CONFIDENTAL B

MT6771
AWB Introduction

Revision

Revision	Date	Description
V1.0	2018.02	Initial version

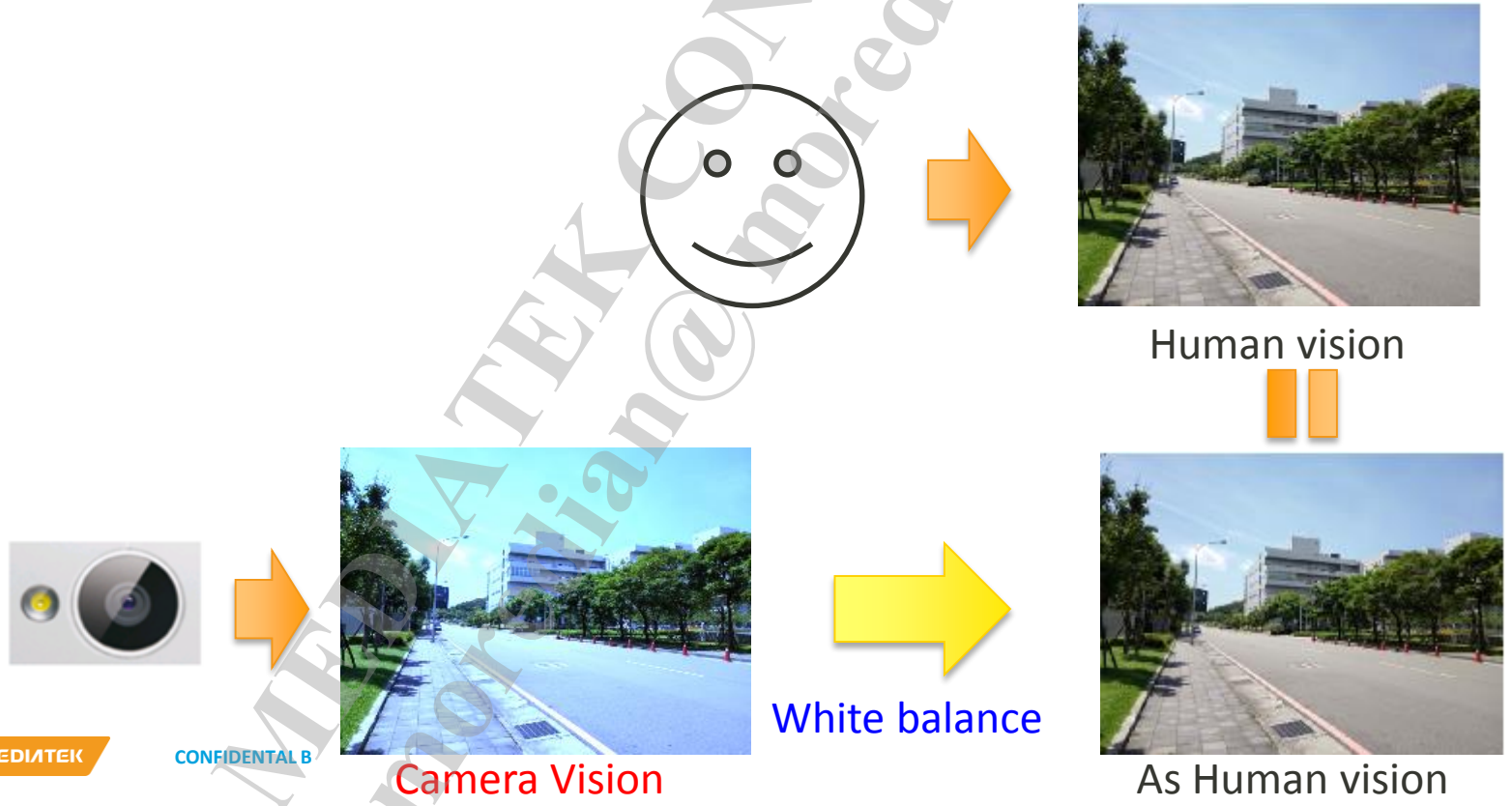
Outline

- Introduction
- AWB
  - {A} HW Statistic
  - {B} Light Source Estimation
  - {C} Predict AWB Gain by Light Source Info.
  - {D} Spatial & Temporal Predictor
  - {E} Get AWB Output Gain
  - {F} Others
- Tuning Parameters
- Debug Parser Tag

Introduction

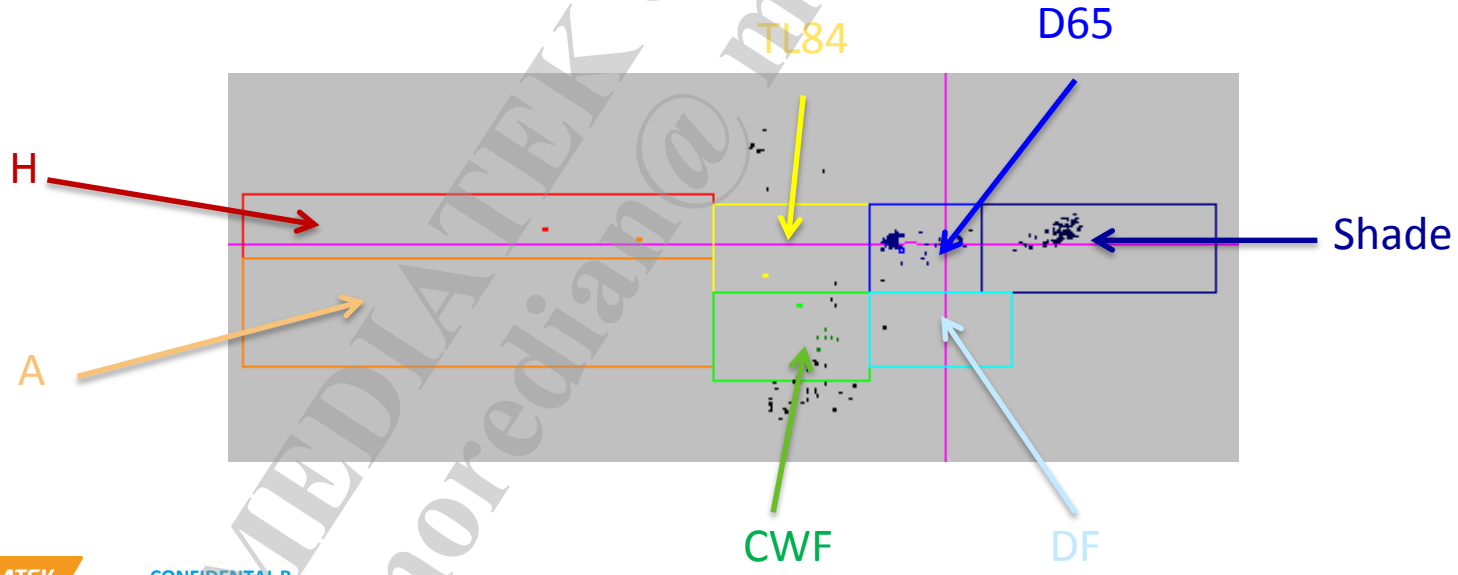
Introduction (1/2)

- What is AWB ?
  - Vision of digital camera is not like human vision on color.
  - Auto white balance is performed to modify image color as human vision.



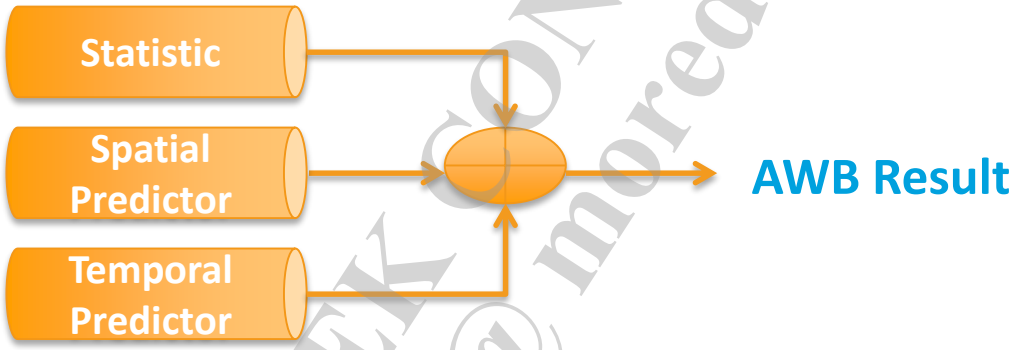
Introduction (2/2)

- How to do AWB
  - Define balance light source color region
  - AWB will balance color fall in this region
  - If there are multi light sources, AWB will weighting each light source and balance it.



