for the project is a PD sensor, PD pixel correction must be enabled
- The pixel correction module does not rely on other modules to be tuned, but if the pixel correction module is not properly tuned, all modules that follow it in the pipeline are impacted
- For pixel correction tuning, only enable the basic modules that are needed to result in final YUV images
- Consider disabling bad pixel correction module while tuning other modules to avoid any negative impact of untuned pixel correction parameters.

RELATED INFORMATION

“Pixel correction parameters” on page 26
“Tune pixel correction” on page 30

6.2.1 Pixel correction parameters

The following table lists the parameters used for tuning the pixel correction module for snapshot mode. The default values for the parameters are suitable for most situations but may require adjustment in some use cases to achieve your specific objectives.

Table 6-2 Pixel correction parameters (Snapshot)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Pdpc Enable (Enable Section)	Enables or disables PD pixel correction.	Set to 1 if the project uses a PD sensor and you want to correct bad PD pixels.	0 Enable = 1 Disable = 0	Yes, if applicable
Dspbc Enable (Enable Section)	Enables or disables single bad pixel correction for zzHDR when using a non-PD pixel sensor.	Set to 1 if the project uses a non-PD sensor with a zzHDR pattern and you want to correct bad pixels.	1 Enable = 1 Disable = 0	Yes, if applicable
Fmax Pixel Q 6 (Pdpc 20 Rgn)	Specifies the threshold for a hot pixel value to be selected for correction.	If the project uses a non-PD sensor, set this value to control how many hot pixels are corrected. A higher value results in fewer pixels detected as bad hot pixels. For a high light region, lowering Fmax has more impact on missed pixels than lowering the offset. For a low light region, lowering the offset has more impact on missed pixels than lowering Fmax.	96 Range: [0, 127]	Use default value, or tune as needed
Fmin Pixel Q 6 (Pdpc 20 Rgn)	Specifies the threshold for a cold pixel value to be selected for correction.	If the project uses a non-PD sensor, set this value to control how many cold pixels are corrected. A higher value results in fewer pixels detected as bad cold pixels.	32 Range: [0, 127]	Use default value, or tune as needed

Table 6-2 Pixel correction parameters (Snapshot) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Hot Pixel Correction Disable (Pdpc 20 Rgn)	Enables or disables correction of hot pixels.	Set to 0 to correct hot pixels	0 Enable = 1 Disable = 0	Yes, if applicable
Cold Pixel Correction Disable (Pdpc 20 Rgn)	Enables or disables correction of cold pixels.	Set to 0 to correct cold pixels	0 Enable = 1 Disable = 0	Yes, if applicable
BPC Offset (Pdpc 20 Rgn)	Specifies the offset adjustment to further refine the threshold for a single bad pixel. Used for processing of a Bayer pattern or a zzHDR pattern with long exposure (T1).	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	512 Range: [0, 16383]	Use default value, or tune as needed
Bcc Offset (Pdpc 20 Rgn)	Specifies the offset adjustment to further refine the threshold for a couplet bad pixel. Used for processing of a Bayer pattern or a zzHDR pattern with long exposure (T1).	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	512 Range: [0, 16383]	Use default value, or tune as needed
BPC Offset T2 (Pdpc 20 Rgn)	Specifies the offset adjustment to further refine the threshold for a sigle bad pixel. Used only for processing of an zzHDR pattern with short exposure (T2).	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	512 Range: [0, 16383]	Use default value, or tune as needed
Bcc Offset T2 (Pdpc 20 Rgn)	Specifies the offset adjustment to further refine the threshold for a couplet bad pixel. Used only for processing of an zzHDR pattern with short exposure (T2).	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	512 Range: [0, 16383]	Use default value, or tune as needed
Correction Threshold (Pdpc 20 Rgn)	Specifies the minimum value change allowed to the bad pixel. If the value change is less than this threshold, the change is not made.	Set this value to adjust how many bad pixels are corrected. A larger value results in fewer pixels corrected.	128 Range: [0, 819]	Use default value, or tune as needed

Table 6-2 Pixel correction parameters (Snapshot) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Remove Along Edge (Pdpc 20 Rgn)	Enables enhanced correction of bad pixels on edges.	Keep the setting at 0 to avoid unexpected artifacts on edges.	0 Enable = 1 Disable = 0	Use default value
Using Cross Channel (Pdpc 20 Rgn)	Enables cross-channel information for detection and correction.	Use default value.	1 Enable = 1 Disable = 0	Rarely tune
Saturation Threshold (Pdpc 20 Rgn)	Specifies the maximum pixel value that is detected as a bad pixel. Past this point, the pixel is saturated.	Set this value to adjust how many bad pixels are corrected. A larger value results in more pixels detected. A larger value can result in false detection.	1023 Range: [0, 16383]	Rarely tune

RELATED INFORMATION

"Examples of pixel correction parameter effects" on page 29

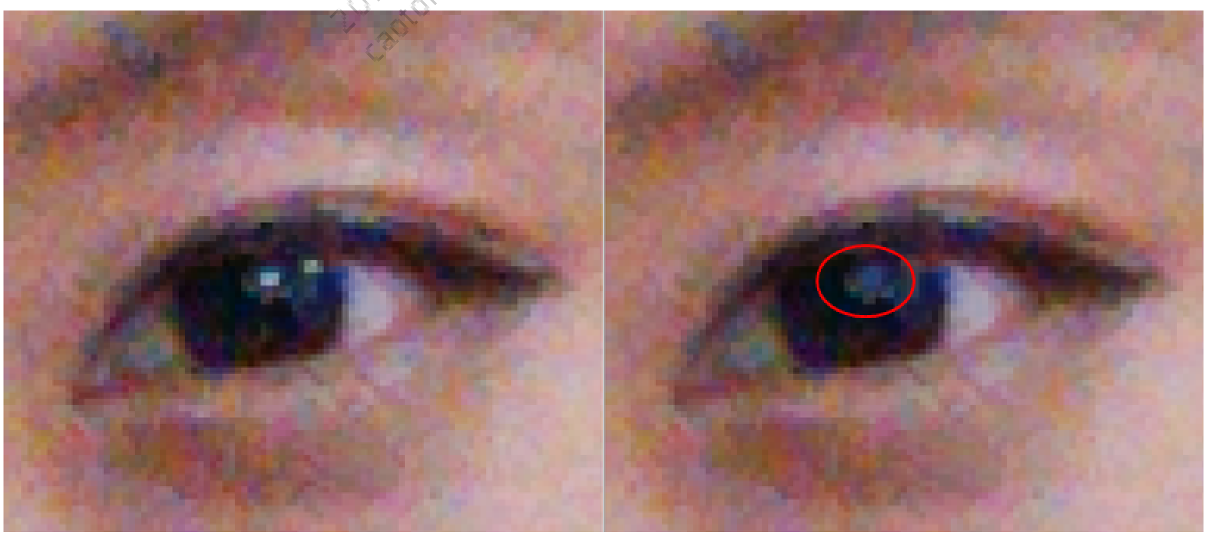
6.2.2 Examples of pixel correction parameter effects

The following examples illustrate the effects of adjusting specific pizel correction parameter values for a snapshot project.

Effects of the Fmax Pixel Q 6 parameter

(L) Weak setting, Fmax Pixel Q 6=96, correct fewer bad pixels. Circles around bad pixels not included in the detection. (R) Aggressive setting, Fmax Pixel Q 6=65, correct more bad pixels

Effects of the Using Cross Channel parameter



(L) Using Cross Channel on, correct more bad pixels. (R) Using Cross Channel off, image feature (circled in red) removed due to false detection. Using cross channel can help in this case.

6.2.3 Tune pixel correction

Prerequisites:

Before tuning the pixel correction module, determine how many lighting regions to define for this module and create them in the Parameter Editor.

The pixel correction module includes processing for sensors with or without phase detection pixels and with a non-zzHDR or zzHDR pattern.

1. On the **Pipeline** tab, click **BPS** > **PDPC**.
2. PDPC is disabled by default. To enable PD pixel correction, open the Parameter Editor, navigate to **Pdpc 20 BPS** > **Enable Section** > **Pdpc Enable**, and change the value from 0 to 1. This completes PDPC tuning for a PD pixel sensor project.
3. If tuning for a non-PD pixel sensor, click the **Region Data** button and tune the following parameters as needed:
   - Fmax Pixel Q 6
   - Fmin Pixel Q 6
   - Hot Pixel Correction Disable
   - Cold Pixel Correction Disable
   - BPC Offset
   - Bcc Offset
   - BPC Offset T 2
   - Bcc Offset T 2
   - Correction Threshold
4. Repeat the previous step for each lighting region you set up.
5. Capture test images to see if any bad pixels are missed or if there is any resolution loss or false color. Make fine tuning adjustments to the parameters as needed.
6. To save the adjustments made to pixel correction parameters, click **File** > **Save To Project**.

RELATED INFORMATION

“Add and edit region triggers” on page 15
“Pixel correction parameters” on page 26

6.3 HDR tuning concepts

Scenes that have a wide dynamic range of luminosity may contain information that is not obvious on devices. As such, the resulting images may lack detail in the light or dark regions of the image. If the camera sensor supports high dynamic range (HDR), an HDR image can be taken by combining images that are underexposed and overexposed as follows:

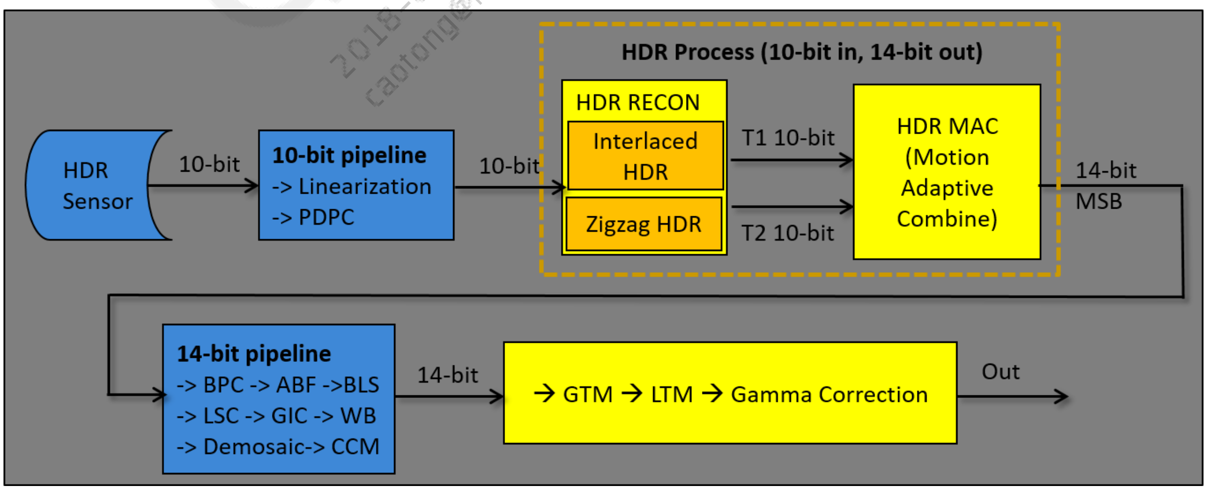
- The underexposed image provides details in the highlights region of the scene.
- The overexposed image provides details from the shadows region of the scene

The images are then reconstructed and blended to produce an image that more accurately reproduces the scene.

(L) Normal or non-HDR image, (R) HDR image

The HDR process consists of two sub-modules:

- HDR Recon – Interlaced HDR and zigzag HDR reconstruction
- HDR MAC – Motion adaptive combination



The input to the process is 10-bit zigzag data where T1 is the image with longer exposure and T2 is the image with shorter exposure. The output is scaled-up HDR Bayer to the full 14-bit range.

6.3.1 HDR parameters

The following table lists the parameters used for tuning the HDR module. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 6-3 HDR parameters

Name	Description	Data range	Default	Effect
recon_min_factor (iHDR only)	This value represents min to mid factor to force vertical edge direction.	0 to 31 allowed, but 0 to 16 meaningful.	12	Scaling factor to force vertical direction: if 16*d_min>min_factor*d_sec_min (min and mid not sufficiently apart), force vertical edge direction.
recon_flat_region_th (iHDR only)	This value represents the threshold for the flat region in edge detection.	0 to 1023	64	If d_max<th, consider current 2x2 pixel as flat region and force vertical edge direction.
recon_h_edge_th1	This value represents threshold 1 for the edge switching function.	0 to 1023	32	Controls the threshold of edge switch
recon_h_edge_dth_log2	log2(th2-th1) for horizontal edge switching function.	4 to 8	4	Controls the slope of the soft switching.
recon_motion_th1	This value represents threshold 1 for the motion switching function.	0 to 1023	512	Controls the threshold to recognize having motion.
recon_motion_dth_log2	Brief description: log2(th2-th1) for motion switching function.	4 to 8	8	Controls the slope of the motion switching.
recon_dark_th1	This value represents threshold 1 for the dark noise switching function.	0 to 1023	10	Controls the detail/noise level trade-off of dark region.
recon_dark_dth_log2	log2(th2-th1) for dark noise switching function.	0 to 4	4	Controls the slope of the dark switching.
hdr_zrec_prefilt_tap0	the strength for low-pass pre-color difference filter	0-63	0	Control the strength of filter to remove color aliasing

Table 6-3 HDR parameters (cont.)

Name	Description	Data range	Default	Effect
hdr_zrec_g_grad_th1	This value represents threshold 1 for green gradient	0-4095	32	Controls the threshold of green gradient to do directional interpolation or bilinear interpolation
mac_motion0_th1	This value represents the threshold 1 of motion switch in HDR MAC	0 to 1023	60	Controls motion switch to use longer pixels or shorter pixels
mac_motion0_th2	This value represents the threshold 2 of motion switch in HDR MAC	0 to 255	24	
mac_motion0_strength	This value represents the motion adaptation strength value.	0 to 16	8	Controls how strongly motion would affect T1/T2 combination.
mac_low_light_th1	This value represents the threshold 1 value for low light switching.	0 to 16383	0	Controls the low light switching.
mac_low_light_strength	This value represents the log light switching strength value.	0 to 16	0	Controls how strongly lowlight would affect T1/T2 combination.
mac_high_light_th1	This value represents the threshold 1 value for high light switching.	0 to 16383	232	Controls the high light switching
mac_high_light_dth_log2	This value represents the difference value for high light switching.	2 to 14	10	Control the slope of the switching function

Table 6-4 HDR reserved parameters

Name	Description	Data range	Default	Effect
hdr_recon_en	This flag when set to 1 enables HDR RECON module.	0, 1	0	This flag should be enabled when sensor is zzHDR pattern.
hdr_mac_en	This flag when set to 1 enables HDR MAC module.	0, 1	0	This flag should be enabled when sensor is zzHDR pattern.
hdr_msb_align	MSB Align for HDR module	0, 1	1	zzHDR default as MSB

Table 6-4 HDR reserved parameters (cont.)

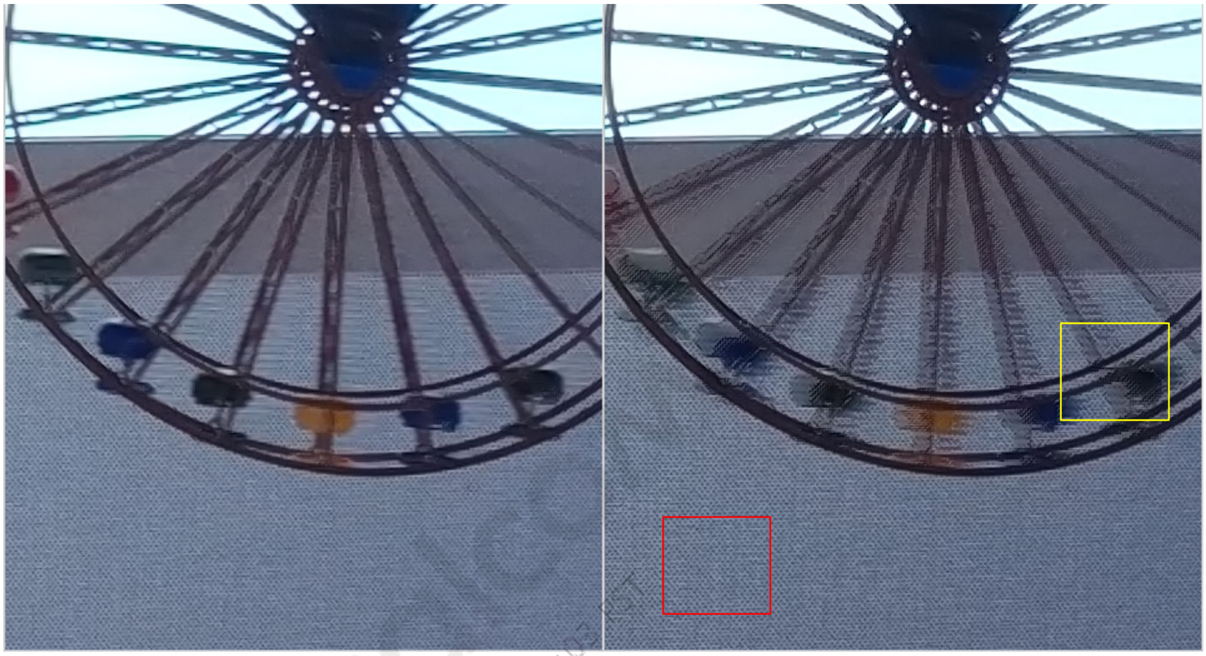
Name	Description	Data range	Default	Effect
hdr_zrec_g_grad_dth_log2	log2(th2-th1) for green interpolation switching function.	0 to 12	5	Controls the slope of the soft green interpolation switching
hdr_zrec_rb_grad_dth_log2	log2(th2-th1) for red/blue interpolation switching function.	0 to 12	5	Controls the slope of the soft red/blue interpolation switching
recon_edge_lpf_tap0 (iHDR only)	Tap0 value for low-pass filter applied before edge detection.	0 to 5	3	
mac_motion_dilation	Size of motion dilation max filter, 5 means -5 to 5, i.e. 11-tap filter.	0 to 5	5	Controls the dilation (motion mask) size.
mac_motion0_dt0	A parameter to avoid dividing by zero.	1 to 63	1	To avoid dividing by zero.
mac_low_light_dth_log2	log2(th2-th1) for low light switching.	2 to 14	6	Controls the slope of low light switching.
mac_smooth_enable	Enable smoothing for final output in MAC.	0 to 1	1	Enables smoothing filter.
mac_smooth_th1	Th1 of the smoothing switch in motion region.	0 to 256	192	Control th1 to smooth in motion region
mac_smooth_dth_log2	log2(th2-th1) for motion adaptive smoothing.	2 to 8	6	Controls the slope of motion adaptive smoothing.
mac_smooth_tap0	Tap0 value for the low-pass filter in motion adaptive smoothing.	0 to 5	5	Controls the filter strength

6.3.2 Examples of HDR parameter effects

These examples illustrate the effects of adjusting specific HDR parameter values.

Effects of the recon_motion_th1 parameter

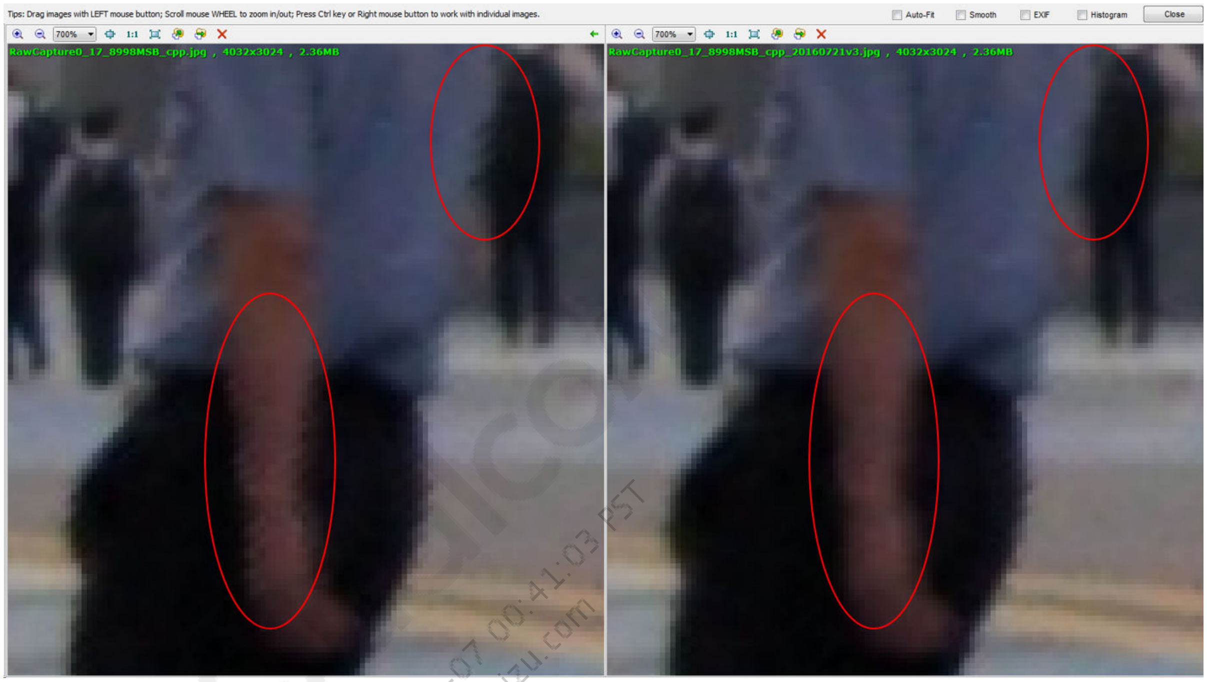
The recon_motion_th1 parameter controls the threshold 1 motion switching function. In the image at right in the following example, the parameter is set at 1023 which results in less interpolated T1 and more scaled T2, thereby producing more resolution and more zigzag artifacts. In the image at left, the parameter is set at 184 which results in more interpolated T1 and less scaled T2, thereby producing less zigzag artifcats and less resolution.



(L) recon_motion_th1 = 184, (R) recon_motion_th1 = 1023

Effects of the recon_dark_th1 parameter

The recon_dark_th1 parameter represents threshold 1 for the dark noise switching function and controls the detail/noise level trade-off in dark regions. In the example below, increasing the parameter value reduces the zig-zag artifacts in the dark regions.



(L) recon_dark_th1 = 0, (R) recon_dark_th1 = 10

Effect of mac_motion parameters

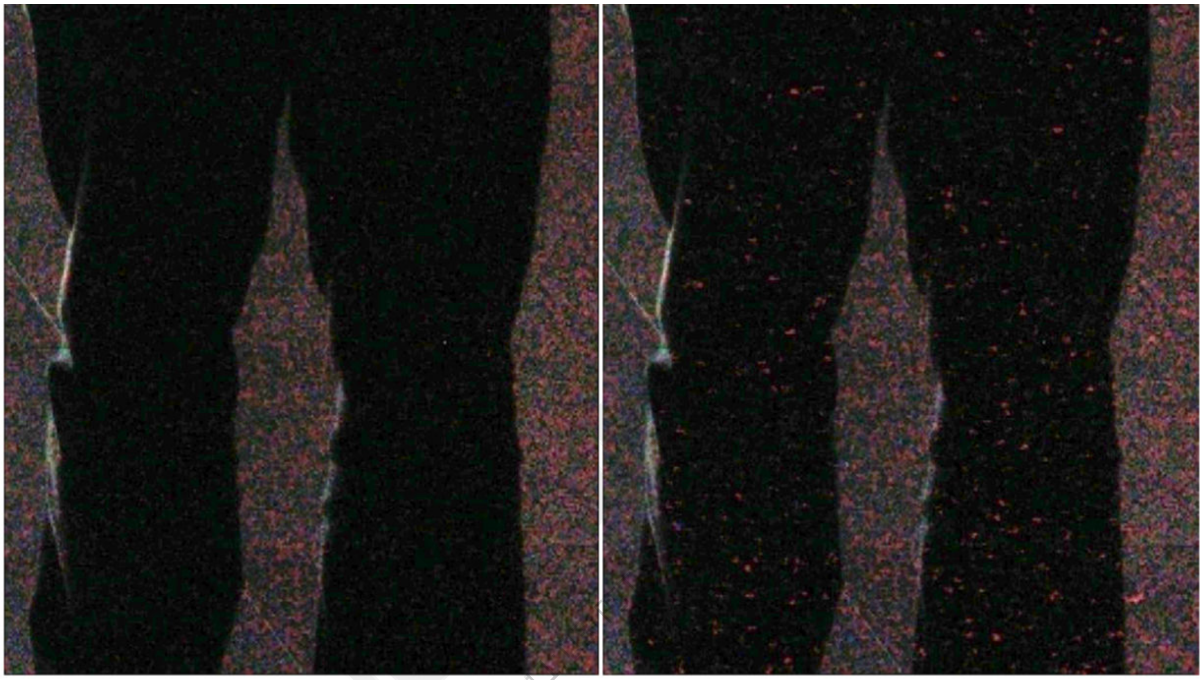
The mac_motion parameters work together to control the motion switch in HDR MAC to use longer or shorter pixels and to control the motion adaption strength value.

(L) Shorter exposure pixels, sharper but noiser (R) longer eposure pixels, more motion blur

In the examples above, the image at left has lower values for the mac_motion threshold parameters and the image at right has higer values. The range for the mac_motion threshold paraemeters is 0 to 1023, the lower the value the shorter the pixels. The following table shows the parameter values for each example.

Parameter	Example (L)	Example (R)
mac_motion0_th1	16	60
mac_motion0_th2	10	24
mac_motion0_strength	16	8

If mac_motion0_th1 and mac_motion0_th2 are set too small, it will over-detect dark regions as motion regions and, as a result, blend more noisy T2 in the MAC output. Compare the two images in the following example where the image on the right has smaller values for the parameters and shows more noise in the dark regions.

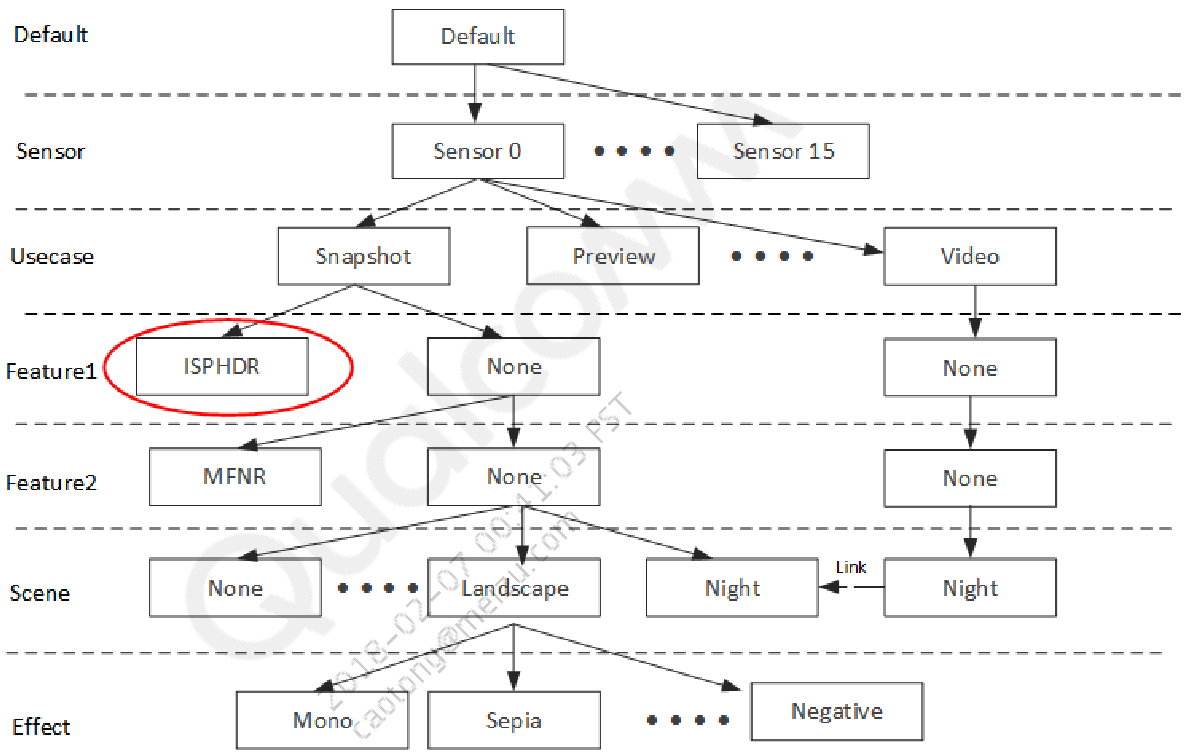


(L) mac_motion_th1 = 16, mac_motion_th2 = 10 (R) mac_motion_th1 = 8, mac_motion_th2 = 5

6.3.3 Tune HDR

Prerequisites:

If the camera sensor supports high dynamic range (HDR) and you want to offer an HDR setting on the device, add the ISPHDR mode to the sensor|usecase that will offer the feature. Do this for each sensor|usecase that will offer an HDR setting.



Add the ISPHDR mode for each sensor and use case that will offer an HDR setting

Tuning HDR primarily involves retuning other modules to compensate for the sensitivity afforded by the HDR module. The following is the general procedure for HDR tuning.

1. Accept the defaults for the HDR modules.
2. Retune the coordinating BPS modules. The tone curve tuning in the GTM and LTM modules is the most important for dynamic range.
   - PDPC – To remove bad pixels, tune PDPC so as not to harm the resolution but still remove bad pixels. Set the T1 and T2 offsets as T2 offset = T1 offset * exposure ratio. The T1 offset parameters are `bpc_offset` and `bcc_offset`. The T2 offset parameters are `bpc_offset_t2` and `bcc_offset_t`.
   - GIC – Retune GIC to remove dots and zigzags by HDR and tone curve. Green imbalance is more obvious in a zzHDR sensor.
     - Using `noise_std_lut` as same as ABF is recommended.
     - Using `gic_correction_strength` is recommended by exp_sensitivity_ratio.

    - Lower `thin_line_noise_offset` is recommended by exp_sensitivity_ratio.
    - When exp_sensitivity_ratio is getting higher, the lower data is filled in and has detail information that should not be removed.
  - ABF – Retune ABF to remove dots and zigzags by HDR and tone curve. Retune the noise_std_lut when T1/T2 has a gain difference.
  - GTM – Tone curve tuning is the most important for dynamic range. Adjust the tone curve to generate 10-bit output by high dynamic range.
  - HNR –Noise levels will be different because of changes to exposure ratio and tone mapping curves, so retuning is recommended after GTM changes. It is permissible to change only strength by intensity. Multiply to lnr_gain_arr based on noise level by intensity.
    - Increase in noise profile = gtm_hdr / gtm_no_hdr
    - lnr_gain_arr[0…33] *= Increase in noise profile[64/2]

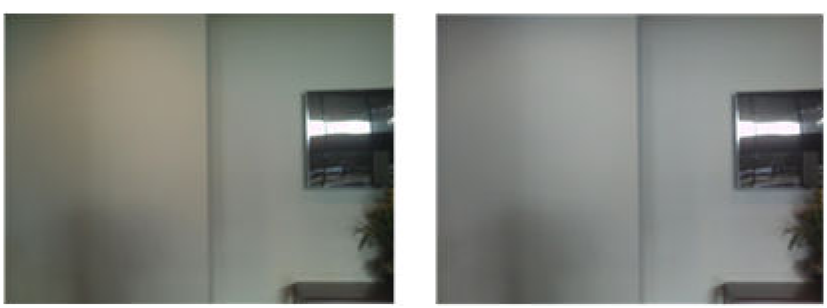
3. Simulate and test.
4. Adjust HDR parameters as necessary.
   a. To achieve good resolution, tune the `recon_h_edge_th1`/ `recon_motion_th1`/ `recon_dark_th1` parameters in the HDR module.
   b. To remove motion effect, tune the `mac_motion0_th1`/`mac_motion0_th2`/ `mac_motion_strength` parameters in the HDR module.
   c. Tune other HDR parameters as needed when there are critical HDR issues with the default values.
5. Repeat procedure until performance goals are met.

RELATED INFORMATION

"Tune ABF" on page 62
"Tune GIC" on page 53
"Tune pixel correction" on page 30

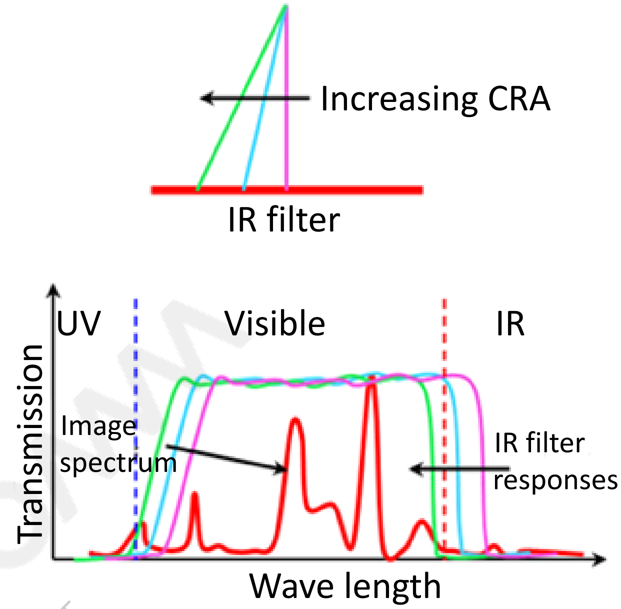
6.4 LSC tuning concepts

Based on the size and quality of the camera lens, the center of an image may appear brighter than the corner regions. Lens shading correction (LSC) corrects brightness attenuation and tint (uneven color shading).

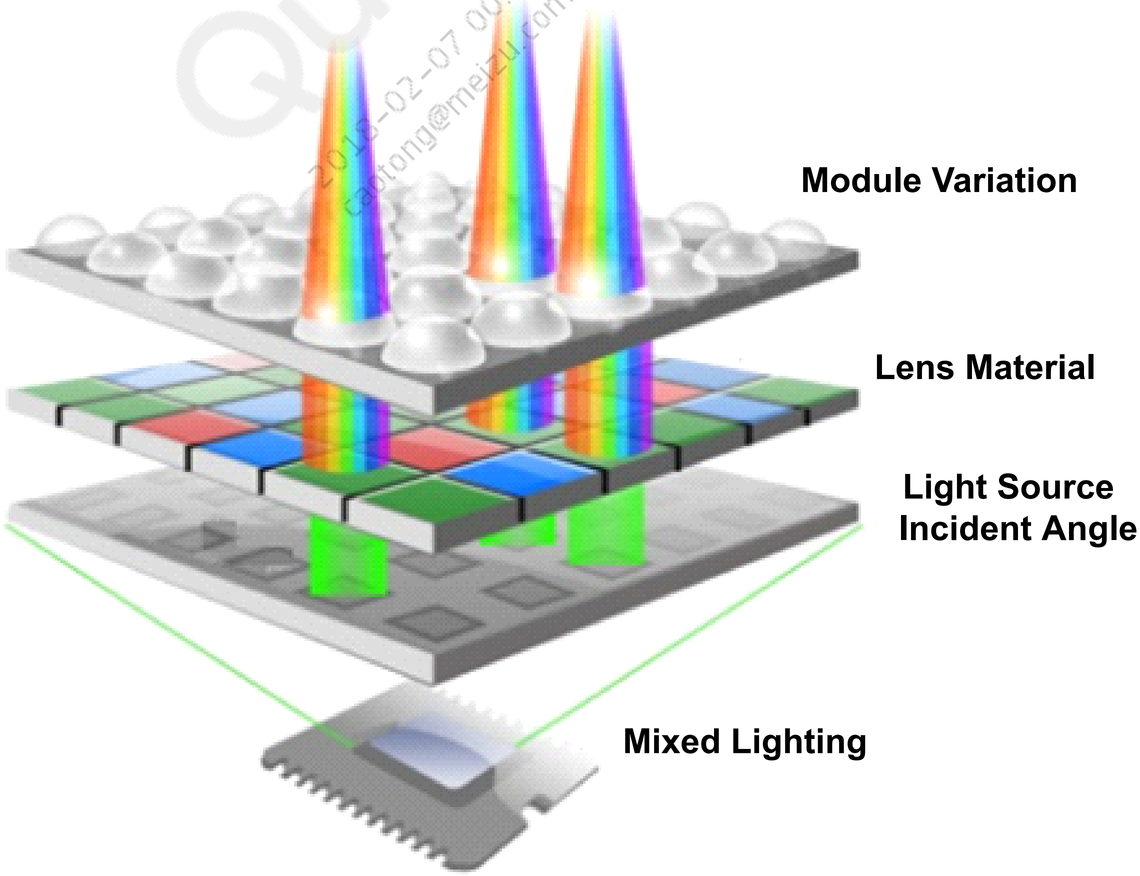


(L) Original; (R) Corrected

Tint usually results from using an infrared (IR) filter when capturing images.

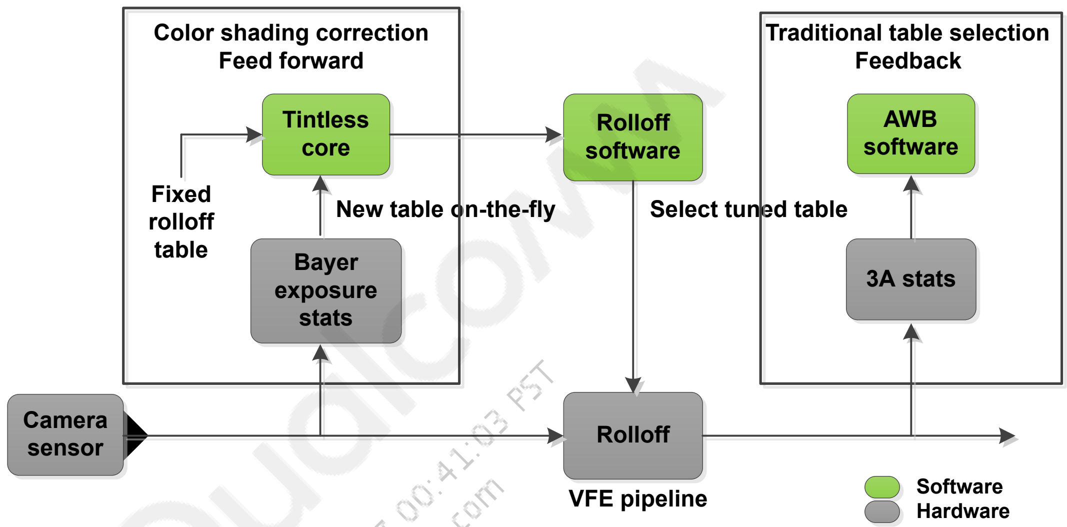


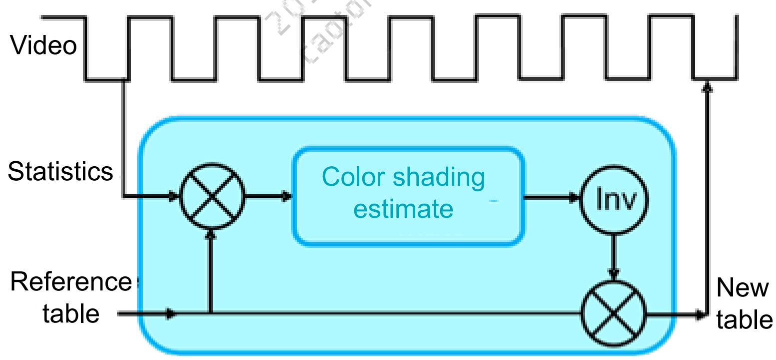
Sensor module variation, lens material and lighting conditions could also cause tint.

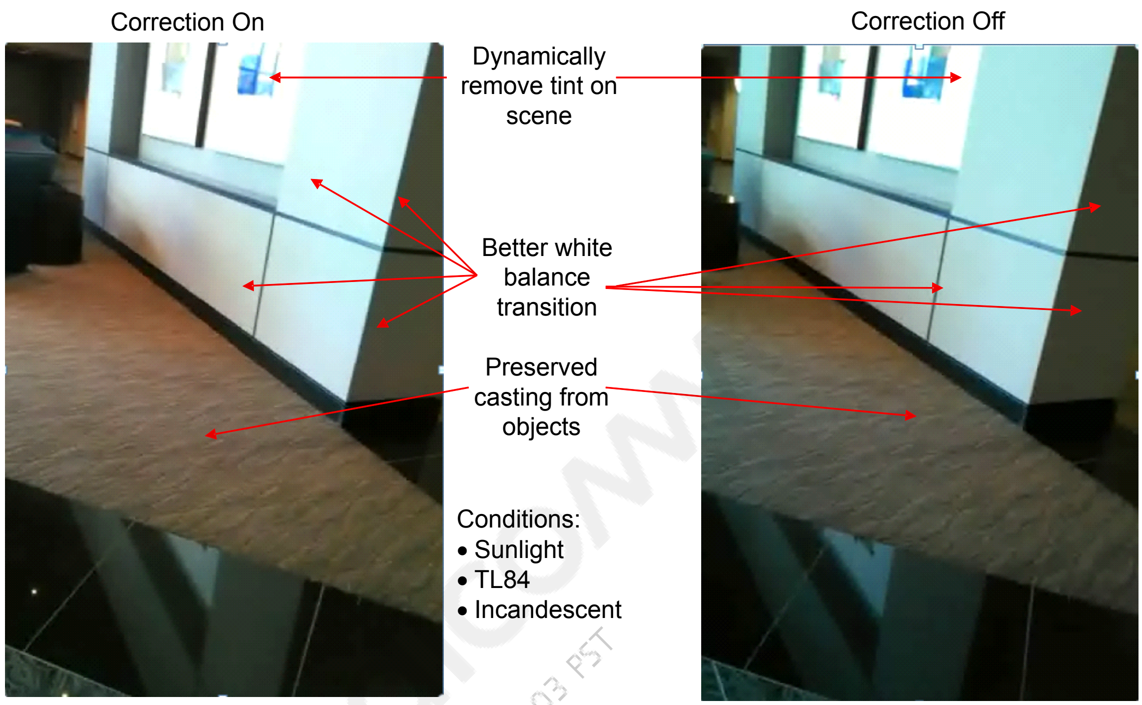


The following figure illustrates where color shading is corrected in the VFE. The main steps in the process are:

1. Values from the reference table are applied and residual stats are obtained.
2. Color tint (shading) is estimated based on the residual stats.
3. The inverse value of color tint is multiplied by the reference table value to obtain the corrected value.



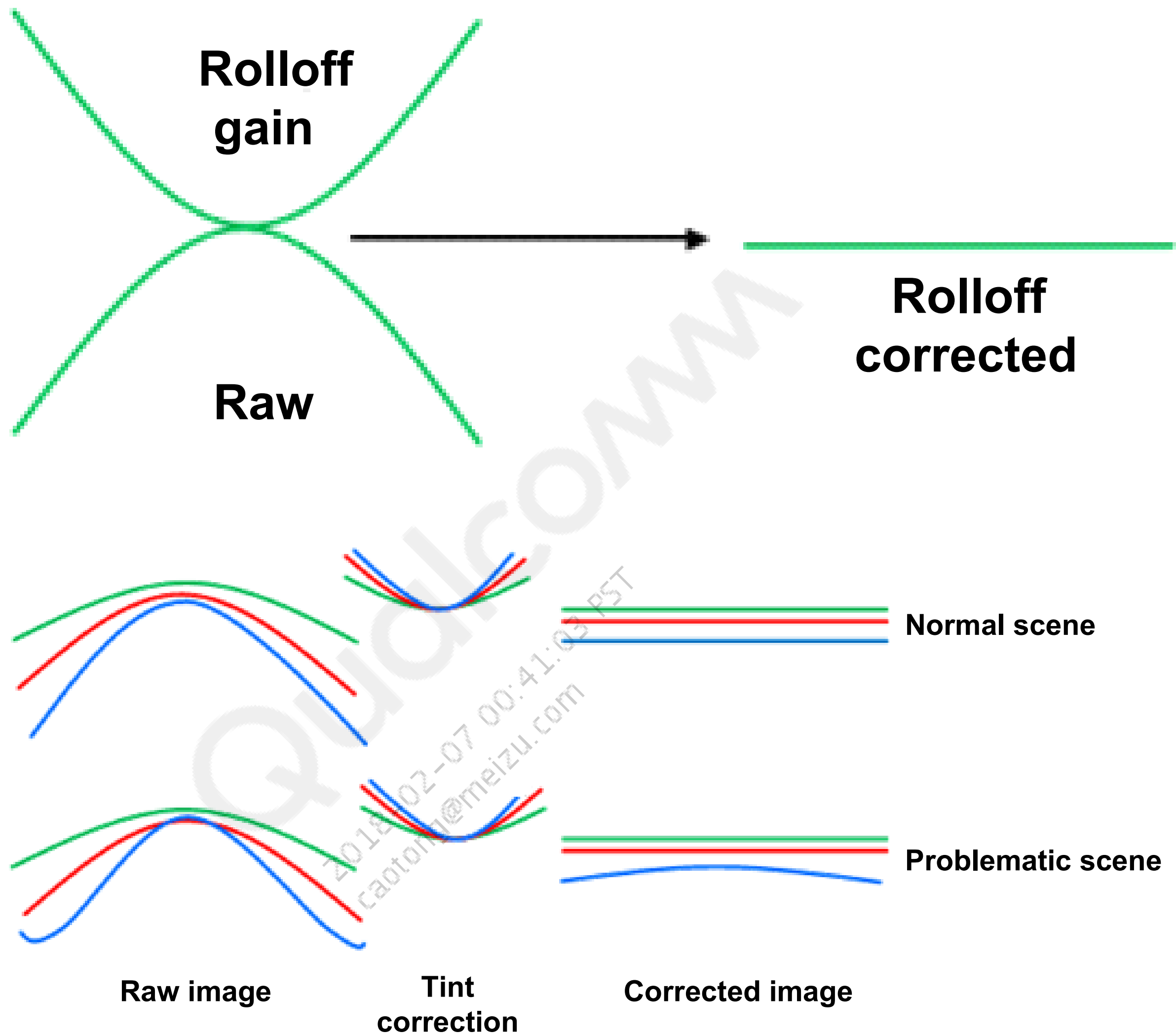




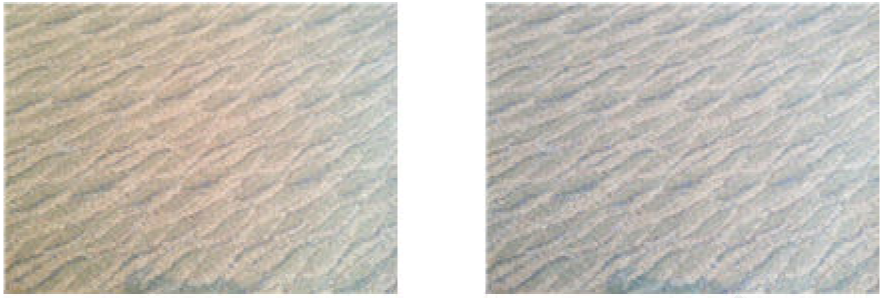
The conventional method of using AWB along with multiple calibration tables cannot compensate for individual module variation, uncalibrated or different illuminants, etc. Dynamic color shading correction estimates color shading from the BE stats and continuously updates the reference table. It provides a more robust solution and reduces the calculation burden by using a single calibration table.

