

ISP Tuning Guide

Revision 0.2

Table of Contents

1. Terms and Abbreviations	4
2. Introduction	5
2.1. Environment requirement of the tuning tool	5
2.2. Tuning overview	5
3. Getting started	6
3.1. Input an image	6
3.2. Workspace	7
3.3. ISP setting	10
4. LSC	14
4.1. Calibrate CT shading table	14
4.2. Reduce compensation	15
4.3. Criteria setting	16
5. AWB Calibration	17
5.1. Tool Overview	17
5.2. Interface Function Introduction	17
5.3. Standard Operation Procedure(SOP)	19
5.4. Detailed Function Description	20
5.5. Notes	20
5.6. FAQ Troubleshooting	21
6. ISP pipeline tuning	22
6.1. OB Calibration	22
6.2. Auto Bad Pixel Correction (ABC)	25
6.3. TNR	28
6.4. SNR	44
6.5. IRP 1 CFAi	50
6.6. Dynamic Tone Control	57
6.7. GTM	59
6.8. NRE Multi-Layer NR	62
6.9. NRE Edge Refine	73
6.10. LTM	89
6.11. HWIRP Stage 2	99
6.12. CFR Tuning Guide	118
6.13. Post Processing	123
6.14. Sharpness	130

7. Auto Exposure &Auto White Balance Tuning Guide	155
7.1. Auto Exposure Tuning Guide	155
7.2. Auto White Balance Tuning Guide	155
8. Process a batch of files	193
8.1. Create Setting File	193
8.2. Run Consecutive images and Independent Images	193
8.3. Run Independent simulate data	194
8.4. Generate Stage Output	194

1. Terms and Abbreviations.

This chapter lists and defines terms and abbreviations used throughout this document.

ABC	Auto Bad Pixel Correction
AD gain	analog and digital gain
AE	auto exposure
AFD	anti-flicker detection
AWB	auto white balance, including R, G, B gain value
ASharp	Adaptive Sharpness
BCCM	Blended Color Correction Matrix
BDS	Bayer domain scaler
BLC	Black Level Clamp
BV	brightness value
CCB	Cb Cr 444 Blur
CCM	Color Correction Matrix
CEAI	Color Filter Array Interpolation
CFR	Color Fringe Reduction
CS	Chroma suppression
DTC	dynamic tone control
ER	edge refine
FW	Firmware
GTE	Geometry Transformation Engine
GTM	Global Tone Mapping
HW	hardware
IQ	Image Quality
IRP	Image Reproduction Pipeline
ISP	Image Single Processor
LCE	Local Contrast Enhancement
LSC	Lens shading compensation
LTM	Local Tone Mapping
MWB	manual white balance
NRE	Noise Reduction Engine
OB	black offset
OTP	One-Time Programmable memory
PP	post processing
SW	Software
TNR	temporal noise reduction
WDR	wide dynamic range

2. Introduction

This document will introduce how to tune image quality by the tuning tool, and the procedures to tune better image quality for your device.

2.1. Environment requirement of the tuning tool

The internet should be connected. Otherwise tuning tool wouldn't be open correctly.

Suggestion Hardware:

- Intel Core i5 or higher level CPU.
- 8G RAM – higher is recommended.

Suggestion Software:

- Microsoft Windows 7 /8 /10 64 bit
- .Net Framework 4.7.2
- Visual Studio 2013 C++ Redistributable Package
<https://www.microsoft.com/en-us/download/details.aspx?id=40784>
- Visual Studio 2010 C++ Redistributable Package
<https://www.microsoft.com/en-us/download/details.aspx?id=14632>

2.2. Tuning overview

The tuning tool supports the tuning and simulation of image pipeline functions as follows:

- Pipe configuration
- LSC
- ABC
- Bayer crop (Window of interest (WOI) in Project Setting)
- TNR
- SNR
- BLC
- CFAi
- GTM
- RGB2YUV
- NRE
- LTM
- CCM
- Tone
- Chroma suppression
- Sharpness
- 2D LUT (Post Processing)

3. Getting started.

When execute gravityxr tuning tool, user need to create a new tuning project by enter the project setting page to set the image information. And please click the "Update" button on the upper left to update the settings. Once update the image information or update the tuning parameters, user could save out tuning project (.json) which describing the project information through 'save project file' button in project column and a folder containing the parameters of each ISP blocks. If there is already an existing project, user can press 'load project file' button to load the existing project (.json).

3.1. Input an image.

Refer to Fig.3.1, Entry the ISP setting page, select the "Image" tab, press the "Load" button, and load a Raw file for image quality simulation. Note that the format of the loaded file must be consistent with the format set in the "Image Setting". Then select the "Condition" tab (Fig. 3.2), input the metadata of raw, and press the "Update" button to finish loading the Raw. The tool will take a while to process the raw image.

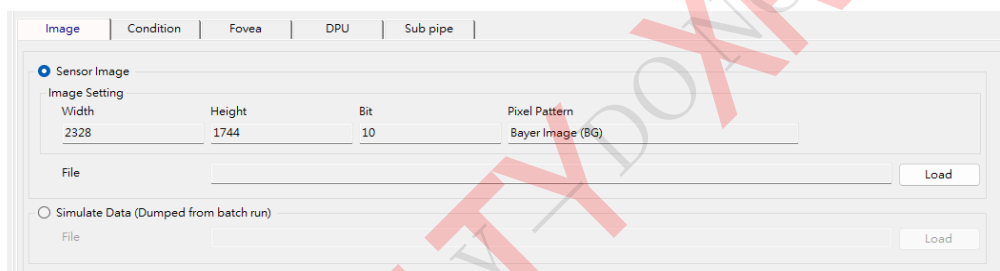


Fig. 3.1 load raw image

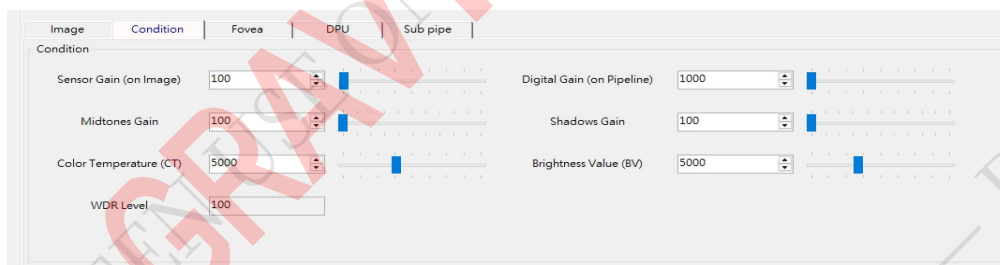


Fig. 3.2 raw condition setting page

3.2. Workspace

1. **Project and output strip:** In the green region in Fig. 3.3, is for saving/ loading the project file (.json) which including off-line simulation settings and binary file (.bin) which is applied to update ISP parameters to GravityXR pipeline.
2. **Modules list:** In the red region in Fig. 3.3, including buttons for setting project file, loading input image, and lists of all ISP modules. User can select functions in modules list, and shows the features of the functions in modules page. Including 5 sections:
 - **Project setting:** set project name / version, input and output format.
 - **ISP setting:** load input image file or simulated data which be dumped for TNR batch run, related with shooting scene.
 - **ISP Info:** list image quality parameter in json format.
 - **Batch run:** runs simulation in batch. Results of full pipeline, stage output, and TNR simulated file (.dat) can be created in this function.
 - **Function:** LSC function.
 - **ISP pipeline:** shows GravityXR pipeline. The status of each module is indicated by the color of checkbox

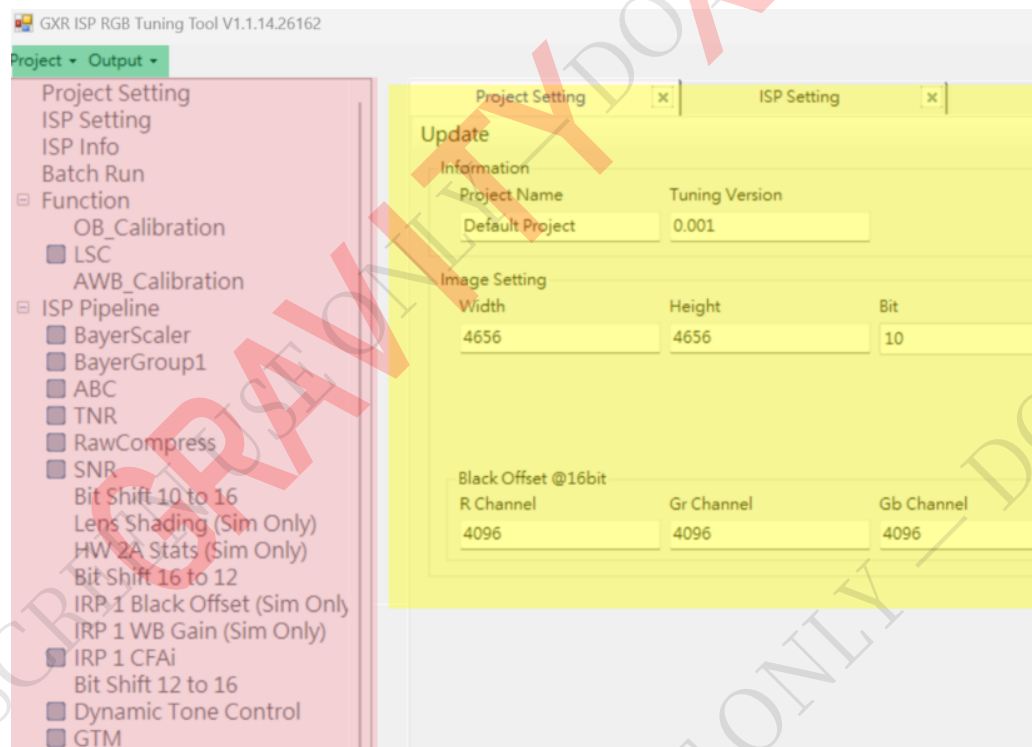


Fig. 3.3 Workspace of GravityXR tuning tool

3. **Status icons:** In the blue region in Fig. 3.3, legends indicate the module's status of current simulation. Descriptions of each icon is shown below:
 -  indicates the parameters are the same as default setting.
 -  indicates the parameters of loaded project file are updated.
 -  indicates the parameters of loaded project are different from default setting.
 -  indicates the module is disable in simulation.
4. **Module page:** In the yellow region in Fig. 3.3, shows tuning pages of modules selected in modules list. Users can close page by clicking  on the right of module's name, and updates parameter to ISP by clicking 'Update' button in the top-left of module page. If parameters aren't updated before being closed, parameter updated last time when users open it again.

3.2.1. Set input and output format of project file.

Open project setting page to sync the basic sensor setting for project simulation. It contains four parts:

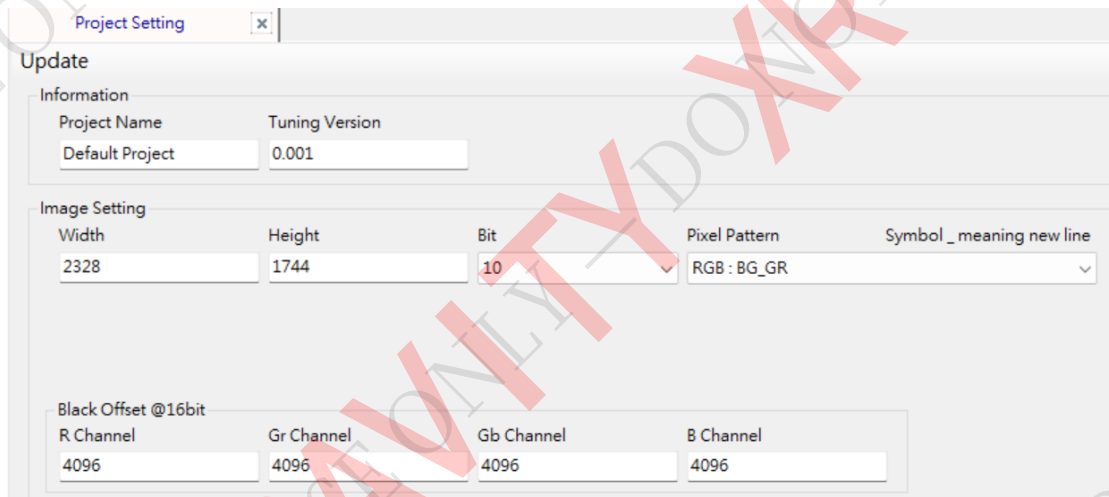


Fig. 3.4 Module page of project setting

1. **Information:** user can define a project name and a tuning version (range 0.001~999.999) to discriminate projects. These values will be shown in the filename of project file and binary file. The module number and the tuning version will be recorded in the first and the third values in binary file's IQ version.
2. **Image setting:** properties of input raw are set here, e.g., image width, image height, bit-depth (8-/10-bits for normal pipeline), Bayer pattern, and black offset. Please note that these setting are only for off-line simulation.
3. **Window of interest (WOI):** user can set a WOI by filling its top-left coordinate (axis X is horizontal and axis Y is vertical), width and height. These values should be 4-base. We suggest user to check the ratio of WOI with the scale-down resolution of Bayer scaling. Please note that these setting are only for off-line simulation.

4. **Black offset:** users need to synchronize the black offset of input raw with 16-bit base. Please note that these setting are only for off-line simulation.

3.2.2. Preview and display

There are icons on the top of preview window (Table 3.1) that help users to check the result of current refinement.

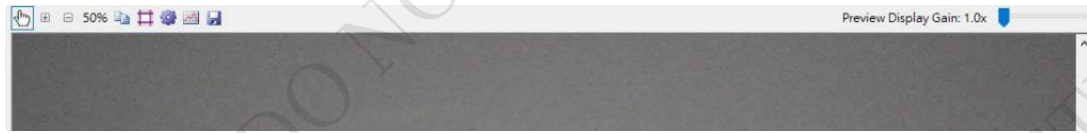


Fig. 3.5 UI in preview window

Table 3.1 Summary of icons in preview window

Icon	Tag	Usage
	Move	Left-click on the displayed image and drag it to the region you want to check.
	Zoom In / Out	Left-click on the displayed image to zoom in/out to check details and edges.
	Compare	Compare the result of current setting with the setting which is previously updated to ISP.
	Selection	Select regions for “Data Analysis”.
	Setting	Set the format of selection: 4 region arrangements in “Type” with 3 margins in “Interval” are provided.
	Data Analysis	Please refer to Data analysis
	Save BMP	Save the displayed image on preview windows in BMP format. Users also can save the input/output of selected module in its stage format by “Saving Type (T)” in “Save Image” page

3.2.3. Simulation

Since the windows version tuning tool is developed based on the .NET architecture. In order to achieve correct simulation results, please aware that it is possible to open tow tools of the same version in the windows environment at the same time and operate properly, however if two tools with different versions are opened at the same time, there may be unexpected simulation results. And when multiple tools with the same version are opened at the same time, only one tool can perform batch processing one time.

3.3. ISP setting

3.3.1.ISP Setting: Fovea

Image	Condition	Fovea	DPU	Sub pipe
pWoiStartW	0			
pWoiWidth	2328			
pBayerScalerOutputWidth	2328			
pBayerGroup1OutputWidth	1640			
pBayerGroup1FoveaLeft	517			
pBayerGroup1FoveaWidth	1294			
pBayerGroup1MaxRatioW	0			
pBayerReGroup1OutputWidth	1640			
pWoiStartH	0			
pWoiHeight	1744			
pBayerScalerOutputHeight	1744			
pBayerGroup1OutputHeight	1440			
pBayerGroup1FoveaTop	260			
pBayerGroup1FoveaHeight	1224			
pBayerGroup1MaxRatioH	0			
pBayerReGroup1OutputHeight	1400			

Fig 3.6 Fovea Configuration Tab

Overview: This interface configures dimensions related to the Bayer domain, including settings for the Crop, Scaler, and Group functional modules. These configurations determine the processing dimensions for subsequent modules and are critical to tuning performance. The purpose of each configurable parameter is described below.

- **pWoiStartW, pWoiStartH, pWoiWidth, pWoiHeight:** Crop-related parameters. If no Crop is needed, set pWoiStartW and pWoiStartH to 0, and set pWoiWidth and pWoiHeight to the Sensor output dimensions.
- **pBayerScalerOutputWidth, pBayerScalerOutputHeight:** Scaler-related parameters. If no Scaler is needed, set these to the values of pWoiWidth and pWoiHeight respectively.
- **pBayerGroup1OutputWidth, pBayerGroup1OutputHeight:** Group1-related parameters that define Group1 output dimensions.

- **pBayerGroup1FoveaLeft, pBayerGroup1FoveaTop, pBayerGroup1FoveaWidth, pBayerGroup1FoveaHeight:** Group1-related parameters that define the boundary and size of the Group1 Fovea region.
 - pBayerGroup1FoveaLeft range: [0, pBayerScalerOutputWidth – pBayerGroup1FoveaWidth]
 - pBayerGroup1FoveaTop range: [0, pBayerScalerOutputHeight – pBayerGroup1FoveaHeight]
 - pBayerGroup1FoveaWidth must be less than pBayerScalerOutputWidth.
 - pBayerGroup1FoveaHeight must be less than pBayerScalerOutputHeight.
- **pBayerGroup1MaxRatioW, pBayerGroup1MaxRatioH:** Group1 maximum scale-down ratio. A setting of 0 is recommended for automatic calculation. A manual value of 65536 corresponds to a 1× ratio.

3.3.2.ISP Setting: DPU

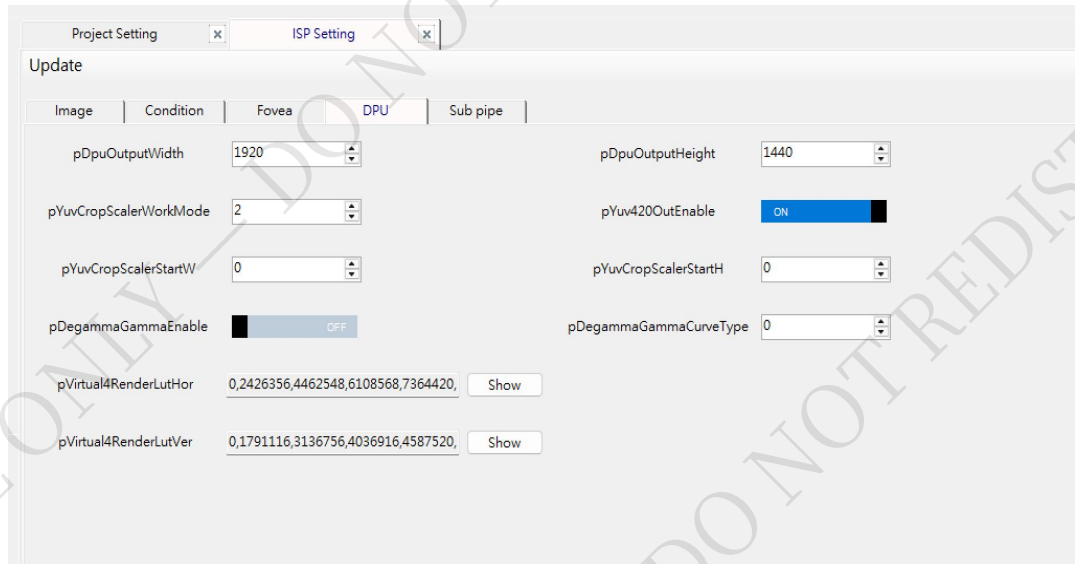


Fig 3.7 DPU Configuration Tab

- **pDpuOutputWidth, pDpuOutputHeight:** Output dimensions for the DPU path
- **pYuvCropScalerWorkMode:** 0 = Crop, 1 = Scaler, 2 = Disable
- **pYuv420OutEnable:** Determines the YUV format output to the DPU path. ON = YUV420, OFF = YUV422
- **pYuvCropScalerStartW, pYuvCropScalerStartH:** Takes effect when pYuvCropScalerWorkMode = 0, defining the Crop region boundary. The output size is determined by pDpuOutputWidth and pDpuOutputHeight. Set to 0 or leave unconfigured otherwise.
- **pDegammaGammaEnable:** Degamma and Gamma enable switch.
- **pDegammaGammaCurveType:** 0 = sRGB, 1 = gamma2.2, 2 = gamma2
- **pVirtual4RenderLutHor, pVirtual4RenderLutVer:** For display only.

3.3.3.ISP Setting: Sub Pipe

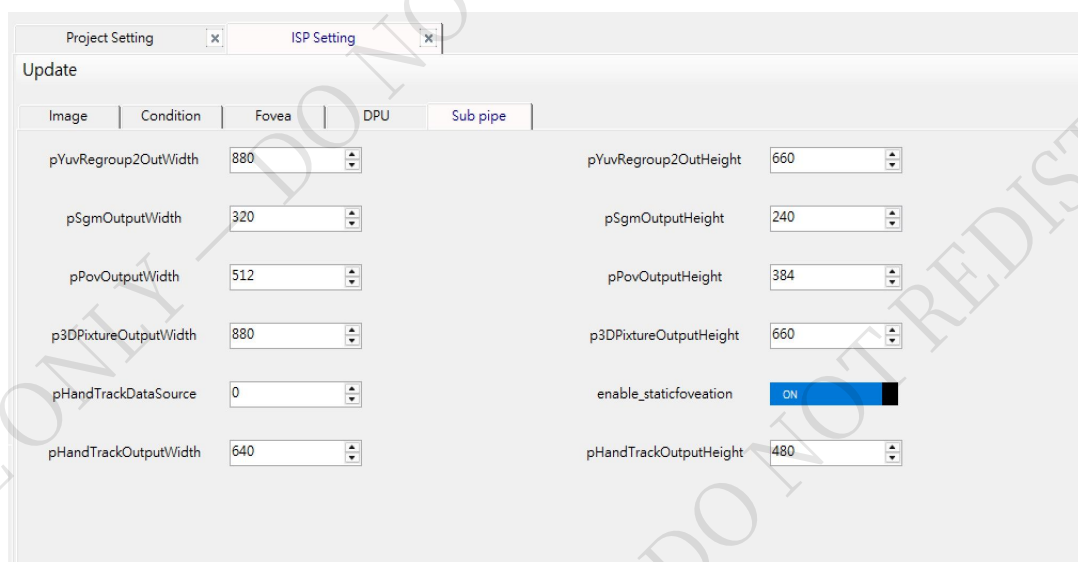


Fig 3.8 Sub Pipe Configuration Tab

- **pYuvReGroup2OutWidth, pYuvReGroup2OutHeight:** YuvRegroup2 parameters; these dimensions correspond to the original input size for all Sub channels.
- **pSgmOutputWidth, pSgmOutputHeight:** Output dimensions for the SGM channel.
- **pPovOutputWidth, pPovOutputHeight:** Output dimensions for the Depth/Matting/Pov channel.
- **p3DPictureOutputWidth, p3DPictureOutputHeight:** Output dimensions for the 3DPicture channel.
- **pHandTrackDataSource:** 0 = Use YuvRegroup2 output as input; 1 = Use Dewarp output as input.
- **Enable staticfoveation:** Default = ON
- **pHandTrackOutputWidth, pHandTrackOutputHeight:** Output dimensions for the HandTrack channel.

4. LSC.

By generate CT table to achieve automatic compensate LSC in different ambient color temperature.

4.1. Calibrate CT shading table.

The LSC tuning file contains 7 CT shading tables, and each table records its color temperature. The following sections show what equipment are required and how to set the basic shading table in the tuning file.

1. Environment requirement: light box or light source box is needed to satisfy multi color temperature environment, recommend are 2400K, 3000K, TL84, CWF , 5000K(D50),D65 and D75.
2. Calibrate basic shading table:

Step 1. Get uniform raw image which gray level of g-channel of center 1/5 area is between 600 to 640 from 1x sensor AD gain and use diffuser when needed. Please make sure that there is no other ambient light source except light box or light source box. The captured image could be like Fig 4.1. Repeat Step1, get shading raw from the recommend light source, 2400K, 3000K, TL84, CWF , 5000K(D50),D65 and D75



Fig 4.1 captured shading image.

Step 2. Refer to Fig 4.2, input the CT value in Map page and click the calibration button to load the relative raw image to calibrate basic shading table and repeat Step2 to calibrate all CT images from Step1. Be aware that there will be an error message when input image does not meet the criteria.

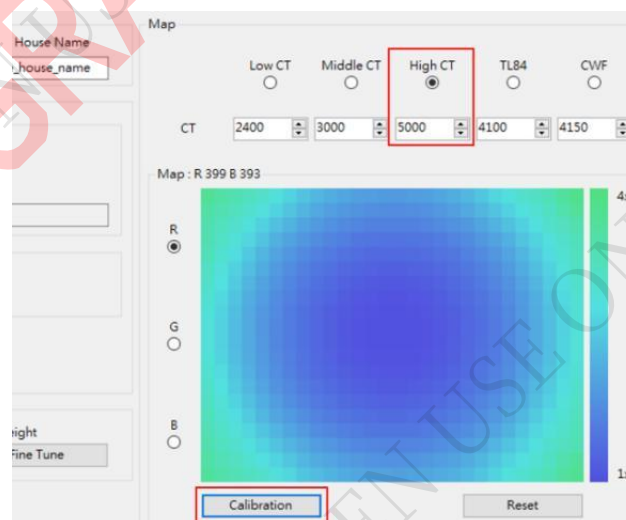


Fig 4.2 Calibrate CT shading table. By fill in the CT value and load the relative raw image from calibration button.

4.2. Reduce compensation.

For example, in order to avoid high ISO noise in the Fig 4.4 when input ISO = 100, down ratio = $1024 / 1024 = 1$ which means no decrease, and if input ISO = 2800, down ratio = $256 / 1024 = 0.25$ which means the gain value in LSC table will multiply 0.25.

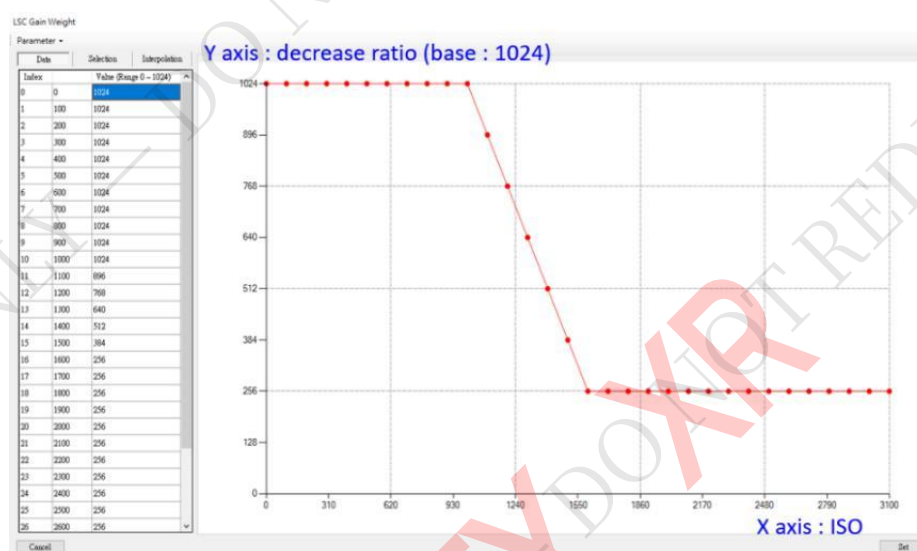


Fig 4.3 Decreased gain ratio curve.

4.3. Criteria setting

Not recommend adjusting criteria setting, unless the module is not evenly enough. By entering the 'calibration criteria' which next to 'update' button and check the box of 'Shading Criteria'. You could adjust the vignetting percentage, color shading percentage, central intensity range which means the brightness of the center of raw and symmetry threshold which means the balance of four corners. Or you could reset to default by check the 'reset to default' box and press 'apply' button.

Note 1.

Vignetting percentage: divide image to 5x5 blocks and calculate Y values of 4 corner's block and center block. The vignetting percentage defined to be the smallest percentage ratio of Y over center block.

Note 2.

Color shading: first to calculate the R/G, B/G, and R/B value and then calculate the difference ratio of these 3 factors of each block over center block.

5.AWB Calibration

Document Version

V1.0 (For Final HTML Tool)

5.1. Tool Overview

This AWB calibration tool is designed for ISP tuning of camera and imaging devices. It supports grayscale card region selection on RAW images that have undergone Demosaic and LSC (Lens Shading Correction), automatically computes RGB channel mean values, and supports data logging and storage for seven standard light sources, fulfilling core AWB calibration data acquisition needs.

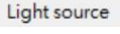
Core Functions

- Load and preview RAW images corrected with Demosaic+LSC; drag the mouse to select the grayscale card region and automatically compute accurate R/G/B channel mean values.
- Independently store RGB mean values for seven standard light sources.
- Visual interface, easy to operate, requires no software installation.

5.2. Interface Function Introduction

The tool interface is divided into three core areas with clear layout and convenient operation:

5.2.1. Control Area (Top)

Control	Function Description
Image Upload Button 	Select and load RAW images processed with Demosaic+LSC correction
Start Selecting Gray Card 	Enable mouse selection mode for selecting the grayscale card area
Light Source Menu 	Select the target calibration light source (D75/D65/D50/CWF/TL84/A/H)

5.2.2. Image Preview Area (Left)

Display the uploaded RAW image

Support mouse drag selection, red box is the selected area

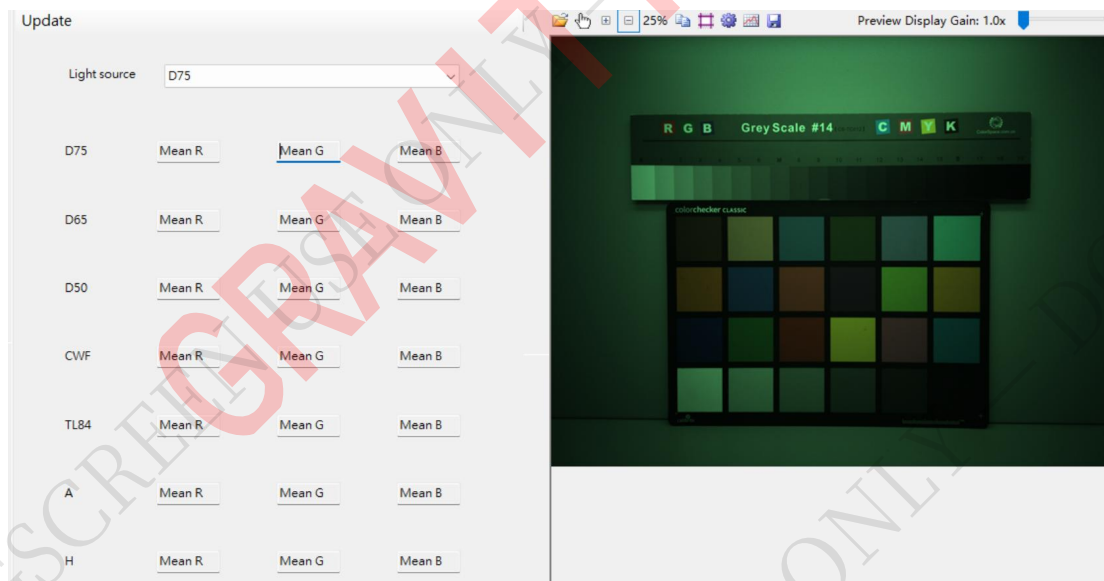


Fig 5.1 tool interface

5.2.3. Light Source Data Area

Seven standard light source modules are built in (D75, D65, D50, CWF, TL84, A, H). Each module includes three text boxes to store the R, G, and B mean values for the corresponding light source.

5.3. Standard Operation Procedure (SOP)

Step 1: Upload Calibration Image

- Click the image upload button in the control area
- Select a local RAW image processed with Demosaic + LSC correction
- The image is automatically loaded to the preview area

Step 2: Enable Selection Mode

- Click the **【Start Selecting Gray Card】** button, a confirmation dialog will appear to enable selection.

Step 3: Select Gray Card Area

- On the left image, hold down the left mouse button and drag to select the effective area of the standard grayscale card (the red box is the selected range). After releasing the mouse, the tool automatically calculates the RGB mean values of the selected area and pops up a calculation result prompt.

Step 4: Select Target Light Source

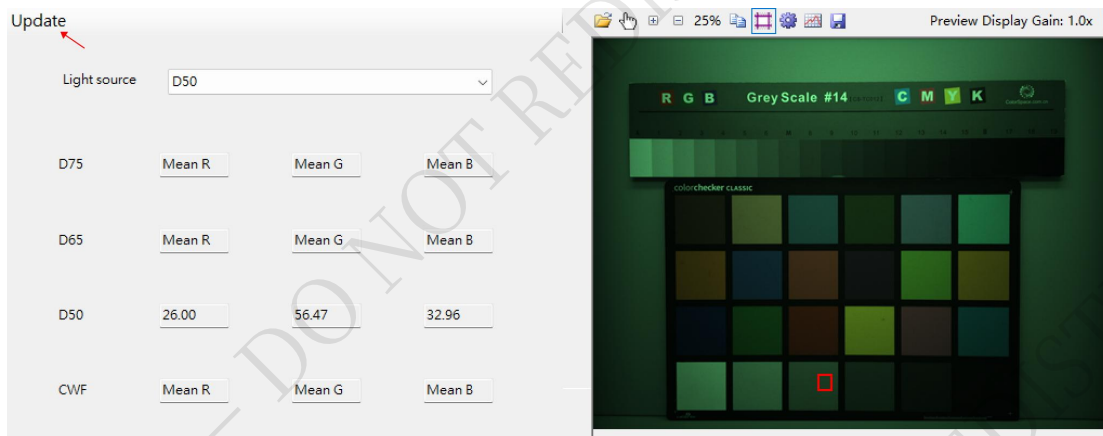
- In the drop-down menu of the control area, select the calibration light source corresponding to the current image (such as D65, A light, etc.).

Step 5: Save Data to Corresponding Light Source

- The calculated RGB mean values will be automatically written into the R/G/B text boxes of the corresponding light source on the right.

Step 6: Multi-light Source Calibration (Cycle Operation)

- Switch RAW images of different light sources, repeat steps 2 ~ 6 to complete the collection of RGB mean values of 7 light sources. Click Update to save all data after calibration.



5.4. Detailed Function Description

Data Calculation Rules

The tool automatically counts all pixels in the selected area:

- R channel average value
- G channel average value
- B channel average value

The numerical precision is retained to 2 decimal places, meeting the AWB calibration accuracy requirements.

5.5. Notes

- Image Requirements: Use only images processed with Demosaic (Demosaicing) and LSC (Lens Shading Correction). Uncorrected images will result in data errors.
- Selection Specification: The selected area must completely cover the neutral gray area of the grayscale card, avoiding background and edge noise.
- Light Source Matching: The uploaded RAW image must correspond to the light source selected in the drop-down menu one-to-one. Data mismatch is prohibited.
- Data Modification: The values in the text boxes support manual modification, which can be used to fine-tune calibration data.

5.6. FAQ Troubleshooting

Problem Phenomenon	Solution
No calculation result after selection	Check whether the image is loaded normally, re-select a larger effective area
Data not written to corresponding light source	Confirm that the target light source has been selected in the drop-down menu and the selection calculation has been completed
No red display in selection	Click 【Start Selecting Gray Card】 again to enable the selection mode

6. ISP pipeline tuning

Introduction of ISP pipeline tuning

In this section, tuning principles and common tool components are introduced.

6.1. OB Calibration

1. Cover the sensor, capture and save the RAW image.
2. Open the file.



Fig 6.1 File

3. Load the RAW image.



Fig 6.2 RAW image

4. Click "Set".

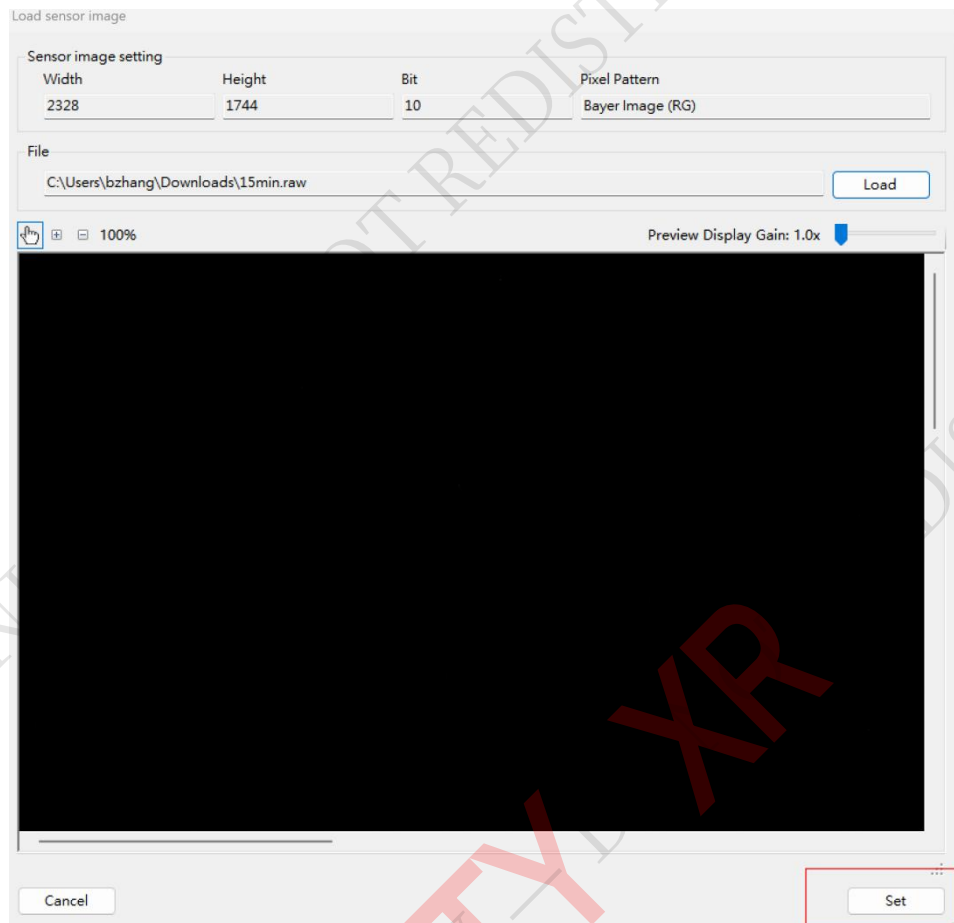


Fig 6.3 Set

5. Output the 16-bit OB values for the four channels R/GR/GB/B, then click "Update".

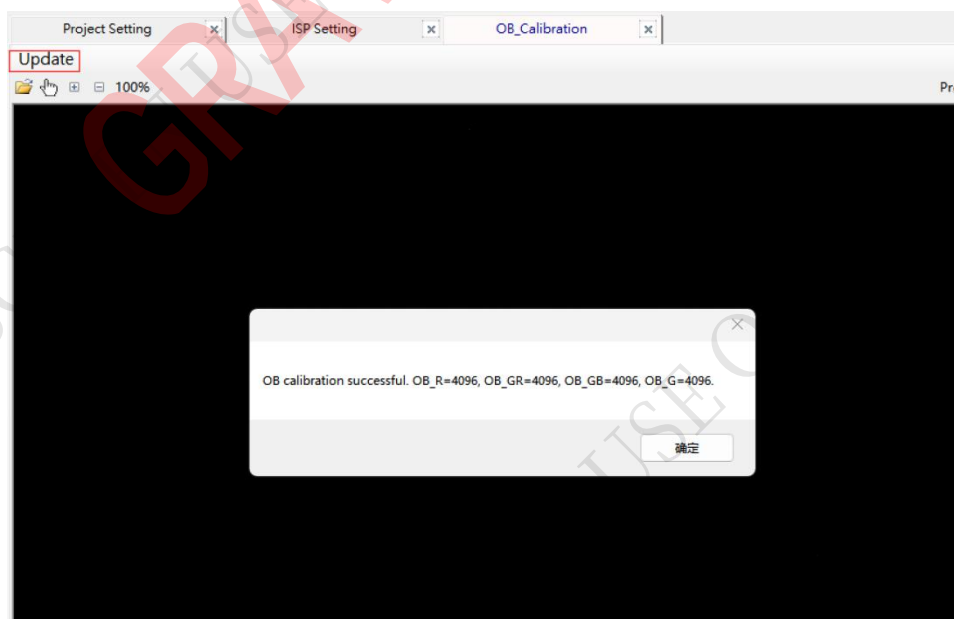


Fig 6.4 Update

6. The results are automatically saved to the Black Offset field in Project Settings.

The screenshot shows the 'Project Setting' dialog box with the 'Update' button highlighted. The 'Information' section contains 'Project Name' (Default Project) and 'Tuning Version' (0.001). The 'Image Setting' section contains 'Width' (4656), 'Height' (4656), 'Bit' (10), 'Pixel Pattern' (RGB : BG_GR), and 'Symbol _ meaning new line'. The 'Black Offset @16bit' section is highlighted with a red box and contains the following data:

Black Offset @16bit	Gr Channel	Gb Channel	B Channel
R Channel	4096	4096	4096

Fig 6.5 Project Setting

6.2. Auto Bad Pixel Correction (ABC)

Auto bad pixel correction (ABC) module deals with defective pixels which cannot be calibrated or found in advance such as hot pixel. In general, this kind of pixel is performed in single or cluster-wise on image. Besides, the case of bad pixel in ISP includes white bad pixel (WBP) and dark bad pixel (DBP) in Bayer domain.

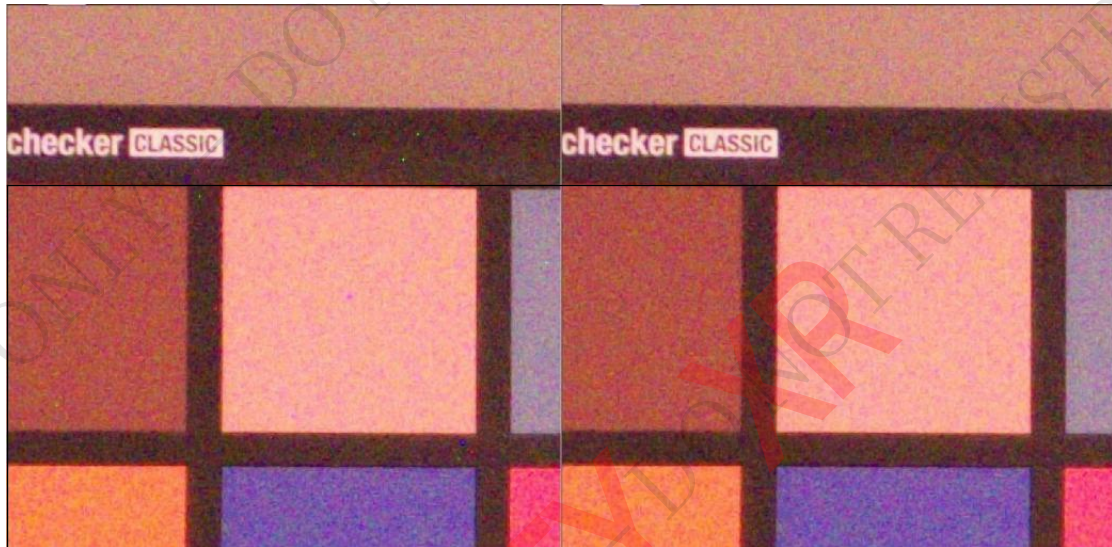


Fig 6.6 (Left) ABC off,defected pixel on image.
(Right) ABC on,defected pixel removed.

6.2.1. Tuning process:

- . Tuning order: Threshold → Cluster → Preservation.
- . Parameter setting (recommend):
 - Threshold: Level 6-11. (WBP/DBP)
 - Cluster: Level 1-4.
 - Preservation: Level 4-6. (WBP/DBP)

6.2.2. Threshold

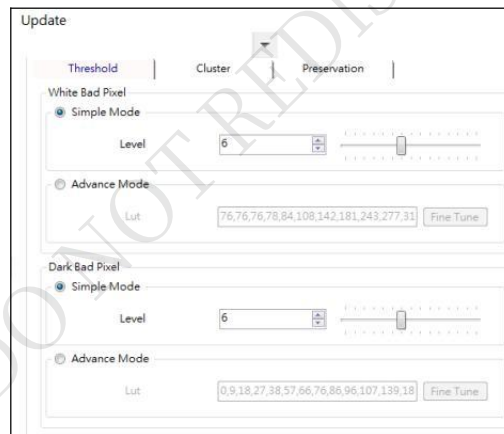


Fig 6.7 Tuning UI for ABC Threshold

- . Define the threshold of outlier that a pixel will determine to bad pixel.
- . Tune for WBP (white bad pixel) and DBP (dark bad pixel) in separate.
- . Level Range: 0-14; default= 6.
- . Low-level means only pixels with very high peak in signal be detected as bad pixel; as level increase, the pixel with middle or low peak also falls into outlier and detected as bad pixel. (Fig. 6 .8)



Fig 6.8 : (Left) Original Image. (Middle) Correct few of pixels in very high peak when tuning threshold with low level. (Right) Correct more pixels include middle or low peak in signal when tuning threshold with high level.

6.2.3. Cluster

Please refer to Fig. 6 .9 the cluster page.

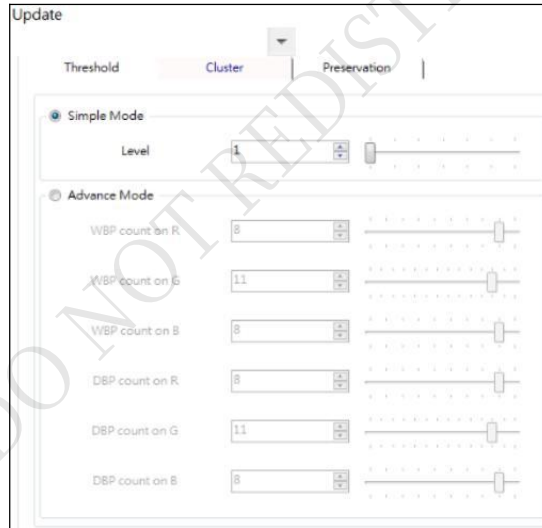


Fig 6.9 Tuning UI for ABC Cluster

- . Control the cluster level for handling cluster-wise bad pixel. (Fig. 6 .10)
- . Level Range: 1-7; default=1.
- . Enable to fine tune cluster parameter for R/G/B channel respectively in advance mode.



Fig 6.10 : (Left) Original Image. (Middle) Unable to remove cluster-wise pixel with weak cluster level. (Right) Remove cluster-wise pixels after applying adequate cluster level.

6.2.4. Preservation

Please refer to Fig 6.11. the preservation page.

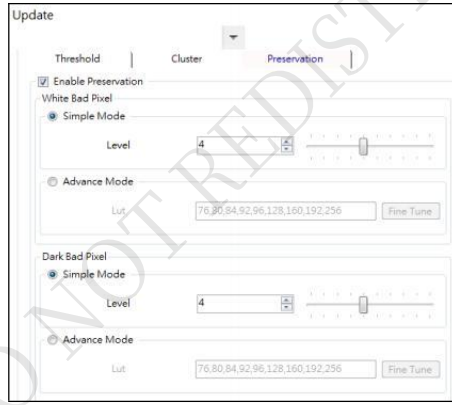


Fig 6.11 Tuning UI for ABC Preservation

- . Use to preserve some bad pixels with false detection such as spot-like feature (ex: stainless brush). (Fig. 6 .12)
- . Tune for WBP (white bad pixel) and DBP (dark bad pixel) in separate.
- . Level Range: 0-9.
- . Increase this level, more bad pixels originally with false detected would turn out to normal pixels.

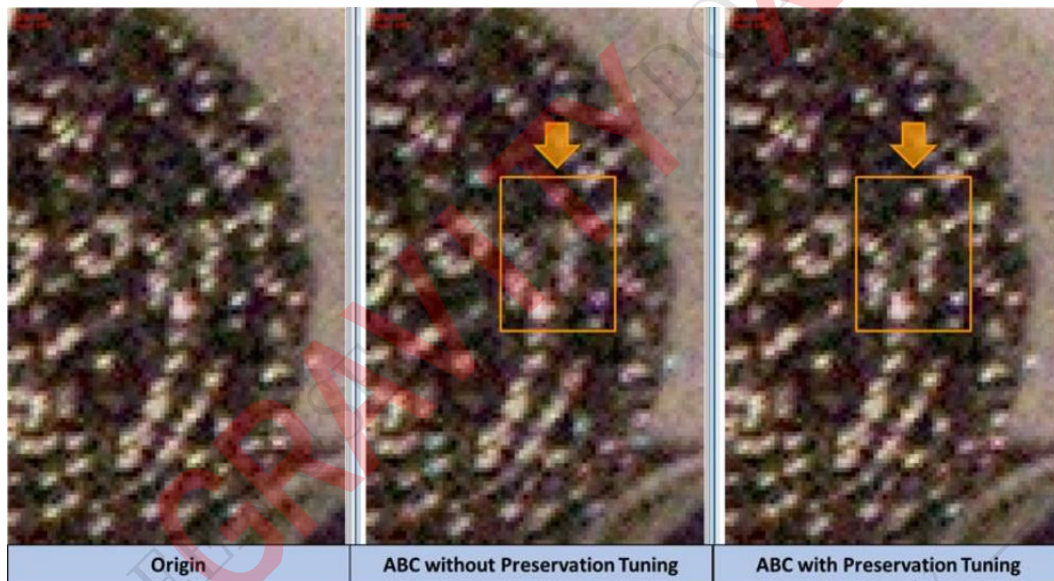


Fig 6.12: (Left) Original Image. (Middle) Detail loss on spot-like feature when taking no preservation tuning. (Right) Keep more detail after preservation tuning.

6.3. TNR

The main goal of TNR (temporal noise reduction) is to decrease temporal flicker noise while keeping detail as much as possible. In addition, when videos have local motion or global motion, Good TNR tuning can try to prevent obvious ghost effect and de-noise temporal flicker.

6.3.1. TNR Tuning Concept

- TNR's component
 - Temporal filter: Naive de-noise temporal flicker noise, but in motion region, this filter will generate ghost image.
 - Spatial filter: mainly use in motion region or when temporal filter's result isn't good. But this filter can't handle temporal flicker. And its de-noise type isn't good.
 - Confidence: decide using ratio of temporal filter or spatial filter.
 - Locally Refine filter:

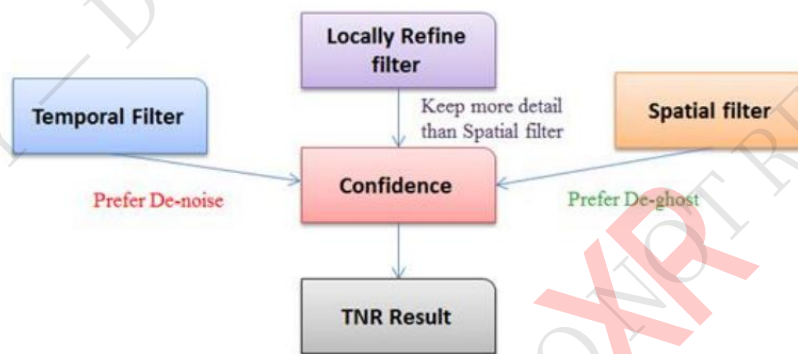


Fig 6.13 TNR tuning flow

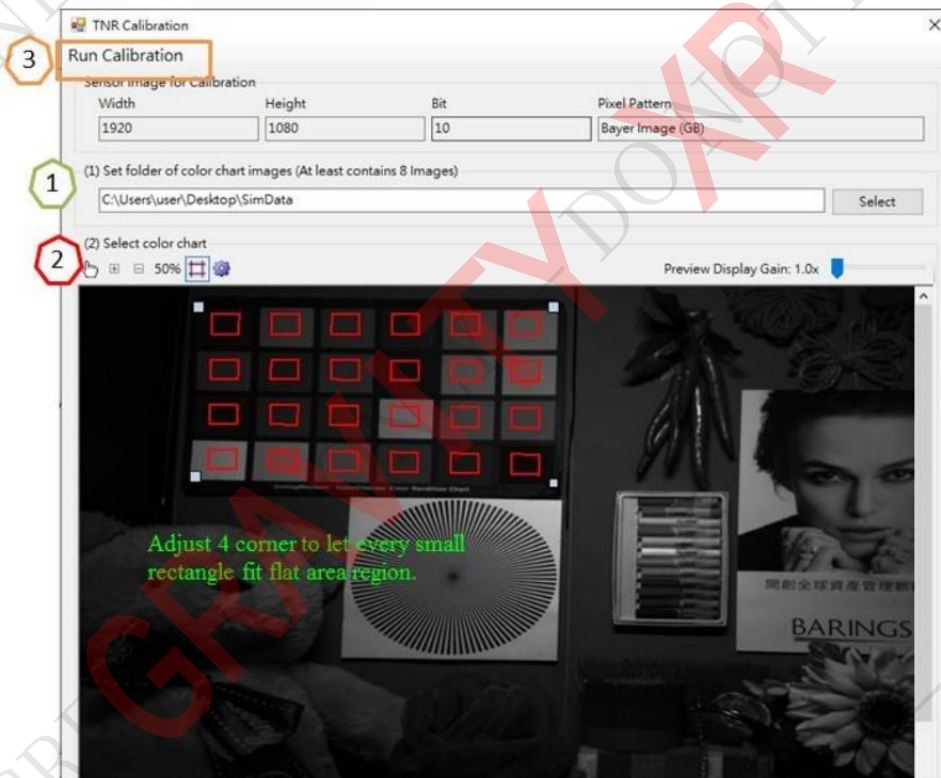
6.3.2. Tuning flow

- Tuning naive TNR(Temporal filter), see aggressively de-noise effect.
- Base on naive TNR effect to adjust spatial filter and local refine TNR. Avoid large gap between three filters.
- Adjust confidence map, use 24 color chart to get initial parameter of spatial filter & confidence map. Fine-tune initial parameter to get de-noise or de-ghost effect.
- Temporal filter's result is clean but in motion region might easily see ghost effect. Locally refine filter can keep more detail than spatial filter. But the result has more noise compare with temporal filter.

- Color chart calibration
 - Capture static 24 color chart's frame (Base on different ISO or noise level)(frame number > 8).
 - Select color chart regions(manual adjust), measure noise standard deviation & related intensity.
 - Use standard deviation & intensity to calculate linear curve fitting($y = ax + b$, y is variance, x is intensity).
 - In dark area or light area, need to manual fine-tune because curve fitting have its limitation during extrapolation case.
 - Use motion scene to verify or fine-tune depend on user's need.

6.3.3. TNR Calibration flow

1. Load static raw image folder(>= 8 frames)
2. Select rectangle(manual adjust 4 corner)
3. Run calibration
4. Pre-calibration value will auto set in current UI



4

Denoise Threshold R	0,20,30,44,54,63	Fine Tune
Denoise Threshold G	0,21,30,43,53,61	Fine Tune
Denoise Threshold B	0,20,30,48,53,61	Fine Tune

STD R	0,0,5,9,12,15,17,18,20,22,23,24,26,27,2	Fine Tune
STD G	0,4,8,11,14,16,17,19,21,22,23,24,26,27	Fine Tune
STD B	0,0,8,11,13,15,17,19,20,22,23,24,26,27	Fine Tune

Fig 6. 14 TNR Calibration Step

6.3.4. Capture static raw sequence image

1. Put color chart in center region
2. Uniform lighting
3. Capture in certain ISO gain, ex: ISO 100, ISO 400.....
4. Capture multiple raw sequence(≥ 10)



Fig 6.15 Capture scene sample

6.3.5. Tuning stage UI

- Decide previous frame's weight
- Range:[0~127]
- Guideline: start from 64, gradually increase previous frame weight. It is a convergent process. User can tune fast convergence or slow convergence. Frame weight large => use more previous image (clean, but easily get ghost)
- Frame weight low => use more current image(noisy, but can keep more detail)

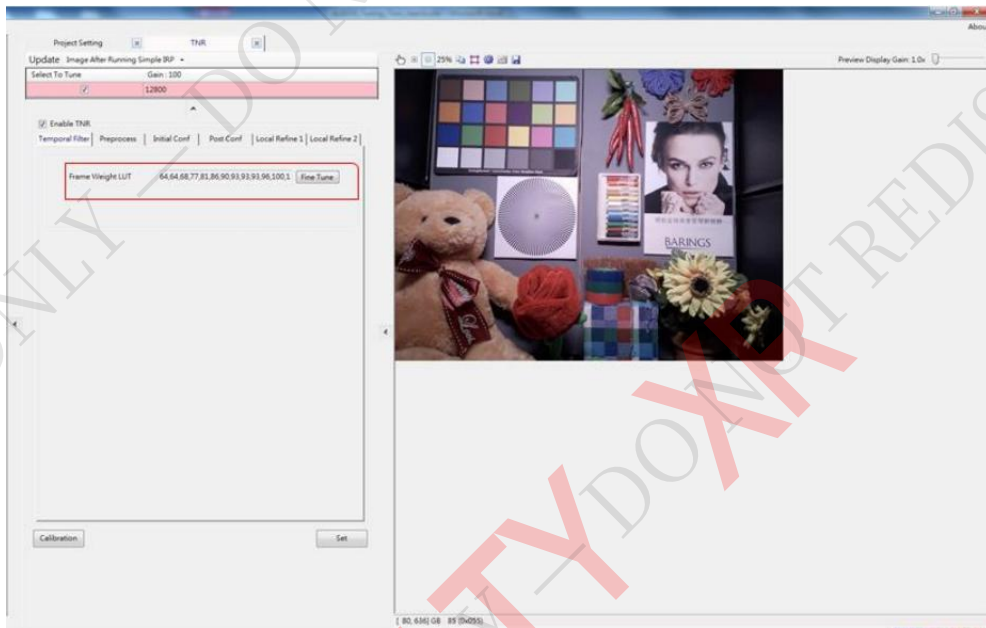


Fig 6.16 Tuning temporal filter

■ Sequence frame weight tuning

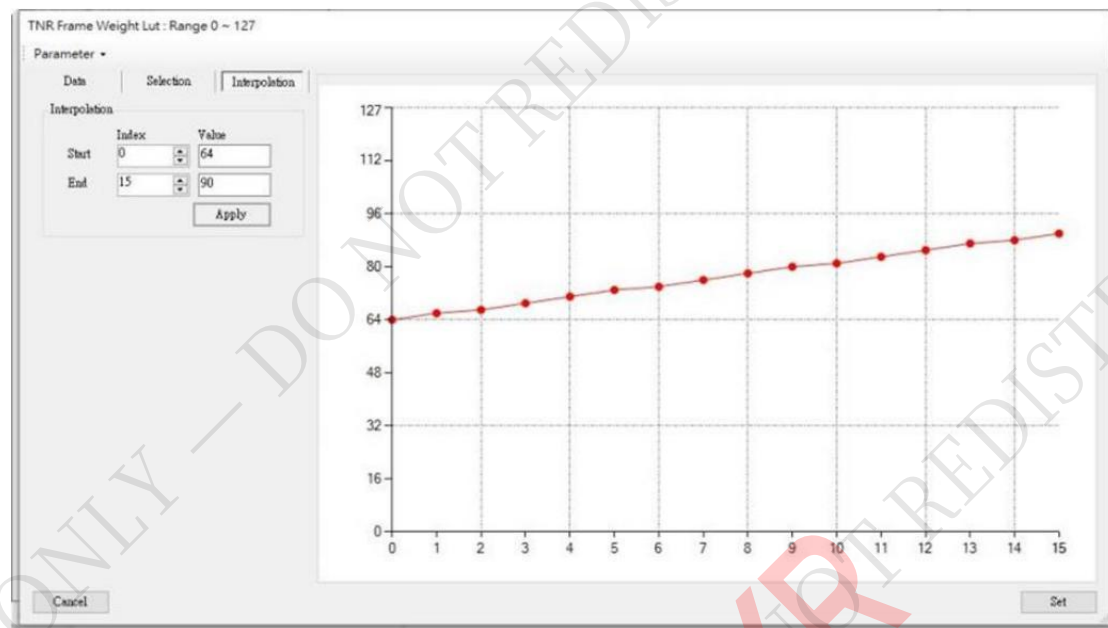


Fig 6.17 Interpolation of sequence frame weight tuning

Ps: Avoid tuning previous frame ratio too large. When image become clean, Bayer codec will compress and de-compress unaccepted pattern (> 110)

6.3.7. Tuning – spatial filter

- Blending weight
 - ☐ Reference intensity
 - ☐ Control ratio of pure average intensity & original intensity
 - ☐ 0 : use more pure average intensity ; 127 : use more original intensity
- De-noise Threshold R/G/B
 - ☐ Use calibration step to pre-define initial threshold
 - ☐ Fine-tune threshold depend on image quality
 - ☐ Intensity node : 0, 128, 256, 512, 768, 1023
 - ☐ Set threshold large => de-noise strong
 - ☐ Set threshold small => de-noise weak
- Intensity Weighting
 - ☐ Define user wanted de-noise's intensity range
 - ☐ Intensity node range [0~255] [256~511] [512~767] [768~1023]
 - ☐ Set large => de-noise strong, but will look like average filter, detail will lose.