Architecture Overview

- Block Diagram
  - Basic flow
    - Statistic Gain + Spatial Predictor + Temporal Predictor



  - Special condition
    - Feature Detect

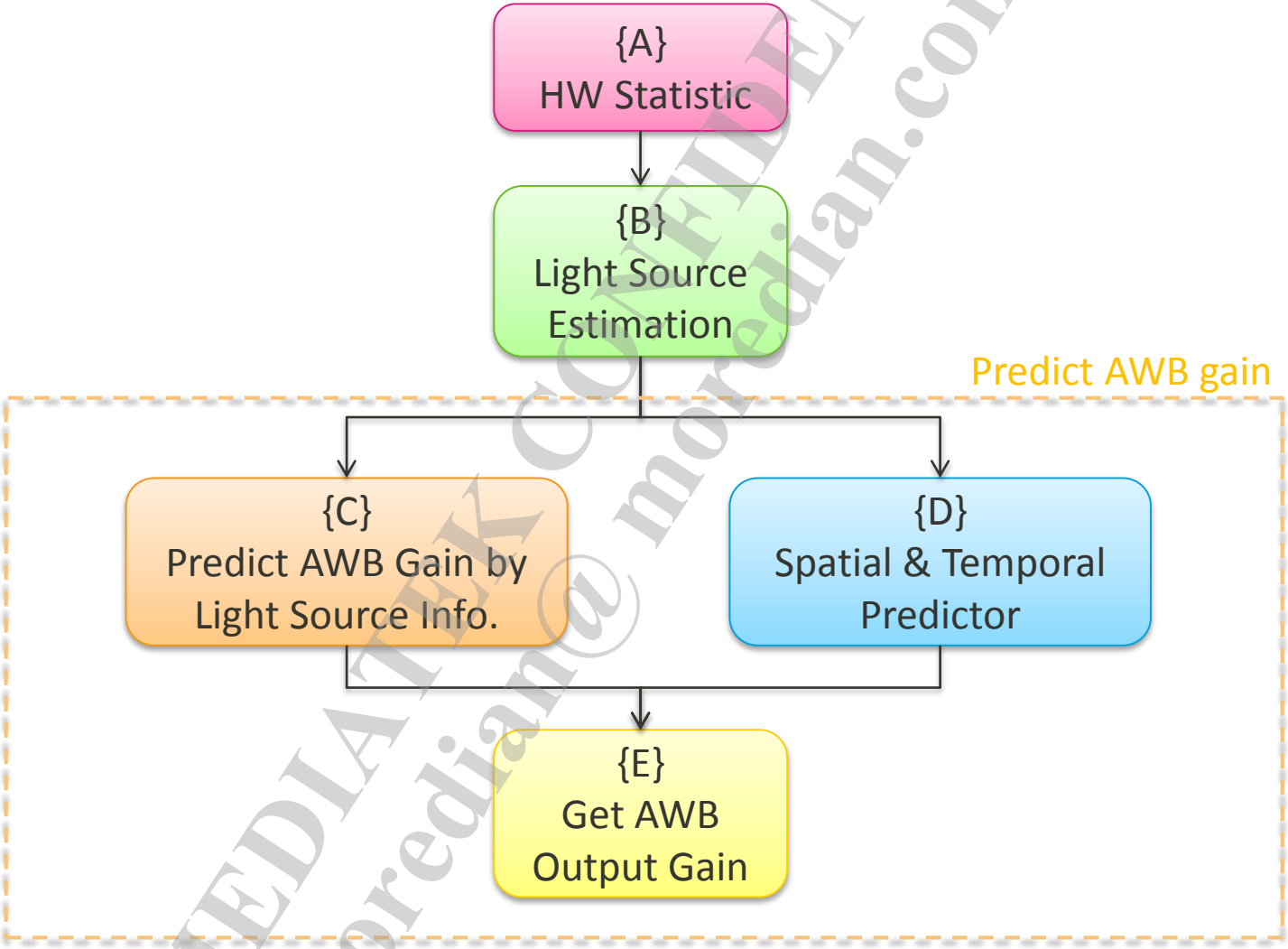
Terms

- **Statistic Gain**
  - Definition: AWB gain got from statistic white point
  - Usage: The base of AWB algorithm for getting correct result in a scene with enough white points
- **Spatial Predictor**
  - Definition: The mechanism to get AWB Default Gain by referring to the ambient brightness
  - Usage: When ambient brightness is high or there is no white point to refer to, Spatial Predictor can replace AWB Statistic to enhance the correctness of AWB.

Terms

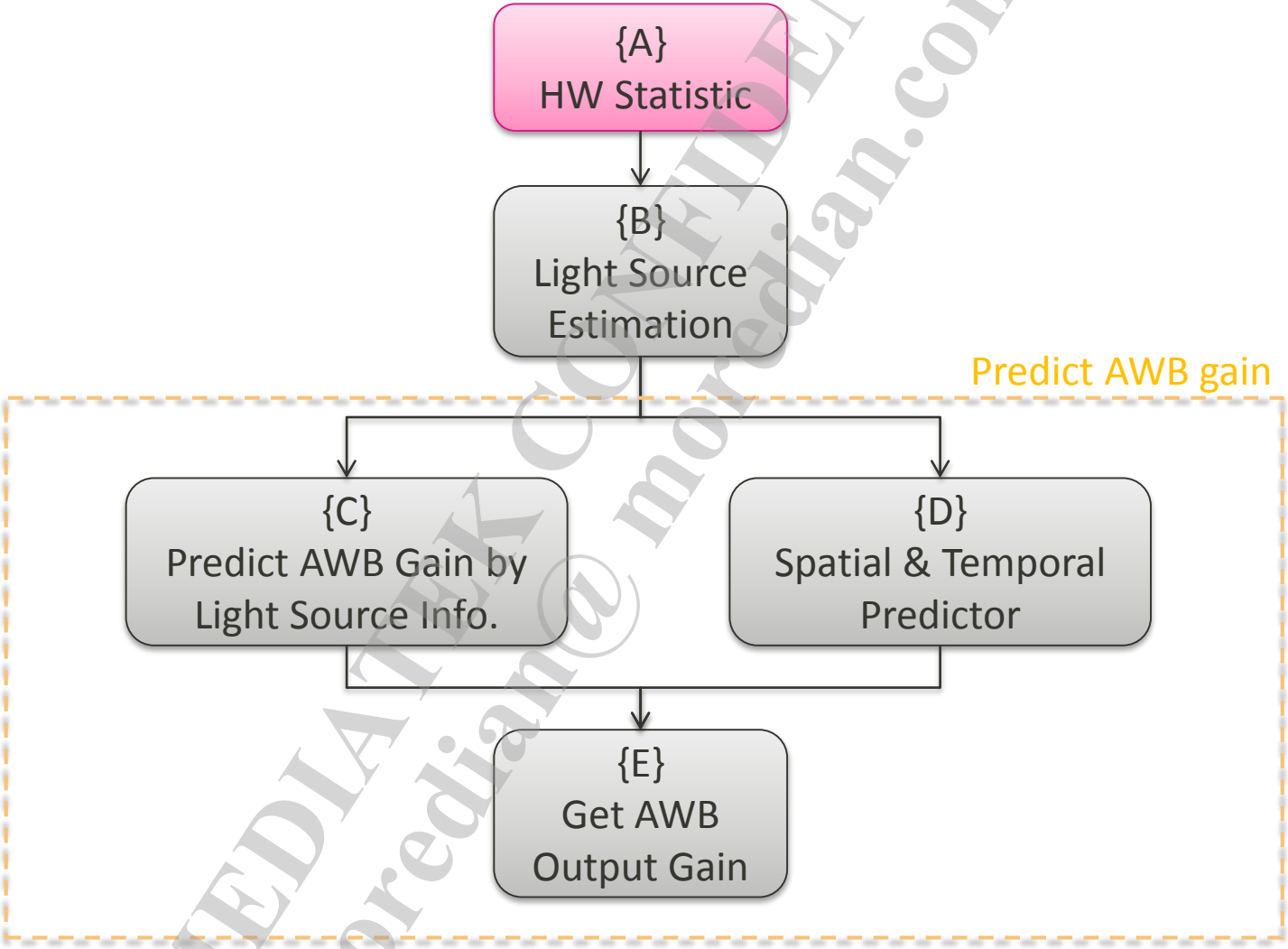
- **Temporal Predictor**
  - Definition: The mechanism to get AWB Gain by referring to the AWB results of the past four suitable frames
  - Usage: When there is no white point to refer to, Temporal Predictor can replace AWB Statistic to enhance the correctness of AWB.
- **Feature Detection**
  - Definition: The extra small light source window in the AWB standard light source window
  - Usage: For supplementing AWB statistic; under all standard light sources, processing special cases which meet AWB light source conditions