(L) Conventional method (AWB + multitable); (R) Dynamic color shading correction (tintless)

LSC is applied to Bayer raw images to reduce the computation load. Instead of modeling the curve and performing polynomial calculations, a linear piecewise approximation with the help of LUTs is used. Each color has its own LUT, so the brightness attenuation and color shift problems can be solved simultaneously.



Examples of lens shading correction

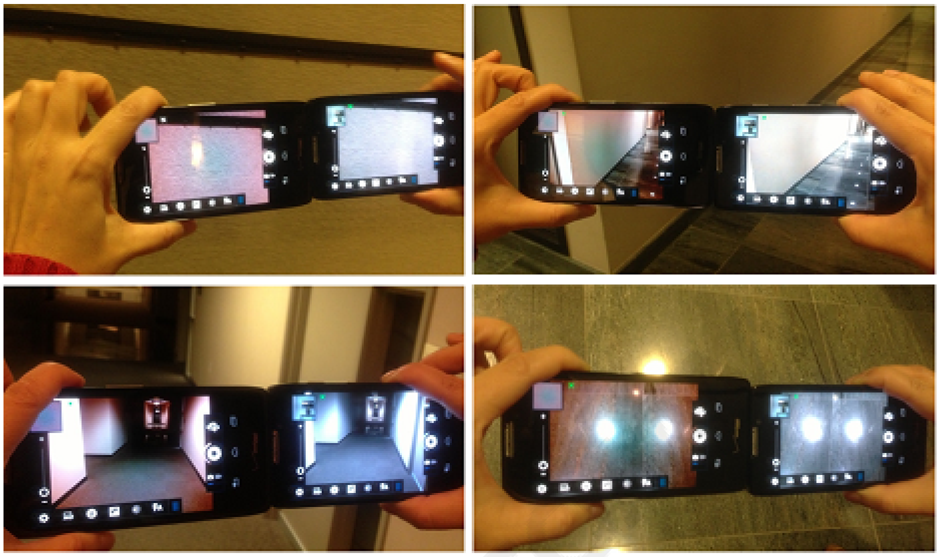


(L) Tintless off, indoor. (R) Tintless on, indoor

(L) Tintless off, mixed light. (R) Tintless on, mixed light

(L) Tintless off, lightbooth; (R) Tintless on, lightbooth

The following figure illustrate the same image using a commercial solution (left) compared to tintless.



6.4.1 Tune LSC

Based on the size and quality of the camera lens, the center of an image may appear brighter than the corner regions. Lens rolloff tuning corrects brightness attenuation and nonuniform color shading (tint).

Prerequisites:

Before tuning the LSC module, determine how many lighting regions you need to define and create them in the Parameter Editor.

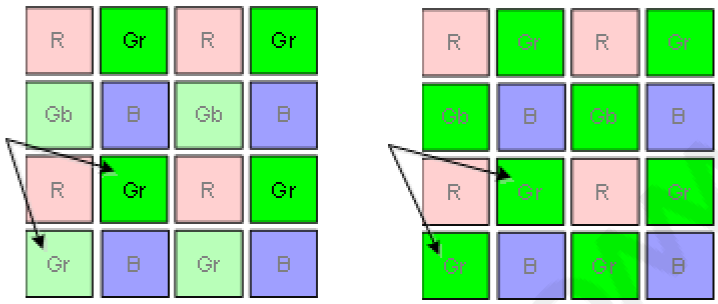
1. On the **Pipeline** tab, click **LSC** and then click a region in the Regions group.

   NOTE Tintless correction is enabled by default. To disable tintless correction, go to the Parameter Editor and navigate to **SW_CONTROL** > **Tintless 20 Sw** > **Enable Section** > **Tintless En**, and enter a value of 0 to disable this parameter. Low light and normal light conditions use the same set of images. Different percentage correction values are applied to the two different lighting conditions to generate LSC compensation tables.

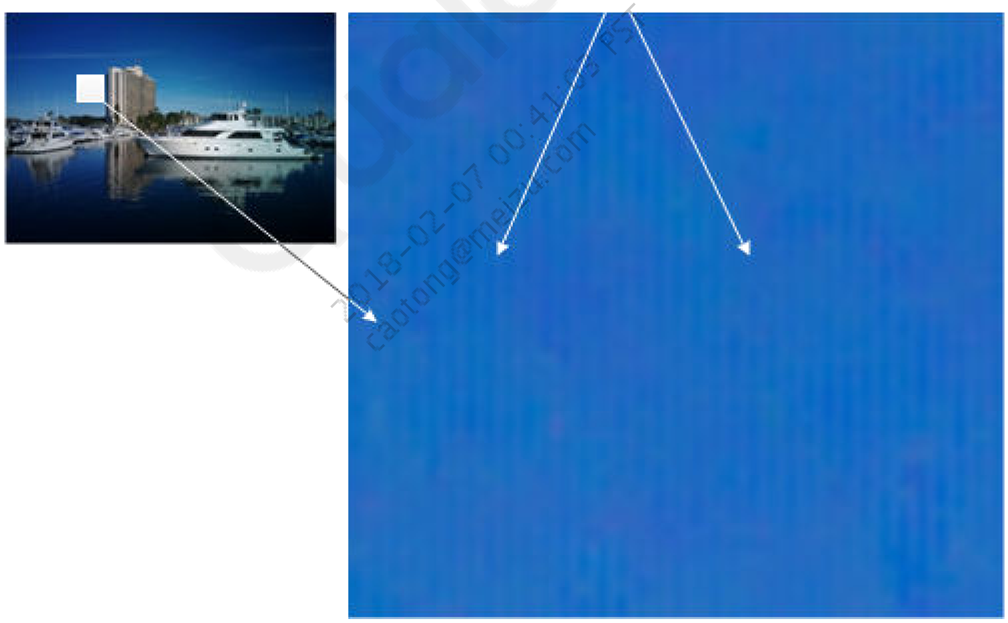
2. In the **LSC** section, click **Load Image** and select a rolloff raw image. Click **Open**.
3. Use the **Radius** percentage slider to adjust the area (from the center of the image) to be corrected. Use the **Correction** percentage slider to adjust the amount of correction to apply. Click **Optimize**.
4. Continue to apply additional radius percentage and correction percentage settings until the updated rolloff curve achieves a more flattened appearance.
5. Repeat this procedure for the remaining regions.
6. When complete, click **File** > **Save To Project** to preserve the tuned LSC settings.

6.5 GIC tuning concepts

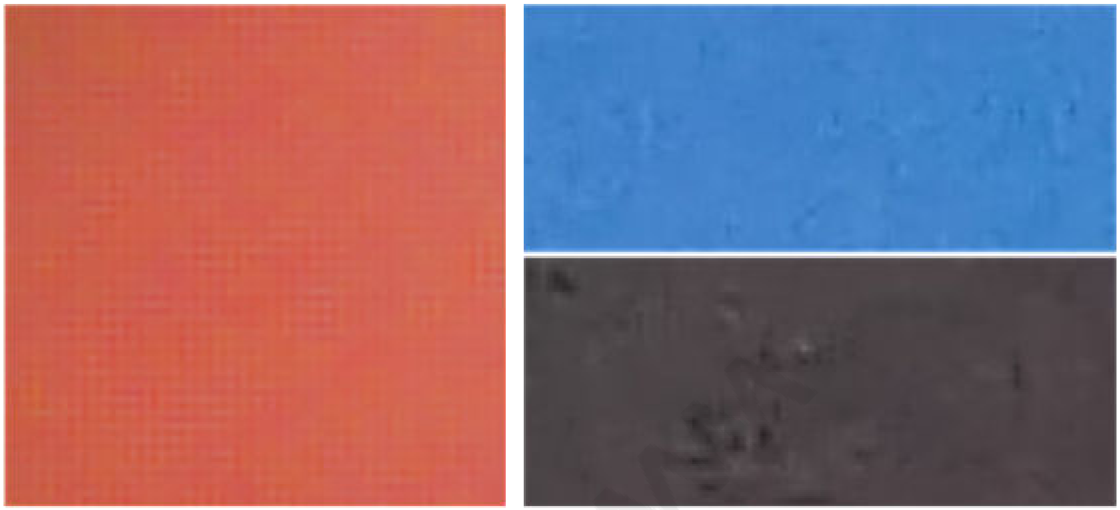
Some Bayer sensors may have a strong imbalance between the Gb and Gr channels. The artifacts caused by this imbalance may not be fully corrected by the ABF noise reduction block in the Bayer domain. Green imbalance correction (GIC) removes the banding artifacts caused by the different sensitivities between Gr and Gb channels.



(L) Sensor showing gr/gb mismatch due to different channel sensitivity creating a gr/gb imbalance artifact. (R) Sensor without gr/gb imbalance, channels have same gr/gb sensitivity.

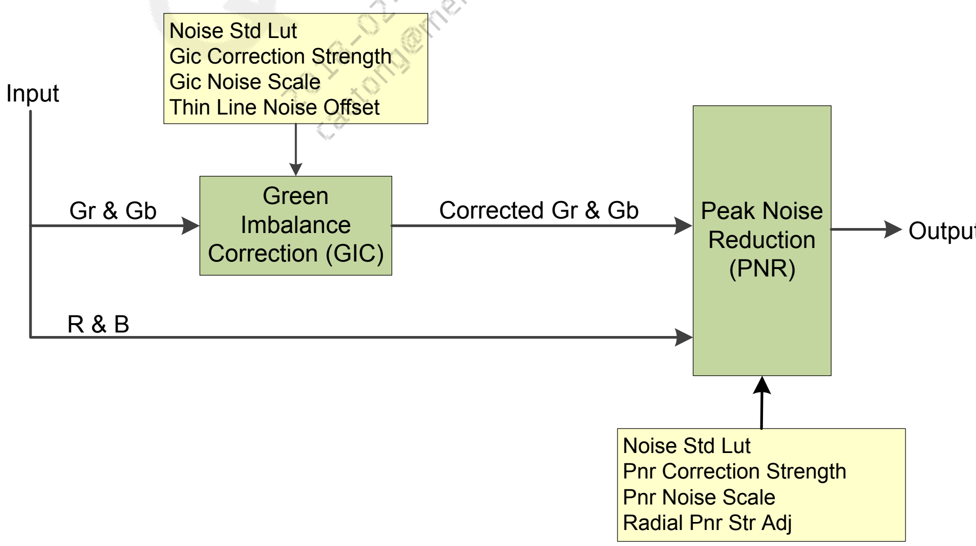


(R) Enlarged sky insert from left image shows vertical banding caused by an imbalance between the gr and gb channels in the Bayer domain.



(L) Global green imbalance pattern. Global grid patterns, uniform throughout the image. (R) Local green imbalace artifacts. Local cross hatch artifacts that typically show up in uniform areas; very common in noisy sensors.

The Qualcomm Spectra 2xx GIC module combines green imbalance correction with peak noise reduction (PNR). The PNR submodule is used to remove peak noise in flat regions. Imbalance between the Gb and Gr channels is corrected first. Corrected Gb and Gr channels then work with R and B channels to reduce peak noise. The following figure shows the GIC and PNR flow, and the parameters used to correct green imbalance.



RELATED INFORMATION

"GIC parameters" on page 49
"Tune GIC" on page 53

6.5.1 GIC parameters

The following table lists the parameters used for tuning the GIC module. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 6-5 Basic GIC tuning parameters

Parameter name	Description	Tuning	Factory default	Tuning required
Gic Global Enable (Enable section)	Enables/disables entire GIC/PNR module	Set to 1 to use GIC or PNR (Disable – 0)	1	Use default value
Enable Gic (Gic 30 Rgn Section)	Enables/disables GIC submodule	▪ Disable – 0 ▪ Enable – 1	1	Use default value
Enable Pnr (Gic 30 Rgn section)	Enables/disables PNR submodule	▪ Disable – 0 ▪ Enable – 1	1	Use default value
Noise Std Lut	Table that plots the local noise intensity (in terms of standard deviation) at 65 points; known as the noise profile	▪ The table is computed by the Chromatix tool. Generally, the noise profile does not need tuning. Compute for each sensor. ▪ Higher values result in weaker GIC and stronger PNR	Range = [0, 512.0]	Advanced tuning – Use the computed values; if it is necessary to tune this parameter, request guidance from the CE team
Gic Noise Scale	Noise scale factor for green imbalance correction used for Gb/Gr channel difference clamping; defines amount of noise	Higher value results in weaker correction (higher threshold for what must be corrected)	1.0 Range = [0, 15.98]	Yes
Gic Correction Strength	Adjusts green imbalance correction strength; use to increase or decrease noise strength	▪ Lower value results in stronger correction ▪ Maximum noise reduction – 0 ▪ No noise reduction –1	0.6 Range = [0, 1]	Yes
Thin Line Noise Offset	Detects thin lines and offsets edge noise to prevent over-correction of Gr/Gb mismatch	Higher value results in weaker thin line detection; thin lines may be impacted by GIC	400 Range = [0, 16383]	Yes

Table 6-5 Basic GIC tuning parameters (cont.)

Parameter name	Description	Tuning	Factory default	Tuning required
Pnr Noise Scale	Noise scale factor of PNR for each channel; use to separate flat region and texture region and reduce peak noise	■ Higher value results in more area that can be detected as flat region; amount of correction would be less ■ Take care when tuning this parameter as setting can impact textured region	{1.0, 1.0, 1.0, 1.0} Range = [0, 15.98]	Yes
Pnr Correction Strength	Adjusts peak noise reduction strength; one value adjusts all four channels	Lower value results in stronger peak noise reduction	0.6 Range = [0, 1]	Yes
Radial Pnr Str Adj	Adjusts radial (corner) peak noise reduction strength	■ Adjust PNR strength based on radial distance percentage ■ Smaller value results in stronger PNR in corners	1.0 Range = [0, 15.98]	Advanced tuning – Use the computed values; if it is necessary to tune this parameter, request guidance from the CE team

RELATED INFORMATION

"Examples of GIC parameter effects" on page 51

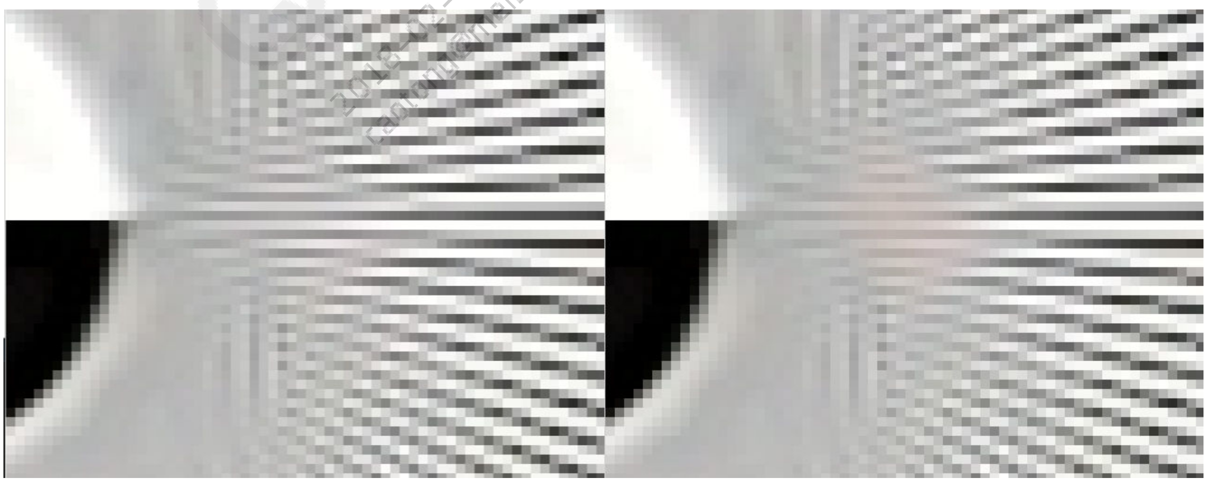
6.5.2 Examples of GIC parameter effects

These examples illustrate the effects of adjusting specific GIC parameter values for a snapshot project.

Effects of the Gic Correction Strength parameter

(L) Gic Correction Strength = 1.0; (R) Gic Correction Strength = 0.6

Effects of the Thin Line Noise Offset parameter

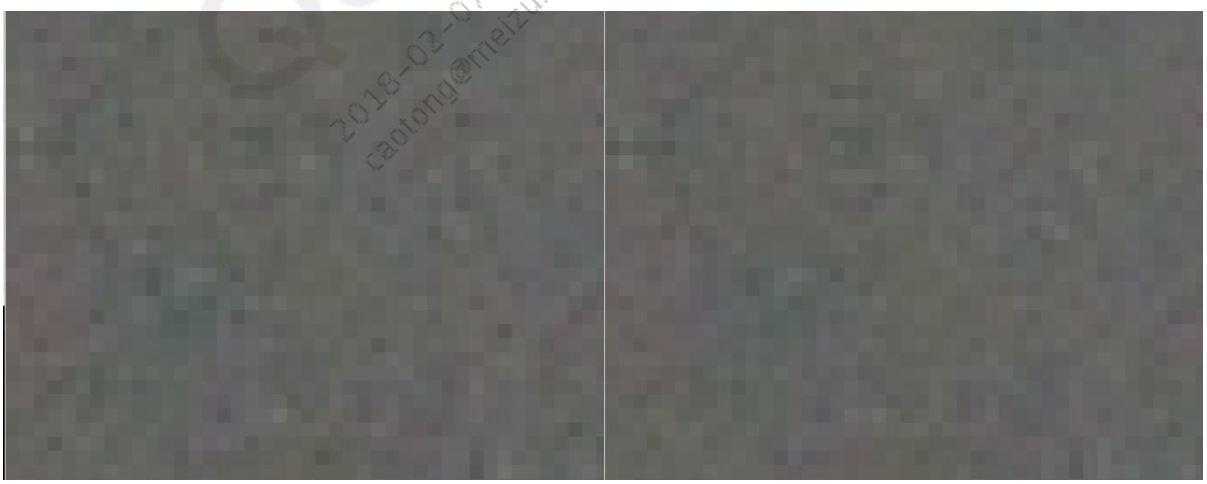


(L) Thin Line Noise Offset = 400 (lower setting preserves thin line); (R) Thin Line Noise Offset = 2000 (higher value results in fuzzier line)

Effects of the Enable Pnr parameter

(L) PNR disabled; (R) PNR enabled

Effects of the Pnr Correction Strength parameter



(L) Pnr Correction Strength = 1.0; (R) Pnr Correction Strength = 0.6

6.5.3 Tune GIC

Prerequisites:

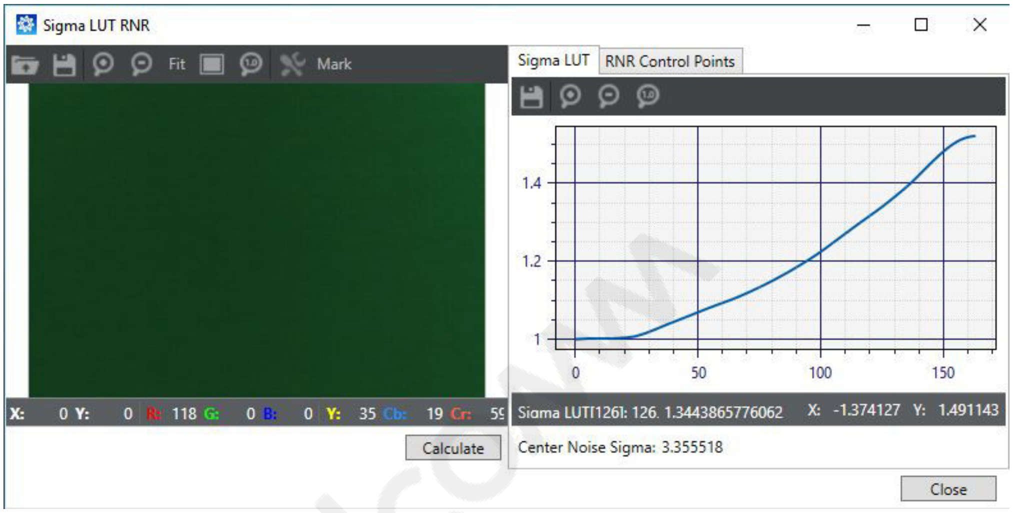
Before tuning the GIC module, determine how many lighting regions you need and create them in the Parameter Editor.

1. On the **Pipeline** tab, click **BPS** > **GIC**.
2. Calibrate the noise profile as follows:
   a. Click a region and click the **Load Image** button.
   b. Select a raw image of an MCC chart taken in the appropriate lighting condition for that region.
   c. Set the MCC markers. Drag the top-left MCC grid marker to the center of the top-left color patch and drag the bottom-right MCC grid marker to the center of the bottom right color patch, so that all of the grid markers align within the color patches.

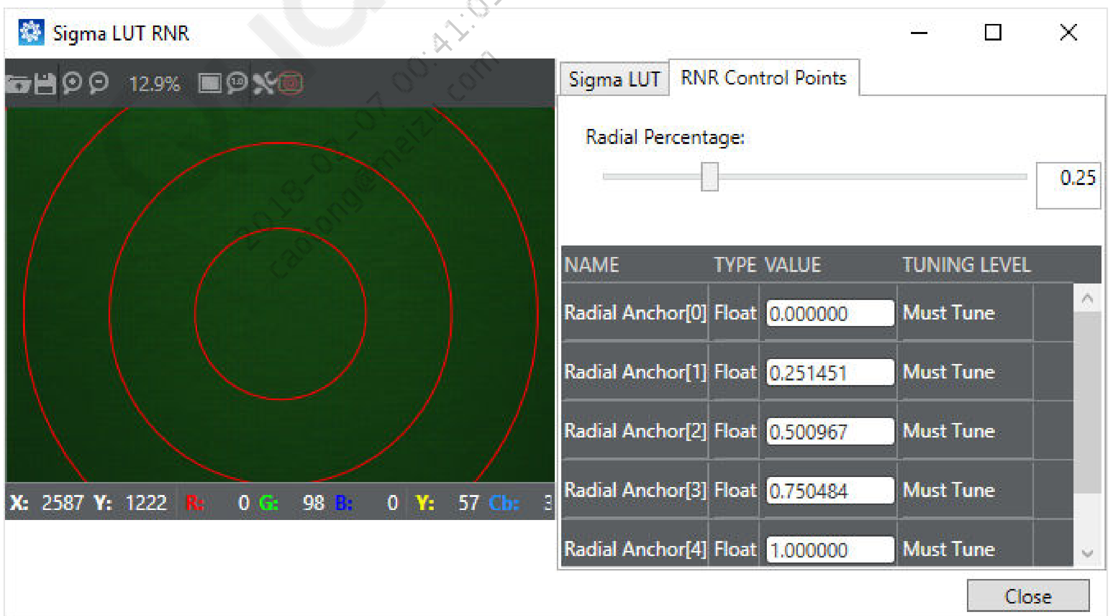
   If the image is taken with a fisheye lens, to set the MCC markers for tuning, hold the Ctrl Key (or Shift Key) while moving the markers

   a. Click **Calculate** to compute the noise profile. View the resulting noise profile graph on the **Noise Profile** tab.

3. To calibrate the radial noise reduction, click **Sigma LUT RNR** and do the following steps:
   a. Click the **Sigma LUT** tab and open a gray chart raw image for the appropriate lighting region. Click **Calculate** to generate the Sigma LUT plot.



   b. Click the **RNR Control Points** tab, and adjust the **Radial Percentage** slider to set the second control point. The tool computes the other points automatically.



4. Click the **Region Data** tab and tune the following GIC parameters as needed:
   - Gic Noise Scale
   - Gic Correction Strength
   - Thin Line Noise Offset
   - Pnr Noise Scale
   - Pnr Correction Strength

- Noise Std Lut (advanced tuning)
- Radial Pnr Str Adj (advanced tuning)

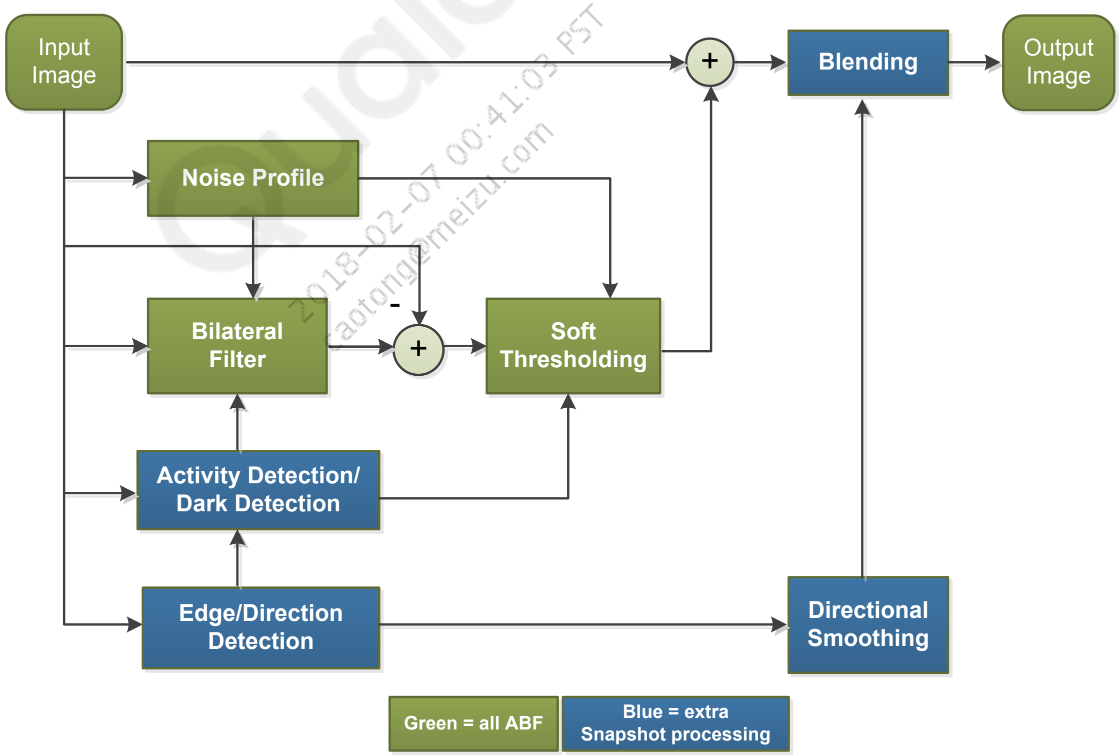
5. Repeat all steps for each lighting region you set up.
6. To save the adjustments made to GIC parameters, click **File** > **Save To Project**.

RELATED INFORMATION

"Add and edit region triggers" on page 15
"GIC parameters" on page 49

6.6 ABF tuning concepts

The Adaptive Bayer Filter (ABF) tuning module performs denoising for the Bayer domain. Regular 2D convolution filtering deteriorates image quality, particularly when applied on Bayer pixels. ABF can adapt its kernel based on neighboring pixel values to reduce noise without blurring sharp edges. ABF includes a bilateral filter followed by soft thresholding. Some additional processing is included to get high quality snapshots, as shown in this illustration.



ABF is not the primary module for denoising in Qualcomm Spectra 2xx. Noise tuning is primarily accomplished with the following modules:

- Hybrid Noise Reduction (HNR) + ANR for snapshot mode
- Advanced Noise Reduction (ANR) + Temporal Noise Filtering (TF) for preview/view mode

Coordination with other modules

In general, the factory default values for ABF parameters are adequate because ABF is not the primary module for denoising. Use mild settings for the ABF filtering parameters to preserve more detail and texture. Allow the ANR, HNR, and TF modules to control the noise.

ABF processes pixels in a linear domain while other modules are in a non-linear YUV domain. Therefore, it is good to have a certain level of noise reduction in the linear domain, especially for getting high quality snapshots.

ABF coordinates with other tuning modules in the following ways:

- The kernel size of ABF is similar to the kernel size of GIC, HNR and the first pass of ANR.
  - For snapshot, the balance is among GIC, HNR, and ABF
  - For video, the balance is among GIC, ABF, and the first pass of ANR
- If using a special RCCB or RGBW sensor, ABF can help reduce noise in the early part of the pipeline
- When doing initial spatial tuning, rely on edge alignment in ASF and disable directional smoothing available in ABF.
- When doing initial spatial tuning, rely on HNR and disable activity-based strength adjustment available in ABF.
- During normal tuning, turn off the min-max filter and rely on these other blocks that have the same function:
  - BPC/BCC/PDPC (single BPC)
  - Peak noise reduction (PNR) in GIC

Key ABF parameters for snapshot mode

A noise profile is a 65-entry LUT that represents noise statistics for a specific sensor. The first to last entries correspond to a noise value (in terms of standard deviation) at intensity levels 0-255. The local average intensity is used to index the noise profile entry. A larger value for the `Noise Stdlut Level` equals a stronger denoising. Calibrate a noise profile for each sensor using the Chromatix tool. Usually the computed noise profile does not require any manual adjustment.

The `Denoise Strength` parameter is used to adjust denoise strength to balance between the four channels. A larger value results in a stronger noise reduction. Unless the noise level is different among the channels, do not tune this parameter.

The `Edge Softness` parameter, together with the noise profile, controls the strength of the bilateral filter. The greater the edge softness value, the more denoising is done. Less noise results in flatter uniform regions, less detail, and softened edges.

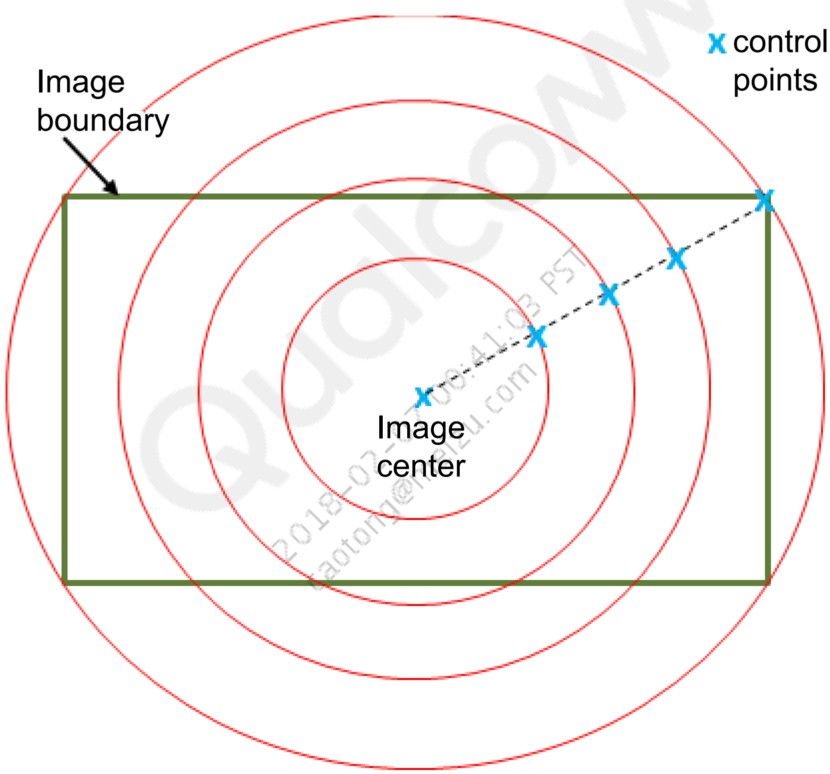
Noise preservation determines how much noise to retain in the soft thresholding step. These settings are helpful for removing dark artifacts, especially crosshatch patterns, but dark details are impacted. There are two parameters that affect noise preservation: `Noise Prsv Base` and `Noise Prsv Anchor`.

The ABF module has some *regional* parameters that must be tuned separately for different lighting conditions. An ABF region is defined by three specific nested triggers: DRC Gain Triggers, HDR AEC Triggers, and AEC Triggers. Define as many regions as you need to accomplish the preferred tuning specificity. It is important to define the regions for ABF before you begin tuning. Refer to *Establishing Regions for Tuning Modules* for examples and instructions.

Radial Noise Reduction parameters

Noise in an image field of view (FOV) is not uniformly distributed. The corner of an image often has more noise due to lens rolloff correction. Radial noise reduction (RNR) is used to uniformly distribute noise in the FOV. ABF is early in the pipeline, so Lens Shading Correction (LSC) information later in the pipeline can be used to adjust denoising.

RNR processing partitions an image into four regions using five control points as illustrated in the following diagram.



The `Radial Anchor Table`, in the ABF Reserve parameters, specifies the five radial control points that are defined by the radius percentage from the center point. The first and fifth points are hard coded as the image center (0.0) and corner (1.0). Use the Chromatix tool to tune the second point and the tool generates the remaining points.

Edge softness and noise preservation can be adjusted based on radial distance. Use the `Radial Noise Prsv Adj` to adjust noise preservation gain and `Radial Edge Softness Adj` to adjust edge softness gain at each control point. A smaller gain results in a larger edge softness or smaller noise preservation, which results in more denoising. To use a stronger ABF at the image corners, program decreasing values into the radial gain adjustments.

Refer to the *ABF Parameters* section for more detail about how to adjust the parameter values.

RELATED INFORMATION

"ABF Parameters" on page 59
"Tune ABF" on page 62

6.6.1 ABF Parameters

The following table lists the parameters used for tuning the ABF module for Snapshot mode. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 6-6 Basic ABF tuning parameters (Snapshot)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Bilateral En (ABF 40 BPS > Enable Section)	Enable flag for snapshot ABF.	Set to 1 to use the ABF module.	1	Take default value
Minmax En (ABF 40 BPS > Enable Section)	Enable flag to enable built-in min-max pixel filter.	If BPC module is on, this feature is likely off (0).	0	Take default value.
AEC Exp Control (ABF 40 BPS > Control Method Section)	Set exposure control triggers to be based on Gain or Lux Index.	Select the appropriate trigger type.	Gain	Yes, required
Cross Plane En (ABF 40 BPS > Chromatix ABF 40 Reserve)	Enable flag for Gr/Gb cross-channel processing.	Always set to 1 to help correct GrGb mismatch.	1	Take default value
Noise Std Lut Table (ABF 40 BPS > ABF 40 Rgn Data)	A table that plots the noise intensity (in terms of standard deviation) at 65 points. Known as the Noise Profile. Affects both the bilateral filter and soft thresholding steps of ABF. Same profile is used for all channels.	The table is computed by the Chromatix tool. Generally, the noise profile does not need tuning. Compute for each sensor. A larger noise value = stronger denoising. In the rare case where the computed noise profile must be adjusted, use the Noise Std chart on the **Graphs 2** tab to adjust individual points.	Computed by the Chromatix tool based upon the uploaded image. Intensity value range = [0, 512]	Take computed values
Denoise strength	Used to adjust denoise strength to balance between the four channels.	Larger value = stronger noise reduction Do not tune unless the noise level is different among the channels.	{1.0, 1.0, 1.0, 1.0} Range = [0.0, 1.0]	
Edge Softness (ABF 40 BPS > ABF40 Rgn Data)	Works with the noise profile to control bilateral filter strength for each channel.	Focus on image quality at center. A larger value = more denoising (e.g., flatter uniform regions, less detail, softened edges) (see ABF examples)	{3.0, 3.0, 3.0, 3.0} Range = [0.0, 15.99]	Yes

Table 6-6 Basic ABF tuning parameters (Snapshot) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Noise Prsv Base (ABF 40 BPS > ABF40 Rgn Data)	▪ Base value for how much noise to retain in each area designated by 5 knee points. ▪ Helpful for removing dark artifacts, especially crosshatch patterns, but dark details are impacted. ▪ Entries 0-4 adjust RG channels ▪ Entries 5-9 adjust BG channels	A larger value = less denoising Bottom graph of **Graph 1** tab plots Noise Prsv Base (Y-axis) against Noise Prsv Anchor (X-axis). Adjust values by changing in Parameter Editor or by moving the points vertically on the graph. Select to show RG or BG channels on the graph. (see *Examples of ABF parameter effects*)	{0, 0, 0, 0, 0, 0, 0, 0, 0, 0} Range = [0.0, 1.0]	Yes
Noise Prsv Anchor (ABF 40 BPS > ABF40 Rgn Data)	Intensity anchor points for Noise Prsv Base. Bottom graph of **Graph 1** tab plots Noise Prsv Base (Y-axis) against Noise Prsv Anchor (X-axis). Select to show RG or BG channels on the graph.	Adjust values by changing in Parameter Editor.	{0, x, x, x, 1.0} Range = [0.0, 1.0]	Yes
Radial anchor base table (ABF 40 BPS > ABF 40 Reserve)	Table that specifies five radial control points that partition the image into 4 regions. Each control point is defined by the radius percentage from the center point.	▪ The first and fifth entries are hard coded as the image center (0.0) and corner (1.0). ▪ Use the Chromatix tool to tune the 2nd point and the tool generates the 3rd and 4th points.	{0,0.4,0.6,0.8, 1.0} Range = [0, 1]	Yes, using Sigma LUT RNR dialog > Radial Control Points tab
Radial Noise Prsv Adj (ABF 40 BPS > ABF40 Rgn Data)	A decreasing table of noise preservation gain entries that correspond to each of the five RNR control points. Used for soft thresholding. Top graph of **Graph 1** tab plots Radial Noise Prsv Adj (Y-axis) against the Radial Anchor points (X-axis).	▪ Default is no adjustment at control points. ▪ Smaller gain results in smaller Noise Preservation, which results in more denoising. ▪ Entries [0] and [4] are hardcoded at 1.0. ▪ Adjust values by changing in Parameter Editor. (see *Examples of ABF parameter effects*)	{1,1,1,1,1} Range = [0.0, 1.0]	Yes
Radial Edge Softness Adj (ABF 40 BPS > ABF40 Rgn Data)	An increasing table of edge softness gain entries that correspond to each of the five RNR control points. Used for bilateral filter. Middle graph of **Graph 1** tab plots Radial Edge Softness Adj (Y-axis) against the Radial Anchor points (X-axis).	▪ Default is no adjustment at control points. ▪ Smaller gain results in larger Edge Softness, which results in more denoising. ▪ Entries [0] and [4] are hardcoded at 1.0. ▪ Adjust values by changing in Parameter Editor.	{1,1,1,1,1} Range = [1.0, 15.99]	Yes

NOTE Parameters that are located in the ABF 40 Rgn Section of the **Parameter Editor** must be tuned for each region that you set up.

RELATED INFORMATION

"Examples of ABF parameter effects" on page 61

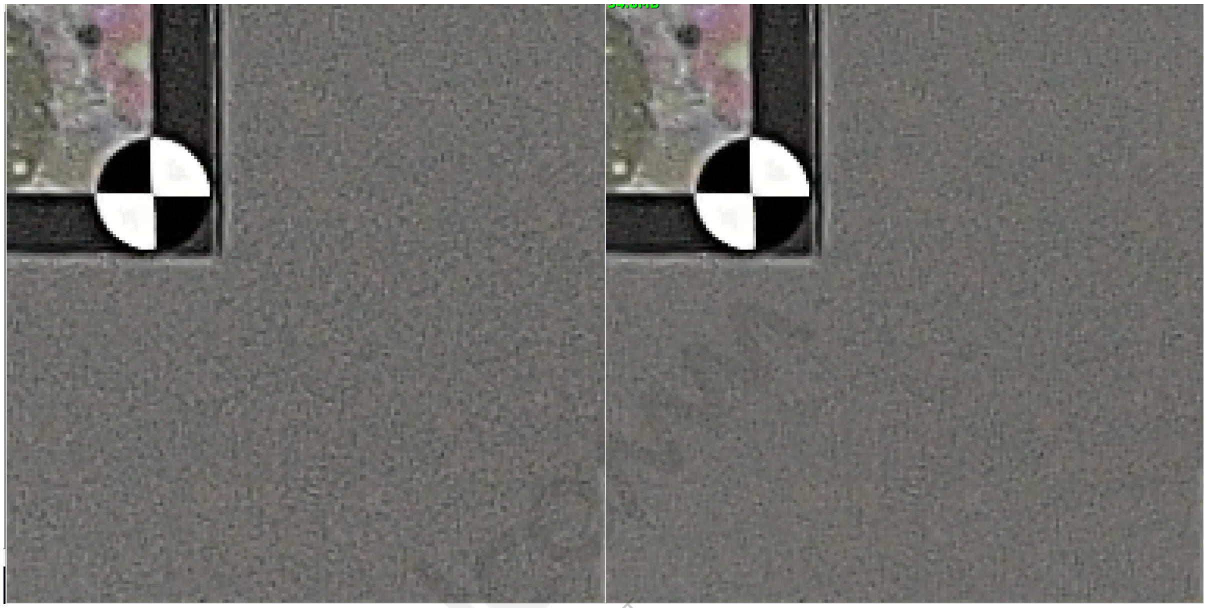
6.6.2 Examples of ABF parameter effects

These examples illustrate the effects of adjusting specific ABF parameter values for a snapshot project

Effects of the edge softness parameter

(L) edge_softness=3, (C) edge_softness=8, (R) edge_softness=12. A larger value results in softened edges, less noise, and less detail.

Effects of the noise preserve base parameter



(L) noise_prsv_base = all 1.0 array, (R) noise_prsv_base = all 0.6 array. The associated radial anchor table = [0.0, 0.4, 0.6, 0.8, 1.0].

The noise_prsv_base parameter defines the percentage of preserved noise at each radial anchor point. A lower value for noise_prsv_base results in stronger noise reduction as shown in the illustration

6.6.3 Tune ABF

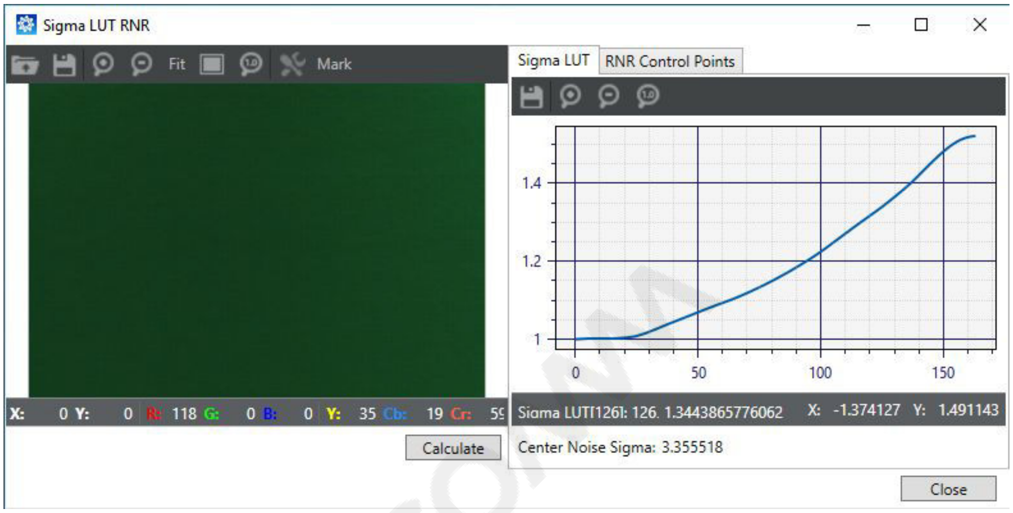
Prerequisites:

Before tuning the ABF module, determine how many lighting regions you need and create them in the Parameter Editor.

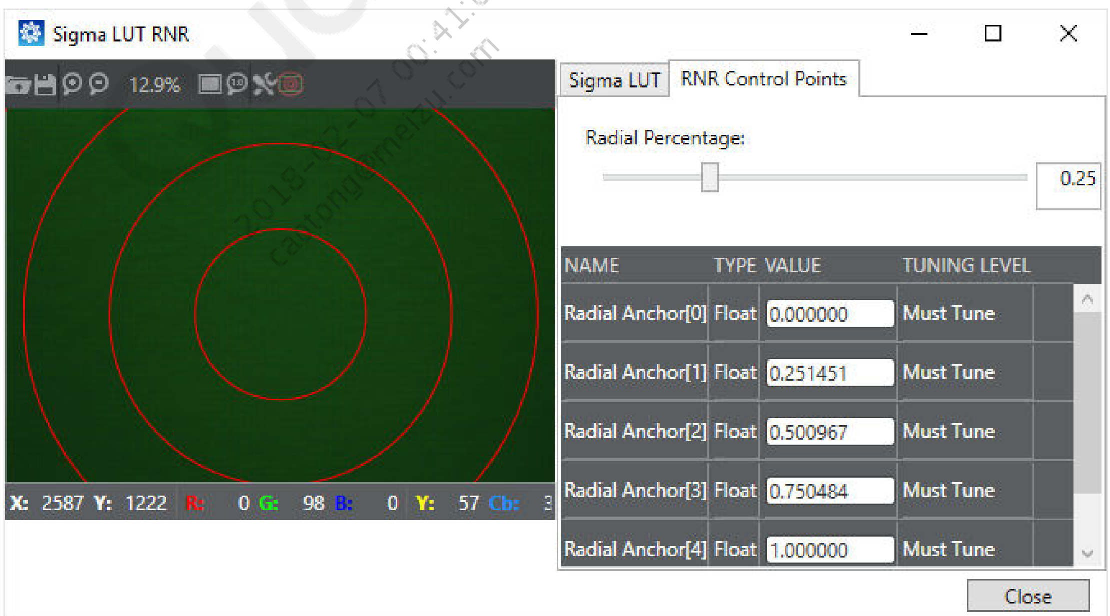
Regular 2D convolution filtering deteriorates image quality, particularly when applied on Bayer pixels. To reduce the noise without blurring sharp edges, use an adaptive Bayer filter (ABF), which can adapt its kernel based on neighboring pixel values.