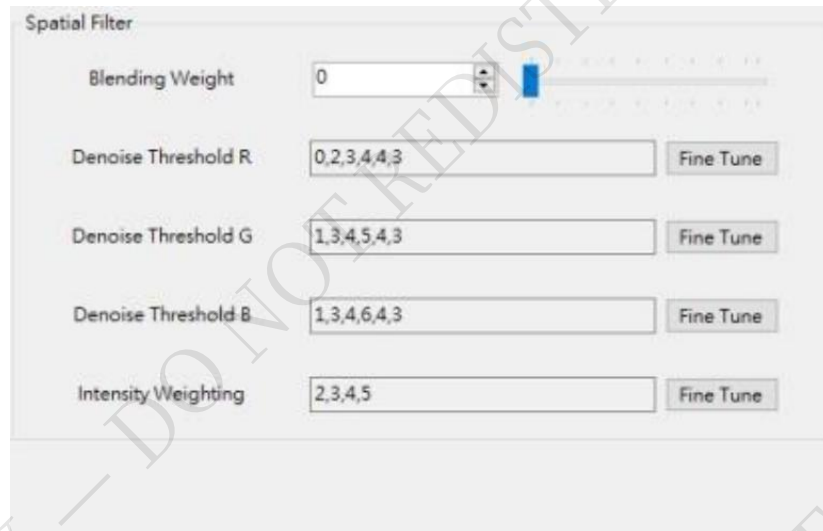


Fig 6.18 Parameter of Spatial filter

Ps: Observe certain intensity

Noisy -> increase this related intensity de-noise threshold

Blur -> decrease this related intensity de-noise threshold

Increase intensity weight can also de-noise, but it is too aggressive. Need to careful tune this value.

- MAE(Edge map)
 - ☐ Apply different de-noise strength on flat area & edge area.
 - MAE Kernel size
 - ☐ 3X3 is used for small image
 - ☐ 5X5 is used for large image & noisy image
 - MAE Gain (MAE Preserve LUT)
 - ☐ MAE Gain directly apply on de-noise threshold R/G/B. It can increase de-noise threshold or decrease de-noise threshold.
 - ☐ Range : [0~63]
 - ☐ MAE node value : 0, 8, 16, 24, 32, 40, 48, 56, 63
 - ☐ 16 represent 1x gain, 32 represent 2x gain etc.
 - ☐ MAE value large => Edge region => Keep detail
 - ☐ MAE value small => might be flat area => De-noise
- Ex: In flat area (MAE node value near 0), increase de-noise strength.
 In edge region (MAE node near 63), decrease de-noise strength.

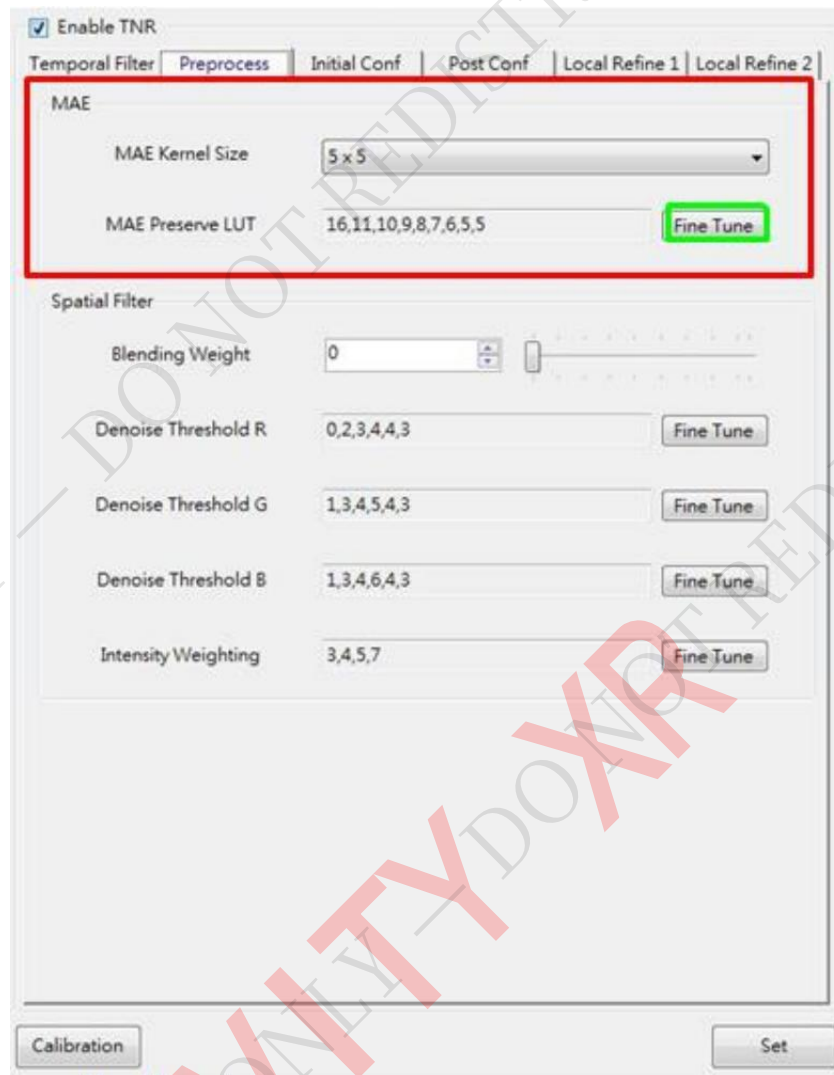


Fig 6.19 Parameter of MAE (For Edge preserving Tuning)

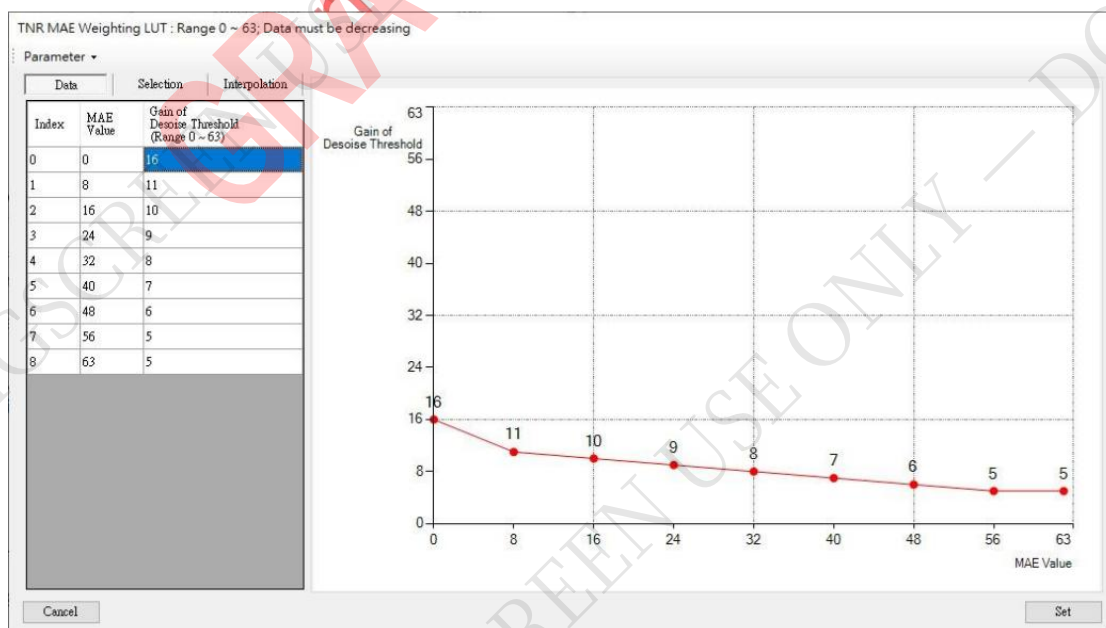


Fig 6.20 MAE LUT

6.3.8. Tuning – Confidence

- Decide ratio of temporal filter or spatial filter or locally refine TNR filter.
- Confidence Mask size
 - ☐ 3x3 is for quite clean image
 - ☐ 5x5 is for noisy image
- Color difference
 - ☐ Mode 0 : Color Diff, pure pixel difference(for quite clean image) => de-ghost
 - ☐ Mode 1: Color Mean Diff, use N x N average, for noisy image => strongly de-noise effect.
 - ☐ Mode 2: Pre img – Cur img, use N x N average of current image, keep previous image unchanged. => slightly de-noise.
 - ☐ Mode 3: Pre img – Blend, use blending result of current image (with blur) & current image (original) via MAE value. => de-noise slightly weaker.
- Discount LUT
 - ☐ Use gain to shrink or enlarge color difference to avoid ghost or strongly de-noise flat area.
 - ☐ Apply on color difference's gain, 64 represent 1x gain.
 - ☐ Apply different gain on flat area/ Texture/ Edge.

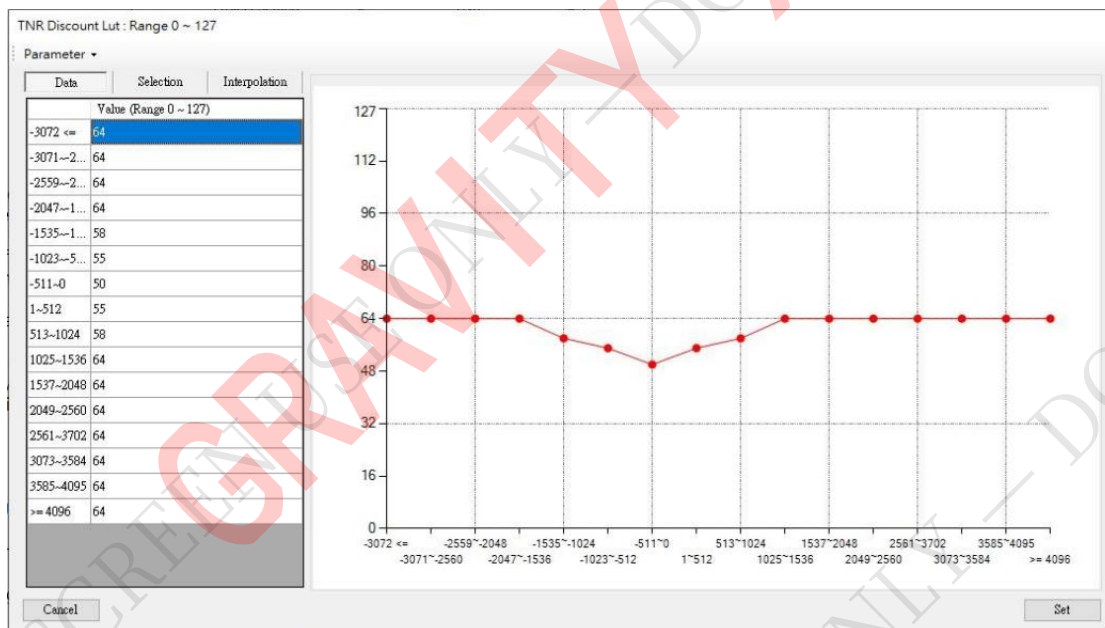


Fig 6.21 Discount LUT



Fig 6.22 Variance difference for discount LUT tuning

Variance

- Lookup STD
 - ☐ Reference STD's intensity
 - ☐ Avg Intensity : for noisy image
 - ☐ MAE blending : for clean image
- STD R/G/B
 - ☐ Decide R/G/B intensity's mapping STD value
 - ☐ Use pre-calibration mechanism to set, and fine-tune.
 - ☐ STD is large, confidence is high => easy for de-noise
 - ☐ STD is small, confidence is low => easy for de-ghost.
 - ☐ STD R/G/B intensity node : 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, 256, 288, 320, 352, 384, 416, 448, 480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800, 832, 864, 896, 928, 960, 992, 1023.
- Variance
 - ☐ Use calibrate variance or max(Calibrate Var, Measure Var)
 - ☐ Calibrate Var: user-defined value, use this mode & STD R/G/B, user can tune aggressively de-ghost style.
 - ☐ Max (Calibrate Var, Measure Var) : edge region will de-noise easily.
- Anti-ghost
 - ☐ Color difference & Variance's gap. Use gap to find mapping confidence.
 - ☐ Need to consider consistency.

The screenshot shows a window titled "Variance" with the following controls:

- Lookup STD:** A dropdown menu currently set to "MAE blending".
- STD R:** A text input field containing the values "1,1,1,3,4,5,7,10,11,14,14,14,16,17,18,19". To its right is a "Fine Tune" button.
- STD G:** A text input field containing the values "3,4,5,6,7,10,10,10,11,12,13,13,14,14,15". To its right is a "Fine Tune" button.
- STD B:** A text input field containing the values "2,3,4,5,6,7,9,10,12,13,14,15,16,17,18,19". To its right is a "Fine Tune" button.
- Variance:** A dropdown menu currently set to "MAX(Calibrate VAR, Measure VAR)".
- Anti-Ghost:** A text input field containing the values "127,127,126,125,120,119,110,100,95,81". To its right is a "Fine Tune" button.

Fig 6.23 Tuning Variance (STD) Parameter

Post Confidence

- Post Processing Mask size
 - ☐ 3X3 & 5X5
- Post Processing Mode
 - ☐ Min (Strongly de-ghost, but need to consider discontinuity)
 - ☐ Max (Strongly de-noise, notice ghost effect)
 - ☐ Blend Avg filter (Neutral method)
 - ☐ Blend min & Max (need to carefully tune Anti-Ghost LUT to balance)
 - ☐ Blend ori & min (Slightly prefer de-ghost) (recommend default)
 - ☐ Blend ori & max (Slightly prefer de-noise)
- Scale / offset
 - ☐ Decide confidence's allowable range. Filter small value(offset) & enlarge confidence gap(Scale)
 - ☐ Scale * confidence – offset, clip into range [0~127]
- Confidence Lower Bound/ Upper Bound
 - ☐ Use TNR's minimum/maximum ratio
 - ☐ Set Confidence Lower = 0 & Confidence Upper = 0 => Tune Spatial filter result or locally refine TNR filter result.
 - ☐ Set Confidence Lower = 127 & Confidence Upper = 127 => Tune Temporal filter.
 - ☐ In normal case, set lower bound = 0 & upper bound = 127
 - ☐ This parameter can let user quickly see each module's effect(Temporal / spatial/Locally Refine TNR)

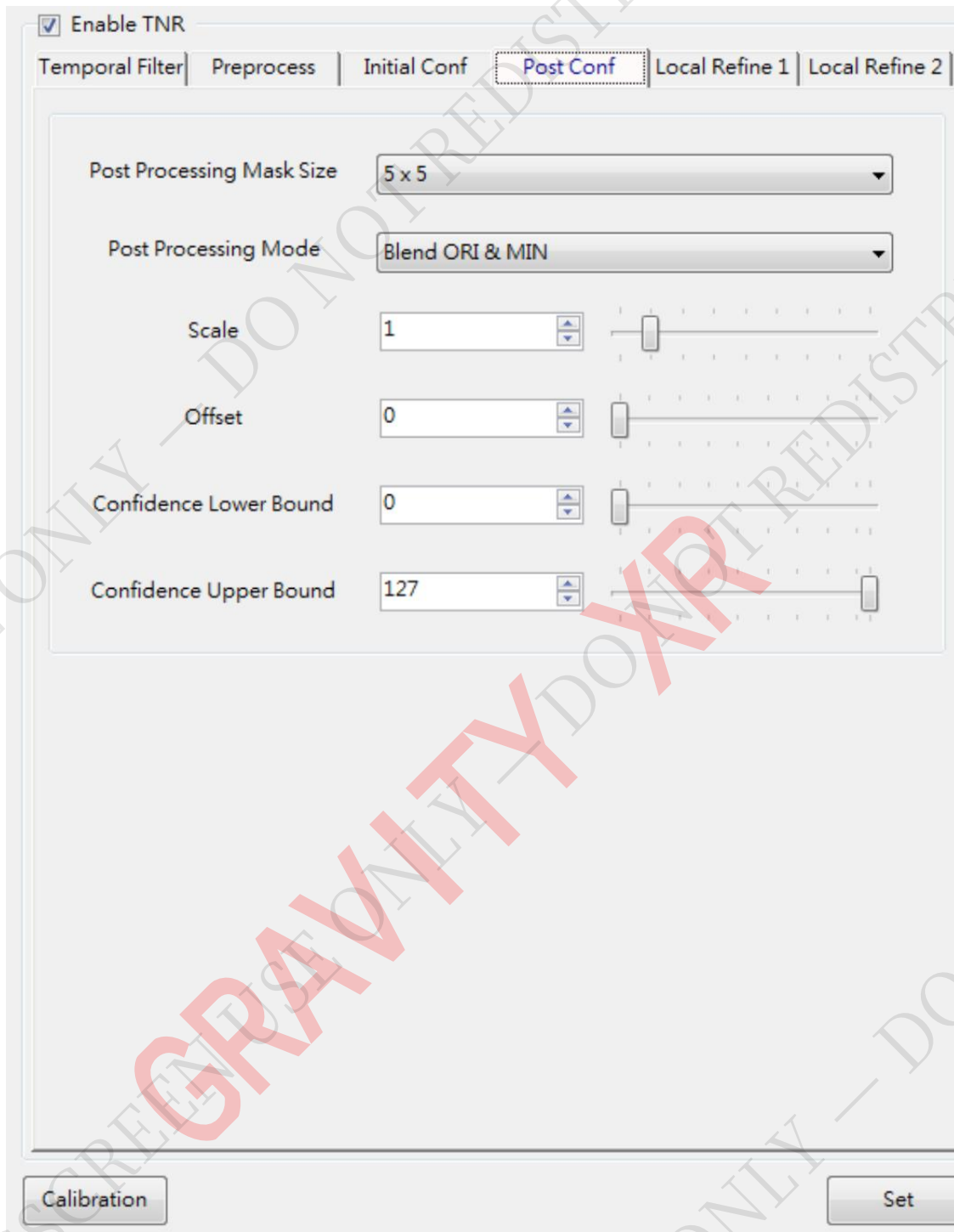


Fig 6 .24 Post Confidence parameter

6.3.9. Tuning – Locally Refine TNR filter

- Locally Refine TNR (LRTNR) can preserve more detail than pure spatial filter. We usually use this filter in motion region, and use pure TNR in static region.

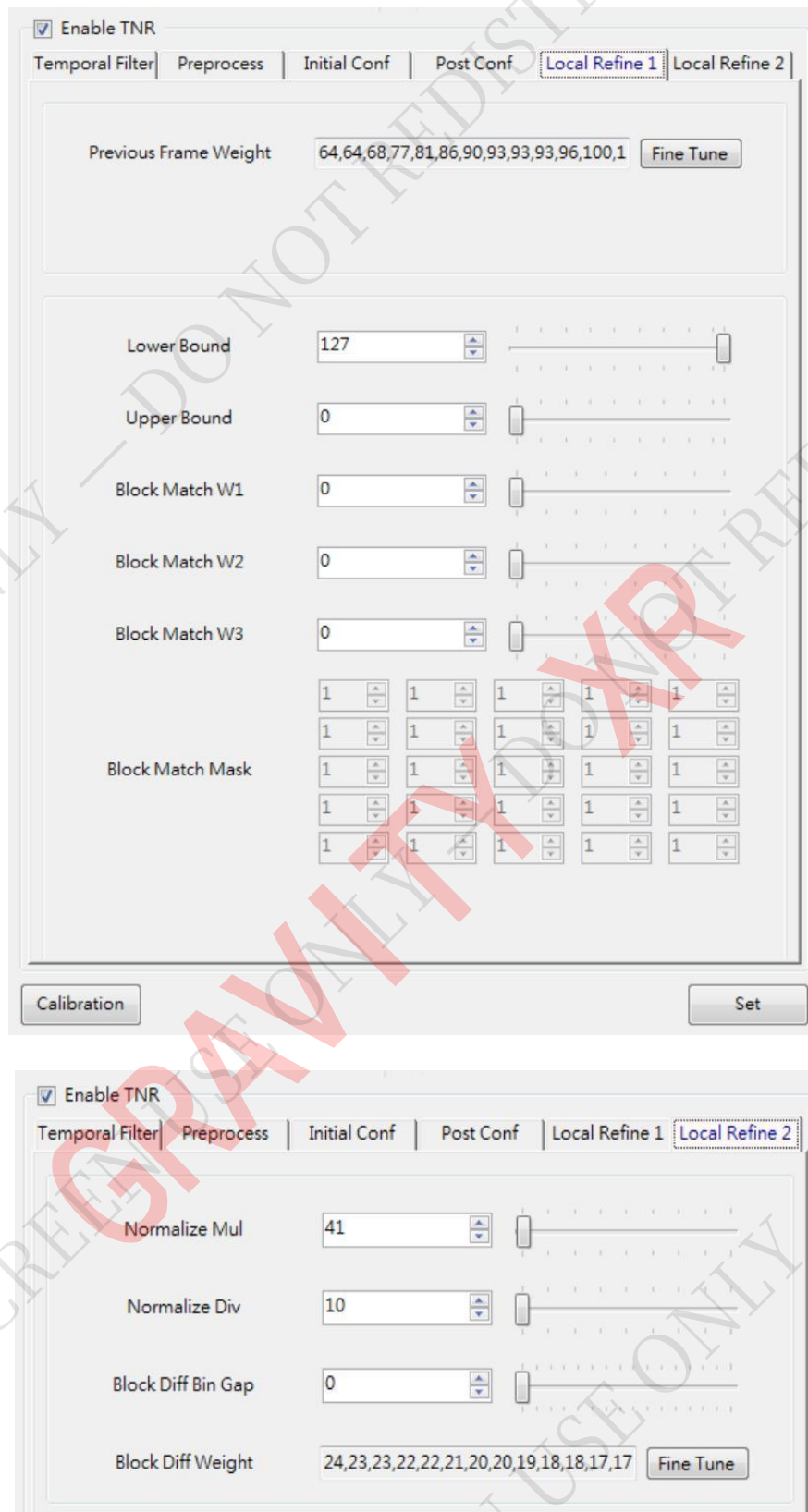


Fig 6.25 Locally Refine TNR parameter. Top : Local Refine 1; Bottom Local Refine 2

- Previous Frame Weight
 - Locally Refine TNR's previous frame weight, this parameter is similar to Temporal Filter's Frame Weight LUT.
- Lower Bound/ Upper Bound

- ☐ Use Locally Refine TNR's lower bound / upper bound.
- ☐ Similar to confidence lower bound/upper bound.
- ☐ Set lower bound = 0 & upper bound = 0 => don't use LRTNR result, it only uses pure TNR & spatial filter.
- ☐ Set lower bound = 127 & upper bound = 127 => use LRTNR
- ☐ Must meet criteria : Confidence Lower bound + Lower bound < 128
- ☐ Must meet criteria : Confidence Upper bound + Upper bound < 128
- Block Match W1/W2/W3 (Block Match Mask)
 - ☐ For 5x5 patch match mask weighting.
 - ☐ W1/W2/W3 is power, the actual weight is 2^{W1} or 2^{W2} or 2^{W3} , ex : set W1 = 0, weight is 1.
 - ☐ $SH[0] = 2^{W1}$, $SH[1] = 2^{W2}$, $SH[2] = 2^{W3}$

SH[2]	SH[2]	SH[2]	SH[2]	SH[2]
SH[2]	SH[1]	SH[1]	SH[1]	SH[2]
SH[2]	SH[1]	SH[0]	SH[1]	SH[2]
SH[2]	SH[1]	SH[1]	SH[1]	SH[2]
SH[2]	SH[2]	SH[2]	SH[2]	SH[2]

Fig. 6.26 SH is a weighting table for each pixel difference $|Src_i - Ref_i|$

- Normalize Mul/ Normalize Div
 - ☐ Block match normalization's hardware coefficient
 - ☐ Ex: if all weight is 1 in 5x5 mask, it will divide 25. We use $41 / 2^{10} \approx 1/25$. Normalize Mul will set 41 and Normalize Div will set 10.
 - ☐ Normalize Mul range: [0~8191] ; Normalize Div range : [0~13].
- Block DiffBin Gap
 - ☐ Control patch difference's gap, ex: patch difference << 1, which mean difference 100 and 101 will map to the same weight.
 - ☐ Range : [0~15]
- Block diff Weight
 - ☐ Range : [0~24]
 - ☐ Actual weight is 2's power, block diff Weight is power.
 - ☐ Horizontal axis is block difference, ex: index 0 => Block difference is 0.
 - ☐ Block difference is large => two patch are not similar, their mapping weight need to set small.
 - ☐ Block difference range [0~31].

6.3.10. Batch run Tuning (Debug information)

- Batch run tuning for TNR can generate debug information to help user fine-tune TNR's result.
- Step:
 1. Generate batch run of TNR simulate data
 2. Load sim_data for further fine-tune.

Batch Run Step

1. Select "Batch Run"
2. Load sequence of Bayer RAW
3. Generate setting files, (ex : ISP settings -> Condition(set parameter here))
4. After generating setting files, batch run TNR's sequence result
5. Type : Consecutive frames
6. Run to stage : Raw TNR
7. Output : Simulate Data(Select for fine-tune usage)
8. Run

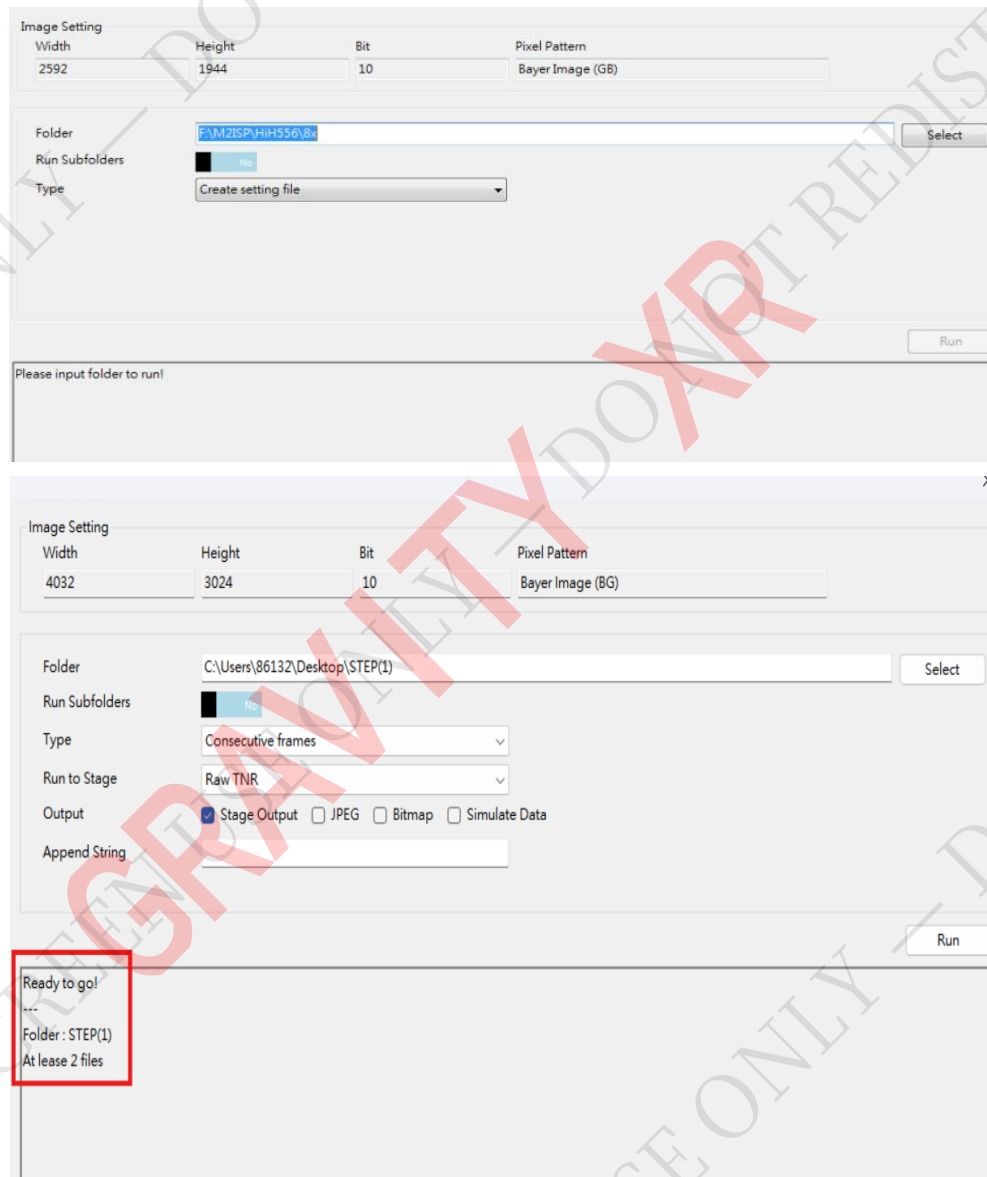


Fig 6.27 Batch run step

02_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,791 KB
03_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,549 KB
04_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,426 KB
05_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,330 KB
06_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,283 KB
07_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,229 KB
08_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,177 KB
09_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,153 KB
10_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,143 KB
11_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,141 KB
12_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,118 KB
13_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,104 KB
14_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,081 KB
15_5000K_Exp_015124_Gain_00800.sim_dat	2023/7/26 上午 1...	SIM_DAT 檔案	17,061 KB

Fig 6.28 Generate TNR sim_data sequence

Use simulate Data for further fine-tune

1. Select ISP setting
2. Simulate Data (Dump from Batch Run step)
3. Load *.sim_data format for fine-tune

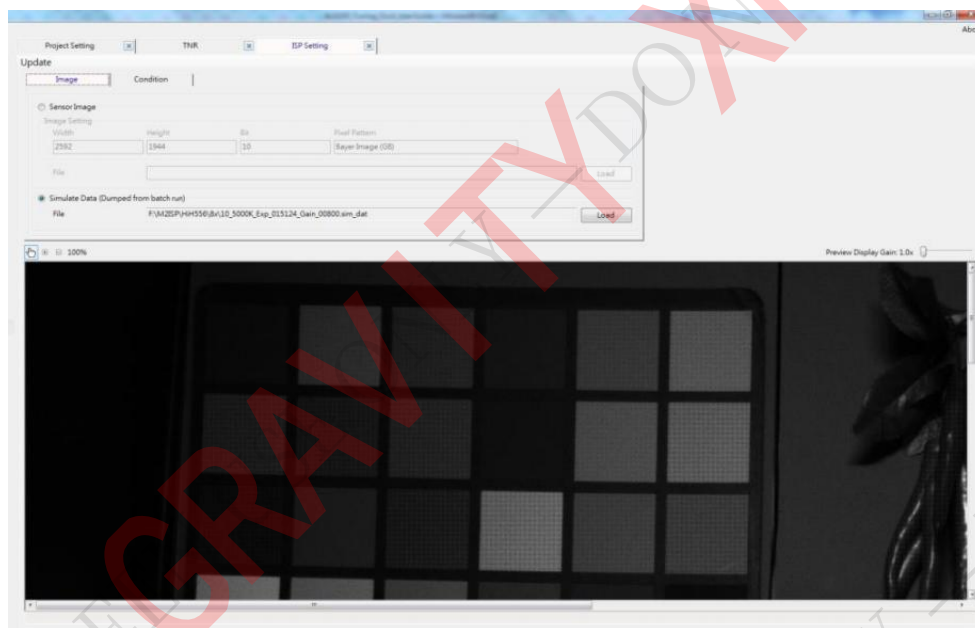


Fig 6.29 Load TNR simulate data for fine-tune

Debug information

1. Normal simulation
2. Image after running simple IRP
3. Spatial filter
4. Confidence map
5. Temporal filter
6. Local Refine

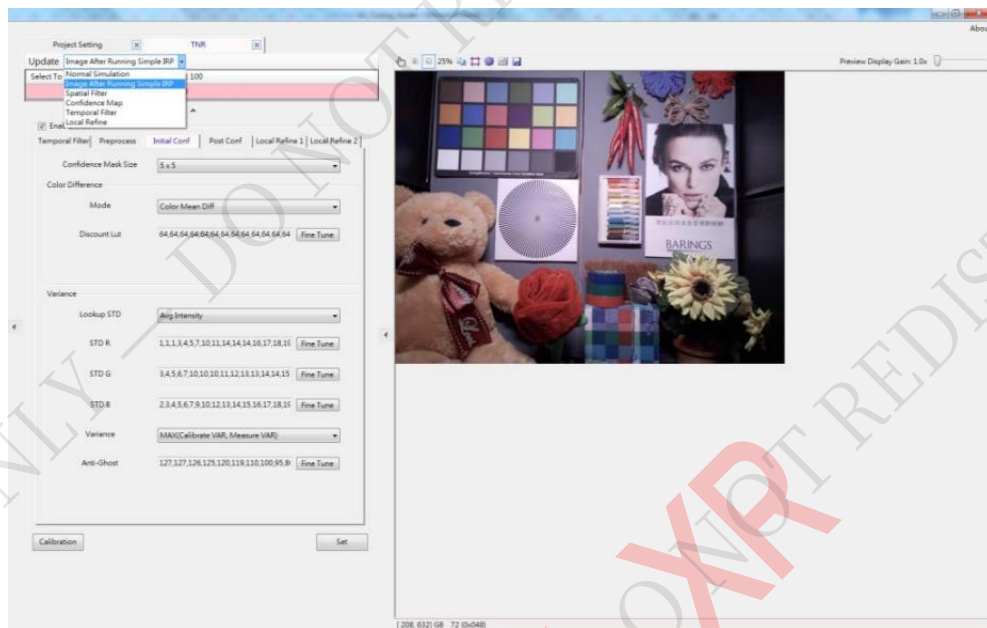


Fig 6.30 TNR debug information

6.4. SNR

- Spatial Noise Reduction (SNR) is a denoise technique in Bayer domain.
- Raw denoise kernel accepts only 10-bits raw image after TNR (Temporal Noise Reduction), the output is also 10-bits raw image.
- Tuning Order:
 - Local Denoise → Edge Filter → Pixel Compensation
- Local Denoise
 - Tune the denoise level for R、Gr、Gb、B channels.
 - Tune the denoise level for each pixel intensity zone.
- Edge Filter
 - Define the edge threshold to classify the edge direction.
 - Tune the denoise level for each edge direction.
- Pixel Compensation
 - Resume the signal for smooth or texture area.

- ❑ Resume the signal for each pixel intensity zone.

6.4.1. Local Denoise

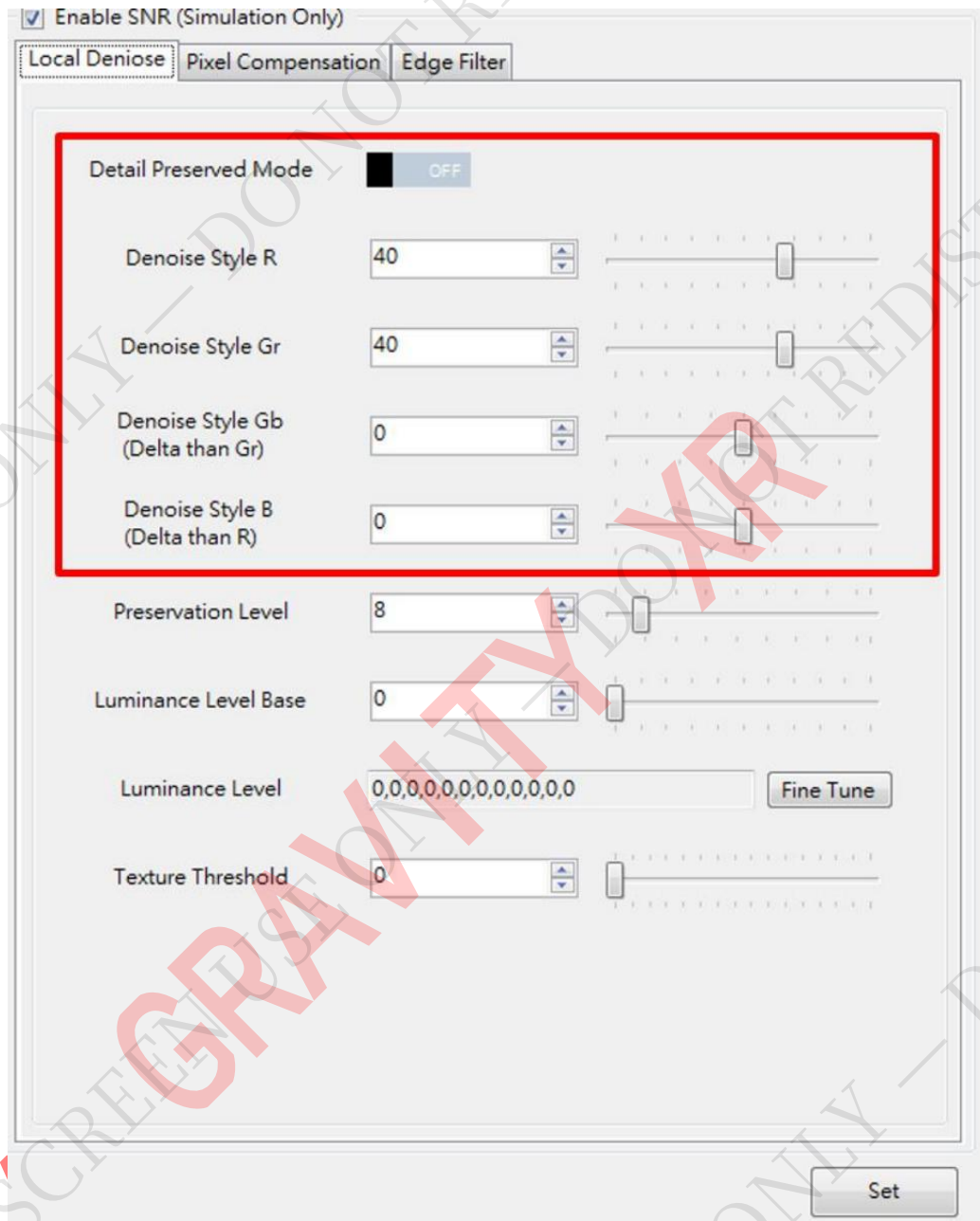


Fig 6.31 Local Denoise UI

- Detail Preserved Mode
 - ❑ On: preserve cluster detail.
 - ❑ Off: smooth cluster detail.
- Denoise Style R/Gr
 - ❑ Denoise level, increase this level, more noise reduction.
 - ❑ Default = 30
 - ❑ Range:0~60
- Denoise Style Gb/B

- ❑ The offset to the Denoise Style R/Gr
- ❑ increase this offset, more noise reduction than R/Gr channels.
- ❑ Default = 0
- ❑ Range: -24~24

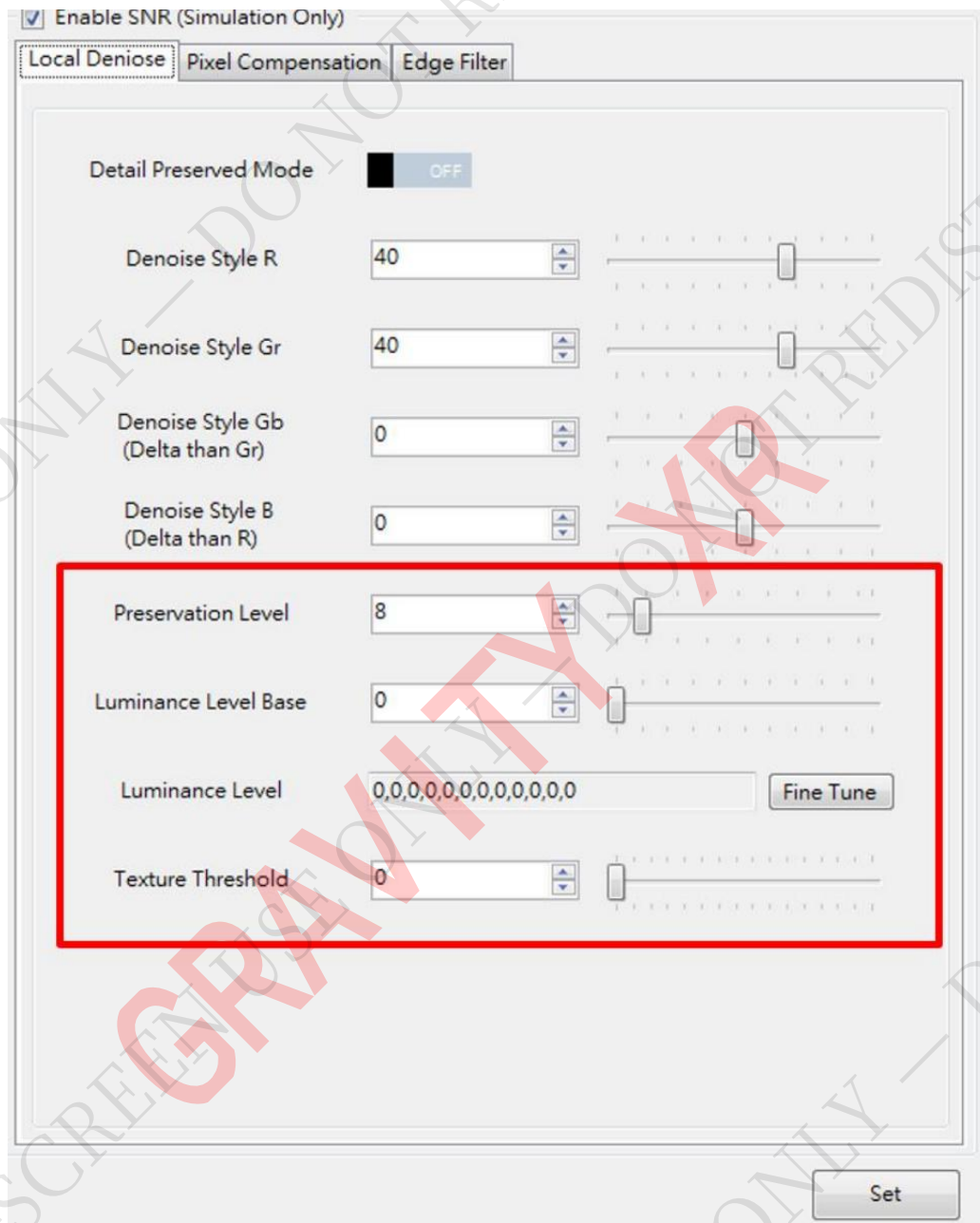


Fig 6.32 Local Denoise UI

- Preservation Level
 - ❑ Preserve detail level
 - ❑ decrease this value, keep more detail.
 - ❑ Increase this value, more noise reduction.
 - ❑ Range:0~76, default = 7
- Luminance Level Base
 - ❑ The base of denoise level to all luminance.
 - ❑ Increase this value, more noise reduction.

- ☐ Range:0~10, default = 0
- Luminance Level
 - ☐ The denoise level to each luminance zone.
 - ☐ Increase this value, more noise reduction.
 - ☐ Range:0~15, default = 0
- Texture Threshold
 - ☐ Increase this threshold, more pixels are classified into smooth regions.
 - ☐ Range:0~65535

6.4.2. Edge Filter

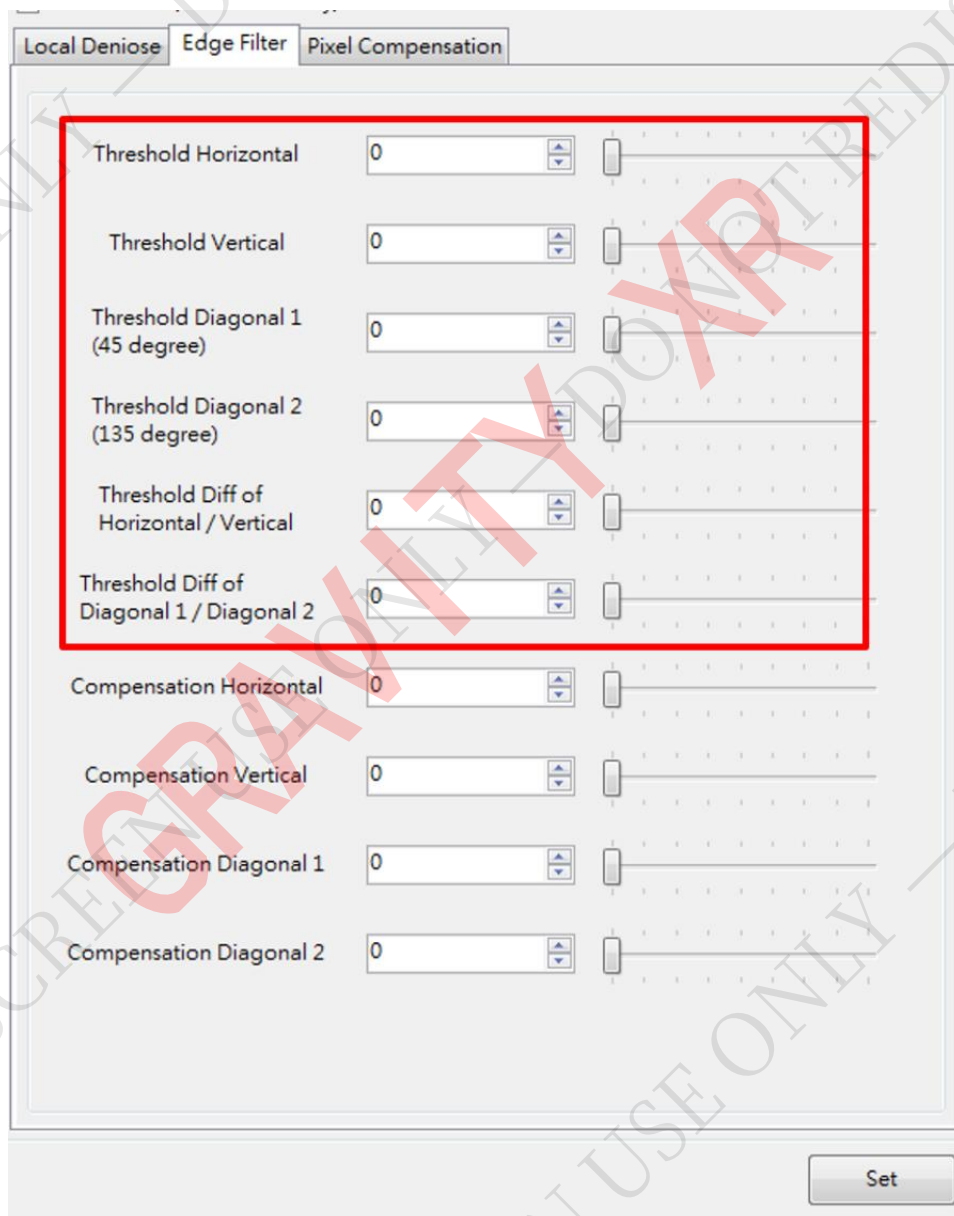


Fig 6.33 Edge Filter UI

- The threshold to classify the edge direction
 - ☐ Threshold Horizontal
 - ☐ Increase this threshold, more pixels are classified into horizontal

- direction.
 - Range:0~2047, default = 0
- ❑ Threshold Vertical
 - Increase this threshold, more pixels are classified into vertical direction.
 - Range:0~2047, default = 0
- ❑ Threshold Diagonal1(45 degree)
 - Increase this threshold, more pixels are classified into 45 degree direction.
 - Range:0~2047, default = 0
- ❑ Threshold Diagonal2(135 degree)
 - Increase this threshold, more pixels are classified into 135 degree direction.
 - Range:0~2047, default = 0
- ❑ Threshold Diff of Horizontal/Vertical
 - Increase this threshold, more pixels are easily classified into smooth area.
 - Range:0~2047, default = 0
- ❑ Threshold Diff of Diagonal1/Diagonal2
 - Increase this threshold, more pixels are easily classified into smooth area.
 - Range:0~2047, default = 0

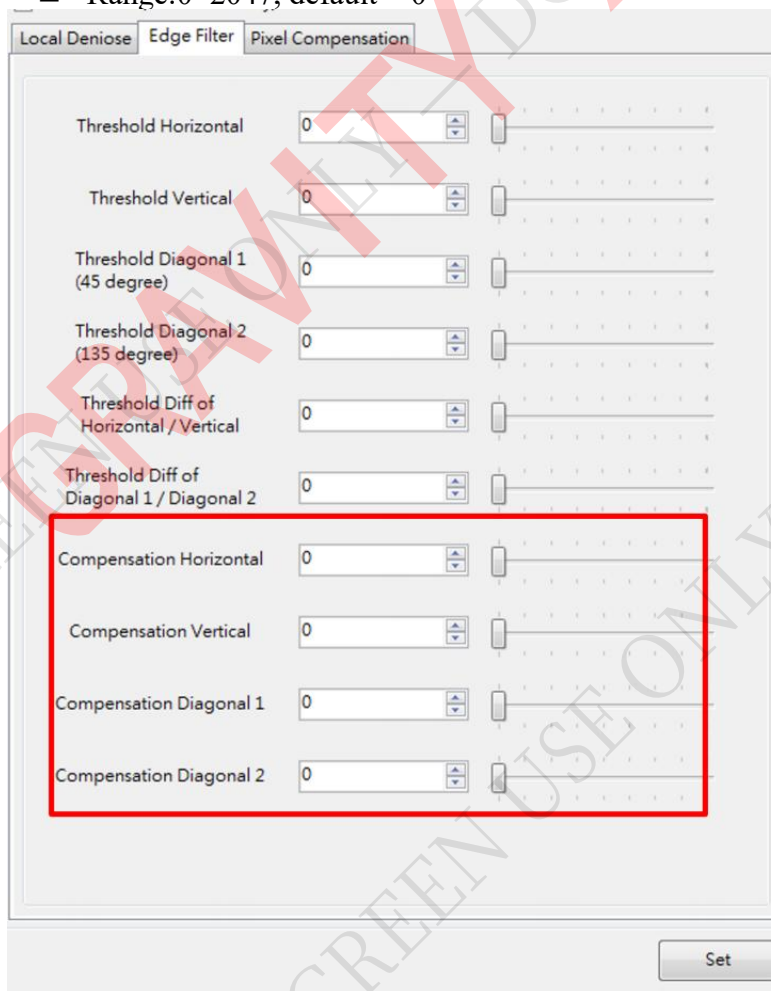


Fig 6.34 Edge Filter UI

- The de-noise level to each edge direction
 - Compensation Horizontal
 - Increase this level, more de-noise along the horizontal edge.
 - Range:0~24, default = 0
 - Compensation Vertical
 - Increase this threshold, more de-noise along the vertical edge.
 - Range:0~24, default = 0
 - Compensation Diagonal1(45-degree)
 - Increase this threshold, more de-noise along the 45-degree edge.
 - Range:0~24, default = 0
 - Compensation Diagonal2(135-degree)
 - Increase this threshold, more de-noise along the 135-degree edge.
 - Range:0~24, default = 0

6.4.3. Pixel Compensation

Please refer to Fig. 6 .35 the pixel compensation UI



Fig 6.35 Pixel Compensation UI

- Max Bound
 - The max limitation of resuming signal.
 - Range:0~1023, default = 10
- Smooth
 - The percentage to resume signal for smooth area.
 - Range:0~1023(100%), default = 0
- Texture
 - The percentage to resume signal for texture area.
 - Range:0~1023(100%), default = 0
- Compensation Level
 - The percentage to resume signal for each intensity zone.
 - Range:0~1023(100%), default = 0

6.5. IRP 1 CFAi

The purpose of CFAi is to convert Bayer pattern become full resolution RGB. Bayer pattern is placed RGB color filter in one plane but different position. Here needs algorithm to interpolate missing colors and induce parameters for controlling interpolation behavior. In below show the tuning parameters and follows.

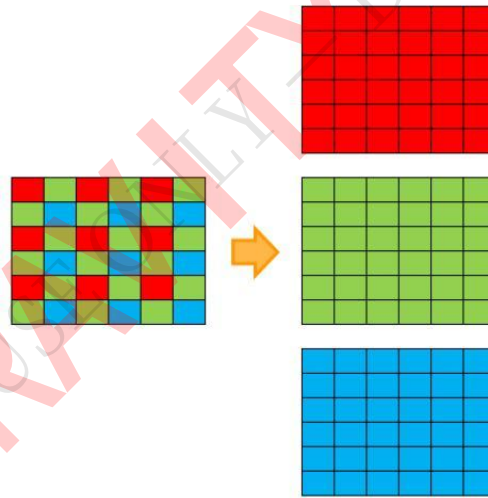


Fig 6.36 Bayer pattern to full resolution RGB

6.5.1. Tuning follow 1

Select Blur, Normal and Sharp Mask, where 0:blur, 1:normal, 2:sharp

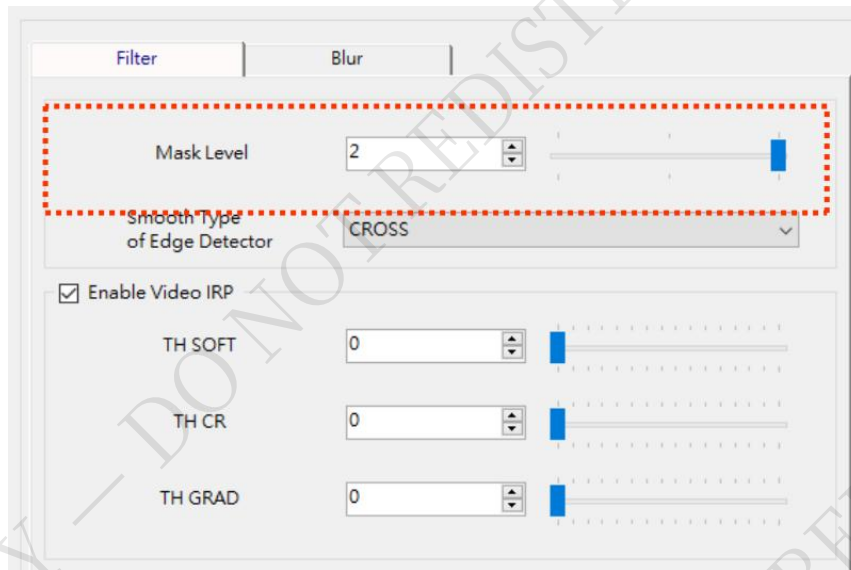


Fig 6.37 CFAi filter



Fig 6.38 CFAi filter

A blur mask interpolates pixels by lowpass filter, and always blurring image. Normal and sharp combine unsharp masking for clearing image, however, induce overshooting at strong edge.

6.5.2. Tuning follow 2

Decide Luminance Edge threshold: Set smaller value to force stronger line connection

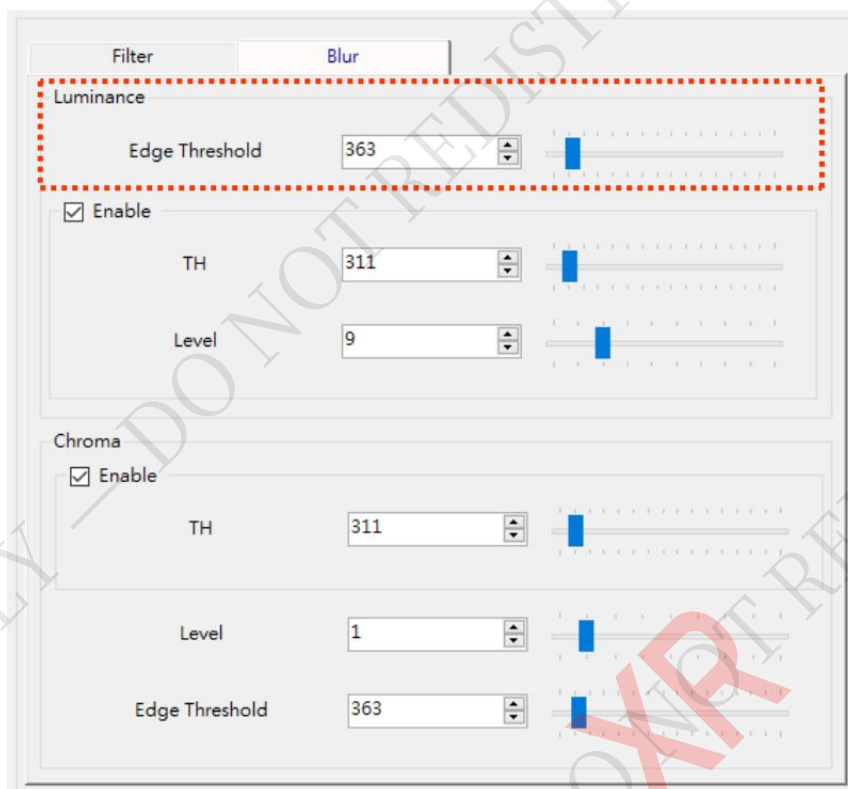


Fig 6.39 CFAi Blur

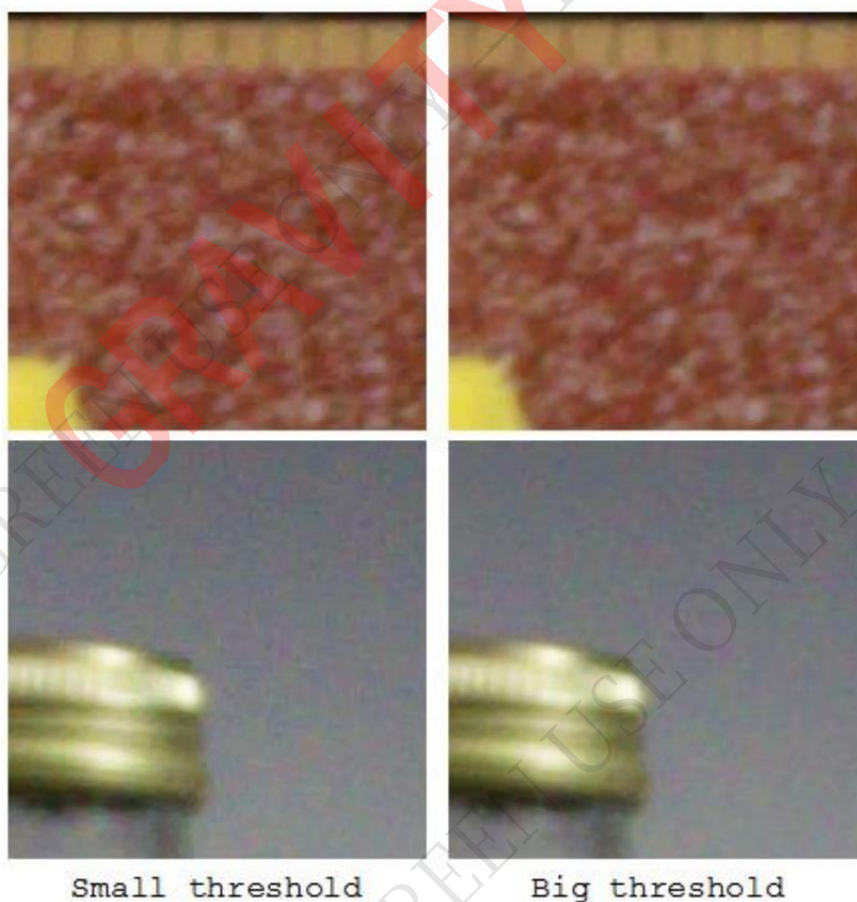


Fig 6.40 CFAi Blur Edge Threshold

A small threshold will give an aggressive connection between pixel and pixel. It makes lines seems continuous and smoothing but also on noisy region. In general, this setting should higher than noise level.