AWB Flow



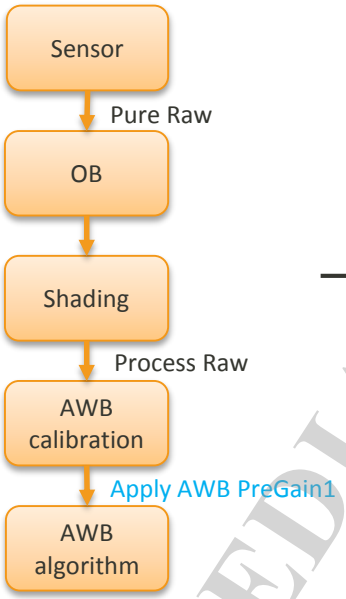
{A} HW Statistic

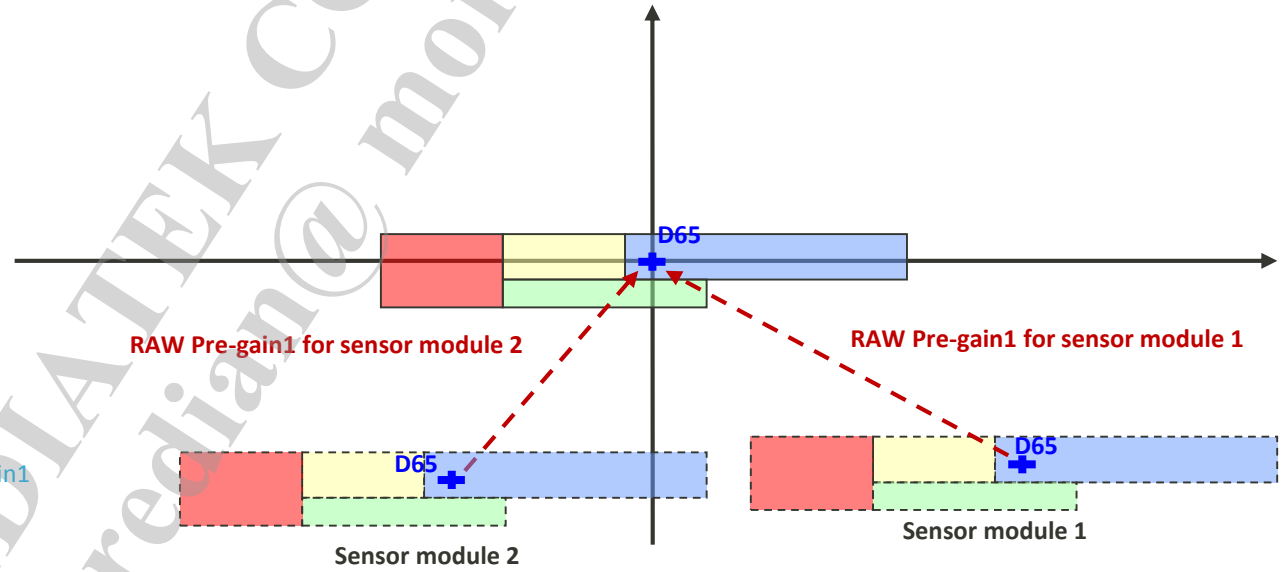
AWB Flow



HW Statistic

- **RAW Pre-Gain 1**
  - When MTK AWB Calibration is adopted, for different units,
    - AWB PreGain1 will be different.
    - AWB Light Source Window Area will be the same.
      - MTK AWB Calibration eliminates unit variance by changing Sensor Response instead of changing the Light Source Window Area of the unit.