1. On the **Pipeline** tab, click **BPS** > **ABF**.
2. Calibrate the noise profile as follows:
   a. Click a region and click the **Load Image** button.
   b. Select a raw image of an MCC chart taken in the appropriate lighting condition for that region.
   c. Set the MCC markers. Drag the top-left MCC grid marker to the center of the top-left color patch and drag the bottom-right MCC grid marker to the center of the bottom right color patch, so that all of the grid markers align within the color patches. **Note**: If the image is taken with a fisheye lens, to set the MCC markers for tuning, hold the Ctrl Key (or Shift Key) while moving the markers.
   d. Click **Calculate** to compute the noise profile. View the resulting noise profile graph on the **Noise Profile** tab.

3. To calibrate the radial noise reduction, click **Sigma LUT RNR** and do the following steps:
   a. Click the **Sigma LUT** tab and open a gray chart raw image for the appropriate lighting region.



   b. Click **Calculate** to generate the Sigma LUT plot.
   c. Click the **RNR Control Points** tab, and adjust the **Radial Percentage** slider to set the second control point. The tool computes the other points automatically.



   d. Click **Close**.
4. (Optional) bilateral filtering is enabled by default. To disable bilateral filtering, open the Parameter Editor, navigate to **ABF 40 BPS** > **Enable Section** > **Bilateral En**, and change the value from 1 to 0.
5. (Optional) The built-in min-max pixel filter is disabled by default. To turn on min-max pixel filter, open the Parameter Editor, navigate to **ABF 40 BPS** > **Enable Section** > **Minmax En**, and change the value from 0 to 1.

6. (Optional) Gr/Gb cross-channel processing is enabled by default. To disable cross-channel processing, open the Parameter Editor, navigate to **ABF 40 BPS** > **ABF 40 Reserve** > **Cross Plane En**, and change the value from 1 to 0.
7. As needed, click the **Region Data** tab and tune the following ABF parameters:
   - Edge Softness
   - Denoise Strength
   - Noise Prsv Base
8. As needed, tune the following ABF RNR parameters in the **Graphs 1** tab:
   - Radial Noise Prsv Adj
   - Radial Edge Softness Ad
   - Noise Prsv Anchor
9. Use the default values for advanced parameters related to activity detection, dark desaturation, edge direction detection, and directional smoothing. If it becomes necessary to tune those parameters, request guidance from the CE team.
10. Repeat all steps for each lighting region you set up.
11. (Optional) To see the results of the tuning adjustments, click **Simulate**. Otherwise, simulate after the entire pipeline is tuned.
12. To save the adjustments made to ABF parameters, click **File** > **Save To Project**.

RELATED INFORMATION

"ABF Parameters" on page 59
"Add and edit region triggers" on page 15

6.7 Demosaic tuning concepts

Use the default values for the demosaic module. The demosaic module filters and reconstructs a complete RGB color image from color filter array (CFA) samples.

A digital color image typically comprises three color samples, namely red (R), green (G), and blue (B), at each pixel location. However, using three separate color sensors for measuring the three R, G, and B color values at each pixel location in a digital camera is very expensive. Thus, most digital cameras employ a single-chip image sensor, where each pixel in the image sensor is covered with an R, G, or B color filter for capturing the color information. The mosaic of tiny color filters covering the pixels in a single-chip image sensor is referred to as the color filter array (CFA). The most commonly used CFA is the Bayer mosaic formed by the replication of two-by-two sub-mosaics, with each sub-mosaic containing two green, one blue, and one red filter.

6.8 HNR tuning concepts

The HNR block is a blending architecture to reduce luminance noise, but keep texture detail. This block is located at the end of the Bayer Processing Segment (BPS), and therefore is available only for snapshot. It consists of DCT-based frequency domain noise reduction, gradient smoothing, and spatial domain blending. Unlike conventional spatial domain noise reduction, DCT-based frequency domain distinguishes weak texture and noise. Using this, HNR reduces noise while preserving edges and texture.

(L) Conventional spatial domain noise reduction; (R) DCT-based frequency domain noise reduction

HNR attributes:

- Spatial and frequency-domain noise reduction
  - Noise reduction controls: level, radial, chroma, skin, flatness, and frequency
- Blending with input image with weighting factors
  - Blending controls: level, chroma, and skin
- Gradient smoothing for jaggy edge treatment
  - Gradient smoothing: percentage
- Multiple controls in both noise reduction and blending stages.
- Luma noise reduction only
- 10-bit input and 10-bit output

Parameters to enable controls

Parameters in the enable section:

- `Hnr nr enable`: noise reduction enable
- `Hnr blend enable`: noise blending enable

Parameters in the reserve section:

- `Lnr en`: level-based noise reduction enable
- `Rnr en`: radial-based noise reduction enable
- `Cnr enable`: chroma-based noise reduction enable
- `Snr enable`: skin-color based noise reduction enable
- `Fd snr enable`: face detection assisted SNR enable
- `Fnr enable`: flatness control enable
- `Lpf3 en`: gradient smoothing enable
- `Blend cnr en`: chroma-based blending enable
- `Blend snr en`: skin-color based blending enable

6.8.1 HNR Parameters

The following table lists the parameters used for tuning the HNR module. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 6-7 Basic HNR tuning parameters

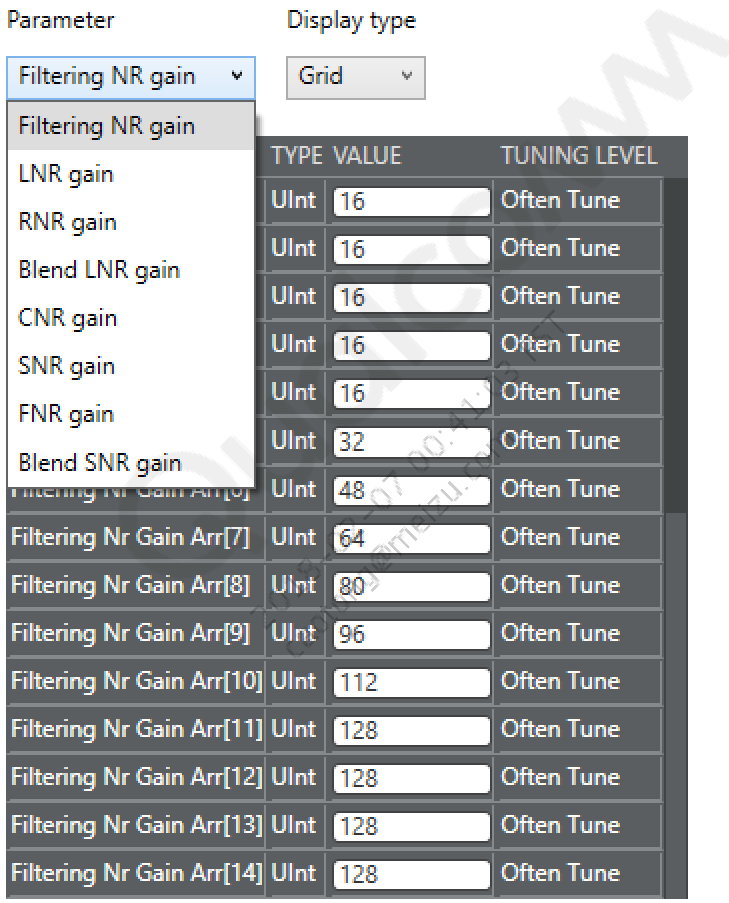
Parameter Name	Description	Tuning	Factory Default	Must Tune?
Filtering nr gain arr	Overall strength of HNR. Length: 33	Lower value = stronger noise reduction Suggestion: Piece-wise smooth curve	Chromatix tool calibrates with raw MCC chart image. All 128 array Min: 0, Max: 128	Yes, required
Lnr gain arr	Adjust strength according to intensity level. Length: 33	Lower value = stronger noise reduction Suggestion: Piece-wise smooth curve	All 32 array Min: 0, Max: 32	Yes, required
Radial noise prsv adj	Adjust strength based on radial distance. Length: 7	Lower value = stronger noise reduction Suggstion: Monotonically decreasing	Chromatix tool calibrates using a flat-field image. All 1.0 array Min: 0, Max: 1.0	Yes, required
Blend lnr gain arr	Blending weight of input luma channel. Length: 17	Higher value = noiser input Suggestion: "U" shape curve to preserve more detailes on both dark and highlight regions	All 32 array Min: 0, Max: 128	Yes, required
Cnr gain arr	Partition UV plane to 4x4 mesh with 25 points. Length: 25	Lower value = stronger noise reduction	All 32 array (disabled) Min: 0, Max: 32	Use default
Snr gain arr	Adjust noise reduction strength based on skin color. Length: 17	Lower value = cleaner flat regions Suggestion: Use mild adjustment	All 32 array (disabled) Min: 0, Max: 32	Use default
Fnr gain arr	Adjust noise reduction strength based on local variations. Length: 17	Lower value = cleaner flat regions Suggestion: Use mild adjustment	All 32 array (disabled) Min: 0, Max: 32	Use default
Blend snr gain arr	Blending weight of input luma channel. Length: 17	Lower value = smoother skin region	All 32 array (disabled) Min: 0, Max: 32	Use default

Table 6-7 Basic HNR tuning parameters (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Blend cnr adj gain	Blending weight of input luma channel based on defined regions. Sharing the region definition with noise reduction stage. Length: 1	Lower value = smoother output in defined chroma regions	32 Min: 0, Max: 63	Use default
Lpf3 percent	Controls threshold for smoothing. Length: 1	Higher value = smoother edges	16 Min: 0, Max: 255	Use default
Face boundary	Lower bond factor for facial areas detected by face detection. Length: 1	Lower value = smaller facial areas	1.2 Min: 0, Max: 8	Use default
Face transition	Lower bond factor for facial areas detected by face detection. Length: 1	Larger value = larger transition areas	2.0 Min: 0, Max: 8	Use default

6.8.2 Tune HNR

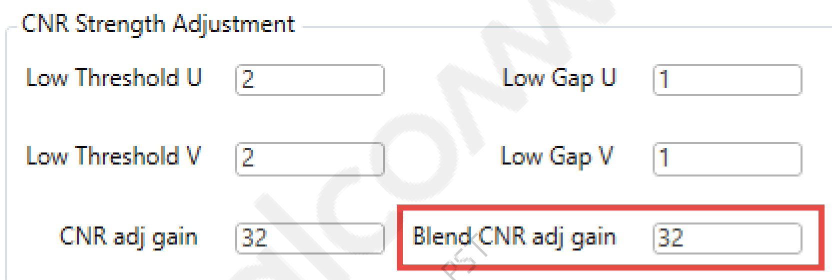
1. Enable features in the Enable and Reserve sections, as needed. In the Parameter Editor navigate to **BPS** > **Hnr 10 BPS** > **Enable Section** (for the Enable section), and **BPS** > **Hnr 10 BPS** > **Chromatix Hnr 10 Reserve** (for the Reserve section).
2. Tune the important parameters first. In Chromatix 7 navigate to the **Pipeline** tab, click **BPS** > **HNR** and then click a region in the Regions group. Use the parameter drop down box to adjust the values of the following parameters, as needed.



- `Filtering nr gain arr`: noise reduction strength
- `Lnr gain arr`: level-based noise reduction adjustment
- `Blend lnr gain arr`: level-based noise blending adjustment
- `Radial noise prsv adj`: radial-based noise reduction adjustment

NOTE Adjust `Radial noise prsv adj` by clicking the **Radial Anchor Data** button and adjusting the values, as needed. It is shared between all of the regions.

3. Tune for different preferences or special cases. Use the parameter drop down box to adjust the values of the following parameters, as needed.
   - `Cnr gain arr`: chroma-based noise reduction strength adjustment
   - `Snr gain arr`: skin color-based noise reduction strength adjustment
   - `Fnr gain arr`: adjust noise reduction strength in flat regions
   - `Blend snr gain arr`: skin color-based blending adjustment
   - `Blend cnr adj gain`: chroma-based blending adjustment
     - Navigate to the **CNR Strength Adjustment** box and adjust `blend cnr adj gain`, as needed.



   - `Lpf3 percent`: reducing jaggy artifacts along edges
     - Navigate to the **LPF3 Percent** box and adjust the value, as needed.

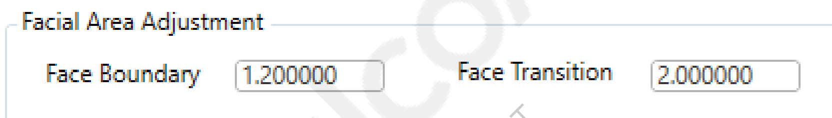
4. Tune noise reduction strength for facial areas.
   a. Adjust skin color definition. Click on the **Region Data** button to adjust the following parameters, as needed.

Skin Hue Min	Float	-1.390000
Skin Hue Max	Float	-0.400000
Skin Y Min	Float	0.125000
Skin Y Max	Float	1.000000
Skin Saturation Min Y Max	Float	0.080000
Skin Saturation Max Y Max	Float	0.800000
Skin Saturation Min Y Min	Float	0.320000
Skin Saturation Max Y Min	Float	1.000000
Skin Boundary Probability	UInt	8
Skin Percent	Float	12.000000
Skin Non Skin To Skin Q Ratio	Float	0.500000

   - `Skin hue min`
   - `Skin hue max`

   - `Skin y min`
   - `Skin y max`
   - `Skin saturation min y max`
   - `Skin saturation max y max`
   - `Skin saturation min y min`
   - `Skin saturation max y min`
   - `Skin boundary probability`
   - `Skin percent`
   - `Skin non skin to skin q ratio`

   b. Adjust facial area. Navigate to the **Facial Area Adjustment** box and adjust **Face Boundary** and **Face Transition**, as needed. Note that `face boundary` must be less than or equal to `face transition`.

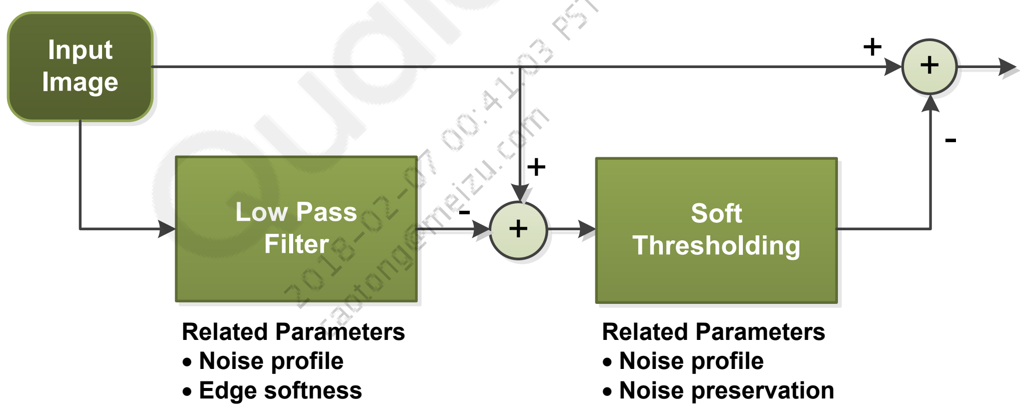


5. Repeat the procedures above for the remaining regions.
6. When complete, click **File** > **Save To Project** to preserve the tuned HNR settings.

7 IFE tuning

7.1 ABF tuning concepts

The Adaptive Bayer Filter (ABF) tuning module performs denoising for the Bayer domain. Regular 2D convolution filtering deteriorates image quality, particularly when applied on Bayer pixels. ABF can adapt its kernel based on neighboring pixel values to reduce noise without blurring sharp edges. ABF includes a low pass filter followed by soft thresholding.



ABF is not the primary module for denoising in Qualcomm Spectra 2xx. Noise tuning is primarily accomplished with the following modules:

- Advanced Noise Reduction (ANR) + Temporal Noise Filtering (TF) for preview/view mode
- Hybrid Noise Reduction (HNR) + ANR for snapshot mode

Coordination with other modules

In general, the factory defaults for ABF parameters should be adequate because ABF is not the primary module for denoising. Use a mild setting for the ABF bilateral filtering strength to preserve more detail and texture. Allow the ANR, HNR, and TF modules to control the noise.

ABF processes pixels in a linear domain while other modules are in a non-linear YUV domain. Therefore, it is good to have a certain level of noise reduction in the linear domain, especially for getting high quality snapshots.

ABF coordinates with other tuning modules in the following ways:

- The kernel size of ABF is similar to the kernel size of GIC, HNR and the first pass of ANR.
  - For snapshot, the balance is among GIC, HNR, and ABF
  - For video, the balance is among GIC, ABF, and the first pass of ANR
- If using a special RCCB or RGBW sensor, ABF can help reduce noise in the early part of the pipeline
- When doing initial spatial tuning, rely on edge alignment in ASF and disable directional smoothing available in ABF
- When doing initial spatial tuning, rely on HNR and disable activity-based strength adjustment available in ABF
- During normal tuning, turn off full BPC and rely on these other blocks that have the same function:
  - BPC/BCC/PDPC (single BPC)
  - Peak noise reduction (PNR) in GIC

Key ABF parameters for preview/video mode

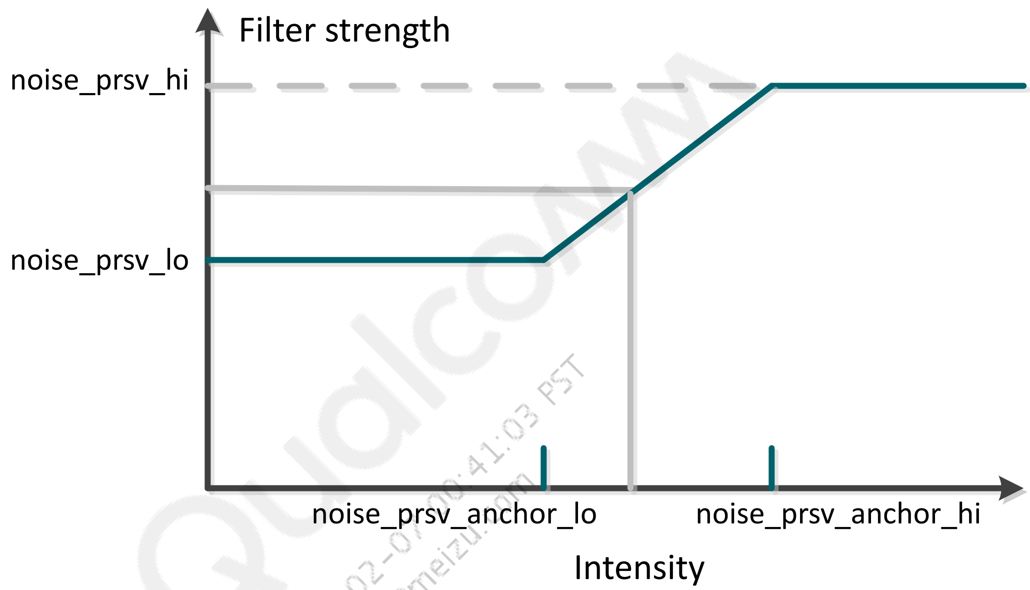
A noise profile is a 65-entry LUT that represents noise statistics for a specific sensor. The first to last entries correspond to a noise value (in terms of standard deviation) at intensity levels 0-255. The local average intensity is used to index the noise profile entry. A larger value for the `Noise Stdlut Level` equals a stronger denoising. Calibrate a noise profile for each sensor using the Chromatix tool. Usually the computed noise profile does not require any manual adjustment.

The `Edge Softness` parameter, together with the noise profile, controls the strength of the bilateral filter. The greater the edge softness value, the more denoising is done. Less noise results in flatter uniform regions, less detail, and softened edges.

Noise preservation determines how much noise to retain in the soft thresholding step. These settings are helpful for removing dark artifacts, especially crosshatch patterns, but dark details are impacted. There are four parameters that affect noise preservation: `Noise Prsv Lo` and `Noise Prsv Hi` set the minimum and maximum for how much noise to preserve; `Noise Prsv Anchor Lo` and `Noise Prsv Anchor Hi` set the minimum and maximum thresholds for local intensity.

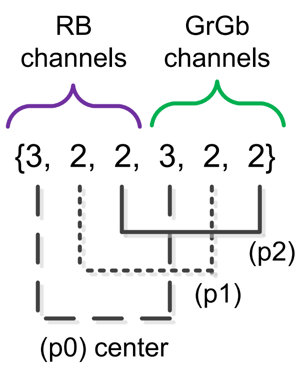
Noise preservation is determined by local intensity as follows:

- If local intensity ≤ Noise Prsv Anchor Lo, noise preservation = Noise Prsv Lo
- If local intensity ≥ Noise Prsv Anchor Hi, noise preservation = Noise Prsv Hi
- Otherwise, linear interpolation is used to determine the noise preservation value as shown in the following graph



Bilateral filter coefficients are adjusted based on pixel location. Pixels are grouped into three categories: the center pixel (p0), near neighbors (p1), and far neighbors (p2). The coefficient for each pixel is left shifted by (n-1) bits, where n=`Distance Kernel`.

The format of the distance kernel array is illustrated below:



Typically the distance kernel array does not need to be tuned, it is recommended to use the default value {3,2,2,3,2,2}. Adjust edge softness and other parameters instead.

The `Curve Offset` parameter causes more filtering of similar pixels resulting in smoother images. The four entries in the curve offset table correspond to the RGGB channels. A larger value smooths out details and noise, producing very soft images. Unless the sensor is extremely noisy, use the default values.

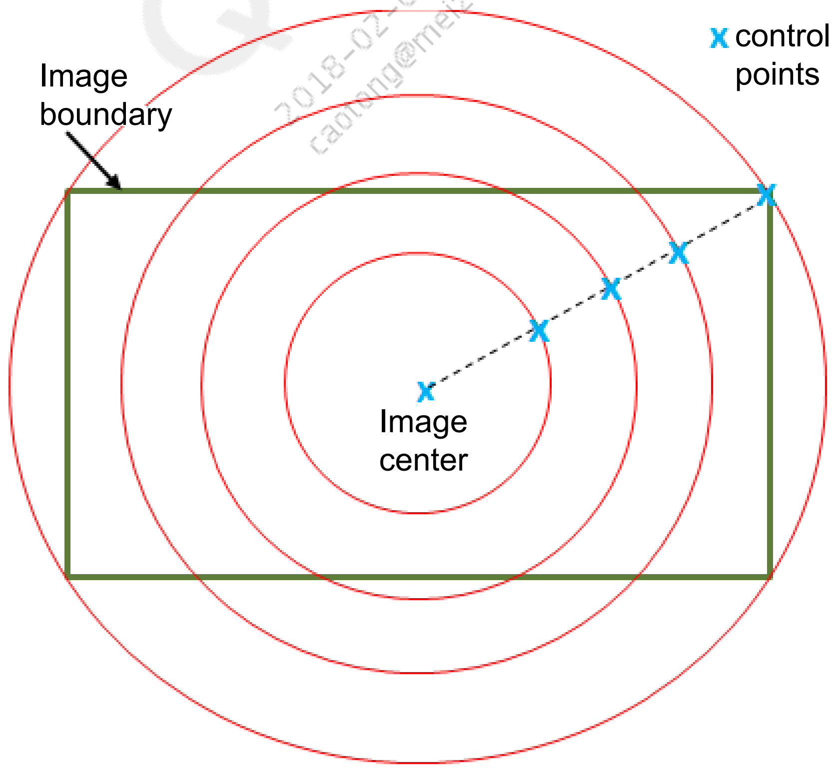
A key step of the bilateral filter is an update based on the weighted average of block matching vs pixel matching. Pixel matching tends to give results with more noise and detail. However, pixel matching can produce a more uniform noise pattern than block matching for some sensors under very low light conditions. Set `Blk Pix Matching RB` and `Blk Pix Matching G` to adjust RB and GrGb channels, respectively.

Most of the ABF parameters are *regional*, which means they must be tuned separately for different lighting conditions. An ABF region is defined by three specific nested triggers: DRC Gain Triggers, HDR AEC Triggers, and AEC Triggers. Define as many regions as you need to accomplish the preferred tuning specificity. It is important to define the regions for ABF before you begin tuning. Refer to *Add and edit triggers in the parameter editor* for examples and instructions.

RNR parameters

Noise in an image field of view (FOV) is not uniformly distributed. The corner of an image often has more noise due to rolloff correction. Radial noise reduction (RNR) is used to uniformly distribute noise in the FOV. ABF is early in the pipeline, so LSC information later in the pipeline can be used to adjust denoising.

RNR processing partitions an image into four regions by five control points as illustrated in the following diagram.



The `Radial Anchor Table`, in the ABF Reserve parameters, specifies the five radial control points that are defined by the radius percentage from the center point. The first and fifth points are hard coded as the image center (0.0) and corner (1.0). Use the Chromatix tool to tune the second point and the tool generates the remaining points.

Edge softness and noise preservation can be adjusted based on radial distance. Use the `Radial Base Table` to define radial based adjustment. Radial gain entries [0] - [4] adjust edge softness at each control point and entries [5] - [9] adjust noise preservation at each control point.

- At each control point [i] edgeSoftness'[i] = edgeSoftness/((base_table[i]+1)/256). i=0-4
- Noise preservation is first determined by local intensity, and then adjusted with the radius. Noise preservation'[i] = noise preservation * (base_table[i]+1)/256. i=5-9

A smaller radial gain results in a larger edge softness or smaller noise preservation, which results in more denoising. To use a stronger ABF at the image corners, program decreasing values into the base table.

Refer to the *ABF Parameters* section for more detail about how to adjust the parameter values.

RELATED INFORMATION

"ABF Parameters" on page 77
"Tune ABF" on page 86

7.1.1 ABF Parameters

The following tables lists the parameters used for tuning the ABF module for Preview/Video mode. The default values for the parameters are suitable for most situations but may require adjustment in some use cases to achieve your specific objectives.

Table 7-1 Basic ABF parameters (Preview/Video)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
ABF34 Enable (Enable Section)	Enable flag for ABF34.	Set to 1 to use the ABF module.	1	Take default value
SBPC Enable (Enable Section)	Enable flag to turn on single BPC and turn off full BPC. Useful for low power mode.	For low power mode, set to 1. Otherwise, use default of 0.	0	Take default value unless low power mode.
AEC Exposure Control (Control Method section)	Set exposure control triggers to be based on Gain or Lux Index.	Select the appropriate trigger type.	Gain	Yes, required
Cross Plane Enable (Chromatix ABF34 Reserve)	Enable flag for cross channel processing for GrGb channels. Affects pixel neighborhood definition.	Always set to 1 to help correct GrGb mismatch.	1	Take default value
Noise Stdlut Level Table (ABF34 Rgn Data)	A table that plots the noise intensity (in terms of standard deviation) at 65 points on the image. Known as the Noise Profile. Affects both the bilateral filter and soft thresholding steps of ABF. Same profile is used for all channels.	The table is computed by the Chromatix tool. Generally, the noise profile does not need tuning. Compute for each sensor. A larger noise value = stronger denoising.	Computed by the Chromatix tool based upon the uploaded image. Intensity value range = [0, 256]	Take computed values
Edge Softness (ABF34 Rgn Data)	Works with the noise profile to control bilateral filter strength. One value controls all four channels.	Focus on image quality at center. A larger value = more denoising (e.g., flatter uniform regions, less detail, softened edges) (see ABF examples)	3.000000 Range = [0, 15.996]	Yes

Table 7-1 Basic ABF parameters (Preview/Video) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Noise Prsv Lo Noise Prsv Hi (ABF34 Rgn Data)	▪ Minimum and maximum values for how much noise to retain. ▪ Helpful for removing dark artifacts, especially crosshatch patterns, but dark details are impacted. ▪ Noise Prsv Lo[0] and Noise Prsv Hi[0] adjust RB channels ▪ Noise Prsv Lo[1] and Noise Prsv Hi[1] adjust GrGb channels	▪ A larger value = less denoising ▪ Set Noise Prsv Lo smaller than Noise Prsv Hi so dark pixels have stronger denoising	{0,0} Range = [0.0, 1.0]	Yes
Noise Prsv Anchor Lo Noise Prsv Anchor Hi (ABF34 Rgn Data)	Minimum and maximum thresholds for local intensity.	▪ If intensity ≤ Noise Prsv Anchor Lo, noise preservation = Noise Prsv Lo ▪ If intensity ≥ Noise Prsv Anchor Hi, noise preservation = Noise Prsv Hi The interpolation range corresponds to intensity 40-56 for 8u values.	{160,224} Range = [0, 1023]	Yes
Blk Pix Matching Rb Blk Pix Matching G (ABF34 Rgn Data)	▪ Control the weighted average of block matching vs pixel matching in the bilateral filter update. ▪ Set blk_pix_matching_rb to adjust RB channels ▪ Set blk_pix_matching_g to adjust G channels	▪ A larger value = more pixel matching and less block matching. ▪ Pixel matching tends to give results with more noise and detail. ▪ 0=pure block matching, 4=pure pixel matching, values 1-3 are a combination of block and pixel matching (see ABF examples)	{0,0} Range = [1, 4]	Yes
Distance Ker (ABF34 Rgn Data)	▪ A bit-shifting value corresponding to distance weight. ▪ Bilateral filter coefficients are adjusted based on pixel	▪ Recommend using the default values. Adjust Edge Softness instead. ▪ A larger value for center pixel = less denoising.	{3,2,2,3,2,2} Range = [1, 4]	Take default values

Table 7-1 Basic ABF parameters (Preview/Video) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
	location (p0-center pixel, p1-adjacent pixel, p2-diagonal pixel) ■ Coefficients of each group (p0, p1, p2) are left shifted by (n-1) bits, (n = Distance Ker value) ■ Distance Ker table entries [0] - [2] control the RB channels and entries [3] - [5] control the GrGb channels	■ A larger value for groups p1/p2 = more denoising. ■ Non-zero value n means left shift (n-1). (see ABF examples)		
Curve Offset (ABF34 Rgn Data)	■ An offset that causes more filtering of similar pixels resulting in smoother images. ■ The four Curve Offset entries in the table correspond to the RGGB channels	■ Recommend using the default values unless the sensor is extremely noisy. ■ A larger value = more denoising. ■ Larger value smooths out details and noise, producing very soft images. (see ABF examples)	{0,0,0,0} Range = [1, 127]	Take default values
Radial anchor table (ABF34 Reserve)	Table that specifies five radial control points that partition the image into 4 regions. Each control point is defined by the radius percentage from the center point.	■ The first and fifth entries are hard coded as the image center (0.0) and corner (1.0). ■ Use the Chromatix tool to tune the 2nd point and the tool generates the 3rd and 4th points.	{0,0.4,0.6,0.8} Range = [0, 1]	Yes, using Sigma LUT RNR dialog > Radial Control Points tab
Radial gain table (ABF34 Rgn Data)	■ A table of gain adjustment entries that correspond to each of the five RNR control points. ■ Radial gain entries [0] - [4] adjust edge softness ■ Radial gain entries [5] - [9] adjust the noise preservation	■ Default is no adjustment at control points ■ Smaller Radial Gain results in larger Edge Softness or smaller Noise Preservation, which results in more denoising. ■ Radial gain entries [0] and [5] are hardcoded at 255 so no	{255, 255, 255, 255, 255, 255, 255, 255, 255, 255} Range = [0.0, 1.0]	Yes

Table 7-1 Basic ABF parameters (Preview/Video) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
		adjustment can be made to the center point.		
BPC FMax (ABF34 Rgn Data)	Upper offset factor used in single BPC applied to center pixel.		16 Range = [0, 63]	Yes, if low power mode
BPC FMin (ABF34 Rgn Data)	Lower offset factor used in single BPC applied to center pixel.		16 Range = [0, 63]	Yes, if low power mode
BPC MaxShft (ABF34 Rgn Data)	Upper offset (as right-shifting) used in single BPC applied to neighbor pixels.		4 Range = [0, 16]	Yes, if low power mode
BPC MinShft (ABF34 Rgn Data)	Lower offset (as right-shifting) used in single BPC applied to neighbor pixels.		4 Range = [0, 16]	Yes, if low power mode
BPC Offset (ABF34 Rgn Data)	Common absolute offset value used in single BPC applied to both center pixel and neighboring pixel.		64 Range = [0, 4095]	Yes, if low power mode

RELATED INFORMATION

"Examples of ABF parameter effects" on page 81

7.1.2 Examples of ABF parameter effects

The following examples illustrate the effects of adjusting specific ABF parameter values for a preview or video project.

Effects of the edge softness parameter

(L) edge_softness=3, (C) edge_softness=8, (R) edge_softness=12. A larger value results in softened edges, less noise, and less detail.

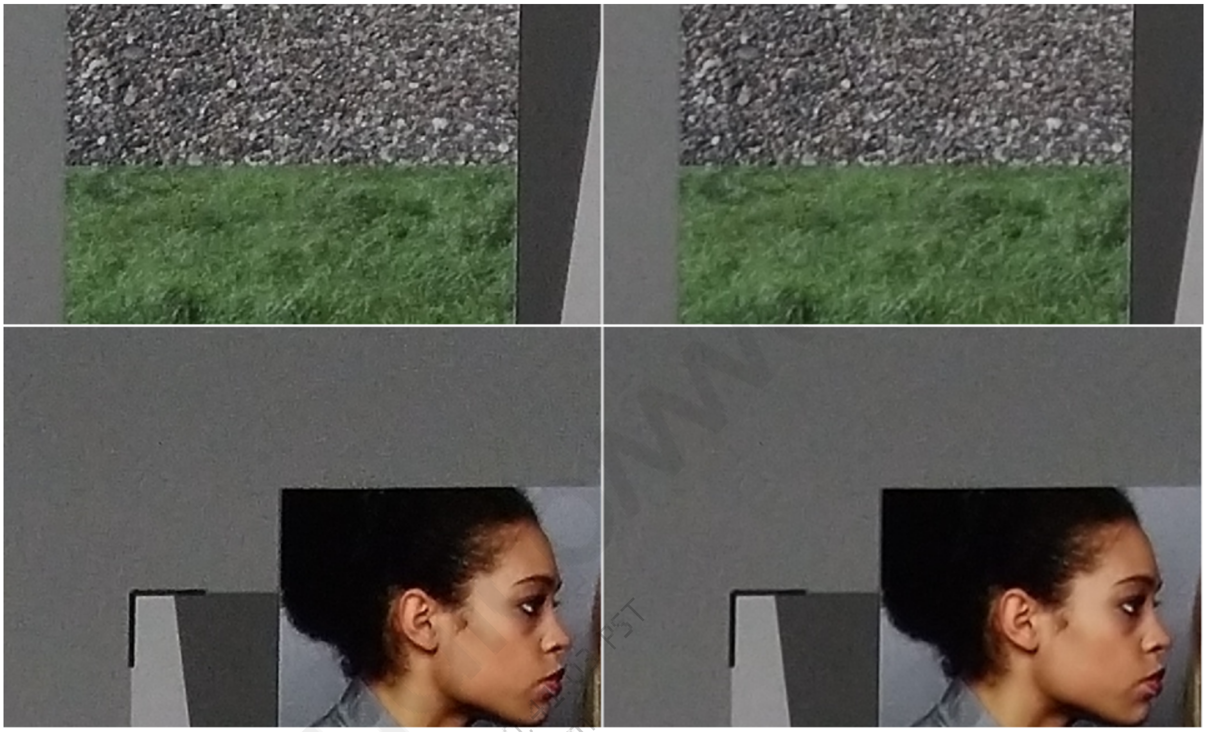
Effects of the distance kernel parameters

(L) distance_ker={{3,2,1} {3.2.1}}, (R) distance_ker={{4,2,1} {4.2.1}}. Right column images are noisier, with many artifacts. Left column images using default values give a good balance of noise/detail in most cases.

Effects of the curve offset parameters

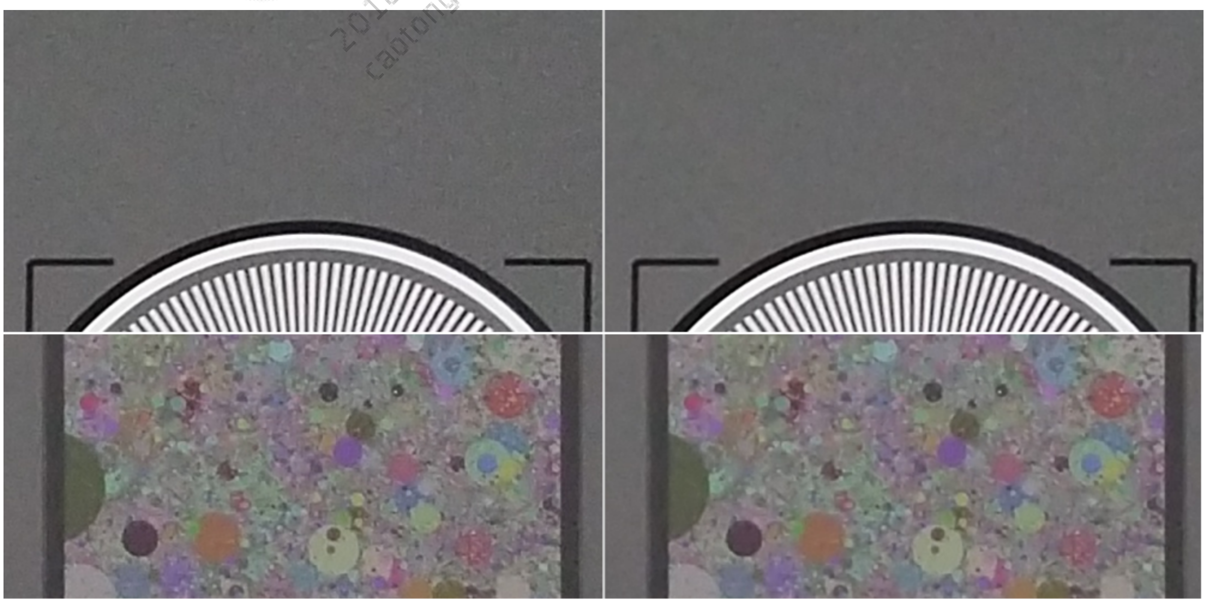
(L) curve_offset={20, 20, 20, 20}, (R) curve_offset={0,0,0,0}. Higher curve_offset smooths out the details.

Effects of the block/pixel matching parameters



(L) Pure pixel matching, (R) Pure block matching

Effects of noise preservation strength parameters



(L) noise_prsv_lo = {0.2, 0.2}, noise_prsv_hi = {0.2, 0.2}, (R) noise_prsv_lo = {0.0, 0.0}, noise_prsv_hi = {0.2, 0.2}. The setting for the right-hand images results in artifact reduction and some loss of detail.

Effects of RNR radial gain table

(L) RNR off in ABF. Radial_gain_table = {255, 255, 255, 255, 255} {255, 255, 255, 255, 255}, (R) Radial_gain_table = {255, 255, 114, 67, 43} {255, 255, 114, 67, 43}

- Effective corner noise reduction (top row of plots cropped at image corner)
- No impact on details at image center (bottom row of plots cropped at image center)
- With the default control points, the center 50% FOV has no RNR adjustment

(L) Radial_gain_table = {255, 255, 213, 117, 142} {255, 255, 213, 117, 142}, (R) Radial_gain_table = {255, 255, 183, 135, 98} {255, 255, 225, 225, 225}

- All images are cropped at image corner.
- The left-hand setting has RNR adjustment for both edge softness and noise preservation. The right-hand setting has adjustment only for edge softness. Note the results are similar.
- In most cases, the preferred RNR adjustment is achieved with just the edge softness adjustment.

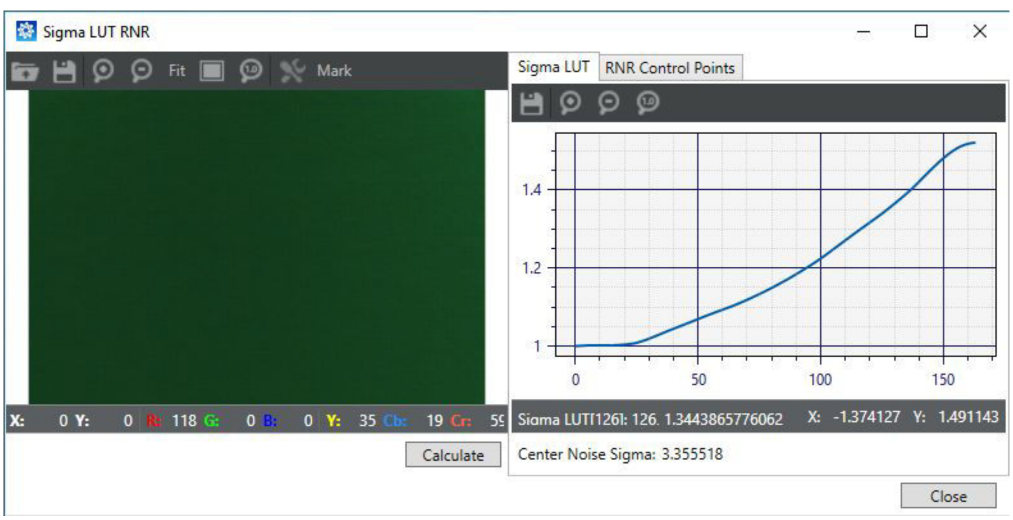
7.1.3 Tune ABF

Prerequisites:

Before tuning the ABF module, determine how many lighting regions to define and create them in the Parameter Editor.

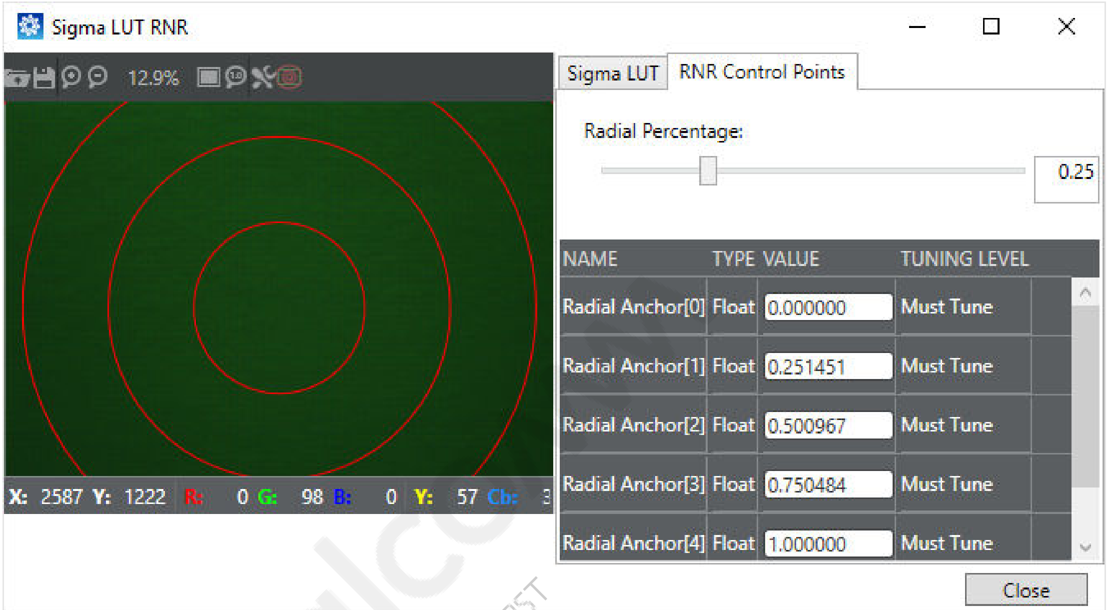
Regular 2D convolution filtering deteriorates image quality, particularly when applied on Bayer pixels. To reduce the noise without blurring sharp edges, use an adaptive Bayer filter (ABF), which can adapt its kernel based on neighboring pixel values.

1. On the **Pipeline** tab, click **IFE** > **ABF**.
2. Calibrate the noise profile as follows:
   a. Click a region and click the **Load Image** button.
   b. Select a raw image of an MCC chart taken in the appropriate lighting condition for that region.
   c. Set the MCC markers. Drag the top-left MCC grid marker to the center of the top-left color patch and drag the bottom-right MCC grid marker to the center of the bottom right color patch, to align all of the grid markers within the color patches. **Note**: If the image is taken with a fisheye lens, to set the MCC markers for tuning, hold the Ctrl Key (or Shift Key) while moving the markers.
   d. Click **Calculate** to compute the noise profile. View the resulting noise profile graph on the **Noise Profile** tab.
3. Calibrate the radial noise reduction. Click the **Sigma LUT RNR** button and do the following steps:
   a. Click the **Sigma LUT** tab and open a gray chart raw image for the appropriate lighting region.



   b. Click **Calculate** to generate the Sigma LUT plot.

c. Click the **RNR Control Points** tab, and adjust the **Radial Percentage** slider to set the second control point. The tool computes the other points automatically.



d. Click **Close**.

4. (Optional) ABF processing is enabled by default. To disable ABF processing, open the Parameter Editor, navigate to **ABF 34 IFE** > **Enable Section** > **ABF Enable**, and change the value from 1 to 0.

5. (Optional) Single BPC is disabled by default (and full BPC is on). To turn off full BPC, open the Parameter Editor, navigate to **ABF 34 IFE** > **Enable Section** > **Sbpc Enable**, and change the value from 0 to 1.

6. (Optional) Gr/Gb cross-channel processing is enabled by default. To disable cross-channel processing, open the Parameter Editor, navigate to **ABF 34 IFE** > **ABF 34 Reserve** > **Cross Plane En**, and change the value from 1 to 0.

7. As needed, click the **Region Data** tab and tune the following ABF parameters:

Parameter	Description
Edge Softness	Works with the noise profile to control bilateral filter strength. One value controls all four channels.
Noise Prsv Lo and Noise Prsv Hi	Minimum and maximum values for how much noise to retain. Helps to remove dark artifacts.
Noise Prsv Anchor Lo and Noise Prsv Anchor Hi	Controls minimum and maximum thresholds for local intensity.
Blk Pix Matching Rb and Blk Pix Matching G	Controls the weighted average of block matching vs. pixel matching.
Distance Kernel	Entries [0] - [2] control distance weight for the RB channels and entries [3] - [5] control distance weight for the GrGb channels
Curve Offset	Smooths out details and noise for each channel.
Radial Gain parameters	Entries [0] - [4] adjust edge softness and entries [5] - [9] adjust noise preservation.