6.5.3. Tuning follow 3

Reduce small pattern: the blurring pattern under TH

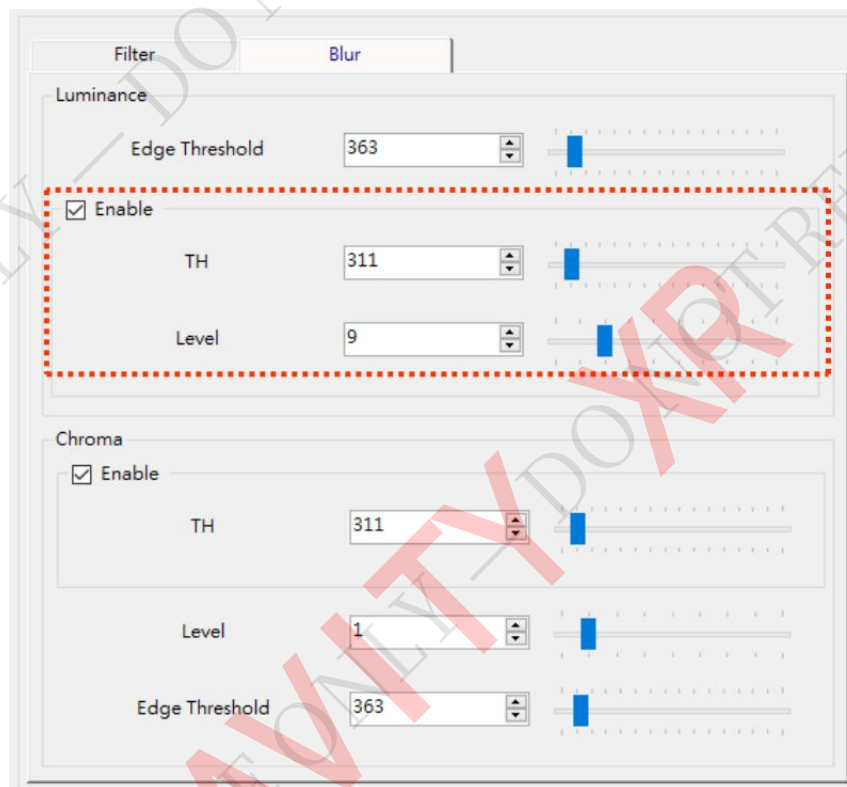


Fig 6.41 CFAi Blur level



Fig 6.42 CFAi Blur level example

If there are intensive lines in image and the line frequency is out of filter handing, the algorithm could not solve the lines correctly. In this situation, you can choose to leave pattern or to blur them. For blurring the pattern, set the TH to cover pattern region and level for blurring strength.

6.5.4. Tuning follow 4

Video IRP mode

- For solving more lines
- Choose a responsible setting for initial

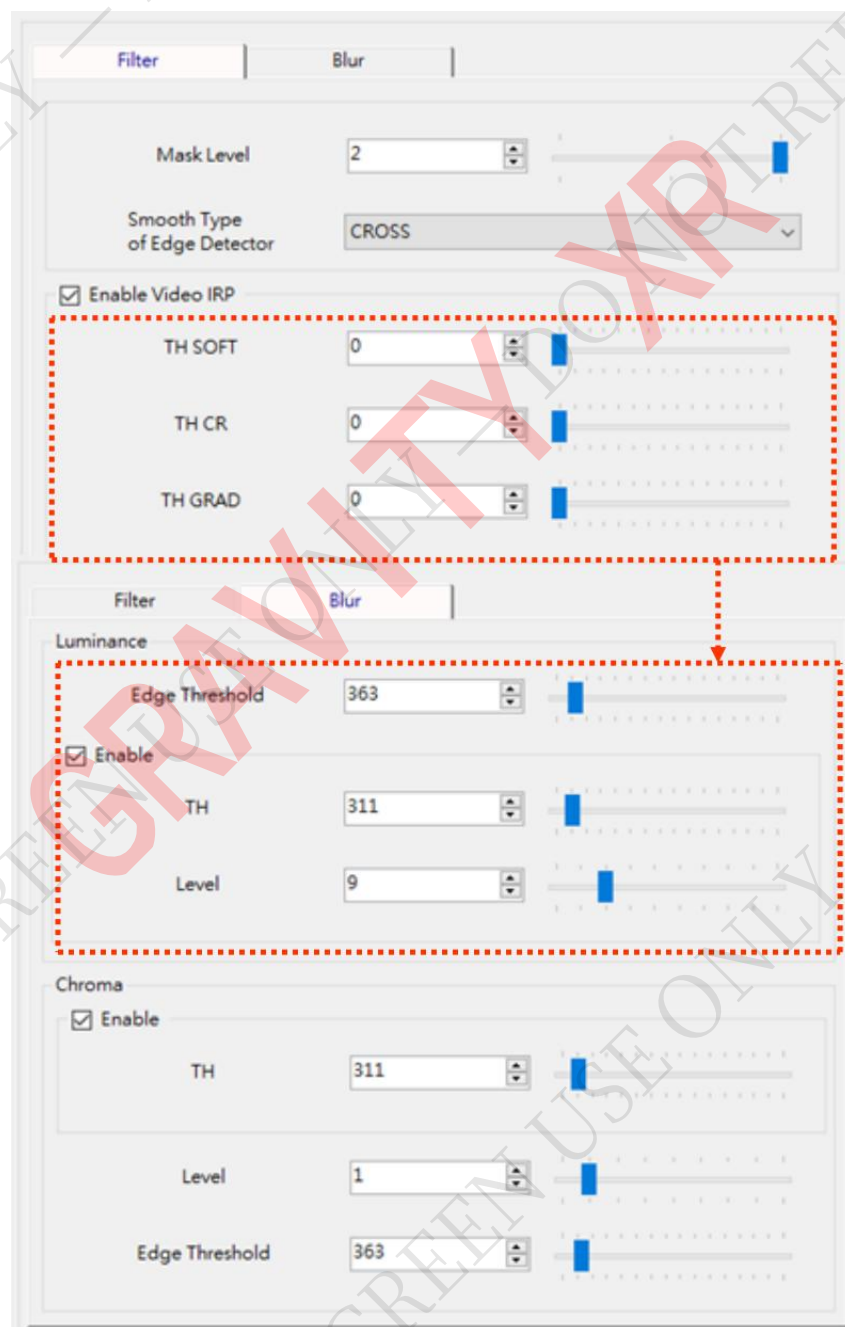


Fig 6.43 CFAi Video IRP mode

Video IRP mode is used to save more lines when frequency is out of handing. Even can not handing but still want to take risk to resolve. Here is the option for this purpose.

However, these parameters are not certainly rules to control quality changing. Suggest to select one of setting in profiles and then go adjusting. Below are profiles with different style of these 3 parameters' combinations. These 3 parameters also relate to Luminance parameters and should to be adjusted together.

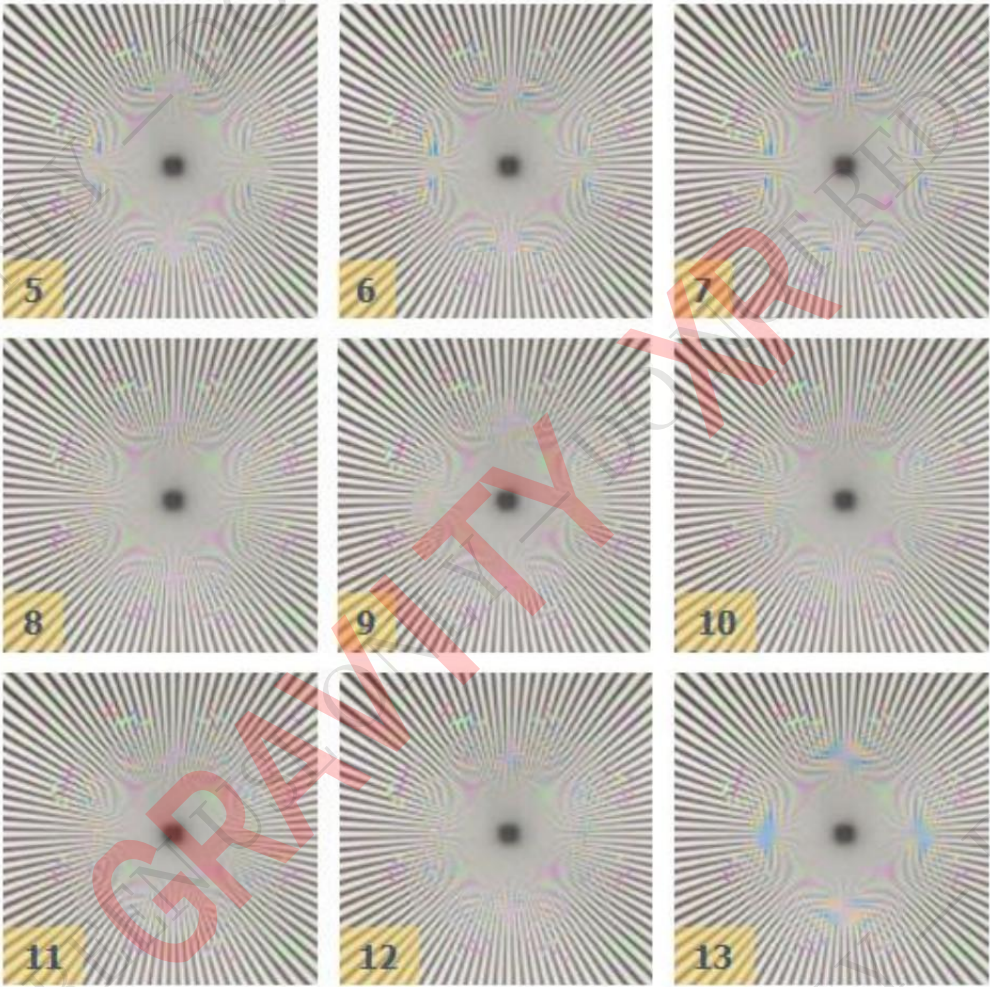


Fig 6.44 CFAi Video IRP mode example
Responding setting as below:

[5] mask level = 2 [VideoIRP] TH SOFT = 110 [VideoIRP] TH CR = 10 [VideoIRP] TH GRAD = 360 [Luminance] Edge Threshold = 140 [Luminance] TH = 180 [Luminance] level = 10	[6] mask level = 2 [VideoIRP] TH SOFT = 55 [VideoIRP] TH CR = 10 [VideoIRP] TH GRAD = 180 [Luminance] Edge Threshold = 70 [Luminance] TH = 90 [Luminance] level = 10	[7] mask level = 2 [VideoIRP] TH SOFT = 15 [VideoIRP] TH CR = 10 [VideoIRP] TH GRAD = 220 [Luminance] Edge Threshold = 60 [Luminance] TH = 10 [Luminance] level = 12
[8] mask level = 2 [VideoIRP] TH SOFT = 85	[9] mask level = 2 [VideoIRP] TH SOFT = 85	[10] mask level = 2 [VideoIRP] TH SOFT = 285

[VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 320 [Luminance] Edge Threshold = 320 [Luminance] TH = 20 [Luminance] level = 10	[VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 320 [Luminance] Edge Threshold = 320 [Luminance] TH = 220 [Luminance] level = 10	[VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 120 [Luminance] Edge Threshold = 80 [Luminance] TH = 30 [Luminance] level = 10
[11] mask level = 2 [VideoIRP] TH SOFT = 285 [VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 320 [Luminance] Edge Threshold = 80 [Luminance] TH = 30 [Luminance] level = 10	[12] mask level = 2 [VideoIRP] TH SOFT = 285 [VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 520 [Luminance] Edge Threshold = 80 [Luminance] TH = 30 [Luminance] level = 10	[13] mask level = 2 [VideoIRP] TH SOFT = 285 [VideoIRP] TH CR = 20 [VideoIRP] TH GRAD = 320 [Luminance] Edge Threshold = 180 [Luminance] TH = 130 [Luminance] level = 10

6.5.5. Tuning follow 5

Reduce color aliasing:

- Suggest set small values for most of case.
- Depended on issue scene for special tuning.

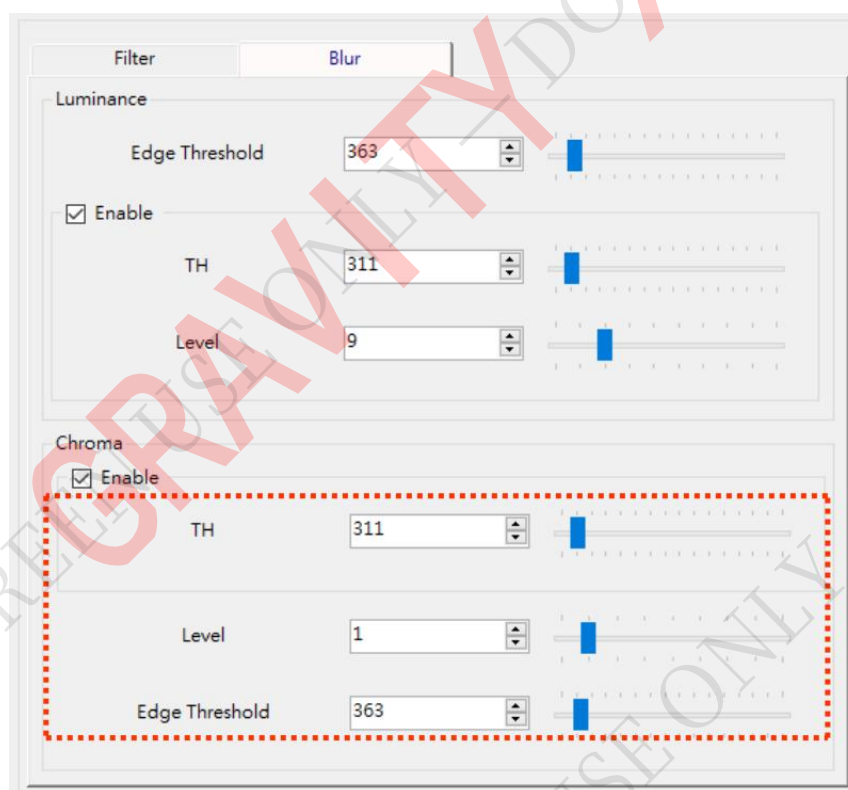


Fig 6.45 CFAi Chroma filter

In this option is to reduce color aliasing by blurring chrominance. However, too strong setting will cause more error color instead.

6.6. Dynamic Tone Control

- DTC is used to automatic generate LTM curve、Scale curve and GTM curve according to the midtones gain and shadows gain.
 - DTC Limitation(default)
 - midtones gain 1.0x~8.0x
 - shadows gain 1.0x~4.0x.
- If midtones gain and shadows gain are around 1.0x, DTC will be bypass, and directedly apply the LTM curve、Scale curve and GTM curve which are User defined in the LTM and GTM tuning block.

6.6.1. BV trigger

- There are three trigger zones by BV.
- You could select which zone to tune, and define the BV range by user.
 - BV -5000~-1500, define as night scene.
 - BV -1000~ 1400, define as indoor scene.
 - BV 2400~12000, define as outdoor scene.
- Weighting indicates which interval falls into.
- This trigger zones are sync with DTC、GTM and LTM blocks.

Select To Tune	BV : 5000	Weighting
☐	-5000 ~ -1500	0
☐	-1000 ~ 1400	0
☒	2400 ~ 12000	1

Fig 6.46 BV Trigger Zone

6.6.2. DTC Setting

Please refer to Fig. 6.39 DTC setting

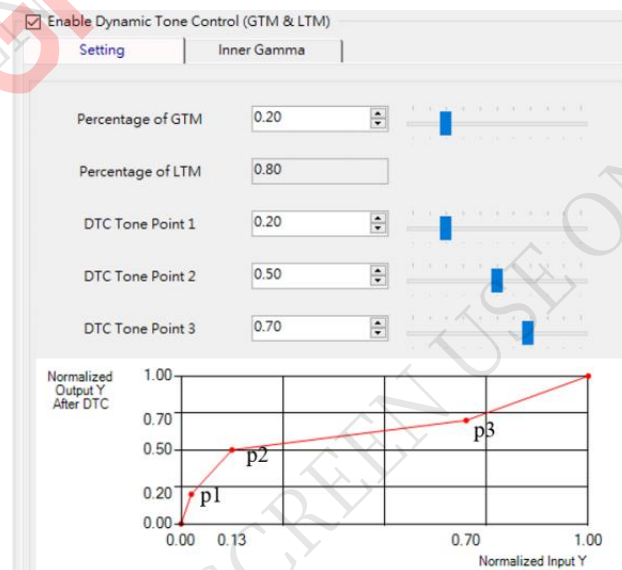


Fig 6.47 DTC Setting

- Enable DTC, otherwise disable GTM and LTM. (Simulation only)
- Tune GTM percentage
 - ☐ Range:0~1
 - ☐ The percentage of brightness boost by GTM.
 - ☐ Recommend setting
 - Night scene :0.5
 - Indoor scene: 0.35
 - Outdoor scene: 0.2
- Tune Tone Points1~3
 - ☐ Range:0~1
 - ☐ Control the tendency of DTC gain curve
 - 0~Point1: represent low key range, where the midtones gain and shadows gain taking effect.
 - Point1~Point2: represent middle key range, where the midtones gain taking effect.
 - Point2~Point3: represent middle to high key range, where the midtones gain is progressively reduced to 1.0x;
 - ☐ Recommend setting: (0.2,0.5,0.7)

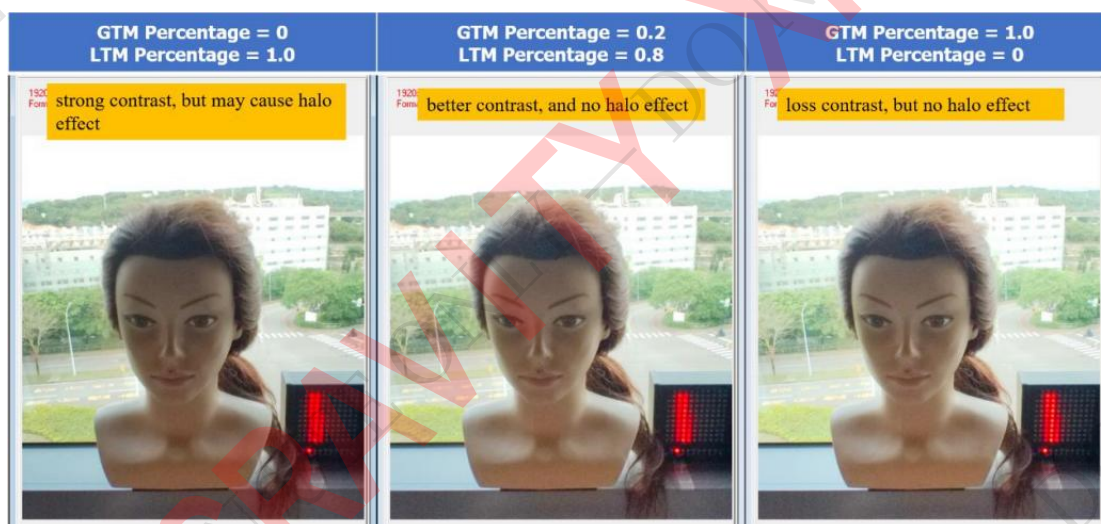


Fig. 5.61 The difference of GTM percentage when midtones gain = 4.0x 、 shadows gain = 2.0x

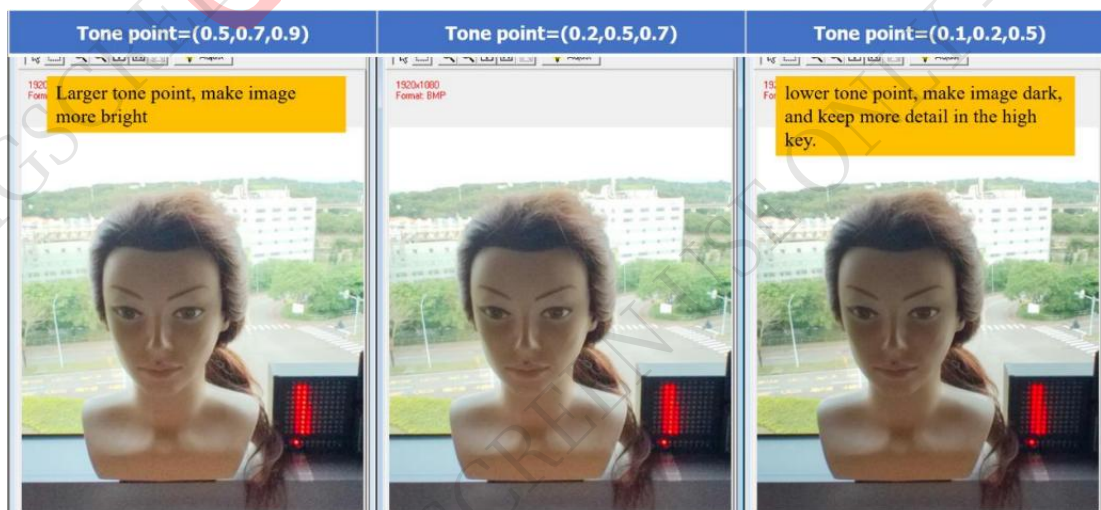


Fig 6.48 The difference of Tone Point

6.6.3. Inner Gamma Setting

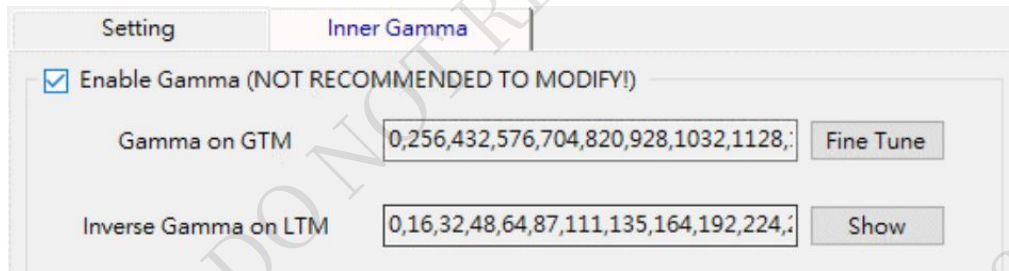


Fig 6.49 Inner gamma setting

- The inner gamma is applied after GTM to compress image into 12bits.
- Keep low key detail by inner gamma.
- The inverse gamma will be automatic generated, and applied after LTM to inverse inner gamma.
- Range:
 - Inner gamma: 16 bits in,12 bits output mapping.
 - Inverse inner gamma: 12 bits in,12 bits output mapping.
- Suggest to enable Gamma on GTM, otherwise apply linear gamma on GTM
- Not recommended to modify.

6.7. GTM

- GTM applies global gain curve to each pixel to enhance brightness according to the pseudo Y index.
- If the midtone gain and shadows gain are around 1.0x, User can define the global gain curve in this block.
- After apply gain curve, there is an inner gamma on GTM to compress 16bits RGBs into 12bits.
- Convert RGBs to YCC420s format for NRE.

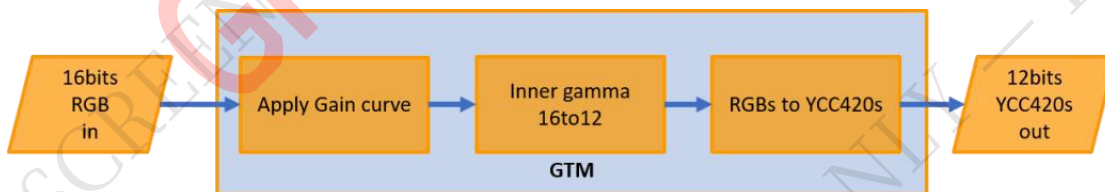


Fig 6.50 The flow of GTM

6.7.1. Tool Bar UI

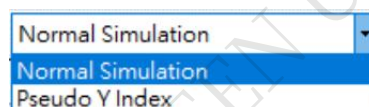


Fig 6.51 UI of Preview Option

- Preview Option
 - Normal Simulation:

- show preview as GTM result.
- Pseudo Y index:
 - show preview as pseudo Y index map for tuning.
 - Move cursor on the preview, get pseudo Y index for tuning user defined gtm curve (Fig 6.52).
 -



Fig 6.52 Pseudo Y index Map from preview

6.7.2. RGBs to Pseudo Y

Please refer to Fig 6.45 the RGBs to pseudo Y.

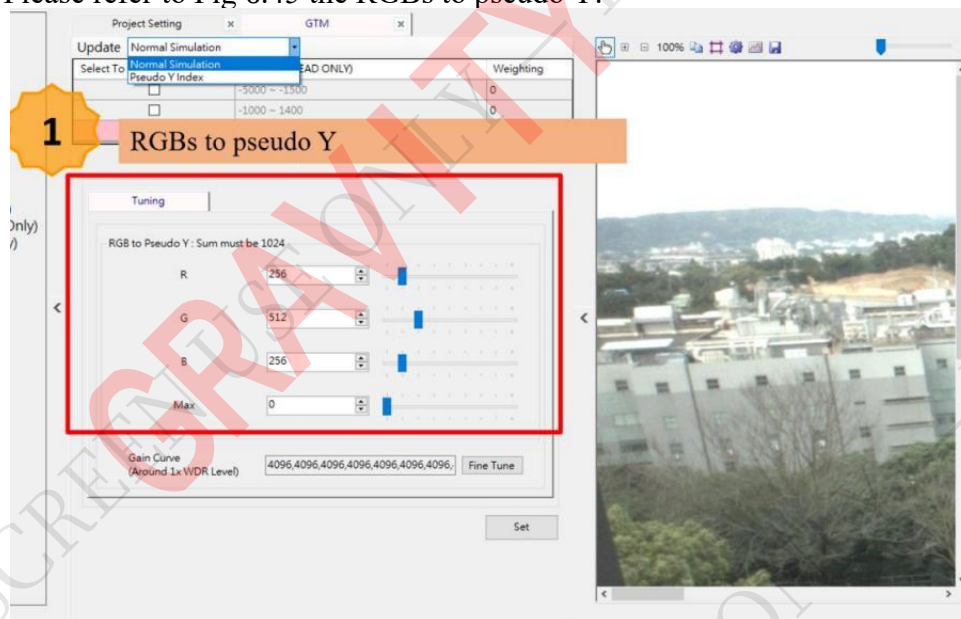


Fig 6.53 RGBs to Pseudo Y Interface

- RGBs to Pseudo Y
 - Before Global tone mapping, calculate pseudo Y to LUT.
 - Finetune c1~c4, the coefficients of R,G,B and max(R,G,B).
 - Range:0~2047
 - RGB to Pseudo Y coefficients, sum must be 1024.
 - $\text{Pseudo Y} = (R \cdot c1 + G \cdot c2 + B \cdot c3 + \max(R, G, B) \cdot c4) \gg 10;$
 - Default = (256,512,256,0)



Fig 6.54 The diff of coefficients

6.7.3. User Defined GTM Gain Curve

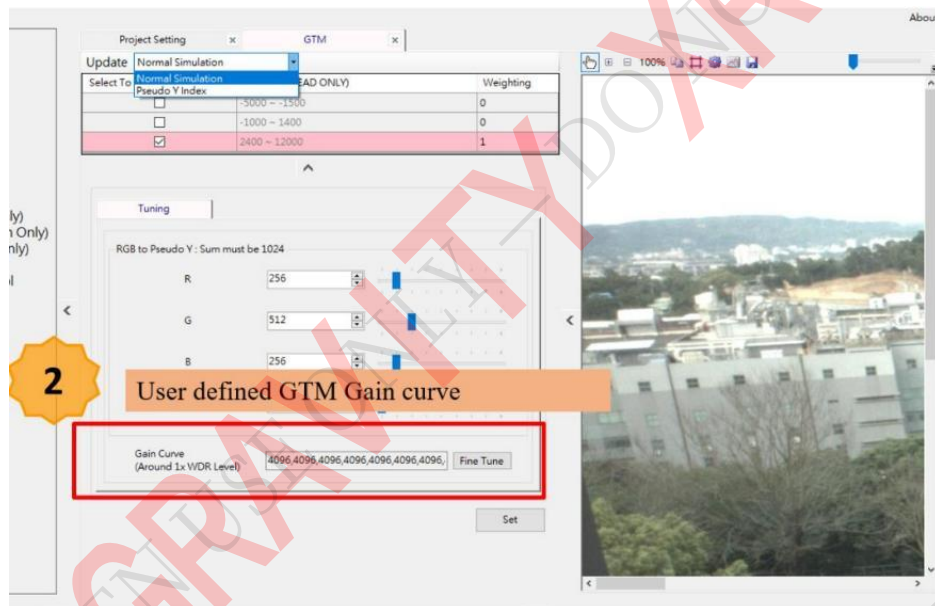


Fig 6.55 GTM Gain Curve Interface

- User Defined Gain Curve
 - ❑ Finetune user defined gain curve according to the pseudo Y index.
 - ❑ Effective conditions:
 - midtone gain and shadows gain are around 1.0x.
 - ❑ Range: 4096(1.0x) ~ 131071(31.99x)
 - ❑ pseudo Y index could be got from preview in the tuning tool.
 - ❑ Default = 4096(1.0x) for all pseudo Y index.
- Example for user defined gain curve
 - ❑ boost up dark area and face area.
 - Find the pseudo Y index around face and dark area. (around 0~20)
 - Increase gain around pseudo Y index 0~20.
 - Gain curve should be smooth to avoid brightness inversion.

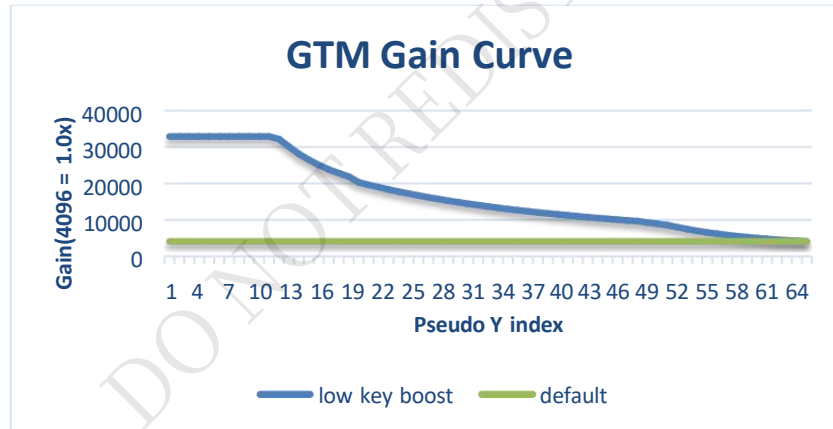


Fig 6.56 gain curve

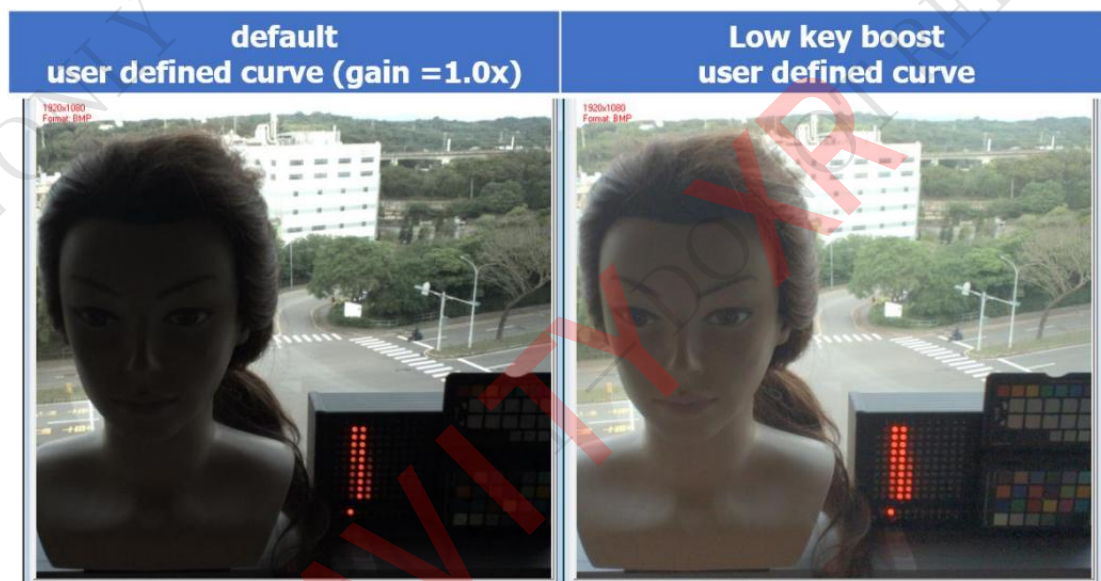


Fig 6.57 GTM result

6.8. NRE Multi-Layer NR

This function allows user to remove noise in each channel (Y, U, and V). NRE is 4 layers parameter. Every layer corresponded to previous bottom layer. So L4 parameter should be decided first. Only Y has Edge Refinement.

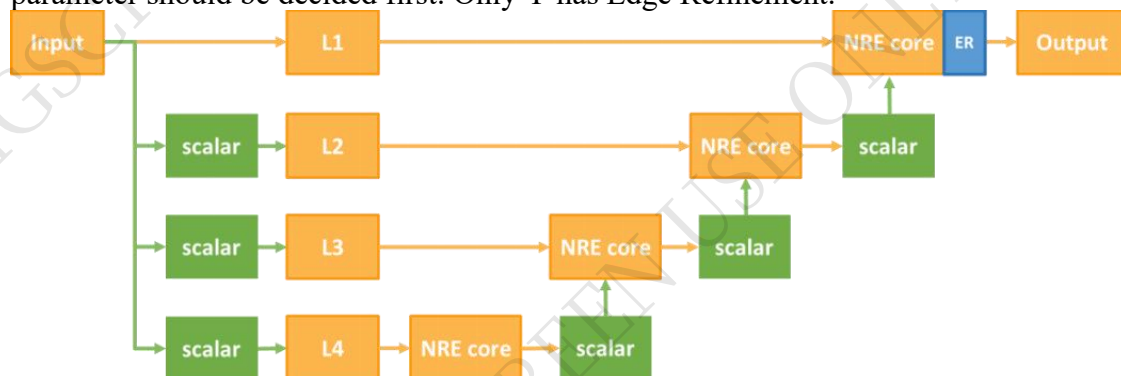


Fig 6.58 NRE Architecture

Multi-layer architecture reduces noise by band. Different layer is response to different noise band.

- User can adjust level on each layer to control quality.
- Reduce low frequency noise required larger mask, or need more layers.

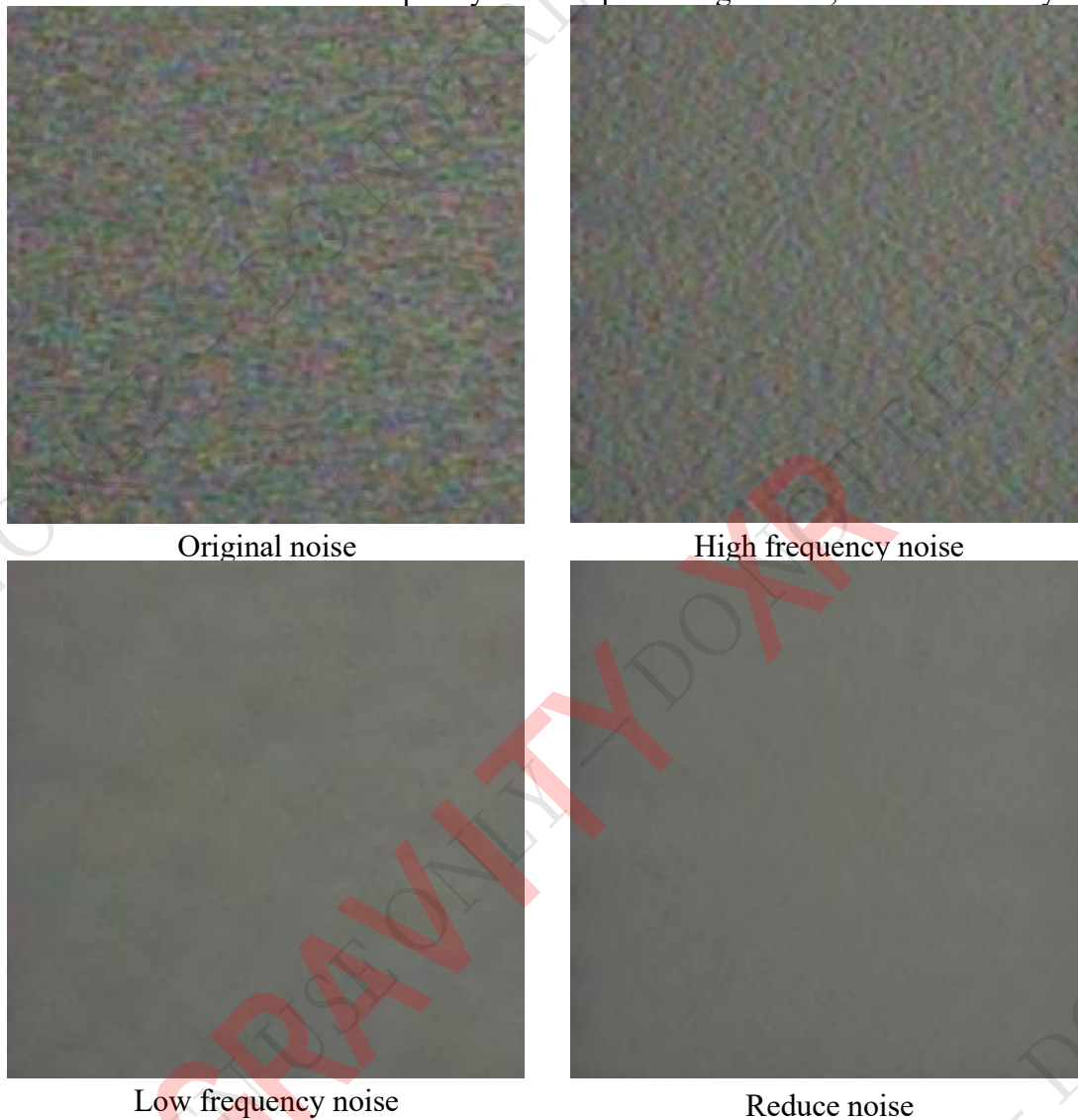
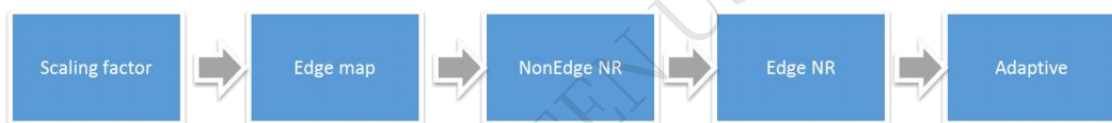


Fig 6.59 NRE Architecture CbCr

6.8.1. Major follow

- Major tuning follow as below
- From layer 4 to layer 1
- For Y, Cb, Cr



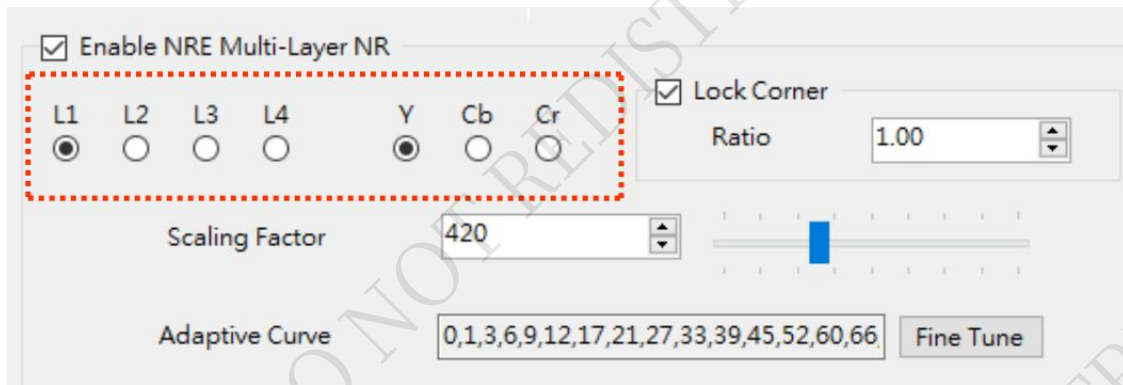


Fig 6.60 NRE tuning flow

6.8.2. Tuning follow 1

- Decide equivalent mask size by choose “Scaling Factor”
 - ☐ Larger scaling factor means larger mask size, and larger mask can remove lower frequency noise
 - ☐ To check suitable scaling factor
 - Select last layer output
 - Make Blend Mode to 2
 - Make Filter 2x to max level
 - Adjust a min scaling factor that flat area noise is not obviously
 - ☐ If you are set the strongest parameter but low frequency noise still left, then you should to increase scaling factor.



Fig 6.61 NRE scaling factor

6.8.3. Tuning follow 2

- Show edge map and adjust Edge Th
 - ☐ Adjust Edge Th on different pixel intensity
 - ☐ Set corner ratio for auto adjust corner para
 - Corner ratio would be related to shading gain on corner

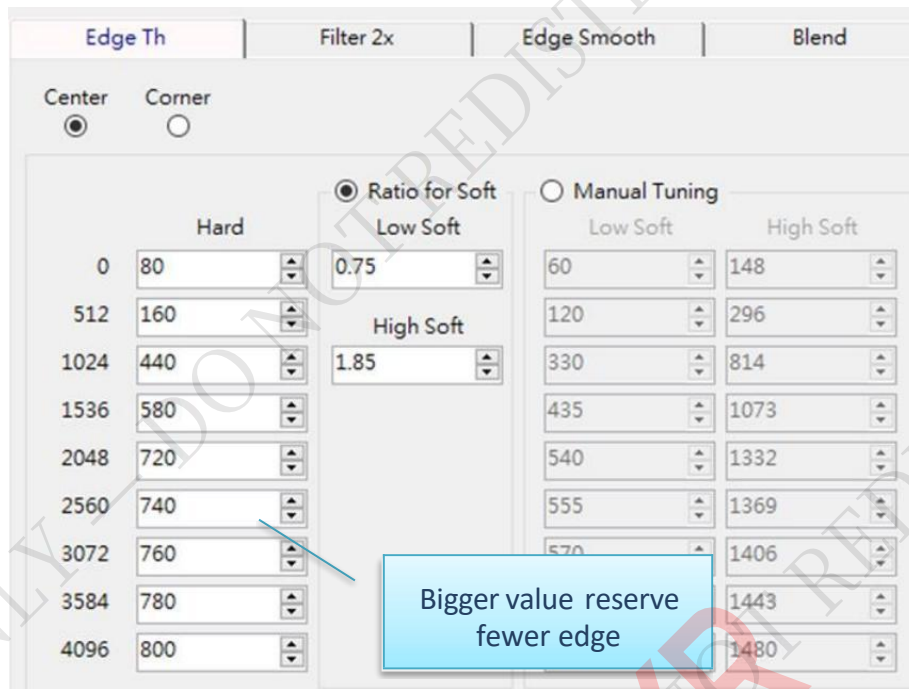
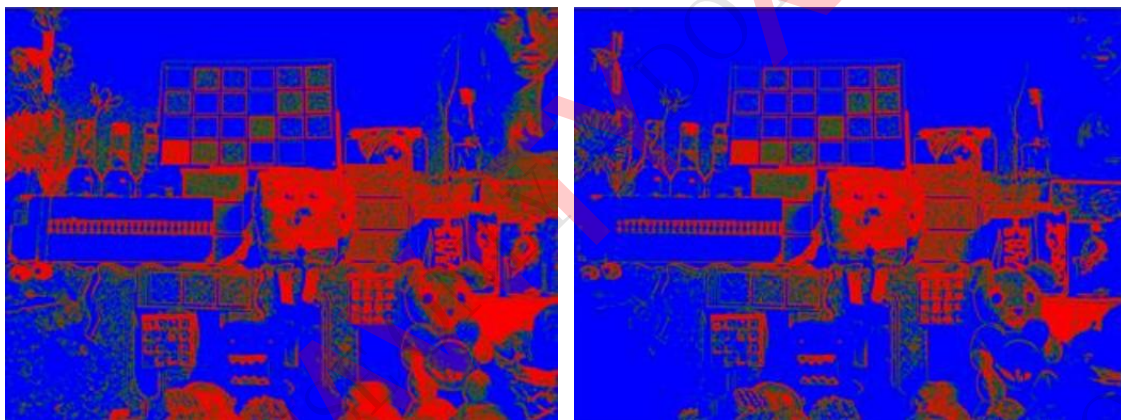


Fig 6.62 NRE edge threshold



Same level on center & corner

Suppress corner

Fig 6.63 NRE edge threshold example

6.8.4. Tuning follow 3

- Relation of Filter 2x, Edge smooth and Blend
 - Take edge map, pixel position, pixel intensity ...etc info to decide weighting for blending 2 filters' result.
 - For example: If you disable Filter 2x but blend all weight on Filter 2x, even set strongest setting on Edge smooth you will get a noisy image.

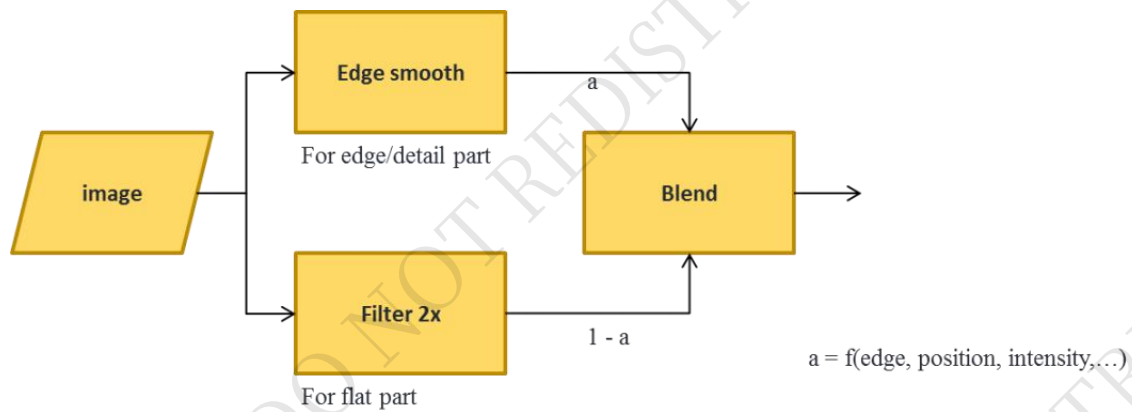


Fig 6.64 NRE filter & blend

6.8.5. Tuning follow 4

- Filter 2x
 - ☐ Mode: filter type, 0~3 → weak ~strong
 - ☐ Threshold: smaller value preserve more edge. 5 segments by intensity.
- For design purpose, is used on flat area. Suggest configure as stronger level.

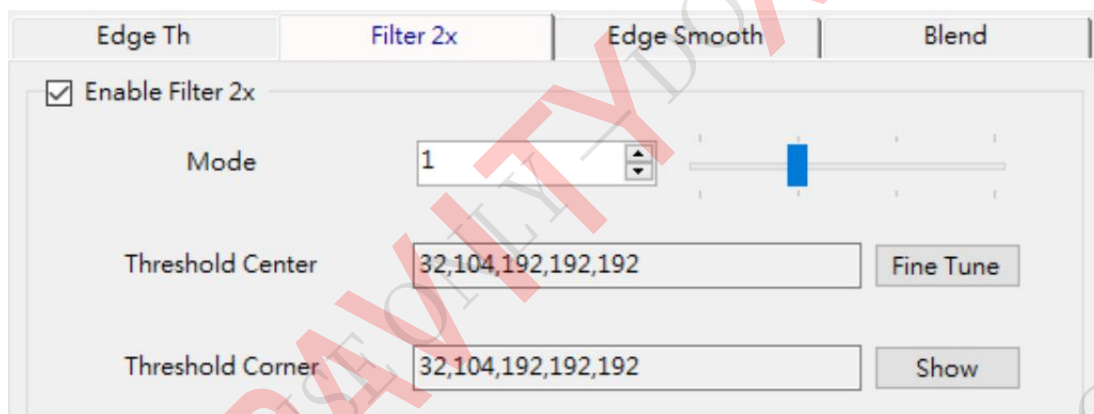


Fig 6.65 NRE Filter 2x

6.8.6. Tuning follow 5

- Edge Smooth
 - ☐ Threshold: smaller value preserve more edge. 5 segments by intensity.
 - ☐ N: transition level 0~15, bigger value makes less jaggy but blurring.
 - ☐ NonEdge Mode: applied on non-edge part, 0~4 → weak ~strong
 - ☐ NonEdge Weight: applied on non-edge part, 0~255 → blur ~clear
- For design purpose, is used on edge or detail area. Suggest configure to weak level.

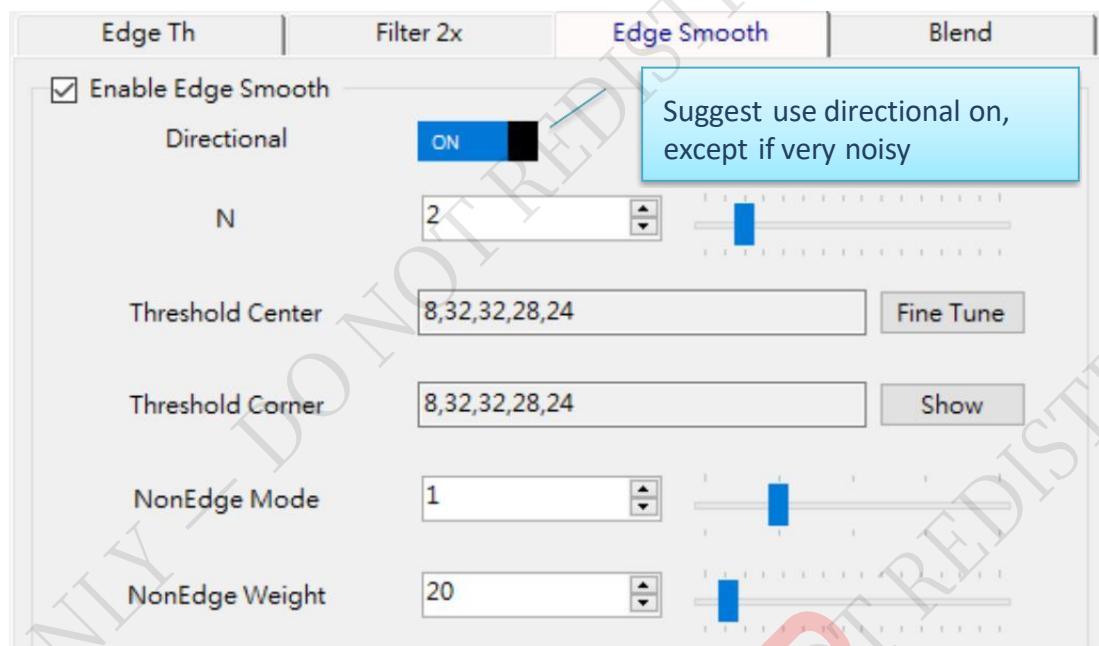
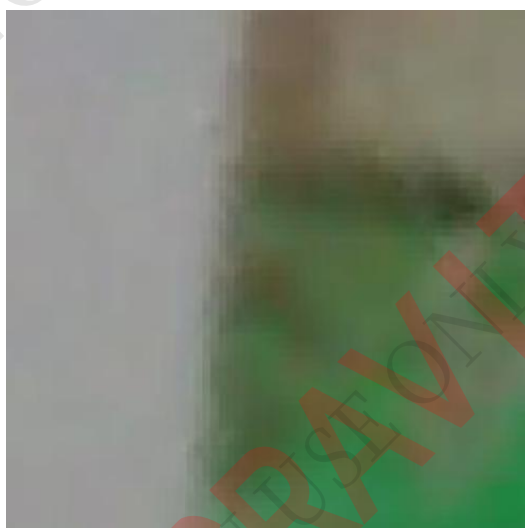


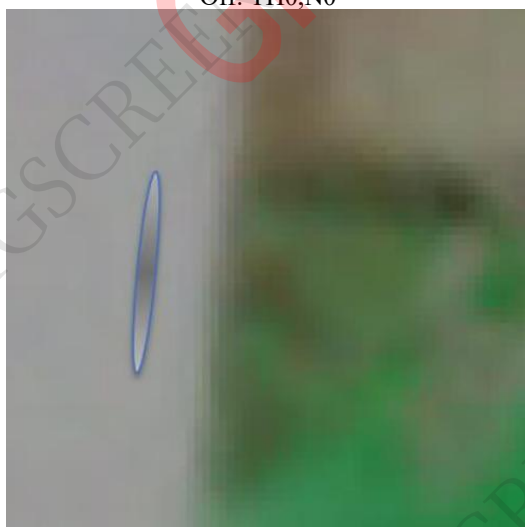
Fig 6.66 NRE edge smooth



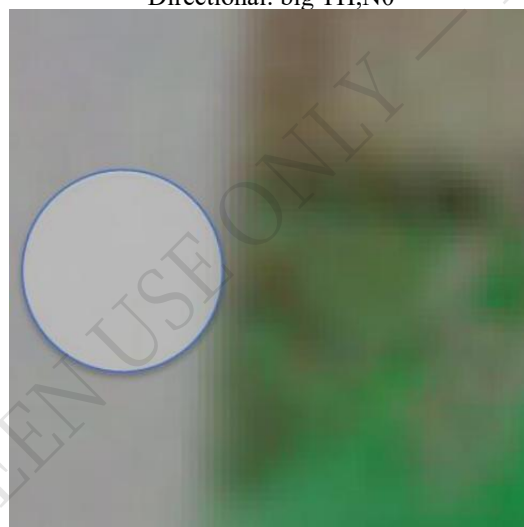
Off: TH0,N0



Directional: big TH,N0



Directional: big TH, big N



Local: big TH, N0



Local: big TH, big N

Fig 6.67 NRE edge smooth example

6.8.7. Tuning follow 6

- Vertical filter
 - There are 2 Vertical filter units behind Filter 2x and Edge Smooth on layer 2,3,4
 - Filter 2x and Edge Smooth are process along horizontal. Vertical filter are process along vertical.
 - Due to horizontal and vertical are processed by different filters, the balance of these 2 directions should be take care. If one of filter is much strong than the other, image will induce moving blur feeling on this direction.

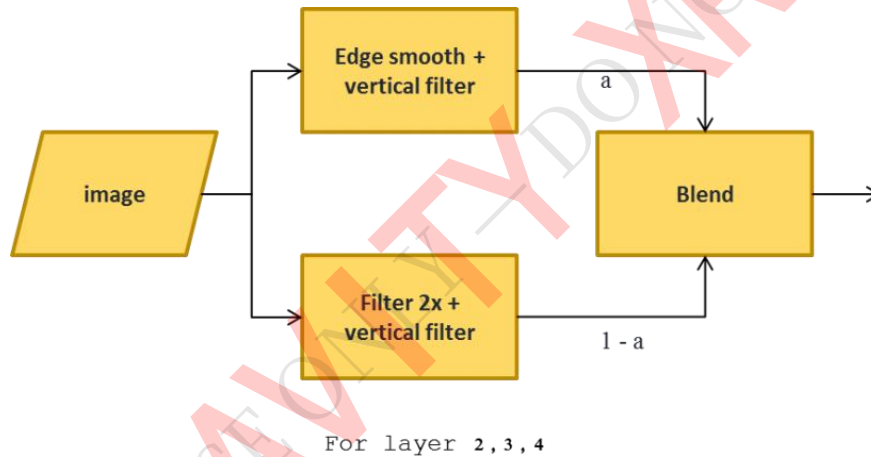


Fig 6.68 NRE Vertical filter & blend

- Weight: apply weighting on Edge by pixel intensity
 - 5 segments from dark to bright. Typical value is 1024. Small value mapping to strong NR.
- Edge: 1~65535, small value mapping to strong NR.
 - Different level on 8 directions.
- Alpha: basic NR level 0~65535. No matter condition, at least applying this NR level. Big value mapping to strong NR
 - Different level on 8 directions.
 - We often use this item to control H/V balance

☒ Enable VF Smooth

Weight	2624,1424,1024,1024,1024	Fine Tune
Edge	240,240,160,480,160,160,320,320	Fine Tune
Alpha	150,150,150,150,150,150,150,150	Fine Tune

Fig 6.69 NRE Vertical filter



Edge 1, Alpha 0

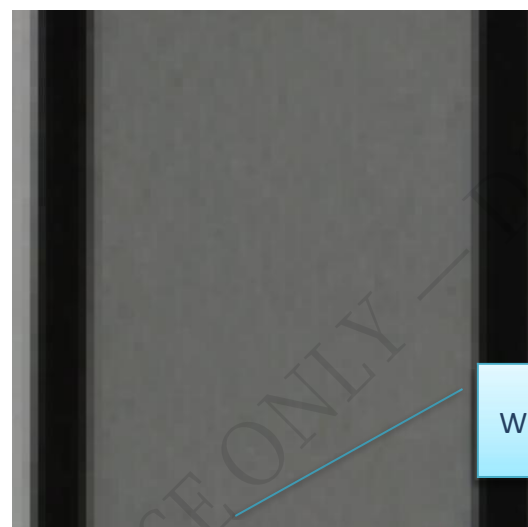


Edge 1, Alpha 8192

Strong
NR



Edge 100, Alpha 0



Edge 100, Alpha 8192

Weak NR

No Basic
NR

Fig 6.70 NRE Vertical filter example

Basic NR

6.8.8. Tuning follow 7

- Blend
 - ☐ Hard : in general is 255
 - ☐ Soft : maybe 75% of Hard
 - ☐ NonEdge : weighting is changed depended on pixel intensity
 - 0 is weighted to Filter 2x
 - 255 is weighted to Edge smooth
 - Directional : different level on 8 ways
 - ☐ Adaptive Weight : Stronger on corner

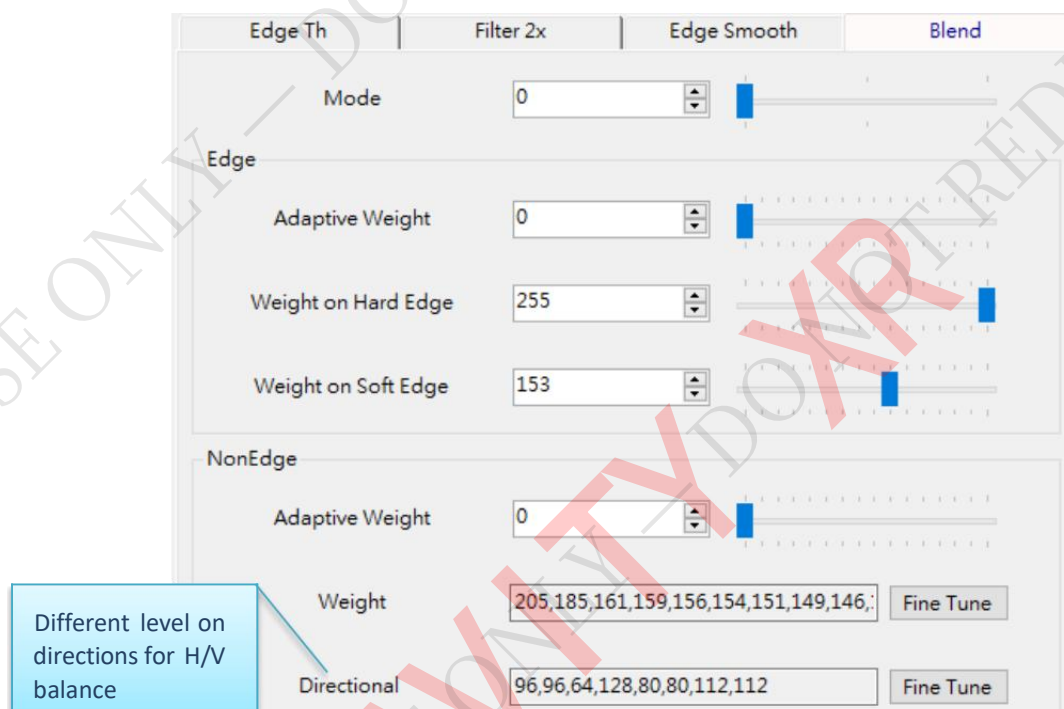
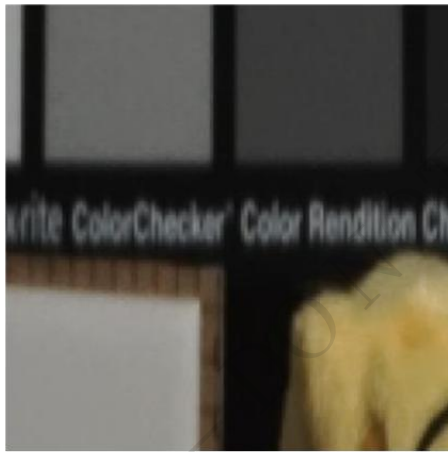


Fig 6.71 NRE Blend

You can switch blend mode to see the effect come from Edge Smooth & Filter 2x. That will help judge what filter needs to be fine tune.