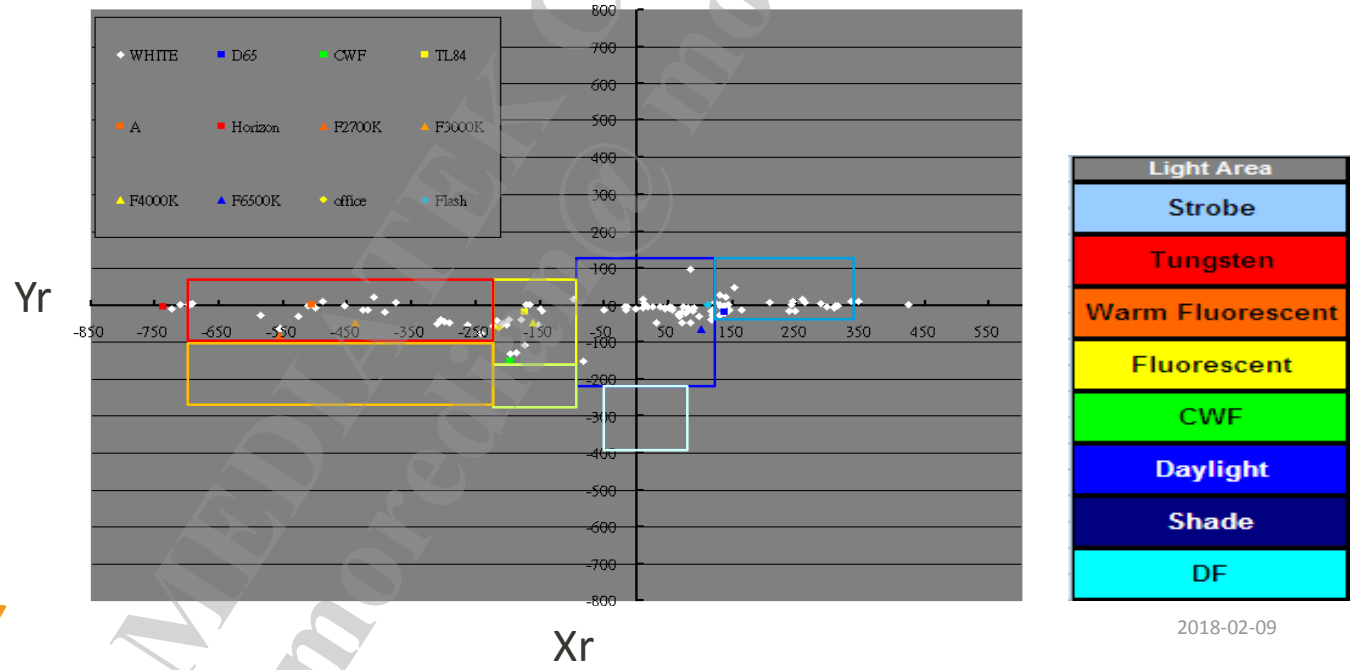




HW Statistic

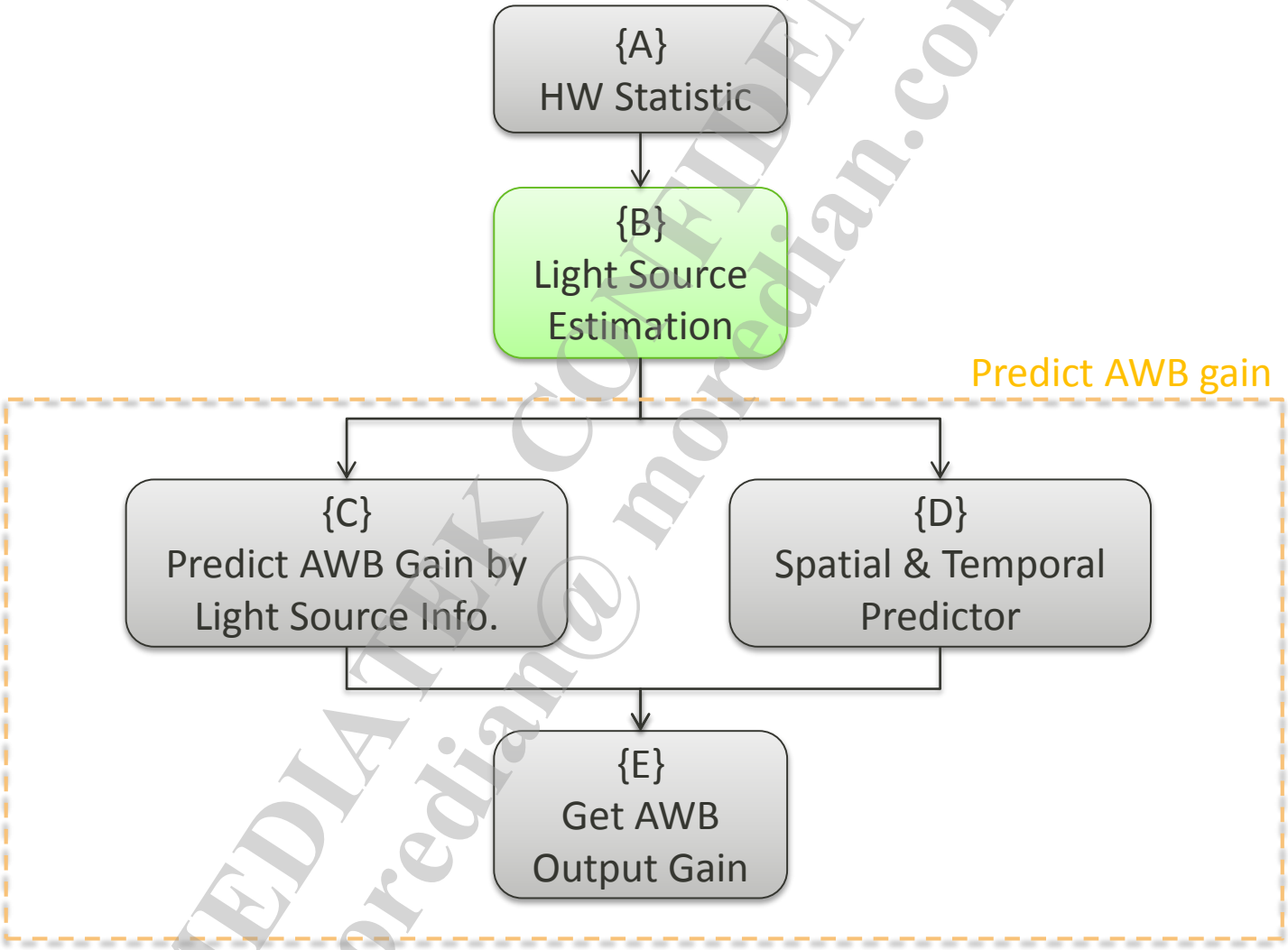
- **Color Space Conversion**

- **Light Source Estimation**
  - Estimate all Main stat. block location to know light source



{B} Light Source Estimation

AWB Flow

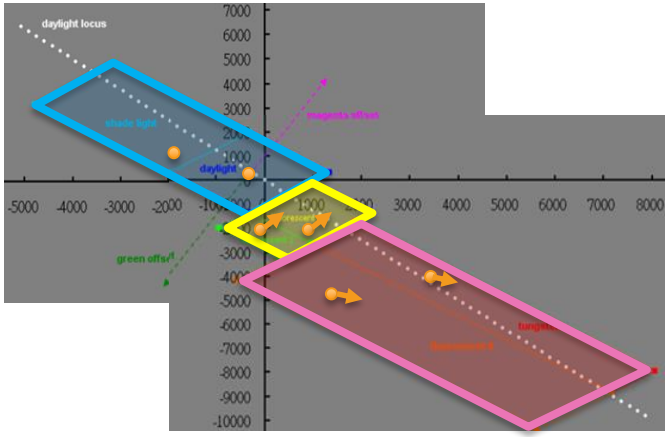


Three Illuminants

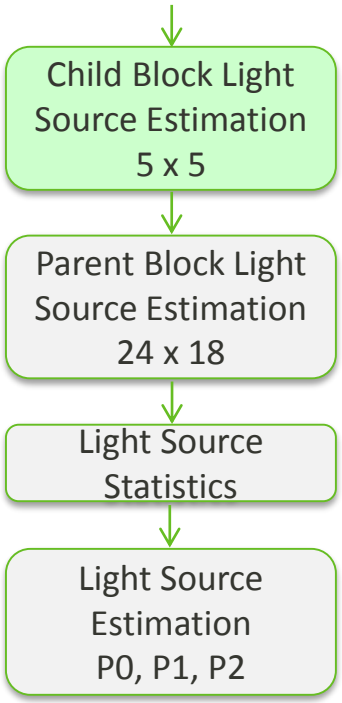
- 120*90 CB statistic data is compensated by HW
  - One-illuminant calibration: Pregain1
  - Multi-illuminant calibration : Pregain1, Pregain1M, Pregain1L depend on light index
    - 2 illuminants : 5000K and 3000K
    - 3 illuminants: >5000K, TL84, <3000K

SW: $stat_{awb}$ * pregain1M / pregain1

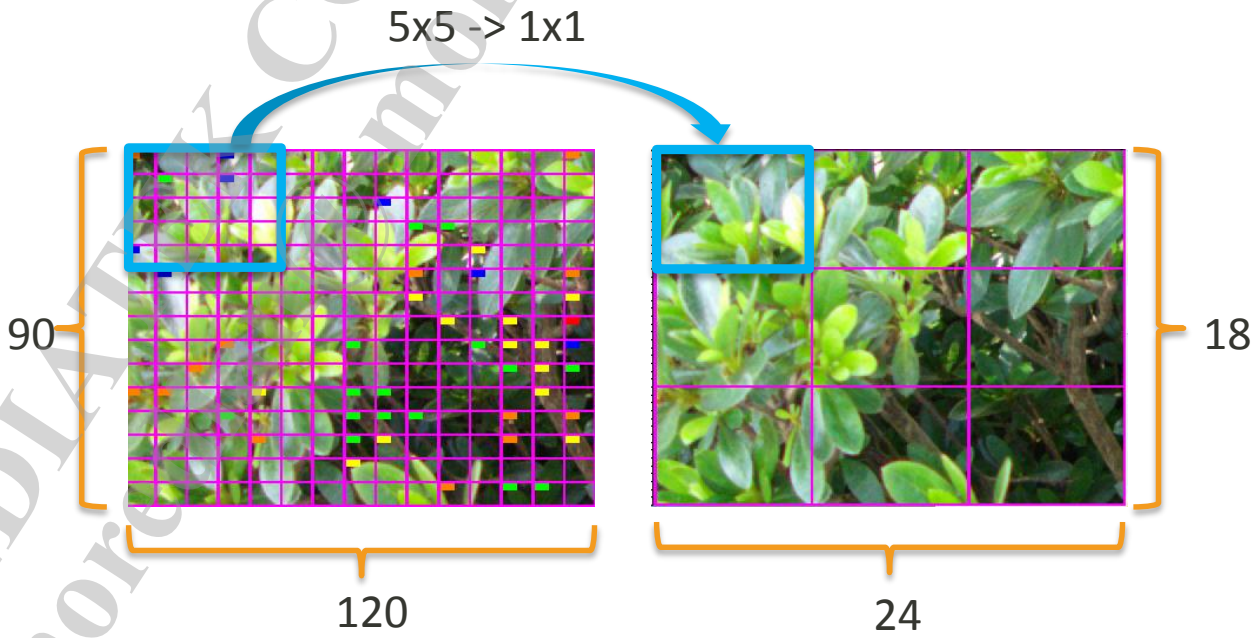
LitIdx	Light Source	1 illuminant	2 illuminant	3 illuminant
0	Strobe	pregain1	pregain1	pregain1
1	T	pregain1	pregain1L	pregain1L
2	WF	pregain1	pregain1L	pregain1L
3	F	pregain1	(pregain1+ pregain1L)/2	pregain1M
4	CWF	pregain1	(pregain1+ pregain1L)/2	pregain1M
5	Daylight	pregain1	pregain1	pregain1
6	Shade	pregain1	pregain1	pregain1
7	DF	pregain1	pregain1	pregain1



Child Block Light Source Estimation

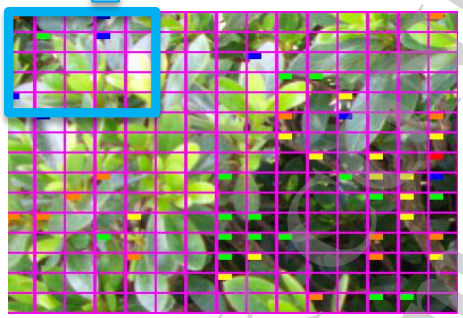


- Parent block number = **24x18**
- Children block number = 5x5 in each parent block
  - Total blocks = (24x5) x (18x5) = 120x90 windows



Parent Block Light Source Estimation

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2



6	E	5	4	4
4	4	4	5	7
4	4	4	4	5
5	4	4	4	4
5	4	4	4	4

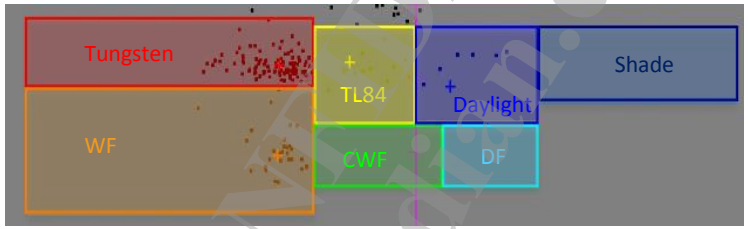
0: Strobe → 0
1: Tungsten → 0
2: Warm Fluorescent → 0
3: Fluorescent → 0
4: CWF → 17 ….. Dominate!
5: Daylight → 5
6: Shade → 1
7: Daylight Fluorescent → 1
Error → 1

AWB
v5.0

Light Source Statistics

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

Separate 24x18 statistic data into 7 light sources:



WeightedSumR[Light] += SumR[i] x Weight[i]
WeightedSumG[Light] += SumG[i] x Weight[i]
WeightedSumB[Light] += SumB[i] x Weight[i]