8. Repeat all steps for each lighting region you set up.

9. (Optional) To see the results of the tuning adjustments, click **Simulate**. Otherwise, simulate after the entire pipeline is tuned.
10. To save the changes, click **File** > **Save To Project**.

RELATED INFORMATION

"Add and edit region triggers" on page 15
"ABF Parameters" on page 77

7.2 Linearization tuning concepts

Because of image photosensor characteristics, sensor output response to scene lightness may not necessarily be linear and may be different for various lighting conditions. For example, the sensor photonic response curve would be nearly flattened on bright light saturation. For dark environments, sensor output would not be totally zero due to dark current.

The black level offset value depends on the specific sensor being used and is also a function of the integration time, overall gain settings corresponding to the exposure settings, and temperature. The black level of the sensor response must be corrected for each channel on every pixel for different lighting conditions.

The black level offset uses a 2- or 4-channel black level correction. The black level offset values are then subtracted from the pixel output to get the corrected value.

7.2.1 Tune linearization

Linearization tuning adjusts the level of brightness in the darkest part of the image. Channel values from three images are averaged to calculate the black level offset values. These offset values are then subtracted from the pixel output to obtain the corrected value.

1. On the **Pipeline** tab, click **Linearization**, and then click a region in the Regions group.
2. (Optional) Deselect the **Bls Enable** check box you would like to use default black level values for black level subtraction (BLS).
3. Calculate linearization values:
   a. Click a region in the Regions group.
   b. Click **Load Image**, select the applicable black level reference image, and click **Open**.
      The image values are listed in the table fields with the average values listed underneath.
   c. Click **Recalculate** to recalculate linearization values for the R/G and G/ B values.
   d. Click **OK**.
4. Repeat process for each region.
5. Click **File** > **Save to Project** to save any changes made.

7.3 PDPC/DSBPC tuning concepts

The PDPC module includes processing for both phase detection pixel correction (PDPC) and dynamic single bad pixel correction (DSBPC). PD pixel correction and DSBPC may be enabled separately in the PDPC module, if appropriate. Use the table below to determine the pixel correction processes to tune for your project.

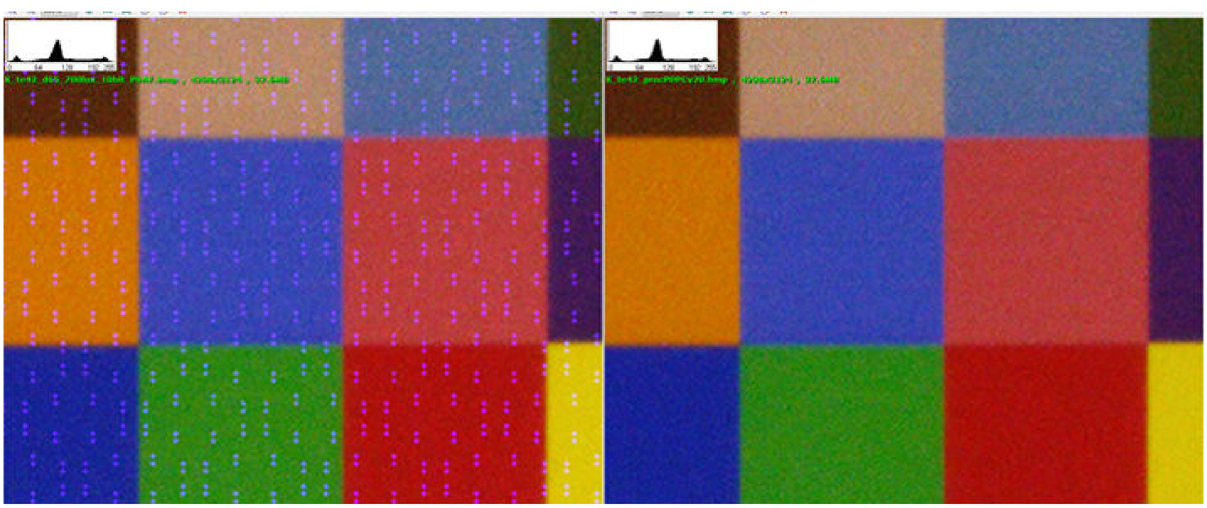
Table 7-2 IFE pixel correction processes matrix

		PD pixel sensor	Non-PD pixel sensor
No zzHDR	Non PD pixel	Not applicable	Not applicable
	PD pixel	PDPC	Not applicable
zzHDR	zzHDR T1 (long exposure)	PDPC/DSBPC	DSBPC
	zzHDR T2 (short exposure)	DSBPC	DSBPC

PD pixel correction concepts

A CMOS sensor can have phase pixels to speed up autofocus. One difference between phase pixels and the normal pixels in a sensor is that the location of the phase pixels is known. The PDPC module takes advantage of this location information to correct defective PD pixels that occur in those locations. Camera software detects phase pixel information in the sensor or driver information file and passes it to the simulator. Currently, the PDPC module only corrects bad PD pixels if they are on a red or blue pixel. Green pixels are not corrected at this time.

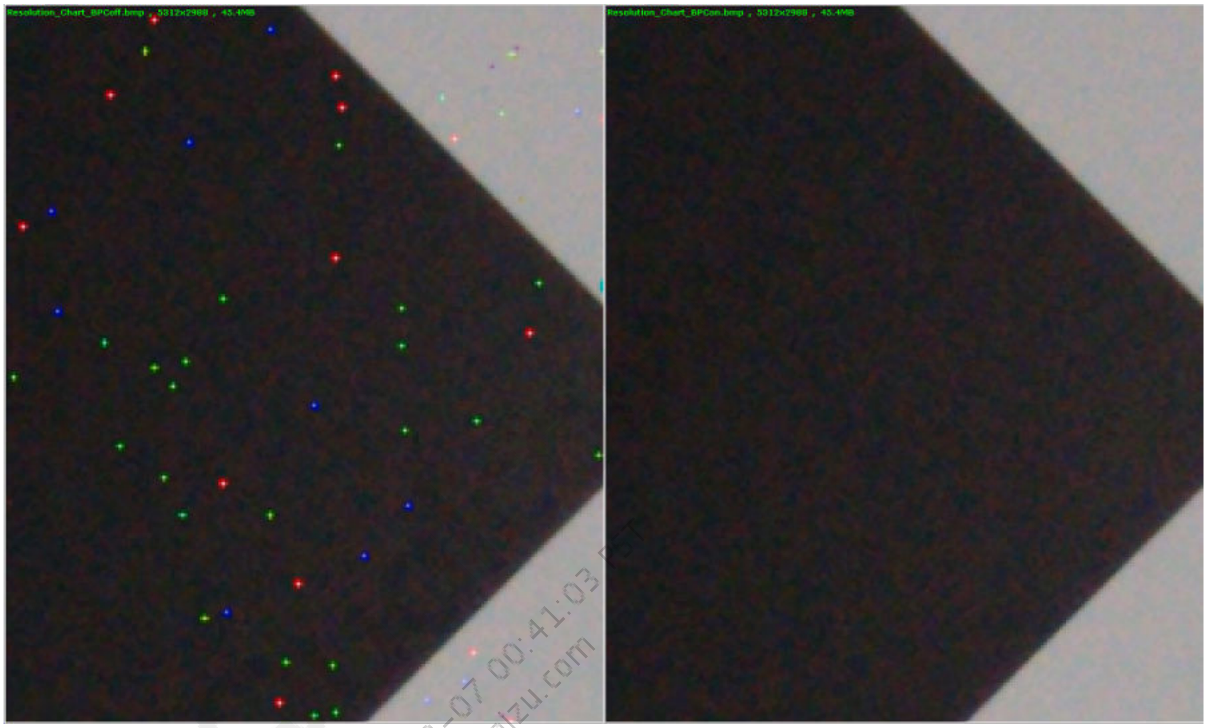
Because the PDPC module knows the location of a defective phase pixel, it has all the information it needs to make a correction based on the neighboring pixels. Thus, there is no parameter tuning required for PD pixel correction. The PD pixel correction supports sensors that use the non-zzHDR and zzHDR patterns.



(L) Original image, PDPC off, defective PD pixels on blue locations; (R) PDPC on, defective PD pixels corrected

DSBPC concepts

Use DSBPC processing to correct bad pixels for a non-PD pixel sensor project that uses the zzHDR pattern.

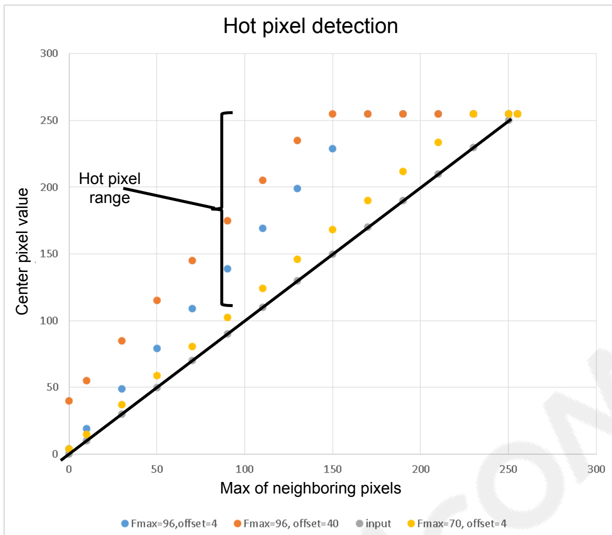


(L) Original image, DSBPC off; (R) DSBPC on, defective pixels corrected

To correct hot pixels, the module evaluates the value of the center pixel with the formula: (Fmax * MP) + BPC Offset

- MP is the maximum pixel value from among the neighboring pixels of the same channel
- Fmax is a user-defined threshold factor that specifies which pixels to correct
- BPC offset is a user-defined adjustment value that further refines the threshold for a bad pixel
- If the value of the center pixel is greater than the result, replace the center pixel with the value of MP. If the center pixel value is less than or equal to the result, do not change the center pixel value.

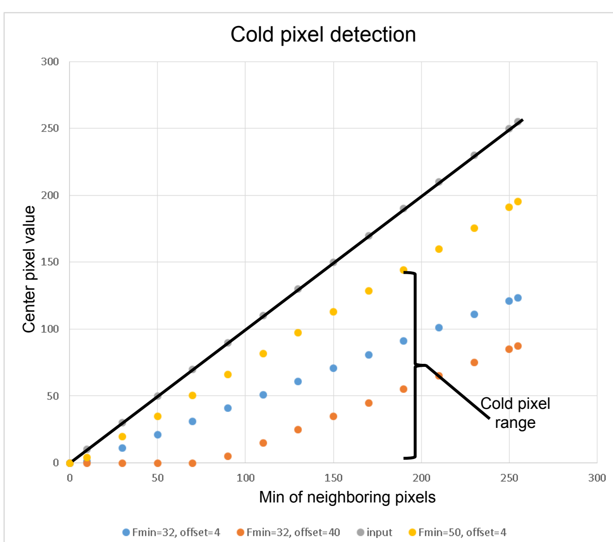
The following figure is a sample plot of the hot pixel equation.



Similarly, to correct cold pixels, the module evaluates the value of the center pixel with the formula: (Fmax * mP) + BPC Offset

- mP is the minimum pixel value from among the neighboring pixels of the same channel
- Fmin is a user-defined threshold factor that specifies which pixels to correct
- BPC offset is a user-defined adjustment value that further refines the threshold for a bad pixel
- If the value of the center pixel is less than the result, replace the center pixel with the value of mP. If the center pixel value is greater than or equal to the result, do not change the center pixel value.

The following figure is a sample plot of the cold pixel equation.



Coordinating PDPC with other IFE modules

Keep the following considerations in mind when tuning IFE modules:

- If the sensor used for the project is a PD sensor, PDPC must be enabled.
- PDPC does not rely on other modules to be tuned, but if the PDPC module is not properly tuned, all modules that follow it in the pipeline are impacted
- For PDPC tuning, only enable the basic modules that are needed to result in final YUV images

RELATED INFORMATION

“PDPC/DSBPC parameters” on page 93
“Tune PDPC/DSBPC” on page 95

7.3.1 PDPC/DSBPC parameters

The following table lists the parameters used for tuning the PDPC module for Preview/Video mode. The default values for the parameters are suitable for most situations but may require adjustment in some use cases to achieve your specific objectives.

Table 7-3 PDPC/DSBPC parameters (Preview/Video)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Pdpc Enable (Enable Section)	Enables or disables PD pixel correction.	Set to 1 if the project uses a PD sensor and you want to correct bad PD pixels.	0 Enable = 1 Disable = 0	Yes, if applicable
Dspbc Enable (Enable Section)	Enables or disables single bad pixel correction for zzHDR when using a non-PD pixel sensor.	Set to 1 if the project uses a non-PD sensor with a zzHDR pattern and you want to correct bad pixels.	0 Enable = 1 Disable = 0	Yes, if applicable
Fmax Pixel Q 6 (Pdpc 11 Rgn)	Specifies the threshold for a hot pixel value to be selected for correction. Used only for DSBPC processing of an zzHDR pattern.	If the project uses a non-PD sensor with a zzHDR pattern, set this value to control how many hot pixels are corrected. A higher value results in fewer pixels detected as bad hot pixels. For a high light region, lowering Fmax has more impact on missed pixels than lowering the offset. For a low light region, lowering the offset has more impact on missed pixels than lowering Fmax.	64 Range: [0, 255]	
Fmin Pixel Q 6 (Pdpc 11 Rgn)	Specifies the threshold for a cold pixel value to be selected for correction. Used only for DSBPC processing of an zzHDR pattern.	If the project uses a non-PD sensor with a zzHDR pattern, set this value to control how many cold pixels are corrected. A higher value results in fewer pixels detected as bad cold pixels.	64 Range: [0, 255]	

Table 7-3 PDPC/DSBPC parameters (Preview/Video) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Bp Offset G Pixel (Pdpc 11 Rgn)	Specifies the offset for a green pixel to be selected for correction. Used only for DSBPC processing of an zzHDR pattern (T1, long exposure).	Set this value to adjust how many bad green pixels are corrected. A higher value results in fewer green pixels detected as bad pixels.	16384 Range: [0, 32767]	Use default value, or tune as needed
Bp Offset Rb Pixel (Pdpc 11 Rgn)	Specifies the offset for a red or blue pixel to be selected for correction. Used only for DSBPC processing of an zzHDR pattern (T1, long exposure).	Set this value to adjust how many bad red or blue pixels are corrected. A higher value results in fewer red or blue pixels detected as bad pixels.	16384 Range: [0, 32767]	Use default value, or tune as needed
T2 Bp Offset G Pixel (Pdpc 11 Rgn)	Specifies the offset for a green pixel to be selected for correction. Used only for DSBPC processing of an zzHDR pattern (T2, short exposure).	Set this value to adjust how many bad green pixels are corrected. A higher value results in fewer green pixels detected as bad pixels.	16384 Range: [0, 32767]	Use default value
T2 Bp Offset Rb Pixel (Pdpc 11 Rgn)	Specifies the offset for a red or blue pixel to be selected for correction. Used only for DSBPC processing of an zzHDR pattern (T2, short exposure).	Set this value to adjust how many bad red or blue pixels are corrected. A higher value results in fewer red or blue pixels detected as bad pixels.	16384 Range: [0, 32767]	Use default value

RELATED INFORMATION

"Examples of DSBPC parameter effects" on page 95

7.3.2 Examples of DSBPC parameter effects

The following examples illustrate the effects of adjusting specific DSBPC parameter values for a preview/video project.

Effects of the Fmax Pixel Q 6 parameter

(L) Weak setting, Fmax Pixel Q 6=96, correct fewer bad pixels. Circles around bad pixels not included in the detection. (R) Aggressive setting, Fmax Pixel Q 6=65, correct more bad pixels

7.3.3 Tune PDPC/DSBPC

Prerequisites:

Before tuning the PDPC module, determine how many lighting regions to define for this module and create them in the Parameter Editor.

The PDPC module includes processing for both phase detection pixel correction (PDPC) and dynamic single bad pixel correction (DSBPC). Enable PDPC to correct PD pixels in a non-zzHDR or zzHDR pattern. Enable DSBPC to correct non-PD pixels in a zzHDR pattern.

1. On the **Pipeline** tab, click **IFE** > **PDPC**.
2. PDPC is disabled by default. To enable PD pixel correction, open the Parameter Editor, navigate to **Pdpc 11 IFE** > **Enable Section** > **Pdpc Enable**, and change the value from 0 to 1. This completes PDPC tuning for a PD pixel sensor project.
3. DSBPC is disabled by default. To enable dynamic single bad pixel correction, open the Parameter Editor, navigate to **Pdpc 11 IFE** > **Enable Section** > **Dsbpc Enable**, and change the value from 0 to 1.
4. If tuning for a non-PD pixel sensor, click the **Region Data** button and tune the following DSBPC parameters as needed:
   - Fmax Pixel Q 6
   - Fmin Pixel Q 6
   - Bp Offset G Pixel

- Bp Offset Rb Pixel
- T 2 Bp Offset G Pixel
- T 2 Bp Offset Rb Pixel

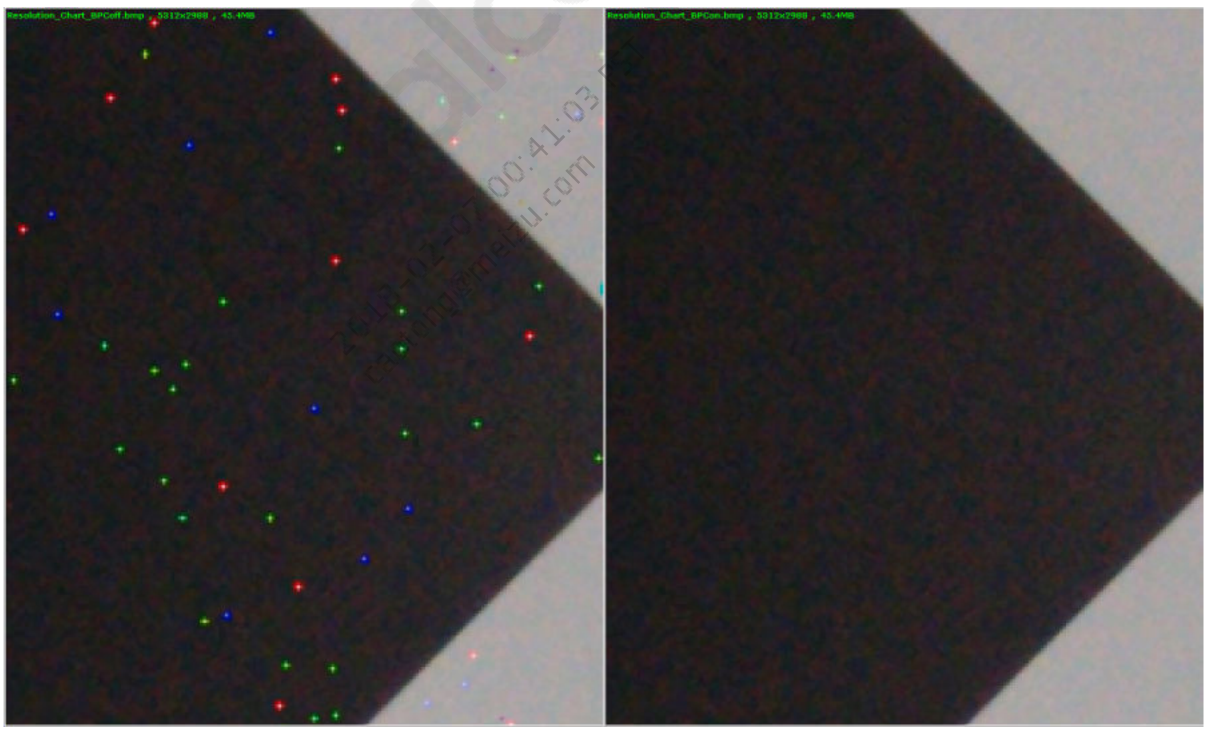
5. Repeat the previous step for each lighting region you set up.
6. Capture test images to see if any bad pixels are missed or if there is any resolution loss or false color. Make fine tuning adjustments to Fmax, Fmin, and offset parameters as needed.
7. To save the adjustments made to PDPC/DSBPC parameters, click **File** > **Save To Project**.

RELATED INFORMATION

“Add and edit region triggers” on page 15
“PDPC/DSBPC parameters” on page 93

7.4 BPC/BCC tuning concepts

Defective pixels may show up on an image as bright points (hot pixels) or as dark points (cold pixels).

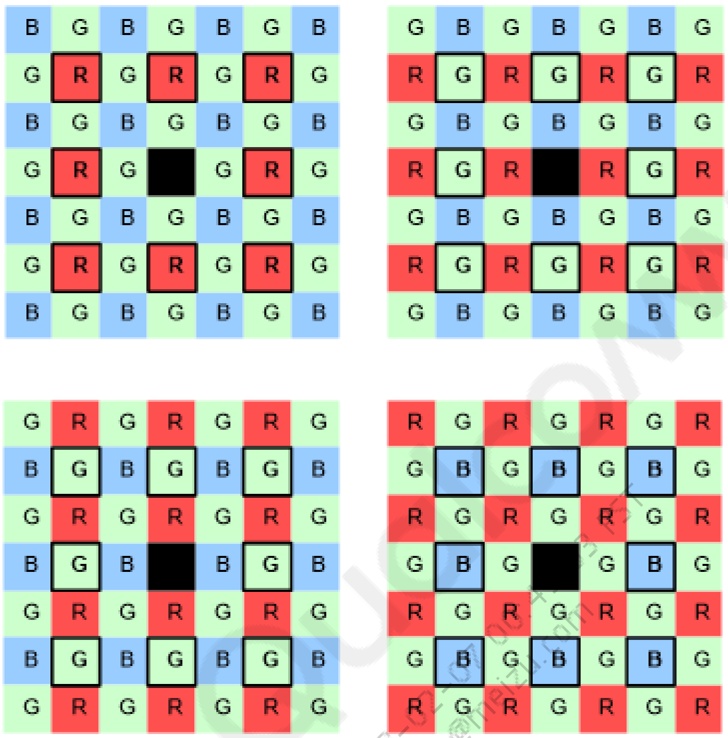


(L) Original image, BPC/BCC off; (R) BPC/BCC on, defective pixels corrected

Defective pixels can be caused by the following:

- Imperfections in the semiconductor processing of the sensor module (CMOS or CCD) – During manufacturing, bad pixels or clusters can be caused by leakage to or from the well, abnormal pixel sensitivity, or resistance losses.
- Dust introduced when lens is mounted – A dust particle can fall on top of the sensor when the lens is mounted.
- Age of the sensor module – Over time, a sensor module accumulates more bad pixels.

For a sensor that uses the Bayer pattern, if the value of a hot pixel value is significantly higher than its neighbors or if the value of a cold pixel value is significantly lower than its neighbors, the bad pixel correction (BPC) and bad couplet correction (BCC) module evaluates it for correction. The following figure shows a single bad pixel in the center (black square) with its eight neighboring pixels (outlined) on the same channel.

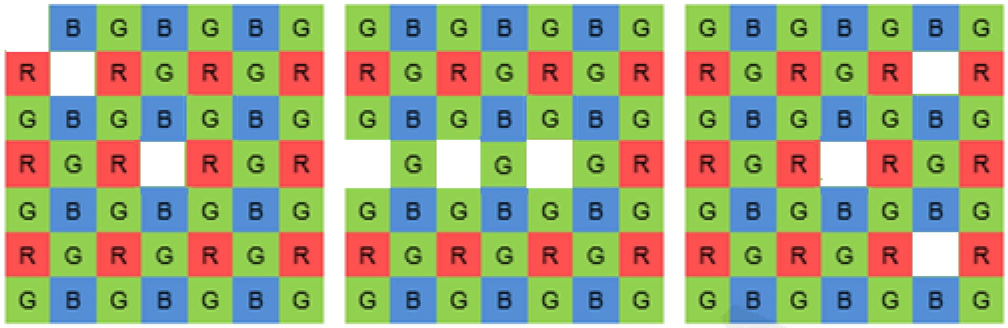


(Top L) Red channel neighbors; (Top R) Gr channel neighbors; (Bottom L) Gb channel neighbors; (Bottom R) B channel neighbors

Two neighboring defective pixels are called couplets. The BCC module can correct couplets that are in the same channel or cross-channel as shown in the image.

(L) Cross-channel couplet, Gb and Gr channels; (R) Same channel couplet, Gb channel

BPC/BCC does not correct clusters of three or more bad pixels, such as those shown in the figure.



Example of clusters of 3+ bad pixels not corrected with BPC/BCC

To correct hot pixels, the module evaluates the value of the center pixel with the formula: (Fmax * MP) + BPC Offset

- MP is the maximum pixel value from among the eight neighboring pixels of the same channel
- Fmax is a user-defined threshold factor that specifies which pixels to correct
- BPC offset is a user-defined adjustment value that further refines the threshold for a bad pixel
- If the value of the center pixel is greater than the result, replace the center pixel with the value of MP. If the center pixel value is less than or equal to the result, do not change the center pixel value.

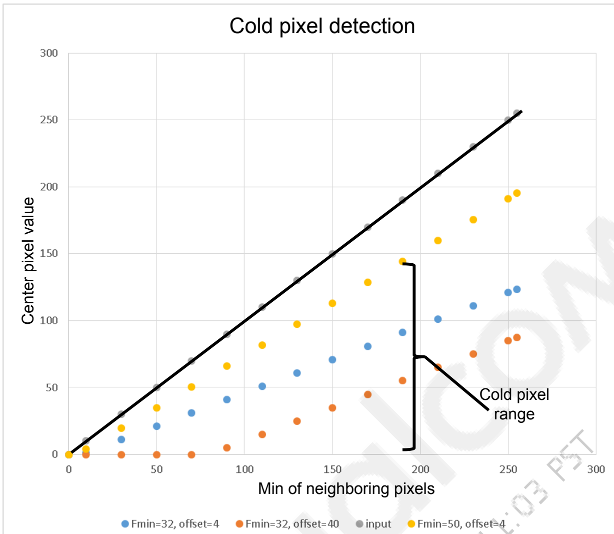
The following figure is a sample plot of the hot pixel equation.

Similarly, to correct cold pixels, the module evaluates the value of the center pixel with the formula: (Fmax * mP) + BPC Offset

- mP is the minimum pixel value from among the eight neighboring pixels of the same channel
- Fmin is a user-defined threshold factor that specifies which pixels to correct

- BPC offset is a user-defined adjustment value that further refines the threshold for a bad pixel
- If the value of the center pixel is less than the result, replace the center pixel with the value of mP. If the center pixel value is greater than or equal to the result, do not change the center pixel value.

The following figure is a sample plot of the cold pixel equation.



For similar single pixel correction on an zzHDR pattern, see the DSBPC submodule in the PDPC module.

Coordinating BPC/BCC with other IFE modules

Keep the following considerations in mind when tuning IFE modules:

- BPC/BCC does not rely on other modules to be tuned, but if the BPC/BCC module is not properly tuned, all modules that follow it in the pipeline are impacted
- For BPC/BCC tuning, only enable the basic modules that are needed to result in final YUV images
- Consider disabling BPC/BCC while tuning other modules to avoid any negative impact of untuned pixel correction parameters.

RELATED INFORMATION

“BPC/BCC parameters” on page 100
“Tune BPC/BCC” on page 102

7.4.1 BPC/BCC parameters

The following table lists the parameters used for tuning the BPC/BCC module for preview/video mode. The default values for the parameters are suitable for most situations but may require adjustment in some use cases to achieve your specific objectives.

Table 7-4 BPC/BCC parameters (preview/video)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Bpcbcc Enable (Enable Section)	Enables or disables bad pixel correction for a Bayer pattern	Set to 1 to correct bad pixels for a Bayer pattern	1 Enable = 1 Disable = 0	Yes, if applicable
Hot Pixel Correction Disable (Bpcbcc 50 Rgn)	Enables or disables correction of hot pixels.	Set to 0 to correct hot pixels	0 Enable = 1 Disable = 0	Yes, if applicable
Cold Pixel Correction Disable (Bpcbcc 50 Rgn)	Enables or disables correction of cold pixels.	Set to 0 to correct cold pixels	0 Enable = 1 Disable = 0	Yes, if applicable
Same Channel Recovery (Bpcbcc 50 Rgn)	Enables or disables correction using cross-channel pixels	Set to 0 to use both same channel and cross-channel	0 Enable = 1 Disable = 0	Yes, if applicable
Fmax (Bpcbcc 50 Rgn)	Specifies the threshold for a hot pixel value to be selected for correction.	If the project uses a non-PD sensor with a Bayer pattern, set this value to control how many hot pixels are corrected. A higher value results in fewer pixels detected as bad hot pixels. For a high light region, lowering Fmax has more impact on missed pixels than lowering the offset. For a low light region, lowering the offset has more impact on missed pixels than lowering Fmax.	64 Range: [0, 127]	Use default value, or tune as needed

Table 7-4 BPC/BCC parameters (preview/video) (cont.)

Parameter Name	Description	Tuning	Factory Default	Must Tune?
Fmin (Bpcbcc 50 Rgn)	Specifies the threshold for a cold pixel value to be selected for correction.	If the project uses a non-PD sensor with a Bayer pattern, set this value to control how many cold pixels are corrected. A higher value results in fewer pixels detected as bad cold pixels.	32 Range: [0, 127]	Use default value, or tune as needed
BPC Offset (Bpcbcc 50 Rgn)	Specifies the offset adjustment to further refine the threshold for a single bad pixel.	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	128 Range: [0, 16383]	Use default value, or tune as needed
BCC Offset (Bpcbcc 50 Rgn)	Specifies the offset adjustment to further refine the threshold for a couplet bad pixel.	Set this value to adjust how many bad pixels are corrected. A higher value results in fewer pixels detected as bad pixels.	128 Range: [0, 16383]	Use default value, or tune as needed
Correction Threshold (Bpcbcc 50 Rgn)	Specifies the minimum value change allowed to the bad pixel. If the value change is less than this threshold, the change is not made.	Set this value to adjust how many bad pixels are corrected. A larger value results in fewer pixels corrected.	128 Range: [0, 819]	Use default value

RELATED INFORMATION

"Examples of BPC/BCC parameter effects" on page 102

7.4.2 Examples of BPC/BCC parameter effects

The following examples illustrate the effects of adjusting specific DSBPC parameter values for a preview/video project.

Effects of the Fmax parameter

(L) Weak setting, Fmax=96, correct fewer bad pixels. Circles around bad pixels not included in the detection. (R) Aggressive setting, Fmax=65, correct more bad pixels

7.4.3 Tune BPC/BCC

Prerequisites:

Before tuning the BPC/BCC module, determine how many lighting regions to define for this module and create them in the Parameter Editor.

The BPC/BCC module is used to correct hot and cold pixels or couplets.

1. On the **Pipeline** tab, click **IFE** > **BPCBCC**.
2. (Optional) BPC/BCC processing is enabled by default. To disable bad pixel and couplet correction, open the Parameter Editor, navigate to **Bpcbcc 50 IFE** > **Enable Section** > **Bpdbcc Enable**, and change the value from 1 to 0.
3. As needed, click the applicable **Region** and tune the following ABF parameters:
   - Hot Pixel Correction Disable
   - Cold Pixel Correction Disable
   - Same Channel Recovery
   - Fmax
   - Fmin
   - BPC Offset

- BCC Offset
- Correction Threshold

4. Repeat the previous step for each region.
5. Capture test images to see if any bad pixels are missed or if there is any resolution loss or false color. Make fine tuning adjustments to the parameters as needed.
6. To save the adjustments made to BPC/BCC parameters, click **File** > **Save To Project**.

RELATED INFORMATION

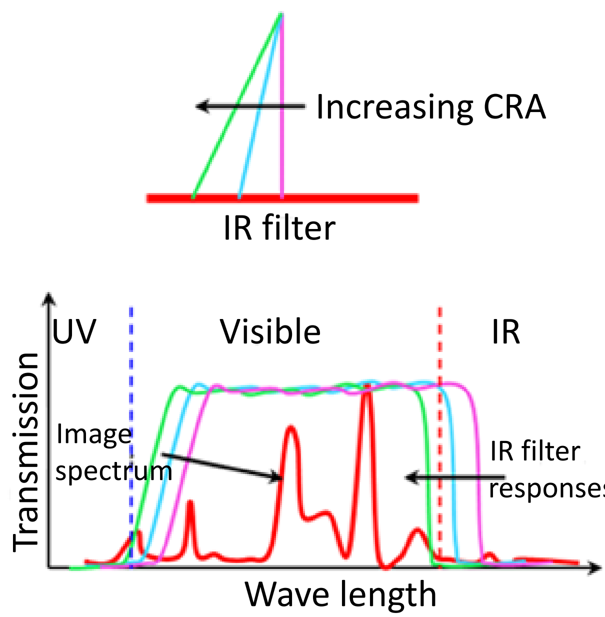
"BPC/BCC parameters" on page 100
"Add and edit region triggers" on page 15

7.5 LSC tuning concepts

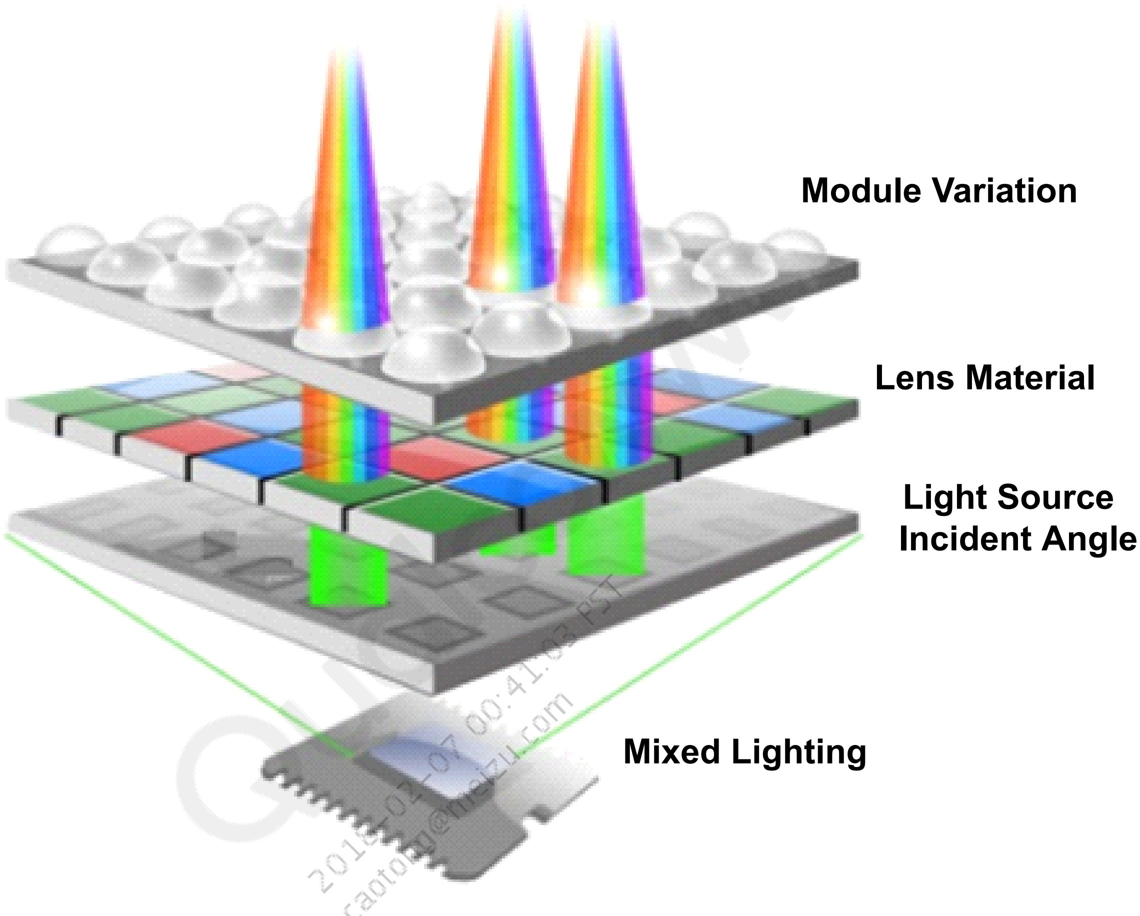
Based on the size and quality of the camera lens, the center of an image may appear brighter than the corner regions. Lens shading correction (LSC) corrects brightness attenuation and tint (uneven color shading).

(L) Original; (R) Corrected

Tint usually results from using an infrared (IR) filter when capturing images.

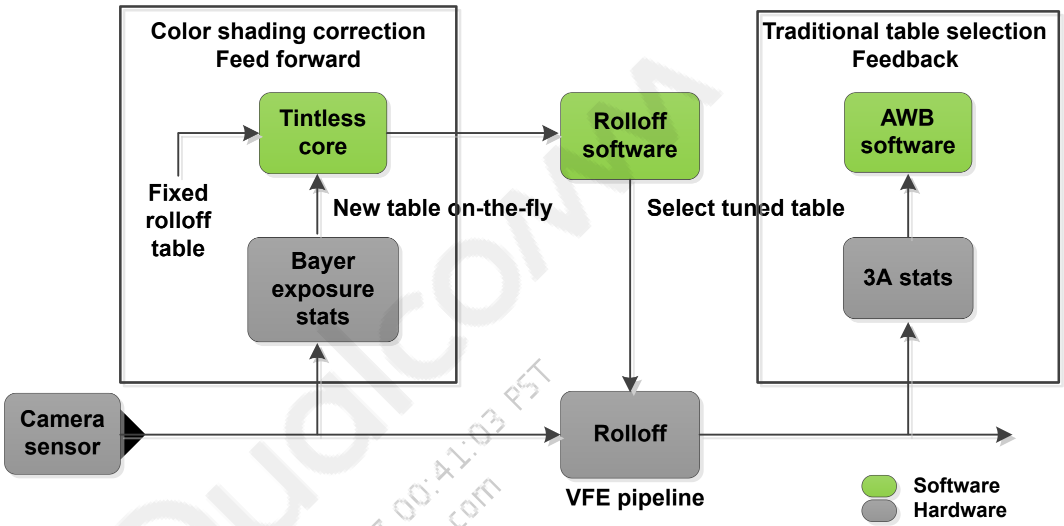


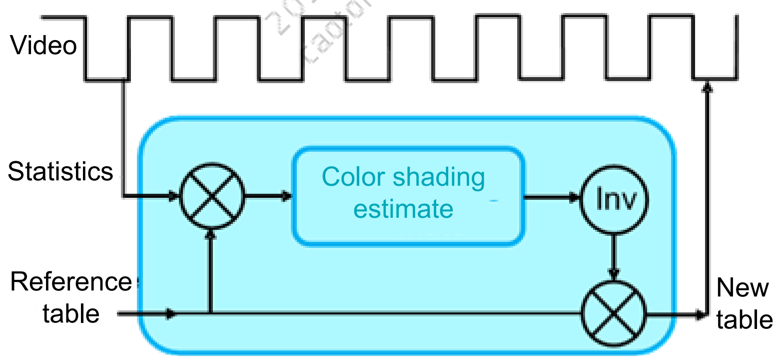
Sensor module variation, lens material and lighting conditions could also cause tint.

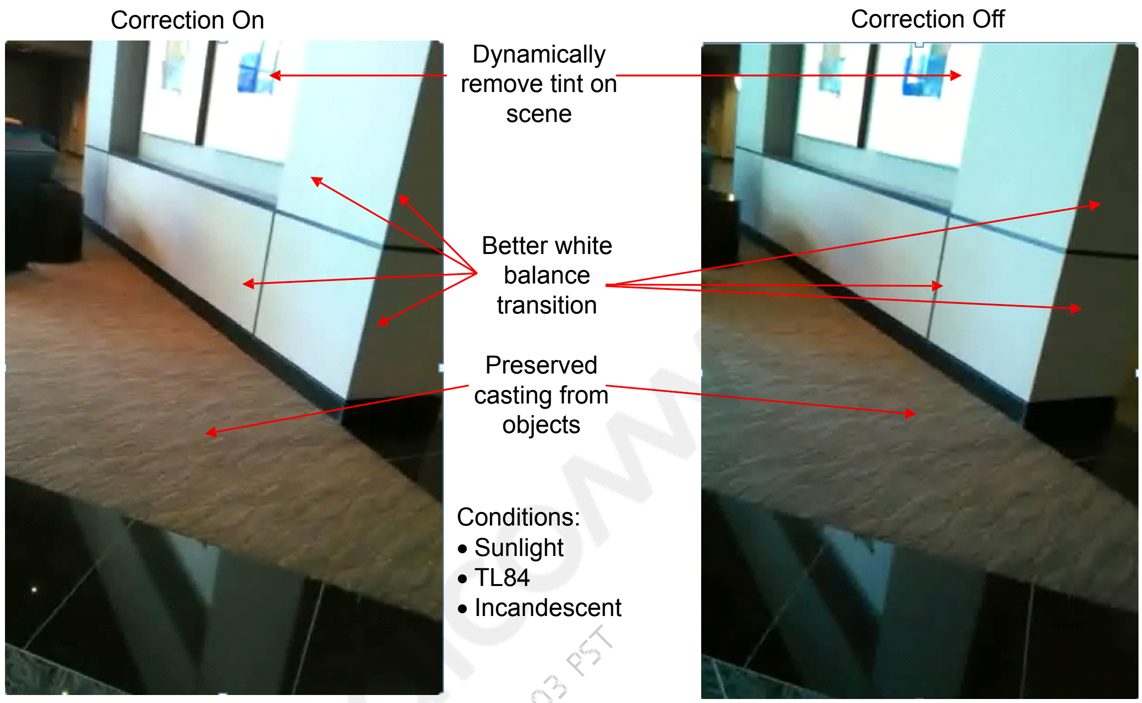


The following figure illustrates where color shading is corrected in the VFE. The main steps in the process are:

1. Values from the reference table are applied and residual stats are obtained.
2. Color tint (shading) is estimated based on the residual stats.
3. The inverse value of color tint is multiplied by the reference table value to obtain the corrected value.



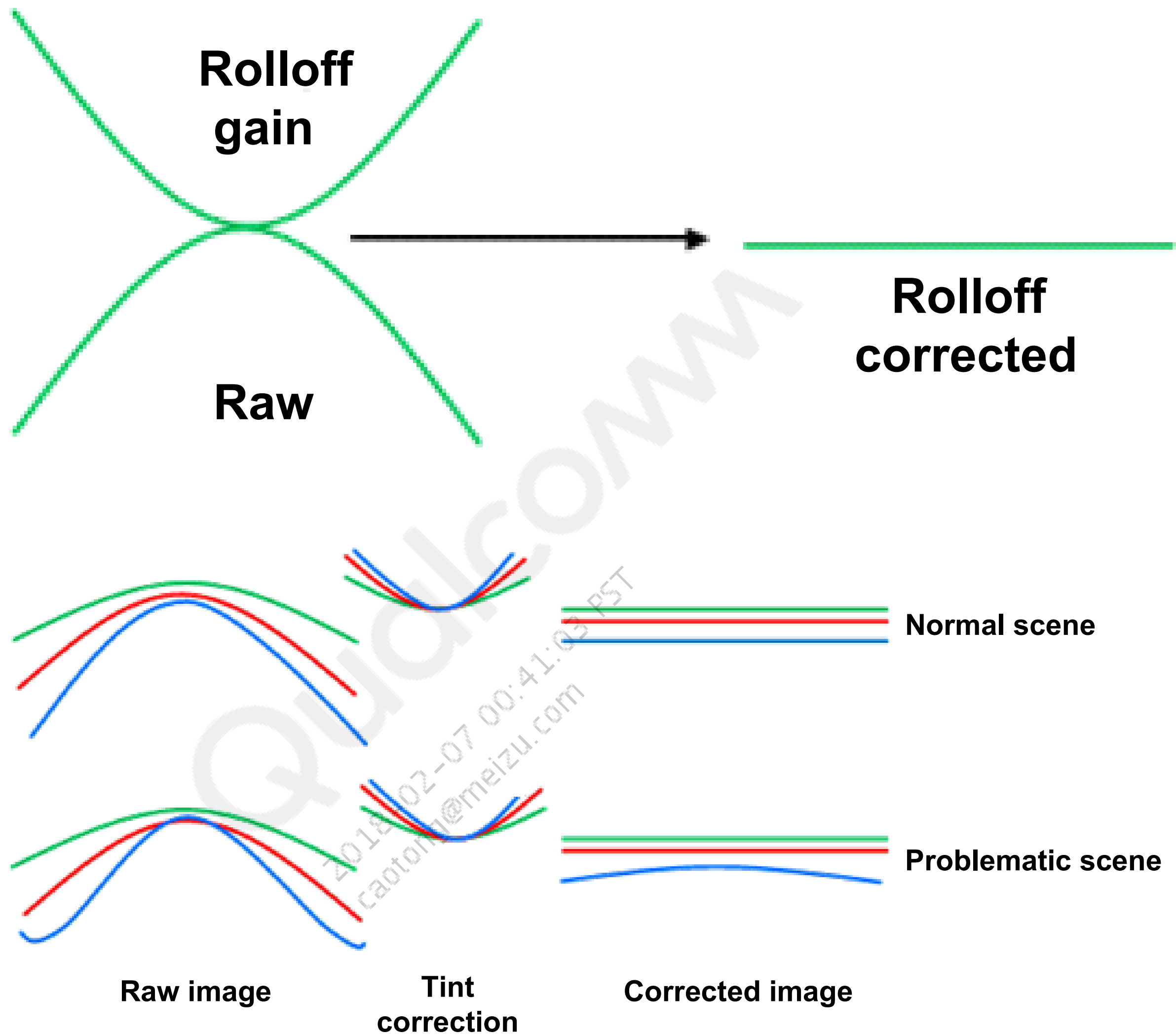




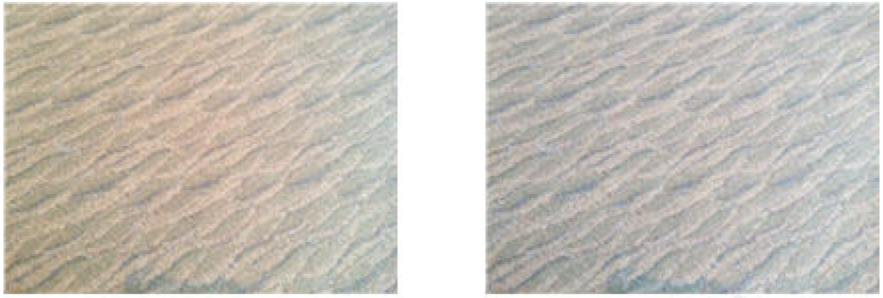
The conventional method of using AWB along with multiple calibration tables cannot compensate for individual module variation, uncalibrated or different illuminants, etc. Dynamic color shading correction estimates color shading from the BE stats and continuously updates the reference table. It provides a more robust solution and reduces the calculation burden by using a single calibration table.