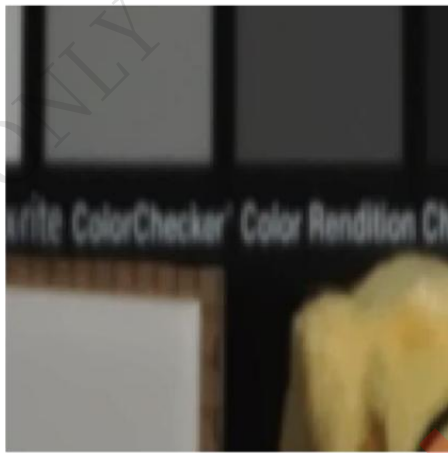
Mode = 1, Edge smooth + Vertical filter



a

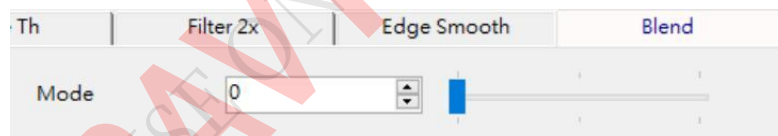


Mode = 0, Blend



1 - a

Mode = 2, Filter 2x + Vertical filter



Switch mode to check respective result.

Fig 6.72 NRE Blend example

6.8.9. Tuning follow 8

Decide Adaptive Curve

- ☐ This curve is uniform. Image center is at segment 0. However, half radius is at segment 8. This is to reserve more segments for corner tuning.
- ☐ In general setting, suggest last segment value is 255
- ☐ The trend of curve could refer to shading gain.

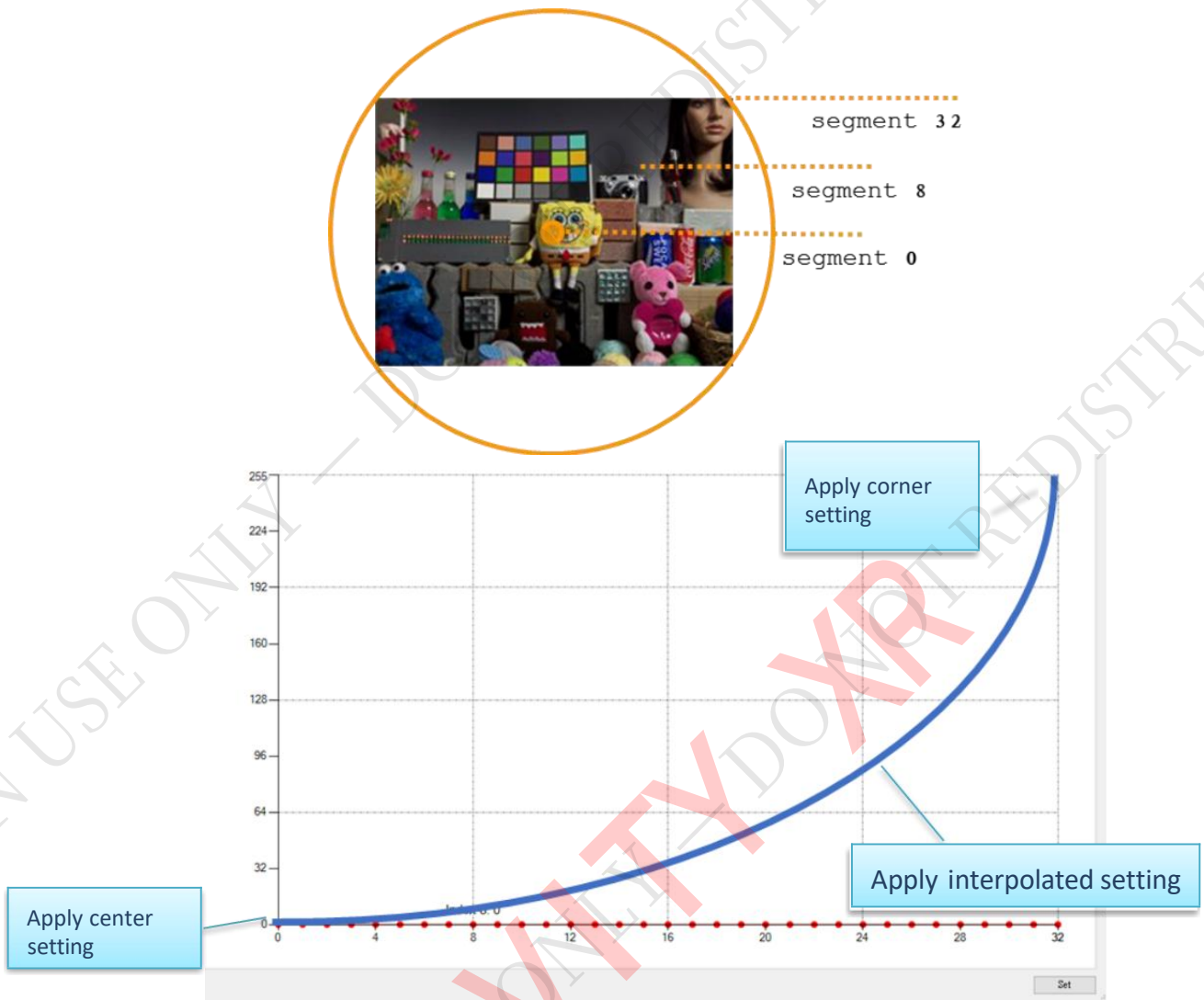


Fig 6.73 NRE adaptive

Each segment is mapping to image's radius as below:

segment	radius
0	0
1	0.176777
2	0.25
3	0.306186
4	0.353553
5	0.395285
6	0.433013
7	0.467707
8	0.5
9	0.53033
10	0.559017
11	0.586302
12	0.612372

13	0.637377
14	0.661438
15	0.684653
16	0.707107
17	0.728869
18	0.75
19	0.770552
20	0.790569
21	0.810093
22	0.829156
23	0.847791
24	0.866025
25	0.883883
26	0.901388
27	0.918559
28	0.935414
29	0.951972
30	0.968246
31	0.984251
32	1

6.9. NRE Edge Refine

We improve the dirty edge by Edge Refinement. And it only works on one layer of the Y channel. The effect is shown in the previous section Fig. 5.86. We try to improve the dirty edge problem. At the same time, we try not to lose the detail of the part. As shown in the Fig 6.76 .

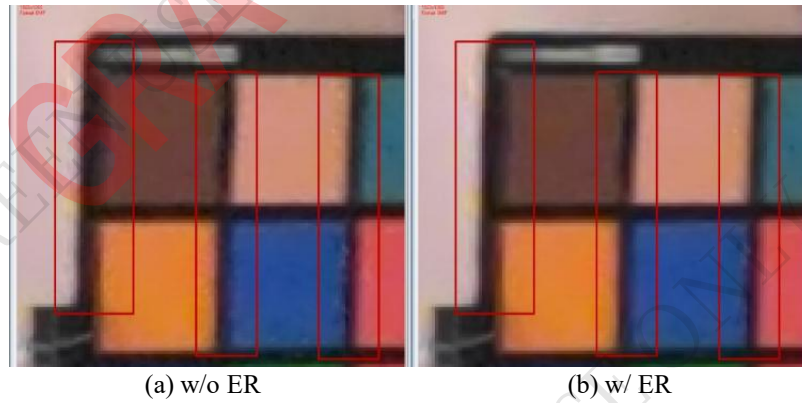


Fig 6.74 The effect of ER.

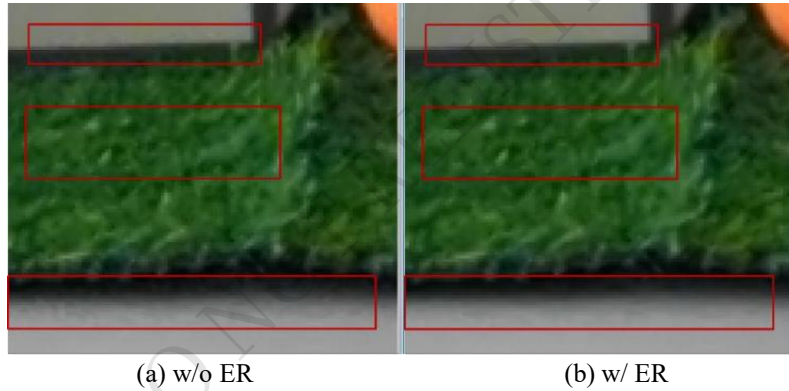


Fig 6.75 The effect of ER.

6.9.1. Interface Introduction

- Normal simulation: this output includes three functions. (YUV codec, NRE and ER).
- Debug information: we provide five debug information for user, as Fig 6.74
- Tuning information: please follow NRE flag setting
- ER Flag (Enable NRE Edge Refine): we can observe the difference between w/o ER and w/ ER.
- Each debug information can be cross-referenced to the title of each sub-page, as Fig 6.75

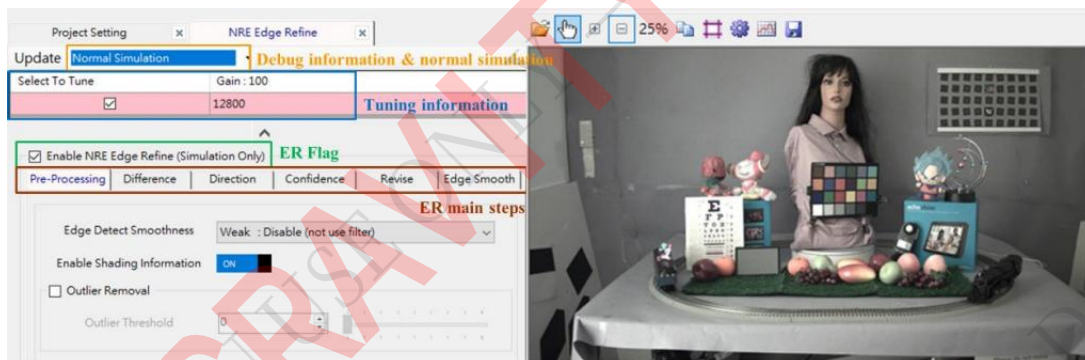


Fig 6.76 About ER interface

Note:
Debug information is displayed if
an image has been loaded.

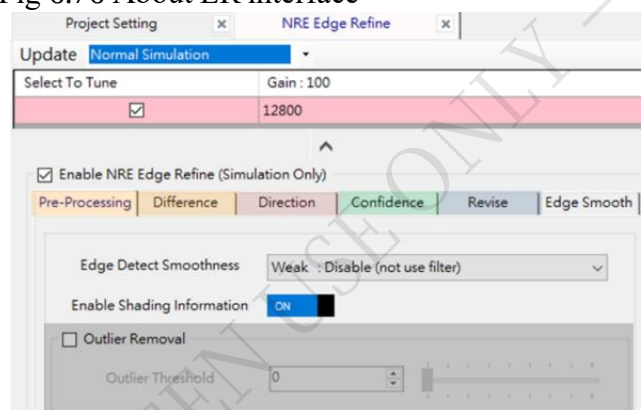


Fig 6.77 ER debug information

6.9.2. Parameter: Pre-Processing section

Please refer to Fig 6.78 the pre-processing page.

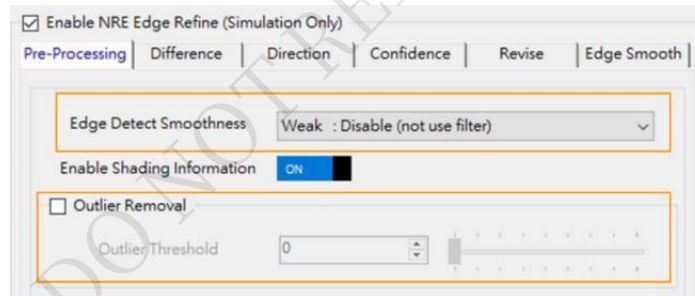
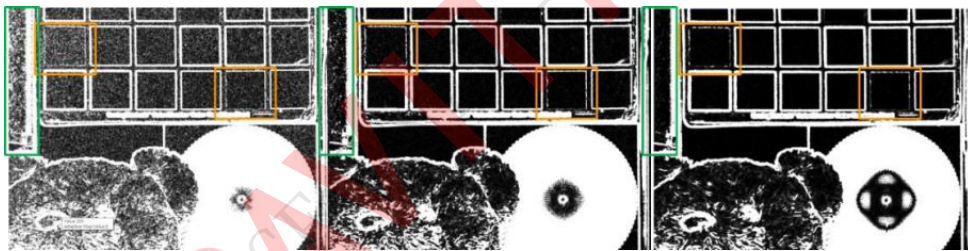


Fig 6.78 The **Pre-Processing** page.

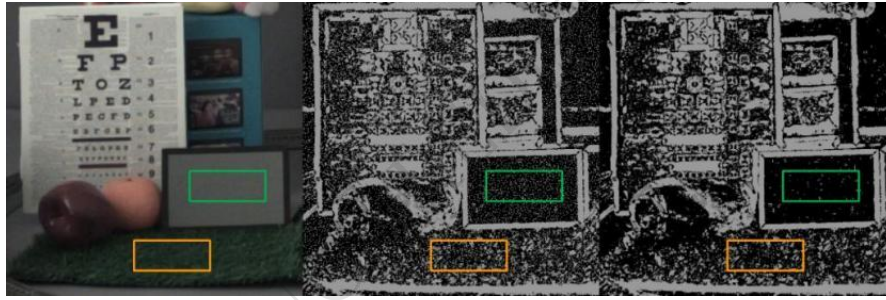
- Edge Detection Smooth:
 - ☐ Avoid noise affecting feature analysis, we use low pass filter to get cleaner image.
 - ☐ Its result could be observed by the debug information ***Difference Map***.
 - ☐ Three modes:
 - not use filter (the most commonly used):
It's easily affected by noise, but the sensitivity of detection is better.
 - Medium / average filter:
It's unaffected by noise, but the sensitivity of detection is also a little weaker.



(a) weak: not use filter (b) medium: gaussian filter (c) strong: average filter

Fig 6.79 The results of the difference map with different filter mode.

- Outlier Removal:
 - ☐ It is used to eliminate abrupt points.
 - ☐ Please note that we seldom use this feature because it is easier to loss the details.
 - ☐ Its result could be observed by the debug information ***Difference Map***.
 - ☐ Tuning tips: Observe the pixel value which you want to eliminate. Set threshold with lower value. As shown in Fig 6.80 .
 - High ISO, or blurred, less-detail images: on
 - Low ISO, or more-detail image (ex: grass, leaves, hair, etc.): off or set "Outlier Threshold" with higher value.



(a) original input (b) outlier threshold = 0 (c) outlier threshold = 100

Fig 6.80 The results of the total weight map with different thresholds.

6.9.3. Parameter: Difference section

Please refer to Fig 6.81 , the difference page

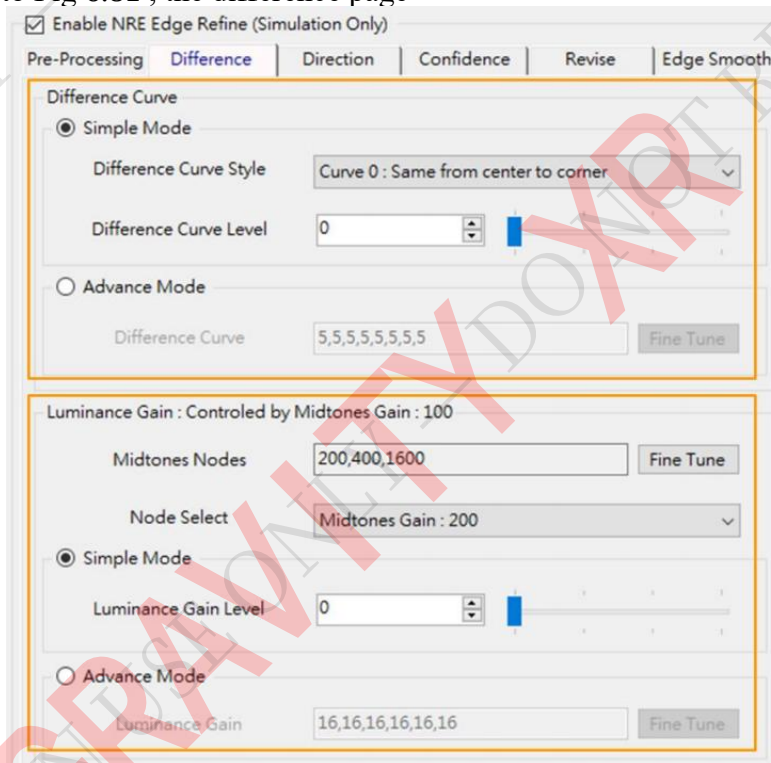
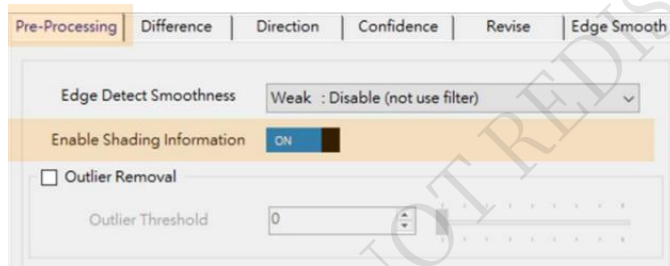
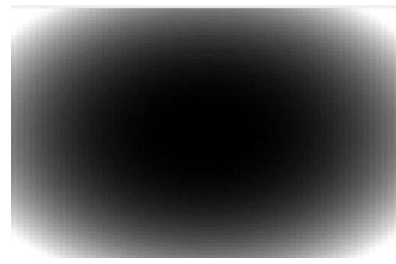


Fig 6.81 The **Difference** page.

- Difference Curve:
 - It affects the detection strength of **Difference Map**.
 - The higher the value seem the stronger gradient of edge.
 - Besides, we provide simple and advance mode for user. In general, you can use the simple mode to observe the results of the difference map.
 - **Simple mode:**
 - Difference Curve Style: three style curves
 - Difference Curve Level: provide four detection intensity for each style curve
 - Note: If you want to have different levels of detection for the center and corners, please remember to open the shading flag (as Fig 6.80).
 - The effect of the difference map, as shown in Fig 6.81 . **In general case, the corner of the image is dirtier than the center. Therefore, we could choose curve 1 as our setting.**

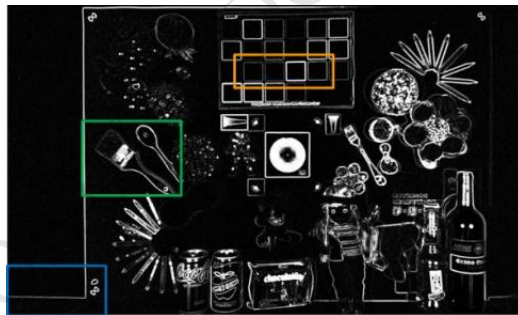


(a) shading flag

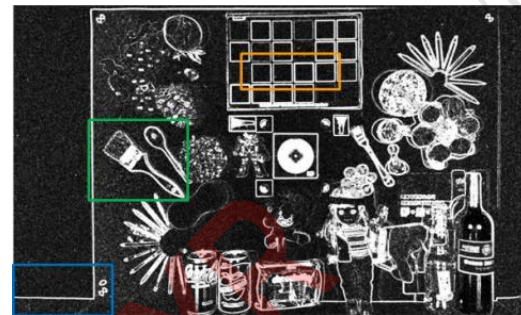


(b) the adaptive map value of NRE

Fig 6.82 shading flag information and adaptive map information



(a) Curve 0 & Level 0



(b) Curve 1 & Level 2

Fig 6.83 Difference map results with different parameters (Difference curve & level).

■ **advance mode: (we can adjust the parameter directly)**

- ☐ If the shading flag is on, it means that you can fill in the corresponding difference value based on the adaptive map value of NRE.
- ☐ If the shading flag is off, it means that the difference map would only be calculated by the first difference value.
- ☐ Value range: 0~255

■ **Luminance Gain:**

The value of this parameter is determined by midtones, and mainly affects the detection intensity of the **Difference Map**.

☐ **simple mode:**

- Luminance Gain Level: four levels
- If the overall brightness difference is not significant, you can use curve 0.
- However, if the overall brightness difference is very large, curve 3 can be used. As shown in the Fig 6.82.

■ **advance mode: (we can adjust the parameter directly)**

- We divide 0~255 into six intervals (8, 16, 32, 64, 128) based on the pixel intensity of the original input.
- The fillable value range is 0~128, and the base is 16, which means the maximum multiplication is 8 times.

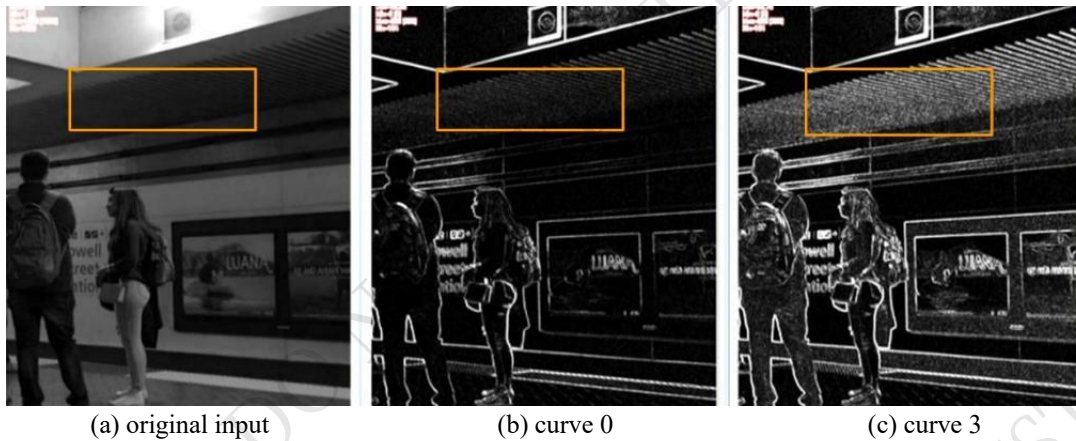


Fig 6.84 Difference map results with different parameters (Luminance Gain).

- Difference Threshold (not recommend to modify)
- If Filter Mode $\neq$ 0, we will merge two difference map information as our final difference map. As shown in the Fig 6.83. Besides, we **only** show this UI when Filter Mode $\neq$ 0, as shown in the Fig 6.84 (a).
- We dive 0~255 into five internals (10, 50, 100, 150) based on **original** input.
- Difference Threshold 1:
The difference map I result is based on filter results image, the range is 0~255.
- Difference Threshold 2:
The difference map II result is based on original image, the range is 0~255.
- Tuning tips:
 - We show the two related pieces of information about Difference Map I & II. As shown in the Fig 6.84 (b).
 - You could fill the appropriate parameter based on original input value.

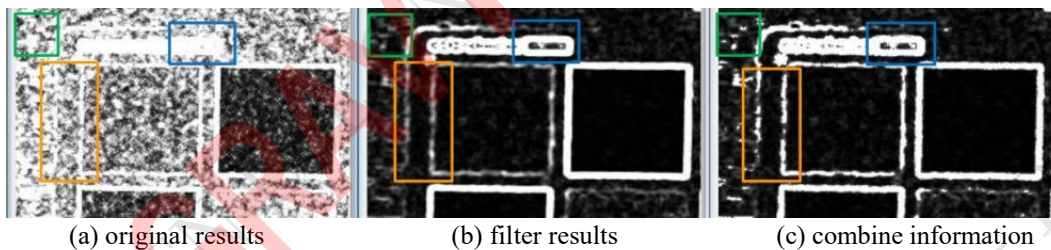
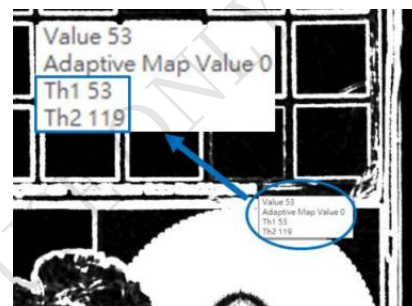


Fig 6.85 The final combine Difference map results if Filter Mode $\neq$ 0.

Difference Threshold : NOT RECOMMENDED TO MODIFY

Threshold 1	40,60,80,100,100	Fine Tune
Threshold 2	80,80,80,150,200	Fine Tune



(a) Mapping to the position of a parameter on a page

(b) Two related pieces of information about Difference Map I & II

Fig 6.86 The Difference Threshold information

6.9.4. Parameter: Direction section

Please refer to Fig 6.87 the direction page.

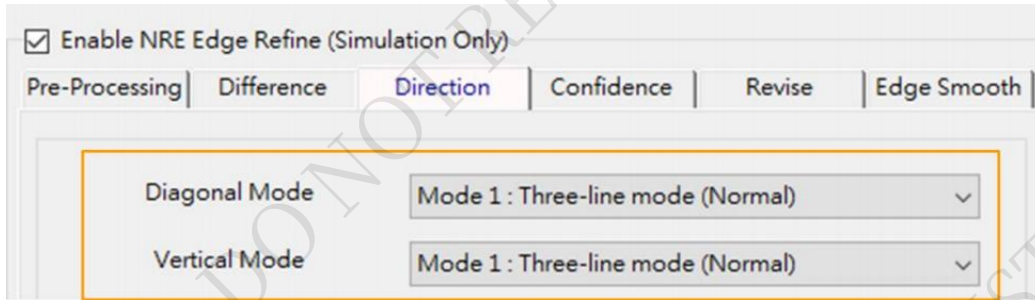


Fig 6.87 The **Direction** page.

In order to avoid non-directional refining, we determine the direction of edge-pixel by edge detection.

- We can observe the **Direction Map** to check results. Besides, we will label different color for each direction. As shown in the Fig 6.86 .
- We provide three modes for the user, the difference between them is shown in the Fig 6.87 -Fig 6.91 .

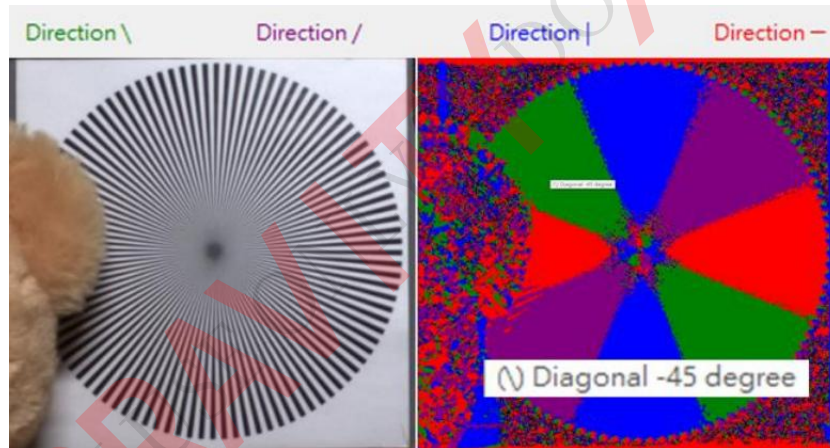


Fig 6.88 Each direction with different label.

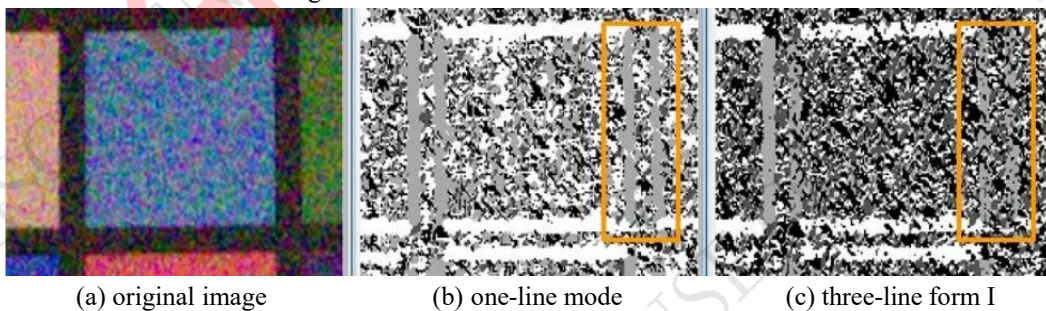


Fig 6.89 The Direction map results for case 1

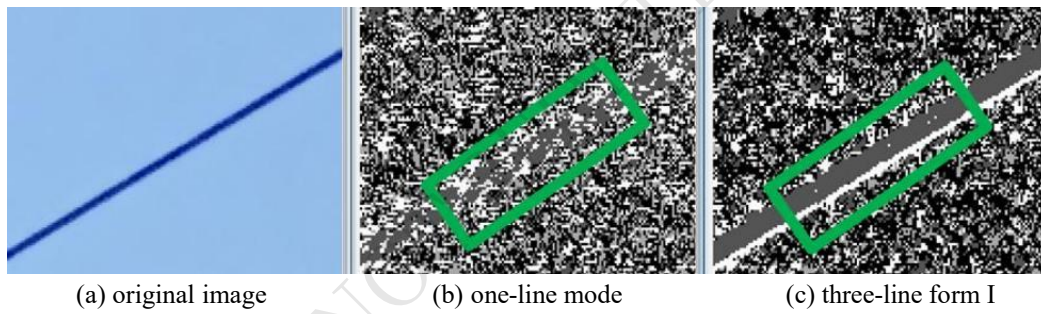


Fig 6.90 The Direction map results for case 2

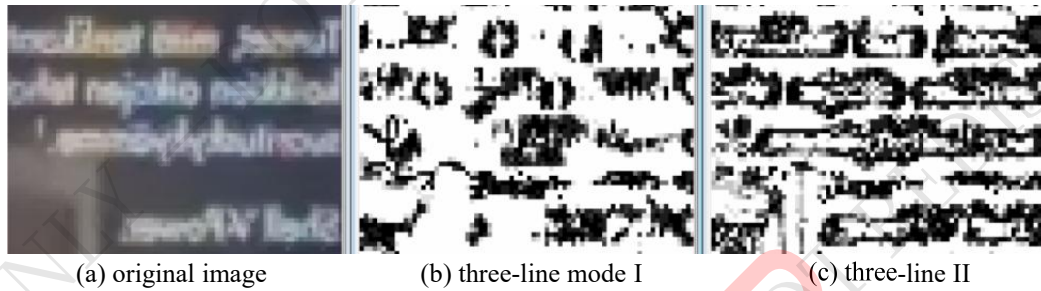


Fig 6.91 The confidence map for case

6.9.5. Parameter: Confidence section

Please refer to Fig 6.92 , the confidence page.

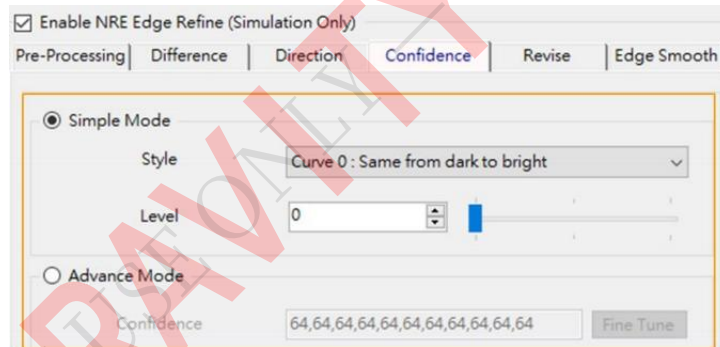


Fig 6.92 The Confidence page.

We calculated the confidence based on the gradient values in each direction and adjust the confidence threshold by the brightness value. Its debug information is *Confidence Map*.

■ simple mode:

- ☐ Confidence style: three style curves, as shown in Fig 6.91 .
 - Curve 1: The intensity of all pixel is with the same threshold.
 - Curve 2: The condition is stricter in dark area.
 - Curve 3: The condition is stricter in bright area.
- ☐ Confidence level: provide three detection intensities for each style curve

■ advance mode:

- ☐ In generally case, the brighter part of the middle area is not particularly adjusted, we mainly adjust the value to higher or lower for the darker areas.
- ☐ Range is 0 ~ 255, and its base is 64.
- ☐ The data range of the Confidence Map results is 0~11. The higher value, the

- The steps of the advance mode tuning as Fig 6.94 .

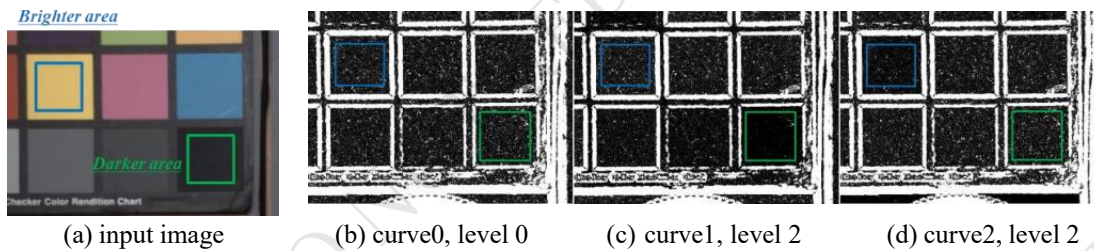


Fig 6.93 Confidence map results with different parameters (Confidence curve & level).

① First, disable ER to check pixel intensity.

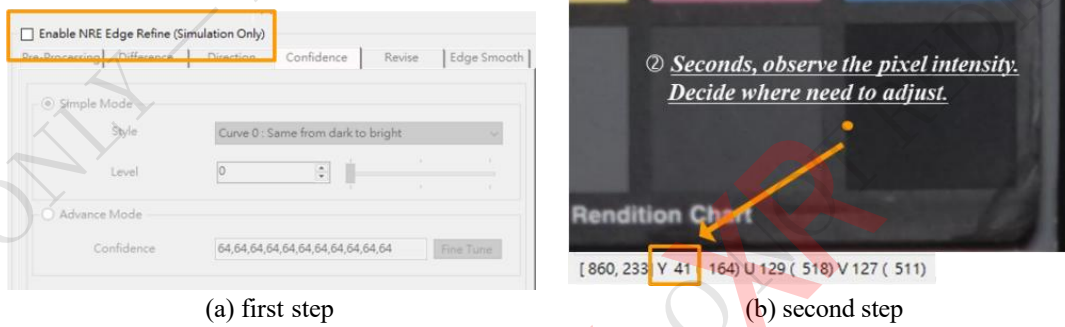


Fig 6.94 The steps of the advance mode tuning.

6.9.6. Parameter: Revise section

Please refer to Fig 6.95 , the revise page.

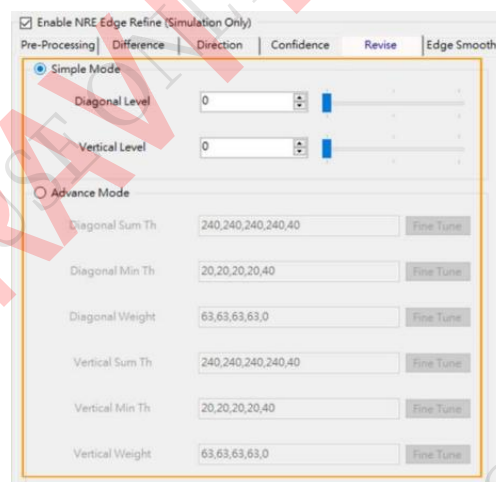


Fig 6.95 The **Revise** page.

We can revise directional confidence standard by Diagonal/Vertical Sum Th, Diagonal/Vertical Min Th. Its debug information is ***Revise Map***.

- **simple mode**

- ☐ Level 0~2: strict to lenient, their results as shown in Fig 6.94.
- ☐ Diagonal and vertical directions can be adjusted separately.
 - Diagonal direction includes $\pm 45^\circ$
 - Vertical direction includes vertical and horizontal.

■ **advance mode:**

- Characteristics of higher confidence are with higher Sum Th, and the minor Min Th. Its weight could be lager.
- In general case, Sum Th, Weight is a decreasing value and Min Th is an increasing value.
- Data range is 0~1000.
- The last parameter of Diagonal/Vertical Sum Th (the effect is as follows):
 - The recommendation can be determined by the overall noise level. AsFig. 6.97.
 - If the image is dirty, you could set a higher value, but also need to check whether it causes the smoothness to be insufficient.
 - If the image is clean, the value can be set lower.
- You can see the index information of the horizontal and vertical parameters where the pixel is. Besides, it also shows the conditional equation of the index. AsFig 6.98

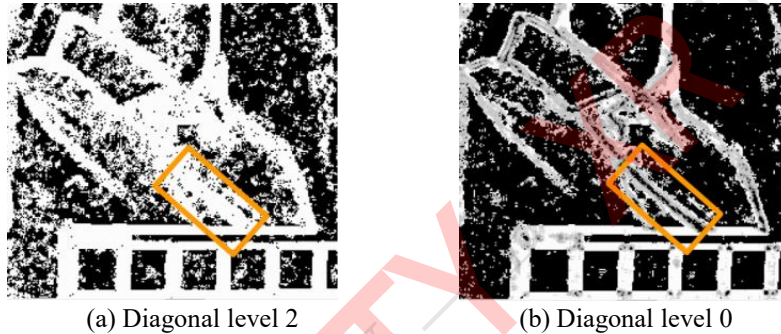


Fig 6.96 The output of Revise map with different simple mode parameters.

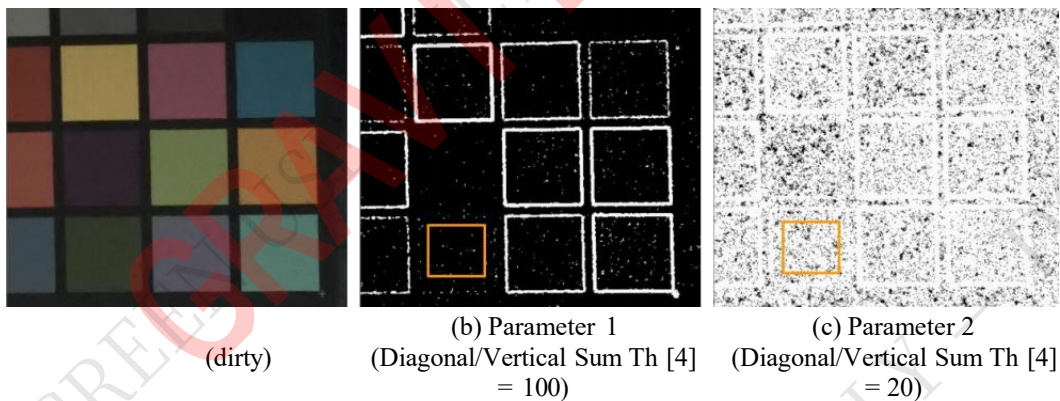


Fig 6.97 The output of Revise map with the last parameter of Diagonal/Vertical Sum Th

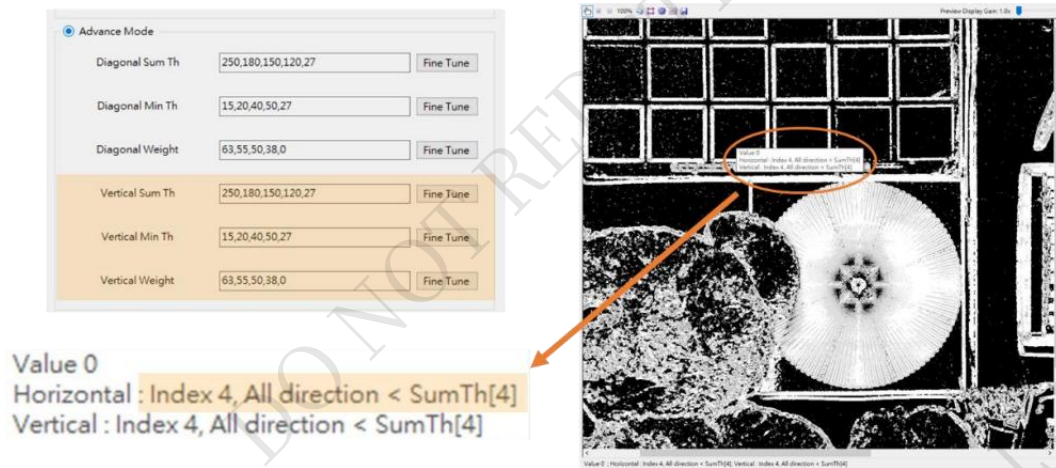


Fig 6.98 Some debug information for tuning.

6.9.7. Parameter: Edge Smooth section

Please refer to Fig 6.99 , the edge smooth page.

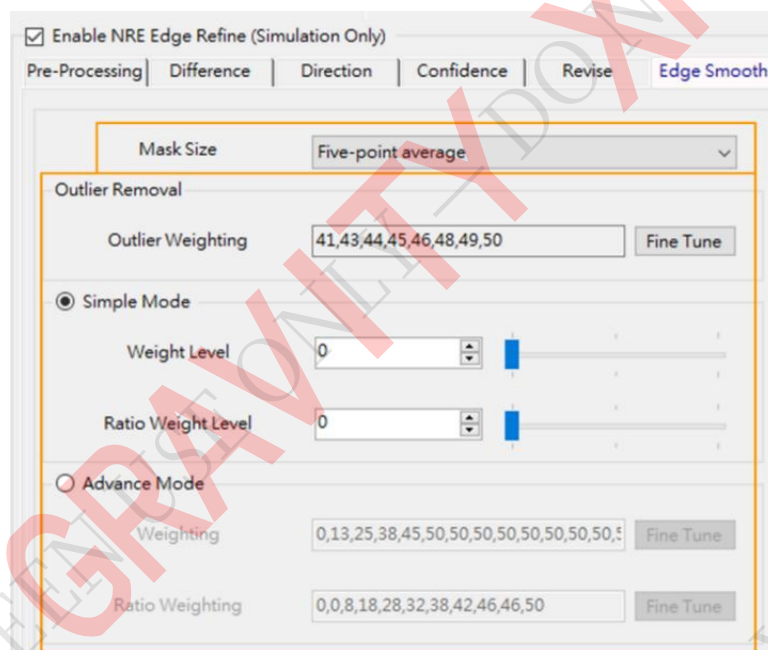


Fig 6.99 The Edge Smooth page.

We determine how to smooth the edge in the part. Such as smooth mode and smooth weighting.

■ Mask size

- There are two selections which are respectively, three-point or five-point average for smooth mode.
- In general case, we usually use five-point average. However, you still can compare the results of the two modes and check which mode works better. As shown in Fig 6.100.
- **Please note:** If you find your input like this (the rightmost picture), please check CFAI parameters or before ER's modulus. Our input is abnormal. This

case is shown in Fig 6.101.

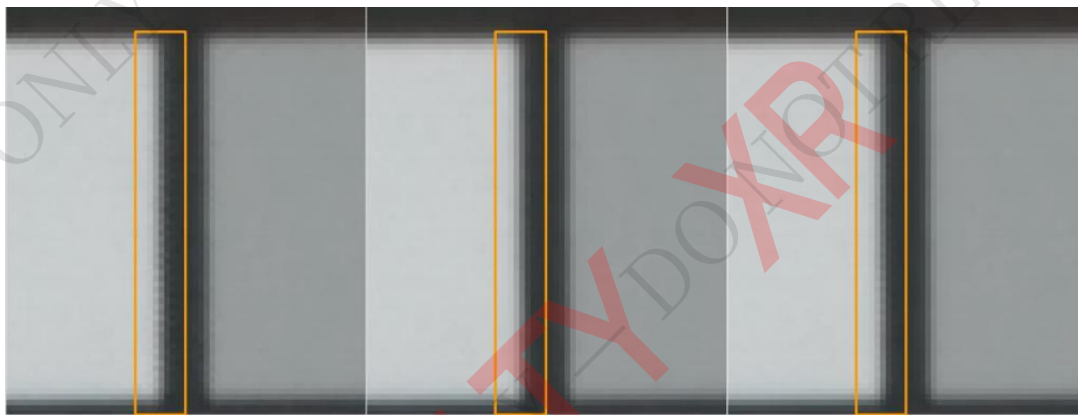


(a) 50lux, w/o ER

(b) 50lux, w/ ER,
three-point average

(c) 50lux, w/ ER,
five-point average

Fig 6.100 The smooth results with different mask size mode



(a) 50lux, w/o ER

(b) 50lux, w/ ER,
three-point average

(c) 50lux, w/ ER,
five-point average

Fig 6.101 The results with different mask size mode

- Outlier weighting (only could be adjusted when outlier removal is open)
 - Debug information can be reference to the **Total Weight Map**.
 - The smoothness of the outlier pixel is determined based on the adaptive curve of the NRE.
 - In general, there is usually more detail in the center of the image, and the image maybe dirtier around it. So, the outlier weight may be weaker at the center and stronger around it. As shown in Fig 6.102.

If shading flag is off, we would only operate by the first value.



(a) Adaptive map, from NRE

(b) w/ shading, total weight
map at corner

(c) w/ shading, total weight
map at center

Fig 6.102 The example tuning for Outlier weighting.

- **Weighting:**
 - You could adjust parameters by referring to the Difference Map results.
 - The higher value of the Difference Map, the higher probability of the edge.
- **Simple mode:**
 - ☐ Weight level: provide three style curves
 - ☐ Level 0~2: Smoothness is from weak to strong.
- **Advance mode:**
 - ☐ You can fill in the parameter values with the corresponding interval based on the value of the Difference Map result.

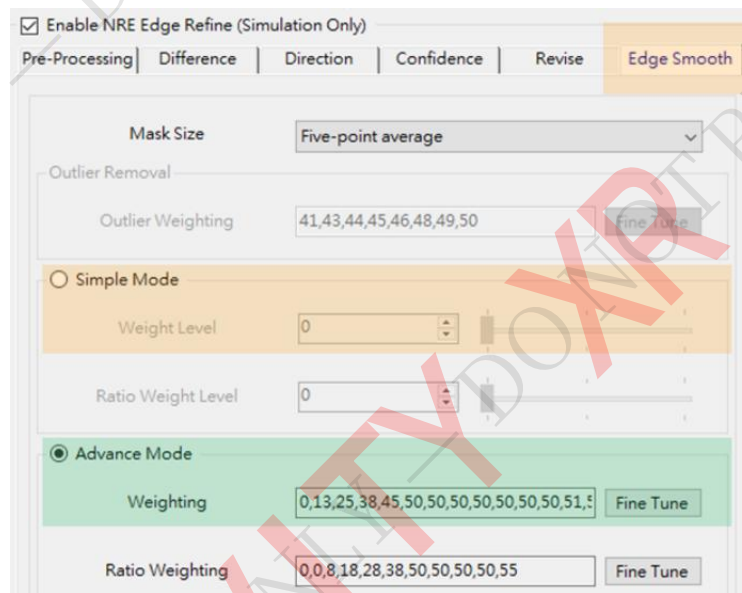


Fig 6.103 Location for Weighting

- **Ratio weighting:**
 - You could adjust parameters by referring to the Confidence Map results.
 - The higher value of the Confidence Map, the higher the confidence level in that direction.

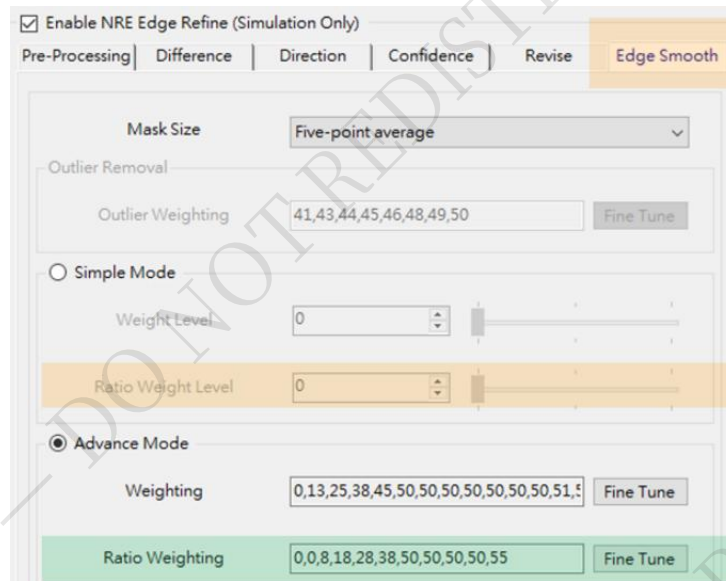


Fig 6.104 Parameter Location for Ratio Weighting

- Overall check parameters:
- The sum of these three weights (outlier weighting, weighting, ratio weighting) is the final total weighting, which will affect the final smoothness.
- If the value of the weighting map is higher, it means that the smoothing will be stronger. As shown below.

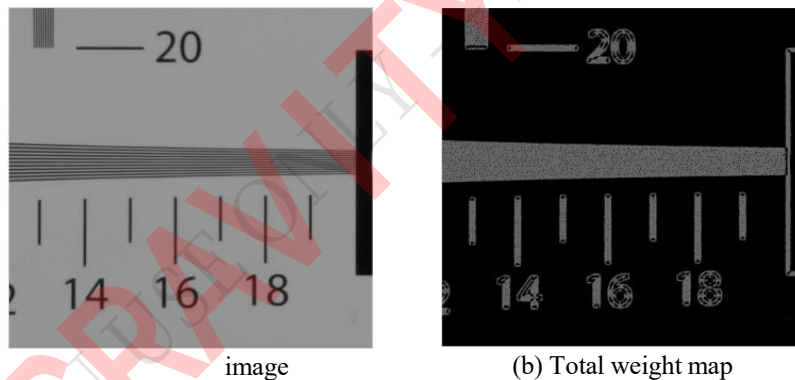


Fig 6.105 Total weight map at hard edge area

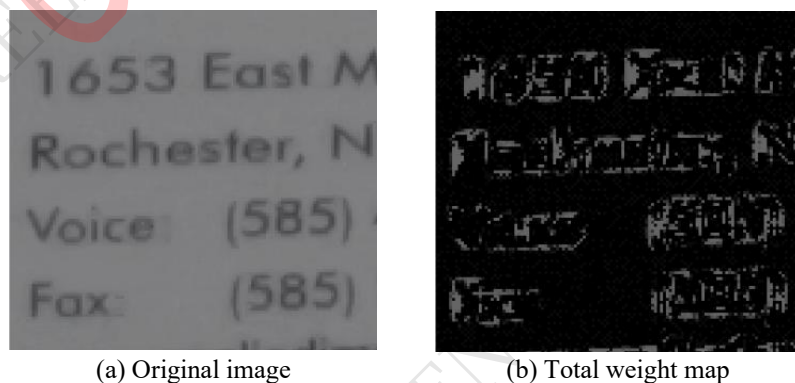


Fig 6.106 Total weight map at text part.

6.9.8. The tuning steps for example:

■ Step1:

First, we observe the noise level of the original image and determine the expected smooth level.

- ☐ Low ISO: The level of smooth could be weaker.
- ☐ High ISO: The level of smooth could be stronger.



(a) 10lux_2700k

(b) 250lux_

Fig 6.107 Two different noise levels.

■ Step2:

Choose debug information, **Total Weight Map**. And observe the results of the current parameters and review the areas where need to be strengthened or weakened. As shown in Fig 6.108 .



Fig 6.108 Total weight map result

■ Step3:

Besides, also need to observe others debug information, there are respectively, **Difference Map**, **Confidence Map**, **Revise Map**. Find out which part causes the detection strength not as strong in the **Total Weight Map**. In the above case, we could find the detection intensity of the **Difference Map** is weaker than others. (As shown in Fig 6.109)



(a) Difference Map

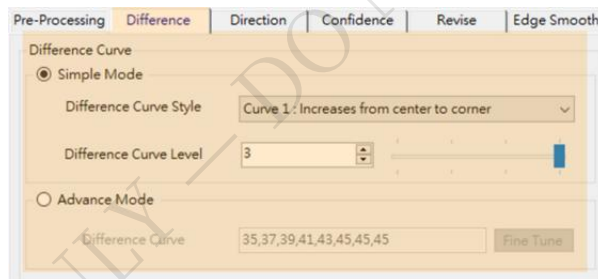
(b) Confidence Map

(c) Revise Map

Fig 6.109 Three debug information .

■ Step4:

- So, we select the **Difference Map**, and observe whether the edge is detected or not. Usually the edge is not effectively detected, which might some characteristics, such as at dark position or the broken edge.
- **Select the Simple Mode, and observe the changes in the Difference Map.** In general, the overall detection intensity can be increased, and the corner of the image can be a little stronger than the center. The fine-tune results as shown in Fig 6.110 (b).



(a) Parameter Location for Difference Curve

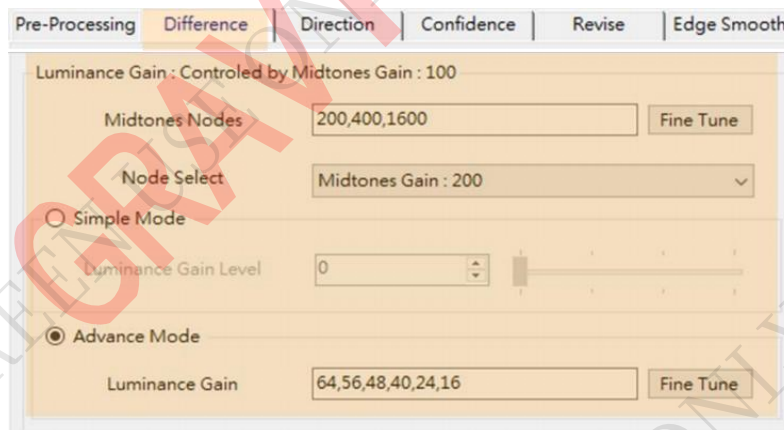


(b) map result

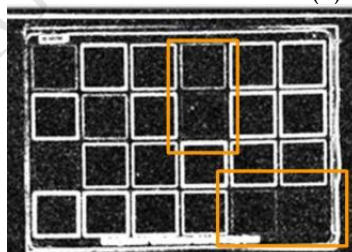
Fig 6.110 Select the Simple Mode, and observe the changes Difference Map

■ Step5:

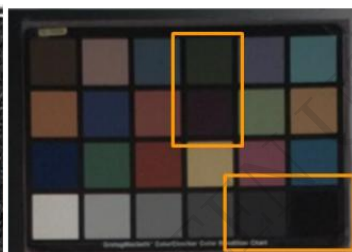
- If the dark area still cannot be detected effectively. We could improve this problem by **adjusting the Luminance Gain Curve**, but it will also result in noise being easily detected. As shown in Fig 6.111. You can use simple mode to quickly try and make it fit the expectation or not.
- Generally, it is not recommended to use this method, because the value of this parameter is determined automatically by midtones.



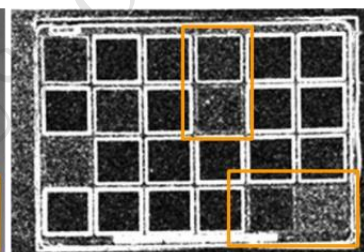
(a) Parameter Location for Luminance Gain



(b) The area is where we want to be enhanced



(c) Compare with original input image.

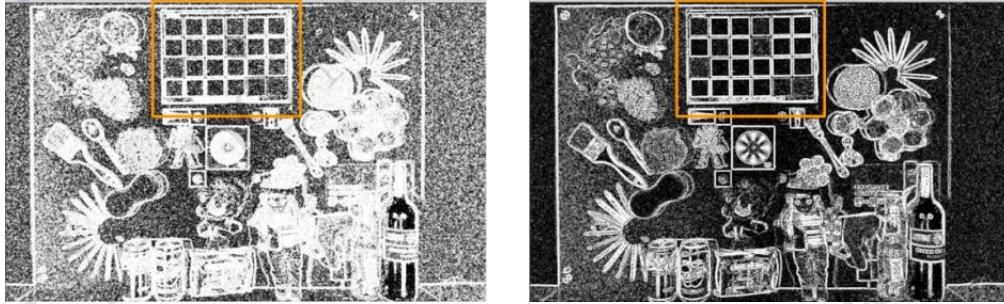


(d) After tuning

Fig 6.111 How to adjusting the Luminance Gain Curve.

■ Step6:

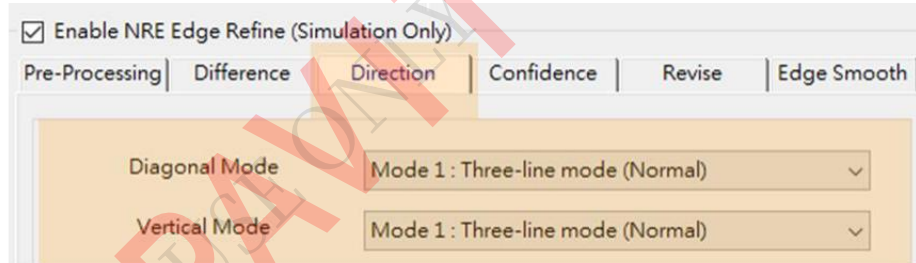
- Then, check the result of the **Revise Map**. If you want to have a stronger smoothing effect, please make sure the most of the edges are detected.
- In addition, you also need to check the total weight map to ensure that it is as expected. **For other flat areas, it doesn't matter if they're wiped at high ISO because the NRE also will smooth them.** As shown in Fig6.112.



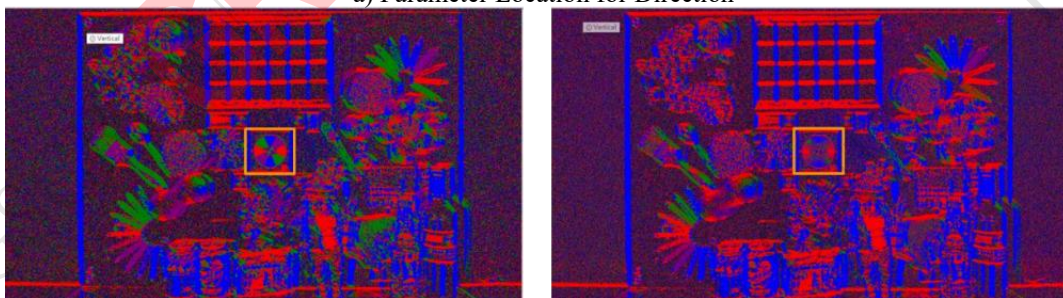
(a) The area is where we want to be enhanced. (b) Compare with original image.
Fig 6.112 Check the result of the Revise

■ Step7:

- Check the **Direction Map** to make sure that direction of the edge you want to smooth is detected. As shown in Fig 6.113
- **Please note:** If a position didn't be direction, it would be not smoothed at all. Besides, our common is Mode 1 (Three-line mode, normal).



a) Parameter Location for Direction



(b) Diagonal, Vertical: Mode1 (c) Diagonal, Vertical: Mode2

Fig 6.113 Check the Direction Map

6.10. LTM

LTM (Local Tone Mapping) applies tone mapping gain to each pixel depending on the local brightness to enhance visibility. There are two stage, statistics stage and estimation stage.