PB: 24*18

The weighting for each Parent Blocks has 3 modes (24x18)

- Mode1 : Linear weighting
- Mode2 : Weighting LUT for different color temperature
- Mode3 : Weighting LUT for different LV

AWB
v5.0

Light Source Statistics

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

- Parent Block Weight
  - Mode 1 (Original)

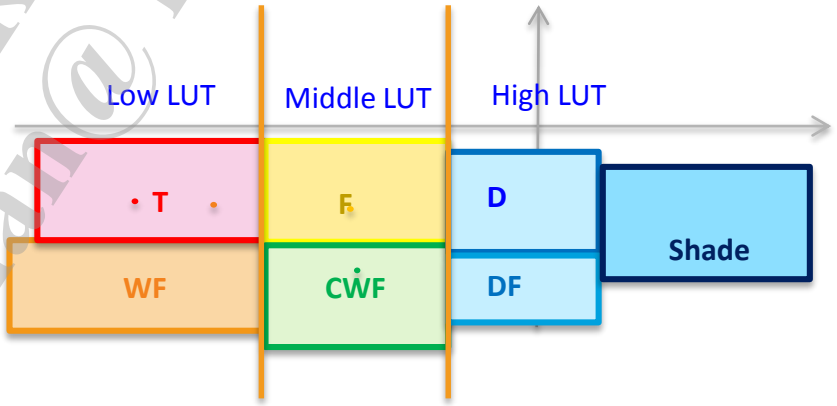
$$Weight = 1 + \frac{(R_{AVG} + G_{AVG} + B_{AVG})}{2^{factor}}$$

High luminance dominate

Scaling Factor	Weight
6	1~12
7	1~6
8	1~3
9	1~2
10	1

factor could be set from 6~10 (default value is 6)

  - Mode 2 (LUT by different color temperature)

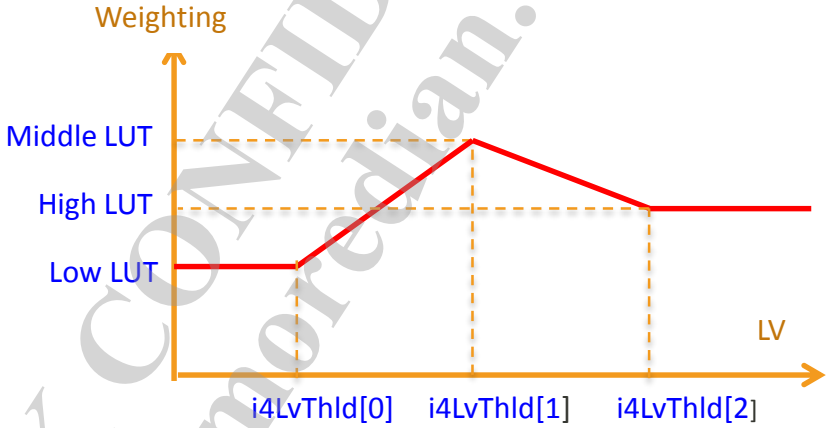


AWB
v5.0

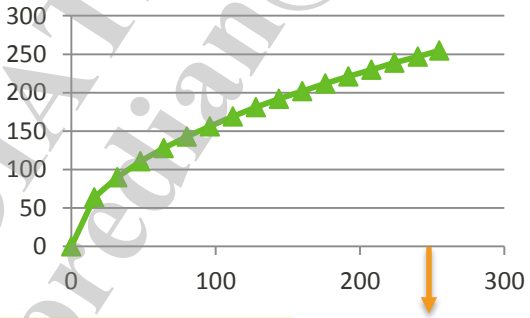
Light Source Statistics

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

- Parent Block Weight
  - Mode 3 (LUT by LV)

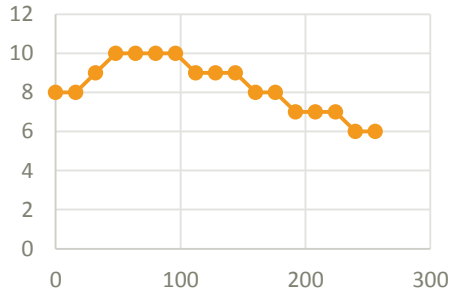


  - mode2 & 3 use Gamma LUT transfer to vision linear domain and then look up weighting by WeightLUT



i4GammaLut

$(R_{AVG} + G_{AVG} + B_{AVG})$



i4WeightLut

AWB
v5.0

Light Source Statistics

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

- Parent Block Weight

```
// Parent block weight parameter
{
    2, // bEnable
    8, // i4ScalingFactor: [6] 1~12, [7] 1~6, [8] 1~3, [9] 1~2, [>=10]: 1
    {20, 70, 120}, // i4LvThld[3]
    // Gamma LUT
    {0, 64, 90, 111, 128, 143, 156, 169, 181, 192, 202, 212, 221, 230, 239, 247, 256},
    // Weighting LUT for High Mid Low color temperature
    {
        { 8, 8, 9, 10, 10, 10, 10, 9, 9, 9, 8, 8, 7, 7, 7, 6, 6}, // Low
        { 8, 8, 9, 10, 10, 10, 10, 9, 9, 9, 8, 8, 7, 7, 7, 6, 6}, // Middle
        { 8, 8, 9, 10, 10, 10, 10, 9, 9, 9, 8, 8, 7, 7, 7, 6, 6} // High
    }
```

Mode selection

Mode1 scale factor

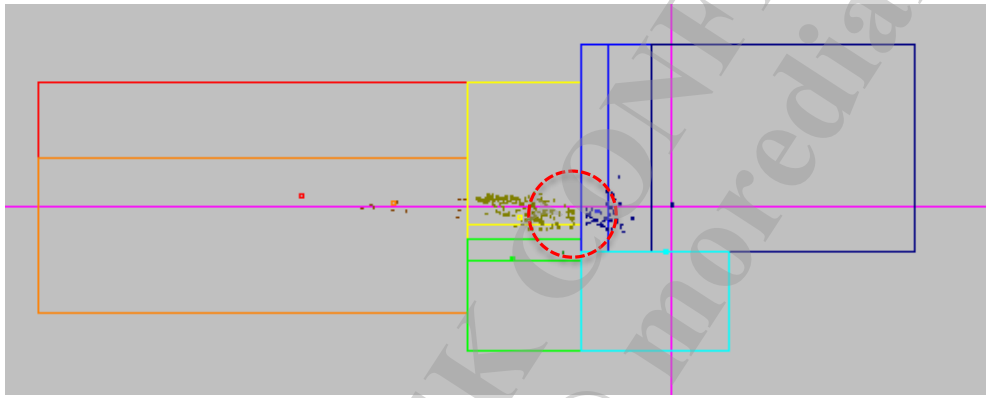
Mode3 LV threshold

Mode2 & 3 used Gamma table

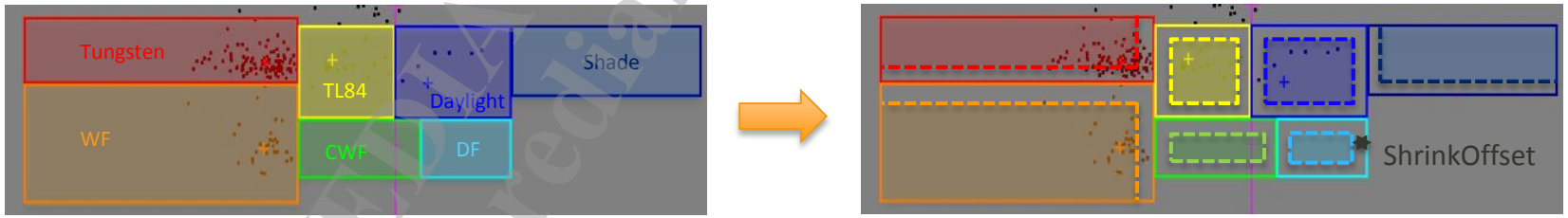
Mode2 & 3 Weight table

Light Source Statistics

- **Statistic Smooth (Shrink Window)**
  - When the statistic data locate on the boundary of AWB windows, it might occur AWB gain not smooth issue.



  - **Purpose:** Improve the AWB result not smooth or consistent



Light Source Statistics

- **Statistic Smooth**
  - Shrink-window offset will be generated automatically according to all light source areas.
  - Recommend use same offset settings for all AWB windows to keep statistic behavior.
  - Distance thresholds are not opened for tuning. FW will decide the Dist. threshold automatically.