(L) Conventional method (AWB + multitable); (R) Dynamic color shading correction (tintless)

LSC is applied to Bayer raw images to reduce the computation load. Instead of modeling the curve and performing polynomial calculations, a linear piecewise approximation with the help of LUTs is used. Each color has its own LUT, so the brightness attenuation and color shift problems can be solved simultaneously.



Examples of lens shading correction

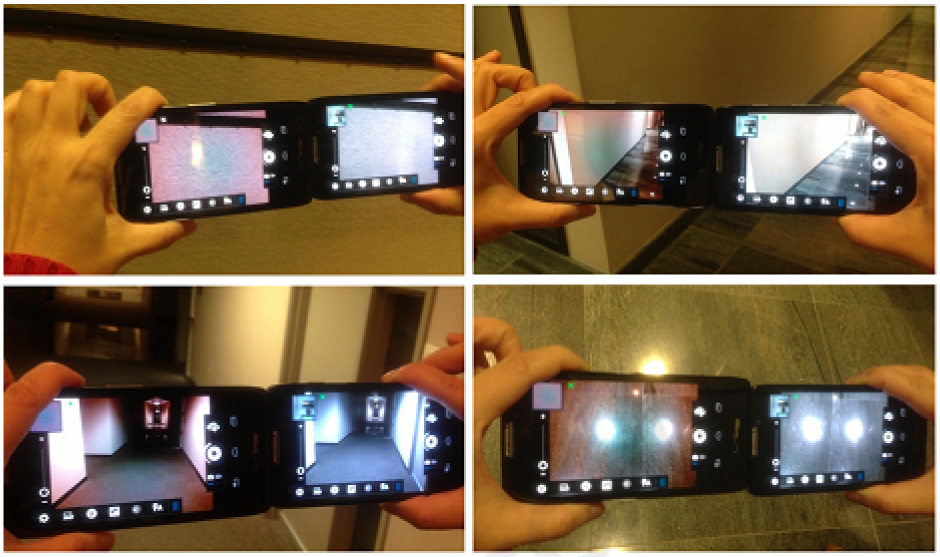


(L) Tintless off, indoor. (R) Tintless on, indoor

(L) Tintless off, mixed light. (R) Tintless on, mixed light

(L) Tintless off, lightbooth; (R) Tintless on, lightbooth

The following figure illustrate the same image using a commercial solution (left) compared to tintless.



7.5.1 Tune LSC

Based on the size and quality of the camera lens, the center of an image may appear brighter than the corner regions. Lens rolloff tuning corrects brightness attenuation and nonuniform color shading (tint).

Prerequisites:

Before tuning the LSC module, determine how many lighting regions you need to define and create them in the Parameter Editor.

1. On the **Pipeline** tab, click **LSC** and then click a region in the Regions group.

   NOTE Tintless correction is enabled by default. To disable tintless correction, go to the Parameter Editor and navigate to **SW_CONTROL** > **Tintless 20 Sw** > **Enable Section** > **Tintless En**, and enter a value of 0 to disable this parameter. Low light and normal light conditions use the same set of images. Different percentage correction values are applied to the two different lighting conditions to generate LSC compensation tables.

2. In the **LSC** section, click **Load Image** and select a rolloff raw image. Click **Open**.
3. Use the **Radius** percentage slider to adjust the area (from the center of the image) to be corrected. Use the **Correction** percentage slider to adjust the amount of correction to apply. Click **Optimize**.
4. Continue to apply additional radius percentage and correction percentage settings until the updated rolloff curve achieves a more flattened appearance.
5. Repeat this procedure for the remaining regions.
6. When complete, click **File** > **Save To Project** to preserve the tuned LSC settings.

7.6 Demosaic tuning concepts

Use the default values for the demosaic module. The demosaic module filters and reconstructs a complete RGB color image from color filter array (CFA) samples.

A digital color image typically comprises three color samples, namely red (R), green (G), and blue (B), at each pixel location. However, using three separate color sensors for measuring the three R, G, and B color values at each pixel location in a digital camera is very expensive. Thus, most digital cameras employ a single-chip image sensor, where each pixel in the image sensor is covered with an R, G, or B color filter for capturing the color information. The mosaic of tiny color filters covering the pixels in a single-chip image sensor is referred to as the color filter array (CFA). The most commonly used CFA is the Bayer mosaic formed by the replication of two-by-two sub-mosaics, with each sub-mosaic containing two green, one blue, and one red filter.

7.7 HDR tuning concepts

Scenes that have a wide dynamic range of luminosity may contain information that is not obvious on devices. As such, the resulting images may lack detail in the light or dark regions of the image. If the camera sensor supports high dynamic range (HDR), an HDR image can be taken by combining images that are underexposed and overexposed as follows:

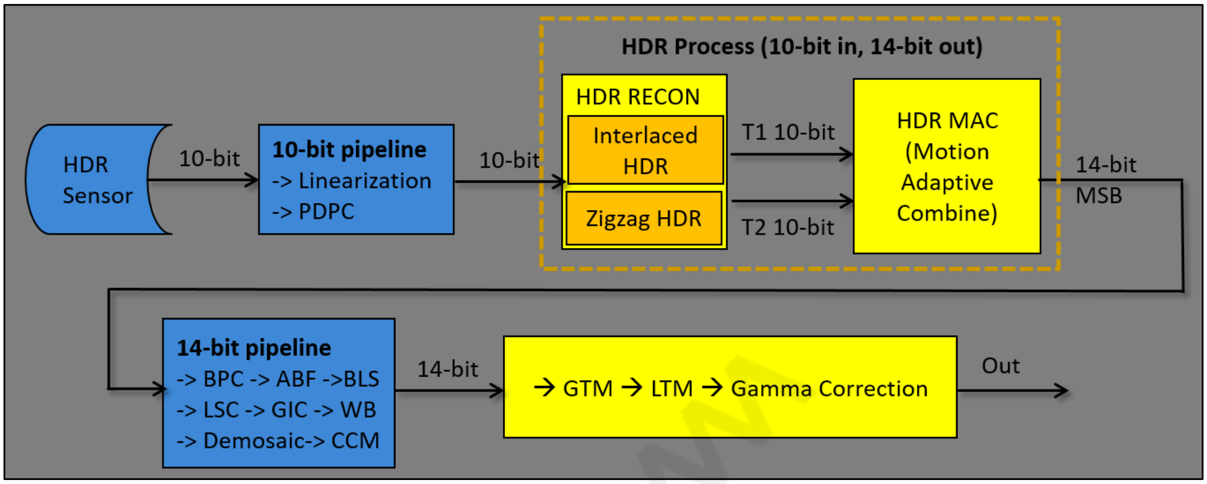
- The underexposed image provides details in the highlights region of the scene.
- The overexposed image provides details from the shadows region of the scene

The images are then reconstructed and blended to produce an image that more accurately reproduces the scene.

(L) Normal or non-HDR image, (R) HDR image

The HDR process consists of two sub-modules:

- HDR Recon – Interlaced HDR and zigzag HDR reconstruction
- HDR MAC – Motion adaptive combination



The input to the process is 10-bit zigzag data where T1 is the image with longer exposure and T2 is the image with shorter exposure. The output is scaled-up HDR Bayer to the full 14-bit range.

7.7.1 HDR parameters

The following table lists the parameters used for tuning the HDR module. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 7-5 HDR parameters

Name	Description	Data range	Default	Effect
recon_min_factor (iHDR only)	This value represents min to mid factor to force vertical edge direction.	0 to 31 allowed, but 0 to 16 meaningful.	12	Scaling factor to force vertical direction: if 16*d_min>min_factor*d_sec_min (min and mid not sufficiently apart), force vertical edge direction.
recon_flat_region_th (iHDR only)	This value represents the threshold for the flat region in edge detection.	0 to 1023	64	If d_max<th, consider current 2x2 pixel as flat region and force vertical edge direction.
recon_h_edge_th1	This value represents threshold 1 for the edge switching function.	0 to 1023	32	Controls the threshold of edge switch
recon_h_edge_dth_log2	log2(th2-th1) for horizontal edge switching function.	4 to 8	4	Controls the slope of the soft switching.
recon_motion_th1	This value represents threshold 1 for the motion switching function.	0 to 1023	512	Controls the threshold to recognize having motion.

Table 7-5 HDR parameters (cont.)

Name	Description	Data range	Default	Effect
recon_motion_dth_log2	Brief description: log2(th2-th1) for motion switching function.	4 to 8	8	Controls the slope of the motion switching.
recon_dark_th1	This value represents threshold 1 for the dark noise switching function.	0 to 1023	10	Controls the detail/noise level trade-off of dark region.
recon_dark_dth_log2	log2(th2-th1) for dark noise switching function.	0 to 4	4	Controls the slope of the dark switching.
hdr_zrec_prefilt_tap0	the strength for low-pass pre-color difference filter	0-63	0	Control the strength of filter to remove color aliasing
hdr_zrec_g_grad_th1	This value represents threshold 1 for green gradient	0-4095	32	Controls the threshold of green gradient to do directional interpolation or bilinear interpolation
mac_motion0_th1	This value represents the threshold 1 of motion switch in HDR MAC	0 to 1023	60	Controls motion switch to use longer pixels or shorter pixels
mac_motion0_th2	This value represents the threshold 2 of motion switch in HDR MAC	0 to 255	24	
mac_motion0_strength	This value represents the motion adaptation strength value.	0 to 16	8	Controls how strongly motion would affect T1/T2 combination.
mac_low_light_th1	This value represents the threshold 1 value for low light switching.	0 to 16383	0	Controls the low light switching.
mac_low_light_strength	This value represents the log light switching strength value.	0 to 16	0	Controls how strongly lowlight would affect T1/T2 combination.

Table 7-5 HDR parameters (cont.)

Name	Description	Data range	Default	Effect
mac_high_light_th1	This value represents the threshold 1 value for high light switching.	0 to 16383	232	Controls the high light switching
mac_high_light_dth_log2	This value represents the difference value for high light switching.	2 to 14	10	Control the slope of the switching function

Table 7-6 HDR reserved parameters

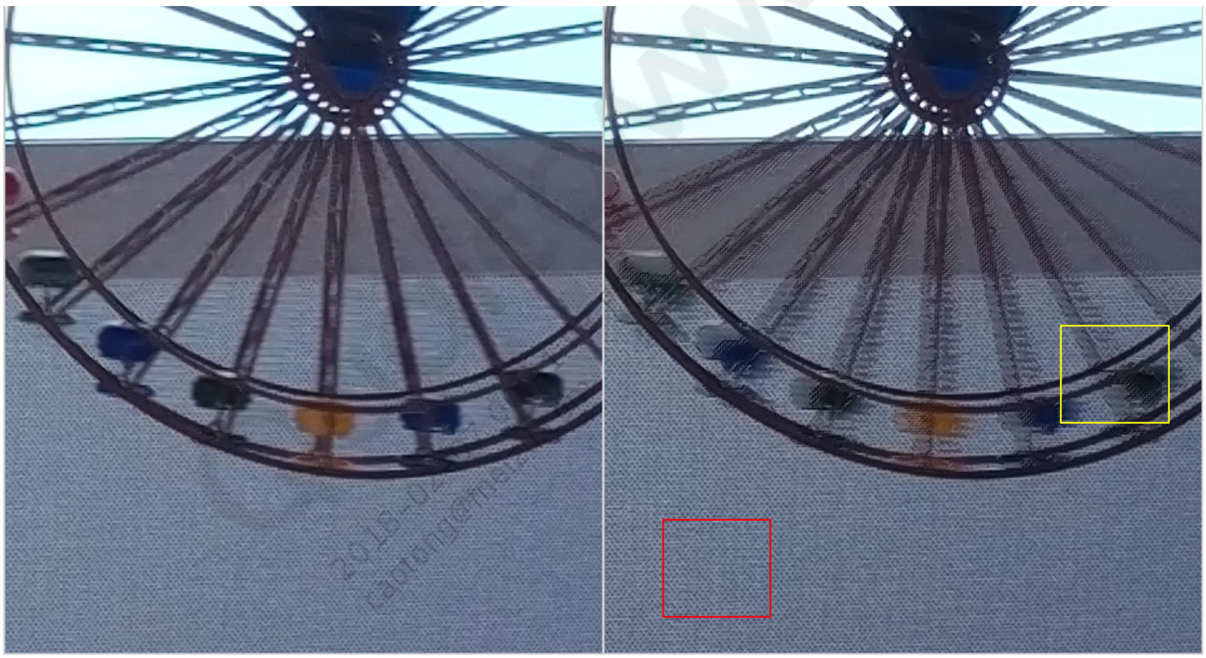
Name	Description	Data range	Default	Effect
hdr_recon_en	This flag when set to 1 enables HDR RECON module.	0, 1	0	This flag should be enabled when sensor is zzHDR pattern.
hdr_mac_en	This flag when set to 1 enables HDR MAC module.	0, 1	0	This flag should be enabled when sensor is zzHDR pattern.
hdr_msb_align	MSB Align for HDR module	0, 1	1	zzHDR default as MSB
hdr_zrec_g_grad_dth_log2	log2(th2-th1) for green interpolation switching function.	0 to 12	5	Controls the slope of the soft green interpolation switching
hdr_zrec_rb_grad_dth_log2	log2(th2-th1) for red/blue interpolation switching function.	0 to 12	5	Controls the slope of the soft red/blue interpolation switching
recon_edge_lpf_tap0 (iHDR only)	Tap0 value for low-pass filter applied before edge detection.	0 to 5	3	
mac_motion_dilation	Size of motion dilation max filter, 5 means -5 to 5, i.e. 11-tap filter.	0 to 5	5	Controls the dilation (motion mask) size.
mac_motion0_dt0	A parameter to avoid dividing by zero.	1 to 63	1	To avoid dividing by zero.
mac_low_light_dth_log2	log2(th2-th1) for low light switching.	2 to 14	6	Controls the slope of low light switching.
mac_smooth_enable	Enable smoothing for final output in MAC.	0 to 1	1	Enables smoothing filter.
mac_smooth_th1	Th1 of the smoothing switch in motion region.	0 to 256	192	Control th1 to smooth in motion region
mac_smooth_dth_log2	log2(th2-th1) for motion adaptive smoothing.	2 to 8	6	Controls the slope of motion adaptive smoothing.
mac_smooth_tap0	Tap0 value for the low-pass filter in motion adaptive smoothing.	0 to 5	5	Controls the filter strength

7.7.2 Examples of HDR parameter effects

These examples illustrate the effects of adjusting specific HDR parameter values.

Effects of the recon_motion_th1 parameter

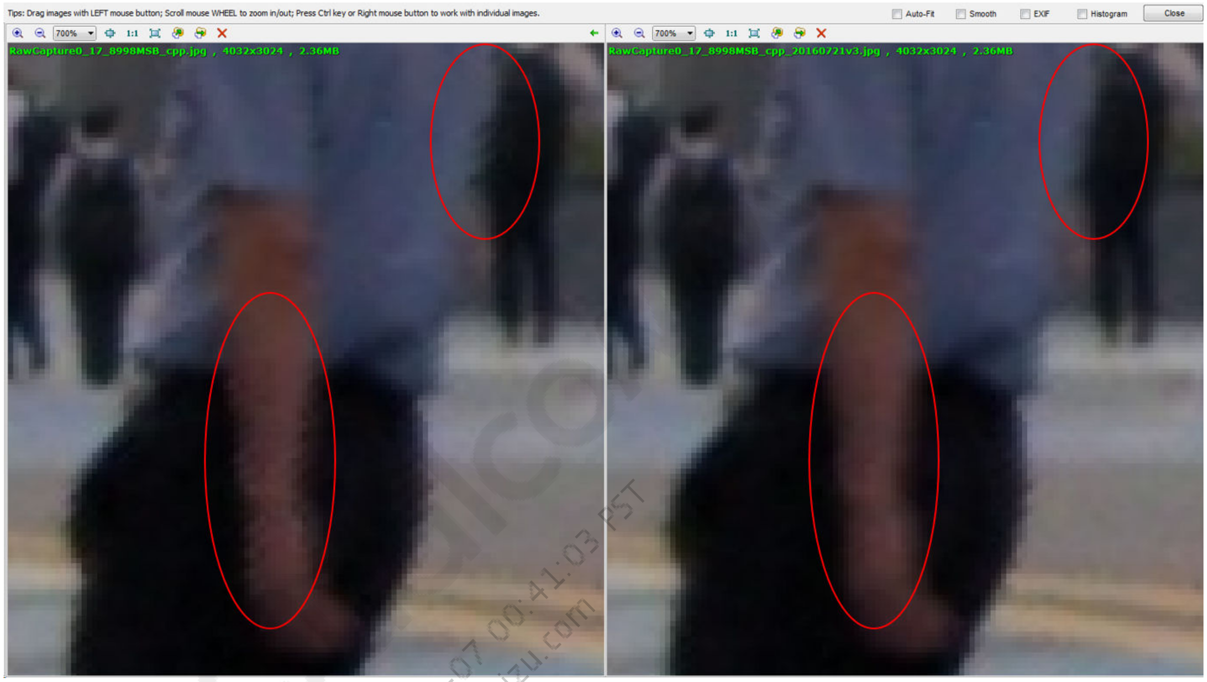
The recon_motion_th1 parameter controls the threshold 1 motion switching function. In the image at right in the following example, the parameter is set at 1023 which results in less interpolated T1 and more scaled T2, thereby producing more resolution and more zigzag artifacts. In the image at left, the parameter is set at 184 which results in more interpolated T1 and less scaled T2, thereby producing less zigzag artifcats and less resolution.



(L) recon_motion_th1 = 184, (R) recon_motion_th1 = 1023

Effects of the recon_dark_th1 parameter

The recon_dark_th1 parameter represents threshold 1 for the dark noise switching function and controls the detail/noise level trade-off in dark regions. In the example below, increasing the parameter value reduces the zig-zag artifacts in the dark regions.



(L) recon_dark_th1 = 0, (R) recon_dark_th1 = 10

Effect of mac_motion parameters

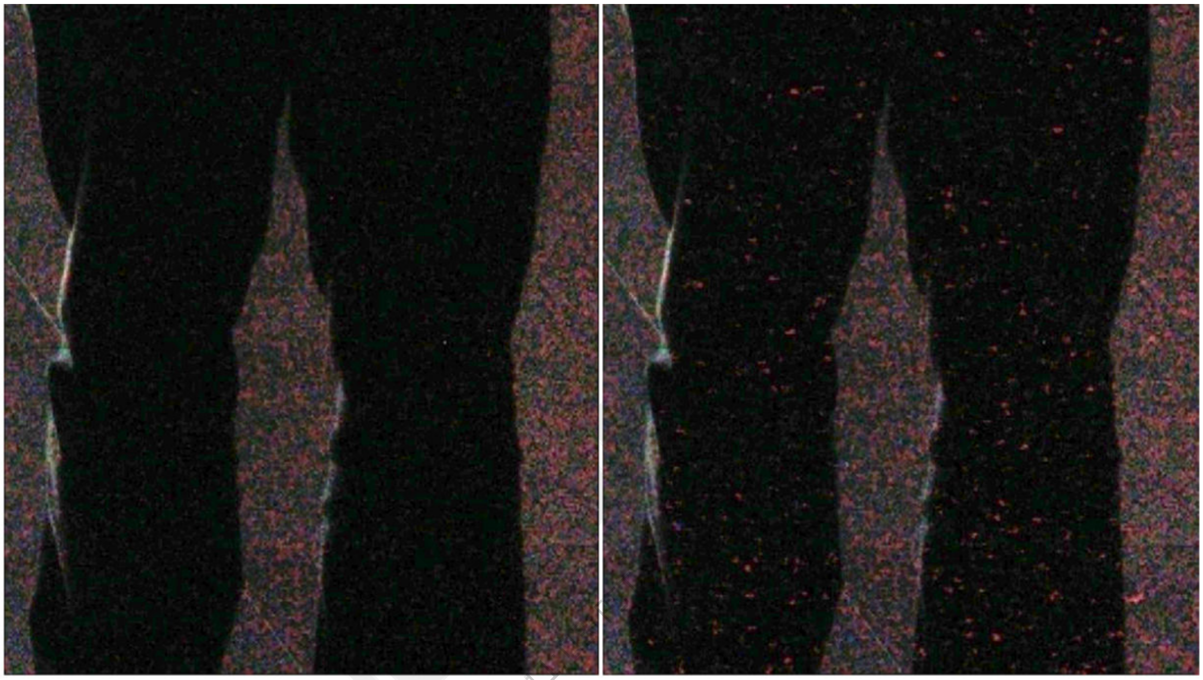
The mac_motion parameters work together to control the motion switch in HDR MAC to use longer or shorter pixels and to control the motion adaption strength value.

(L) Shorter exposure pixels, sharper but noiser (R) longer eposure pixels, more motion blur

In the examples above, the image at left has lower values for the mac_motion threshold parameters and the image at right has higer values. The range for the mac_motion threshold paraemeters is 0 to 1023, the lower the value the shorter the pixels. The following table shows the parameter values for each example.

Parameter	Example (L)	Example (R)
mac_motion0_th1	16	60
mac_motion0_th2	10	24
mac_motion0_strength	16	8

If mac_motion0_th1 and mac_motion0_th2 are set too small, it will over-detect dark regions as motion regions and, as a result, blend more noisy T2 in the MAC output. Compare the two images in the following example where the image on the right has smaller values for the parameters and shows more noise in the dark regions.

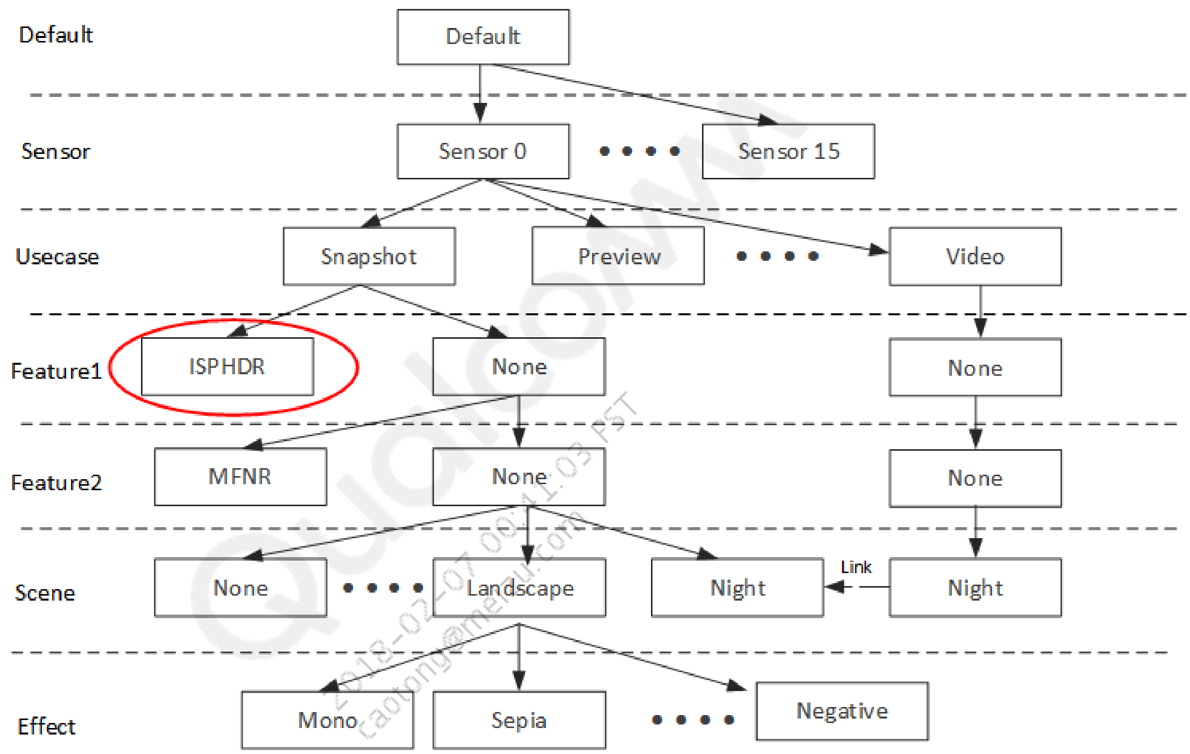


(L) mac_motion_th1 = 16, mac_motion_th2 = 10 (R) mac_motion_th1 = 8, mac_motion_th2 = 5

7.7.3 Tune HDR

Prerequisites:

If the camera sensor supports high dynamic range (HDR) and you want to offer an HDR setting on the device, add the ISPHDR mode to the sensor|usecase that will offer the feature. Do this for each sensor|usecase that will offer an HDR setting.



Add the ISPHDR mode for each sensor and use case that will offer an HDR setting

Tuning HDR primarily involves retuning other modules to compensate for the sensitivity afforded by the HDR module. The following is the general procedure for HDR tuning.

1. Accept the defaults for the HDR modules.
2. Retune the coordinating IPE modules. The tone curve tuning in the GTM modules is the most important for dynamic range.
   - PDPC – To remove bad pixels, tune PDPC so as not to harm the resolution but still remove bad pixels. Set the T1 and T2 offsets as T2 offset = T1 offset * exposure ratio. The T1 offset parameters are `bpc_offset` and `bcc_offset`. The T2 offset parameters are `bpc_offset_t2` and `bcc_offset_t`.
   - ABF – Retune ABF to remove dots and zigzags by HDR and tone curve. Retune the noise_std_lut when T1/T2 has a gain difference.
   - GTM – Tone curve tuning is the most important for dynamic range. Adjust the tone curve to generate 10-bit output by high dynamic range.
3. Simulate and test.

4. Adjust HDR parameters as necessary.
   a. To achieve good resolution, tune the `recon_h_edge_th1`/ `recon_motion_th1`/ `recon_dark_th1` parameters in the HDR module.
   b. To remove motion effect, tune the `mac_motion0_th1`/`mac_motion0_th2`/ `mac_motion_strength` parameters in the HDR module.
   c. Tune other HDR parameters as needed when there are critical HDR issues with the default values.
5. Repeat procedure until performance goals are met.

RELATED INFORMATION

“Tune ABF” on page 86
“Tune PDPC/DSBPC” on page 95

8 IPE tuning

8.1 ANR tuning concepts

Advanced noise reduction (ANR) is a multipass spatial noise filtering process available for snapshot and video modes.

ANR detects edges, preserves edges, and combines the image with a lowpass image to create a new image with lower noise levels. ANR has four passes for snapshot mode (1:64, 1:16, 1:4, 1:1) and three passes for preview and video modes (1:16, 1:4, 1:1).

If the image is calibrated correctly by ANR, it should have a good result as shown in the below example.

(L) No ANR; (R) Good calibration

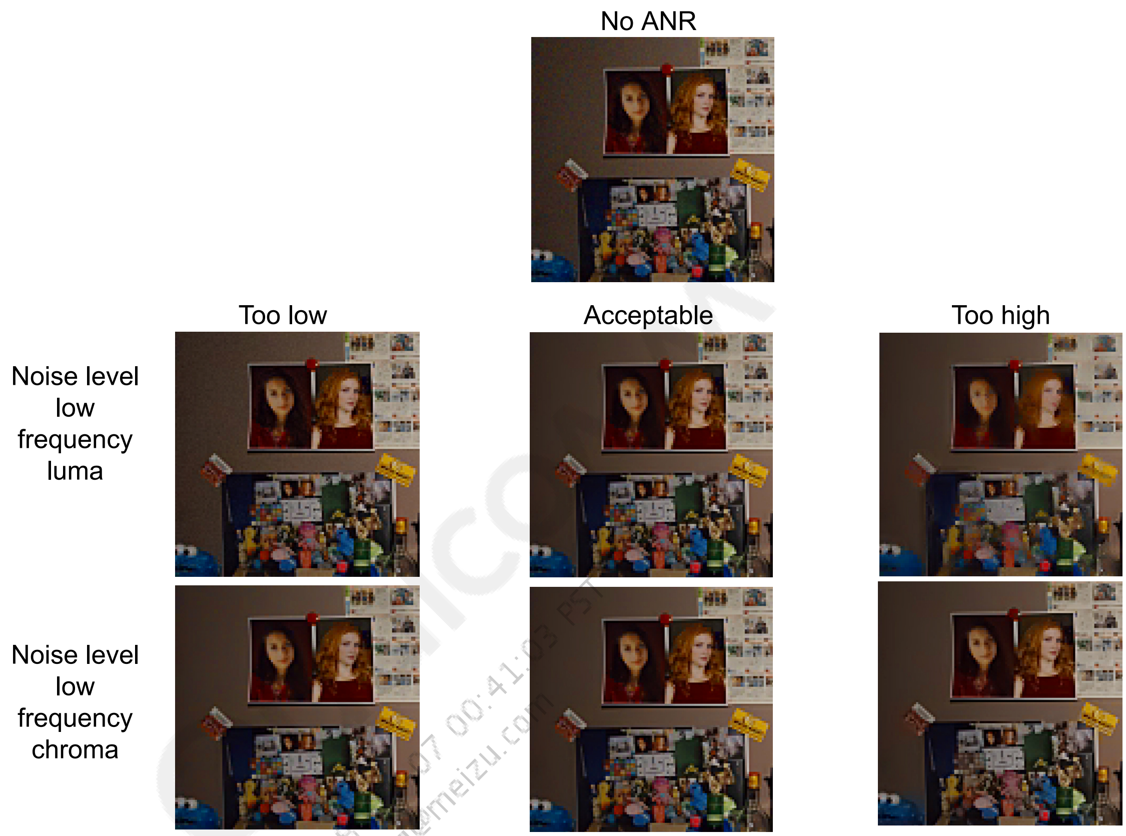
Incorrect calibration can cause over-filtering. This could be caused by damaged MCC images (miscolored areas on the color patches), or if the color patches are not selected correctly (too small or positioned incorrectly).

Inaccurate flat field images may result in radial noise patches and heavily effect calibration noise detection, especially in low frequencies.

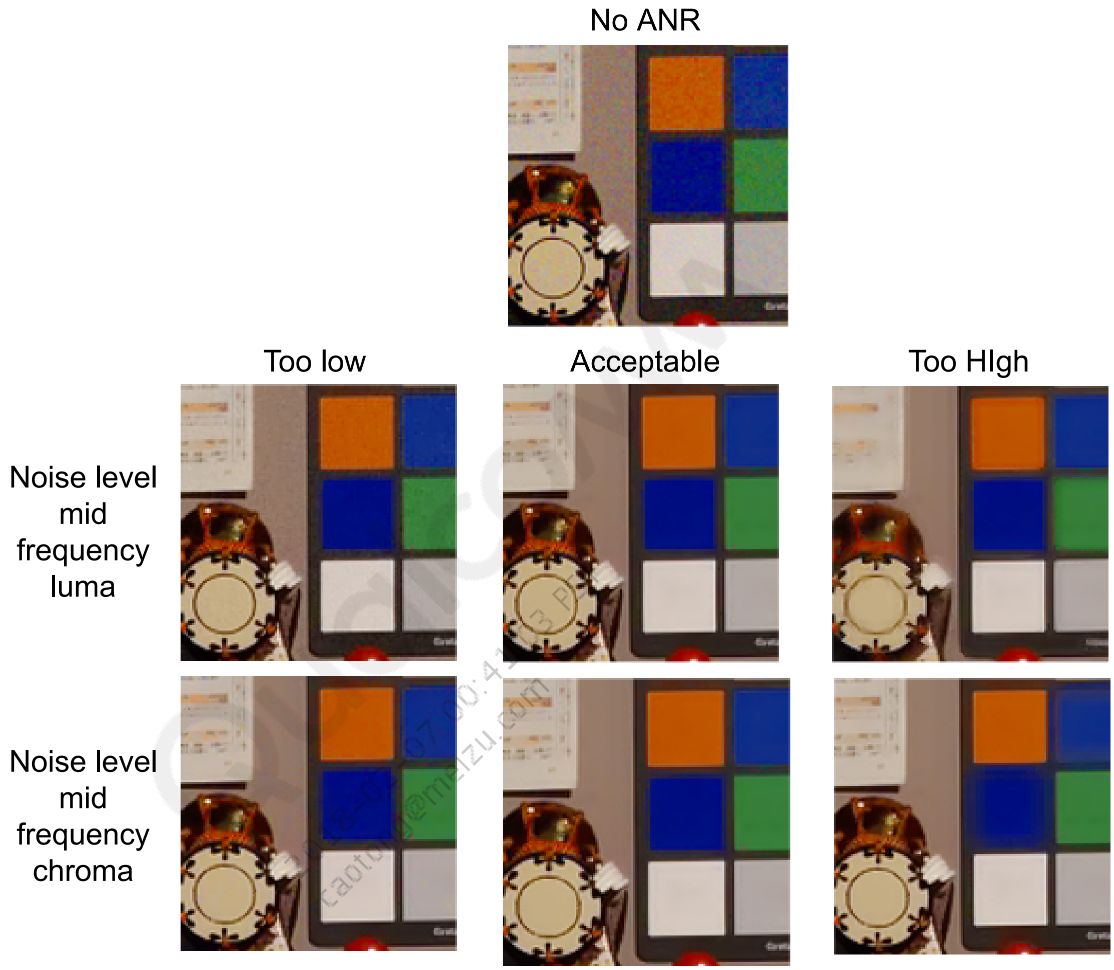
(L) Calibration with accurate flat field image; (R) Calibration with defective flat field image

(L) Less noise in corners; (R) More noise in corners

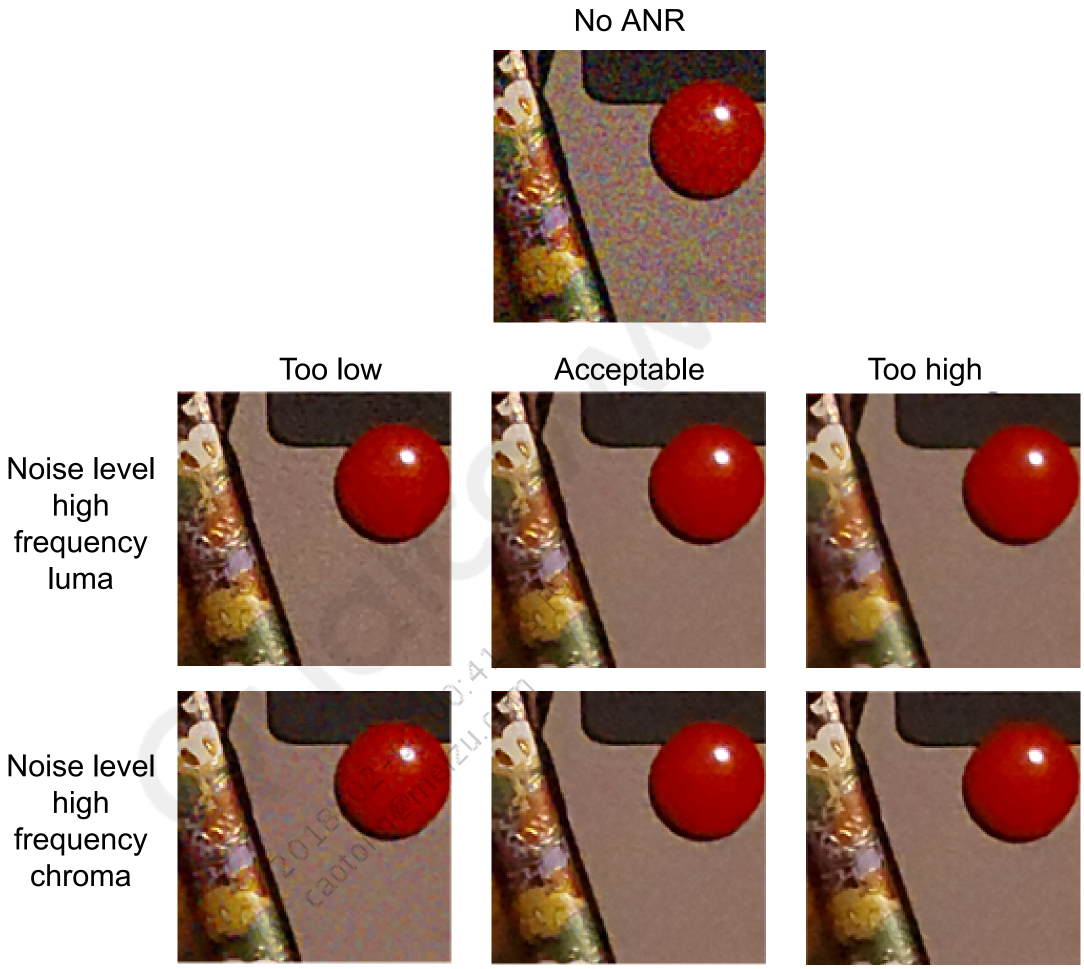
ANR for noise level low frequency with a DC16 image:



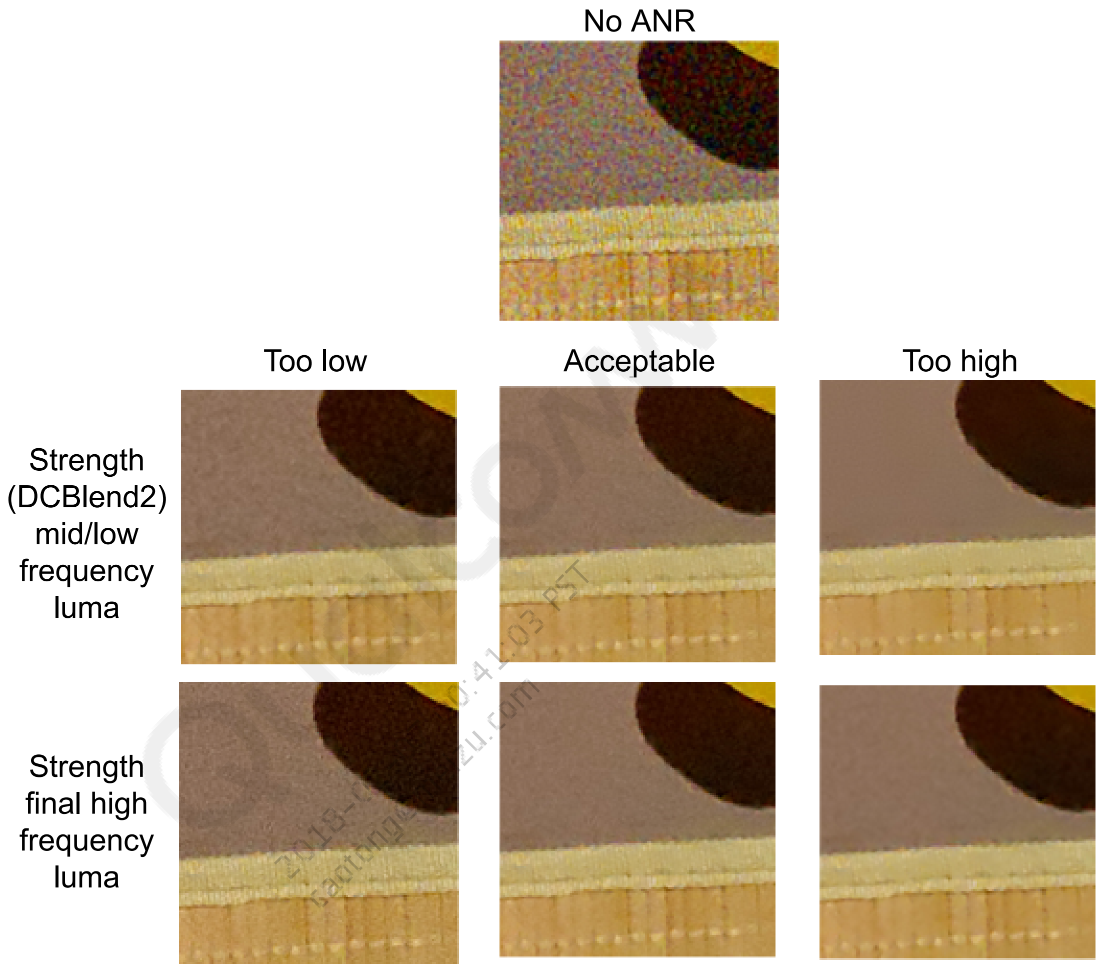
ANR for noise level mid frequency with a DC4 image:



ANR for noise level high frequency with a full image:



ANR for strength (DCBlend2) mid/low frequency:



Lens-dependent noise reduction (LNR) is used for aggressive or subtle radial filter behavior. Typically, running a calibration generates good results. LNR can also be used to tune when flat-field images are not available.

There are four LUTs to adjust, but it is recommended to tune one LUT and apply to the other three. Once an acceptable LUT has been created, adjust the LUTs separately as needed (e.g., more radial filtration on luma only).

Points of interest (POIs) can be used at the image corners and flat areas to understand the amount of scale that should be applied via thresholds.

Inter length control is used when flat images contain noise leftovers due to high derivatives that do not pass the thresholds. Each pass uses inter length control to check if the filtered pass beneath it is flat for a pixel. Inter length control can allow lower thesholds and preserve finer details while maintaining clean flat areas.

Inter length control may cause damage to low contrast details on flat areas, which means the pass below it has been overfiltered. This can be fixed using the **Base Functions** menu. Inter length control may also cause noisy edges, which is caused by a high strength.

Sometimes noise is detected correctly, but a more subtle reduction is preferred to preserve details. Just blending with the unfiltered image may reveal many unwanted peaks. *Filter Final Strength* enables a smart blend of filtered and unfiltered images.

Some areas with defined colors (i.e., faces or sky) require more filtering to make the image more flattering. CNR allows the base functions to be raised up to 5 areas in the YUV space. For skin color, CNR uses face detection indicators to avoid overfiltering areas with similar colors (i.e., wood).

Chroma noise is harder to clean on edges due to high luma derivatives. False colors have two ways to correct this issue that affect only chroma channels:

- Gray edge treatment – Detects strong edges that should have low chromaticity (gray edges) and adjusts the chroma filter to filter it
- Chroma edge treatment – Performs median filter on gray edges and on areas without details that have color noise

(L) Before tuning false color; (R) After tuning false color

RELATED INFORMATION

“ANR parameters” on page 127
“Tune ANR” on page 130

8.1.1 ANR parameters

The following table lists the gray edge parameters used for tuning the ANR module.

Parameter name	Description	Tuning	Default	Must tune?
Chromaticity threshold	Defines the chroma threshold conditions for identifying gray areas. Gray edge treatment will not apply to any pixel with greater chromaticity than the End value.	Defines start/end points	Low default: 2; low range: [0, 182] High default: 3; high range: [0, 182]	Use defaults
Y max derivative threshold	Defines the derivative threshold conditions for identifying gray areas. Gray edge treatment will not apply to any pixel with a lower Y derivitive than the Start value.	Defines start/end points	Low default: 255; low range: [0, 255] High default: 255; high range: [0, 255]	Use defaults
enable_grey_treatment_thr_modification	Enables increasing chroma filter thresholds in base functions. Y, U, and V will be increased to these values correspondingly (according to confidence).	Enable/ disable feature	Default: 1; range: [0, 1]	Use defaults
enable_grey_treatment_isotropic_filter_blend	For full images only. Blends pixel chroma with chroma isotropic filtered pixel according to confidence of gray edge detection.	Enable/ disable feature	Default: 1; range: [0, 1]	Use defaults
enable_grey_treatment_dcblend2_chroma_modification	Blends pixel chroma with chroma filtered lowpass image pixel according to confidence of gray edge detection.	Enable/ disable feature	Default: 0; range: [0, 1]	Use defaults
Detail corner sensitivity Y/UV ■ median_detect_corner_detail_sensitivity_y ■ median_detect_corner_detail_sensitivity_uv	Defines the level of local identified details.	Higher value = less filtering	■ Default: 16; range: [0, 65] ■ Default: 16; range: [0, 64]	Use defaults

Parameter name	Description	Tuning	Default	Must tune?
median_detect_triple_chromaticities_detail_thr_up_down	If a pixel is located on a strip between two other colors, performing median may smear the pixel. If the chromaticity difference in both directions is higher or equal than this value the median is not applied.	Anything higher than this value is defined as a detail	Default: 8; range: [0, 1023]	Use defaults
median_detect_isotropic_neighbors_detail_sensitivity	Local details identification according to neighbors detail detection.	Higher value = less filtering	Default: 64; range: [0, 128]	Use defaults
median_detect_directional_neighbors_detail_sensitivity	Similar to neighbors isotropic detail sensitivity but median still may be performed on directions that are not detected as detail.	Set this value as lower than neighbors isotropic detail sensitivity	Default: 32; range: [0, 128]	Use defaults
bilateral_decision_minimalsize	If flat length on any direction is higher or equal to this value, a bilateral filter will be performed instead of a median. The bilateral filter is generally preferable to the median filter as it uses averaging.		Default: 6; range: [0, 6]	Use defaults
luma_input_indication_thr_modification_scale	The strength is the maximal thresh gain factor on current PASS assuming fully flat pixel detected by input indication from the lower pass.		Default: 48; range: [16, 80]	Use defaults
chroma_input_indication_thr_modification_scale	The strength is the maximal thresh gain factor on current PASS assuming fully flat pixel detected by input indication from the lower pass.		Default: 48; range: [16, 80]	Use defaults

Parameter name	Description	Tuning	Default	Must tune?
luma_center_binarization_minflatval	If the center pixel is flatter than center_binarization_minflatval, it will indicate the pixel as flat without any regard to neighbors. If not, the decision will be done according to neighboring decision.		Default: 2; range: [1, 5]	Use defaults
chroma_center_binarization_minflatval	If the center pixel is flatter than center_binarization_minflatval, it will indicate the pixel as flat without any regard to neighbors. If not, the decision will be done according to neighboring decision.		Default: 2; range: [1, 6]	Use defaults
luma_neighbours_binarization_minflatval	Neighbors are counted as flat if they are flatter than neighbours_binarization_minflatval.		Default: 1; range: [1, 4]	Use defaults
chroma_neighbours_binarization_minflatval	Neighbors are counted as flat if they are flatter than neighbours_binarization_minflatval.		Default: 1; range: [1, 5]	Use defaults
luma_neighbours_agreement_sensitivity	If percentage of neighbores are determind as flat (according to comparison with neighbours_binarization_minflatval) is greater than neighbours_agreement_sensitivity, center pixel will be indicated as flat for higher pass.		Default: 80; range: [0, 128]	Use defaults

Parameter name	Description	Tuning	Default	Must tune?
chroma_neighbours_agreement_sensitivity	If percentage of neighbores are determind as flat (according to comparison with neighbours_binarization_minflatval) is greater than neighbours_agreement_sensitivity, center pixel will be indicated as flat for higher pass.		Default: 80; range: [0, 128]	Use defaults
flat_kernel_blend_weight	Defines the amount of blend of the Flat kernel on flat area.		Default: 128; range: [0, 128]	Use defaults
Luma DCBlend2 Strength	Defines the blend amount of lower passes, i.e., noise reduction in low frequency.		Default: 80; range: [0, 128]	Use defaults
Chroma DCBlend2 Strength	Defines the blend amount of lower passes, i.e., noise reduction in low frequency.		Default: 128; range: [0, 128]	Use defaults
Luma Strength HF	Enables a smart blend of filtered and unfiltered image. Does not blend back peak noise. In that case, the unfiltered pixel may be blended up to *1 – parameter value*.		Default: 256; range: [0, 256]	Use defaults

8.1.2 Tune ANR

Advanced noise reduction (ANR) is a multipass spatial noise filtering for snapshot and video modes.