In the statistics stage, get pseudo Y value from LTM estimation stage. And save stats data which will be used in the next frame estimation stage.

In the estimation stage, estimate LTM Gain after NRE, and apply LTM Gain on RGB Sep image which is converted from YCC420 Sep. The final output is 12 bits RGB Sep image.

6.10.1. LTM Statistics

- LTM stats is used to calculate local average mean map.
- Collect Y into spatial and intensity grids.
- Apply bilateral filter to the grids to weighting average.

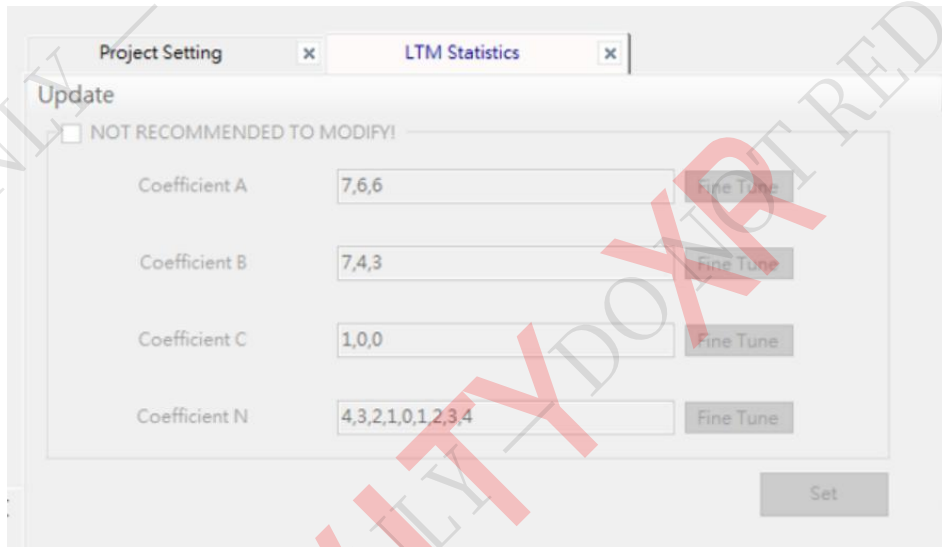


Fig 6.114 The interface of LTM statistics tuning

- Bilateral filter coefficients
 - Coefficient A~C: To create the spatial domain 3x3 filter coefficients. It's recommended not to modify.
n Range:A、 B: 0~7; C: 0~1
 - formula
 - $F0 = 2^{a0} + 2^{b0} - c0$
 - if $a0 = b0 = c0 = 0$, $F0=0$.
 - $F1 = 2^{a1} + 2^{b1} - c1$
 - if $a1 = b1 = c1 = 0$, $F1=0$.
 - $F2 = 2^{a2} + 2^{b2} - c2$
 - if $a2 = b2 = c2 = 0$, $F2=0$.
 - Increase the outer coefficients, make the center grid stats more average with outer grid stats.

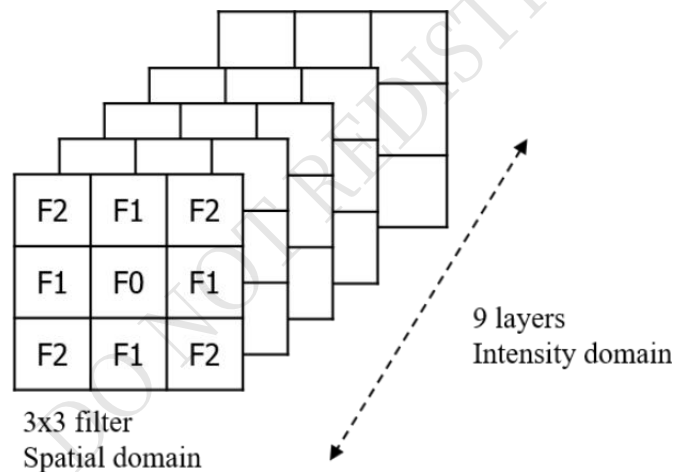


Fig 6.115 filter coefficients layout

■ Bilateral filter coefficients

□ Coefficient N

- The bits shift value to each intensity layers, which is as the weighting of each layers.
- It's recommended not to modify
- Range:0~15
- Increase the value, less weighting on the intensity layer.
- If the values are the same in the outer layer and center layer, smoother to the grid stats. But may cause halo in the LTM result.

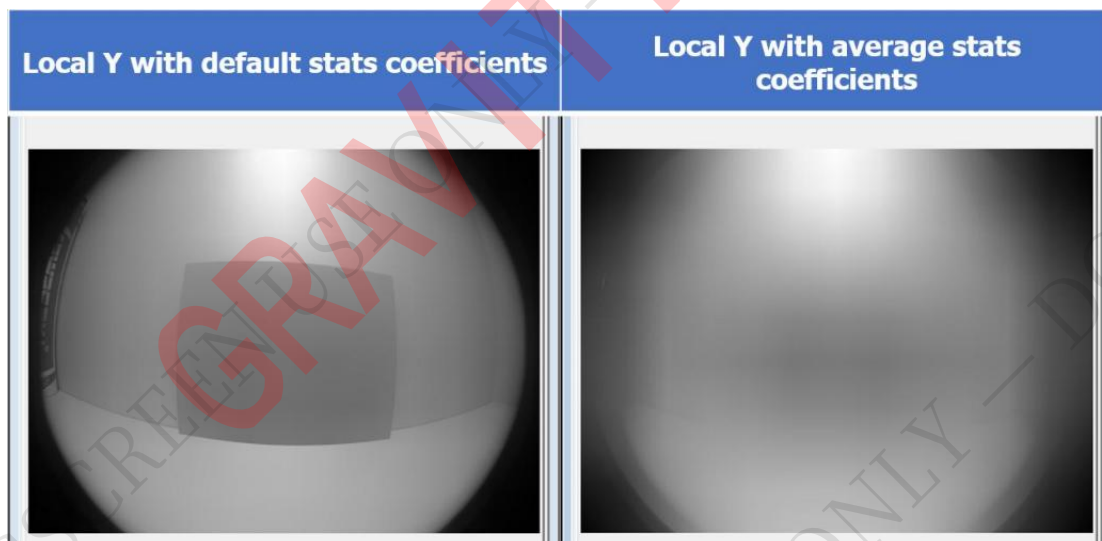


Fig 6.116 local average mean map with different filter coefficients.

6.10.2. LTM Estimation

- First, convert YCC420s to RGBs before apply LTM.
- LTM applies local tone mapping gain to each pixel depending on the local brightness to enhance visibility. Include
 - Local Brightness Enhancement
 - Local Contrast Enhancement
- If the midtone and shadows gain are around 1.0x, use the User defined LTM curve

- and scale table in this block.
- After apply LTM gain, output is 12bits RGBs.
- And then, apply inverse inner gamma on 12bits RGBs. The final output is still 12bits RGBs.

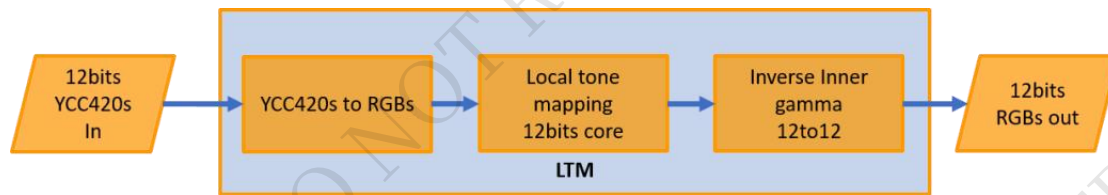


Fig 6.117 The flow of LTM

6.10.2.1. Tool Bar UI

Please refer to Fig. 5.108, when you load a RAW image, there is a toolbar in LTM,

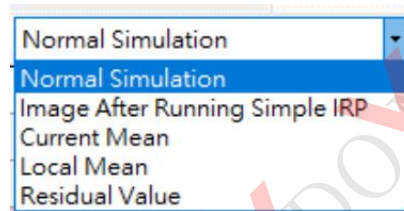


Fig 6.118 UI of

- Preview Type
 - Normal Simulation:
 - show preview as LTM result.
 - Image After Running Simple IRP:
 - show preview as LTM result + IRP S2.
 - Current Mean:
 - Current mean(=Y) is the Y mean of the ltm input image.
 - show preview as Y mean image, which is used to estimate LTM.
 - Move cursor on the preview, get Current Mean Index for tuning user defined LTM Curve. (Fig 6.119)
 - Local Mean:
 - Local mean(=L) is the local average Y mean.
 - show preview as Local Y mean image, which is used to estimate LTM.
 - Move cursor on the preview, get Local Mean Index for tuning user defined Scale Curve. (Fig 6.119)
 - Residual Value:
 - show preview as Residual Value.
 - Residual Value = $(Y-L)*c0*c1$
 - c0 is the boost gain LUT from Brighter/Darker Enhance Curve.
 - c1 is the boost gain from Brighter/Darker Strength.
 - Move cursor on the preview, get Local Enhance Index for tuning Local Brighter/Darker Enhance (Local Contrast Enhance).
 - If Residual Value < 0, tune the Darker Enhance Curve. Otherwise, tune Brighter Enhance Curve. (Fig 6.119)

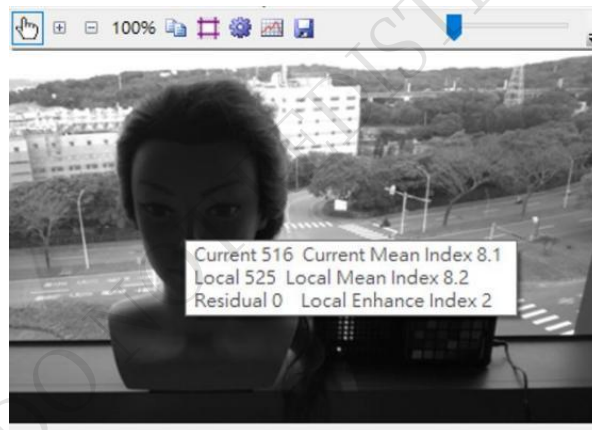


Fig 6.119 Tuning Info from Preview

6.10.2.2. The Order of Tuning

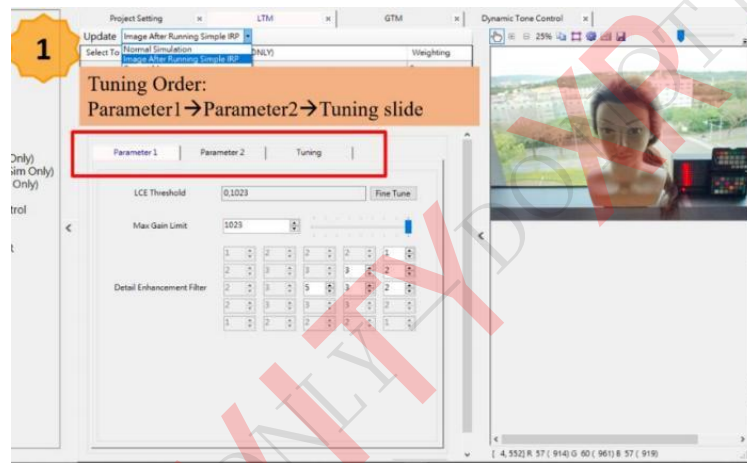


Fig 6.120 The LTM Interface

There are three slides to tune. The Tuning order is Parameter1 → Parameter2 → Tuning slide.

6.10.2.3.

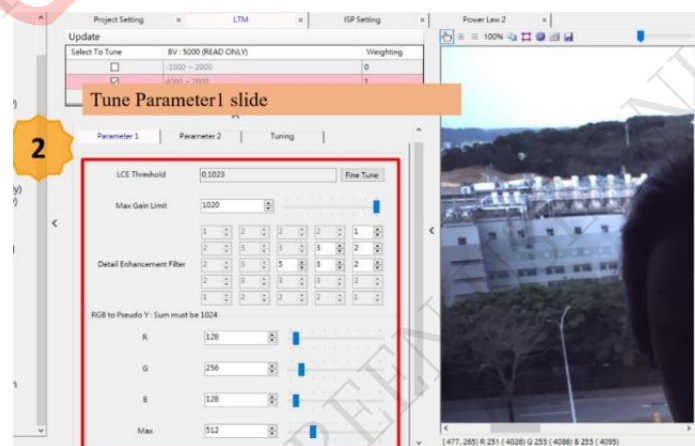


Fig 6.121 The Parameter1 Interface

- LCE Threshold
 - ❑ the min and max clamp of residual value.
 - ❑ Range: 0~4095
- Max Gain Limitation
 - ❑ The max limitation of LTM gain boost.
 - ❑ Range: 0~1023
- Detail Enhancement Filter
 - ❑ The coefficient of 5x5 gaussian filter, to enhance local detail.
 - ❑ Range: 0~7
 - ❑ Recommend the Default = (5,3,3,2,2,1)
 - ❑ The coefficients are smoother, like average filter, more detail enhanced. But may cause halo effect around the texture edge. (Fig 6.120)
- RGBs to pseudo Y
 - ❑ Before LTM core, calculate pseudo Y to estimate and LUT.
 - ❑ Finetune c1~c4, the coefficients of R,G,B and max(R,G,B).
 - ❑ Range:0~2047
 - ❑ RGB to Pseudo Y coefficients, sum must be 1024.
 - ❑ Pseudo Y = $(R*c1 + G*c2 + B*c3 + \max(R,G,B)*c4) \gg 10$;
 - ❑ Default = (128,256,125,512)

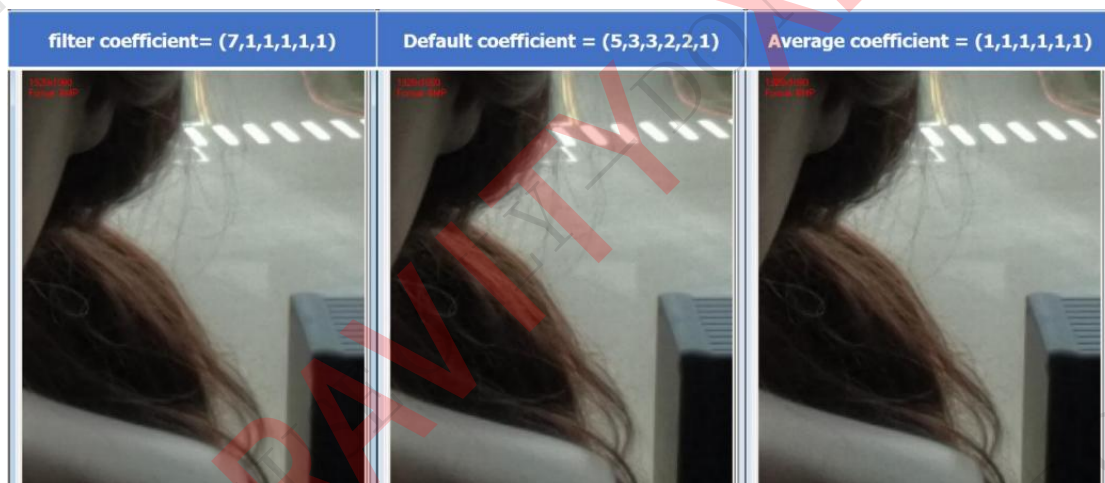


Fig 6.122 The diff of detail enhancement filter coefficients

6.10.2.4. Parameter2

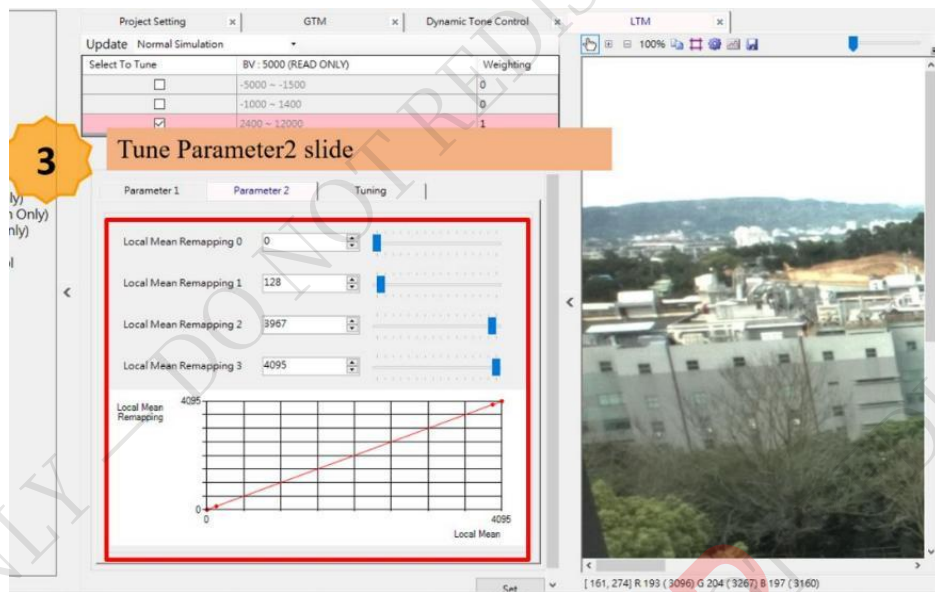


Fig 6.123 The Parameter2 Interface

■ Local Mean Remapping 0~3

- ❑ tune the four points to control tone curve for Local Mean re-mapping.
- ❑ Range: 0~4095
- ❑ Default: (0,128,3967,4095)
- ❑ Decreasing the curve of dark key to boost brightness in the dark area.
- ❑ Increasing the curve of high key to suppress brightness in the bright area.
- ❑ The distance should be larger to make tone smooth changing between P0~P1 and P2~P3.

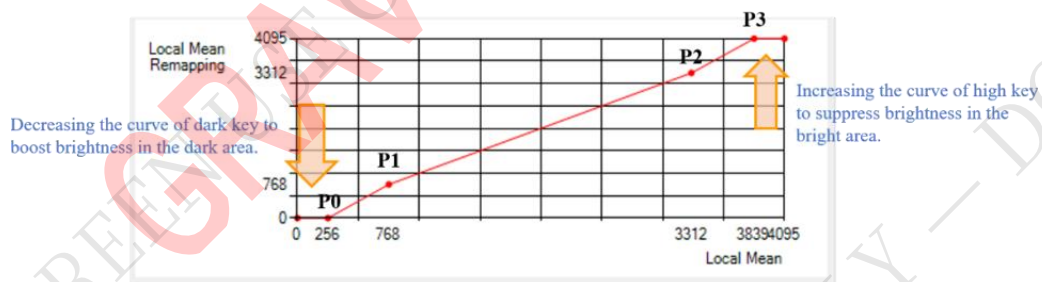
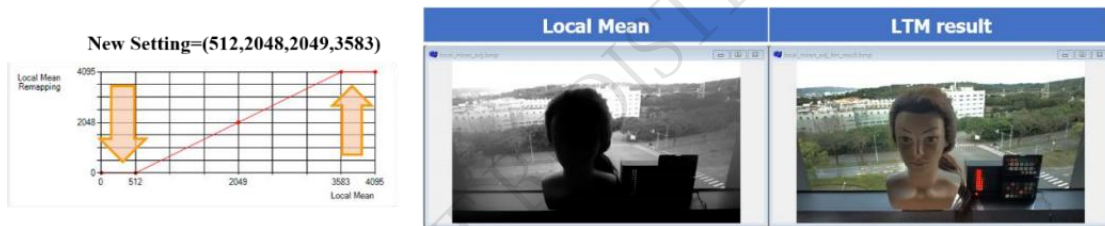


Fig 6.124 local mean remapping curve



Fig 6.125 LTM result with default local mean remapping setting



Decreasing the curve of dark key to boost brightness in the dark area.
Increasing the curve of high key to suppress brightness in the bright area.

Fig 6.126 LTM result with new local mean remapping setting

6.10.2.5. Local Brightness Tuning

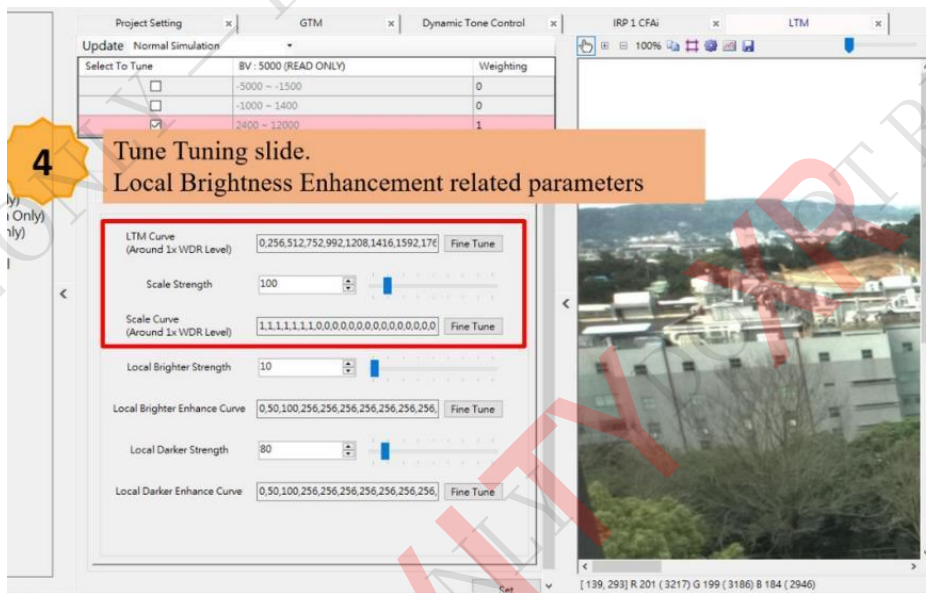


Fig 6.127 Local Brightness Enhancement Interface

- Local Brightness Enhancement related parameters:
 - LTM Curve
 - The user Defined brightness increment ($= \Delta Y(Y)$) of each Current mean index.
 - Range: 0~65535
 - Effective conditions:
 - midtone gain and shadows gain are around 1.0x.
 - Increasing these parameters according to the Current Mean Index, make image brighter. (Fig 6.128)
 - curve should be smooth to avoid brightness inversion
 - Scale Curve
 - The User Defined Brightness Boost Gain ($= \text{gain1}(L)$) of each Local Mean Index.
 - Range: 0(0.0x) ~32767(31.99x)
 - Effective conditions:
 - midtone gain and shadows gain are around 1.0x.
 - Increasing these parameters according to the Local Mean Index, make image brighter. (Fig 6.129)
 - curve should be smooth to avoid brightness inversion
 - Scale Strength

- Increasing strength (= gain2), enhance local brightness (Fig 6.130)
- Range: 0(0.0x) ~800(8.0x)
- Default = 100(1.0x)
- Local Brightness Enhancement = $Y + \Delta Y(Y) * \text{gain1}(L) * \text{gain2}$

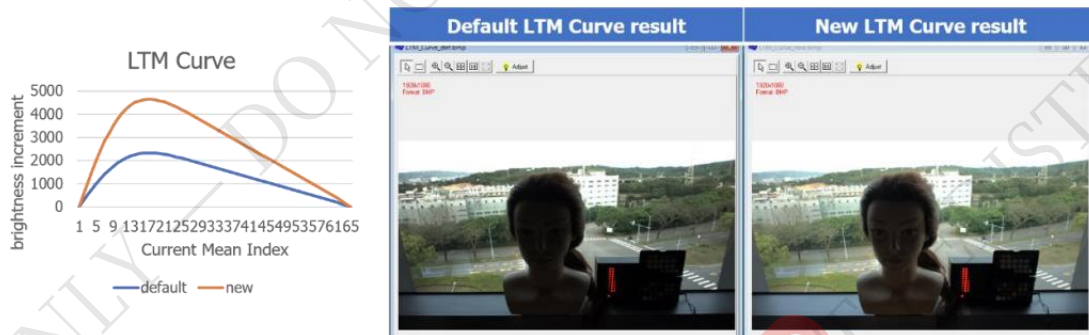


Fig 6.128 LTM result with difference



Fig 6.129 LTM result with difference Scale Curve



Fig 6.130 LTM result with difference Scale Strength

6.10.2.6. Local Contrast Tuning

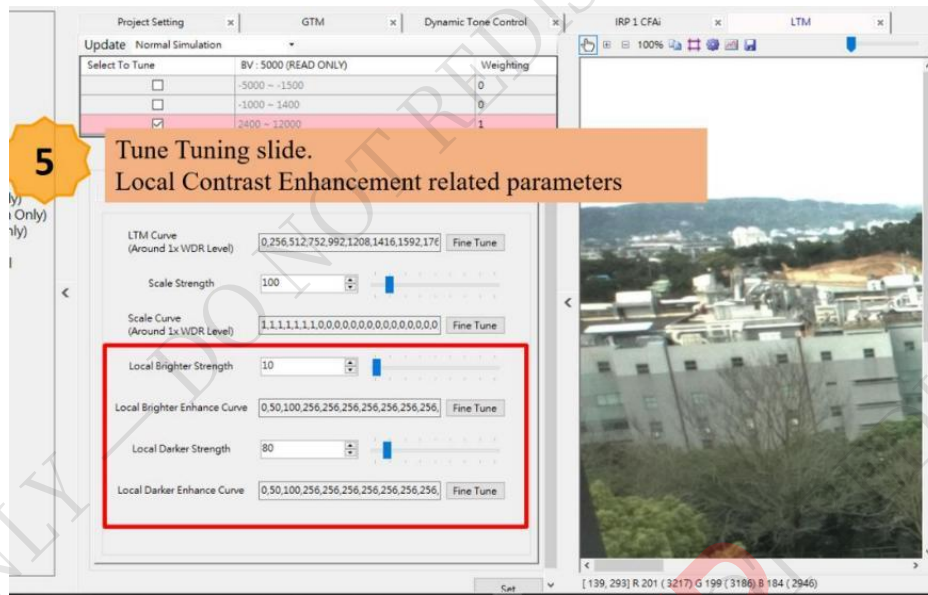
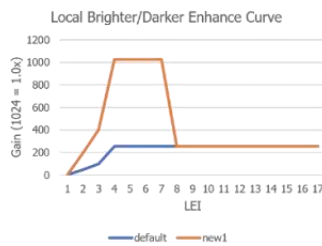


Fig 6.131 Contrast Enhancement

- Local Contrast Enhancement related parameters:
 - Local Brighter/Darker Enhance Curve
 - tune the gain value in this curve according to the Local Enhance Index (LEI)
 - Range:0(0x) ~65535(63.99x)
 - Increase enhancement gain to make local bright/dark area brighter/darker, to enhance local contrast. (Fig 6.132)
 - Local Brighter/Darker Strength
 - The strength to make local bright/dark area brighter/darker, to enhance Fig 6.133)
 - Suggest to lower the gain of the first or second LEI in the enhance curve.
 - Avoid to enhance noise in the dark area.
- Local Contrast Enhancement = $(Y-L) * c0(LEI) * c1$
 - Residual Value = $(Y-L) * c0(LEI) * c1$
 - Current mean(=Y) is the Y mean of the ltm input image.
is the local average Y mean.
 - C0(LEI) is the boost gain from Brighter/Darker Enhance Curve according Local Enhance Index (LEI).
is the boost gain from Brighter/Darker Strength.
 - If Residual Value <0, tune the Darker Enhance Curve.
Brighter Enhance Curve.



Enhance Local Contrast in the low to middle key.

Fig 6.132 Local Brighter/Darker Enhance Curve



Fig 6.133 Different Brighter/Darker

6.11. HWIRP Stage 2

6.11.1. CCM

CCM is applied to map the characteristic of color from sensor to human eye.

■ Notice : Light source condition

- ☐ High color temperature, like D65 (6500K± 200K) @ above 320 lux
- ☐ Middle color temperature, like CWF or TL84 (4000K± 200K) @ above 320 lux
- ☐ Low color temperature, like incandescent (2700K± 200K) @ above 320 lux

CCM Main Page Interface Introduction:

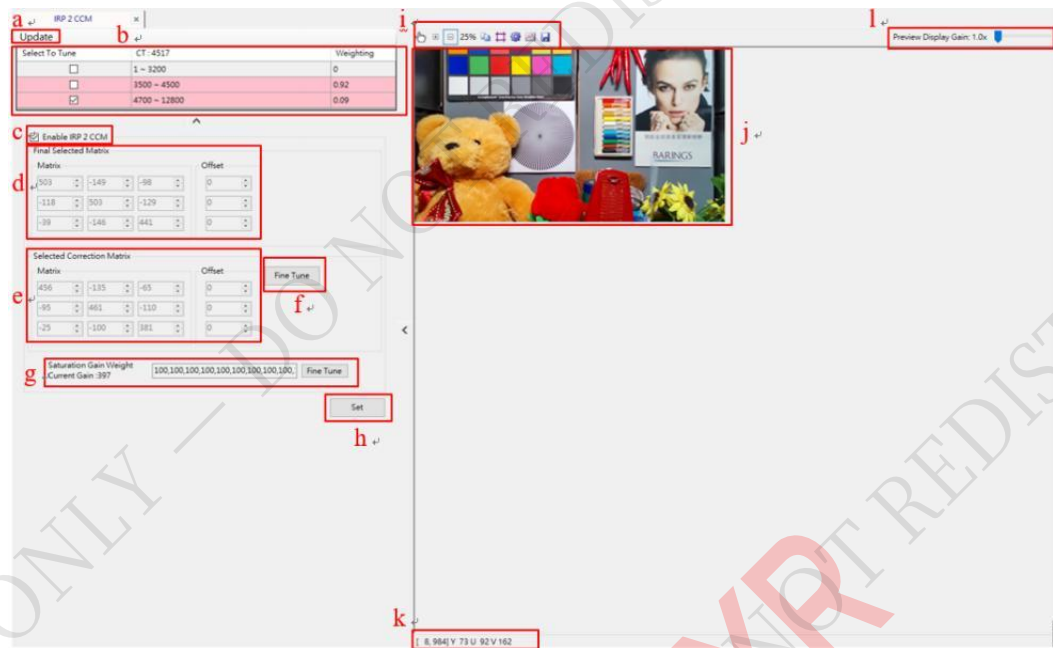


Fig 6.134

- (a) Update button
Click "Update" and the CCM parameter will be updated to tuning tool main form.
- (b) CT (color temperature) interpolation
CCM is divided into five groups according to the color temperature, different color temperatures use the same matrix or matrix which linear interpolation of two color temperatures.

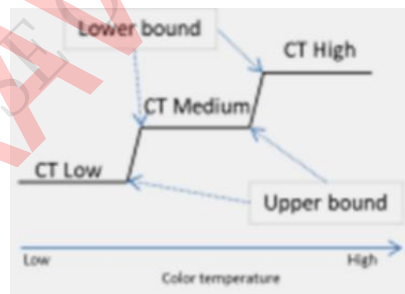


Fig 6.135

Select To Tune	CT: 4517	Weighting
☐	1 ~ 3200	0
☒	3500 ~ 4500	0.92
☐	4700 ~ 12800	0.09

Fig 6.136

1. Select To Tune
Select the CT region to tune CCM.
2. Current CT(color temperature)
Show the color temperature of the current simulation.
3. CT region setting

The CT region can be set according to needs. Usually divided into three color temperatures: high, medium and low. See CCM Notice.

The adjustment method is to directly modify the CT number, then push the keyboard's ENTER. After push the "Update" button, setting will be done.

4. Weighting

Show the weighting to use when linearly interpolating at current simulation.

The example shown above: the current CT is 4517K, so the final CCM use the linear interpolation of the second matrix and the third matrix. The weighting of the second and the third matrix are 0.92 and 0.09. The second and the third region change to red background when the weighting don't equal to zero. So we can tune the second matrix and the third matrix at this case.

(c) Enable IRP2 CCM

Use the CCM parameter (on) or identity matrix (off).

(d) Final Selected matrix

Show the final CCM of the current simulation.

(e) Selected correction matrix

Show the selected matrix according to selected CT region (see [\(b\) CT interpolation table](#)).

(f) Selected correction matrix (Fine Tune), see **CCM Fine Tune Interface Introduction**

Click "Fine Tune" to fine tune selected matrix.

(g) Saturation Gain Weight (Fine Tune)

Fine tune the gain-weight table to adjust saturation weighting according to gain.



Fig 6.137

1. Current Gain means sensor gain * digital gain * midtones gain * shadow gain (See page ISP setting -> Condition)

2. The value of table means weighting, example: 100 means 100%

(h) Set button

Click "Set" to apply parameter to (j) simulation result.

(i) Image toolbar

A1. Move Image

Move the simulation image.

A2. Zoom In Image

Zoom in the simulation image.

A3. Zoom Out Image

Zoom out the simulation image.

A4. Compare Image

Push button and show the simulation image by last updated parameter.

A5. Selection

Click to select regions of interest for analysis.

A6. Setting

Choose selection type for A5.

A7. Data analysis

Data analysis for selected regions of A5.

A8. Save Bitmap

Save simulation image.

- (j) Simulation results
Display current simulation result.
- (k) Mouse information
Show related information on mouse pointer.
- (l) Preview Display Gain
Adjusts the gain of the simulation display.

CCM Fine Tune Interface Introduction:

At fine tune interface, we can choose a target to approach or fine tune matrix directly.

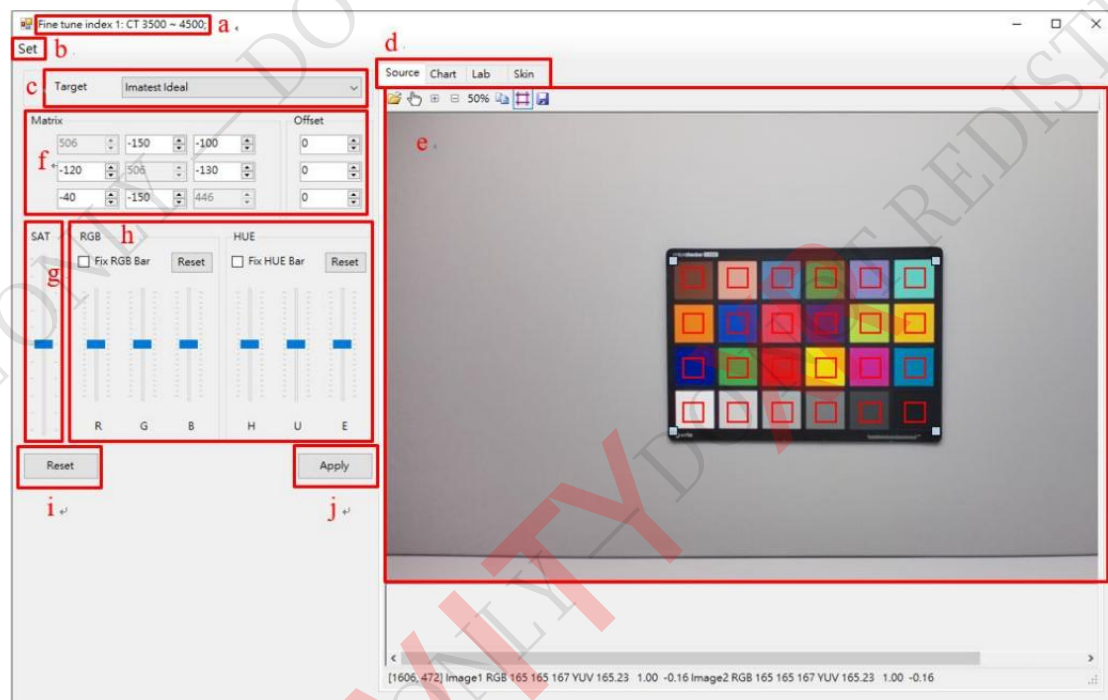


Fig 6.138

- (a) Fine tune
Show the selected CTregion ofCCM main page. The fine tune page only adjusts parameters ofselected region.
- (b) Set
parameters to CCM main page.
- (c) Fine tune page, we can optimize CCM by approaching a selected target.

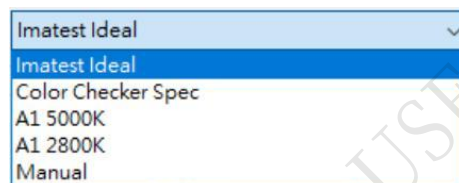


Fig 6.139

There are several target to choose currently. The “Manual” type could choose any target to approach. There will be examples later.

- (d) Analysis toolbar
 1. Source

Load a raw image with 24 color checker to analysis.

2. Chart

Details of the source (final simulation) and target.

User can adjust the weight of optimization algorithm.

Source Chart Lab Skin

Delta_C (Weighted Avg 4.090 Max 16.250) Delta_E (Weighted Avg 5.515 Max 16.673)

	Weight	Source	R	G	B	Target	R	G	B	Delta C	Delta E
☒	1		105	68	54		113.985	80.835	67.575	2.650	5.440
☒	1		209	165	153		194.820	147.900	127.755	5.113	8.046
☒	1		115	145	178		92.820	122.400	156.060	0.746	8.639
☒	1		97	126	75		90.525	107.610	64.515	6.366	8.995
☒	1		150	149	190		129.285	128.010	176.205	4.745	9.089
☒	1		137	201	193		97.410	190.995	170.850	11.543	12.692
☒	1		212	126	58		221.085	122.655	47.685	7.312	7.325
☒	1		82	107	185		70.635	90.780	170.340	3.100	6.766
☒	1		206	92	104		193.290	82.110	97.410	2.039	4.423
☒	1		88	61	104		92.055	57.375	106.335	4.908	4.918
☒	1		171	199	78		160.395	189.210	61.710	2.808	4.602
☒	1		215	181	49		228.225	160.650	41.310	16.250	16.673
☒	5		39	55	134		39.525	62.730	146.880	2.675	4.234
☒	5		76	153	74		70.635	149.940	72.675	0.960	1.546
☒	5		183	51	57		173.655	50.745	56.865	3.842	4.236
☒	1		221	205	37		236.640	198.135	19.635	10.512	10.520
☒	1		187	84	152		188.190	83.895	151.470	0.633	0.648
☒	1		43	134	171		0.000	137.700	168.300	6.577	6.593
☒	0		224	223	222		244.800	245.310	242.250	1.073	7.738
☒	0		202	202	202		200.430	202.215	202.215	0.668	0.671
☒	0		169	169	169		160.905	162.945	163.200	0.775	2.542
☒	0		120	121	121		120.870	121.125	121.635	0.254	0.281
☒	0		67	67	68		82.620	84.150	85.680	0.865	7.173
☒	0		33	33	33		48.705	49.470	50.745	0.783	7.731

Max Count of Over-Saturated 0 Cost Ratio from Mean to Max (0 : All Mean; 1.0 : All Max) 0.00 Optimize

Fig 6.140

3. Lab

Show the distribution of the Current (source before approach), Tuning (source after approach), and Target at CIELab color space.

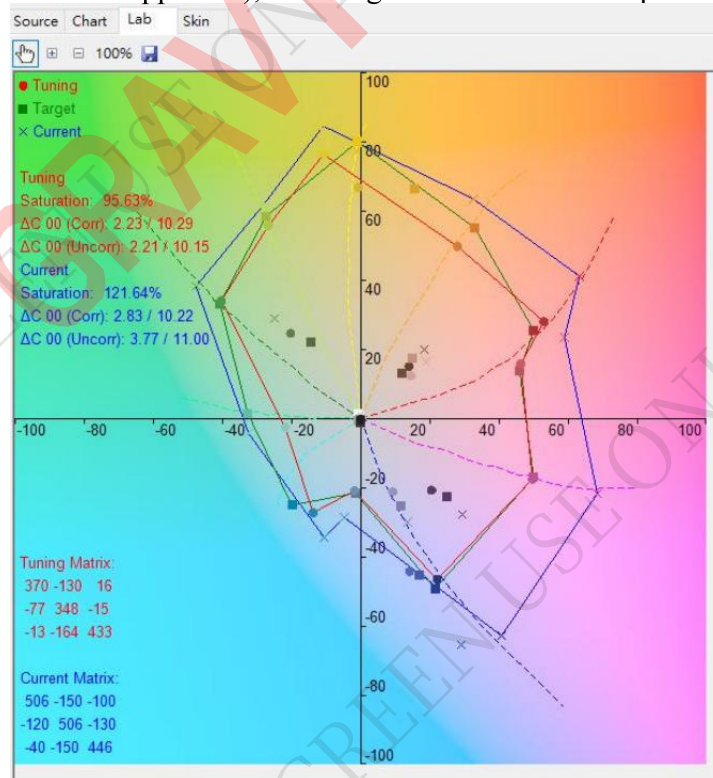
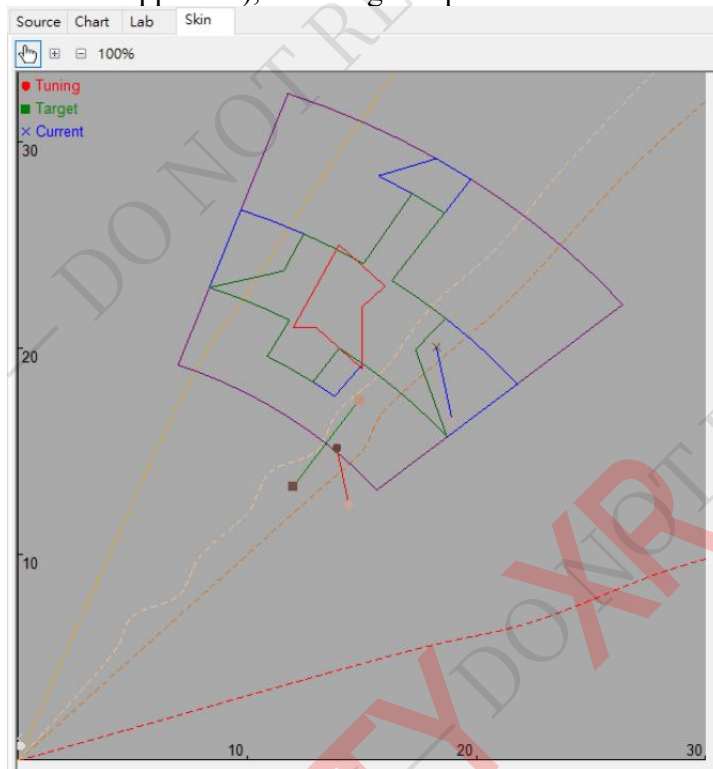


Fig 6.141

4. Skin

Show the skin distribution of the Current (source before approach), Tuning (source after approach), and Target at part of the CIELab color space.



Saturation Gain Weight Interface Introduction:

Adjust saturation weighting according to gain (sensor gain * digital gain * midtones gain * shadow gain).

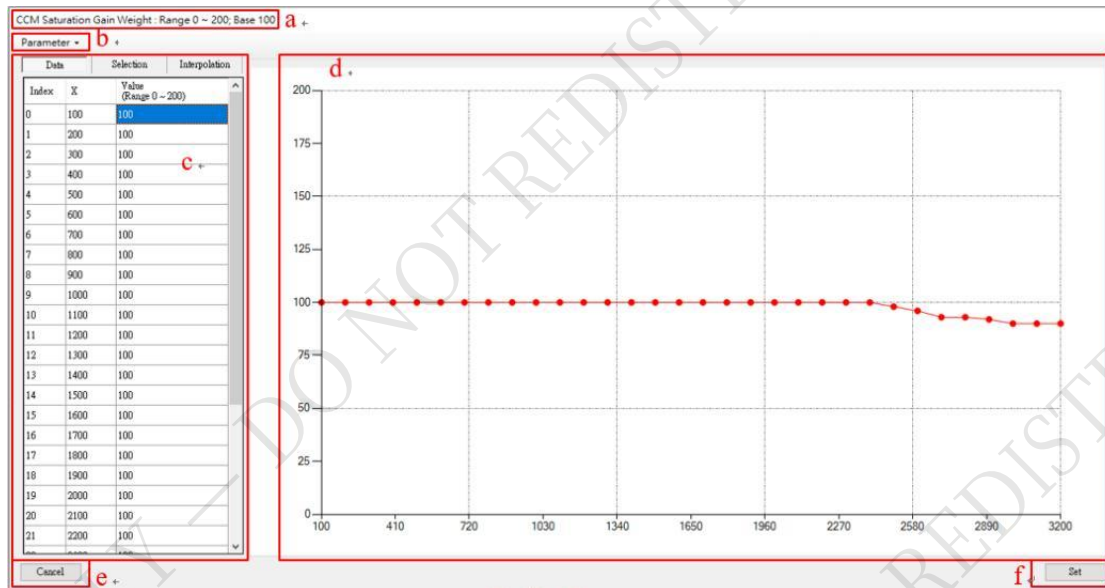


Fig 6.142

(a) Weighting range and gain base

1. The weighting range are between 0 to 200. The value means weighting, example: 100 means 100%.
2. Gain base is 100. 100 means 1x gain.

(b) Parameter

Save or load the weighting table.

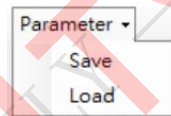


Fig 6.143

(c) Weighting

1. Data

User could edit value directly.

Index: The label of the data on the table.

means 1x gain)

range (0~200) (100 means 100%)

value with range of x once at a time.

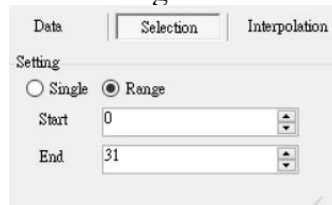


Fig 6.144

3. Interpolation

Linearly interpolate "range" with the range of "x".

The dialog box has three tabs: Data, Selection, and Interpolation. The Interpolation tab is active, showing a table with 'Index' and 'Value' columns. The 'Start' row has '0' in the Index column and '100' in the Value column. The 'End' row has '31' in the Index column and '90' in the Value column. An 'Apply' button is at the bottom right.

	Index	Value
Start	0	100
End	31	90

Fig 6.145

- (d) Weighting edition figure
x-axis: gain
y-axis: value
User could adjust the point directly.
Selection also be applied here.
- (e) Cancel button
Cancel the page without saving.
- (f) Set button
Update the parameters of this page to the CCM main page.

The process to tune CCM:

1. Check LSC calibration, LSC tuning, AWB tuning completed.
2. Load raw image and set condition on ISP Setting page.
Finally, push Update button.

The dialog box has tabs for 'Update', 'Image', and 'Condition'. The 'Image' tab is active. It includes a 'Calculate WB Gain & CT' checkbox (checked), a 'Sensor Image' radio button (selected), and an 'Image Setting' section with fields for Width (2592), Height (1944), Bit (10), and Pixel Pattern (Bayer Image (GR)). A 'File' field contains 'blackbox_bv1465_ad230_mid100_shad173.raw' and a 'Load' button. There is also a 'Simulate Data' option and another 'Load' button.

Fig 6.146

The dialog box has tabs for 'Update', 'Image', and 'Condition'. The 'Condition' tab is active. It shows various gain and level parameters with sliders and input fields: Sensor Gain (on Image) 230, Digital Gain (on Pipeline) 1000, Midtones Gain 100, Shadows Gain 173, Color Temperature (CT) 4140, Brightness Value (BV) 1465, and WDR Level 173.

Fig 6.147

3. Select the corresponding CT area to adjust the parameters on CCM main page. Push the Fine Tune button to switch to the fine tune page.

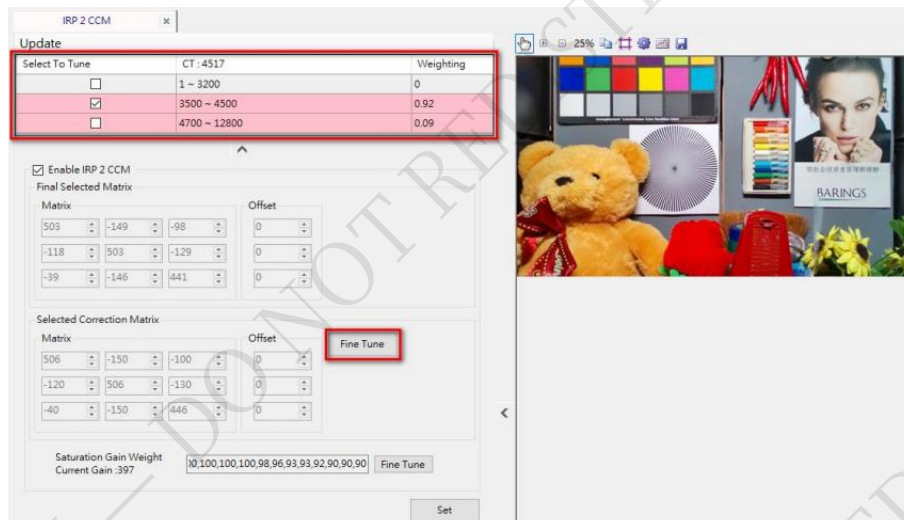


Fig 6.148

4. Matrix tuning with manual mode or optimize mode

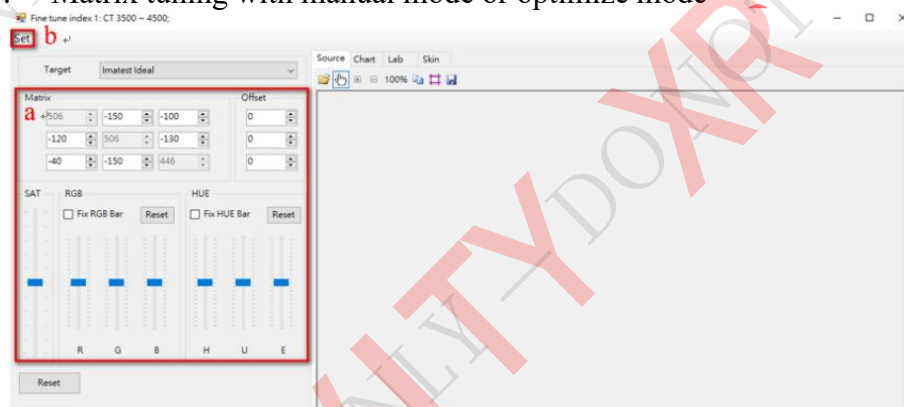


Fig 6.149

Manual mode:

- (a) Users can set matrix or adjust saturation or RGB/HUE manually.
- (b) Click “Set”, the parameters will be updated to CCM main page.

5. Check tuning result

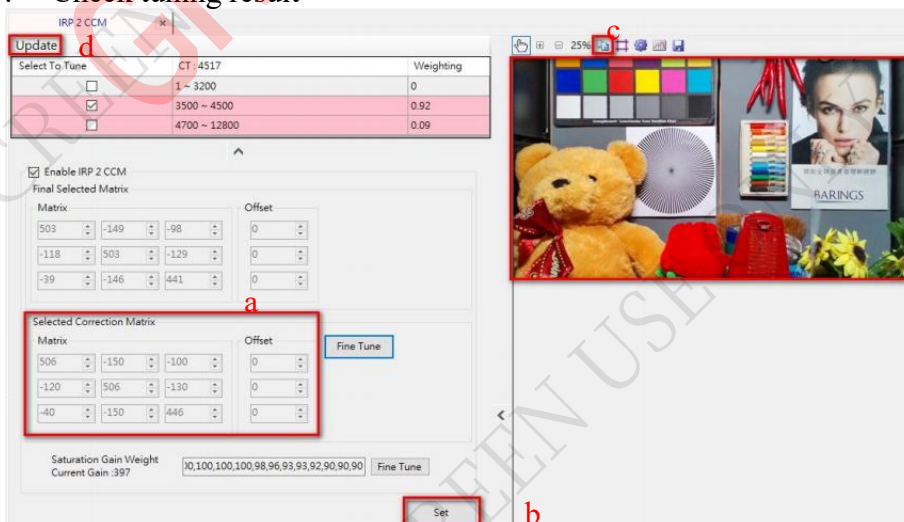


Fig 6.150

- Check if the parameters here have been updated.
- Click “Set”, run simulation with new parameters.
- The simulation result will display on window. User can compare new parameters and parameters of tuning tool main page by image toolbar.
- If tuning complete, click “Update” to update the new parameters to tuning tool main page.

The example of optimize mode:

This example will show how to choose a target to optimize CCM.

- Load color chart raw image. (Note: The CT of the raw image must be in the selected range) Lower sensor adgain is recommended.
- Select the color patch of source image.

After loading image, the color of image may be a bit strange. But it's okay. This is because when loading images here, tuning tool doesn't know what the color temperature is. So, the default WB gain is set from the CCM main form. When users choose target (next step), source will be applied with the same WB GAIN as target. Under the same WB standard, the next will optimize a set of CCM to approach target.

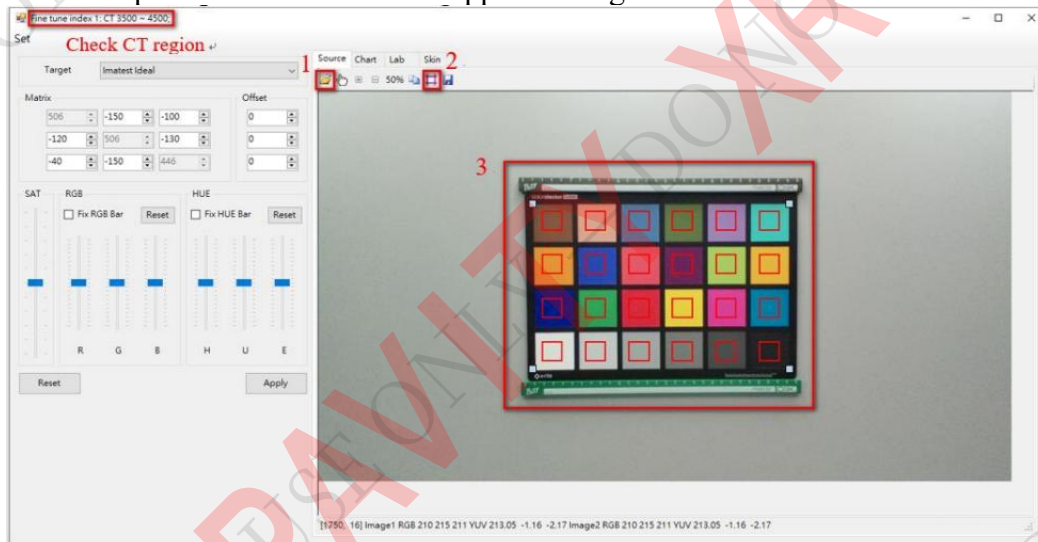


Fig 6.151

- Select the target type. We choose “Manual” in this example. This type can choose any target to optimize CCM.

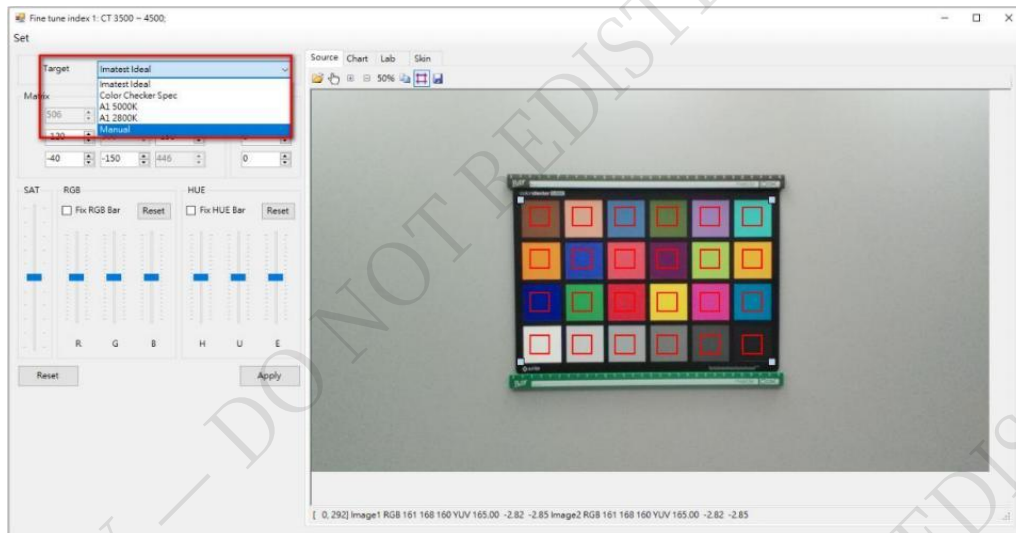


Fig 6.152

4. After choose “Manual” type, the target page will appear to the right of source page (only for “Manual” type). Load target image (jpg or bitmap), and select the color patch again. Note that for better estimation result, the G mean on patch 22 between source image and target image should not have large difference (15%). Otherwise, tuning tool will show some warning message to notify user to confirm chart image first.

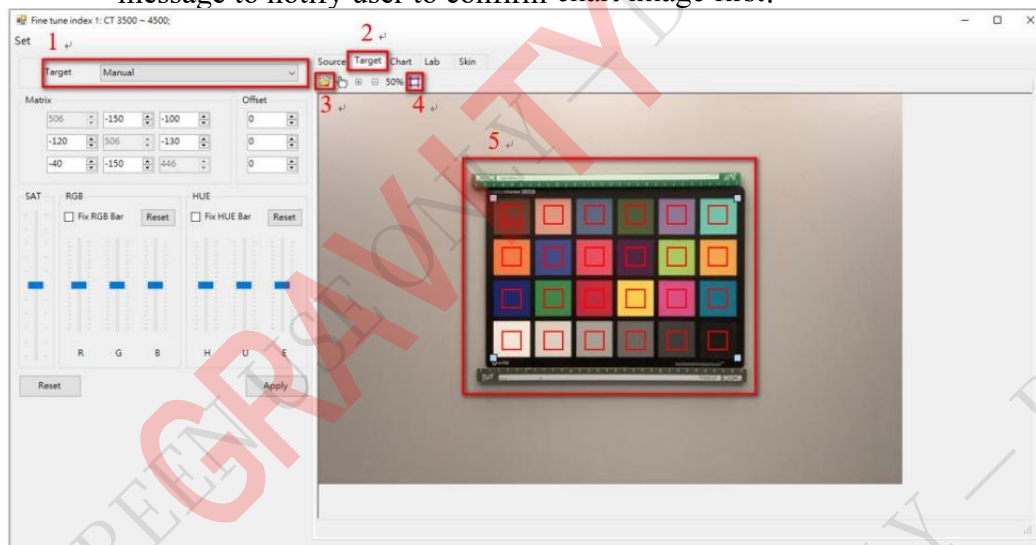


Fig 6.153

5. Click “optimize” button on “Chart” page to calculate matrix.

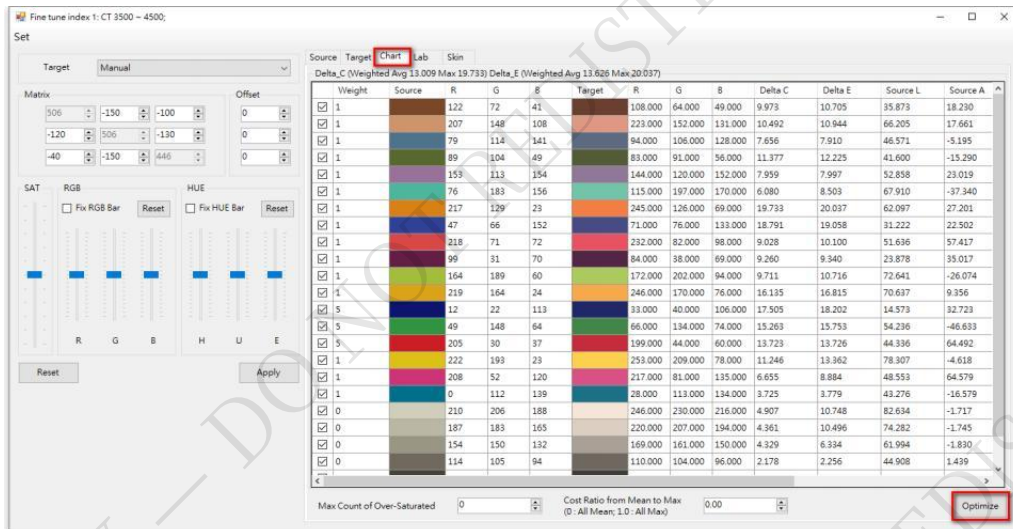


Fig 6.154

- Click “Apply” to re-simulation and check simulation result. Analysis for this simulation or fine tune manually if needed.