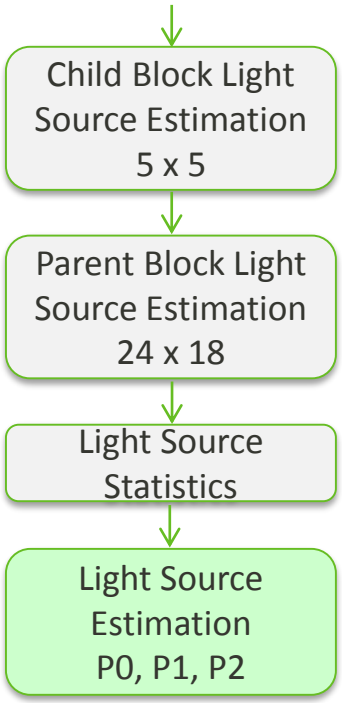
```
// Smooth Statistic
{
    1,    //i4Enable
    {0, 23, 23, 23, 23, 23, 23, 23},  //i4StatWinShrinkOffset
    0,    //Reserved1
    0,    //Reserved2
    0,    //Reserved3
    0,    //Reserved4
    0,    //Reserved5
},
```

Threshold is decided by FW automatically

EXIF

AWB_TAG_SMOOTHSTAT_ENABLE	:	1
AWB_TAG_SMOOTHSTAT_OFFSET_STB	:	0
AWB_TAG_SMOOTHSTAT_OFFSET_T	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_WF	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_F	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_CWF	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_D	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_S	:	23
AWB_TAG_SMOOTHSTAT_OFFSET_DF	:	23
AWB_TAG_SMOOTHSTAT_DIST_THR_STB	:	0
AWB_TAG_SMOOTHSTAT_DIST_THR_T	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_WF	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_F	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_CWF	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_D	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_S	:	46
AWB_TAG_SMOOTHSTAT_DIST_THR_DF	:	46

Light Source Estimation



- Give each light source L_i a Probability $P(L_i)$.
  - If $P(L_i)$ is higher, means that the light source L_i is more important light source in this frame.
  - Otherwise, if $P(L_i)$ is lower, the light source L_i is not the main color temperature now.

- Define $P(L_i)$ combined by 3 types probability:

$P_0(L_i)$: Probability of weighted parent block number

$P_1(L_i)$: Probability of scene luminance level

$P_2(L_i)$: Probability of green/magenta offset

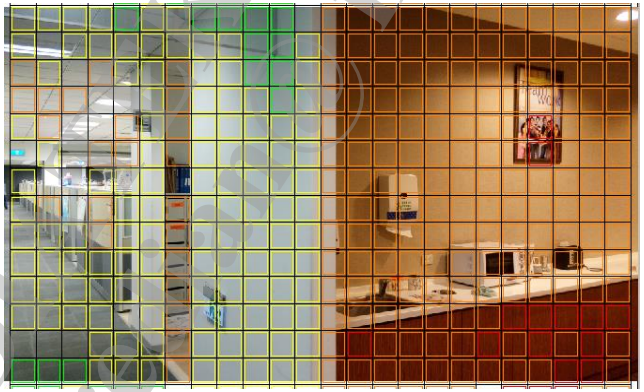
Light Source Estimation – P0 (1/4)

- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

$P_0(L_i)$: Probability of weighted parent block number

$$P_0(L_i) = \frac{WPBN(L_i)}{\sum_j WPBN(L_j)}$$

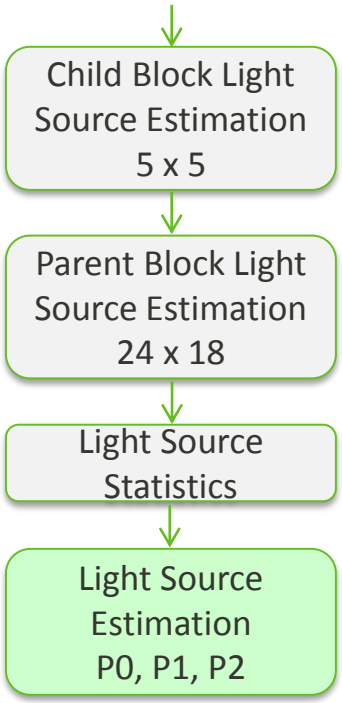
$WPBN(L_i)$: Weighted parent block number of light source candidate L_i.



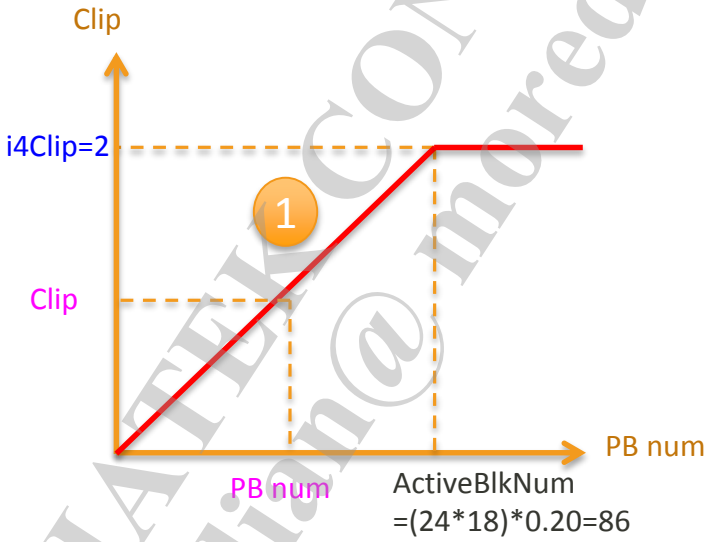
P0(Strobe)	**0**
P0(Tungsten)	**3**
P0(WF)	**50**
P0(Fluorescent)	**41**
P0(CWF)	**6**
P0(Daylight)	**0**
P0(Shade)	**0**
P0(DF)	**0**

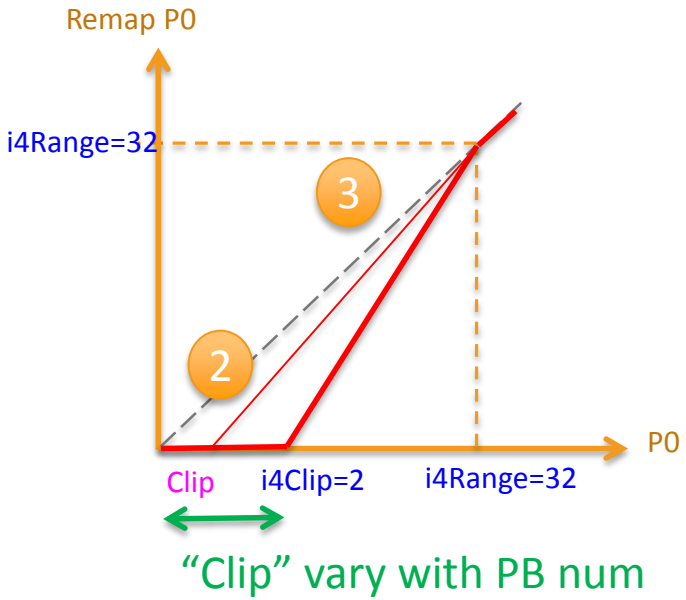
If the parent block number of L_i is larger, and the weight of these parent blocks are also bigger, the $P_0(L_i)$ will be higher!

Light Source Estimation – P0 (2/4)



- **P0 Stability** remapping function
  - Reduce the disturbance from the not dominated light source





PB num = number of all neutral parent blocks

Light Source Estimation – P0 (3/4)

- **P0 Stability**
  - Tuning Parameters

```
// P0 Stability
        {
            1,      //i4Enable
           20,      //i4PbRatio
           32,      //i4Range
            2       //i4Clip
        },
```

用來計算 Active Block Number (BlkThr),
BlkThr = 總ParentBlock數 * i4PbRatio = Round(432*20/100) = 86
如果該光源的NeutralParentBlockNumber 小於86
則重新計算Clip值

P0 remapping的轉折點, 原始P0小於此值會進行P0 remapping

i4Clip(0~10), 用來計算最後的Clip值
Clip = i4Clip * (Current PB Num / Blk_Thr))
1. 如果原始P0 < Clip, P0' = 0
2. 如果原始P0 > Clip, P0' = i4Range * (P0-i4Clip)/(i4Range-i4Clip)

Light Source Estimation – P0 (4/4)

- **P0 Stability**
  - Case : Enable P0 stability to reduce P0

EXIF
AWB_TAG_P0_STB : 0
AWB_TAG_P0_T : 0
AWB_TAG_P0_WF : 2
AWB_TAG_P0_F : 0
AWB_TAG_P0_CWF : 0
AWB_TAG_P0_D : 97
AWB_TAG_P0_S : 1
AWB_TAG_P0_DF : 0

EXIF
AWB_TAG_P0_STB : 0
AWB_TAG_P0_T : 0
AWB_TAG_P0_WF : 0
AWB_TAG_P0_F : 0
AWB_TAG_P0_CWF : 0
AWB_TAG_P0_D : 100
AWB_TAG_P0_S : 0
AWB_TAG_P0_DF : 0