1. On the **Pipeline** tab, click **ANR** and then click a region in the Regions group.
2. Click **Load MCC Images** and load the preferred YUV MCC images.
   a. Click **Submit**.
   b. Drag the top-left MCC grid marker to the center of the top-left color patch. Drag the bottom-right MCC grid marker to the center of the bottom-right color patch. All the grid markers should align within the color patches of the MCC chart image. Ensure ~80% of the patches are selected for best results.
   c. Close the image.
   d. Repeat process for each MCC image.
3. Click **Load Flat Image** and load a flat field image.

   Inaccurate flat field image may result in radial noise patches and heavily effect calibration noise detection, especially in low frequencies.

4. Click **Calibrate**.
5. Click **Calibrate LNR** to run lens-dependent noise reduction (LNR). To manually adjust the LNR, see *(Advanced) Lens-dependent Nosie Reduction (LNR)*.

(Advanced) Reserved Parameters

6. For high and low power modes, click **Reserved Parameters** and in the Power Control group, enable or disable the following features:
   - Enable Chroma Filter Extension
   - Enable Luma Smoothing Treatment and Peak Treatment
   - Enable Chroma Smoothing Treatment and Peak Treatment

   These features are used for 4K60 or 1080p videos.

Basic Level

7. In the **Configuration** group, click **Basic Level**.
   a. Adjust the **Noise Detection** luma and chroma values as needed.

      Noise detection separates details from noise at each frequency range. A higher value applies more filtering, but may have inconsistent edges.

      Noise Level Low Frequency affects noise levels for DC16 images. **Noise Level Mid Frequency** affects noise levels for DC4 images. **Noise Level High Frequency** affects noise levels for full images.

   b. Adjust the **Noise Reduction** luma and chroma values as needed.

      The noise reduction amount defines the amount of blend between the original image and the filtered one. A higher value creates a softer image with less noise. It is recommended to keep the **Chroma** sliders at the max values.

      Adjust **Strength Final High Frequency** for cleaner images with high frequency granularity, retain textures and continuity, and reduce harsh trasitions between flat and non-flat areas.

      Adjust **Strength (DCBlend2) Mid/Low Frequency** for cleaner flat areas, to retain low contrast textures for flat areas, and mid frequency granularity.

   c. Click **Save** to retain all changes.

   d. Click **Close** to return to the main ANR tuning screen.

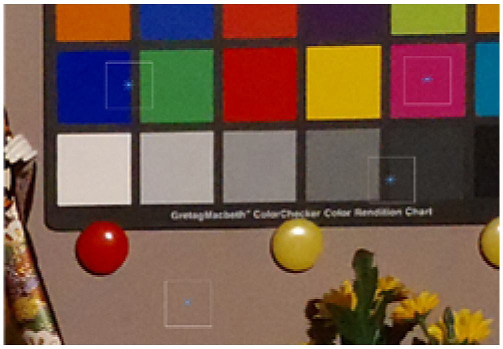
(Advanced) Base Functions

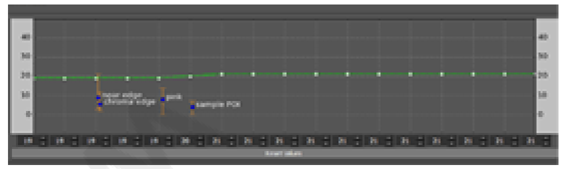
8. In the **Configuration** group, click **Base Functions**.
   a. Tune **Luma Filter** and **Chroma Filter** from lowest pass to highest (i.e., DC64 > DC16 > DC4 > Full).

      If an issue arises, find the lowest pass where the issue occurs, place a point of interest (POI), and try to resolve the issue.

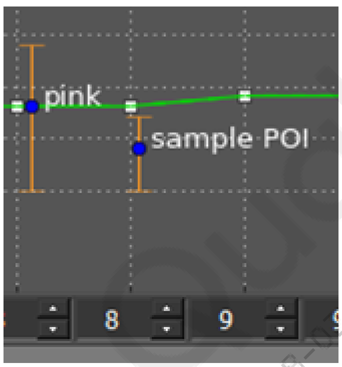
      DC64s typically do not require tuning.

   b. Add POI markers where noise occurs in an image. Review the markers on the threshold graph and if more pixels are distributed above than the function, the pixels were not filtered enough. If overfiltering occurs and more pixels are distributed below the function, adjust the function to lower the filtering level. In the graph, the X-axis represents the luma value and the Y-axis represents the derivative.



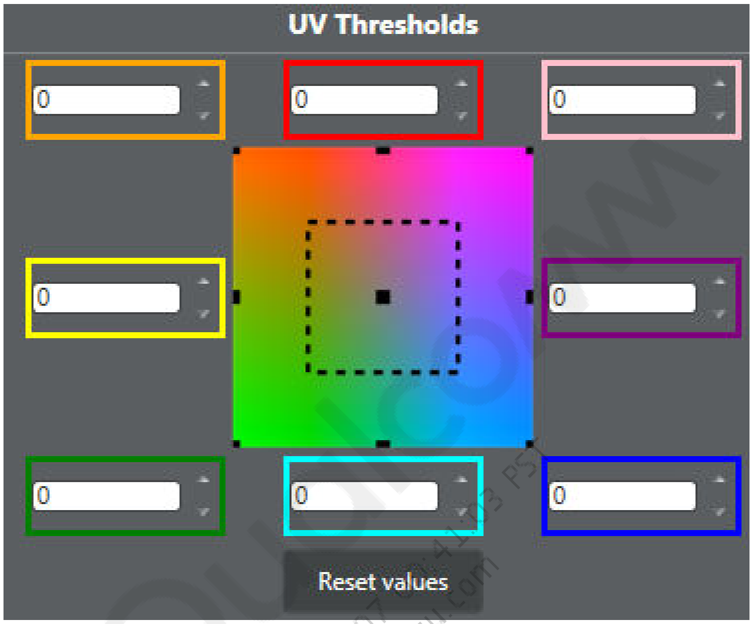


The I marker shows the highest and lowest derivatives around the POI. If the POI line goes above the threshold, the pixels above the threshold will be filtered. If the POI line is entirely under the threshold, all pixels will be filtered.



c. Adjust the **UV Thresholds** for **Luma Filter** and **Chroma Filter** as necessary to offset the threshold according to chroma values. For example, set a higher value for blues to increase filtering for sky-like colors.

The UV limit is the distance of the black squares (see following image) from the center. Larger values have less impact on low color saturation; smaller values have more impact. The luma and chroma UV limits are 64.



d. Set the scale for the functions set by the threshold.

3 (2nd der) sets the second order derivative scale. It is recommended to use this value for contours (gradients) and local peaks detection.

(Advanced) Lens-dependent Nosie Reduction (LNR)

9. In the **Configuration** group, click **Lnr**.

a. Tune the **Center** and **Ellipse parameters A** values, if needed.

The **Center** value defines the ellipse center position in the image. For the **Ellipse parameters A** value, 0.5 is a circle. Anything greater than 0.5 is an ellipse, and anything smaller than 0.5 is a wide ellipse.

b. Evaluate the noise in the image center. If the image is overfiltered or too noisy, adjust the **Base Functions**.

c. Adjust the four LUTs (LumaFilter, ChromaFilter, Y_Factor, and UV_Factor). It is recommended to tune one LUT and use **Copy Lut** and **Paste Lut** to copy and paste the results to the other three LUTs. Further adjustments can be made after.

d. To manually reset the LUT to be linear, set the reset to linear values and click **Reset to linear**.

The left value must be 1. The second value should be the maximum LNR correction, where the thresholds increase at the end of the image.

(Advanced) Inter Length Control

10. In the **Configuration** group, click **Inter Length Control** and adjust values as needed.

False Colors

11. In the **Configuration** group, click **False Colors**
    a. Deselect the **Enable** check box in the **Chroma Edge Treatment** group for all passes (full, DC4, DC16, and DC64).
    b. Enable and enter values for **enable_greytreatment_thr_modification** for DC4 and full passes only.
    c. Enter values for **Detection - Chromaticity Theshold** and **Detection - Y Max Derivative Theshold** for full pass. Only tune for DC4 if it is highly necessary. Do not tune other passes.
    d. Click **Save** to retain all changes.
    e. Click**Close**.

Filter Settings

12. In the **Configuration** group, click **Filter Settings** and adjust the values as needed.

    The Full Luma Filter Kernel feature allows tuning for flat and edge areas. This is for Full pass only. It is recommended to set the edge kernel flatter than the flat kernel.

CNR

13. In the **Configuration** group, click **CNR** and tune from the lowest pass.

    For example, if enabling CNR for Full and DC4, begin with DC4 and continue with Full. For Full, DC4, and DC16, begin with DC16, then DC4, then Full.

    Examine color indentification and check the impact of tuning on additional images to assess and understand the tradeoffs.
14. Repeat process for each region.
15. Click **File** > **Save to Project** to save any changes made.

Postrequisites:

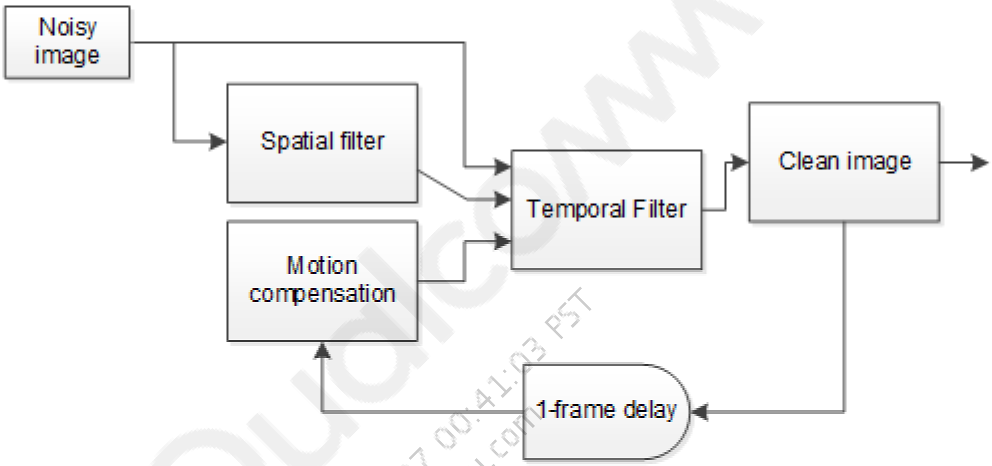
Tune the temporal filter (TF) module (video mode only).

8.2 TF tuning concept

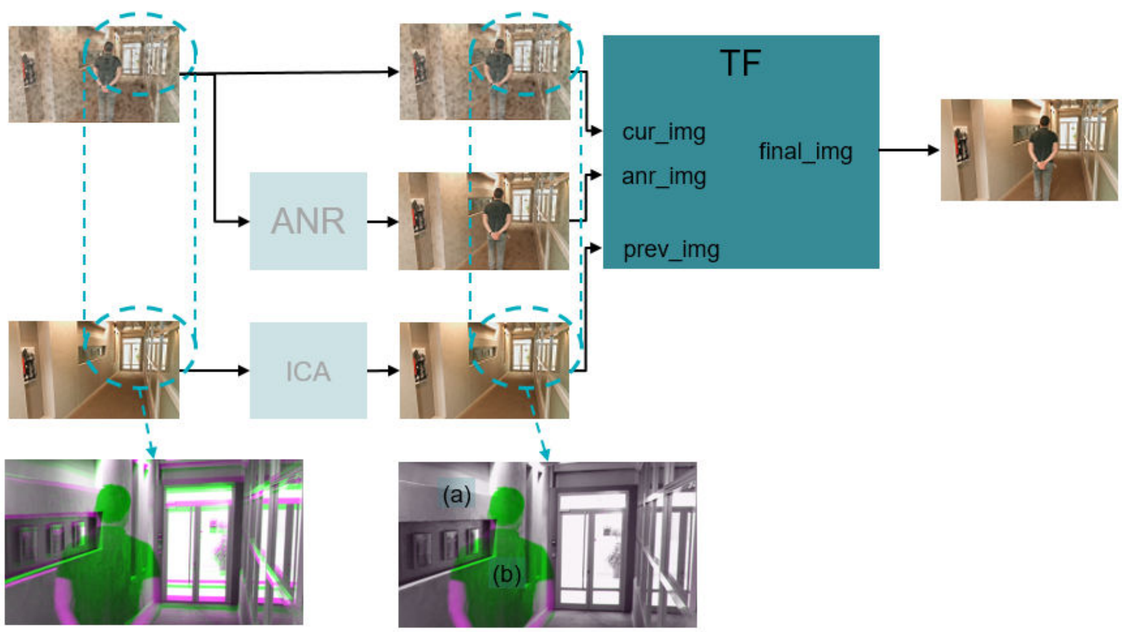
Temporal Filtering (TF) removes unwanted frequencies (noise) within an image.

TF performs the following:

- Temporal noise filtering: Attenuates the noise signals
  - Works on each video frame
  - Blends the current frame with the previous frame



- Warping: Warps the previous frame to align it with the current frame. TF behaves differently accoring to the use case.
  - In normal operation: blend cur_img with prev_img
  - When detecting local motion: blend cur_img with anr_img



Ghost detection calculates TF strength (FS). If the current and previous images have different content, FS = 0, otherwise the objects will appear twice (i.e., as ghosts). If the current and previous images have the same content, FS = 63 (maximum), as the high filtering strength ensures maximum noise reduction.

For blending:

- final = α_1 * prev + (1 - α_1) * cur_spatial
  - α_1 (temporal blending factor) is a tunable function of FS
  - When FS = 0, $\alpha_1 = 0$
  - When FS = 63, α_1 is maximal
  - $\alpha_{1,max}$is big with cleaner output, but slower convergence
  - $\alpha_{1,max}$ is small with faster convergence, but noise reduction is weaker
- cur_spatial = α_2 * ANR + (1 - α_2) * cur
  - α_2 (spatial blending factor) is a tunable function of FS
  - When FS = 0, α_2 is maximal
  - When FS = 63, α_2 is minimal
  - $\alpha_{2,min}$ is big with less noise in the output, but with degradation of details
  - $\alpha_{2,min}$ is small with more details preserved, but less noise reduction

The TF should be tuned after the ANR is tuned. During the TF tuning, the following units must be turned on:

- CAC
- CCM
- ASF
- GLUT
- 2D LUT
- Chroma suppression
- Skin color enhancement
- GRA
- M/N DS

If the ANR tuning changes, the TF blending factors are retuned.

8.2.1 Tune TF

Temporal Filtering (TF) removes unwanted frequencies (noise) within an image.

Prerequisites:

Tune ANR.

1. On the **Pipeline** tab, click **TF** and then click a region in the Regions group.
2. Click **Load MCC Images** and load the preferred image sequence with a minimum of 10 frames.
3. Click **Load Flat Image** and load a flat field image.
4. Click **Calibrate**.
5. Click **Calibrate LNR** to run lens-dependent noise reduction (LNR). To manually adjust the LNR, see *(Advanced) Lnr Setting*.
6. Click **Load Test Image** and select a video file or sequence or images (select frame _0).
7. Select the number of frames to review and click **Run!**.
8. Compare the final images of frames 0 and the last frame. The last frame should have less noise.
9. Compare the ANR image and the final image from TF. The final image should have less noise and more details.

Basic Level

10. In the **Configuration** group, click **Basic Level**.
    a. Adjust the **Spatial Noise Reduction** luma and chroma values as needed.
    b. Adjust the **Temporal Noise Reduction** luma and chroma values as needed.
    c. Adjust the **Motion Detection** luma and chroma values as needed for 1:16 scale and 1:4 scale.
    d. From the reserved parameters, select the **High Quality (High Power) Mode** check box to enable 1:1 scale luma and chroma adjustments, if necessary.
    e. Click **Save** to retain all changes.
    f. Click **Close** to return to the main TF tuning screen.

Base Functions

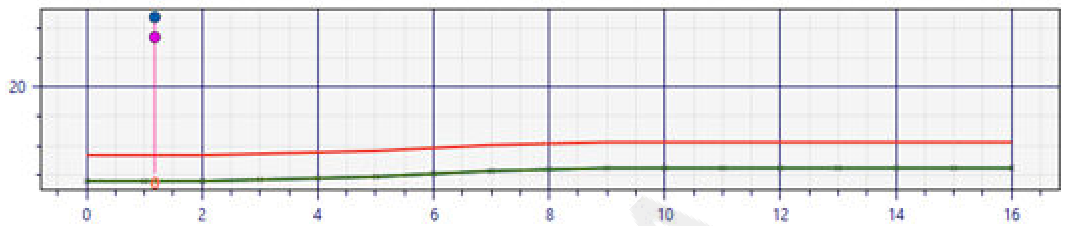
11. In the **Configuration** group, click **Base Functions**.
    a. Tune the gain from lowest pass to highest (i.e., DC16 > DC4). Full resolution is required only in High Quality mode and D64 is not required for videos.

       If an issue arises, find the lowest pass where the issue occurs, place a point of interest (POI), and try to resolve the issue.

       DC64s typically do not require tuning.

b. Add POI markers on stationary and moving objects.

The I marker shows the highest and lowest derivatives around the POI and helps separate noise from details. Stationary object POIs should be below the green **FS=Max** line. Moving objects should be above the red **FS=0** line.



c. Adjust the **FS=Max** and **FS=0** lines using the **Gain** controls (+ and -).

d. Assess the tuning quality by looking at the filtering strength (FS).

Moving objects should have an FS of 0. Stationary objects should have an FS of 63.

Blend Factor

12. In the **Configuration** group, click **Blend Factor**.
13. Adjust the **Spatial blend(a2)** values to control the strength of TF.

The **a2 max** value should be almost 100% where a lower values restore fine details on moving objects.

The **a2 min** value adjusts the amount of spatial denoising to add to temporal denoising on stationary objects. For a weaker value, use 0%. For noisy videos, a value of up to 70% may be required.

(Advanced) Options

14. In the **Configuration** group, click **Options**.

a. Adjust the **ErodeFilterSize** and **DilateFilterSize** values as needed.

0 = disabled; 1 = 5x5; 2 = 3x3.

If the erode filter value is larger than the dilate filter value, there are less ghost artifacts. If the erode filter value is equal to the dilate filter, there is more noise reduction on moving objects. The erode filter value cannot be smaller than the dilate filter value.

b. Adjust other values as needed.

8.3 CAC tuning concept

8.3.1 Tune CAC

Chromatix aberration causes fringes of color to occur along boundaries when different bandwidths of incoming light are refracted slightly by the lens and do not land in the same visual plane on the camera sensor. The CAC module is used to correct these visual artifacts in an image.

Prerequisites:

Before tuning the CAC module, determine the number of lighting regions required for this module and create them in the Parameter Editor.

Do the following to tune CAC:

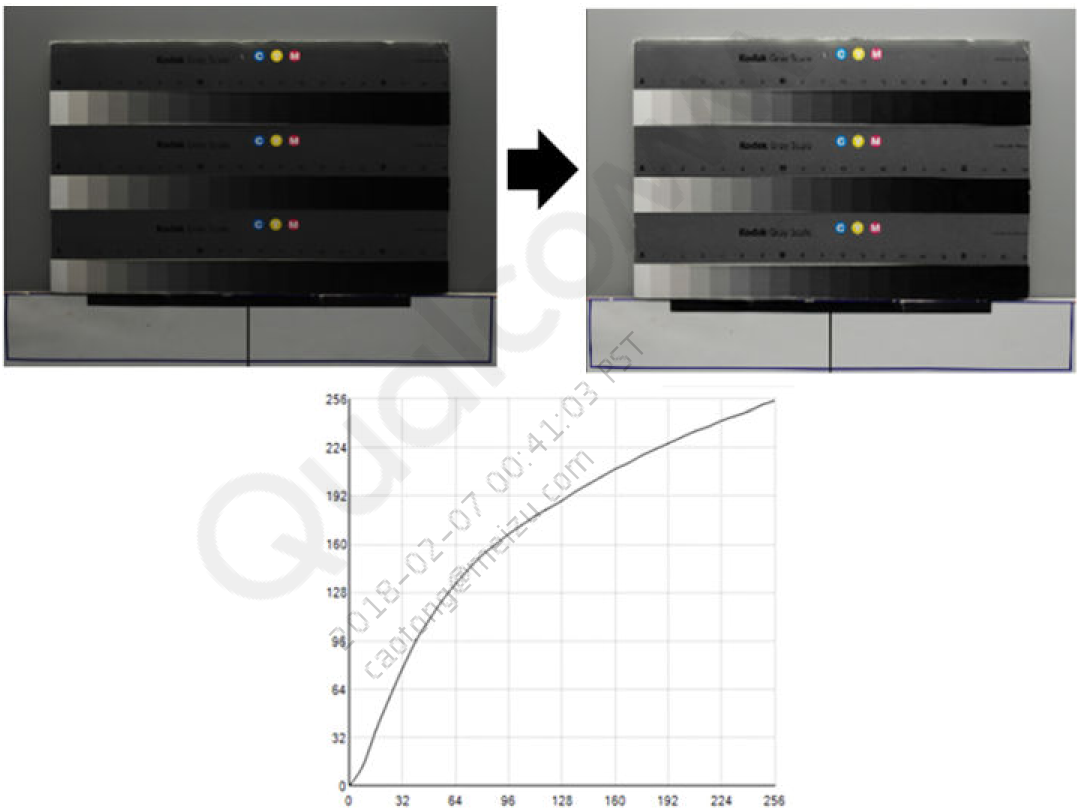
1. In the **Pipeline** tab, expand the**IPE** pipeline.
2. Select **CAC** to show the default region and Region Data.
3. Click **Region Data** > **CAC 22 Rgn** to show the values of various parameters.
4. As needed, tune the following CACv2 regional parameters:

Parameter name	Range of values	Default value	Description	Tuning
Cac En	0 or 1	1	Enables or disables CAC2. Values by default for each region: ■ Enabled: 1 and 2. ■ Disabled: 3,4,5, and 6.	Use the mutually exclusive Software CAC2 and chroma downsampling for low light.
Y Spot Thr	0 to 63	10	Smaller values increase the amount of bright spot correction.	Use default.
Y Saturation Thr	0 to 1023	960	Smaller values increase the amount of correction.	Keep the default value. If an increased correction is needed for the dim areas, set to 230.
C spot Thr	0 to 1023	40	Detects color differences between a pixel and its surrounding pixels.	Use default.
C Saturation Thr	0 to 511	20	Smaller values increase color artifact correction.	Keep the default value.

5. Repeat the previous step for each region.
6. To save the adjustments made to the parameters, click **File** > **Save To Project**.

8.4 Gamma tuning concepts

Traditional video display systems, such as those based on Cathode Ray Tubes (CRTs), have a non-linear, power-law response (roughly a 2.5 power function) to the pixel values. Gamma correction refers to the techniques of pre-adjusting the brightness of the image captured by an image sensor to boost darker signals for better display on video monitors. For example, gamma correction could be roughly a 1/2.5 power function of the pixel value. The figure below shows how gamma curve improves dark area's brightness.



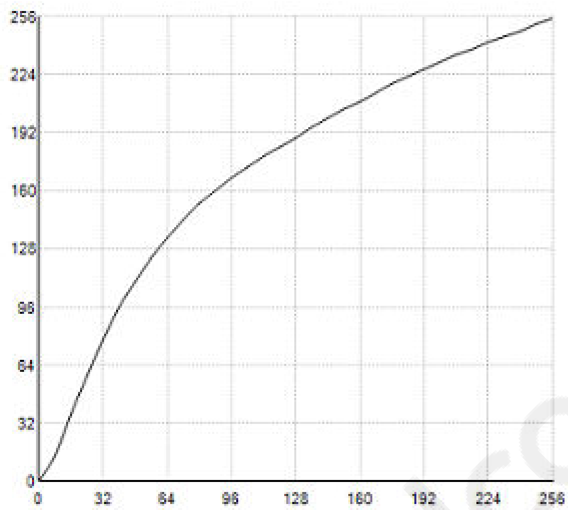
(L) Before; (R) After

Concept and algorithm

The proposed method provides a high quality gamma correction that can approximate practical gamma curves with virtually no loss in visual quality. From various experiments, it has been shown that there is no need to have 4096 entries of LUT for 12-bit data. Limited by the display resolution (8 bit per color per pixel) and human perception resolution, it is sufficient to use a few points to approximate the gamma curve and interconnect these points with straight lines. The linearly interpolated curve has almost the same visual performance as the original curve as long as the number of points that approximates the gamma curve is not too small. Per every frame, this single-pixel, single-color gamma transformation can be performed using a 64-value LUT for each of the colors: red, green, and blue.

Gamma table

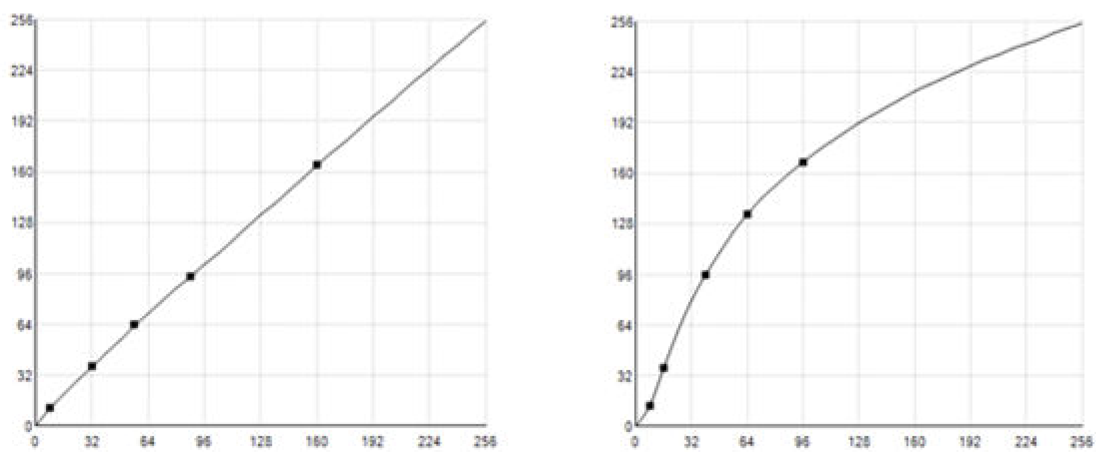
Even though the input data to the Gamma Correction block is 12 bits, it is not necessary to prepare a 4096 entry LUT. 64 entry-LUT is enough and these entries are expanded by linear interpolation.



Gamma tuning effects

Gamma tuning adjusts the pixel sensitivity of the camera sensor to increase the contrast between various shades of a particular color. When a digital image is captured, each level of gray is encoded to a certain pixel brightness value. This pixel encoding can either be linear-encoded or gamma-encoded.

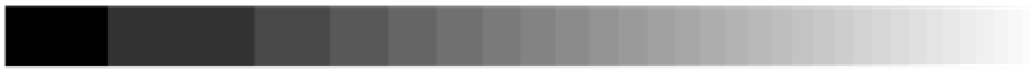
With linear pixel encoding, there is a 1-to-1 relationship between the amount of signal a pixel produces to the amount of photons hitting the pixel. However, gamma encoding is nonlinear, and the pixel signal level for certain tones are boosted or reduced to improve contrast. Reducing the far left (low-light) end and boosting the middle (brighter light) of the exposure curve increases perceived image contrast.



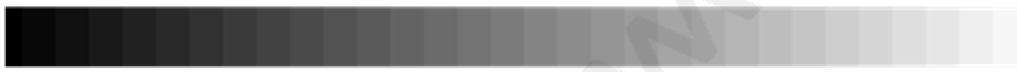
(L) Linear pixel encoding; (R) Gamma pixel encoding

For example, in the image that follows, the original grayscale is shown captured with linear encoding and gamma encoding. With linear encoding, large increments of the dark end of the scale are shown at the same shade, which results in low contrast of darker images. But with gamma encoding, the darker end of the scale breaks the scale into smaller increments at different brightness levels, which increases image contrast.

Original grayscale image



Grayscale image captured with linear encoding



Grayscale image captured with gamma encoding

To tune the gamma of the tuning device to have the same contrast as a reference device, adjust the gamma curve of the tuning device to be the same as the reference device.

RELATED INFORMATION

"Edit gamma curve" on page 142

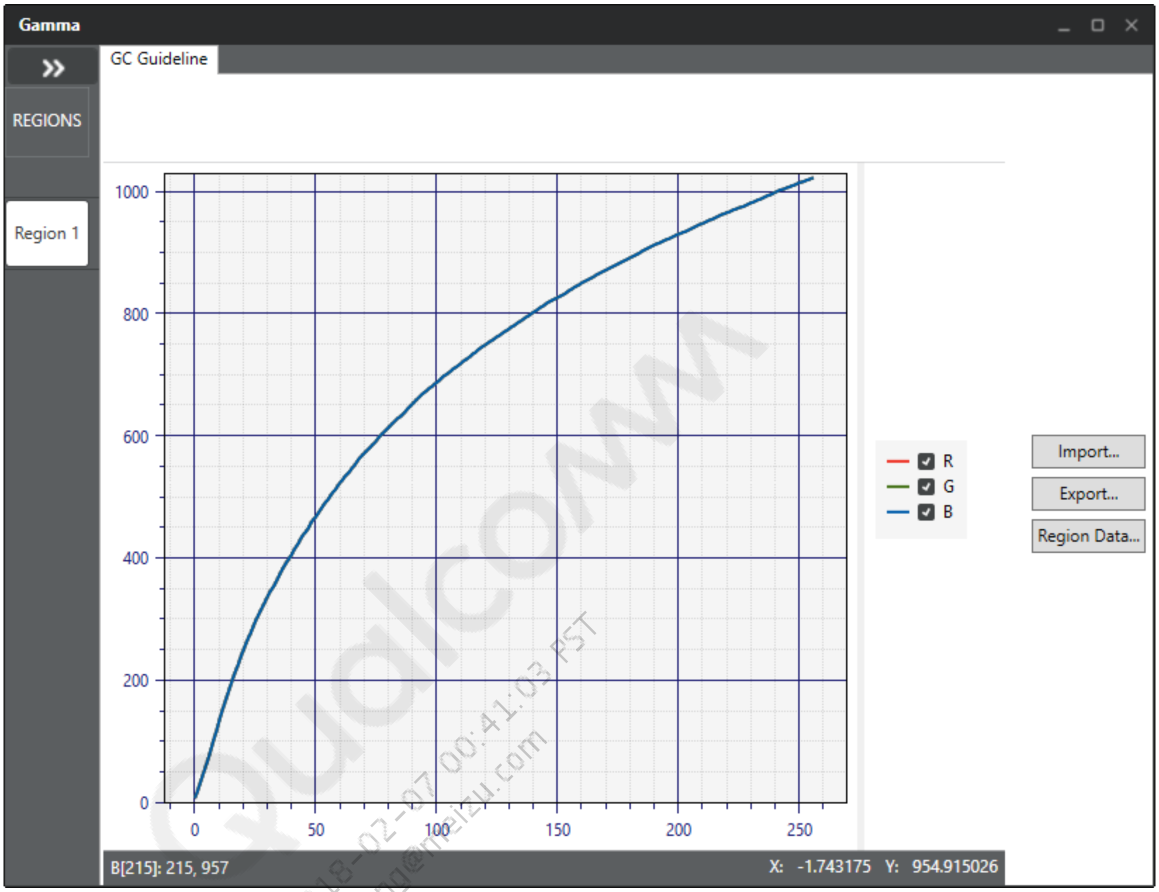
8.4.1 Edit gamma curve

Prepare a gamma curve for each color correction region.

Chromatix provides a gamma curve. Unless you require a custom curve, it is usually adequate to choose the one provided. Use this procedure to view the Chromatix-supplied curve or to create custom curves. Custom curves can be made by editing an existing curve or can be made from an imported gamma table.

1. On the **Pipeline** tab, click **IPE** > **CC** to open the **Color Correction** window
2. At the top of the **Color Correction (CC)** window, click **Edit Gamma**.
3. Click the **Region** to be tuned.

4. Use the R, G, B check boxes to the right of the graph to view the plot of each channel.



5. To export the current gamma curve, do the following:
   a. Click **Export**.
   b. Save the curve data to the preferred location and name the data file. The data is saved in three text files labeled with the specified name followed by R, G, or B.
6. To import a different gamma curve, do the following:
   a. Click **Import**.
   b. Navigate to and select the R, G, B, and RGB text files for the curve. Follow the prompts at the top of the navigation window to select each file in the proper order.
7. To edit the current gamma curve, do the following:
   a. Click **Region Data**.
   b. Edit the gamma curve entries as needed.
8. Open an image with the new gamma table to see the results of the modification.
9. Click **File** > **Save To Project** to save the changes.

Postrequisites:

Whenever you make a change in the gamma curve for a lighting condition, you must re-optimize the CC tuning parameters for the same lighting condition.

RELATED INFORMATION

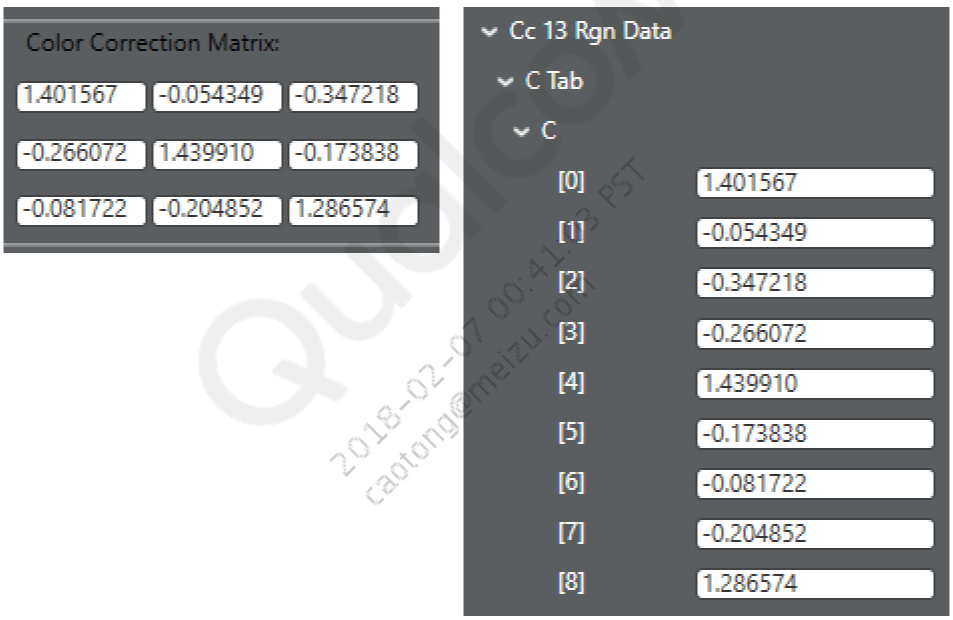
"Gamma tuning concepts" on page 140

8.5 CCM tuning concepts

Due to the characteristics of optics (lens, color filters, etc.) and sensors used in the camera system, the native RGB data may not provide faithful color rendition to the human eye when the image is presented on a particular output media. Therefore, color correction is required.

In modern digital camera designs, color correction is an essential part of the color signal processing chain because it helps the system achieve higher color quality and fidelity. Typically, a 3x3 matrix (in the case of a three-color camera) is used for color correction. Color correction matrix (CCM) tuning minimizes the color reproduction inaccuracies and improves color fidelity.

The color correction parameters in the Color Correction module are in the form of 3x3 matrices.



(L) color correction matrix in CC module window, (R) color correction matrix values in Region Parameters

Color correction optimization fine-tunes the CC matrices to boost color saturation. The diagonal row of values from top left to bottom right are the color saturation parameters for the lighting condition:

- Higher values mean higher color saturation
- Lower values mean lower color saturation

You may want to set up CC matrices to tune for various lighting conditions. Do this by defining Regions defined by triggers. Examples of conditions for which you might need additional matrices include:

- Matrices for different color temperatures (TL84, A, H, D50 and D65)
- Matrices for low-light and outdoor (based on AEC trigger)
- Matrices for LED flash (based on LED Index trigger)

The final CC matrix is first interpolated between the matrices based on color temperature. The calculated matrix is then interpolated according to AEC conditions.

- If the LED is off, the matrix is interpolated between the low-light or outdoor matrix, and is controlled by AEC lighting conditions.
- If the LED is on, the matrix is interpolated between the LED flash matrices, and is controlled by LED trigger points.

Because there can be many CC matrices, you must first determine which matrix to tune. This is done with AWB logs, and A-light or D65 trigger points. These trigger points indicate which matrix dominates in current conditions. If the flash LED is on, tune the LED flash CC matrix first.

The AEC logs show gain and lux index information, and the low-light trigger points determine whether the low-light CC matrix is being used. However, whether the outdoor CC matrix is used is determined by the outdoor trigger points in gamma.

You can obtain the A-light and D65 trigger points, and low-light and outdoor trigger points from the embedded 3A debug information in JPEG images. To confirm the gain and lux index, use the AEC debug log and search for the keywords luxidx and aec_process_pack_out. To find the color temperature, look in the AWB debug log message for the keyword **Bayer AWB CCT**.

RELATED INFORMATION

"Tune CC" on page 146

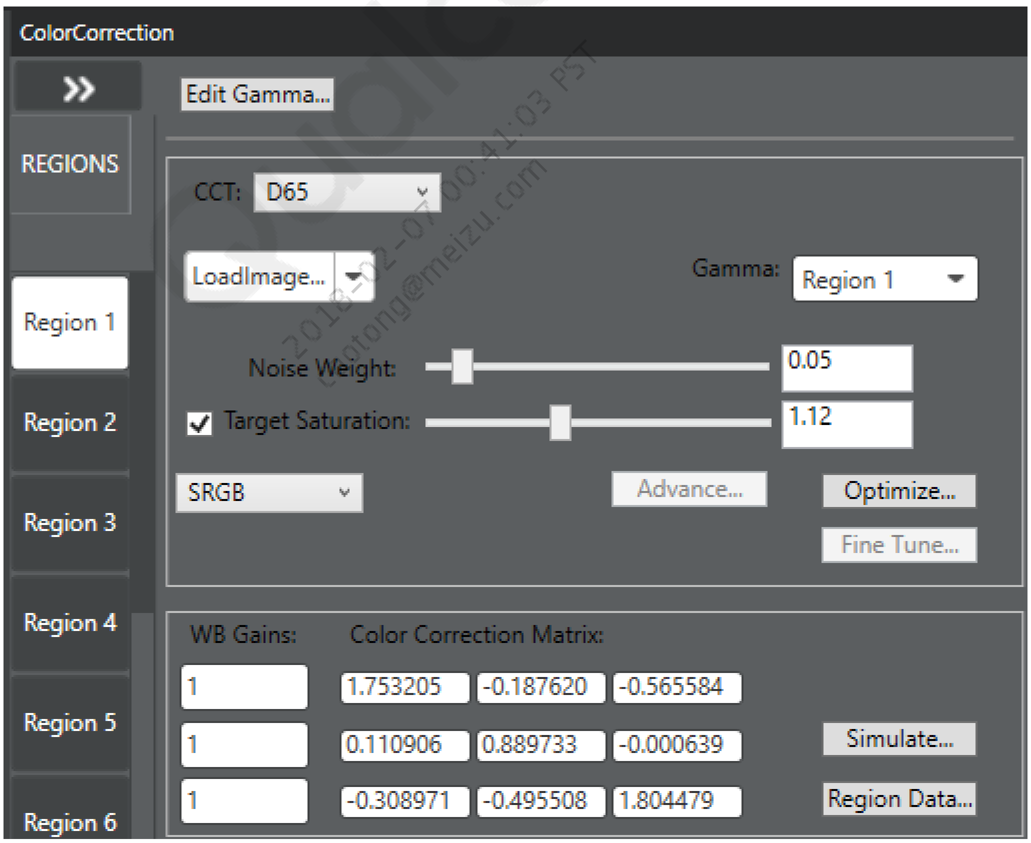
8.5.1 Tune CC

Use the **Color Correction (CC)** module to minimize color reproduction inaccuracies and improve color fidelity.

Prerequisites:

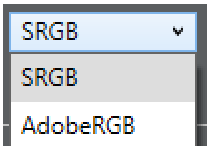
- Set up Regions in the CC section of the **Parameter Editor** to support the various CC matrices you need to tune. For example:
  - Regions can be defined using the AEC Trigger to segregate tuning for outdoor conditions vs indoor conditions.
  - A region can be defined by the LED Index Trigger to allow for specific CC tuning for LED flash.
- The gamma curve must be accurate prior to tuning the color correction matrix (CCM). Any change to the gamma curve requires retuning the CC module.

1. On the **Pipeline** tab, click **IPE** > **CC** to open the **Color Correction** window.

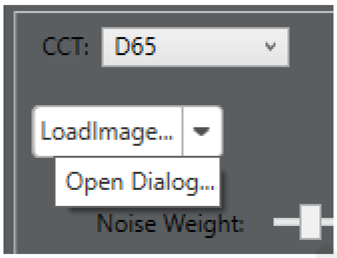


2. If it is necessary to modify the standard gamma curve or to create a new gamma curve, click **Edit Gamma** and make the changes before tuning the color correction matrix.
3. In the **Regions** panel, click the region that you want to tune.
4. In the **CCT** field, select the color temperature you want to tune.
5. In the **Gamma** field, select the applicable gamma curve. The gamma curve must be accurate before continuing with the CCM optimization.

6. Color correction uses the SRBG color transform unless you select AdobeRGB color transform from the list box.



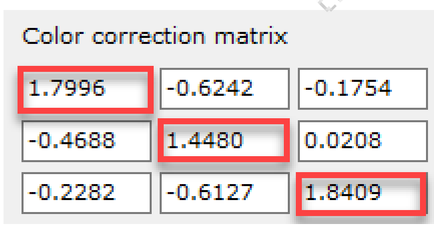
7. Load an MCC chart image that was captured under the appropriate lighting condition.
   a. Click **Load Image** to load an image previously saved to the **Images** panel, or click **Open Dialog** to load a new image.



   b. When the MCC image opens, select **Mark** > **Macbeth** > **Macbeth** and set the grid markers. Drag the top-left MCC grid marker to the center of the top-left color patch and drag the bottom-right MCC grid marker to the center of the bottom-right color patch so that all of the grid markers align within the color patches of the MCC chart image. Close the window.