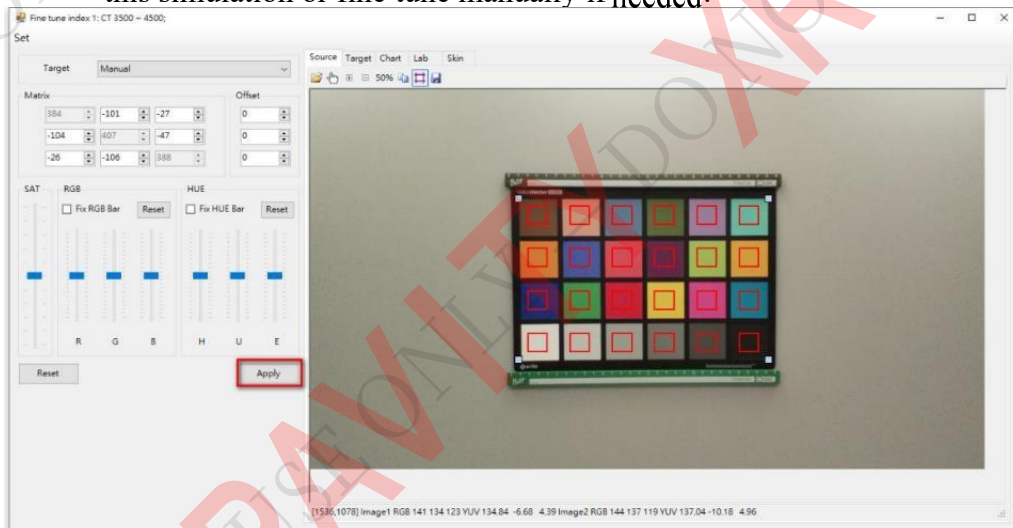


Fig 6.155

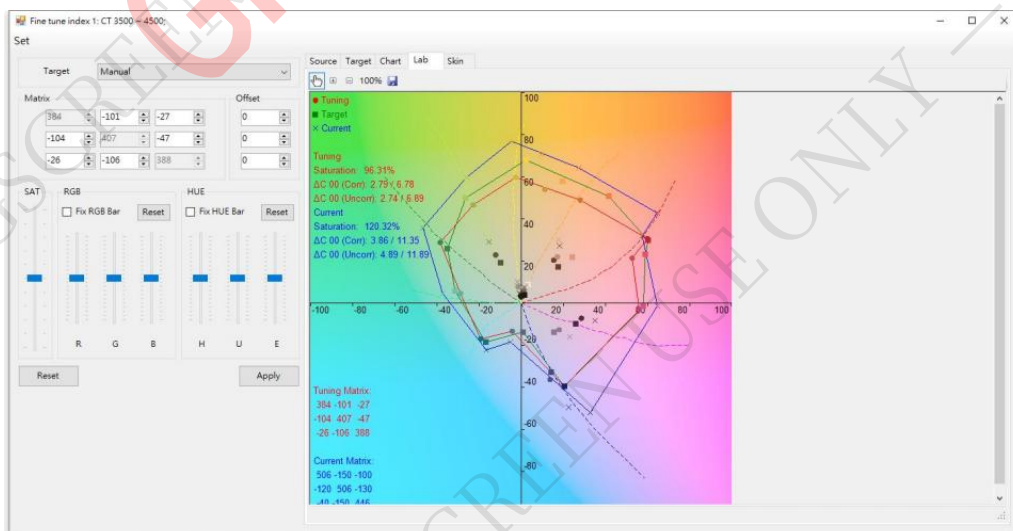


Fig 6.156

7. Click “Set”, the parameters will be updated to CCM main page.

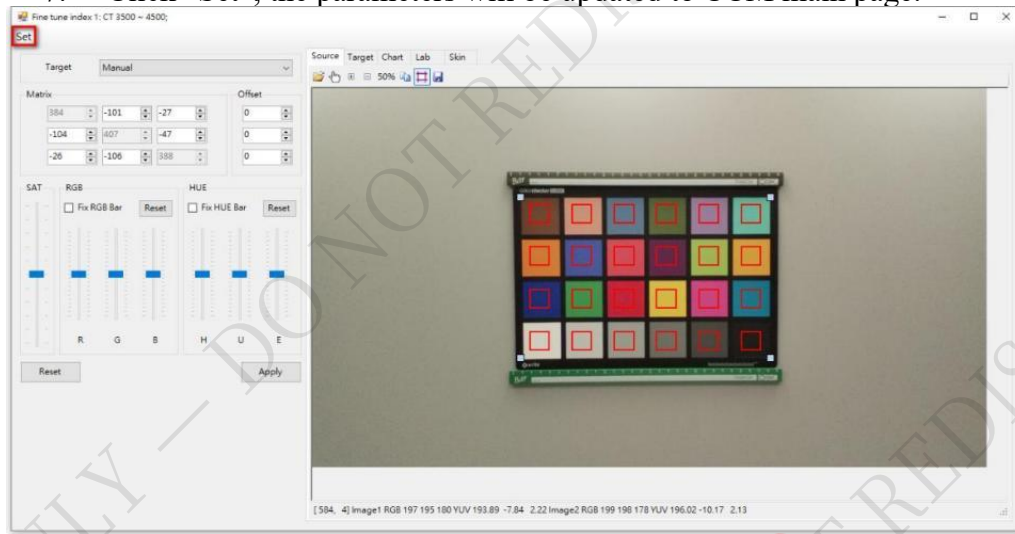


Fig 6.157

For the tuning of the three parameters: “Clip level”, “protect range” and “CCM decimal”, please refer to the article [“Improvement for color tint \(face case\)”](#).

6.11.2. BCCM

The target of blended CCM is prevent the false color be enhanced when raising the colorfulness at CCM module, and reduce CCM colorfulness more when the color closer the neutral color.

This function is an extension of CCM for reducing false color. By interpolating CCM coefficient in different chroma region, we can limit color noise be not enhanced.

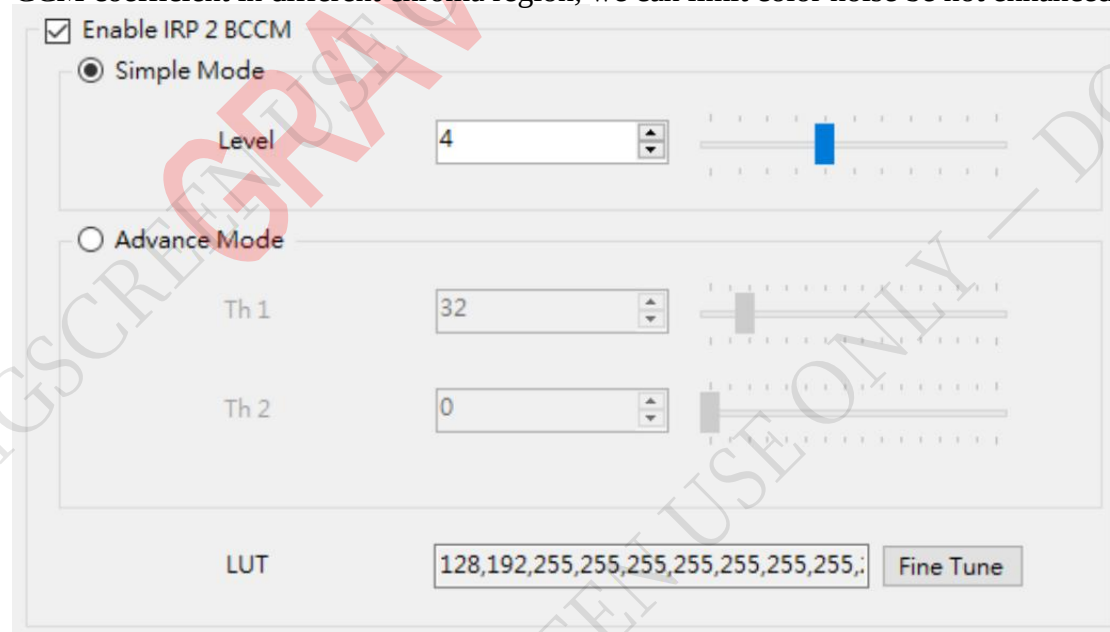


Fig 6.158

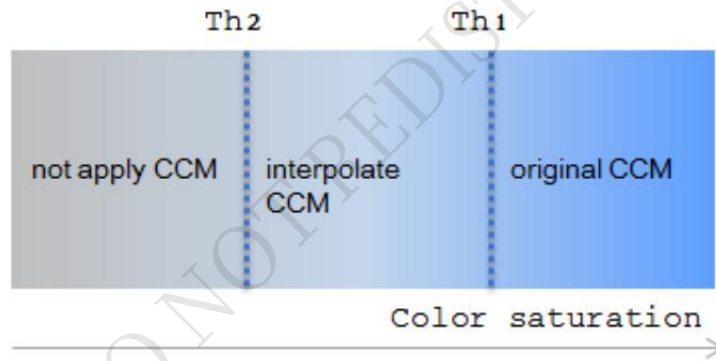


Fig 6.159 Th 1&2 definition



Fig 6.160BCCM example

Although BCCM limit color in noise region but also effect color texture.

6.11.3. Gamma

The human eyes perceive light in a different way than the cameras. The Gamma function provides a nonlinear mapping to fit the human eyes.

(Note that there is only one set of Gamma throughout different gain triggered parameters)

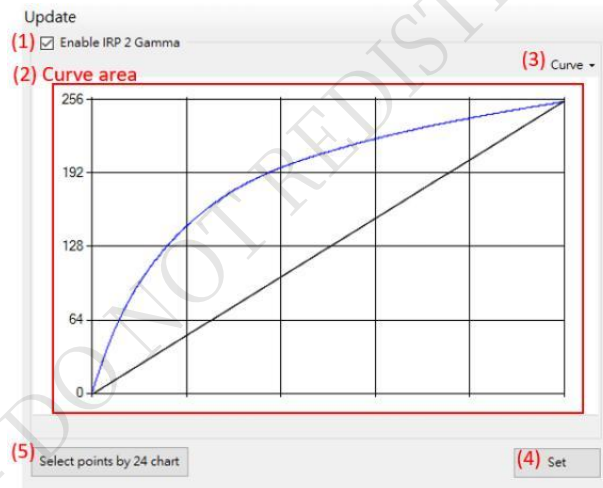


Fig 6.161

■ Enable/Disable Gamma

Disable Gamma will get a linear gamma curve



Fig 6.162

- **Curve area:** Allows user to add control point on the curve and generate smooth non-linear curve. Left click by mouse to add point, and press the “Del” by keyboard to delete the point. When a point is selected, the points show green color, the users can drag around or use the keyboard (up, down, left, right) to generate new curve. It is recommended to use tool default curve.

■ Curve function

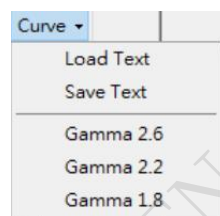


Fig 6.163

Save text: Save the current curve into a text file.

Load text: Load a text file curve. The user can save a text and see the format before load a text.

- **Gamma 2.6/2.2/1.8:** Default gamma curve for reference
 - **Set:** Apply gamma to simulation result
 - **Select points by 24 color chart**
- The user can create a gamma curve by a target 24 color chart image.

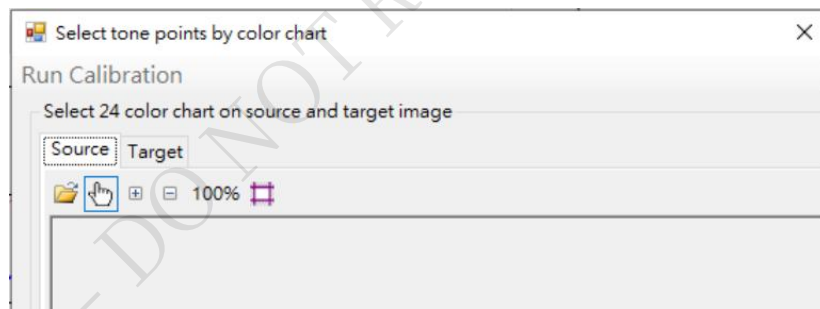
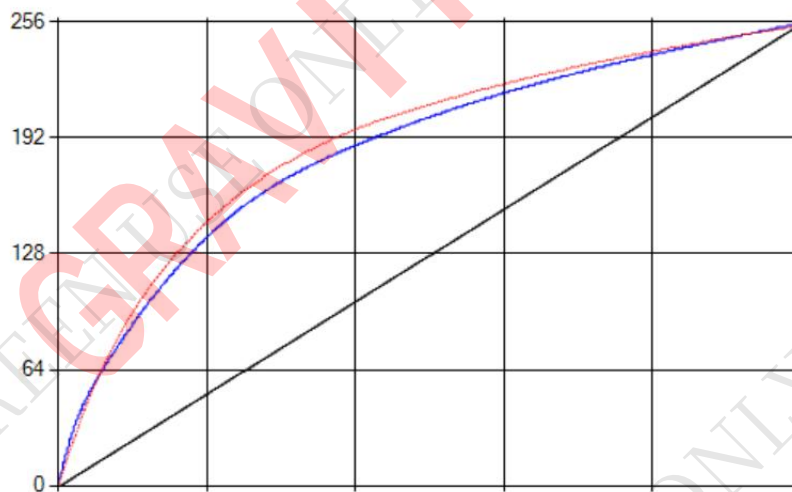


Fig 6.164

Follow steps as below,

- i. Load a raw image for source
- ii. Select 24 patches on source image with
- iii. Load jpeg or bmp image for target
- iv. Select 24 patches on target image with this button: 
- v. Start "Run Calibration"
- vi. The new gamma curve will be displayed in the curve area



Blue: new curve, Red: original curve

6.11.4. RGB2YUV

Convert RGB to YUV, currently set BT601 as default setting.

6.11.5. IRP 2 CS

6.11.5.1. Chroma suppression

The chroma suppression enforce the false color close neutral to become more close neutral color actually in CbCr domain. At LUT mode, the color distance apart from the neutral be suppressed could be tuned by a threshold, and the suppress strength could be tuned by a LUT. Besides, if Y is smaller than a pre-defined threshold, it will do nothing at chroma suppression.

At Linear mode, the Cb/Cr values would be suppressed with linear equation by the different cutoff threshold, slope and offset setting.

【LUT mode】

- Decrease saturation by LUT
- ☐ $(CbCr < Th \ \&\& \ Y > \text{Luminance Th}) \rightarrow \text{mapping CbCr by LUT}$
- ☐ In dark area, CbCr is small and easily to be suppressed. If only want to reduce brightness area, you can set the Luminance Th.

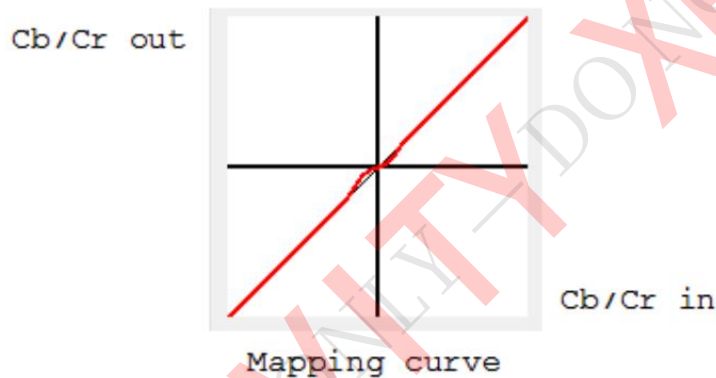


Fig 6.165 Chroma Suppress LUT mode

【Linear

- Mapping CbCr value by linear equation
- Like LUT mode, linear equation mode also could be represented to a transfer function. However, we suggest use LUT mode first, because it is more accurate for controlling.

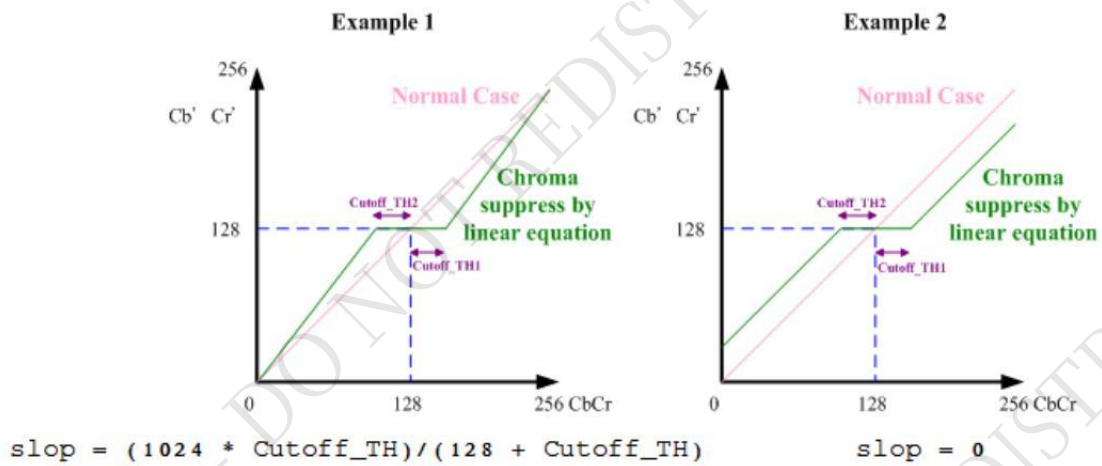


Fig 6.166 Chroma Suppress linear equation mode

6.11.5.2. Cb Cr 444 Blur

CbCr 444 Blur reduces false color by applying a 5 x 5 mask on Cb and Cr channels.

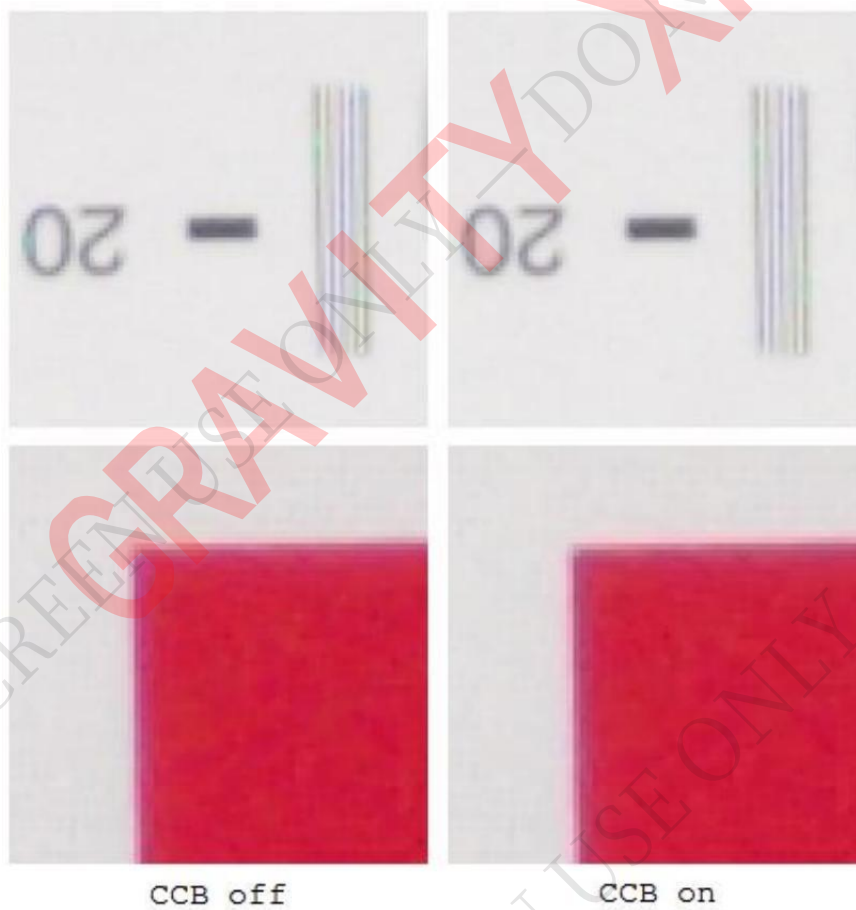


Fig 6.167 Chroma Suppress CCB example

6.11.5.3. Y Guided Chroma Suppression

Y guided chroma suppression enforce the false color close neutral in low or high

luminance area to become more close neutral color actually in CbCr domain.

YCS is an advance version of CS. It reference Y intensity for guiding suppressed level. There are 2 weighting curve of this function, one is defined by Y intensity, the other is by chroma difference. Combine these 2 weightings to decide final suppress level.

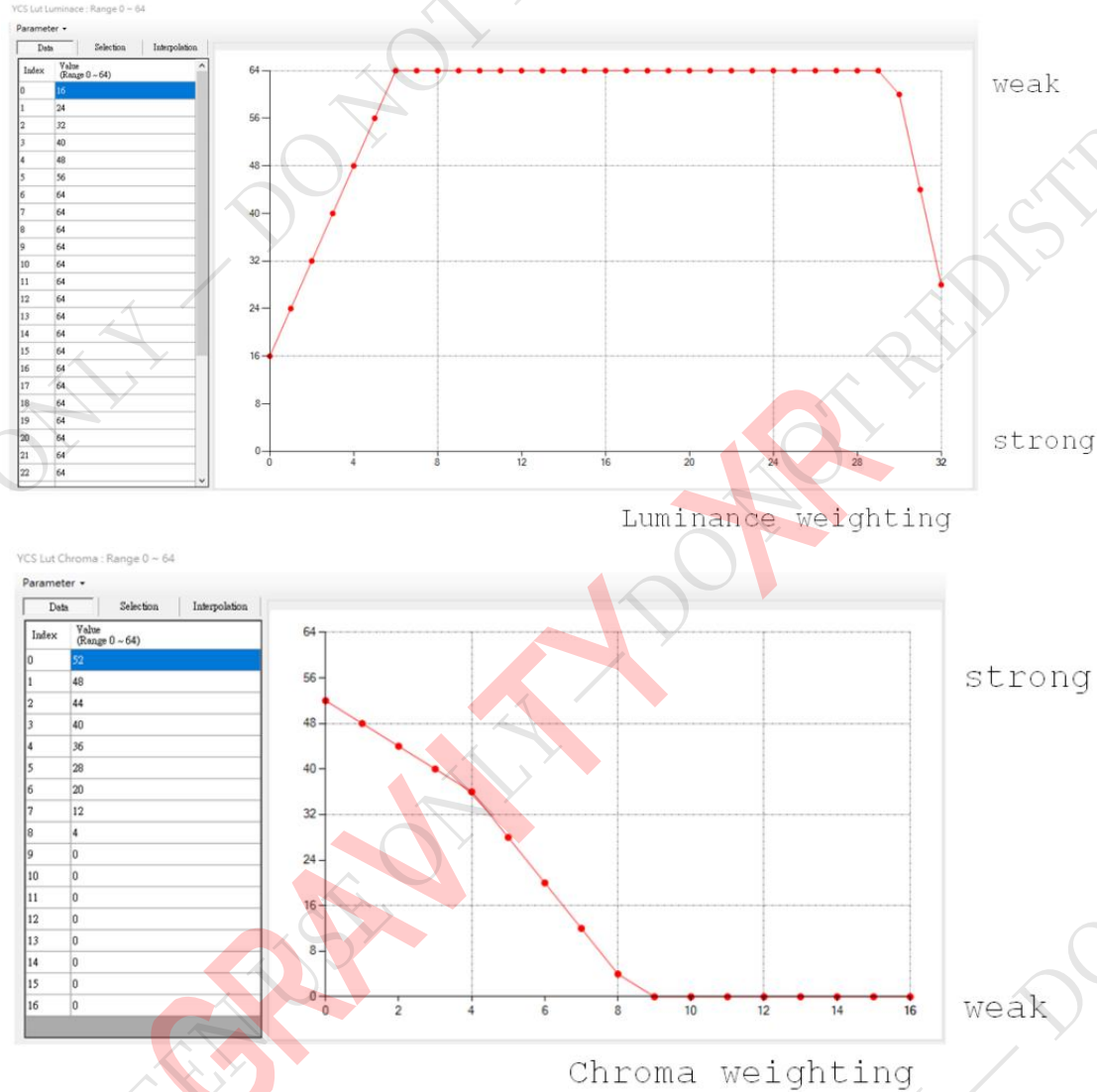


Fig 6.168 Chroma Suppress YCS



Suppress color not refer to Y



Suppress more color on dark and bright region

Fig 6.169 Chroma Suppress YCS example

6.12. CFR Tuning Guide

CFR eliminates color fringes in images caused by lens chromatic aberration or other algorithms. This effect is particularly noticeable along high-contrast edges in high dynamic range scenes. An indoor high dynamic range scene is used below as an example to illustrate the debugging strategy.



Fig 6.170 Classic Scenarios of CFR

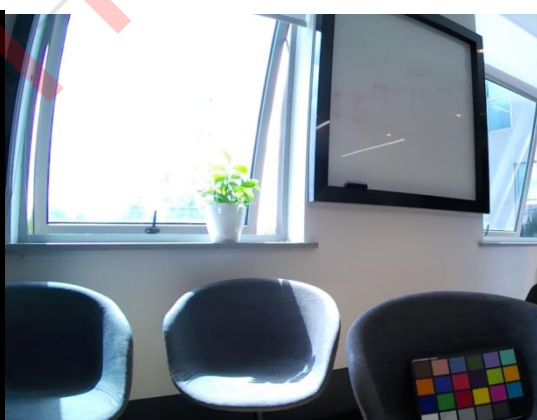


Fig 6.171 CFR Focus Area

- Calibration steps: Chroma → Saturation → Brightness → Contrast → Radial → Intensity

1. Hue Mask Debugging

p_hue1thleft	240	p_hue2thleft	0	p_hue3thleft	0
p_hue1thright	320	p_hue2thright	400	p_hue3thright	400
p_hue1slopeleft	8	p_hue2slopeleft	0	p_hue3slopeleft	0
p_hue1sloperight	7	p_hue2sloperight	0	p_hue3sloperight	0

Fig 6.172 Hue Mask Debugging

Select the chromaticity range to be removed. Three ranges are available for configuration. Each range corresponds to four debugging parameters: **thleft**, **thright**, **slopeleft**, and **sloperight**, which represent the left hue threshold, the maximum hue value of the right mask, the rising slope of the left mask, and the falling slope of the right mask, respectively, as shown in the following figure:

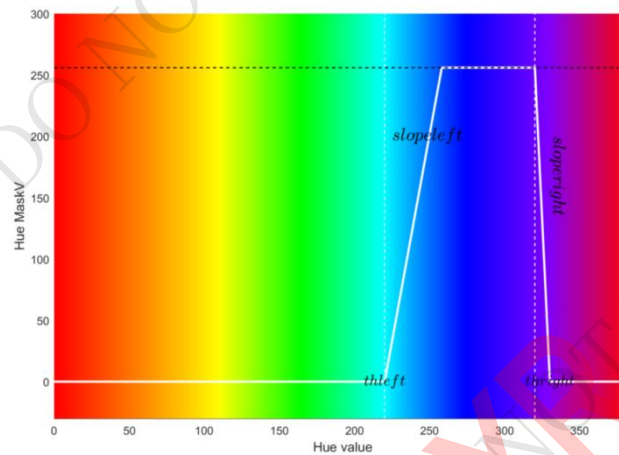


Fig 6.173 Schematic diagram of chroma interval removal parameters

2. Saturation Mask Debugging

p_sauth	50
p_saudelta	5

Fig 6.174 Saturation Mask Debugging

There are two debugging parameters, corresponding to the left saturation threshold and the width of the transition interval from the minimum to the maximum mask value.

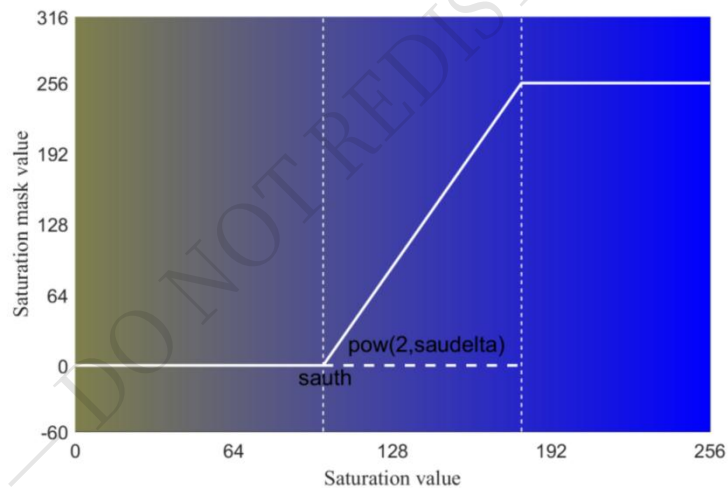


Fig 6.175 Schematic diagram of saturation mask adjustment parameters

3. Brightness Value Mask Debugging

p_yth	120
p_ydelta	5

Fig 6.176 Brightness Value Mask Debugging

There are two debugging parameters, corresponding to the left brightness threshold and the width of the transition interval from the minimum to the maximum mask value.

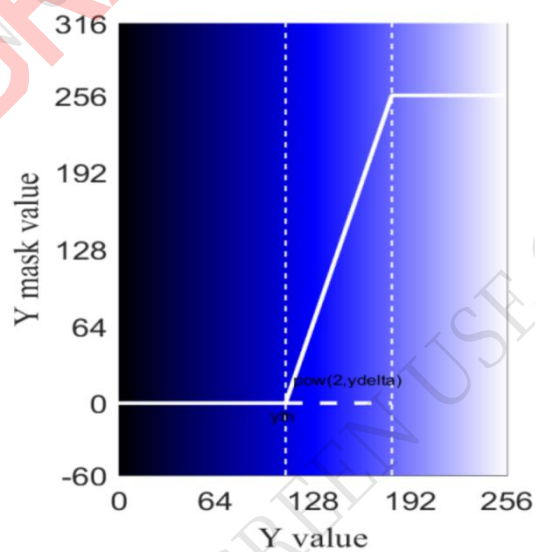


Fig 6.177 Schematic diagram of luminance (Y channel) mask adjustment parameters

4. Contrast Value Mask Debugging

p_consth	100
p_constdelta	6

Fig 6.178 Contrast Value Mask Debugging

There are two debugging parameters, corresponding to the left contrast threshold and the width of the transition interval from the minimum to the maximum mask value.

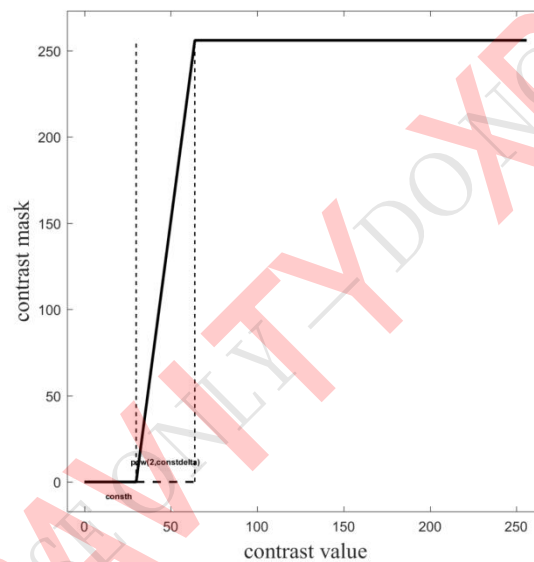


Fig 6.179 Schematic diagram of contrast mask adjustment parameters

5. Radial Direction Strength Adjustment

p_radth1	1
p_radth2	10

Fig 6.180 Radial Direction Strength Adjustment

Let R be the distance from the current pixel to the image center. The formula for the radial direction intensity mask is as follows:

$$rad_mask = \begin{cases} 0 & R \leq radth1 \\ \frac{R \cdot R - 2^{12 \cdot radth1}}{2^{4 \cdot (radth2 - radth1)}} & R > radth1 \text{ and } R < radth2 \\ 256 & R \geq radth2 \end{cases} \quad (1)$$

6. Total Removal Strength Adjustment

A larger value indicates stronger removal. The default value of 16 corresponds to normal removal strength.



Fig 6.181 Total Removal Strength Adjustment

Comparison of Results Before and After Applying CFR with Different Strengths.

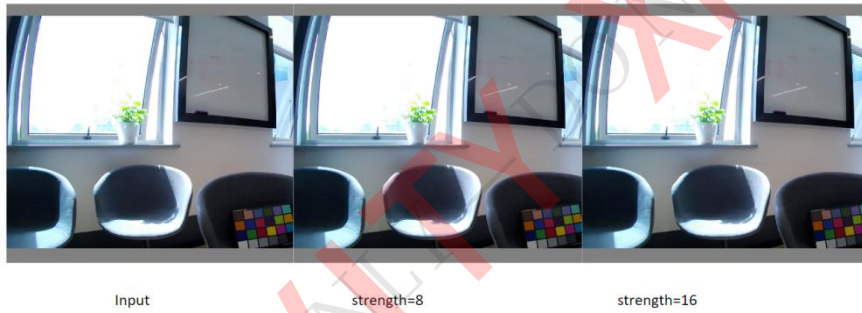


Fig 6.182 Comparison of CFR processing results with different strength values

6.13. Post Processing

Post processing engine combines two operations: 1.AdjustYcontrast, 2.Adjust Cb Cr by 2DLUT method. Y contrast operation can let user tuneY intensity into a new desired output intensity. 2D LUT is a kind of color transformation process, this algorithm uses Cb/Cr as input and combine with Y channel for reference. User can define 32 x 32 Cb/Cr LUT to find desired remapping color. At last, an input Cb/Cr can map to a new desired output Cb/Cr result

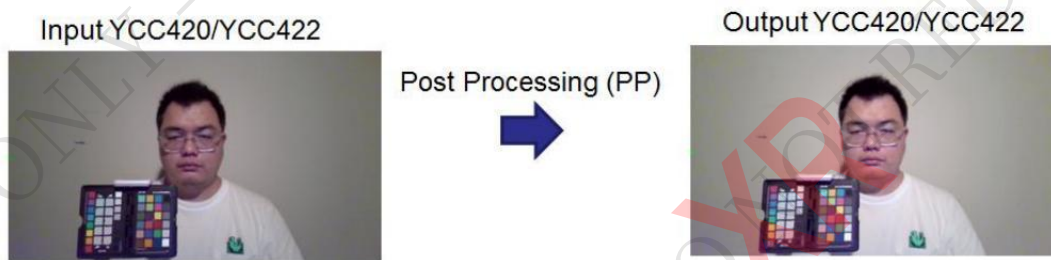


Fig 6.183 Post Processing result

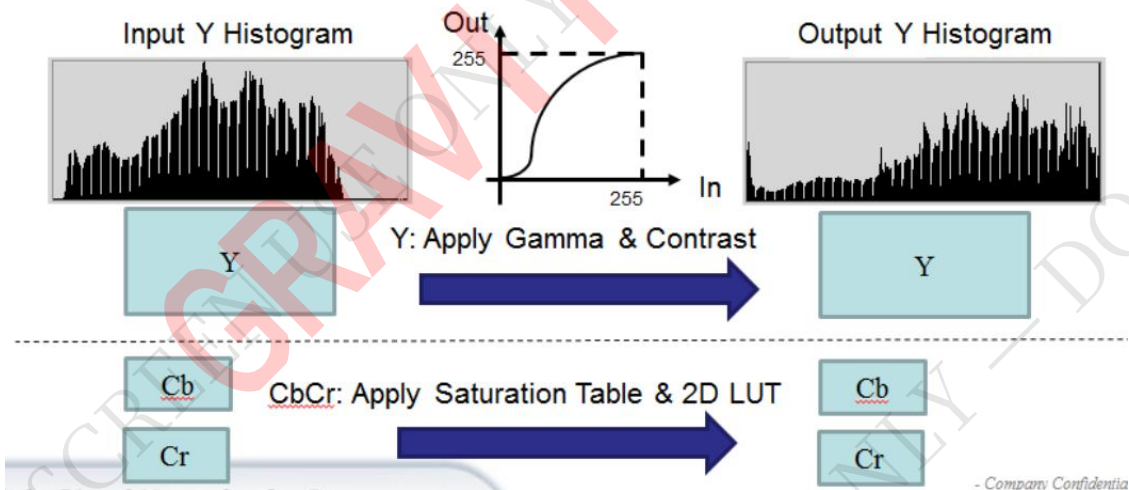


Fig 6.184 Post Processing flow

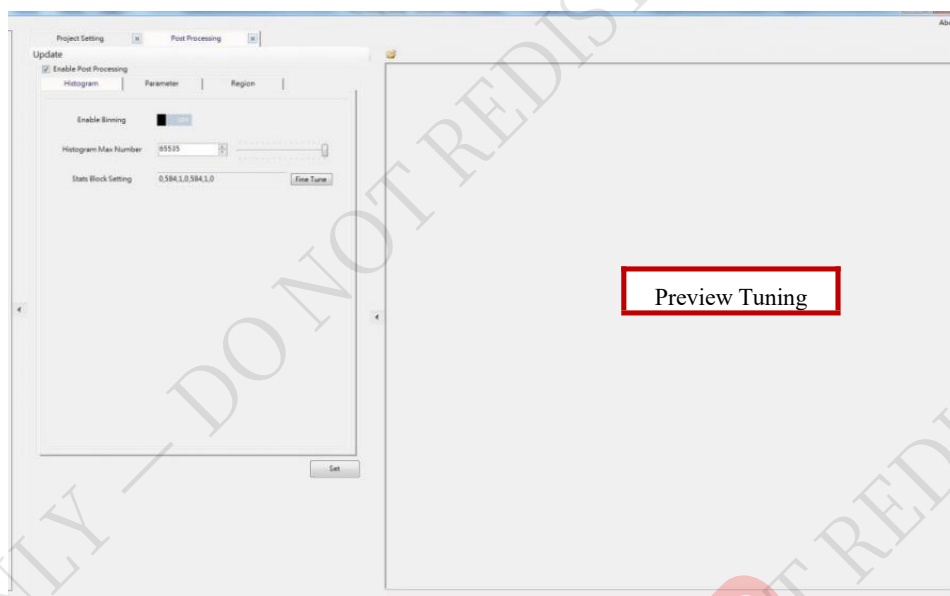


Fig 6.185 Basic GUI of

6.13.1. Tuning Process:

- Set trigger point of Color Temperature
- Select candidate region to tune color

6.13.2. Basic flow control

Please refer to fig. . page

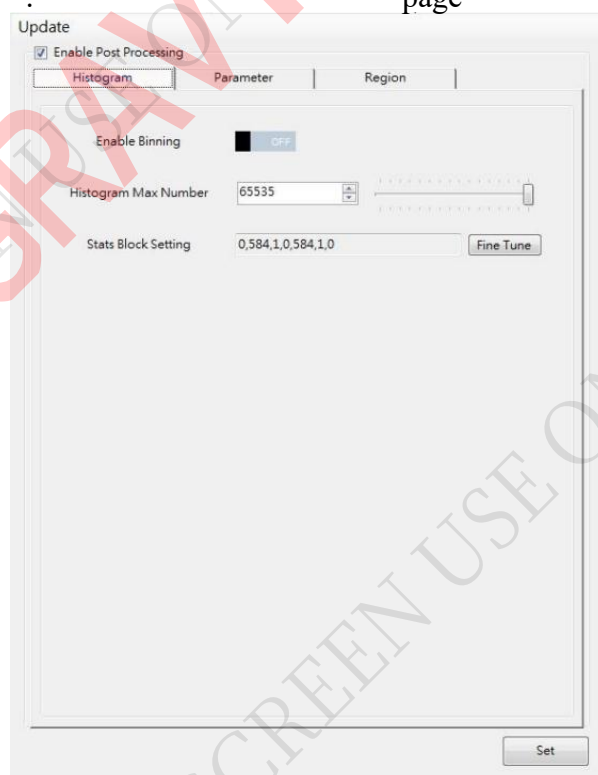


Fig 6. 186 Basic control parameter (Histogram)

- Enable Binning
 - ☐ Calculate binning histogram. For large image, the histogram statistic will overflow, so the image must scale down through the binning method.
- Histogram Max Binning
 - ☐ Max number of histogram's counts. Range from [0~65535]
- Stats Block setting
 - ☐ Set histogram statistic region.

- Enable Adaptive Curve
 - ☐ Control shading weight's influence on 2D LUT.
- Trigger CT
 - ☐ Different color temperature's trigger point.
 - ☐ CT Low/ CT Mid/ CT High

6.13.3. Select Region and Tune color

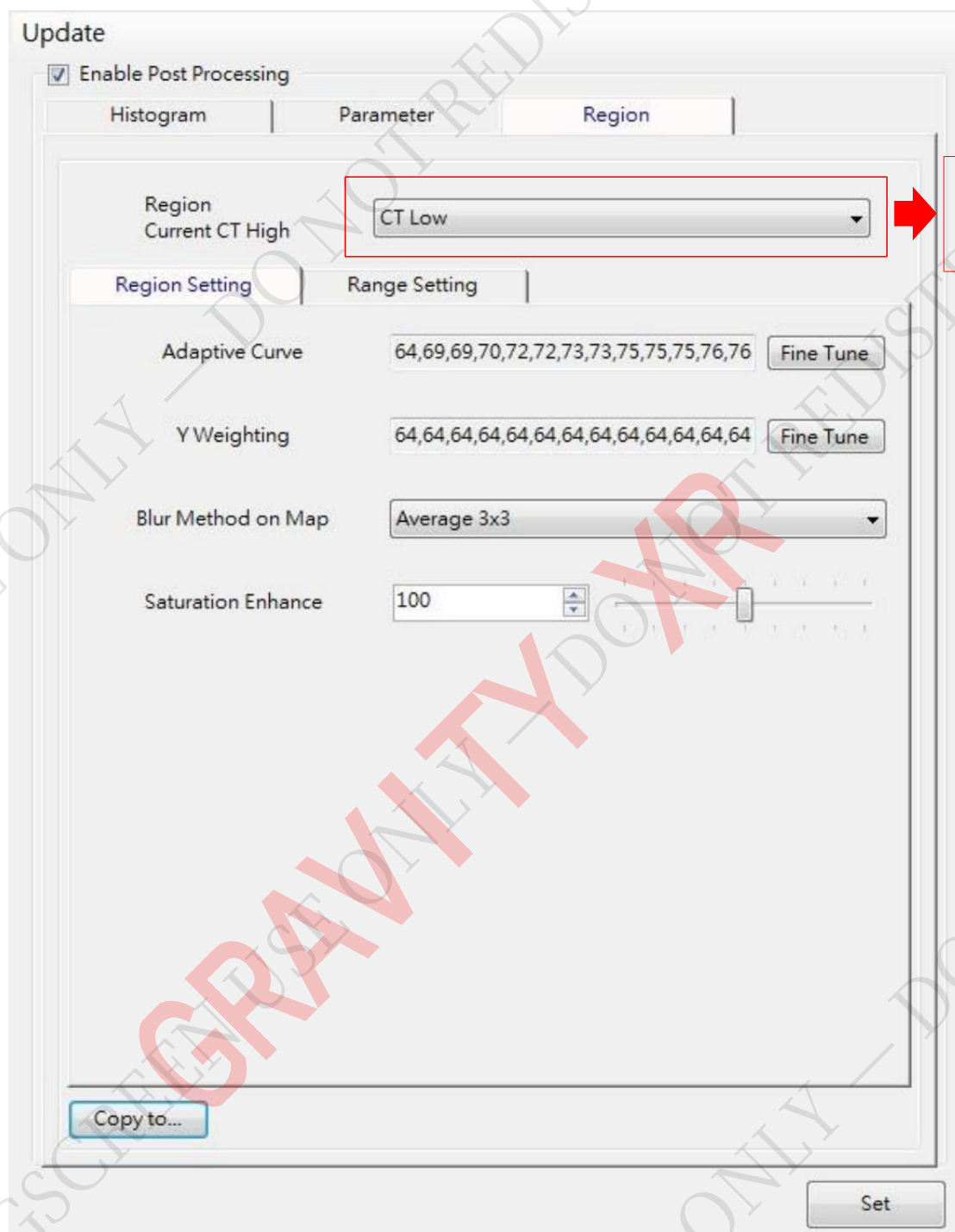


Fig 6.187 Region tuning parameter

■ Adaptive Curve

- ☐ Apply different weighting on center corner region
- ☐ Weight < 64 => Reduce 2D LUT effect.
- ☐ Weight > 64 => Enhance 2D LUT effect.

It separate 33 nodes for user to tune different weighting on center/corner

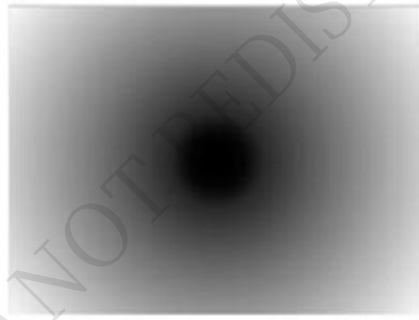


Fig 6.188 Shading(Adaptive curve)

- Y Weighting
 - ☐ Base on Y(Luminance) intensity to reduce/enhance 2D LUT effect.
 - ☐ Node : 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, 256
 - ☐ Weight < 64 => Reduce 2D LUT effect.
 - ☐ Weight > 64 => Enhance 2D LUT effect.
- Blur Method on Map
 - ☐ Apply on 2D LUT to smooth every tuning color range (Range 0~3).
 - ☐ Average 3x3
 - ☐ Gaussian 3x3
 - ☐ Gaussian 5x5
- Saturation Enhance
 - ☐ Globally enhance saturation
 - ☐ Range: [0 ~ 200]. Base is 100.

6.13.4.

Tuning tool converts original hardware Y tone table and 32x32 Cb/Cr LUT into HSV domain for intuitive tuning.

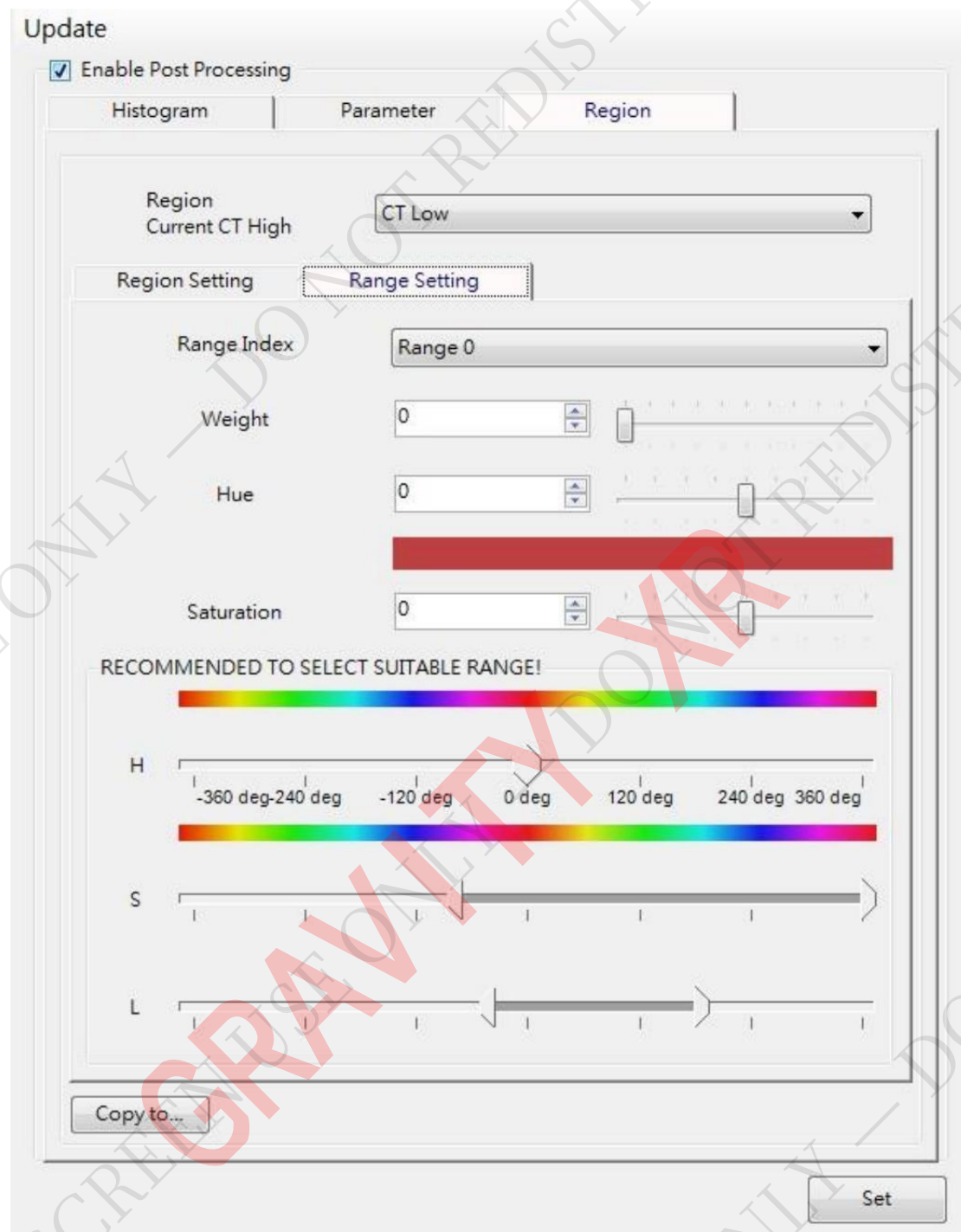


Fig 6.189 Color Range Setting

- Range Index
 - ☐ Support 4 range tuning combination
- Weight
 - ☐ Each range(Range 0~ 4) map's weight
 - ☐ Range : [0~100]
- Hue
 - ☐ 0 : Red
 - ☐ 120 : Green
 - ☐ 240 : Blue
 - ☐ Limitation : +/- 10 degree

- ☐ Range: [-5000~5000]. Base is 100
- Saturation
 - ☐ Range: [-5000~5000]. Base is 100
 - ☐ Limitation : +/- 10%
- Lightness (L)
 - ☐ Range : [0~100]

6.13.5. Helper UI

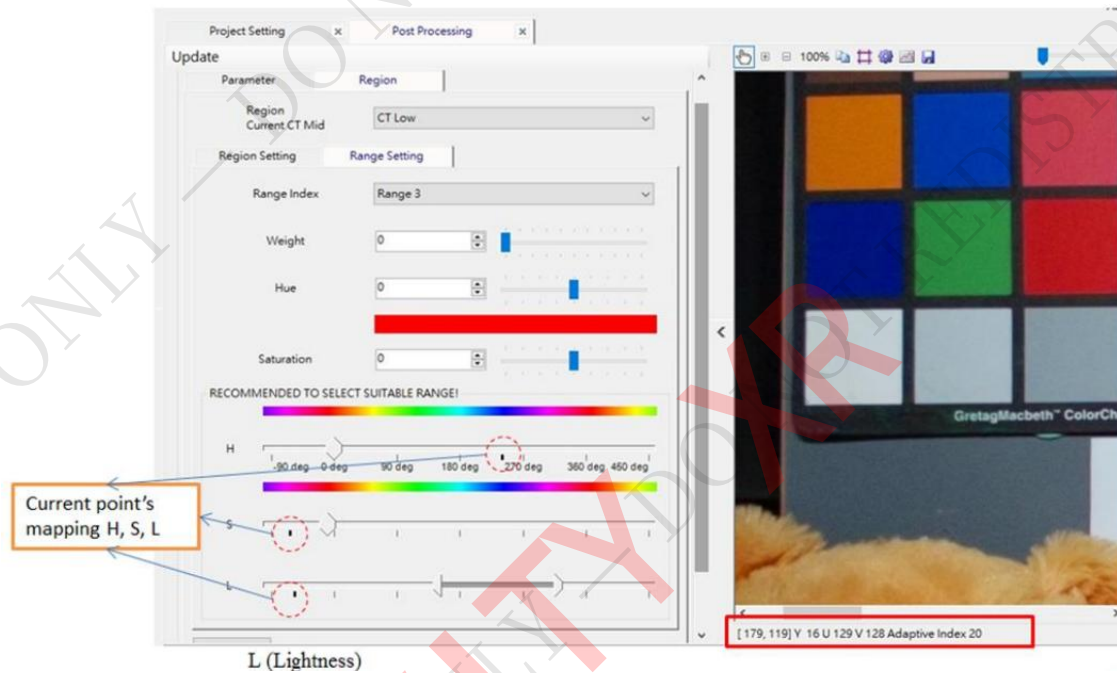


Fig 6.190 Helper UI

6.13.6. Tuning

1. Tuning one range (ex: Range1)
2. Range's weight (ex: apply 60% Range1's map)
3. Mouse move to red rectangle region, it will show YUV information and its
4. Stretchbar in mapping H/S/L (Must contain current H/S/L).
5. Tuning Hue/ Saturation depend on user's preference.

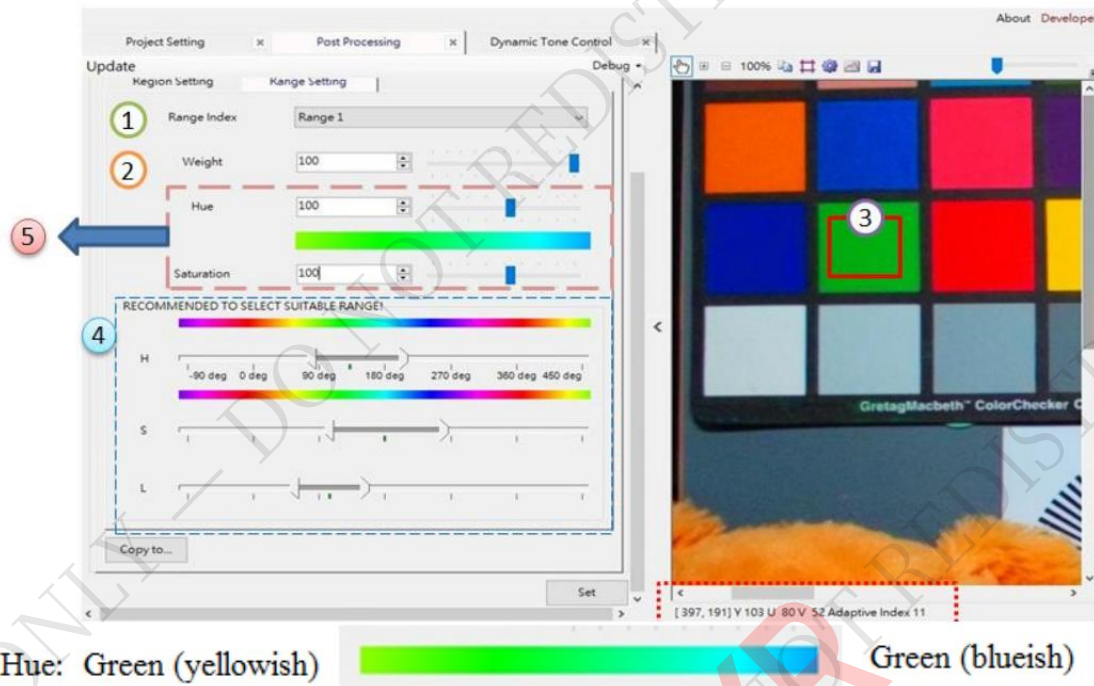


Fig 6.191 Post processing tuning

6.14. Sharpness

Sharpness aims to enhance edge contrast of input image, it doesn't really add resolution. Sharpness only works on Y channel. If user prefers high edge contrast, it can be tuned according to preference, example is shown below:



Fig 6.192 Sharpness overall goal

6.14.1. Tuning Process

■ Tuning order:

Filter→Intensity→Feature→Laplacian→Luminance→Map

It is suggested to tune by sub-page order.

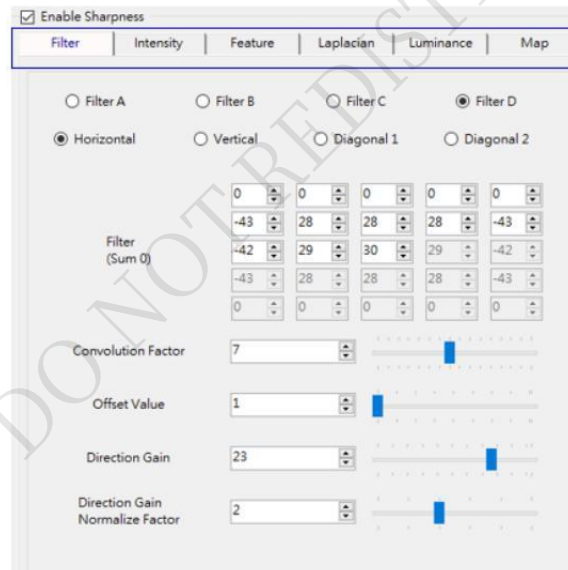


Fig 6.193 Sharpness overall tuning sub-

6.14.2. Tunings for “Filter” sub-page

- 5x5 filter is used to get filter response
 - There are four filter groups: FilterA, FilterB, FilterC, FilterD
 - For each group, four directional masks are used: Horizontal, Vertical, Diagonal1, Diagonal2 (+45 degree)
 - User can set mask coefficients for Horizontal & Diagonal1; HW will get Vertical & Diagonal2 mask coefficients automatically, the configurable coefficients are a0~a12 & b0~b12, tuning tool can let user set desired values, example is shown below:

a0	a1	a2	a3	a4
a5	a6	a7	a8	a9
a10	a11	a12	a11	a10
a9	a8	a7	a6	a5
a4	a3	a2	a1	a0

Horizontal:
a0~a12 can be set by user
They are registers in HW

a4	a9	a10	a5	a0
a3	a8	a11	a6	a1
a2	a7	a12	a7	a2
a1	a6	a11	a8	a3
a0	a5	a10	a9	a4

Vertical:
It will reference a0~a12 automatically
to put in above figure order by HW

b0	b1	b2	b3	b4
b5	b6	b7	b8	b9
b10	b11	b12	b11	b10
b9	b8	b7	b6	b5
b4	b3	b2	b1	b0

Diagonal1:
b0~b12 can be set by user
They are registers in HW

b4	b9	b10	b5	b0
b3	b8	b11	b6	b1
b2	b7	b12	b7	b2
b1	b6	b11	b8	b3
b0	b5	b10	b9	b4

Diagonal2:
It will reference b0~b12 automatically
to put in above figure order by HW

Fig 6.194 Sharpness: Set mask coefficients

- Filter coefficients setting rule:
 1. Let all mask coefficients summation value for a 5x5 filter be zero
EX:

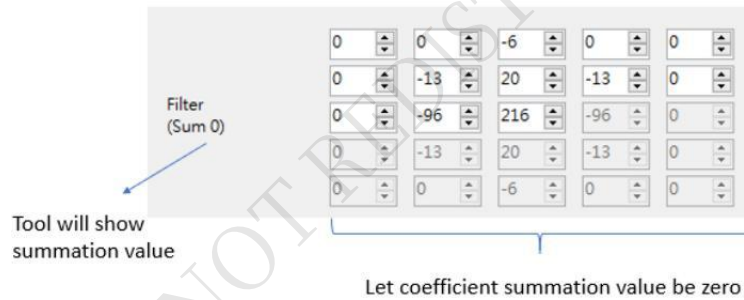


Fig 6.196 Observe mask coefficients

2. Data range limitation:

- a12 (range: -255~+255 ; 9bit)
- b12 (range: -255~+255 ; 9bit)
- a6, a7, a8, a11 (range: -127~+127 ; 8bit)
- b6, b7, b8, b11 (range: -127~+127 ; 8bit)
- a0, a1, a2, a3, a4, a5, a9, a10:(range: -63~+63:
- b0, b1, b2, b3, b4, b5, b9, b10:(range: -63~+63: 7bit)

3. Tuning suggestion:

- High pass filter coefficients need to be designed carefully, not suitable for random key in numbers. If user doesn't have specific preference, it is suggested to use tool's default coefficients.