EXIF
WB_TAG_P0_STB : 0
AWB_TAG_P0_T : 1
AWB_TAG_P0_WF : 2
AWB_TAG_P0_F : 5
AWB_TAG_P0_CWF : 0
AWB_TAG_P0_D : 82
AWB_TAG_P0_S : 8
AWB_TAG_P0_DF : 3

Disable

EXIF
AWB_TAG_P0_STB : 0
AWB_TAG_P0_T : 0
AWB_TAG_P0_WF : 0
AWB_TAG_P0_F : 3
AWB_TAG_P0_CWF : 0
AWB_TAG_P0_D : 89
AWB_TAG_P0_S : 7
AWB_TAG_P0_DF : 1

Enable

Light Source Estimation – P1

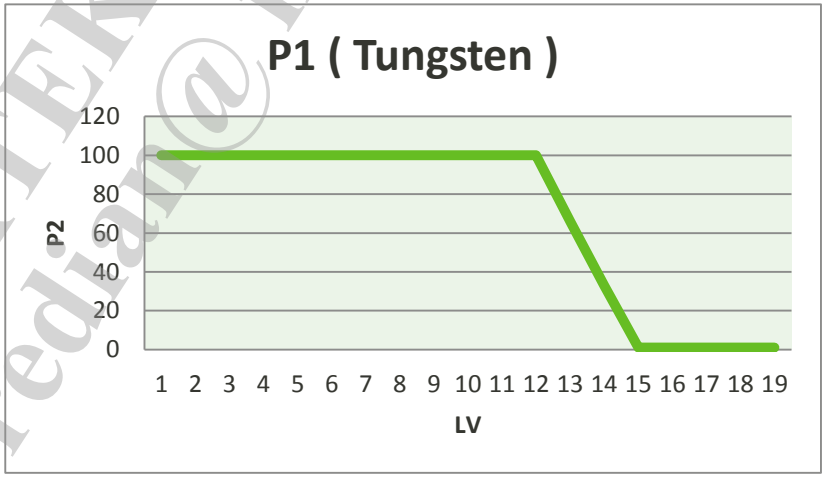
- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

$P_1(L_i)$: Probability of scene luminance level

$$LV = EV - \log_2(ISO\ Speed/100)$$

```
// AWB Light source probability (P1) look-up table (Max: 100; Min: 0)
{                                                                          LV
  //LV0   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
   {100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100}, // Strobe
   {100,  90,  85,  90, 100, 100,  85,  65,  36,  15,  15,  15,   8,   8,   1,   1,   1,   1}, // Tungsten
   {100,  90,  85,  85,  80,  80,  82,  72,  80,  55,  35,  15,   8,   8,   1,   1,   1,   1}, // Warm fluorescent
   {100, 100, 100, 100, 100, 100, 100, 100,  75,  80,  80,  80,  75,  55,  45,  10,   1,   1}, // Fluorescent
   {100, 100,  95, 100, 100, 100,  90, 100,  95,  90,  95,  80,  65,  43,  25,  30,   1,   1}, // CWF
   {100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100}, // Daylight
   {100, 100, 100, 100, 100, 100,  90, 100, 100, 100, 100, 100,  75,  75,  65,  45,  33,   1}, // Shade
   {100, 100, 100, 100, 100, 100, 100,  75,  55,  65,  80,  75,  50,  57,  33,   1,   1,   1}  // Daylight fluorescent
},
```

Light Source



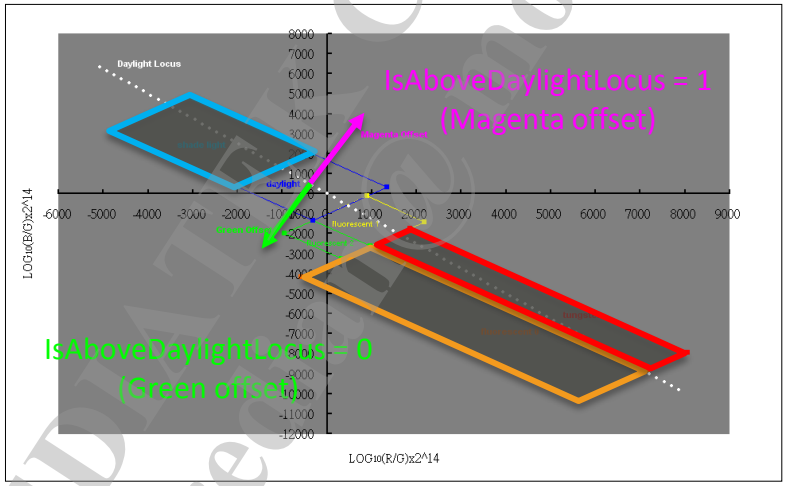
Light Source Estimation – P2 (1/5)

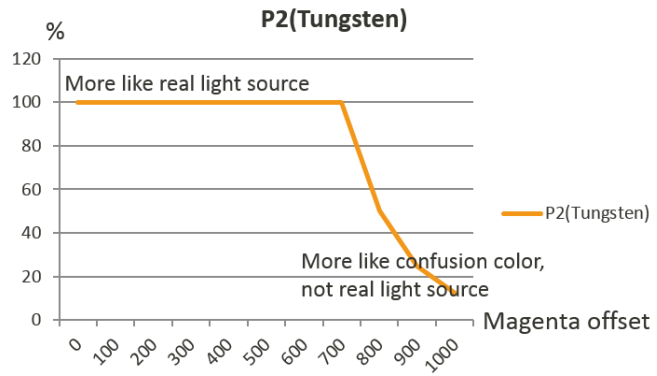
- Child Block Light Source Estimation 5 x 5
- Parent Block Light Source Estimation 24 x 18
- Light Source Statistics
- Light Source Estimation P0, P1, P2

- Tungsten
- WF
- Shade
- Others: 100%

$P_2(L_i)$: Probability of green/magenta offset

Change to log domain for using daylight locus model to check if there is any **Magenta offset** or **Green Offset** in Tungsten, Warm Fluorescent and Shade.





Light Source Estimation – P2 (2/5)

- Tuning Procedure
  - Get Magenta/Green Offset value from EXIF info

EXIF

AWB_TAG_DAY_LOCUS_OFFSET_T : 6072
AWB_TAG_NEW_OFFSET_T : 1175
AWB_TAG_OFFSET_RATIO_T : 50
AWB_TAG_RATIO_OFFSET_T : 952
AWB_TAG_LUT_OFFSET_T : 757
AWB_TAG_IS_ABOVE_DAY_LOCUS_T : **1** → 1 means Magenta offset
AWB_TAG_GM_OFFSET_T : **700** → Magenta offset = 700
AWB_TAG_GM_OFFSET_THR_T : 400
AWB_TAG_WEIGHT_T : **160** → Weight = 160

  - Modify weight of the corresponding offset to affect P2 probability (256 means 100%)

```
// AWB light source weight LUT
{
    //Tungsten
    {
        { // LUT: use magenta offset (0~1000) as index to get tungsten weight (x/256)
          // 0   100  200  300  400  500  600  700  800  900 1000
            256, 256, 256, 256, 256, 256, 256, 160,  96,  80,  32
        },
        { // LUT: use green offset (0~1000) as index to get tungsten weight (x/256)
          // 0   100  200  300  400  500  600  700  800  900 1000
            256, 256, 256, 256, 256, 256, 256, 128,  64,  32,  16
        },
    },
```

Magenta offset vs. weight

Green offset vs. weight

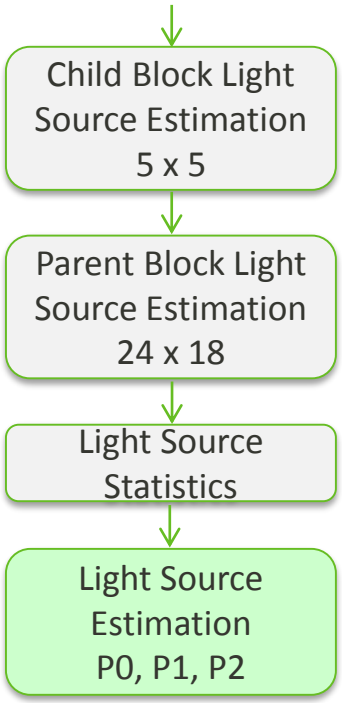
Child Block Light Source Estimation 5 x 5

Parent Block Light Source Estimation 24 x 18

Light Source Statistics

Light Source Estimation P0, P1, P2

Light Source Estimation – P2 (3/5)



- **P2 Stability** remapping function
  - Reduce P2 effect to improve AWB stability
  - Issue : In outdoor confusion color scene, AWB is not stable
    - Parent block number is small
    - P2 effect of is too strong

  - Solution : Reduce P2 effect if …
    - Total Neutral PB number is small
    - LV is small
    - P0 is large

Light Source Estimation – P2 (4/5)