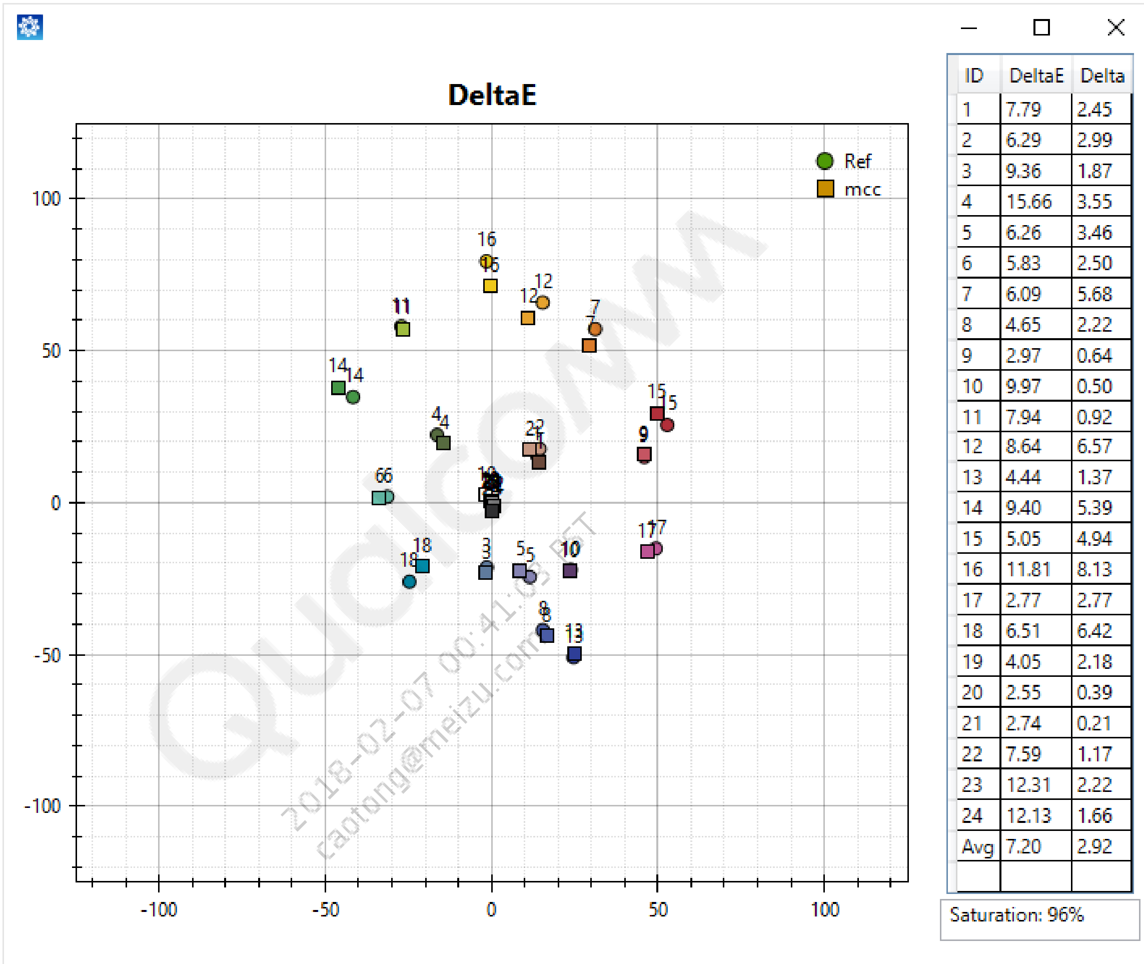
      If the image is taken with a fisheye lens, to set the MCC markers for tuning, hold the Ctrl Key (or Shift Key) while moving the markers

8. On the CC window, click **Optimize**.
9. Examine the three diagonal values of the color correction matrix to see how they changed as a result of the optimization..



For tuning purposes, only the highest value of the three values matters. In this example, the highest value is the bottom-right value. A higher noise weight value produces lower diagonal parameters in the color correction matrix. For the diagonal parameters, higher values indicate higher color saturation and increased noise. The ideal range for the highest diagonal value is 1.8 to 3.

10. Open the image again and select > **DeltaE**. When the Illuminant Selection dialog opens, select the appropriate illuminant. The DeltaE plot opens and shows the changes made by the optimization.



ID	DeltaE	Delta
1	7.79	2.45
2	6.29	2.99
3	9.36	1.87
4	15.66	3.55
5	6.26	3.46
6	5.83	2.50
7	6.09	5.68
8	4.65	2.22
9	2.97	0.64
10	9.97	0.50
11	7.94	0.92
12	8.64	6.57
13	4.44	1.37
14	9.40	5.39
15	5.05	4.94
16	11.81	8.13
17	2.77	2.77
18	6.51	6.42
19	4.05	2.18
20	2.55	0.39
21	2.74	0.21
22	7.59	1.17
23	12.31	2.22
24	12.13	1.66
Avg	7.20	2.92

11. If adjustments are necessary, drag the **Noise weight** slider to change the balance between color reproduction and noise reduction.

    The noise weight setting determines how much significance is given to noise. If the slider is set to 0, color reproduction is given full priority, with no consideration or weight given to noise. Moving the slider to the right minimizes noise, but sacrifices color reproduction.

12. If saturation changes are needed, mark the **Target saturation** check box and drag the slider to change the saturation level.

    The saturation level is not normally adjusted, but use this method to increase saturation resulting from an increase in noise weight.

13. Click **Optimize**. Continue adjusting noise weight and saturation, and clicking **Optimize** until the highest diagonal value in the CCM is in the ideal range.

14. Click **Simulate**. A simulated image appears. If the simulated image is acceptable, continue to the next step. If not, repeat the preceding steps.

15. Click **File** > **Save To Project** to save the settings for this Region/CCT.

16. Repeat this process for each remaining **CCT** in this Region.
17. Repeat this process for each **Region**.

RELATED INFORMATION

"CCM tuning concepts" on page 144

8.6 2D LUT tuning concepts

2D LUT tuning helps render colors accurately and pass color fidelity tests. It can also adjust specific colors, such as blue skies or green grass, without affecting other colors. The design allows easy enhancement of most colors. 2D LUT helps to control color dynamically from point-to-point for hue and saturation. All hue and saturation values can be changed, though only some hue and saturation values can change without affecting other colors. 2D LUT uses hue, saturation, and luminance to enhance colors, where hue and saturation can be adjusted with the lightness (L) value.

The Y offset adjusts brightness and is calculated and applied to the final output. A value of zero is recommended to maintain original colors and brightness.

Y offset = Yin (R, G, B) - Yout (R', G', B')

An image with a sky or grass background is the best test example to use for 2D LUT tuning. In low-light and indoor conditions, ensure chroma noise is not added.

2D LUT replaces MCE and SCE.

The following example has a +20% saturation for green colors.

Using hue and saturation can adjust blue sky images. The following example has a hue of +10 degrees and +15% saturation.

By adjusting specific L values, images can be adjusted to allow only changes to blue sky, or changes to all blue colors.

Coordination with other modules

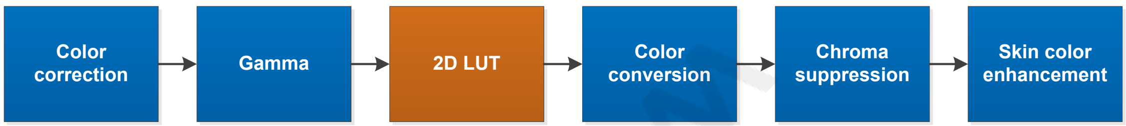
Module	Description
AEC (luma target)*	Tune to match target phone.
AWB*	Tune reasonably well.
Chroma suppression*	It is recommended to disable this module.
Gamma and CCM	Perform global tone and global color assessment.
2D LUT	■ Color conversion (CV) – Use default ■ Memory color enhancement (MCE) – Disable ■ Skin color enhancement (SCE) – Disable
*Perform prior to color tuning.	

8.6.1 Tune 2D LUT

2D LUT adjusts specific colors without affecting others.

Prerequisites:

Tune color correction and gamma.



1. On the **Pipeline** tab, click **2DLut** and then click a region in the Regions group.
2. On the **Main** tab, click **LoadImage** and select an image to tune.
3. On the **Main** tab, in the color wheel, drag knee points to the perferred color (e.g., drag a green knee point to a brighter shade of green) and the hue and saturation will be adjusted accordingly in the **Saturation Lut** and **Hue Lut** tabs. White lines denote the original RGB colors and the black lines denote the target RGB colors.

   Adjust the **Hue Nodes** and **Saturation Nodes** values in the Impacted Range group to adjust how many neighboring knee points are affected during adjustments. If a larger color adjustment is being made, increase the impacted range so the changes are smoother.
4. (Optional) Click the **1-D Lut** tab and set the knee points for `lut_1d_h[25]` and `lut_1d_s[16]` for 1D colors that need adjustments.

   `lut_1d_h[25]` represents the hue angle and `lut_1d_s[16]` represents the saturation. LUTs are always programmed in ascending order, and no two entries can be equal.
5. Obtain the hue and saturation LUT (HSL) values for the original colors and target colors.
   a. On the **Main** tab, set the values for the **Original RGB** and **Target RBG** fields.
   b. Set the 2D delta hue and saturation LUTs. To preserve the gray areas, hue and saturation should not change when the saturation is zero.
   c. Click **Adjust Lut** to adjust the color wheel knee points to match the original and target RGB values.

      The **Saturation Lut** and **Hue Lut** tabs are automatically updated. To preserve the gray areas, hue and saturation should not change when the saturation is zero.
6. (Optional) Click the **Boundary** tab and enter values to set the L boundary points to maintain adjustments within a specified brightness range.

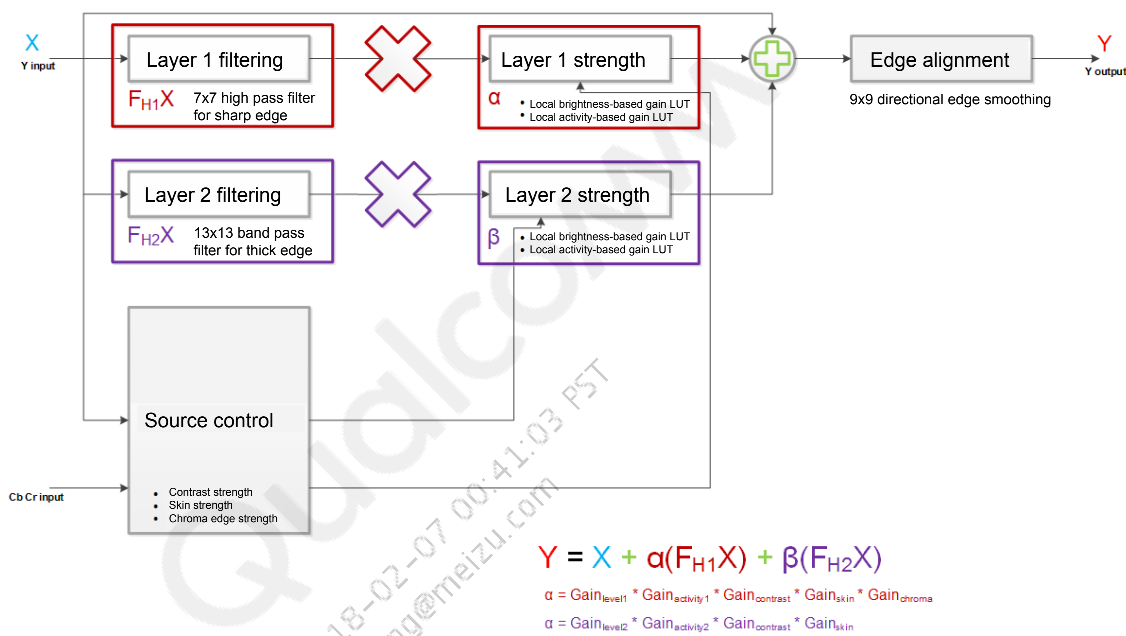
   By default, **L start A** and **L start B**= 0, and **L end A** and **L end B** = 1.0.
7. (Optional) Set the **Y Offset Weight** value for the preferred luma preservation and color enhancement.

   The **Y Offset Weight** range is 0 to 1.0, where the default value is 0. A value of 0 is recommended for best colors. When the value is 1, the Y value is best preserved.
8. Repeat process for each region.
9. Click **File** > **Save to Project** to save any changes made.

8.7 ASF tuning concepts

Edge enhancement increases the local contrast at boundaries which makes the image appear sharper. The ASF block in the CPP hardware uses high-pass filters to enhance the edges. Smoothing can be applied to nontexture regions to suppress noise.

The ASF module is located in PPS of IPE and should be tuned last.



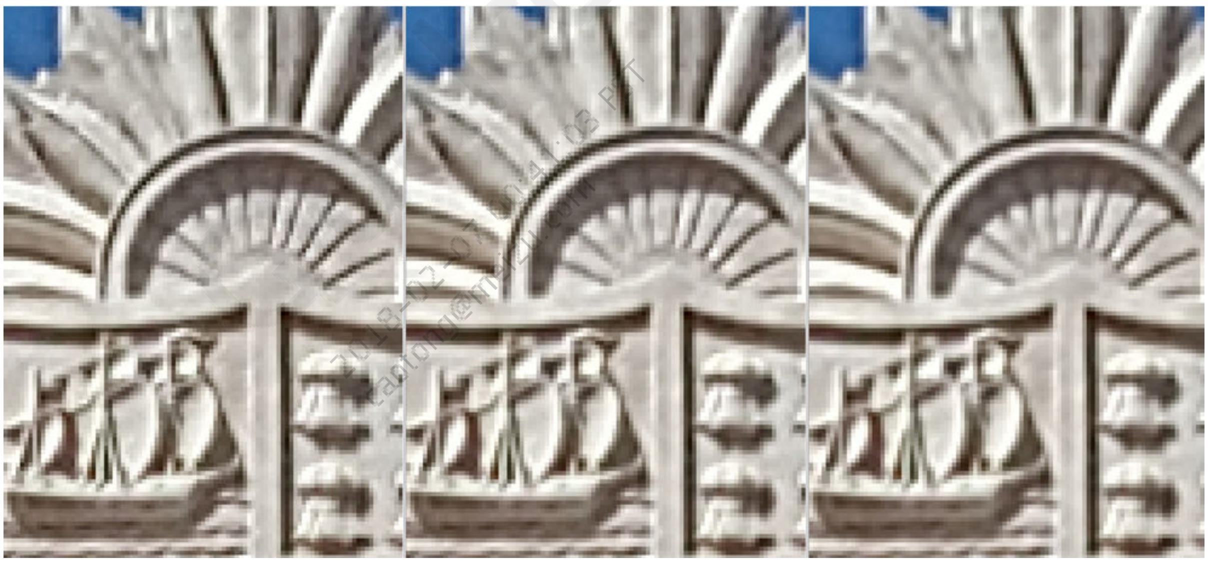
Sharpening gain is calculated by combining sharpening strength based on local brightness and local activity. Sharpening gain is controlled by three LUTs per layer: `layer_1_gain_positive_lut`, `layer_1_gain_negative_lut`, and `layer_1_gain_weight_lut`. For layer 2, the LUTs are `layer_2_gain_positive_lut`, `layer_2_gain_negative_lut`, and `layer_2_gain_weight_lut`.

Strength reduction

Strength reduction is controlled by four parameters: `gain_contrast_negative`, `gain_contrast_positive`, `skingain`, and `skinactivity`. These parameters adjust the contrast strength, skin strength, and chroma edge strength.

Symmetric kernel

Parameter name	Description	Tuning	Default	Must tune?
layer_1_hpf_symmetric_coeff	This controls the first layer 7x7 sharpening coefficients. The thin kernel shows small details and makes the image sharper. The thick kernel smooths edges. The sum of the kernel should be zero. It is not recommended to change the kernel manually. Length: 10; Q10		▪ Thin kernel (for outdoor): 0, 0, 0, -1, -2, -25, -86, -173, -180, 1968 ▪ Mid kernel (for indoor): 0, 0, -3, -5, -10, -57, -106, -132, 65, 1232 ▪ Thick kernel (for low light): 0, -2, -9, -15, -22, -67, -89, -46, 144, 736	Use defaults



(L) Thin kernel; (C) Mid kernel; (R) Thick kernel

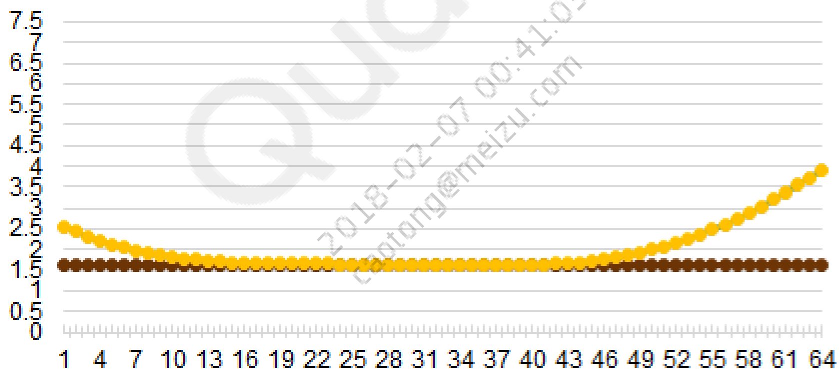
gain_positive/negative_lut

Thses LUTs affect sharpness in the output image where the positive gain LUT affects white or lighter edges and the negative gain LUT affects black or darker edges. Larger values increase sharpness in busy areas of the image. Positive and negative gain LUTs are calculated by:

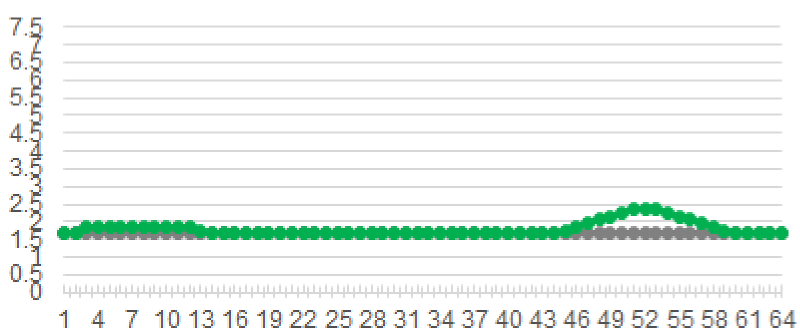
Level = Y intensity value/4

Parameter name	Description	Tuning	Default	Must tune?
layer_1_gain_positive_lut[level]	Level-based sharpening gain LUT for positive halo Length: 64	Higher value = stronger sharpening	All 1.65f Min: 0.0f; Max: 7.9f	Use defaults
layer_1_gain_negative_lut[level]	Level-based sharpening gain LUT for negative halo Length: 64	Higher value = stronger sharpening	All 1.65f Min: 0.0f; Max: 7.9f	Use defaults
layer_1_gain_weight_lut[activity]	Normalized activity-based sharpening gain LUT Length: 64	Higher value = stronger sharpening	From 0.5 to 1.0 Min: 0.0f; Max: 0.996f	Use defaults

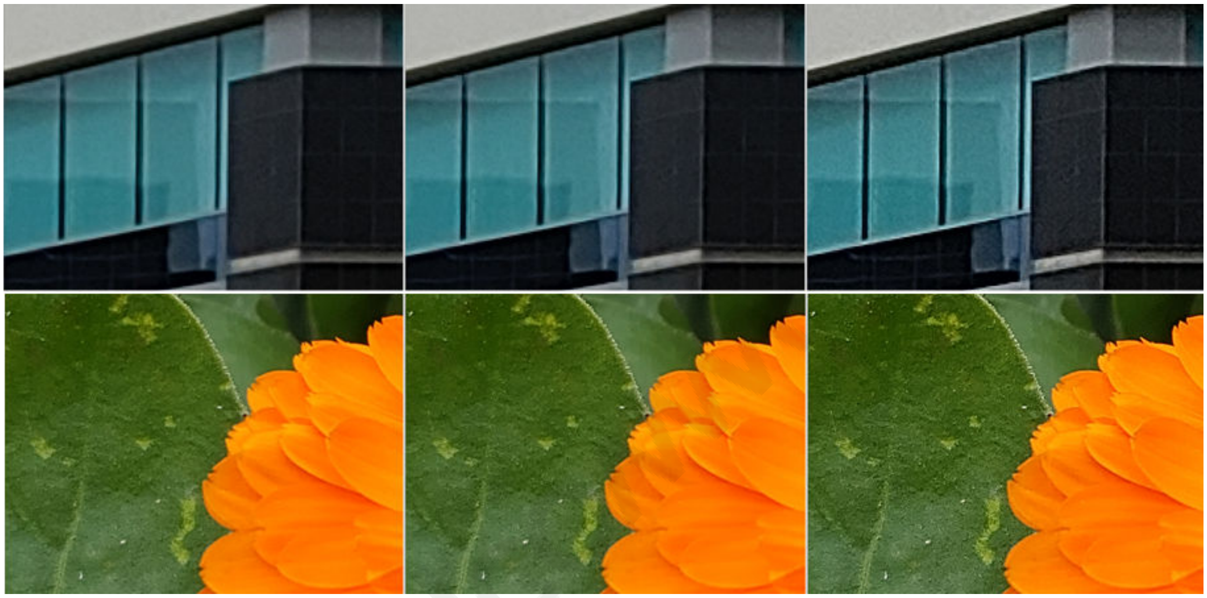
For bright or normal light, there is a U-shaped curve with a stronger enhancement for highlights and shadows (see orange line below). For low light, there is a flat curve with the same enhancement for all intensity levels and the default is all 0.6f (see brown line below).



For manual curve tuning, obtain the level data (Y intensity) for the section that requires enhancement, or release sharpening and tune the level manually.



Increasing gain LUTs increases sharpness but also increases noise.



(L) Flat: 1.00f; (C) Flat: 1.65f; (R) Flat: 3.0f

U-shape generates stronger highlight sharpening and manual shape sharpening is similar.

(L) Flat: 1.00f; (C) Flat: 1.65f; (R) U-shape: 1.65f

(L) Flat: 1.00f; (C) Flat: 1.65f; (R) Manual shape: 1.65f

activity_normalization_lut

This LUT is tuned to reduce noise in dark areas of the image. Activity is calculated by:

Activity = Local variation/4

The activity is normalized 5x5 BPF and LPF input. It can be adjusted with the following parameters.

Change to perform	Parameter to adjust
BPF	layer_1_activity_band_pass_coeff
Scale	layer_1_norm_scale and layer_1_L2_norm_en
Clamp	layer_1_activity_clamp_threshold
Normalization	layer_1_activity_normalization_lut

The following example illustrates the effects of changing the LUT to decrease scaling on darker pixels. The dark areas indicate low levels of activity and the light areas indicate high levels of activity.

(L) Flat: 1.00f; (C) Bright/normal setting; (R) Lowlight setting

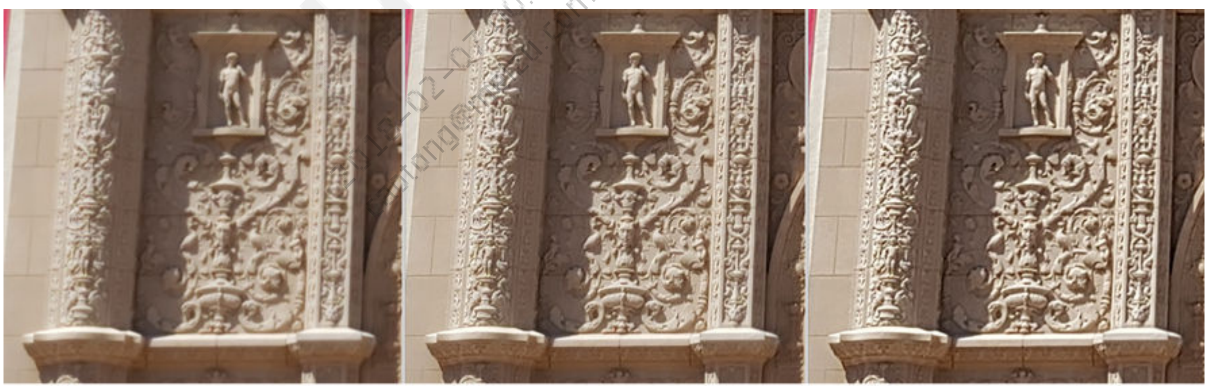
Using smaller normalization values decreases the normalized activity factor, which in turn provides dark noise suppression but also creates less contrast on dark edge areas. Avoid using values close to zero for this LUT.

Layer 2 kernel and LUTs

Parameter name	Description	Tuning	Default	Must tune?
Layer 2 kernel				
layer_2_hpf_sharpening_coeff	This parameter controls the second layer sharpening coefficients. The sum of the kernel should be zero. It is not recommended to change the kernel manually. Length: 6; Q10	5x5 kernel on x2 downscaled image and 2x upsampled	-1, -4, -6, -16, -24, 220	Use defaults
Layer 2 LUTs				

Parameter name	Description	Tuning	Default	Must tune?
layer_2_gain_positive_lut[level]	Level-based sharpening gain LUT for positive halo. This can be the same as layer 1. Length: 64	Higher value = stronger sharpening	All 1.65f Min: 0.0f; Max: 7.9f	Use defaults
layer_2_gain_negative_lut[level]	Level-based sharpening gain LUT for negative halo. This can be the same as layer 1. Length: 64	Higher value = stronger sharpening	All 1.65f Min: 0.0f; Max: 7.9f	Use defaults
layer_2_gain_weight_lut[activity]	Normalized activity-based sharpening gain LUT. This can be the same as layer 1. Length: 64	Higher value = stronger sharpening	All 1.65f Min: 0.0f; Max: 0.996f	Use defaults

In the following example, layer 1 and layer 2 gain LUT parameters are the same, but the effects are different.



(L) ASF input; (C) Layer 1 output; (R) Layer 1 and layer 2 output

(L) Layer 1 only; (C) Layer 2 only; (R) Layer 1 and layer 2

Gain and skin LUTs

Parameter name	Description	Tuning	Default	Must tune?
gain_contrast_positive[contrast]	Contrast-based sharpening gain LUT. The strength is based on contrast and a value of 1 bypasses the effects. Length: 32	Higher value = stronger sharpening	From 1.0 to 0.25 Min: 0.004f; Max: 1.0f	Use defaults
gain_contrast negative[contrast]	Contrast-based sharpening gain LUT. The strength is based on contrast and a value of 1 bypasses the effects. Length: 32	Higher value = stronger sharpening	From 1.0 to 0.25 Min: 0.004f; Max: 1.0f	Use defaults
skingain	Skin color based sharpening gain LUT. Length: 17	Higher value = stronger sharpening	From 1.0 to 0.76 Min: 0.004f; Max: 1.0f	Use defaults
skinactivity	Skin color based sharpening gain activity LUT. Length: 17	Higher value = stronger sharpening	From 1.0 to 0.5 Min: 0.004f; Max: 1.0f	Use defaults

The contrast value for gain_contrast_positive is calculated using:

Contrast = Y min max diff/4

Define the areas that may have halos and reduce the contrast strength to remove them. Typically, a positive halo appears worse than a negative halo.

For skin gain, define the skin color and face area and reduce the gain strength on the face. Face areas can be smoothed with FD information, which is set via the face_boundary and face_transition parameters.

For skin activity, the activity strength is based on skin color. Define the skin color and face area and reduce the activity data on the face. Face areas can be smoothed with FD information, which is set via the face_boundary and face_transition parameters.

(L) All 1.0 (skin color control off); (C) Skin color control on; (R) Skin color control and FD data on

Smoothing

Parameter name	Description	Tuning	Default	Must tune?
smoothing_strength	The blending factor between original pixel and the edge-smoothed pixel. Length: 1	Higher value = stronger sharpening	0.5f Min: 0.0f; Max: 0.999f	Use defaults

If the output image edges are not clean or have a checkered-like effect, use the smoothing_strength parameter to blend the edges of pixels. The edge is smoothed by direction (horizontal, diagonal, ect.). If an edge is not detected by other parameters, this parameter will not work.

The following images show coarse edges that are aligned without reducing the texture. Details are not affected by edge alignment.

(L) Edge alignment off; (C) Edge alignment; strength 1.0; (R) Edge alignment; strength 0.6

Clamps

Clamping parameters help correct the overshooting that can occur in edge sharpening.

Parameter name	Description	Tuning	Default	Must tune?
layer_1_clamp_UL layer_2_clamp_UL	Manual fixed positive clamping-level sharpening Length: 1	Higher value = stronger sharpening	255 Min: 0; Max: 255	Yes
layer_1_clamp_LL layer_2_clamp_LL	Manual fixed negative clamping-level sharpening Length: 1	Higher value = weaker sharpening	-255 Min: -255; Max: 0	Yes

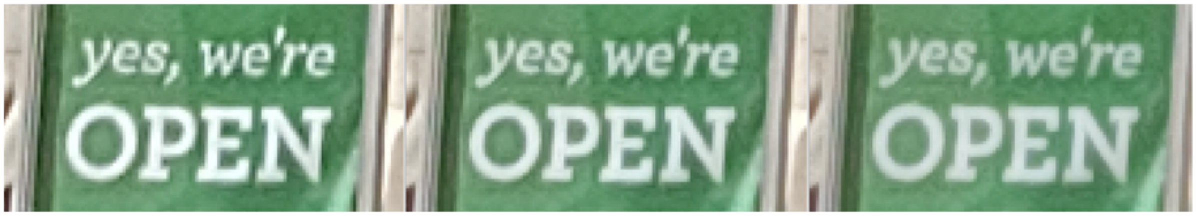
It is recommended to use halo control parameters over clamps where possible.

Tighter clamps, values with a smaller absolute value, should be used in the final steps of ASF tuning. These tighter clamp values can reduce halo effects, which typically occur in normal to low lighting conditions.

The following example shows the effects of two different clamp values.

Layer_1_clamp_UL

255 16 8



Layer_1_clamp_LL

255 16 8

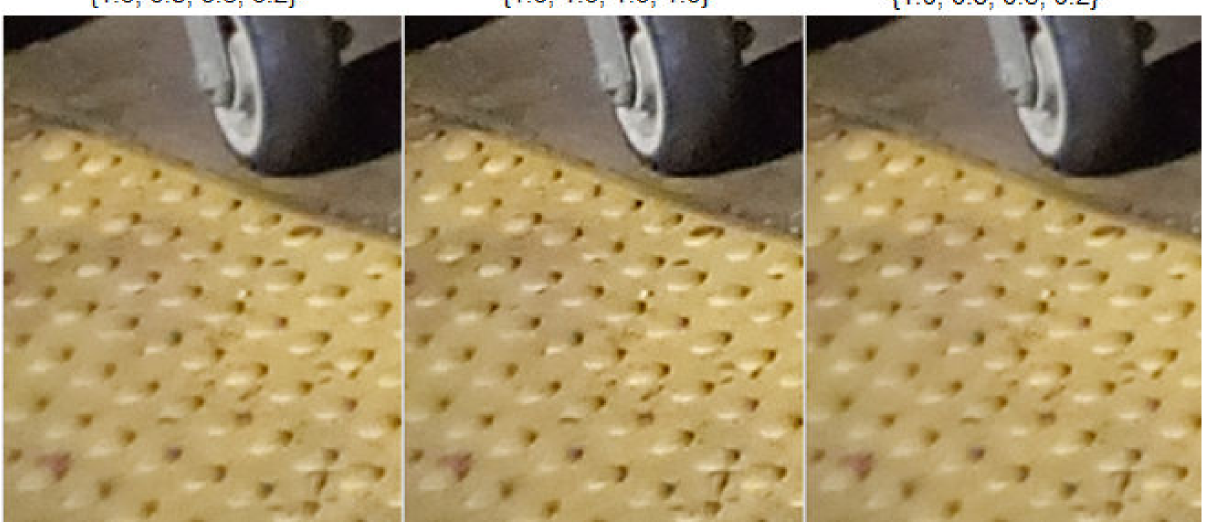
Radial adjustments

Parameter name	Description	Tuning	Default	Must tune?
radial_activity_adj	The correction factor for activity based on radial distance. Length: 4	Higher value = stronger sharpening	From 1.0 to 0.6 Min: 0.0f; Max: 7.96f	Yes
radial_gain_adj	The correction factor for gain based on radial distance. Length: 4	Higher value = weaker sharpening	From 1.0 to 1.0 Min: 0.0f; Max: 7.96f	Yes

Apply the radial_activity_adj parameter to activity and the radial_gain_adj parameter to gain. Both parameters can be increased to enhance corner edges.

(L) Corner: all 1.0; (R) Corner: both activity/gain

To preserve image corner details, use the radial_activity_adj parameter for radial processing.



(L) Gain adjust only; (C) Activity adjust only; (R) Gain and activity adjust

Chroma edge

Parameter name	Description	Tuning	Default	Must tune?
gain_chroma_positive[chroma_halo] gain_chroma_negative[chroma_halo]	Contrast-based sharpening gain LUT Length: 32	Higher value = stronger sharpening	From 1.0 to 0.25 Min: 0.004f; Max: 1.00f	Use defaults, but can be tuned to preference

The gain_chroma_positive/negative parameters control strength based on the chroma edge. Define the area where a halo may occur around chroma and reduce the strength around the chroma edge when there is no contrast (between chroma to chroma). These parameters are primarily for positive halo control surrounding chroma.

(L) All 1.0 (chroma halo control off); (C) Curve 1 for positive/negative; (R) Curve 1 for positive, Curve 2 for negative

Edge alignment

Edge alignment can be adjusted with the `smothing_strength` parameter.

Parameter name	Description	Tuning	Default	Must tune?
flat_threshold	Apply edge smoothing only when the maximum of each edge value is larger or equal to the flat_threshold parameter value. Increasing this threshold causes edges to not be detected so edge smoothing will not work. Length: 1	Higher value = less edge detection and less smoothing	8 Min: 0; Max: 255	Use defaults
similarity_threshold (corner_threshold)	The edge smoothing value is applied only when the maximum value is greater or equal to the minimum value * similarity_threshold. Length: 1	Higher value = less edge detection and less smoothing	2 Min: 0; Max: 63.999f	Use defaults
texture_threshold (max_smoothing_clamp)	Gray colors are not smoothed as they are considered texture. The edge smoothing value change is clamped by texture_threshold. Length: 1	Higher value = stronger smoothing	8 Min: 0; Max: 255	Use defaults

Parameters

Parameter	If disabled
layer_1_enable	1st layer is disabled
layer_2_enable	2nd layer is disabled
radial_enable	No gain applied for radial location
contrast_enable	No gain applied for contrast
chroma_gradient_enable	No gain applied for chroma
skin_enable	No gain applied for skin
face_enable	No face detection data used for skin
edge_alignment_enable	No edges are smoothed

To bypass ASF using parameters:

1. Set layer_1_clamp_UL/layer_2_clamp_UL to 0.
2. Set layer_1_clamp_LL/layer_2_clamp_LL to 0.
3. Set layer_1_sp to 0.
4. Set smoothing_strength to 0.

8.7.1 Tune ASF

ASF is an adaptive filter that can apply smoothing and sharpening to different parts in images, depending on the detection of edges. It smooths areas with less detail to remove noise and sharpens edges to increase the sharpness of the image.

1. On the **Pipeline** tab, click **ASF** and then click a region in the Regions group.
2. Click **Load Image** and select a resolution chart image that was captured under the specified lighting condition.
3. Click **Region Data** and expand the **Asf 30 Rgn Data** navigation tree.
   a. Set layer 1 symmetric kernel parameters based on light conditions.
   b. Tune layer 1 gain strenght based on level and activity to enhance steep edges and textures.
   c. Set layer 2 kernel as the default.
   d. Tune layer 2 gain strength based on level and activity to enhance thick edges and local edge enhancements.
   e. If haloing occurs, tune gain contrast to reduce gain strength around the edges.
   f. If skin appears dirty, tune skin gain to reduce sharpening on skin.
   g. Tune smoothing strength to make edges smoother.
   h. Tune clamp if haloing persists after tuning gain contrast.
4. Click **Simulate**.

Kernel

5. On the **Kernel** tab, select **HPF Symmetric Coeff**, **LPF Symmetric Coeff**, or **Activity Band Pass Coeff** from the list and adjust the values as needed.

LUT

6. Click **LUT**.
   a. On the **Gain LUT** tab, use the three sliders to adjust **Overall Gain**, **Shadow Boost**, and **Highlight Boost**.
   b. On the **Negative Gain** tab, select adjust the **Negative Gain**, **Negative Shadow**, and **Negative Highlight** sliders to change gain LUT scaling.
   c. On the **Detail** tab, adjust the **Detail**, **Noise**, and **Halo** sliders as needed.

      If haloing occurs around chroma edges when contrast is not high, tune gain chroma. If edge alignment is unnatural for textures, tune the thresholds for edge alighnment not to detect textures as edges.
   d. If image corners are noisy or have less details, adjust the **Radial Gain** and **Radial Activity** sliders on the **Radial** tab.

7. Click **Sigma LUT RNR** to adjust the RNR control points.

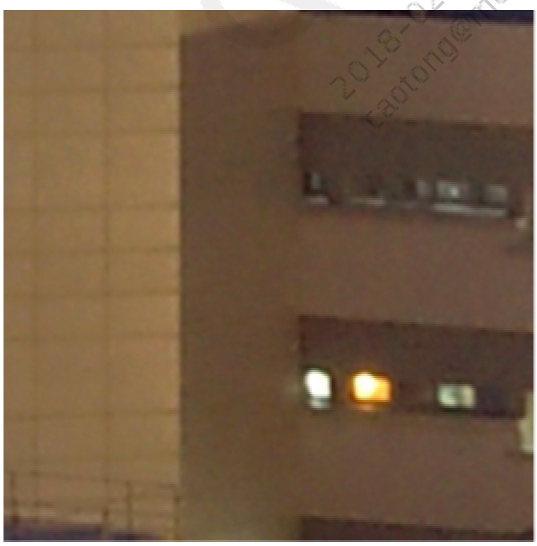
   The radial control circles split the image into four knee points (Radial Anchor[0] though Radial Anchor[3]), where each point has a different noise reduction value. Moving the slider right increases the circle radius and moves the location of the noise.

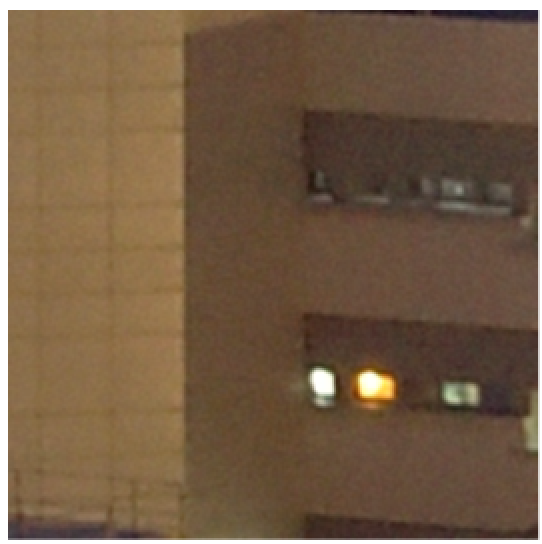
To confirm the module performs as expected:

8. Capture a TE42 image.
9. Verify the following:
   a. Flat area has no dots or artifacts from this module.
   b. Details are visible and look acceptable.
   c. Edges are strong.
   d. Corners are not noisy or less sharpened from the rest of the image.
   e. Halos are not present.
   f. Skin appears smooth.
10. Repeat process for each region.
11. Click **File** > **Save to Project** to save any changes made.

8.8 GRA tuning concepts

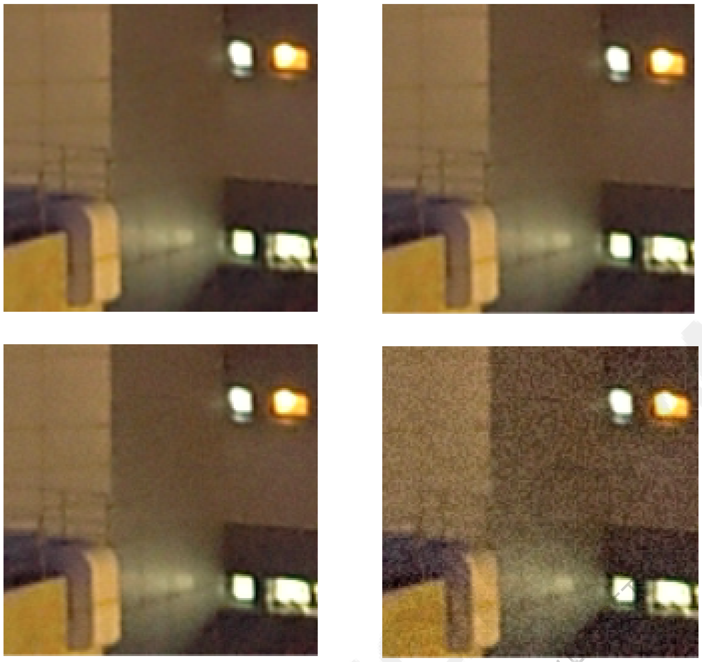
The grain adder (GRA) module controls the amount of fine grain added to an image. The default settings are adequate in most cases, so tuning the GRA module is at the descretion of the user.





(L) GRA off; (R) GRA on

If changes are made to noise reduction modules such as ANR, HNR, and TF, there may be a need to adjust the **Grain Strength** parameter.



(Top L) GRA off; (Top R) Low GRA strength; (Bottom L) Medium GRA strength; (Bottom R) High GRA strength

The GRA module includes three LUTs that affect luma **(Y Weight LUT**) and chroma (**CB Weight LUT**, **CR Weight LUT**). For example, use the chroma LUTs to apply less grain for skin (Cr) or sky (Cb) colors. If tuned, it should be the last module tuned in the pipeline.

RELATED INFORMATION

“GRA parameters” on page 168
“Tune GRA” on page 169

8.8.1 GRA parameters

The following table lists the parameters used for tuning the GRA module. The default values for the parameters are suitable for most situations but may require adjustment in some cases to achieve your specific objectives.

Table 8-1 Basic GRA tuning parameters

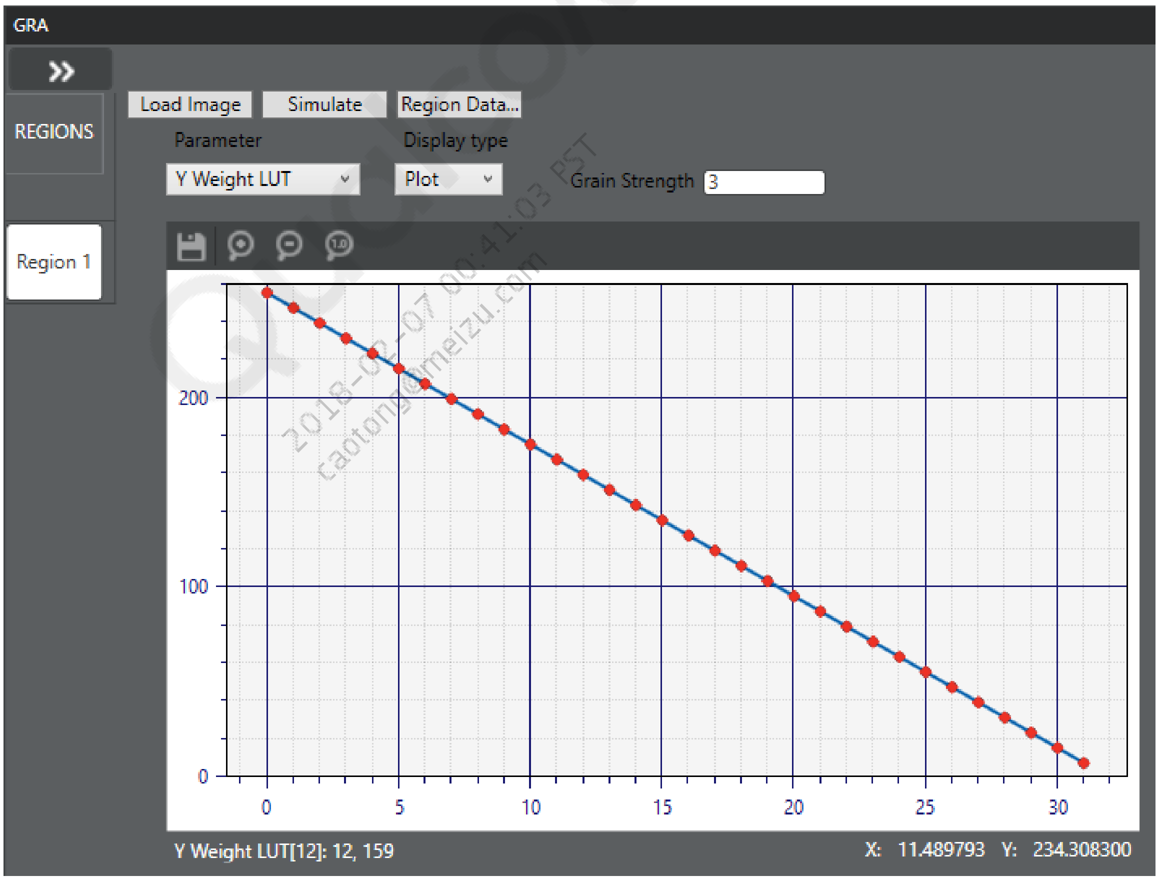
Parameter name	Description	Tuning	Factory default	Tuning required
Grain Strength	Number of value bits to generate for grain. For example, for value of 3, the grain noise added is in the range from -7.	Adjust as needed to control grain. Higher value = more grain Lower value = less grain	3	Use default value
Y Weight LUT	32-value lookup table for Y (luma) grain weight. Changes grain strength according to pixel luminance value.	Higher value = more grain	Range: 0-255	Use default value
Cb Weight LUT	32-value lookup table for Cb (chroma) grain weight Changes grain strength according to pixel luminance value.	Higher value = more grain	255 Range: 0-255	Use default value
Cr Weight LUT	32-value ll ookup table for Cr (chroma) grain weight Changes grain strength according to pixel luminance value.	Higher value = more grain	255 Range: 0-255	Use default value

8.8.2 Tune GRA

Prerequisites:

Before tuning the GRA module, determine how many lighting regions you need and create them in the Parameter Editor.

1. On the **Pipeline** tab, click **IPE** > **GRA**.
2. Select the **Region** you want to tune.
3. Click **Load Image** and upload a raw image of a scene to evaluate the grain.
4. Evaluate the image and the three GRA lookup tables to determine if any adjustments are necessary. Use the **Parameter** list to select the appropriate LUT and select Plot or Grid as the **Display type**.



5. To adjust the strength, enter a new value in the **Grain Strength** field.
6. To adjust LUT values, select the appropriate LUT, and do one of the following:
   - If using **Display type** = Plot, drag the appropriate point to the adjusted value.
   - If using **Display type** = Grid, enter a new value in the appropriate field.
7. Click **File** > **Save to project** to save the changes.
8. Upload the image again to evaluate the effect of the changes.