■ Other filter control values:

- ☐ Convolution Factor: (Data range: 0~15)
- ☐ after 5x5 convolution, how manybits for shifting to get response; for example:

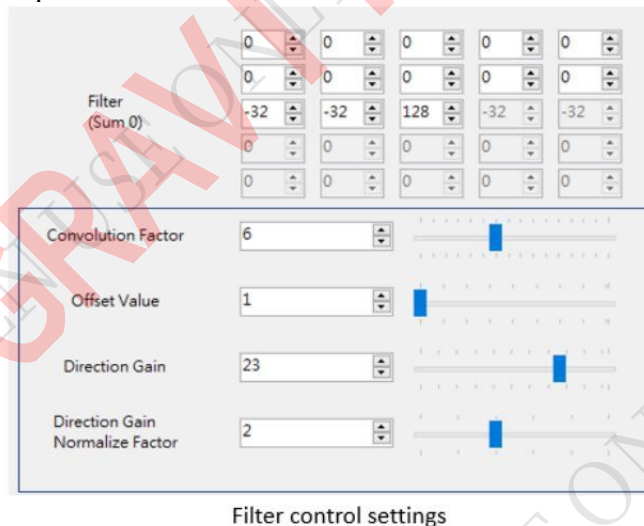


Fig 6.197 Filter control related values

After convolution, needing to do bit shift to limit the data range, convolution factor means the bit shift amount. In often, user can estimate the summation value of positive coefficients, in above example, summation value is 128, convolution factor is 6, it means that convolution result will do bit shift of 6 (equal to divide by 64). This step aims to shrink data range before further signal estimation steps. For suggestion, if positive coefficient summation value is around 64, let

convolution factor be 5; if positive coefficient summation value is around 128, let convolution factor be 6; if positive coefficient summation value is around 256, let convolution factor be 7; these settings are often used for general tunings, the signal range will not be enlarged too much or shrunk too much. User can reference this suggestion.

- Offset value: (Data range: 0~ 511)
 - It will do data offset after convolution, when convolution factor is too low, then data range is enlarged after convolution, the offset value needs to set as high value, that's why this value's data range is not so limited. In often usage, if user use tool's default mask coefficient & convolution factor, because that data range is controlled, then offset value is suggested to be tuned in range of 0~3. Higher offset value help remove too small signal after convolution, it can help reduce smooth area noise. Example is shown below:



Fig 6.198 simulation example of different filter offset value

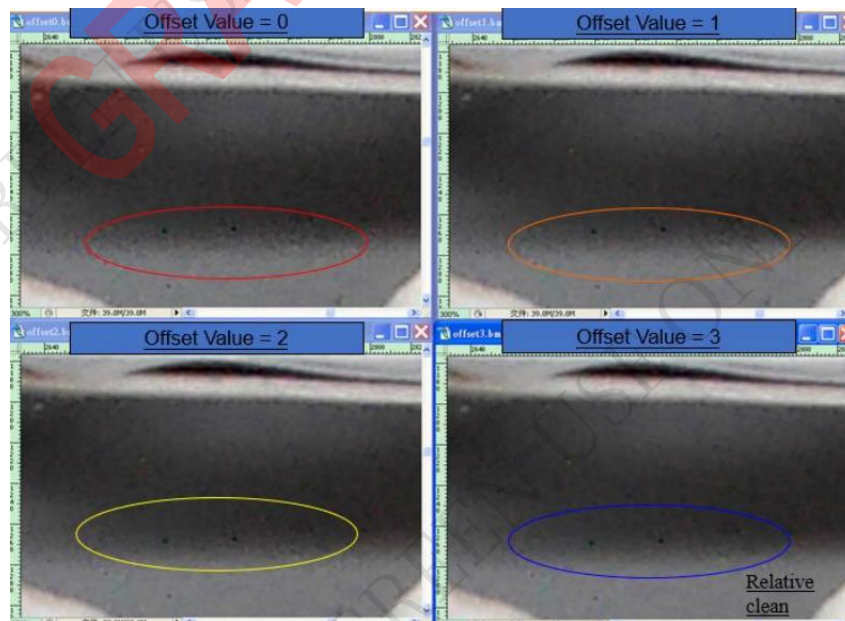


Fig 6.199 simulation example of different filter offset value

- Direction gain: (Data range: 0~ 31)
 - Each direction can set different direction gain. Higher gain value can let response be increased. For suggestion, letting each direction using same gain value will be preferred. If user prefers to enhance specific direction's filter response, then let each direction have obvious different gain value. For example: only let horizontal or vertical direction has gain value, other direction gain value set as zero, the comparison is shown below:

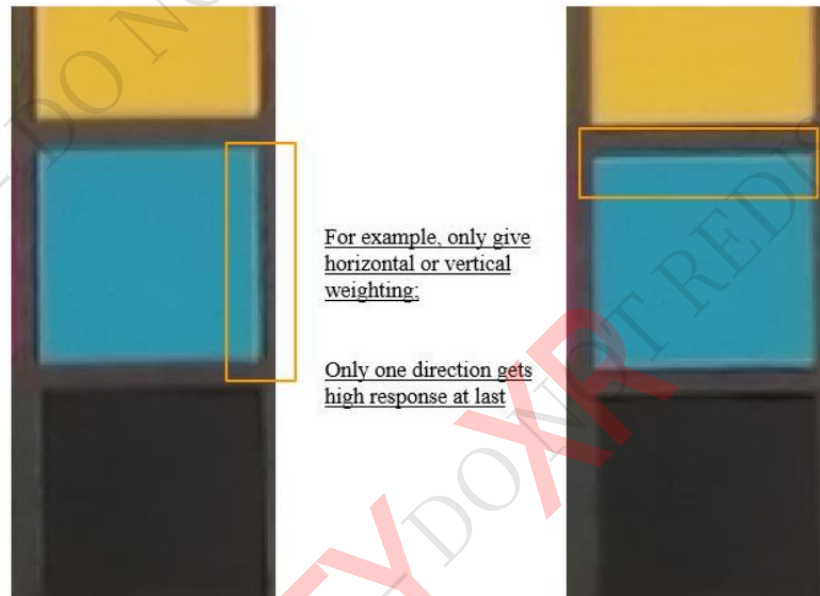


Fig 6.200 simulation example of different filter directional gain

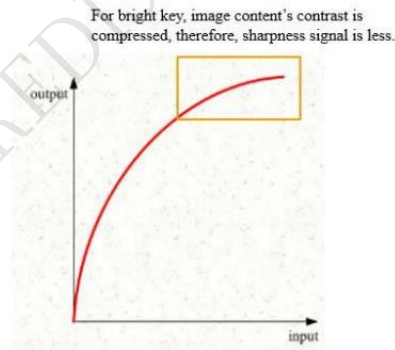
- Direction Gain Normalize Factor (Data range: 0~5)
 - After multiplying with direction gain, signal needs to do bit shift for controlling data range. This value means the bit shift amount. For example, direction gain is 16, however, direction gain normalized factor is only set as 3, it means that signal after direction gain is enlarged by 2x, because only do bit shift of 3 (equal to divide by 8).
- Overall suggestion for tuning of this sub-page
 - For this subpage, all settings are rarely tuned, user can just keep default mask coefficient, set all of direction gain as 31 (equal weighting and let it be higher), set offset value as 1, for other items, if no specific preference, can just use tool's default value.

6.14.3. Tunings for “Intensity” sub-page

Because that sharpness's input image often already be processed by specific luminance processing; (for example, high key is already highly compressed by gamma, since input has low contrast in high key, it is hard to get high response for sharpness.) In order to get required sharpness response amount, user can choose preferred LUT content to transform input data before sharpness value estimation. It is noted that input image's intensity is not really changed by this LUT at last, just use the transformed data for sharpness analysis. An example of high key input problem is shown below:



High key has lower contrast because of gamma transform.



Sharpness input already apply gamma curve

Fig 6.201 Example of high key already be compressed too much before sharpness

- Goal: Let input intensity transform into output intensity by a LUT before further sharpness estimation; (just for internal analysis usage, it will not change output image's pixel intensity by this LUT)
- Output intensity end point amount: 9
(The mapping input intensity position is shown in below figure;
(Linear interpolation will be done to get full mapping relationship))

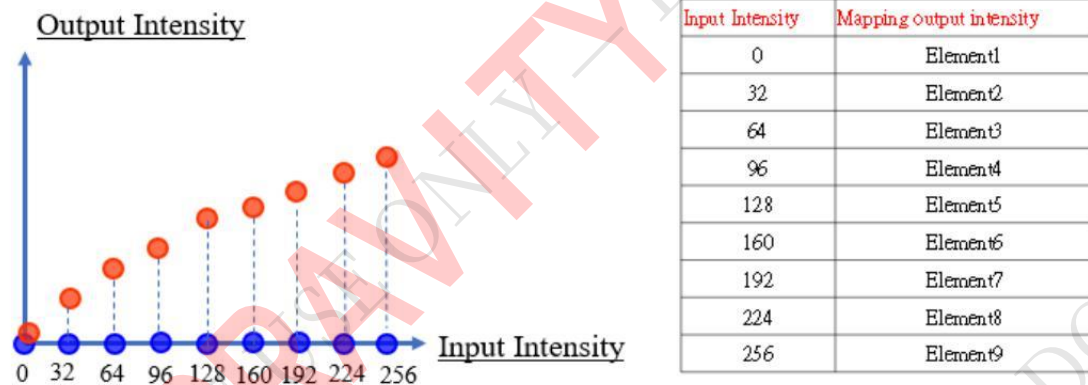


Fig 6.202 Tuning item in LUT of intensity sub-page

Two methods for tuning this LUT:

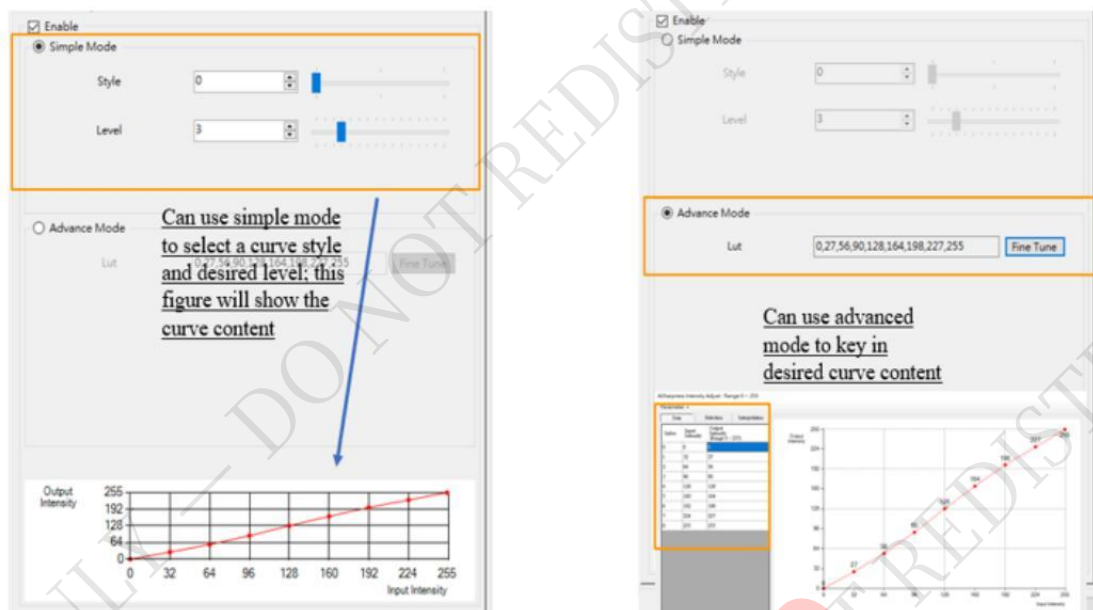


Fig 6.203 Tuning intensity LUT by simple

An example of enlarging contrast for high key is shown below:

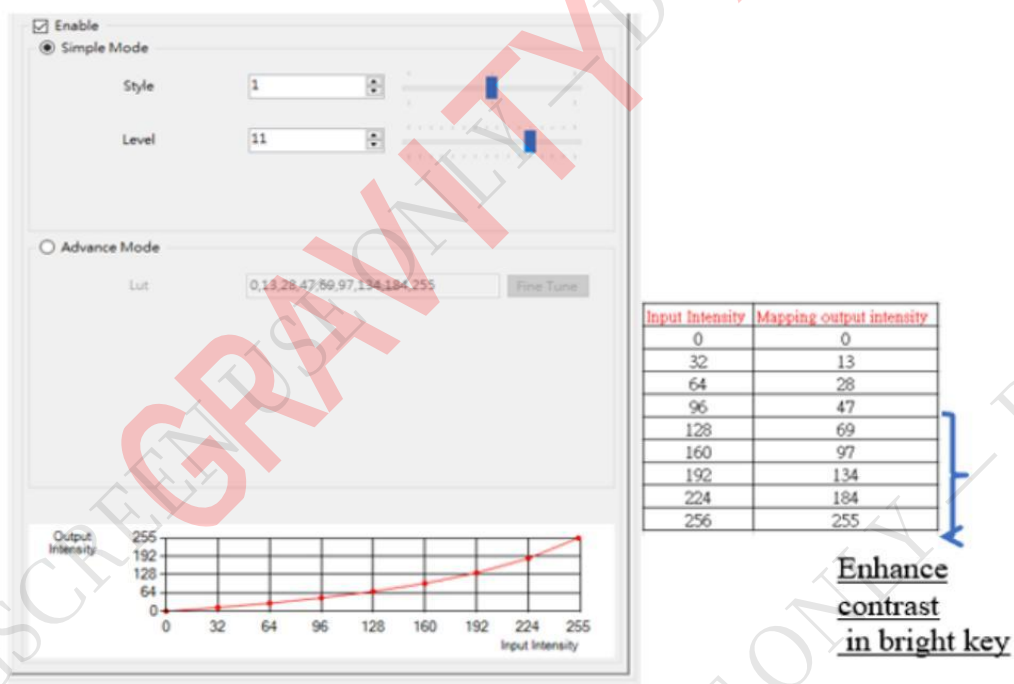


Fig 6.204 Example of high key boosting by LUT



Fig 6.205 Example of high key boosting by LUT

Suggestion for tuning:

→ If input image is high compressed for specific luminance region, user can compensate its impact by tune corresponding intensity LUT. In general, the intensity LUT is suggested to be “bypass”.

6.14.4. Tunings for “Feature” sub-page

For descriptions and given tuning examples, it is supposed that user just using tool’s default settings in “Filter” sub-page.

There are three main jobs for this sub-

1. Determines sharpness style. (thin to thick sharpness)
2. Determines sharpness feature gain for smooth area to hard edge
3. To avoid single outlier pixel over-enhancement or not

Avoiding too much sharpness enhancement for outlier pixel can be done by “Enable outlier removal”. Tuning method and example are shown below:

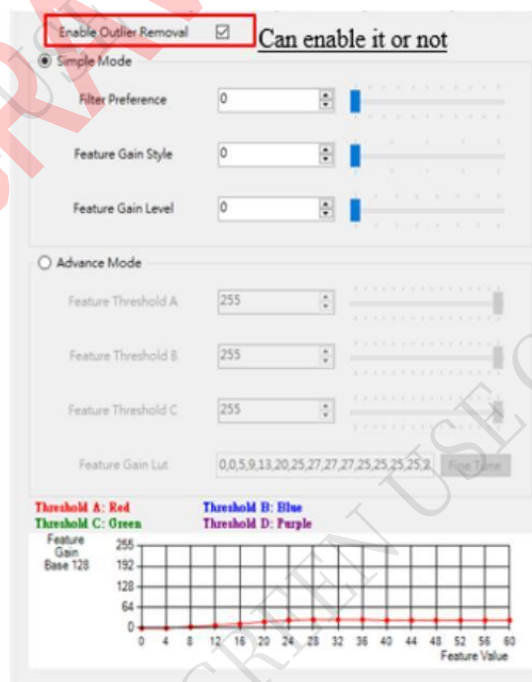


Fig 6.206 Tune outlier removal by check box

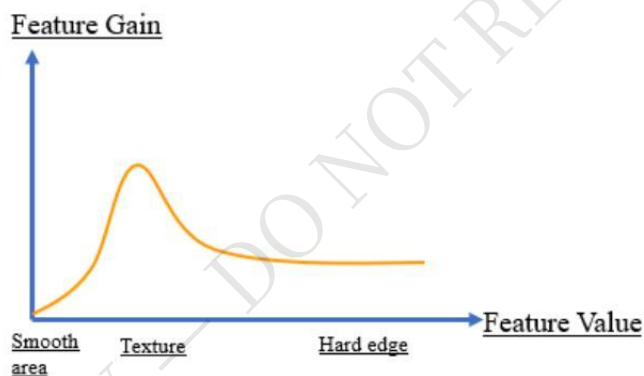
- Enable → Avoid over-enhancement for single outlier noisy pixel
- Suggestion for tuning: let it be “Enable”



Avoid single outlier noisy pixel too much sharpness enhancement

Fig 6.207 Example of outlier

- Feature gain content: 16 end points
(As below figure “Feature Value” is fixed, only can tune “Feature Gain”)



16 end points mapping method

Index	Feature Value	Feature Gain (Scale: 1~255) (Range 0 ~ 255)
0	0	0
1	4	16
2	8	32
3	12	40
4	16	50
5	20	64
6	24	64
7	28	64
8	32	64
9	36	64
10	40	64
11	44	64
12	48	64
13	52	64
14	56	64
15	60	64

Fig 6.208 For different feature value, assign

Suggested style for tuning feature

- For low feature value end points, mapping feature gain end points, assign low feature gain value
→ Prevent from smooth area noise too obvious
- For middle feature value end points, mapping feature gain end points, assign high feature gain value
→ Let texture be sharp enough
- For high feature value end points, mapping feature gain end points, assign
→ Prevent from too obvious halo at hard edge; need to enhance but not too much

Two

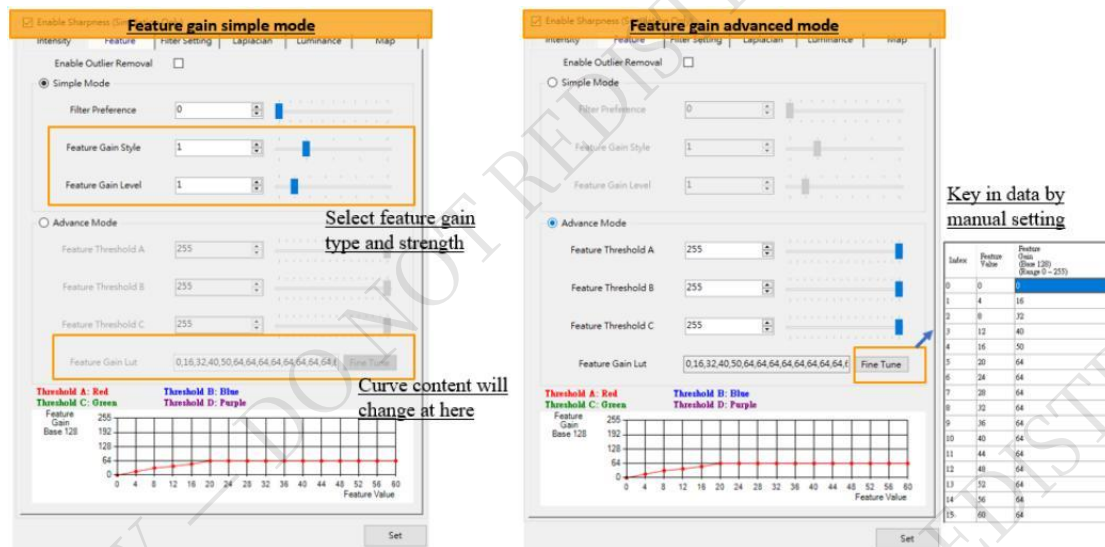


Fig 6.209 Two kinds of feature gain tuning method

Suggested feature gain tuning method:

- To control smooth area strength → often tuned by this LUT's element 1, 2
- To control texture area strength → often tuned by this LUT's element 3, 4, 5
- To control hard edge area strength → often tuned by this LUT's element 6~ end
- Suggested to let first node be zero.

Tuning examples are shown below:

Example: avoid too much noise enhancement in smooth area:

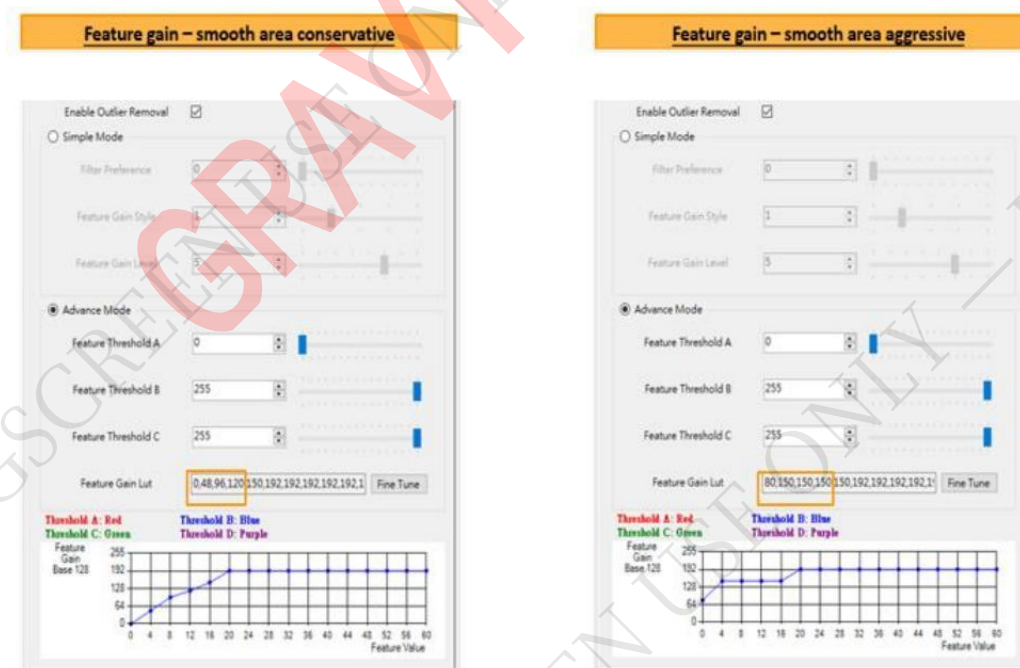


Fig 6.210 Example of feature gain tuning



Fig 6.211 Example of feature gain tuning



Fig 6.212 Comparison of different feature gain setting

Suggestion for tuning:

- low iso cases: “Filter Preference” is set as level 6~7,
- high iso cases: “Filter Preference” is set as level 9~10.
- “Filter Gain Style” is often set as: 2.
- “Feature Gain Level” is 2~3 for noisy case; 4~5 for clean image case.

- Feature threshold tuning:

Depend on input pixel's feature value, determining the selected filter group (there are four filter groups, therefore, needing three threshold values to divide them; in here, using tool's default mask setting results for example)

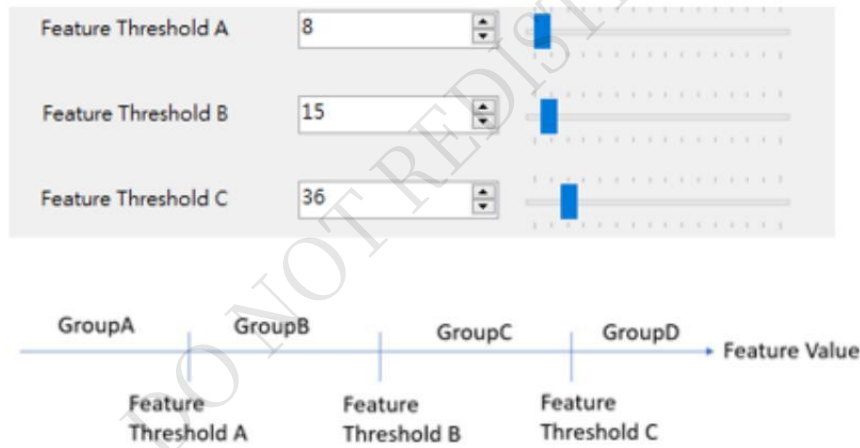


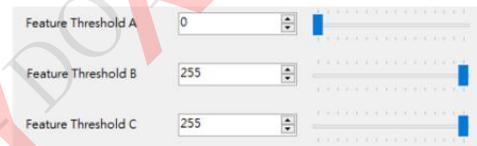
Fig 6.213 Concept of feature threshold

- ☐ GroupA: choosing thin sharpness style
- ☐ GroupB, Group C: suitable for texture area
- ☐ GroupD: choosing thick & uniform sharpness style to prevent from ugly & non-smooth hard edge

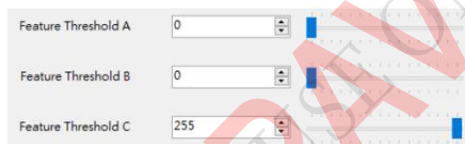
Examples of tuning results are shown below:



(mainly using GroupA)



(mainly using GroupB)



(mainly using GroupC)



(mainly using GroupD)

Fig 6.214 Feature threshold setting example

For hard edge,
smooth and
uniform
response is
better,
therefore,
rgroupC, group D
are preferred.

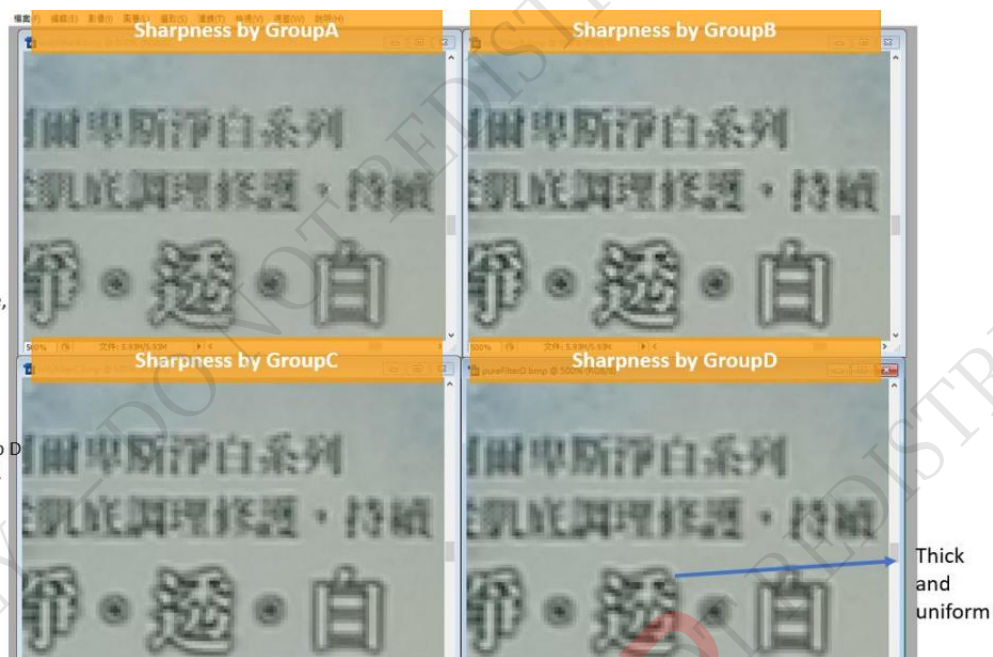


Fig 6.215 Corresponding result

Hard edge area
behavior
– GroupC~D
is better



Fig 6.216 Corresponding result by example setting

For texture, somewhat higher variance sharpness help let texture look natural, therefore, groupB is better than group D.

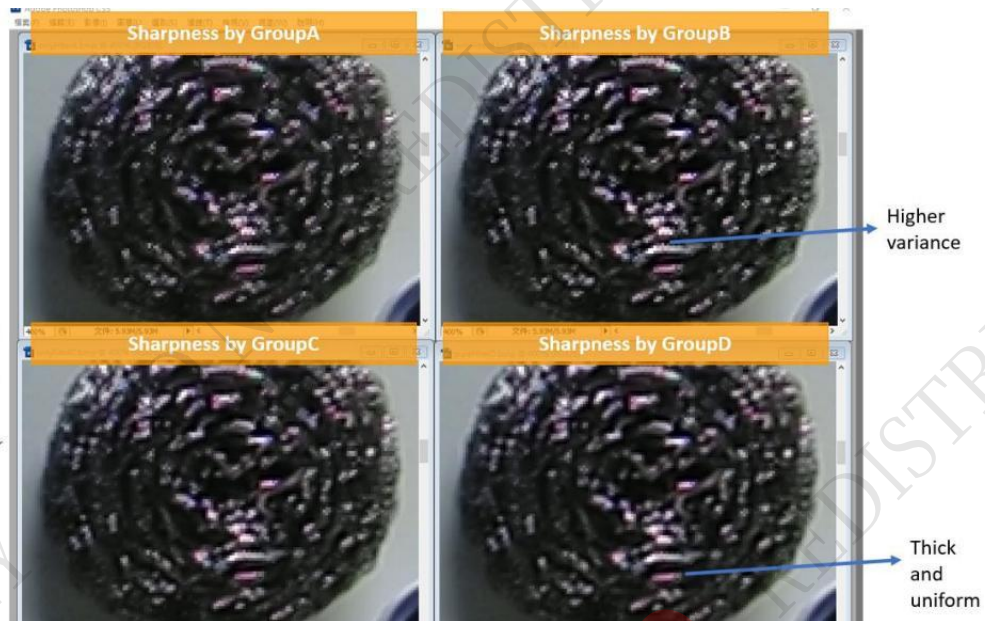


Fig 6.217 Corresponding result

Smooth area behavior
– GroupA~B is better



Fig 6.218 Corresponding result by example setting

For texture, somewhat higher variance sharpness help let texture look natural, therefore, groupB is better than group D.

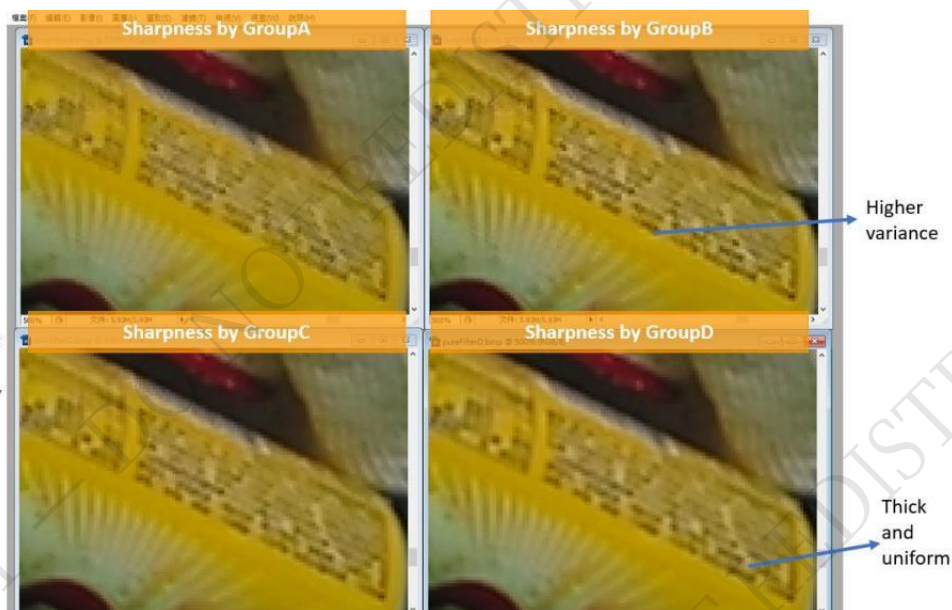


Fig 6.219 Corresponding result

- Suggested setting1: (use all kinds of filter to handle with different feature pixels)
Smooth area's feature value is often less than 5, while hard edge's feature value is often larger than 25 in past experience. For feature value locates in 5~25, Group B, C can handle with them. (mainly for soft texture ~ strong texture) For smooth area, since only using GroupA, even enhancing somewhat noise, noise still looks thin. (not thick style). For hard edge, GroupD is used to let hard edge look uniform and smooth.

- Suggested setting 2: (enhance fine texture as higher priority; EX: low iso case)
Use Group B for most of pixels in image; and utilize Group C to help keep hard edge's smooth.

Example:

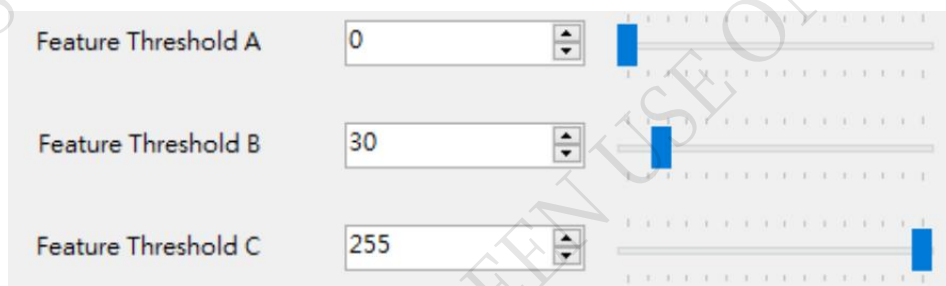


Fig 6.220 example of filter threshold setting

- Suggested setting 3: (protect edge smooth as highest priority in high iso case)
For high iso, often tune conservative sharpness and also need to prevent from ugly strong edge; Mainly use Group D to help achieve this style

Example:

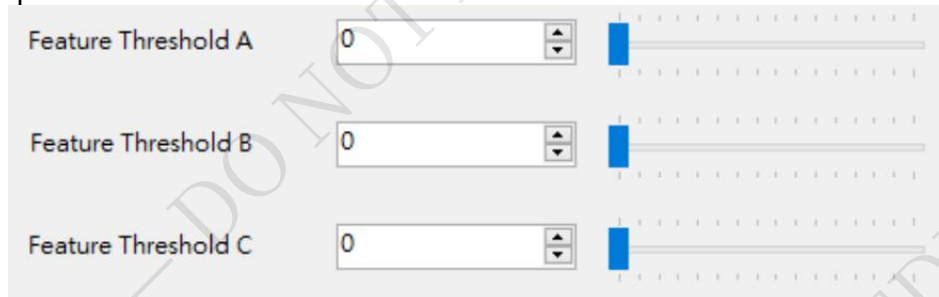


Fig 6.221 example of filter threshold setting

6.14.5. Tunings for “Laplacian” sub-page

For this sub-page, sharpness enhancement value can be adjusted depending on preference.

- For positive value & negative value of sharpness enhancement, adjusting their response curves
- Image center and corner can set difference response curves for reducing corner blur causing from lens (EX: let corner response curve have higher value than center response curve)

Lens may cause image corner blur, example is shown:

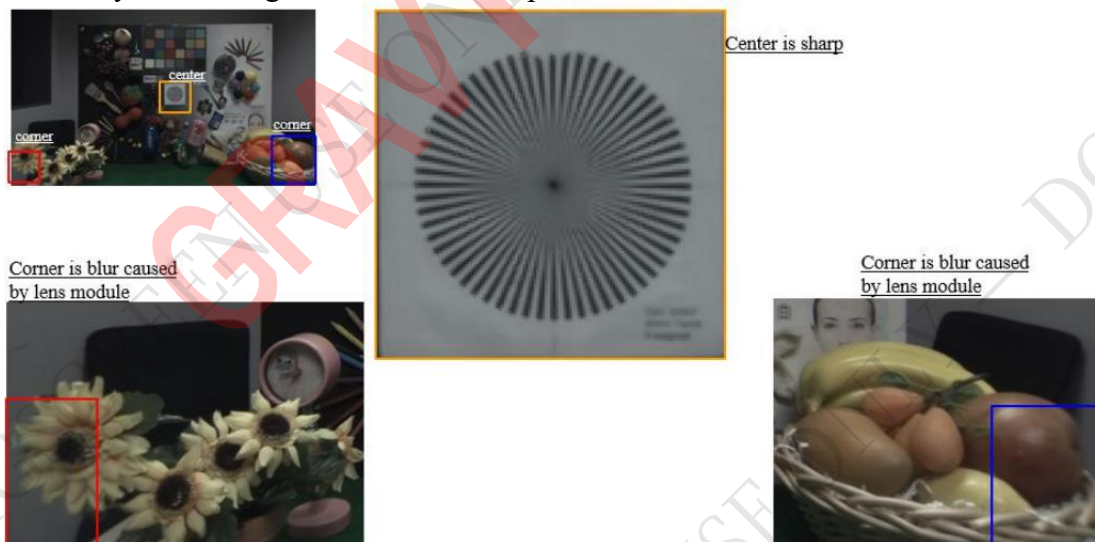
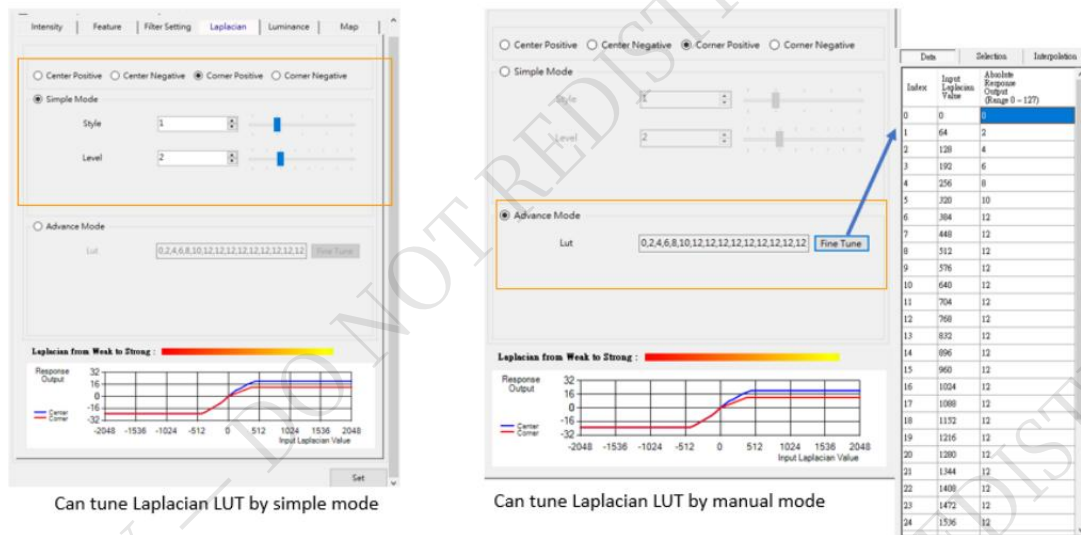


Fig 6.222 Lens causes image corner blur

To compensate it, user can let image corner pixel adopting relative aggressive sharpness than image center. Four curves need to be tuned: center positive, center negative, corner positive, corner negative. Two different tuning method is shown below:



X axis: input response value
Y axis: output mapping response value

Fig 6.223 Two methods for Laplacian value tuning

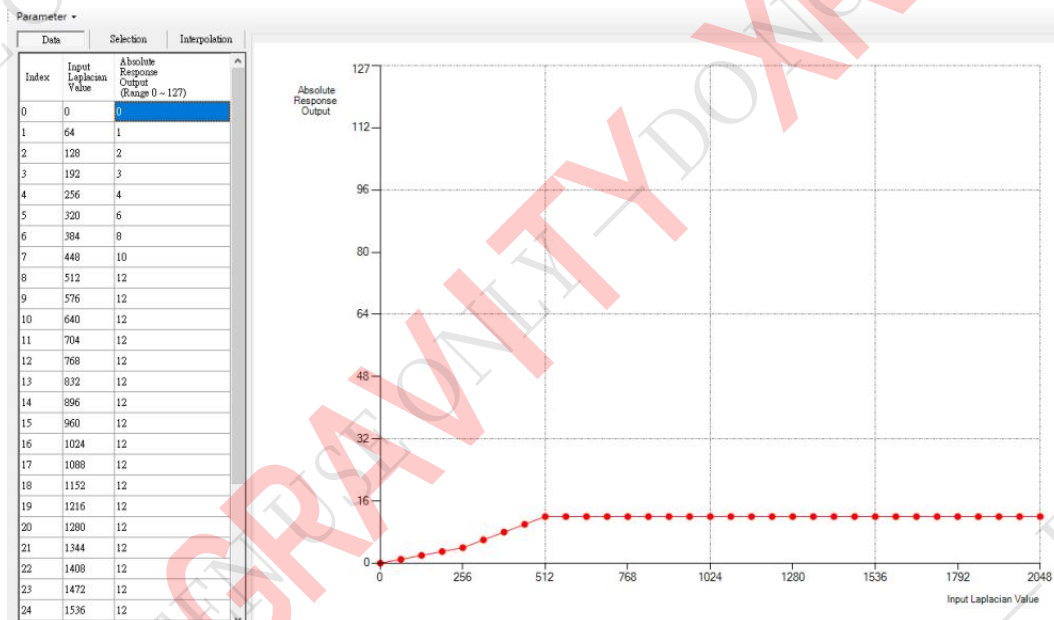


Fig 6.224 Description for Laplacian response curve

In this curve:

- X-Axis: input Laplacian value
- Y-Axis: desired response

User can tune it to determine positive halo & negative halo response amount; center response & corner response amount can also be different. From low response to high response, curve is divided by 33 end points, mapping to Laplacian value of 0, 64, 128, 192.....2048. For each Laplacian value end point, user can set desired response value, data range is 0~127. Higher value means higher corresponding response for sharpness preference. Because that Laplacian value may be negative, when tuning negative curve, the value show on UI will be noted as negative value.

Tuning example is shown below:

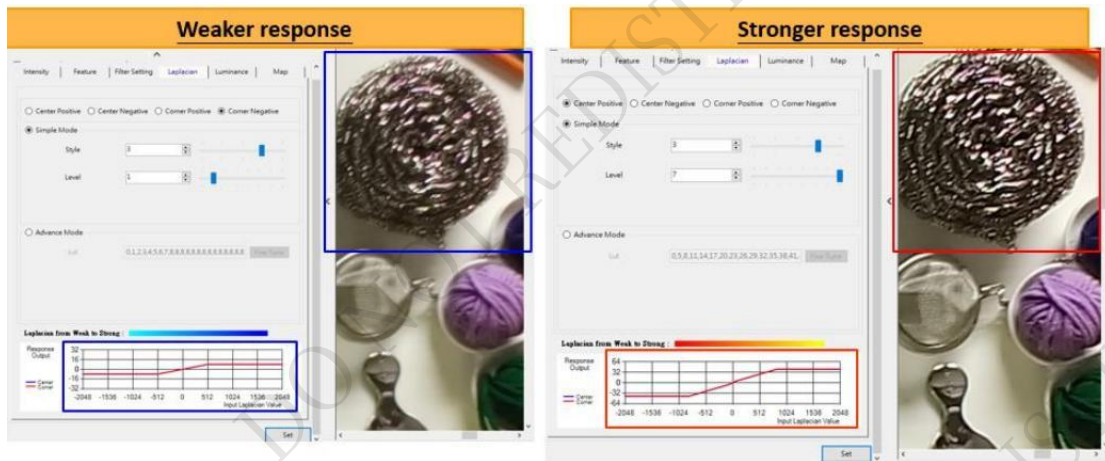


Fig 6.225 Example of Laplacian value tuning

Suggested settings for Laplacian curve:

1. Smooth increasing from zero to a desired strength (EX: 10~20 at end)
2. Final strength is suggested to be less than 20 to avoid too obvious halo

General suggested setting is shown in below figure:

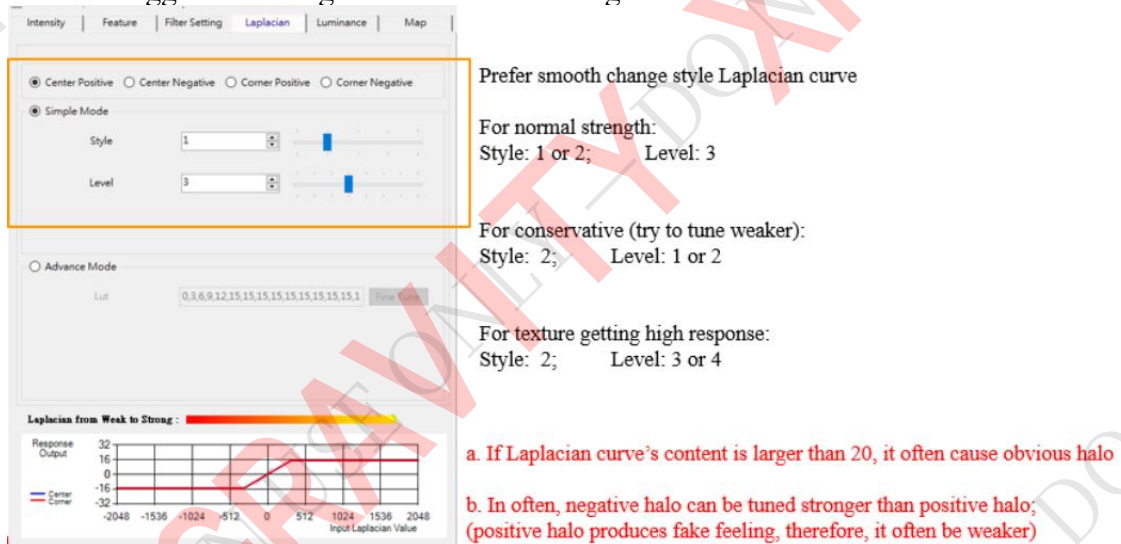


Fig 6.226 Suggestion of Laplacian curve tuning

6.14.6. Tunings for “Luminance” sub-page

For each pixel, user can tune overshooting control by two factors, called it as alpha1 & alpha2;

■ alpha1:

- references from pixel intensity, in high luminance area or dark area, sharpness level can be further reduced or amplified by alpha1.

- Tuning method:

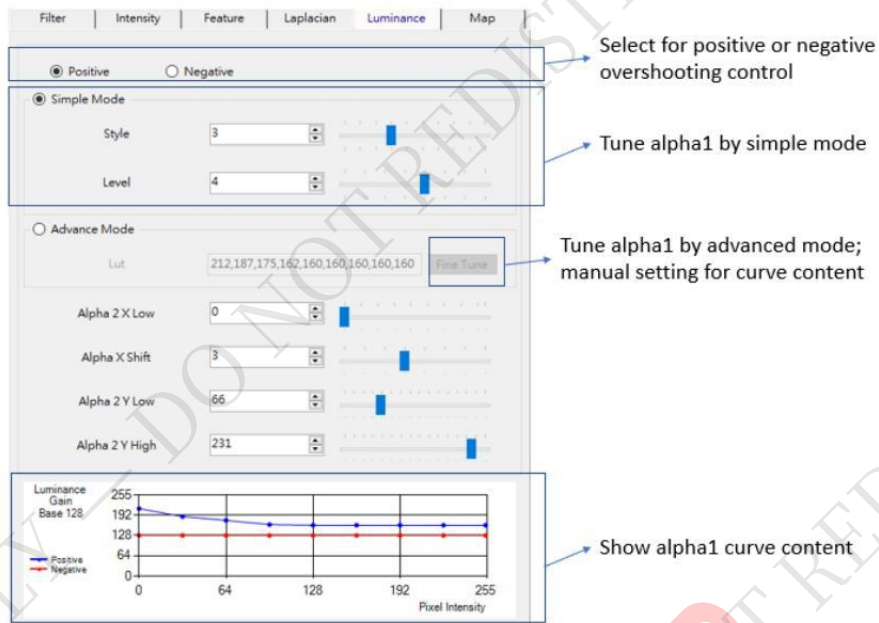


Fig 6.227 Icons for tuning Alpha1

1. Both of positive & negative shooting control need to be tuned individually
2. User can use simple mode tuning or set curve content by advanced mode.
3. There are 9 end points mapping to corresponding luminance value as below figure:

Input Intensity	Mapping output luminance gain
0	Element1
32	Element2
64	Element3
96	Element4
128	Element5
160	Element6
192	Element7
224	Element8
256	Element9

Fig 6.228 End points for Alpha1 Curve

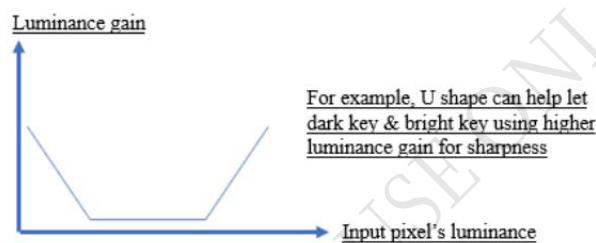


Fig 6.229 meaning of Alpha1 curve

Tuning example if shown below:

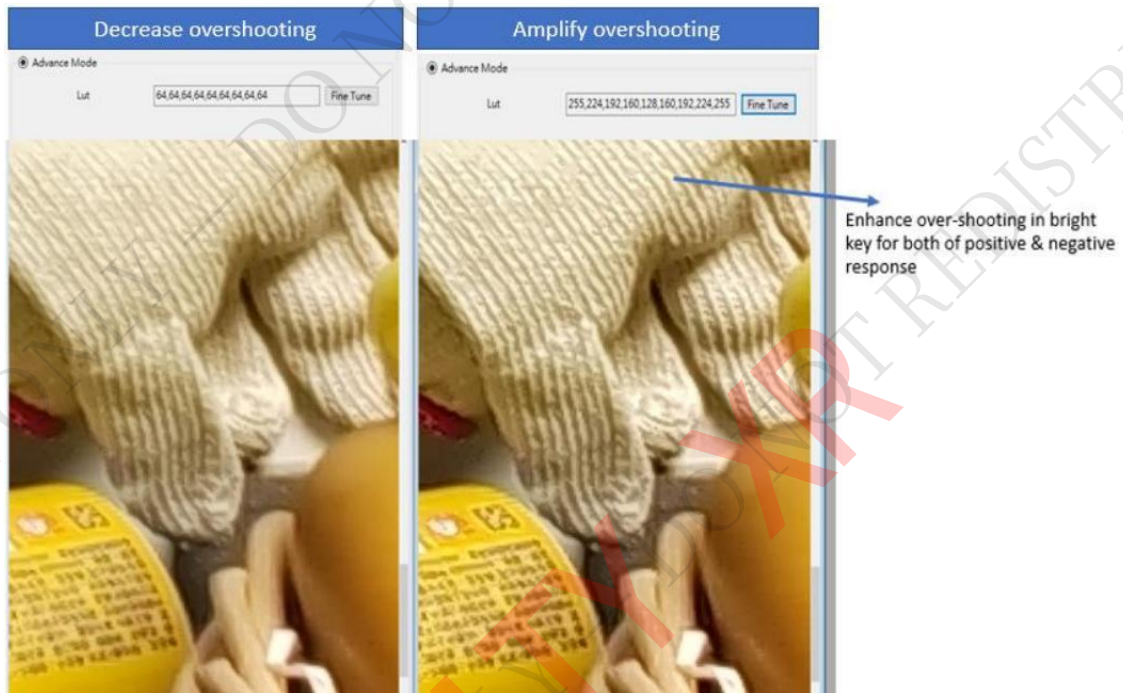


Fig 6.230 Tuning example of luminance shooting control

Suggestion for tuning:

1. Usually, dark key & high key can put relative stronger shooting; (U shape)
2. Another usual using method:
→ Bypass this preference, just let each element be 128

■ alpha2:

- ☐ reference from absolute response amount, doing overshooting control according to current response value. Idea is shown below:

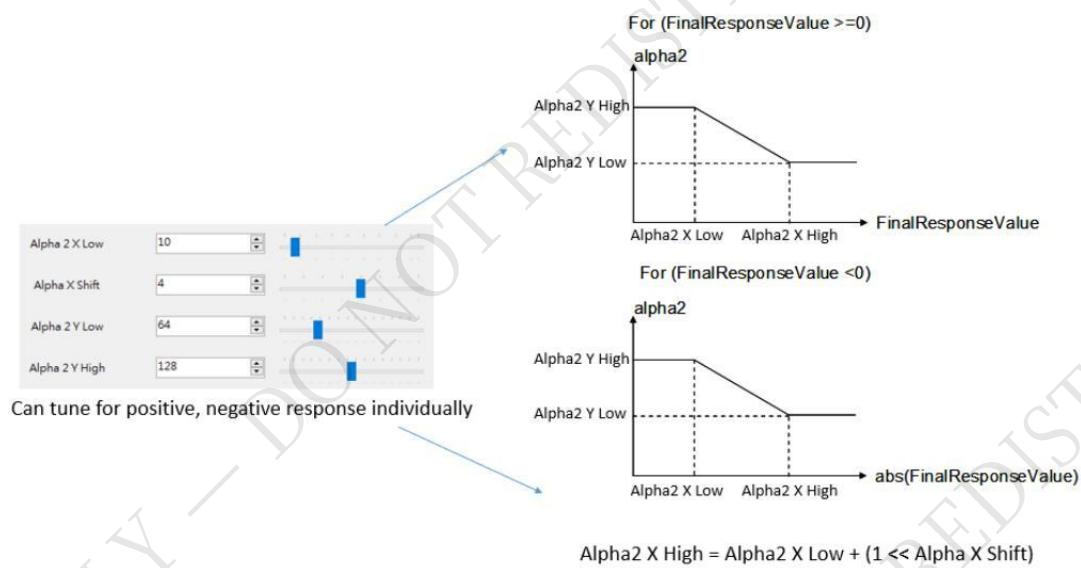


Fig 6.231 Concept of alpha2 setting



Fig 6.232 Example of alpha2 tuning

Using Alpha2 X Low & Alpha X Shift to determine the response region for suppression, later, using Alpha2 Y High, Alpha2 Y Low to determine the response decreasing amount.

At last, sharpness overshooting control factor will be determined by mean value of alpha1 & alpha2. Both of alpha1 & alpha2 can be tuned for positive or negative response control individually.

Suggestion for tuning:

→ Usually, if no specific purpose, user can just let Alpha2 Y Low & Alpha2 Y High use same value (namely don't want to do suppression by Alpha2); if these two values are lower and using identical value, then it will let shooting control mainly come from Alpha1.

6.14.7. Tunings for “Map” sub-page

Map means corner response's weighting; for below example, right picture's corner part is much bright, it means that image corner will adopt more weighting for corner response curve's content. For an input pixel, it needs to look up center response curve to know the center version sharpness value, it also needs to look up corner response curve to know the corner version sharpness value in section of “Laplacian” curve tuning. At last, by input pixel's map value, do weighting average for center and corner sharpness value to get determined response. If user prefers to let image corner pixel get higher response, at first, needing to let corner curve have higher values, later, needing to tune map as below example's right figure style to let image corner pixel can blend with corner response by higher weighting.

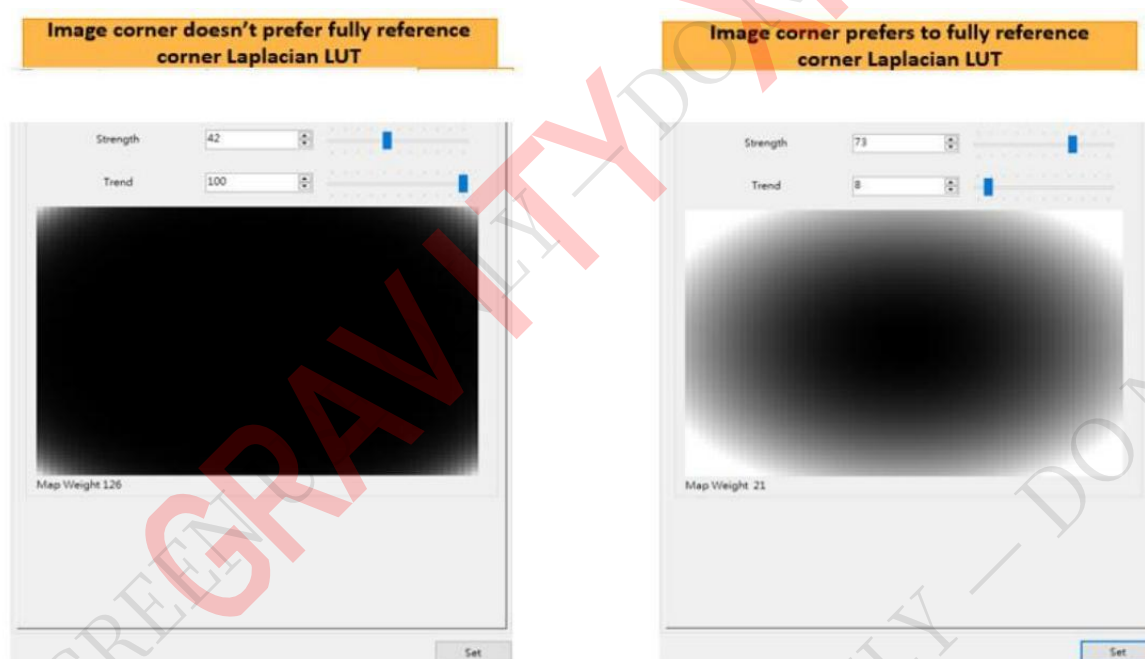


Fig 6.233 Map tuning example

Choose desired map content by tuning “strength” & “trend” on UI of Map page. Tuning example is shown below:

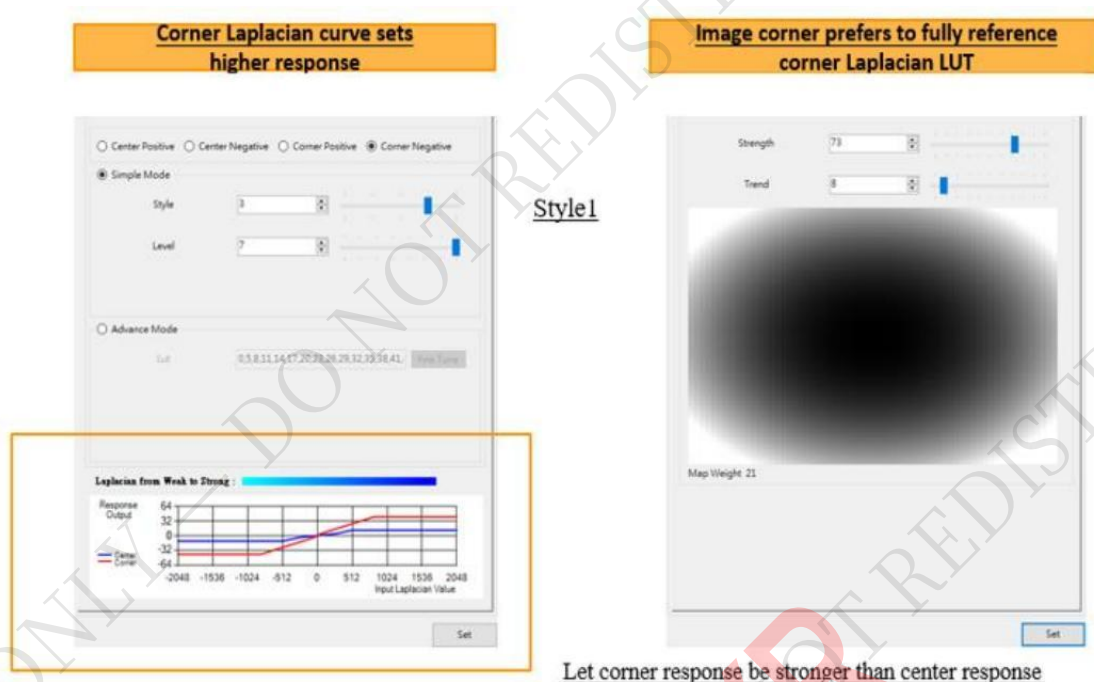


Fig 6.234 Map tuning example



Fig 6.235 Map tuning example

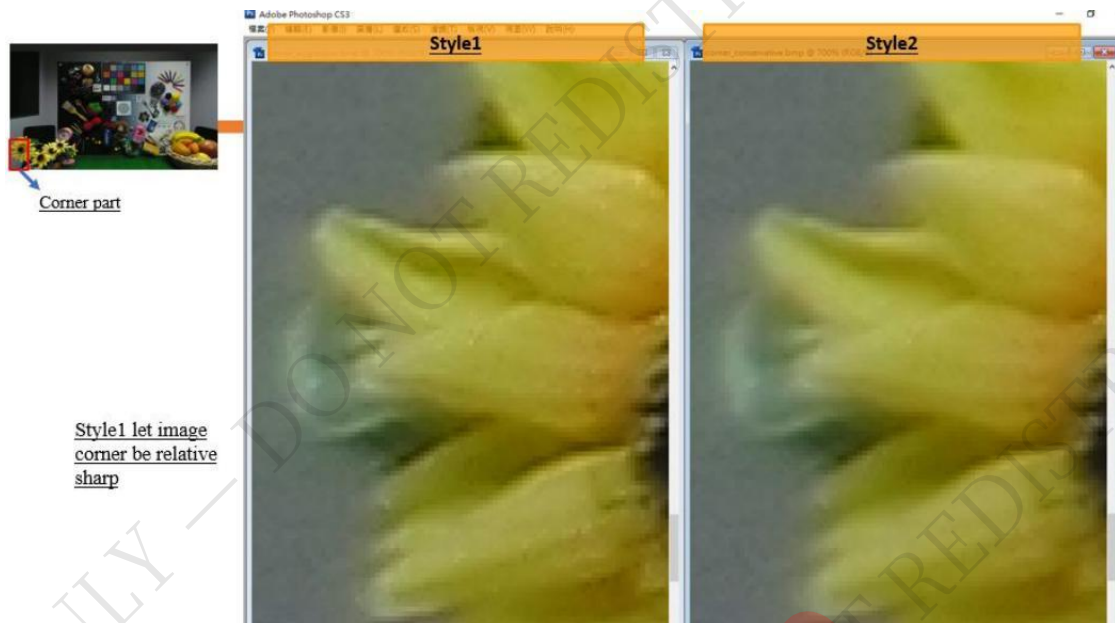


Fig 6.236 Map tuning example

In order to let user get mapping relationship with input image and map, Help UI is designed, example is shown below:

By referencing image & map relationship, user can reference it to help tune map content.

Suggestion for tuning:

→ Usually needing a smooth change style, suggested setting is just like below:



Fig 6.237 Suggested map style — corner be relative aggressive and map is smooth

7. Auto Exposure & Auto White Balance Tuning Guide

7.1. Auto Exposure Tuning Guide

Revision 1.0

7.1.1. Introduction

This document is a tuning guide for the auto exposure algorithm. The following content will introduce the relevant tuning parameters and application scenarios of the auto exposure algorithm.

7.1.2. Metering

Field	Value
aeMeteringMode	1
manualISOLevel	0
manualExptime	0
manualADGain	0
manualAETarget	100
manualATFMode	8
aeTargetMode	1
aeExpoMode	2
meteringSigma	50.00

Field	Value
wdrEnable	1
luxidkth_0	180
luxidkth_insert_01	200
luxidkth_1	230
luxidkth_insert_12	240
luxidkth_2	250
luxidkth_insert_23	290
luxidkth_3	320
luxidkth_insert_34	360
luxidkth_4	360
luxidkth_insert_45	370
luxidkth_5	370
luxidkth_insert_56	390
luxidkth_6	390
luxidkth_insert_67	420
luxidkth_7	420
luxidkth_insert_78	420
luxidkth_8	470
luxidkth_insert_89	470

Field	Value
hist_sat_low	100
hist_sat_high	120
hist_dark_low	50
hist_dark_high	60
hist_sat_pushup_range_start0(scale1000)	0
hist_sat_pushup_range_start1(scale1000)	0
hist_sat_pushup_range_start2(scale2000)	0
hist_sat_pushup_range_end0(scale1000)	30
hist_sat_pushup_range_end1(scale1000)	30
hist_sat_pushup_range_end2(scale1000)	30
hist_sat_pushdown_range_start0(scale1000)	0
hist_sat_pushdown_range_start1(scale1000)	0
hist_sat_pushdown_range_start2(scale1000)	0
hist_sat_pushdown_range_end0(scale1000)	10
hist_sat_pushdown_range_end1(scale1000)	10
hist_sat_pushdown_range_end2(scale1000)	10
hist_dark_range_start0(scale1000)	0
hist_dark_range_start1(scale1000)	0
hist_dark_range_start2(scale1000)	0
hist_dark_range_end0(scale1000)	100
hist_dark_range_end1(scale1000)	100
hist_dark_range_end2(scale1000)	100
max_adjust_ratio	120
min_adjust_ratio	80
max_midtones_gain	1000

Fig 7.1 Metering tab

1. Common

1. Metering Mode: aeMeteringMode & aeMeteringSigma

Parameter	0	1
Mode	Global Metering	Center-weighted

		Metering
--	--	----------

Tip: When using center-weighted metering, the parameter aeMeteringSigma (0-1000) controls the size of the central circle. The smaller the sigma of the Gaussian kernel, the smaller the central circle, the greater the weight difference between the center and the surrounding areas, and the faster the convergence rate.

2. manualISOLevel

Parameter	0	1	2	3	4	5	6	7	8
Mode	Auto	100	200	400	800	1600	3200	6400	12800

The default value is 0. Parameter values 1 to 8 represent that the AE is fixed to the corresponding ISO value.

3. manualExptime & manualADGain & manualAETarget & ae_target_mode

- **manualExptime**: Represents the manual exposure time, which allows the user to set the exposure time manually according to the actual scene. A non-zero value for this parameter means the exposure time is set manually (in microseconds, μ s).
- **manualADGain**: Refers to the manual analog gain (scale 100). A non-zero value for this parameter means the gain is set manually. When used in conjunction with manualISOLevel (a non-zero value), manualADGain takes higher priority.
- **manualAETarget**: Is the manual exposure target value, which is used to specify the desired image luminance. aeTargetMode needs to be set to 0 (scale 100). A setting range of [100,25500] means the target is set manually.
- **aeTargetMode**: Is the exposure target value mode selection.

Parameter	0	1
Mode	Manual Fixed Mode	Auto Mode

4. manualATFMode

Parameter	0	1	2	8
Mode	OFF	50HZ	60HZ	ATF Algorithm Result

5 . WDRMode

It is not recommended to modify these parameters; keep the default values.

2. WDR Settings

1. wdrEnable

Parameter	0	1
Mode	OFF	ON

2. WDR Lux Selection Interval

Adjust the interval to control the corresponding WDR scene parameters under different lux values, with a total of 10 scenarios. Example: When $\text{luxIdx} \leq \text{luxIdxth_0}$, the WDR parameters of case0 are selected; when $\text{luxIdxth_0} < \text{luxIdx} \leq \text{luxIdxth_insert_01}$, the WDR parameters are interpolated from case0 and case1; when $\text{luxIdxth_insert_01} < \text{luxIdx} \leq \text{luxIdxth_1}$, the WDR parameters of case1 are selected, and so on. The recommended range for the above relevant parameters is 0 to 600.

3. WDR Case Parameter Configuration

Parameter	Description	Range
hist_dark_high	The maximum appropriate luminance of the dark area adjusted under the current environment; the larger the value, the larger the dark high target	0 to 255
hist_dark_low	The minimum appropriate luminance of the dark area adjusted under the current environment; the larger the value, the larger the dark low target	0 to 255
hist_sat_high	The maximum appropriate luminance of	0 to 255

	the mid-high bright area adjusted under the current environment; the smaller the value, the smaller the bright high target	
hist_sat_low	The minimum appropriate luminance of the mid-high bright area adjusted under the current environment; the smaller the value, the smaller the bright low target	0 to 255
hist_dark_range_end	Used to calculate the moving average luminance of pixels whose proportion in the histogram is in the range of start% to end% from the dark end as the current luminance of the dark area, and finally calculate the dark target. The higher the end percentage of the pixel proportion range, the smaller the dark target value (scale 1000)	0 to 1000
hist_dark_range_start		0 to 1000
hist_sat_pushdown_range_end	Used to calculate the moving average luminance of pixels whose proportion in the histogram is in the range of start% to end% from the start as the current luminance of the mid-high bright area, and finally calculate the bright high target. The earlier the proportion, the smaller the bright high target value	0 to 1000
hist_sat_pushdown_range_start		0 to 1000
hist_sat_pushup_range_end	Used to calculate the moving average	0 to 1000

hist_sat_pushup_range_start	luminance of pixels whose proportion in the histogram is in the range of start% to end% from the start as the current luminance of the mid-high bright area, and finally calculate the bright low target. The earlier the proportion, the smaller the bright low target value (scale 1000)	0 to 1000
max_adjust_ratio	The safe target value must be $\leq$ the reference target value multiplied by max_adjust_ratio	100 to 200
min_adjust_ratio	The safe target value must be $\geq$ the reference target value multiplied by min_adjust_ratio	0 to 100
max_midtones_gain	Maximum value of midtones_gain	100 to 3200
max_shadows_gain	Maximum value of shadows_gain	100 to 3200
max_isp_midtones_gain	Maximum value of isp_midtones_gain, generally equal to max_midtones_gain	100 to 3200

7.1.3. Arbitration (SDR Target)

1. Target Table

targetTable	expoTable				
	Normal	Preview	Night	Face	Sport
▶	1	1	1	1	1

Fig 7.2 TargetTable (1)

The above parameters mean that the table of the corresponding scene is enabled when the value is 1, otherwise disabled.

NORMAL PREVIEW NIGHT FACE SPORT			
	luxidx	value	default0
▶	0	7000	0
	22588	6500	0
	24933	5800	0
	27059	5500	0
	29404	4800	0
	31132	3500	0
	35067	2300	0
	39737	1200	0

Fig 7 . 3 TargetTable (2)

Note: At present, only the Normal Scene takes effect after debugging, and the switching of other scenes is not open yet.

The luxidx column in the TargetTable above is equivalent to the expindex column in the ExpoTable. The value in the value column is the SDR target value. A larger luxidx number indicates a darker current environment, and a larger value number indicates a brighter target luminance. Generally, the value needs to decrease from top to bottom, and the minimum value is not less than 100. The current environment BV is distinguished according to the calculated luxindex, and the corresponding index & target value (scale 100) is adjusted according to the current environment.

Recommendation: Generally adjust rows 1 and 2 for outdoor scenes, rows 3-5 for indoor scenes, and rows 5-8 for dark environments. Calculations are based on raw block statistics. When setting the target value, the gamma effect must be considered, and the algorithm uses a default reverse inference with a 2.2 gamma value.

2. Expo Table

targetTable expoTable			
AETABLE_PRECII	AETABLE_CAPTL	AETABLE_CAPTL	AETABLE_CAPTL
1	1	1	1

Fig 7 . 4 ExpoTable(1)

The above parameters mean that the table of the corresponding scene is enabled when the value is 1, otherwise disabled.

PREVIEW_AUTO	CAPTURE_AUTO	CAPTURE_NIGHT	CAPTURE_HDR
expindex	exptime	gain	prio
516	11	1	1
23563	10000	1	0
28253	10000	4	1
30598	20000	4	1
38388	50000	16	0
40733	50000	32	1
41672	66000	32	0
44017	66000	64	1

Fig 7. 5 ExpoTable(2)

Note: At present, only CAPTURE_AUTO and PREVIEW_AUTO take effect after debugging, and the switching of other scenes is not open yet.

At present, the AE algorithm will automatically switch the corresponding exposure table strategy based on IMU motion detection, with Normal for CAPTURE_AUTO and the Motion Table for PREVIEW_AUTO. The expindex setting of each node in the exposure table should match exposure time and gain, that is, the parameter combination set for the first node (the calculation formula will be marked below). Prio: When the expindex is between two nodes, whether the exposure change is prioritized through exposure time or gain change; 0 for gain change, 1 for exposure time change.

Formula: $100 * \text{expindex} = \min \text{expindex} + \text{LOG}(\text{exposure time} * \text{gain} / (\min \text{exposure time} * \min \text{gain})) / \text{LOG}(1.3)$

Among them, min expindex = 516, min exposure time = 11, min gain = 1 (i.e., the set values in the first row).

7.1.4. Multi Camera

1. Multi Cam Mode

Five operating modes are available, with parameter values ranging from 0 to 4 respectively:

- Mode 0: Binocular cameras do not affect each other, and each performs auto exposure (selected when the effect of left and right eye corner cases is not considered).
- Mode 1: Master-slave mode, only the master camera converges, and the slave camera uses the same settings. (Good effect stability when the sensor difference is small, with corner cases such as large brightness differences between the content of the left and right eyes).
- Mode 2: Dynamically select the master camera, then the slave camera follows and converges with reference to the exposure of the master camera, and the convergence logic of the master camera remains unchanged (Recommended).

- Mode 3: The slave camera only mixes the master camera statistics according to weights in the overlapping area.
- Mode 4: The statistics of the overlapping area and edge area of the master and slave cameras are mixed into a new set of statistics according to weights (Verification stage).

2. Other Parameters

Parameter	Description	Range
BVStart	The weight is calculated by interpolation between BVStart and BVEnd according to the current BV value; generally, the darker the environment, the higher the weight. The weight value is set to slaveWeightStart when BV is less than BVStart	-5000 to 5000
BVEnd	The weight value is set to slaveWeightEnd when BV is greater than BVEnd	-5000 to 5000
slaveWeightStart	Slave weight when BV is less than BVStart	0 to 1.0
slaveWeightEnd	Slave weight when BV is greater than BVEnd	0 to 1.0
varianceStableTh	The master and slave camera selection will be adjusted only if the difference in the statistical variance of the binocular cameras is greater than varianceStableTh	0 to 50
ymeanStableTh	The master and slave camera selection will be adjusted only if the difference in the average	0 to 500

	luminance of the binocular cameras is greater than ymeanStableTh	
stableFrame	The final master and slave camera selection is confirmed only when the selection remains unchanged for consecutive stableFrame frames	0 to 50
cameraMixROI.left.start	The starting horizontal block position of the overlapping area of the left cameraExample: When the statistical block is 16*16, this parameter can be set to 3	0 to 32
cameraMixROI.left.end	The ending horizontal block position of the overlapping area of the left cameraExample: When the statistical block is 16*16, this parameter can be set to 16	0 to 32
cameraMixROI.left.top	The starting vertical block position of the overlapping area of the left cameraExample: When the statistical block is 16*16, this parameter can be set to 0	0 to 24
cameraMixROI.left.bottom	The ending vertical block position of the overlapping area of the left cameraExample: When the statistical block is 16*16, this parameter can be set to 16	0 to 24

cameraMixROI.right.start	The starting horizontal block position of the overlapping area of the right cameraExample: When the statistical block is 16*16, this parameter can be set to 0	0 to 32
cameraMixROI.right.end	The ending horizontal block position of the overlapping area of the right cameraExample: When the statistical block is 16*16, this parameter can be set to 13	0 to 32
cameraMixROI.right.top	The starting vertical block position of the overlapping area of the right cameraExample: When the statistical block is 16*16, this parameter can be set to 0	0 to 24
cameraMixROI.right.bottom	The ending vertical block position of the overlapping area of the right cameraExample: When the statistical block is 16*16, this parameter can be set to 16	0 to 24

Note: slaveWeightStart, slaveWeightEnd, varianceStableTh, ymeanStableTh are used for Mode 2; cameraMixROI.left.start, cameraMixROI.left.end, cameraMixROI.left.top, cameraMixROI.left.bottom, cameraMixROI.right.start, cameraMixROI.right.end, cameraMixROI.right.top, cameraMixROI.right.bottom are used for Mode 3.

7.1.5. ConfigInfo (Sensor Configurations)

Parameter Configuration

Parameter	Description	Range
aeEnable	Whether to enable AE,	0,1

	otherwise return directly	
pipelay	Number of frame intervals from the issuance of exposure parameters by AE to the actual effective exposure of the sensor	0 to 20
fNumberX1000	Aperture (scale 1000), fixed at 2200 in internal projects	Fill in according to the sensor
maxfps	(scale 100). When exptimePerLineNs is 0, calculate hwmintime with 100000000/maxfps, which is used to limit the minimum exposure time. The larger one of hwMinExposureTime and swMinExposureTime is used as the limit of the minimum exposure time	3000
minfps	(scale 100). When min_fps is not 0, calculate maxExposureTimeFromFPS with 100000000/minfps, and compare it with maxExposureTimeFromLine calculated by maxLineCnt and exptimePerLineNs; the smaller one is used as hwMaxExposureTime	1600
maxGain	(scale 100), maximum AD gain	100 to 6400
minGain	(scale 100), minimum AD gain	100 to 6400

maxLineCnt	Used to calculate maxExposureTimeFromLine below	1 to 65535
minLineCnt	Used to calculate hwMinExposureTime below	1 to 65535
swMaxExposureTime	Compare hwMaxExposureTime with swMaxExposureTime; the smaller one is used as the final maxExposureTime	1 to 11111
swMinExposureTime	Compare hwMinExposureTime with swMinExposureTime; the larger one is used as the final minExposureTime	1 to 11111
exptimePerLineNs	(scale 1000). When exptimePerLineNs is not 0: maxExposureTimeFromLine= exptimePerLineNs*maxLineCnt/1000.0f, hwMinExposureTime=exptimePerLineNs*minLineCnt/1000.0f; When exptimePerLineNs is 0 and maxLineCnt and minfps are not 0: maxExposureTimeFromLine =(1000000000/minfps/maxLineCnt*100/1000)*maxLineCnt, hwMinExposureTime=(1000000000/minfps/maxLineCnt*100/1000)*minLineCnt;	1 to 10000

maxSensorGain	Maximum gain of the sensor (again)	100 to 1600
---------------	------------------------------------	-------------

7.1.6. Converged

Parameter Configuration

Parameter	Description	Range
convSpeedLevel	5 convergence gears: normal, direct, fast, smooth, slow corresponding to 0 to 4 respectively	0 to 4
inConvYmeanTh	Dual-threshold judgment logic for the range of convergence entry and exit conditions. This parameter is the convergence entry threshold; when switching from non-convergence to convergence, it is considered that convergence is achieved when the target gap is within this threshold.	0 to 25600default: 300
outConvYmeanTh	Dual-threshold judgment logic for the range of convergence entry and exit conditions. This parameter is the convergence exit threshold; when switching from convergence to non-convergence, it is considered that re-convergence is required when the target gap is outside this threshold.	0 to 25600default: 400

convBVCntTh	This threshold indicates the number of consecutive frames that the BVGAP meets the requirements before triggering the convergence completion flag	0 to 100default: 3
convTgtCntTh	This threshold indicates the number of consecutive frames that the target Y gap meets the requirements before triggering the convergence completion flag	0 to 100default: 3
Curvxxx_TimeSmooth	β value of the damped convergence model, mainly controlling convergence stability and total convergence time	0 to 1
Curvxxx_firstpower	Slope of the first linear convergence function of the three-stage convergence model, determining the anti-overshoot and stability control parameters for convergence at low gap	0 to 1
Curvxxx_quad	Constant of the third quadratic convergence function of the three-stage convergence model, i.e., controlling the step rate of the fast convergence segment at large gap, determining the time to quickly reach the convergence target for the first time in the three-stage model	0 to 1

Curvxxx_cube	Constant of the second cubic convergence function of the three-stage convergence model, i.e., controlling the step rate of the smooth convergence segment at medium and low gap, determining the convergence smoothness of the three-stage model	0 to 1
--------------	--	--------

7.2. Auto White Balance

Tool Overview

This tool is a visual ISP (Image Signal Processing) debugging tool developed based on Python 3.x+PyQt5, specially designed for Auto White Balance (AWB) and Color Correction Matrix (CCM) algorithms. It strictly follows the core AWB logic of "Grey World Theory + Pixel RGB Statistical Distribution" and the CCM correction principle of "24-color Chart Calibration + Color Difference Constraint", realizing the full-process debugging from statistical data analysis, region division, weight configuration to matrix optimization through a modular interface.

The tool supports independent operation on Windows systems (can be packaged as EXE) without relying on additional development environments, with core advantages as follows:

- 1. Parameter Visualization: All key AWB/CCM parameters are presented in the form of tables/input boxes, which are intuitive and editable;
- 2. Full-process Coverage: Covering the entire debugging link from original parameter configuration to final effect verification;
- 3. Data Persistence: Supporting the loading and saving of parameter files in .bin format to ensure the reusability of debugging results;
- 4. Version Traceability: Facilitating parameter iteration management through the dual version identifiers of PARAM_VERSION and DEBUG_LEVEL.

Detailed Explanation of Core Interfaces and Functional Modules

The tool adopts a multi-tab layout and is divided into 5 core functional areas. The functions, corresponding parameters and debugging significance of each component are explained in detail in the order of tabs below:

7.2.1. AWB Core Parameter Configuration

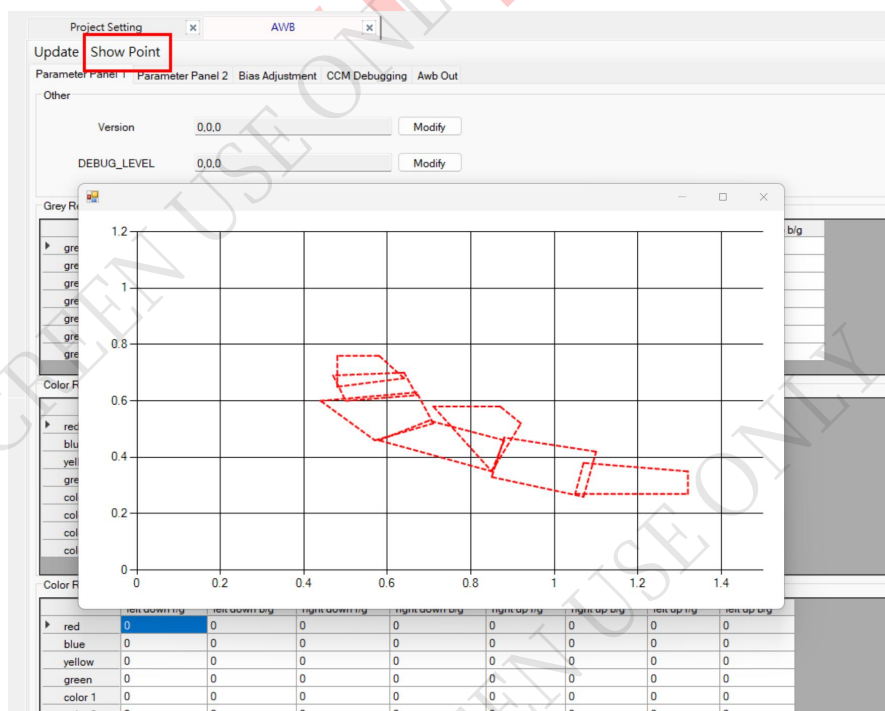
This tab focuses on the basic parameter configuration of the AWB algorithm, including the Version Parameter Input Area, Light Source Calibration Table and Gray Area Boundary Table, which is the starting point of AWB debugging.

1. Version Parameter Input Area

Component Name	Corresponding Parameter	Function Description	Debugging Significance
PARAM_VERSION Input Box	Self.param.PARAM_VERSION (3 integers)	Input format is "Value 1, Value 2, Value 3", the default is [1,1,1],	Used for parameter version management, which can

		supporting manual editing and real-time synchronization to the parameter structure	distinguish parameter configurations in different debugging stages for convenient iteration traceability (e.g., V1.0.0 corresponds to 1,0,0)
Parameter Version (DEBUG_LEVEL) Input Box	Self.param.DEBUG_LEVEL (3 integers)	Complementary to the function of PARAM_VERSION, used to mark the debugging state	Assisting team collaboration and avoiding the confusion of parameters in different states

Push the "Show Point" button to draw the Grey and color regions and the points from AWB stats on the (r/g,b/g) platform.



2. Light Source Parameter Table

	R/G	B/G	R/G shift	B/G shift
H	123	32	-25	12
U30	99	37	-16	14
TL84	68	45	-4	6
CWF	72	46	-4	6
D50	63	57	-6	7
D65	57	66	0	0
D75	53	71	2	-2

	left down r/g	left down b/g	right down r/g	right down b/g	right up r/g	right up b/g	left up r/g
grey region 1	111	27	132	27	132	35	107
grey region 2	85	33	107	26	110	42	88
grey region 3	85	35	92	52	87	58	71
grey region 4	58	46	85	35	88	46	70
grey region 5	57	46	71	52	67	63	44
grey region 6	50	60	67	62	64	70	47
grey region 7	48	65	64	68	58	76	48

Fig 7 .6 Light Source Parameter

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	7 rows (number of light sources) × 4 columns (parameter items)	Rows: 7 standard light sources including H, U30, TL84, CWF, D50, D65, D75; Columns: R/G, B/G, R/G shift, B/G shift Storing the reference color cast data under each light source	Light source calibration is the foundation of the AWB algorithm. Calibrating the reference values enables the algorithm to identify the "ideal white" under different light sources, and adjusting the offset values modifies the colors to conform to the subjective perception of the human eye.
Core Parameters	R/G, B/G reference values:	Table cell values are directly	Different light sources have

	first two columns; R/G, B/G offset values: last two columns	mapped to the parameter structure, supporting integer input (range: reference values [0~400], offset values [- 200~200])	different spectral distributions (e.g., H light source is yellowish, D65 light source is close to natural light). Calibrating reference values allows the algorithm to recognize the "ideal white" under different light sources, and offset adjustment modifies colors to match human subjective visual perception.
--	--	--	--

3. Gray Area Boundary Table

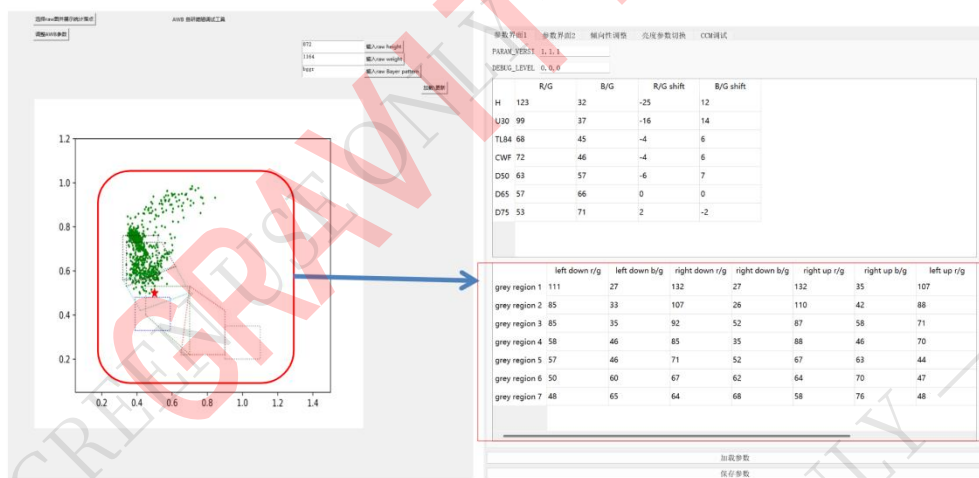


Fig 7.7 Gray Area Boundary

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	7 rows (number of light sources) × 8 columns (boundary)	Rows: corresponding to 7 standard light sources; Columns:	The gray area is the core statistical region of the AWB algorithm

	coordinates)	left down r/g, left down b/g, right down r/g, right down b/g, right up r/g, right up b/g, left up r/g, left up b/g Defining the boundary coordinates of the "gray area" under each light source (in the R/G-B/G two-dimensional space)	(the basis of the grey world hypothesis). Accurately defining the gray area boundary can eliminate the interference of other pixels and ensure the accuracy of AWB calculation.
Core Parameters	GreyRegions (7×8 integer matrix)	Each cell stores the R/G or B/G ratio of the gray area vertex, used to screen the "gray pixels" in the image (i.e., pixels meeting the gray area conditions)	The gray area is the core statistical region of the AWB algorithm (the basis of the grey world hypothesis). Accurately defining the gray area boundary can eliminate the interference of other pixels and ensure the accuracy of AWB calculation.

7.2.2. Color Region and Misleading Color Region Configuration

This tab focuses on the "effective pixel screening" of the AWB algorithm. By defining color regions and misleading color regions, the accuracy of AWB statistics is further optimized to avoid the influence of invalid pixels on color cast correction.

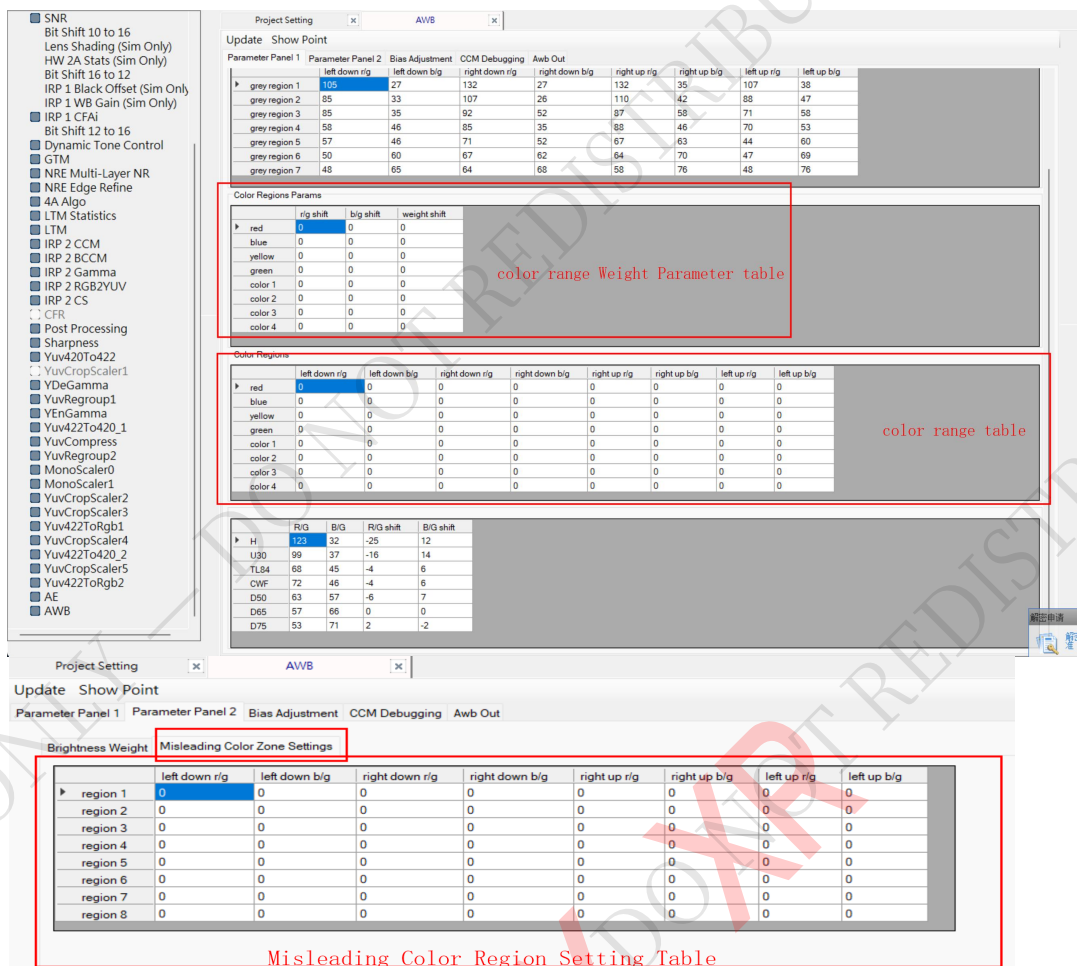


Fig 7.8 effective pixel screening

1. Color Range Table

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	8 rows (number of color regions) × 8 columns (boundary coordinates)	Rows: 8 color regions including red, blue, yellow, green, color1~color4; Columns: consistent with the column definition of the gray area table (8 boundary coordinates) Defining the boundaries of common color regions in the image (e.g., R/G-B/G range of red	Marking "non-gray pixels", the algorithm will specially process the pixels in this region during statistics. A maximum of 8 color regions can be set on the two-dimensional plane.

		objects, green plants), and assigning different weights to the statistical falling points entering the region when calculating the final AWB gain	
Core Parameters	colorRegions (8×8 integer matrix)	Used to mark "non-gray pixels", the algorithm will specially process the pixels in this area during statistics	This table is used to define regions on a 2D plane, with a maximum of 8 color regions configurable.

2. Color Range Weight Parameter Table

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	8 rows (number of color regions) × 3 columns (weight items)	Rows: consistent with the color range table; Columns: r/g shift, b/g shift, weight shift Configuring offset values and weights for each color region to adjust the influence of the color region	Adjusting the statistical data values and weights within the color region to optimize the screening effect of effective pixels
Core Parameters	r/g shift	Parameter range [-200~200], which can directly adjust the value of statistical data in the color region	Directly adjusting the statistical data values within the color region
/	b/g shift	Parameter range [-200~200], which	Directly adjusting the statistical data

		can directly adjust the value of statistical data in the color region	values within the color region
/	weight shift	Parameter range [-200~200], which can directly adjust the weight of statistical data in the color region	Directly adjusting the weight of statistical data within the color region

3. Misleading Color Region Setting Table

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	8 rows (number of misleading color regions) × 8 columns (boundary coordinates)	Rows: 8 misleading color regions including region1~region8; Columns: consistent with the column definition of the gray area table Defining color regions that are "easily misjudged as gray" (e.g., light purple, cream yellow), and the statistical points entering them are not involved in the calculation	Solving color cast problems in special scenarios (e.g., light purple walls are easily misjudged as gray, leading to AWB correction color cast)
Core Parameters	misColorRegions (8×8 integer matrix)	The algorithm will forcibly exclude the pixels in this area during statistics to avoid their interference with gray point judgment	Solving color cast problems in special scenarios (e.g., light purple walls are easily misjudged as gray, resulting in color cast in AWB)

			correction)
--	--	--	-------------

7.2.3. Tendency Adjustment (Brightness Weight Optimization)

This tab focuses on the "scene adaptability" of the AWB algorithm. By adjusting the light source weight under different brightness levels, the performance of AWB in complex brightness environments (e.g., strong light, low light scenarios) is optimized.

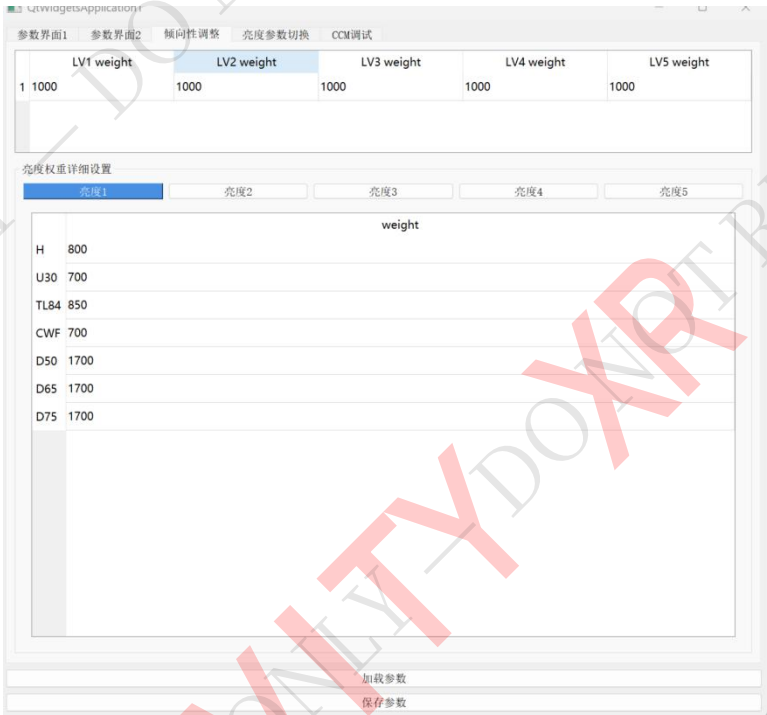


Fig 7 .9scene adaptability

1. Brightness Weight Master Table (the top table of the interface)

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	1 row × 5 columns (brightness levels)	Columns: LV1 weight~LV5 weight (5 brightness gears) Configuring the weight of each brightness gear (range [0~5000])	Adapting to the weights of different brightness scenarios. According to the scene BV brightness, the AWB gain is calculated under two adjacent

			brightness levels of the BV value respectively, and the final gain is obtained by weighting the AWB gain according to the weight.
Core Parameters	lvWeight (5 integers)	The larger the weight value, the more important the AWB correction of the corresponding brightness gear	Adapting weights for different brightness scenarios. Based on the scene BV brightness, AWB gain is calculated for two adjacent brightness levels of the BV value, and the final gain is derived by weighting the AWB gains according to the set weights.

2. Brightness Weight Switch Button Group

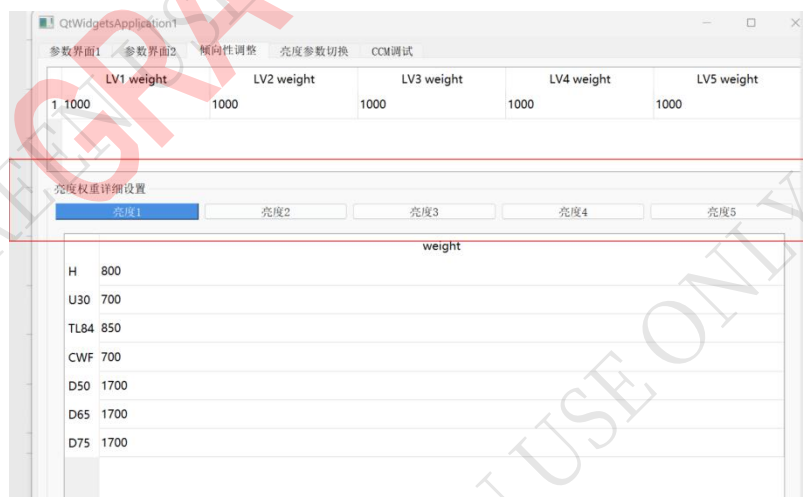


Fig 7 .10 Brightness Weight Switch Button Group

Component Name	Function Description	Debugging Significance

Brightness 1 ~ Brightness 5 Buttons	Clicking the button to switch and display the detailed light source weight table of the corresponding brightness gear	Supporting independent debugging by brightness, avoiding mutual interference of parameters in different brightness scenarios
-------------------------------------	---	--

3. Detailed Light Source Weight Table (the bottom table of the interface, used for mixed light source debugging)

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	7 rows (number of light sources) × 1 column (weight value)	Rows: 7 light sources including H, U30, TL84, CWF, D50, D65, D75; Columns: weight (weight value, range [0~5000]) Configuring the weight of each light source under the current brightness gear	For mixed light sources, each typical light source has a weight. In mixed light scenarios, the higher the weight of a light source, the closer the color effect is to that under the light source.
Core Parameters	cctWeightTable (5×7 integer matrix, 5 brightness levels × 7 light sources)	For mixed light sources, each typical light source has a weight. In mixed light scenarios, the higher the weight of a light source, the closer the color effect is to that under the light	Realizing the refined adaptation of the AWB algorithm and improving the color restoration accuracy in complex scenarios

		source	
--	--	--------	--

7.2.4. Brightness Parameter Switch (AWB Tendency Fine Adjustment)

This tab is the core optimization link of AWB debugging. By adjusting the shiftRG/shiftBG matrix, color cast correction under different brightness-RG-BG combinations is realized, solving the problems of "slight color cast still exists after basic parameter configuration" or "small-range modification of subjective color tendency".

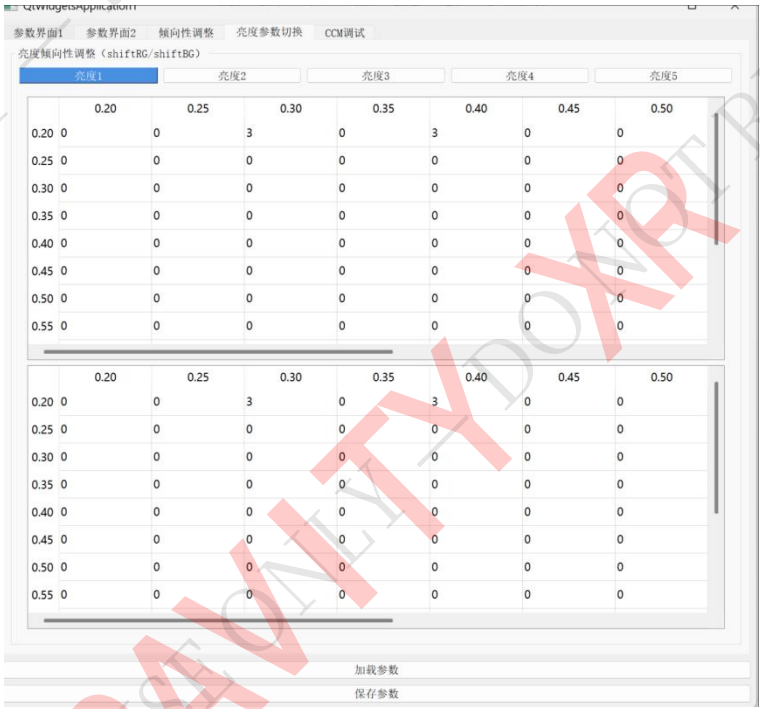


Fig 7.11 Brightness Parameter Switch

1. Brightness Switch Button Group

Component Name	Function Description	Debugging Significance
Brightness 1 ~ Brightness 5 Buttons	Linked with the button function of the "Tendency Adjustment" tab to switch the current debugging brightness gear	Supporting independent adjustment of the shift matrix by brightness to ensure that the color cast of each brightness gear can be accurately corrected

2. shiftRG Matrix Table (the top table of the interface)

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of Rows/Columns	13 rows $\times$ 13 columns (shiftLevel=13)	Row and column coordinates: corresponding to 13 gears of R/G and B/G values (range 0.2~0.85, step 0.05). The vertical axis represents r/g, the horizontal axis represents b/g, and the value scaler = 100 Configuring the R/G value offset (range [-200~200]) under different R/G and B/G values at the current brightness level	Correcting the color cast for a specific range of R/G and B/G values (e.g., if the image is reddish when R/G=0.5 and B/G=0.5 at a certain brightness level, the offset of the corresponding cell can be reduced)
Core Parameters	shiftRG (5 $\times$ 13 $\times$ 13 integer matrix, 5 brightness levels $\times$ 13 $\times$ 13 gears)	The cell value of the table is the offset, a positive number means increasing the R channel gain, and a negative number means decreasing the R channel gain	Correcting the color cast for a specific range of R/G and B/G values (e.g., if the image is reddish when R/G=0.5 and B/G=0.5 at a certain brightness level, the offset of the corresponding cell can be reduced)

3. ShiftBG Matrix Table (the bottom table of the interface)

Table Attribute	Corresponding Parameter	Function Description	Debugging Significance
Number of	13 rows $\times$ 13	Consistent with	Cooperating with

Rows/Columns	columns (shiftLevel=13)	the above Configuring the B/G value offset (range [- 200~200]) under different R/G and B/G values at the current brightness level	the shiftRG matrix to realize the color cast coverage of the entire R/G- B/G space
Core Parameters	shiftBG (5×13×13 integer matrix, 5 brightness levels × 13×13 gears)	The cell value of the table is the offset, a positive number means increasing the B channel gain, and a negative number means decreasing the B channel gain	Cooperating with the shiftRG matrix to realize the color cast coverage of the entire R/G- B/G space

7.2.5. CCM Debugging (Color Correction Matrix Configuration)

This tab focuses on the core parameter configuration of the CCM algorithm. By adjusting the 3×3 matrix, the spectral response difference between the sensor and the human visual system is corrected, which is a key link for color restoration.

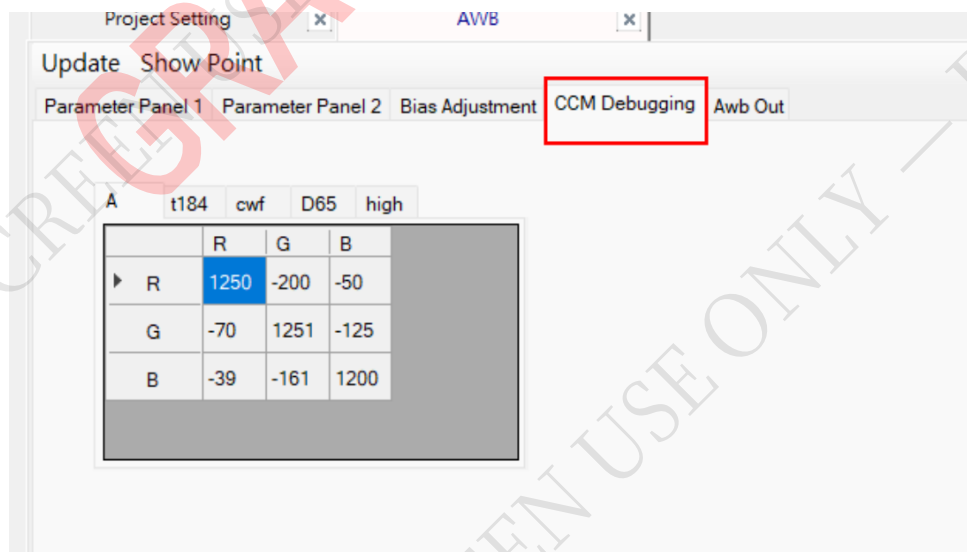


Fig 7.12 CCM Debugging

1. CCM Matrix Group (5 3×3 matrices)

Component Name	Corresponding Parameter	Function Description	Debugging Significance
Matrix Grouping	5 GroupBoxes, corresponding to 5 light sources including A, TL84, CWF, D65 and high respectively	Each GroupBox contains a 3×3 table, with both rows and columns representing the R, G and B channels. Parameter scaler = 1000 Storing the color correction matrix corresponding to each light source, with the matrix element range [-4000~8000]	Solving the problem of "color cast still exists after AWB correction" (e.g., if the sensor has an over-strong response to red, the main diagonal element of the R channel can be adjusted); adapting to the 24-color chart calibration process and optimizing the color restoration accuracy
Core Parameters	ccmParams (5×9 integer matrix, 5 light sources × 9 matrix elements)	Matrix operation logic: Corrected RGB = Original RGB × CCM Matrix (following the white balance constraint to ensure that the white color after AWB correction remains white after CCM processing)	Solving the problem of "color cast still exists after AWB correction" (e.g., if the sensor has an over-strong response to red, the main diagonal element of the R channel can be adjusted); adapting to the 24-color chart calibration process and optimizing the color restoration accuracy

2. Matrix Parameter Adjustment Rules (refer to the core logic of the document)

Color Cast Problem	Adjustment Object	Example Operation
Blue with purple tint	Reduce C13 (B→R cross term)	Change C13 of the A light source matrix from -50 to -80
Green with yellow tint	Reduce C12 (G→R cross term)	Change C12 of the TL84 light source matrix from -70 to -100
Insufficient red saturation	Increase C11 (R main diagonal term)	Change C11 of the D65 light source matrix from 1251 to 1300
Skin tone with yellow tint	Increase C32 (G→B cross term)	Change C32 of the high light source matrix from -161 to -130

Key Parameter Correlation and Debugging Logic

1. AWB Parameter Correlation

The core logic of the AWB algorithm is "statistics based on gray area pixels → color cast calculation → correction through weight and shift matrix", with the parameter correlation as follows:

- 1. Basic Layer: lightSourceTable (light source reference) → Defining the R/G and B/G values and offsets of the "ideal white";
- 2. Screening Layer: greyRegions (gray area), colorRegions (color region), misColorRegions (misleading color region) → Screening effective gray point pixels;
- 3. Optimization Layer: lvWeight (brightness weight), cctWeightTable (detailed light source weight) → Adjusting the response priority of different scenarios;
- 4. Fine-tuning Layer: shiftRG/shiftBG (offset matrix) → Correcting residual color cast.

2. Collaborative Logic of CCM and AWB

- 1. Execution Order: AWB correction is performed first, followed by CCM correction (CCM acts on the linear RGB data after AWB processing);

- 2. Constraint Relationship: The CCM matrix follows the white balance constraint (ensuring that the white color after AWB correction remains white after CCM processing);
- 3. Debugging Order: Complete AWB debugging first (ensuring no obvious color cast), then perform CCM debugging (optimizing color restoration accuracy).

Standard Debugging Process (Step-by-Step Debugging from Scratch)

1. AWB Debugging Steps

- 1. Initialize Parameters: Open the tool and confirm that PARAM_VERSION and DEBUG_LEVEL are the current debugging versions;
- 2. Configure Light Source Reference: Enter the R/G and B/G reference values of the seven light sources in the light source table of "Parameter Interface 1" (can be obtained from calibration data);
- 3. Define Gray Area Boundary: Adjust the boundary coordinates of the greyRegions table according to image features to ensure that the gray area can cover most gray pixels;
- 4. Exclude Interfering Colors: In "Parameter Interface 2", define color regions and misleading color regions, debug color objects, and eliminate the interference of easily confused colors;
- 5. Adjust Brightness Weight: In "Tendency Adjustment", set lvWeight and cctWeightTable according to the target scene (e.g., indoor, outdoor, mixed light);
- 6. Fine-tune Shift Matrix: In "Brightness Parameter Switch", adjust the shiftRG/shiftBG matrix for each brightness gear to correct residual color cast;
- 7. Save Parameters: Click the "Save Parameters" button to generate a .bin file for subsequent verification.

2. CCM Debugging Steps

- 1. Precondition: Complete AWB debugging to ensure no obvious color cast in the image (otherwise CCM cannot perform accurate correction);
- 2. Collect Calibration Images: Collect 24-color chart images with the sensor under the target light source (e.g., TL84, D65);
- 3. Test Color Cast: Test the color deviation of the 24-color chart with tools such as Imatest to determine the color channels that need to be adjusted;
- 4. Adjust Matrix Parameters: In the "CCM Debugging" tab, adjust the corresponding elements for the matrix of the target light source according to the color cast problem;

- 5. Verify Effect: Reload the parameters and test the 24-color chart image, and iterate the adjustment until the average color difference meets the requirements;
- 6. Save Parameters: Update PARAM_VERSION synchronously and save the .bin file.

Explanation of Parameter Loading and Saving

1. Load Parameters

- 1. Click the "Load Parameters" button at the bottom of the tool and select an existing .bin parameter file;
- 2. The tool will automatically read all parameters in the file, including version parameters, AWB table parameters and CCM matrix parameters, and synchronize them to the corresponding interface components;
- 3. After successful loading, a "Parameter Loading Completed" prompt will pop up, and adjustments can be made directly based on the existing parameters.

2. Save Parameters

- 1. After debugging, click the "Save Parameters" button and select the save path and file name (the default name is AWBParam.bin);
- 2. Before saving, the tool will automatically verify the input format of PARAM_VERSION and DEBUG_LEVEL (must be 3 integers separated by commas), and a prompt will pop up if the format is wrong;
- 3. After passing the verification, the tool will write all parameters into the .bin file in a fixed order, including:
 - AWB Basic Parameters → Color Region Parameters → Weight Parameters → Shift Matrix Parameters → Version Parameters → CCM Matrix Parameters;
- 4. After successful saving, a "Parameter Saving Completed" prompt will pop up, and the file can be used for subsequent loading or deployment to the ISP chip.

Appendix

Algorithm Parameter Table

Parameter Name	Introduction
colorRegionsCount	Number of color range settings
misColorRegionsCount	Number of misleading color

	region settings
CCTLevel	Number of color temperature gears
LightLevel	Number of brightness levels
lightSourceNum	Number of light sources used for calibration
ccmNum	Number of CCM matrices
shiftLevel	Number of levels representing tendency adjustment
AWB_HW3A_BLOCK_IN FO_SIZE	Number of bits for storing statistical information
AWB_BLOCK_XCNT	max size 64
AWB_BLOCK_YCNT	max size 48
AWB_HW3A_PER_BLOCK K_SIZE	Number of storage bits per statistical block
AWB_BLOCK_SIZE	$(AWB_BLOCK_XCNT * AWB_BLOCK_YCNT)$
AWB_HW3A_PATCH_W	Width of the statistical block
AWB_HW3A_PATCH_H	Height of the statistical block
AWB_HW3A_PATCH_SIZE	$(AWB_HW3A_PATCH_W * AWB_HW3A_PATCH_H)$
AWB_HW3A_PATCH_W_	max block width

MAX	
AWB_HW3A_RATIO_MAX	RG/BG(x100) RATIO MAX VALUE
AWB_STATS_RG_SAT	rg saturation lower limit
AWB_STATS_BG_SAT	bg saturation lower limit
AWB_LS_SHIFT_TABLE_BV_PARA0	Brightness level boundary value for gear division of the tendency adjustment table
AWB_LS_SHIFT_TABLE_CT_PARA0	RG\BG level boundary value for gear division of the tendency adjustment table
LV1_weight	Weight of image brightness gear 1
LV2_weight	Weight of image brightness gear 2
LV3_weight	Weight of image brightness gear 3
LV4_weight	Weight of image brightness gear 4
LV5_weight	Weight of image brightness gear 5
LV1_H_weight	Weight of light source H under image brightness gear 1
LV1_U30_weight	Weight of light source U30 under image brightness gear 1

LV1_TL84_weight	Weight of light source TL84 under image brightness gear 1
LV1_CWF_weight	Weight of light source CWF under image brightness gear 1
LV1_50_weight	Weight of light source D50 under image brightness gear 1
LV1_65_weight	Weight of light source D65 under image brightness gear 1
LV1_75_weight	Weight of light source D75 under image brightness gear 1
LV2_H_weight	Weight of light source H under image brightness gear 2
LV2_U30_weight	Weight of light source U30 under image brightness gear 2
LV2_TL84_weight	Weight of light source TL84 under image brightness gear 2
LV2_CWF_weight	Weight of light source CWF under image brightness gear 2
LV2_50_weight	Weight of light source D50 under image brightness gear 2
LV2_65_weight	Weight of light source D65 under image brightness gear 2

	2
LV2_75_weight	Weight of light source D75 under image brightness gear 2
LV3_H_weight	Weight of light source H under image brightness gear 3
LV3_U30_weight	Weight of light source U30 under image brightness gear 3
LV3_TL84_weight	Weight of light source TL84 under image brightness gear 3
LV3_CWF_weight	Weight of light source CWF under image brightness gear 3
LV3_50_weight	Weight of light source D50 under image brightness gear 3
LV3_65_weight	Weight of light source D65 under image brightness gear 3
LV3_75_weight	Weight of light source D75 under image brightness gear 3
LV4_H_weight	Weight of light source H under image brightness gear 4
LV4_U30_weight	Weight of light source U30 under image brightness gear 4

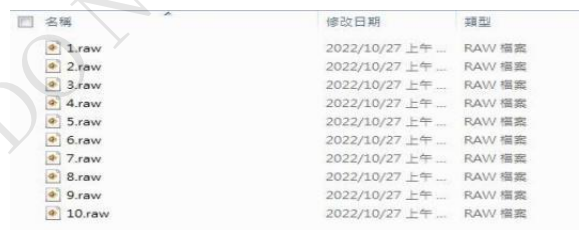
LV4_TL84_weight	Weight of light source TL84 under image brightness gear 4
LV4_CWF_weight	Weight of light source CWF under image brightness gear 4
LV4_50_weight	Weight of light source D50 under image brightness gear 4
LV4_65_weight	Weight of light source D65 under image brightness gear 4
LV4_75_weight	Weight of light source D75 under image brightness gear 4
LV5_H_weight	Weight of light source H under image brightness gear 5
LV5_U30_weight	Weight of light source U30 under image brightness gear 5
LV5_TL84_weight	Weight of light source TL84 under image brightness gear 5

8. Process a batch of files

There are three kinds of batch run mode in our tool, a) to run consecutive frames b) to run independent frames c) to run independent data. Fig. 8 .1

8.1. Create Setting File

Refer to fig. 8.1 create a folder and put all your images that need to be simulated in it.



名稱	修改日期	類型
1.raw	2022/10/27 上午...	RAW 檔
2.raw	2022/10/27 上午...	RAW 檔
3.raw	2022/10/27 上午...	RAW 檔
4.raw	2022/10/27 上午...	RAW 檔
5.raw	2022/10/27 上午...	RAW 檔
6.raw	2022/10/27 上午...	RAW 檔
7.raw	2022/10/27 上午...	RAW 檔
8.raw	2022/10/27 上午...	RAW 檔
9.raw	2022/10/27 上午...	RAW 檔
10.raw	2022/10/27 上午...	RAW 檔

Fig 8 .1

8.2. Run Consecutive images and Independent Images

If you want to run TNR function and reference previous frame, you have to choose consecutive images mode. Because it will reference previous frame, it will have cleaner image, refer to fig. 8.2, compared to independent images.

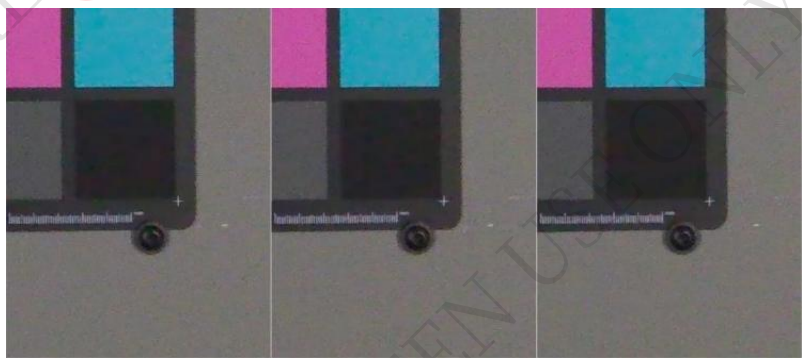


Frame 2

Frame 5
Fig 8 .2

Frame 10

Refer to fig. 8 .3, if you do not want to reference previous frame's TNR result, you can choose independent images. Because it won't reference previous frame, it will have noisier image compared to consecutive images.



Frame 2

Frame 5
Fig 8.3

Frame 10

8.3. Run Independent simulate data

Refer to fig. 8.4, If you want to tune functions after TNR, you have to generate simulate data at first. Choose simulate data check box and run batch file.

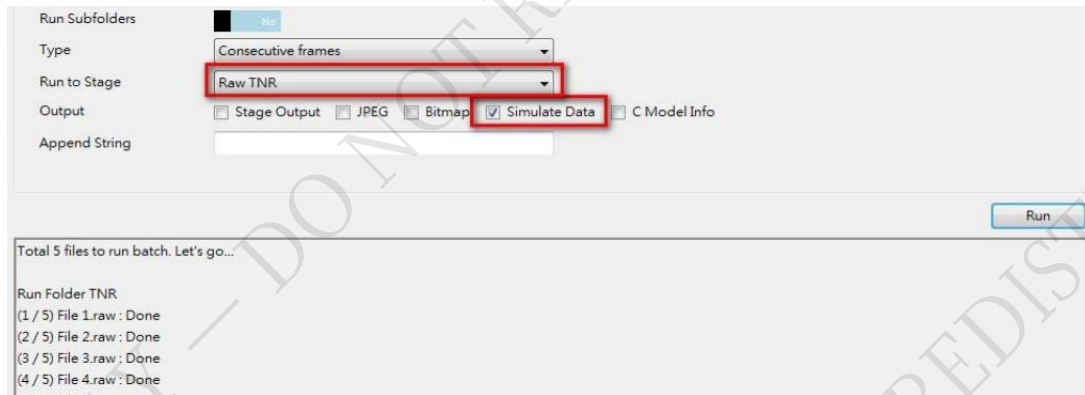


Fig 8 .4

Load simulation data path and then choose independent simulation data. Press “run” button to see batch run results.

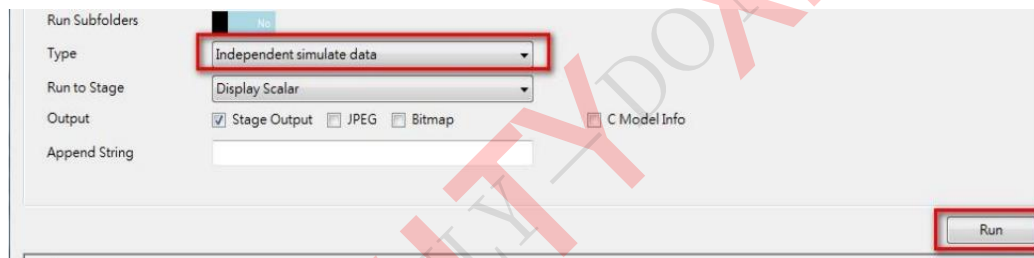


Fig 8 .5

8.4. Generate Stage Output

We can choose to run stage output through batch run. First, choose batch run mode. Then, choose pipeline stage output.