RELATED INFORMATION

"GRA tuning concepts" on page 166

A Installing and Working with Chromatix

A.1 System requirements

- OS – 64-bit version of Windows 10
- CPU – 64-bit Intel processor, preferably minimum of 2 GHz with multiple cores
- Memory – Minimum 16 GB of RAM
- Monitor – Minimum resolution of 1024x768, preferably color-calibrated

A.2 Install Chromatix

Prerequisites:

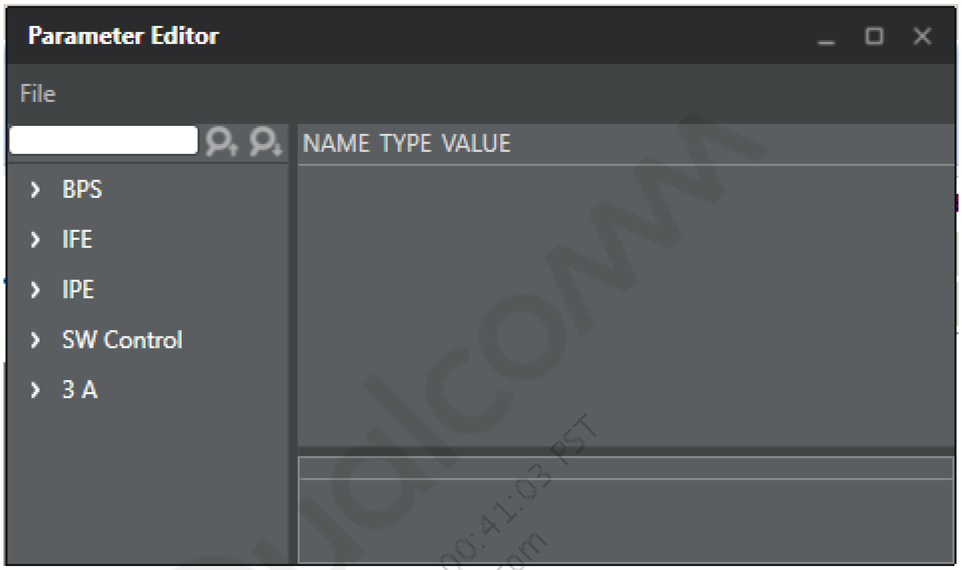
If a previous version of Chromatix 7 is on the computer, uninstall the previous version.

1. Download the Chromatix installer from the QTI CreatePoint tool.
2. Double-click the Chromatix7Setup.exe file to launch the installer.
3. Read the license terms and conditions. Click **I accept the terms of the license agreement**.
4. Click **Install**. The installer saves the files to C:/Qualcomm/Chromatix7.
5. Double-click the Chromatix.exe file to launch the application.

A.3 Edit Chromatix parameter data

Parameter Editor dialog box.Parameters that are not accessible in ISP tuning menus can be edited directly in the

1. With the tuning project open in Chromatix, click **Utilities**, then click **Parameter Editor**.



2. Expand the applicable pipeline from the list (e.g., BPS, IFE, IPE, or 3A) in the parameter tree along the left-hand side of the dialog box.
3. Expand the applicable module tree to edit parameters.

Common Navigation	Controls
Expand navigation tree	Click top-most tree to expand + E
Collapse navigation tree	Click top-most tree to collapse + C

4. Click a specific editable field to make adjustments.

   Changes are saved as edits are made.

A.4 Using the Chromatix Tools menu

Add user files to the Tools menu

Use this procedure to add any type of file (text, spreadsheet, folder, or shortcut) to the user-defined tools section of the **Tools** menu. Files you frequently access when tuning may be useful to add to the menu.

1. From the Chromatix main menu, select **Tools** > **Edit**. This action opens an Explorer window to the user-defined tools directory.
2. Add a file of your choice to the tools directory, then close the Explorer window. The file name is added to the tools menu for easy access.

B Capturing Images for Camera Tuning

B.1 Test charts and equipment required for capturing images

Table B-1 Test charts for capturing images

Chart or equipment	Applicable tuning procedures
 Macbeth color checker (MCC)	▪ CCM ▪ Linearization ▪ ABF noise profiling ▪ ANR noise profiling ▪ Bayer AWB tuning
 18% gray chart	▪ AWB reference point ▪ AWB low-light LUT

Table B-1 Test charts for capturing images (cont.)

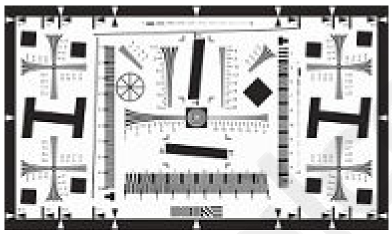
Chart or equipment	Applicable tuning procedures
 Diffuser filter	■ LSC tuning ■ Linearization
 ISO12233 resolution chart	■ ASF sharpness tuning

Table B-2 Equipment for capturing images

Equipment	Purpose
 Light booth	Provides the required types of light sources: ▪ Daylight – D75, D65, D50 ▪ Fluorescent – CW, TL84 ▪ Incandescent – A and H
 Light meter	▪ Measures lighting intensity ▪ Must be able to show the lighting intensity in lux
 Neutral density (ND) filter	▪ Used when the light booth cannot control lighting intensity

B.2 Measure the lux level with a light meter

When capturing images for camera tuning, the lux level of each image must be recorded. Perform the following steps to properly measure the lux level.

1. Set the light meter to Ev mode where the meter shows the intensity of the light that falls on its sensor (white bulb).
2. Place the light meter on the chart and ensure that the light meter sensor is facing the camera. The following example reads 520 lux.

B.3 Capture images and videos to use for tuning with Chromatix

Qualcomm Spectra 2xx adopts a flexible tuning interface. For example, users can use 6 or 10 regions to cover scene brightness. For spatial processing modules (denoising, sharpening, etc.) by default, 6 regions are used to cover the gain range. Images are captured at 1000/500/200/100/50/10 lux levels for regions 1 through 6. If you require more regions, capture more images at other lux levels.

As a prerequisite, ensure AEC is working correctly before capturing raw images.

NOTE Images imported into Chromatix must not have spaces in their titles, or an error message appears.

Set	Test scenes	Tuning blocks
1	Black	Linearization
2	Diffuser	■ LSC ■ HNR radial processing
3	MCC images	■ Color correction ■ ABF noise profiling ■ HNR and ASF skin color detection
4	Flat field images	Color correction
5	ISO 12233 resolution chart	ASF
6	MCC image/video sequences	■ ANR ■ TF ■ MF
7	Flat field image/video sequences	■ ANR ■ TF ■ MF
8	Natural scene image/video sequences	■ ANR ■ TF ■ MF
9	Outdoor landscape sequence	MF
10	Outdoor portrait sequences	MF
11	Dot chart images	ICA
12	Natural scene video with sufficient details	LRME
13	Natural scene video with insufficient details	LRME

Refer to the Suggested Images spreadsheet attachment for the appropriate chart and lighting conditions with which to capture the raw images for tuning a specific ISP block. Follow the image capture procedure specific to the ISP block or the chart type.

RELATED INFORMATION

“Capture black level images” on page 177
“Capture lens rolloff images” on page 177
“Capture MCC images” on page 180
“Capture flat field images” on page 180

"Capture gray chart images in the light booth" on page 181
"Capture ISO12233 resolution chart images" on page 182
"Capture gray chart images outdoors on a sunny day" on page 183
"Capture MCC image and video sequences" on page 184
"Capture flat field image and video sequences" on page 185
"Capture natural scene image and video sequences" on page 186
"Capture outdoor landscape and portrait sequences" on page 186
"Capture dot chart images" on page 187
"Capture natural scene LRME video" on page 188
"Capture ANR images" on page 188
"Capture TF image and video sequences" on page 189
"Capture MF image and video sequences" on page 189

B.3.1 Capture black level images

Related tuning block: Linearization

Capture two raw images in a completely dark room with the lens covered by a dark object.

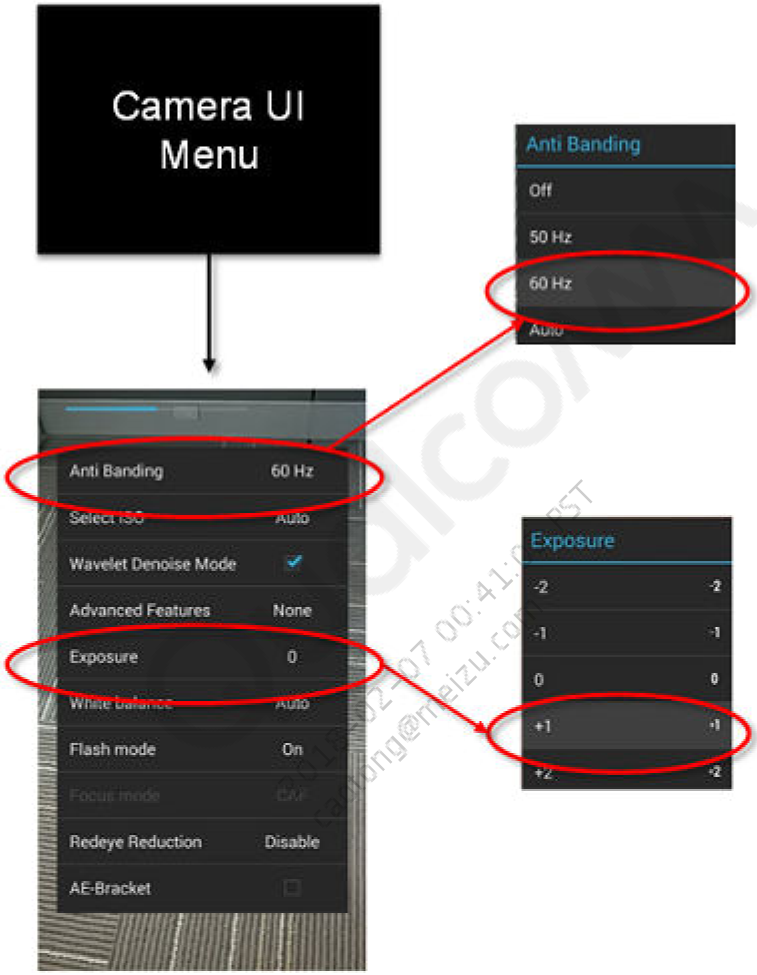
1. Under each lighting condition, let the camera's AEC stabilize.
2. Turn off AEC.
3. Turn off all lights in the test room and capture a raw image with the lens covered with a dark object (e.g., black tape).
4. Turn on AEC.
5. Repeat Steps 1 through 4 with all light conditions.

B.3.2 Capture lens rolloff images

Related tuning block: LSC and HNR

20 lux images are used for low-light LSC. The 200 lux TL84 image is used for HNR radial calibration.

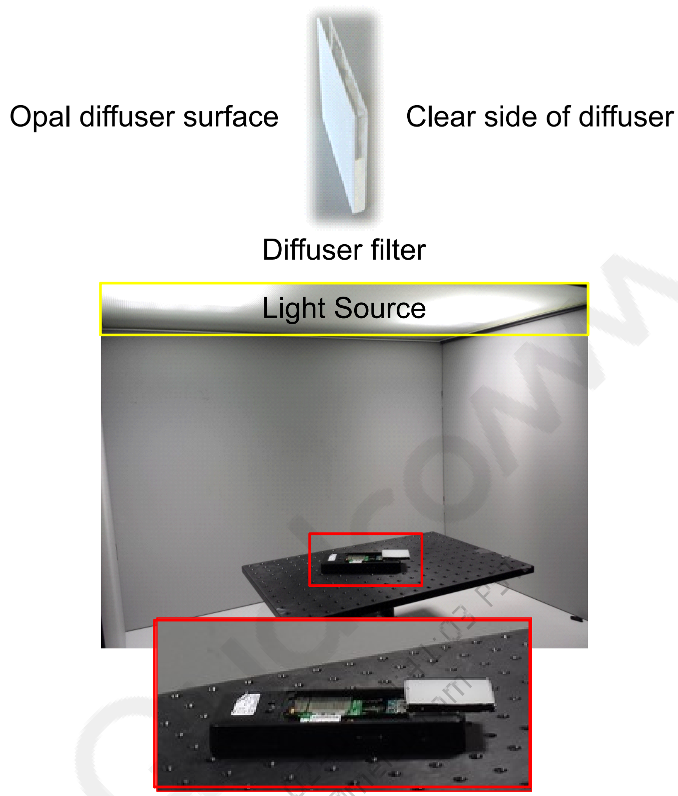
1. Enable the antibanding option in the camera UI menu for the frequency of the commercial power in your area.
2. Change the exposure setting to +1.



NOTE Diffuser images for lens rolloff compensation are dark and can cause problems when optimizing the lens rolloff parameters. Increasing the exposure setting to +1 eliminates this problem.

3. Place the opal side of diffuser filter on the camera lens.

4. Aim the camera directly at the light source and capture raw images.



5. Change the exposure setting back to 0 after capturing all required images. Leave antibanding enabled.

B.3.3 Capture MCC images

Related tuning blocks:

- Color correction (use 20 lux images for low-light CC)
- ABF noise profiling (use 1000/500/200/100/50/10 lux TL84 images)
- Green imbalance correction (GIC)

NOTE If the image is taken with a fisheye lens, to set the MCC markers for tuning, hold the Ctrl key (or Shift key) while moving the markers.

Perform the following steps for each color temperature and light level combination specified in the Suggested Images spreadsheet whose chart type value is **Macbeth**.

a. Place an MCC at the bottom-right of the light booth.

b. Position the device so that the MCC fills 70% of the camera's FOV.

c. Set the light booth conditions to match those specified in the Suggested Images spreadsheet. If you cannot control the brightness of the light booth, adjust the brightness using ND filters.

d. Capture a raw image.

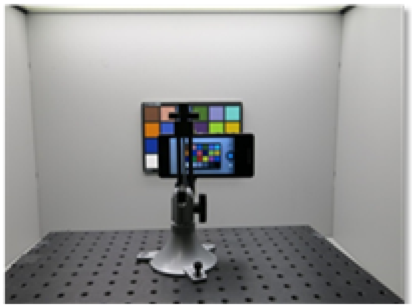
B.3.4 Capture flat field images

Related tuning block: Color correction matrix

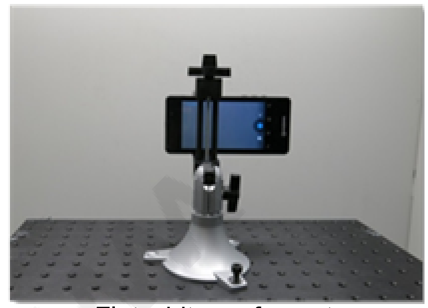
Perform the following steps for each color temperature and light level combination specified in the Suggested Images spreadsheet whose chart type value is **FlatField**.

1. Remove all charts from the light booth.
2. Position the device in the same place it was located when capturing the corresponding MCC image.

3. Set the light booth conditions to match those specified in the Suggested Images spreadsheet. If you cannot control the brightness of the light booth, adjust the brightness using ND filters.
4. Capture a raw image.



MCC chart at light source D65



Flat white surface at light source D65

B.3.5 Capture gray chart images in the light booth

Related tuning blocks:

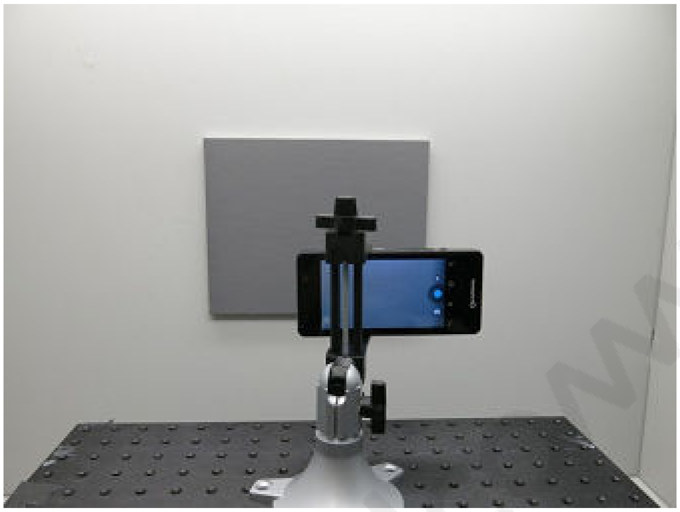
- AWB reference point
- AWB low-light LUT

JPEG format gray chart images are required for 3A tuning. Refer to the Suggested Images spreadsheet for the specific color temperature and light level combinations when capturing those images.

Perform the following steps for each color temperature and light level combination specified in the Suggested Images spreadsheet whose chart type value is **GrayChart**.

1. Place an 18% gray chart in the middle center of the light booth.
2. Positon the device so that the chart fills 100% of the camera's FOV.

3. Set the light booth conditions to match those specified in the Image Gallery spreadsheet. If you cannot control the brightness of the light booth, adjust the brightness using ND filters.
4. Capture a raw image.

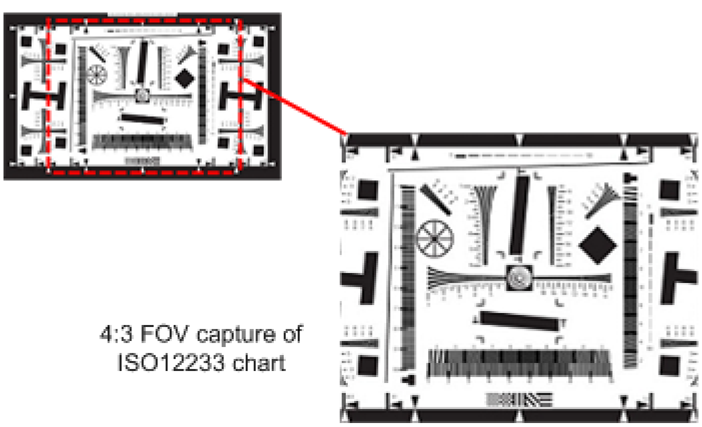


B.3.6 Capture ISO12233 resolution chart images

Related tuning block: ASF

Perform the following steps for each color temperature and light level combination specified in the Suggested Images spreadsheet.

1. Position the ISO12233 resolution chart in the middle center of the light booth.
2. Position the device so that the chart fills 100% of the height of the FOV. The FOV will typically capture the area bound by the red dashed line in the figure.
3. Set the light booth conditions to match those specified in the Suggested Images spreadsheet. If you cannot control the brightness of the light booth, adjust the brightness using ND filters.
4. Capture a raw image.



B.3.7 Capture gray chart images outdoors on a sunny day

Related tuning blocks:

- AWB reference point
- AWB low-light LUT

1. At approximately noon and under direct sunlight, fill 100% of the preview screen with an 18% gray chart.
2. Capture a raw image.

B.3.8 Capture MCC image and video sequences

Related tuning blocks: ANR, TF, and MF

Scene	File format	Requirements	ANR calibration	TF calibration	MF calibration
MCC	Image	▪ 70% FOV ▪ (Advanced) Can capture EV-1 and EV +1	Process with IFE or BPS, with rolloff disabled	NA	NA
	Video sequence	▪ Requires at least 20 frames ▪ 70% FOV ▪ Constant gain/3A during capture ▪ (Advanced) Can capture EV-1 and EV +1	NA	Process with IFE, with rolloff disabled	NA
	Image sequence	▪ Requires at least 10 frames ▪ 70% FOV ▪ Constant gain/3A during capture ▪ (Advanced) Can capture EV-1 and EV +1	NA	NA	Process with BPS, with rolloff disabled

Total # of images to capture = # of images to capture per lighting condition * # of lighting conditions. For most spatial modules, use six regions to cover bright to low light conditions.

B.3.9 Capture flat field image and video sequences

Related tuning blocks: ANR, TF, and MF

Scene	File format	Requirements	ANR calibration	TF calibration	MF calibration
Flat field	Image	–	Process with IFE or BPS, with rolloff enabled	Process with IFE, with rolloff enabled	Process with BPS, with rolloff enabled
	(Advanced) Video sequence	▪ Requires at least 20 frames ▪ Constant gain/3A during capture	NA	Process with IFE, with rolloff disabled	NA
	(Advanced) Image sequence	▪ Requires at least 10 frames ▪ Constant gain/3A during capture	NA	NA	Process with BPS, with rolloff disabled

Total # of images to capture = # of images to capture per lighting condition * # of lighting conditions. For most spatial modules, use six regions to cover bright to low light conditions.

B.3.10 Capture natural scene image and video sequences

Related tuning blocks: ANR, TF, and MF

Scene	File format	Requirements	ANR calibration	TF calibration	MF calibration
Natural scene (lightbooth with various objects)	Image	–	Process with IFE or BPS, with rolloff enabled	NA	NA
	Video sequence	▪ Requires a long sequence (~200 frames) ▪ Scene must include moving objects ▪ Constant gain/3A during capture	NA	Process with IFE, with rolloff enabled	NA
	Image sequence	▪ Requires at least 10 frames ▪ No tripod ▪ Scene must include moving objects (occupying 10-30% FOV) ▪ Constant gain/3A during capture	NA	NA	Process with BPS, with rolloff enabled

Total # of images to capture = # of images to capture per lighting condition * # of lighting conditions. For most spatial modules, use six regions to cover bright to low light conditions.

B.3.11 Capture outdoor landscape and portrait sequences

Scene	File format	Requirements	ANR calibration	TF calibration	MF calibration
Outdoor landscape	Image sequence	▪ Requires at least 10 frames ▪ Scene must include moving people/cars (occupying	NA	NA	Process with BPS, with rolloff enabled

Scene	File format	Requirements	ANR calibration	TF calibration	MF calibration
		up to 20% FOV) ■ Exposure time up to 100 ms with OIS; up to 33 ms without OIS ■ No tripod			
Outdoor portrait	Image sequence	■ Requires at least 10 frames ■ Scene must include people (occupying up to 30% FOV) ■ People should try not to move ■ Background should have details ■ No tripod	NA	NA	Process with BPS, with rolloff enabled

Total # of images to capture = # of images to capture per lighting condition * # of lighting conditions. For most spatial modules, use six regions to cover bright to low light conditions.

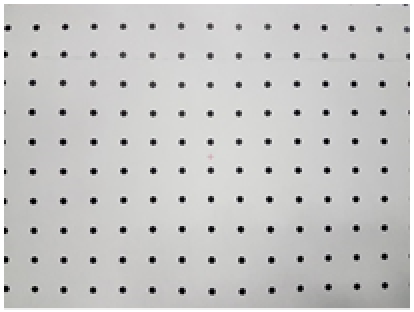
B.3.12 Capture dot chart images

Related tuning block: ICA1

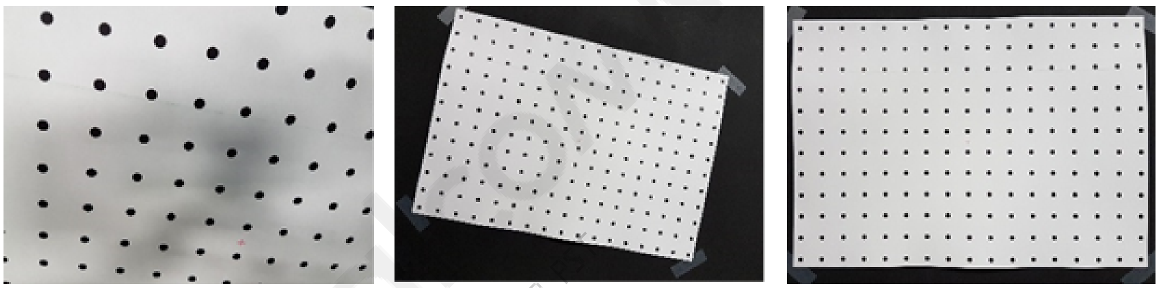
The light can be D65, TL84, or D55 and should be between 200 and 400 lux.

1. Place camera on the tripod and position the camera parallel and aligned (not rotated) with the dot chart.
2. Set the camera to the minimum gain to reduce noise and set the exposure so there is no flickering from the light source.
3. Set the chart to 100%+ FOV. The image should include only the chart.

4. Ensure the captured image has no motion blur.



Good example



Bad examples – (L) too close (C) rotated (R) visible background

B.3.13 Capture natural scene LRME video

Low resolution motion estimation (LRME) finds the motion vector (MV) field between two low-resolution (downscaled) images. LRME should be tuned after IFE tuning. When TF is configured to use LRME, LRME should be tuned first.

For registration tuning, capture video with global motion that includes:

- A lab scene (or something similar) with sufficient details
- Mimicry of slight hand-shake motion
- Some moving objects (e.g., trains, rotating tables, a person moving)
- Flat regions suitable for noise level estimation (ensure the flat region is at least 5% x 5% FOV)

(Advanced) For transform confidence tuning, capture video with insufficient details that includes:

- A single object on a uniform background
- Many occlusions and/or aperture problems (e.g., repetitive patterns such as carpets)

B.3.14 Capture ANR images

ANR is a multipass spatial noise filtering process available for snapshot and video modes.

1. Place an MCC in the center of the light booth.
2. Position the device on a tripod so that the MCC fills 50-70% of the camera's FOV.
3. Disable LSC before capturing (even if with a companion chip).
4. (Advanced) Capture the entire luma range with under/overexposed images (i.e., EV-1 and EV+1).

5. (Advanced) Place the device slightly out of focuse to resolve a scratched chart.
6. Capture an image of flat field (used for LNR and better low-frequency tuning) with lens rolloff enabled.
7. Capture an image of a natural scene (used for subjective fine tuning) with a scene that includes a variety of objects, colors, faces, flat areas, etc.

B.3.15 Capture TF image and video sequences

TF is applied to each video frame and blends the current frame with the previous frame.

The device and MCC chart must be static during video capture.

1. Place an MCC in the center of the light booth.
2. Position the device on a tripod so that the MCC fills 50-70% of the camera's FOV.
3. Disable LSC before capturing (even if with a companion chip).
4. Capture at least 20 frames with constant exposure, gain, and 3A.
5. (Advanced) Capture the entire luma range with under/overexposed images (i.e., EV-1 and EV+1).
6. Capture a single frame of flat field (used for LNR tuning) with lens rolloff enabled.
7. Capture video of a lab scene (used for subjective fine tuning) that includes:
   - A variety of objects, colors, faces, flat areas, etc.
   - Some moving objects (e.g., trains, rotating tables, etc.)
   - 200 frames or more
   - Constant exposure, gain, and 3A during capture
8. (Advanced) Capture flat field video (used for better low-frequency tuning) with LSC disabled (even if with a companion chip) that has at least 20 frames with constant exposure, gain, and 3A during capture.

B.3.16 Capture MF image and video sequences

The device and MCC chart must be static during video capture.

1. Place an MCC in the center of the light booth.
2. Position the device on a tripod so that the MCC fills 50-70% of the camera's FOV.
3. Disable LSC before capturing (even if with a companion chip).
4. Capture at least 10 frames with constant exposure, gain, and 3A.
5. (Advanced) Capture the entire luma range with under/overexposed images (i.e., EV-1, EV0, and EV+1).
6. Capture a single image of flat field (used for LNR tuning) with lens rolloff enabled.
7. Capture video of a lab scene (used for subjective fine tuning) that includes:
   - Steady hand-held footage (no tripod)
   - A variety of objects, colors, faces, flat areas, etc.

  - LSC enabled
  - Some moving objects (e.g., trains, rotating tables, etc.) that occupy 10-30% FOV
  - Constant exposure, gain, and 3A during capture
  - Sequence of 10 images

8. (Advanced) Capture flat field still sequence (used for better low-frequency tuning) with LSC disabled (even if with a companion chip) that includes at least 10 images with constant exposure, gain, and 3A during capture.
9. Capture an outdoor urban/landscape scene with moving people or cars included that take up up to 20% FOV with lighting conditions to properly expose the image. If using OIS, set the exposure time up to 200 ms; if not using OIS, set it to up to 33 ms. Use a steady hand to capture the footage, not a tripod.
10. Capture an outdoor portrait scene with one or more people standing still taking up 30% FOV. Ensure there are background details and that the lighting conditions are enough to properly expose the image.

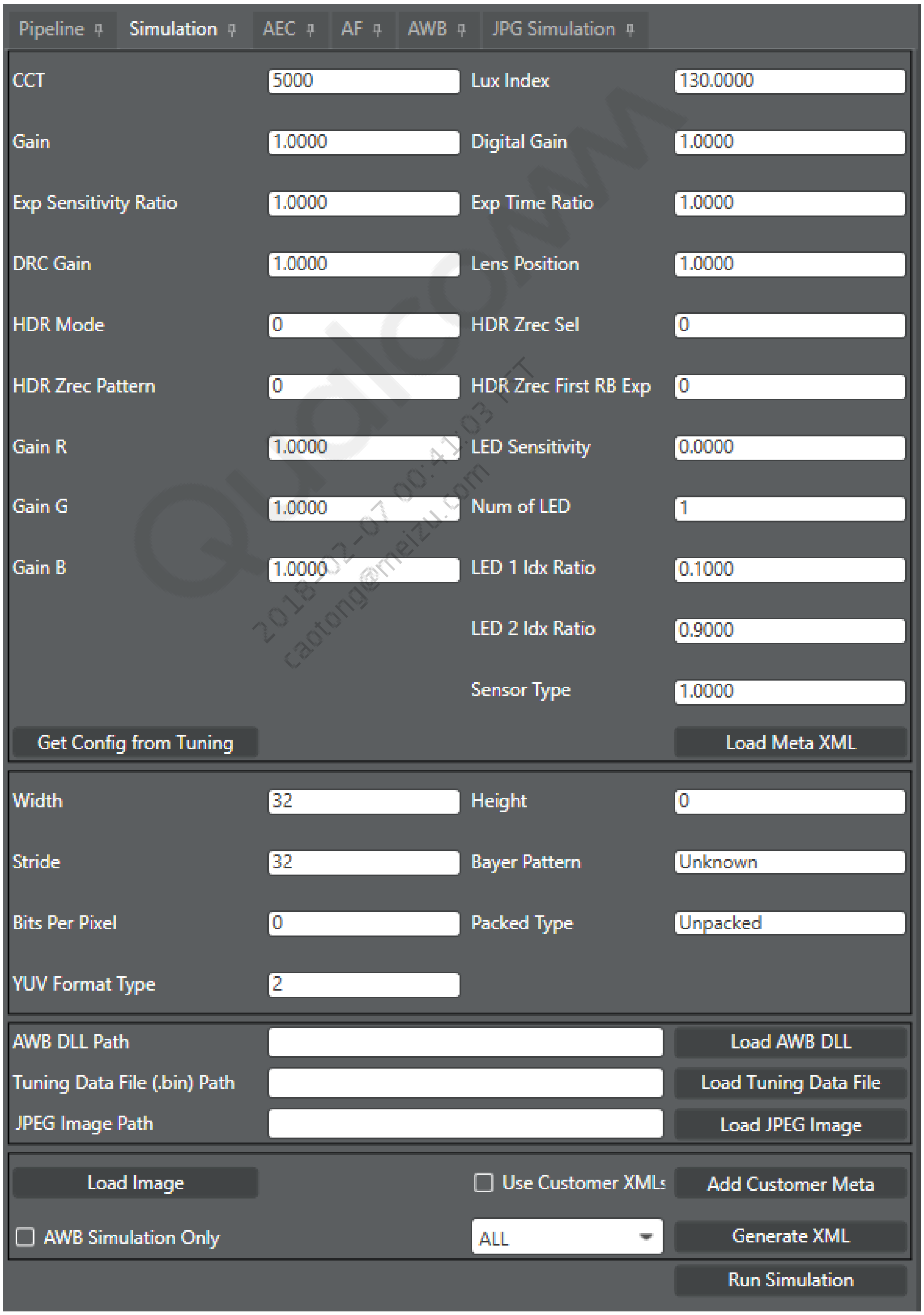
C Simulation Tool

The simulation tool allows you to apply the effect of a tuned project on a raw image to see how the tuned parameters affect the image.

C.1 Configure and run the simulation tool

Use this procedure to run an AWB simulation or a raw simulation, or both.

1. Open the correct tuned project in the Chromatix tool.
2. Click the **Simulation** tab to open the simulation window.



AWB Simulation

3. Click **Load AWB DLL** to navigate to the AWB simulation .dll provided in the software build. Select the .dll file and click **Open**.
4. Click **Load Tuning Data File** to navigate to the tuned data file from the target device. Select the target data file and click **Open**.
5. Click **Load JPEG File** to navigate to the .jpg file that has the 3A debug information. Select the image file and click **Open**.
6. (Optional) If no simulation subfolder exists in the appropriate project folder, click **Generate XML** to create the subfolder. This is generally necessary only the first time a simulation is run for a specific project.
7. Run the simulator to produce a simulated image based on the AWB info associated with the JPEG file. Click **Run Simulation** and wait until **Simulation finished** shows in the simulation window.

   The resulting output gains and CCT are used for the raw simulation.

Raw Simulation

8. (Optional) Load previously saved Meta XML or Stream XML files.
9. Enter or adjust configuration values as needed.
10. If needed, adjust values in the Gain R, Gain G, and Gain B white balance fields.
11. Click **Load Image**, navigate to the raw image to use for the simulation, and click **Open**.
12. (Optional) To reuse the existing XML files in the project `simulation.default` subfolder, mark the **Use Customer XMLs** check box. If this check box is not marked, the simulation overwrites the XML files in the project simulation subfolder.
13. (Optional) If no simulation subfolder exists in the appropriate project folder, click **Generate XML** to create the subfolder. This is generally necessary only the first time a simulation is run for a specific project.
14. Run the simulator to produce a simulated image based on your tuned pipeline settings. Click **Run Simulation** and wait until **Simulation finished** shows in the simulation window.

    The simulated image appears in the image pane for evaluating the result.

RELATED INFORMATION

"Configure and run the JPG simulation tool" on page 193
"Run an AWB simulation from the command line (Spectra 2xx)" on page 197
"Run a raw simulation from the command line" on page 197

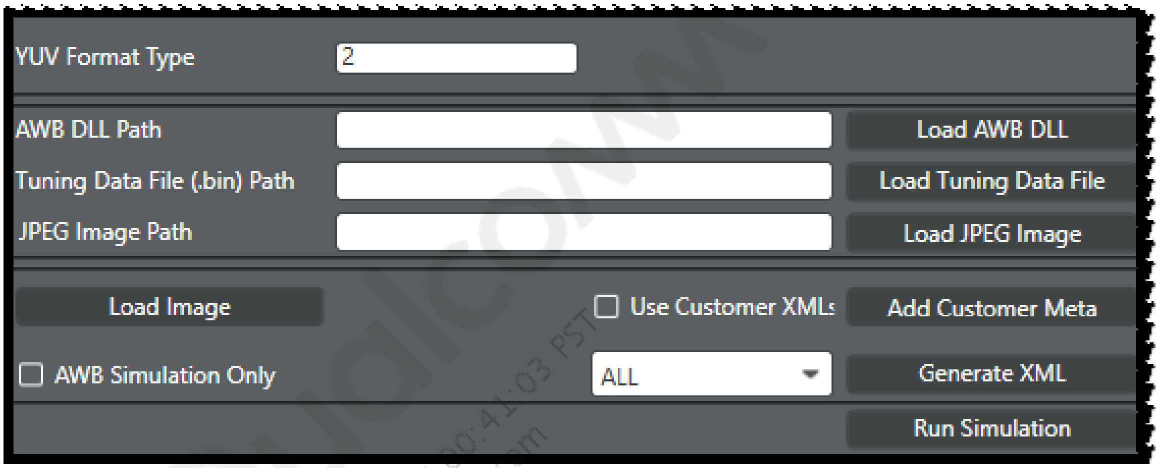
C.2 Configure and run the JPG simulation tool

The JPG Simulator provides a simulation result using a JPEG image in less time than running the conventional Simulator using a RAW image.

The JPG Simulator result is less accurate because it does not process all the ISP tuning modules, but it can be useful when time is of greater importance than accuracy.

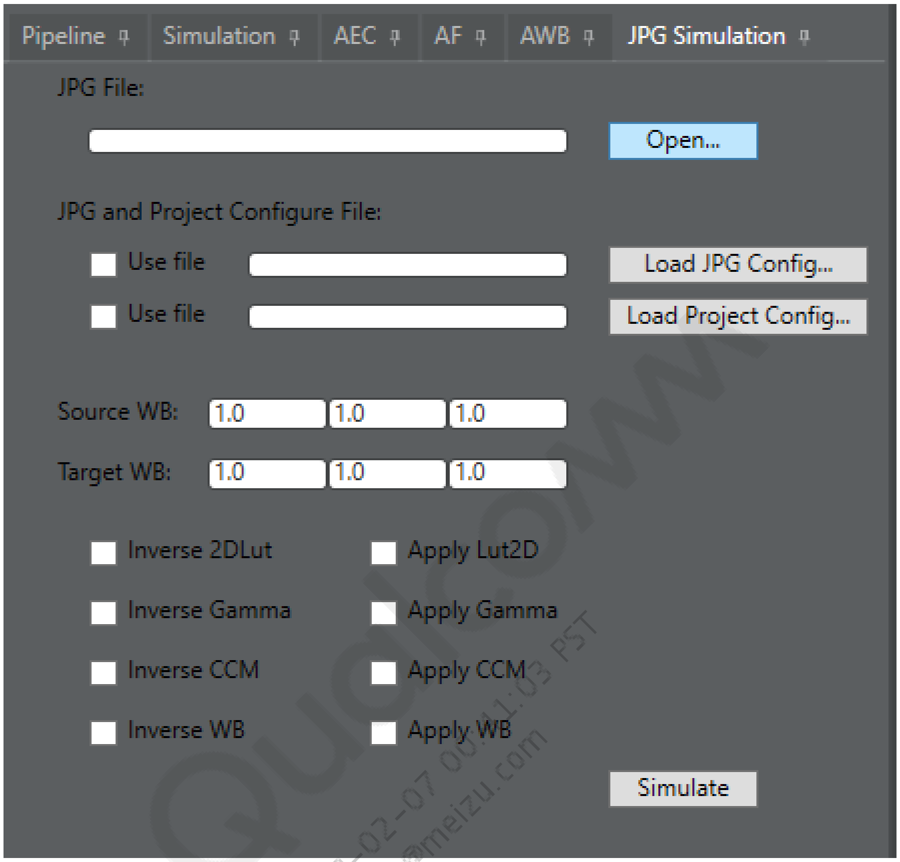
The JPG Simulator considers a limited selection of tuning module updates: 2DLut, Gamma, CCM, White Balance, plus AWB tuning. These five modules affect image color tone and can be selectively reversed and then applied after further tuning.

1. Open the appropriate project in Chromatix
2. Tune AWB parameters as needed and save the project.
3. Copy the project .bin file to the target device.
4. Using that target device, capture a JPEG image with 3A debug data and metadata.
5. Click the **Simulation** tab to open the conventional simulator.



6. Run AWB simulation as follows:
   a. Click **Load AWB DLL** to navigate to the AWB simulation .dll provided in the software build. Select the .dll file and click **Open**.
   b. Click **Load Tuning Data File** to navigate to the tuned data file from the target device. Select the target data file and click **Open**. **NOTE:** For subsequent simulations, it is important to update the .bin file after making any tuning parameter updates.
   c. Click **Load JPEG File** to navigate to the JPEG file from Step 4. Select the image file and click **Open**.
   d. Mark the **AWB Simulation Only** check box.
   e. Click **Load Image** and upload any RAW image (temporary workaround for this release).
   f. Click **Run Simulation** and wait until *Simulation finished* shows in the status window. The resulting R/G/B gains and CCT are updated in the upper portion of the simulator window.

7. With the same project open in Chromatix, click the **JPG Simulation** tab.



8. Click **Open** and upload the JPEG image captured in Step 4.
9. Complete the **JPG Configuration file** field as follows:
   a. For a first-time simulation, clear the **Use file** check box. The simulation process populates this field with the path to the generated image config file (under the image folder).
   b. For a subsequent simulation where a new JPEG image is used, clear the **Use file** check box so that the image config file is regenerated.
   c. For a subsequent simulation where the same JPEG image is used, mark the **Use file** check box to save time by reusing the existing image config file.
10. Complete the **Project Configuration file** field as follows:
    a. For a first-time simulation, clear the **Use file** check box. The simulation process populates this field with the path to the generated project config file (under the project folder).
    b. For a subsequent simulation using updated parameters, clear the **Use file** check box so that the project config file is regenerated.
    c. For a subsequent simulation using unchanged parameters, mark the **Use file** check box so the simulation saves time by reusing the existing project config file.
11. Ensure the following fields are populated:
    - **Source WB:** Populated with the R/G/B gains from the uploaded JPEG image
    - **Target WB:** Populated with the R/G/B gains read from the tuned pipeline (IFE or BPS)

12. Complete the **Inverse** check boxes as follows:
    - When simulating for AWB, ensure that all **Inverse** check boxes are marked.
    - If a particular simulation would benefit from inverting any of the four modules so that updated parameter values can be applied, mark the check box next to the applicable **Inverse<module>** item. Otherwise, clear the check box.
13. Complete the **Apply** check boxes as follows:
    - When simulating for AWB, ensure that all **Apply** check boxes are marked.
    - If a particular simulation would benefit from applying any of the four modules with updated parameter values, mark the check box next to the applicable **Apply<module>** item. Otherwise, clear the check box.
14. Click **Simulate** and wait until *Simulation finished* shows in the Status window.

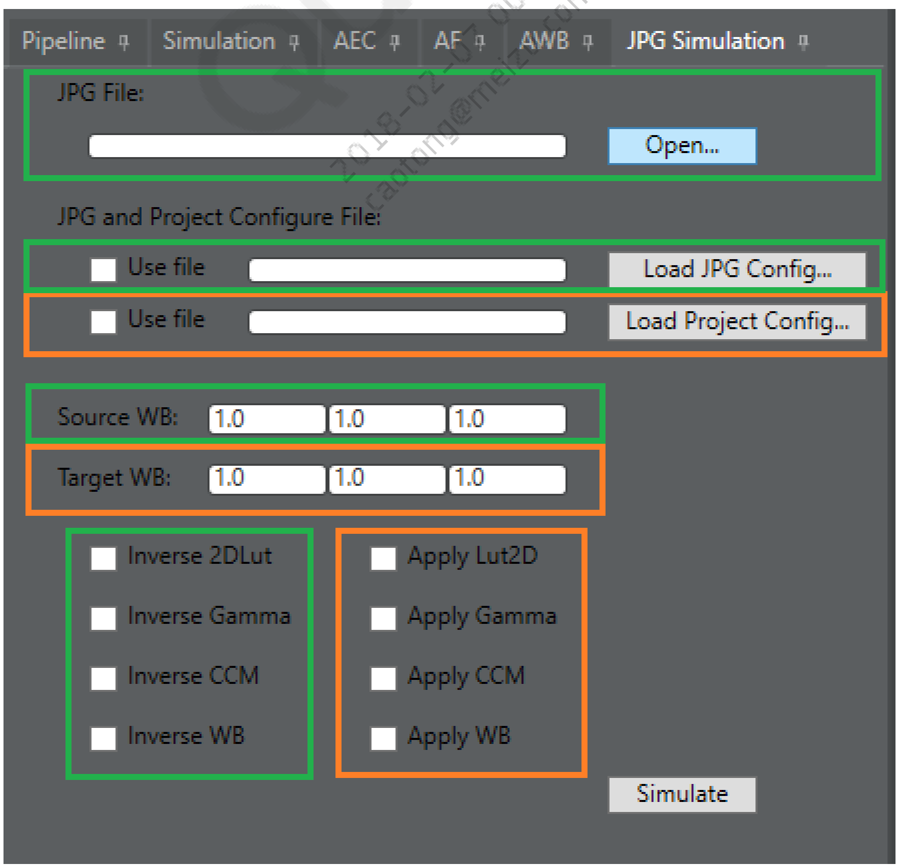
    A simulated image appears in the image pane for evaluating the result.

RELATED INFORMATION

"Configure and run the simulation tool" on page 192

C.2.1 JPG simulator user interface

Notice which fields on the JPG Simulator tab relate to the source image (green boxes) and to the project parameters (orange boxes).



C.3 Run an AWB simulation from the command line (Spectra 2xx)

As an alternative to using the Chromatix Simulation tool UI, run an AWB simulation via the command line. To expedite AWB simulation from multiple images, create and run a batch file with multiple commands.

If you have an existing Chromatix project, use these steps to run an AWB command line simulation that uses the files from that project.

1. Open the command prompt.
2. Navigate to the Chromatix directory.
3. To enable the command line simulation, type something like the following:

```
Chromatix.exe -awbsim -awbdll <C:\Dropbox\C7AwbSim\com.qti.stats.awb.dll
-bin C:\Dropbox\C7AwbSim\imx318tuned.bin -jpg C:\Dropbox\C7AwbSim
\IMG_113921.jpg
-inputMeta C:\Dropbox\myproj_awb\Simulation\Default
\meta_config_default.xml
-outputMeta C:\Dropbox\myproj_awb\Simulation\Default\meta_config.xml
```

Argument	Description
-awbsim	Enable AWB simulation mode
-awbdll <filename>	Path to the awb dll file in the software build
-bin <filename>	Path to the tuned project binary file for the target device
-jpg <filename>	Path to the JPEG image that has the 3A debug data
-inputMeta <filename>	Path to the default metadata file
-outputMeta <filename>	Path to the output file for the updated metadata file

4. Add a set of commands for each JPEG file that you want to use for AWB simulation.
5. Run the command.

RELATED INFORMATION

"Configure and run the simulation tool" on page 192

C.4 Run a raw simulation from the command line

As an alternative to using the Chromatix Simulation tool UI, run a raw simulation via the command line. To expedite a simulation of multiple images.

If you have an existing Chromatix project, use these steps to run a command line simulation that uses the files from that project.

1. Navigate to the project folder and view the `<project>\Simulation\Default \spectracamsim_config.xml` file.

   NOTE If there is no Simulation subfolder in the project folder, open the Simulation tab in the Chromatix GUI and click **Generate XML** to create the simulation default files.

2. Check the default values for the `Input Folder` and `Output Folder` parameters; edit if needed.
3. Upload the raw images you want to simulate into the Input Folder defined in the `spectracamsim_config.xml` file.
4. Verify the `<project>\Simulation\Default\meta_config.xml` file values are set the way you want them; edit if needed.

   NOTE When simulating a video, there is a metadata file for each frame, such as meta_config_0.xml and meta_config_1.

5. Open the command prompt.
6. Navigate to the Chromatix directory.
7. To enable the command line simulation, type the following:

```
QualcommSpectraSim280.exe C:\Dropbox\myproj\Simulation\Default
\spectracamsim_config.xml
```

8. Run the command.

   NOTE The resulting images will appear in the Output Folder defined in the `spectracamsim_config.xml` file.

D References

D.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Camera 3A 7.0 Tuning Guide	80-NK872-14
Qualcomm Spectra 280 Deep Dive	80-P9301-60
Chromatix 7 Mode Tree: Inheritance, Use Cases, Scenes, and Other Scenarios (Part 1)	VD80-P9301-202A
Chromatix 7 Mode Tree: Inheritance, Use Cases, Scenes, and Other Scenarios (Part 2)	VD80-P9301-202B