- **P2 Stability**
  - Case :
    - Number of total Parent Block is small
    - [Tungsten] P0 is small, P2 = 100%, P = 40%
    - [WF] P0 = 96%, P2 = 6%, P = 60%

    Parent block number of Tungsten is small but affect final gain too much

  - Solution :
    - Enable P2 stability function
    - P2 of WF is increased by P2 stability function
    - P of Tungsten is decreased for better AWB stability

	Without P2 Stability		With P2 Stability	
	Tungsten	WF	Tungsten	WF
PB_Num	2	49	2	49
P0	4	96	4	96
P1	90	90	90	90
P2	100	6	100	46
P	40	60	8	92

Light Source Estimation – P2 (5/5)

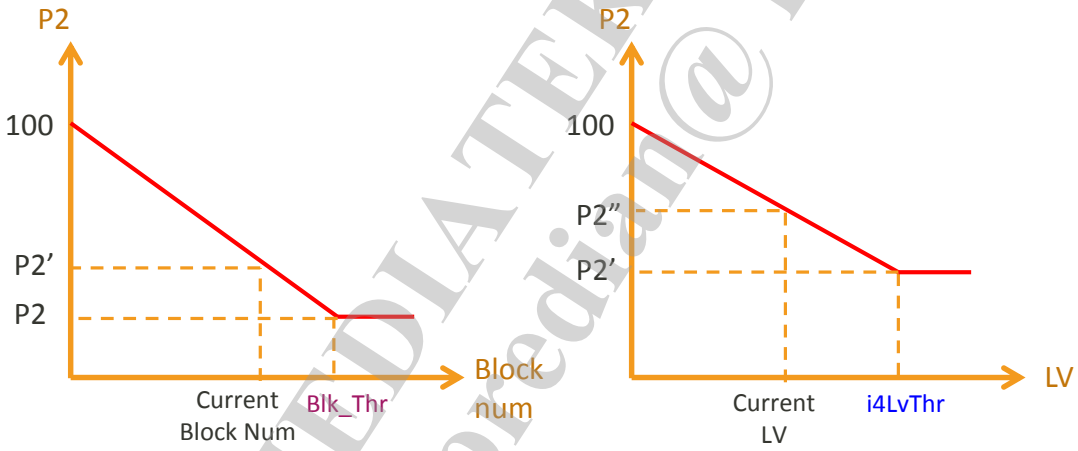
- **P2 Stability**
  - Tuning parameters

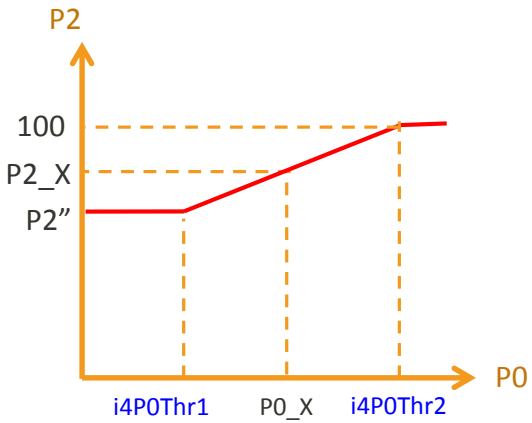
```
// P2 Stability
        {
            1,  //i4Enable
           20,  //i4PbRatio
           20,  //i4LvThr
           80,  //i4P0Thr1
          110   //i4P0Thr2
        },
```

用來計算 Active Block Number (BlkThr),
BlkThr = 總ParentBlock數 * i4PbRatio = Round(432*20/100) = 86
如果整體Parent Block Number小於BlkThr, 則重新計算P2

LV小於i4LvThr, 則重新計算p2

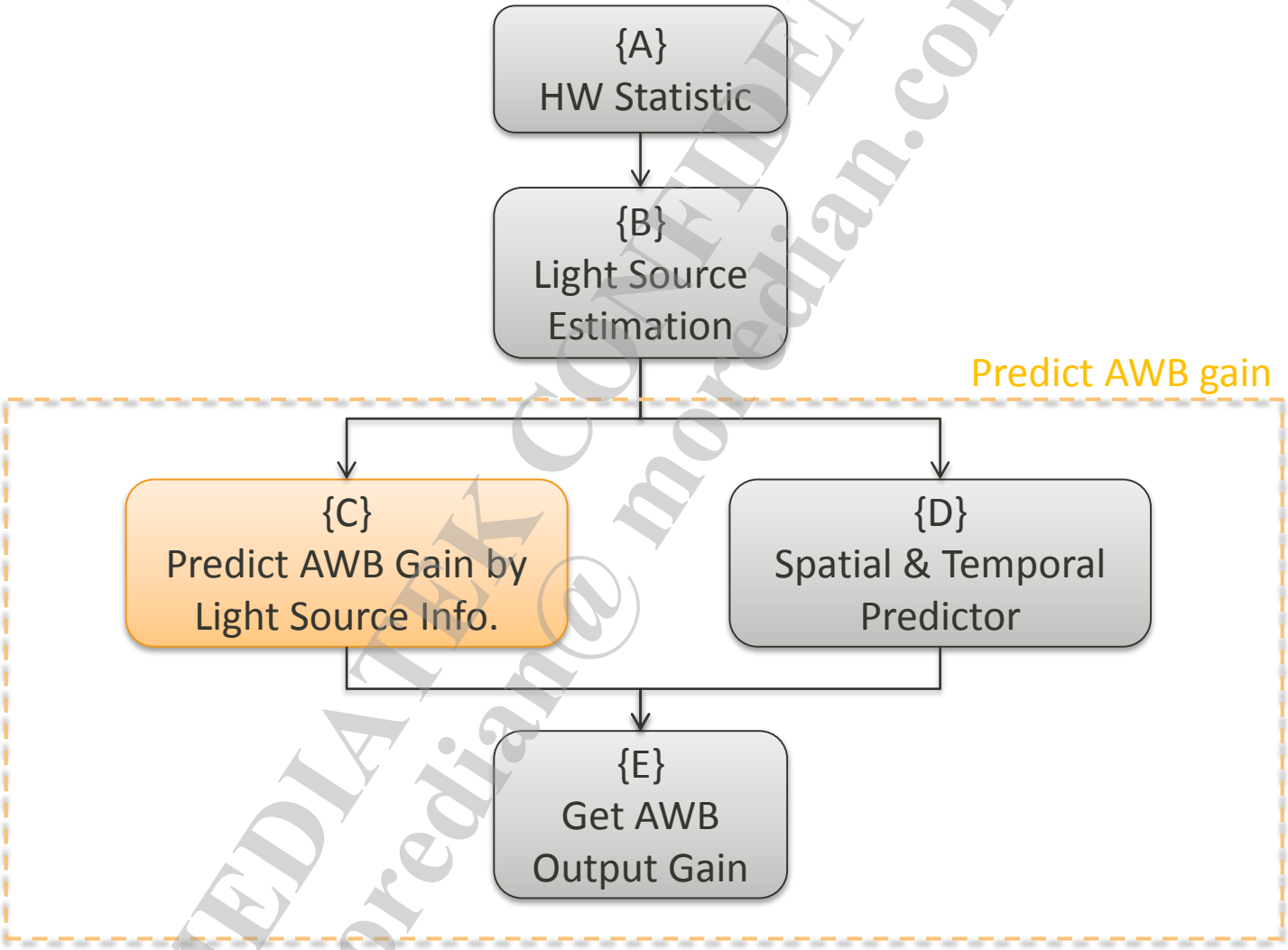
根據原始P0, 重新計算P2,
原始P0越大, 則P2越大





{C} Predict AWB Gain by Light Source Info.

AWB Flow



Preference Color Compensation

Preference Color Compensation (1/11)

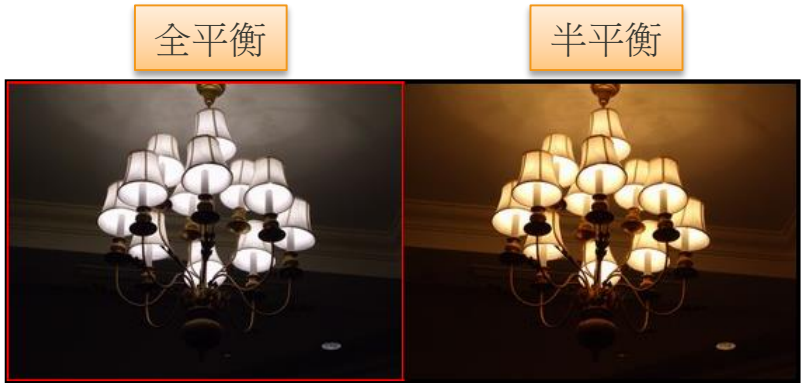
- **Purpose :**

  Preference color is used control WB convergence degree for Tungsten, WF and Shade

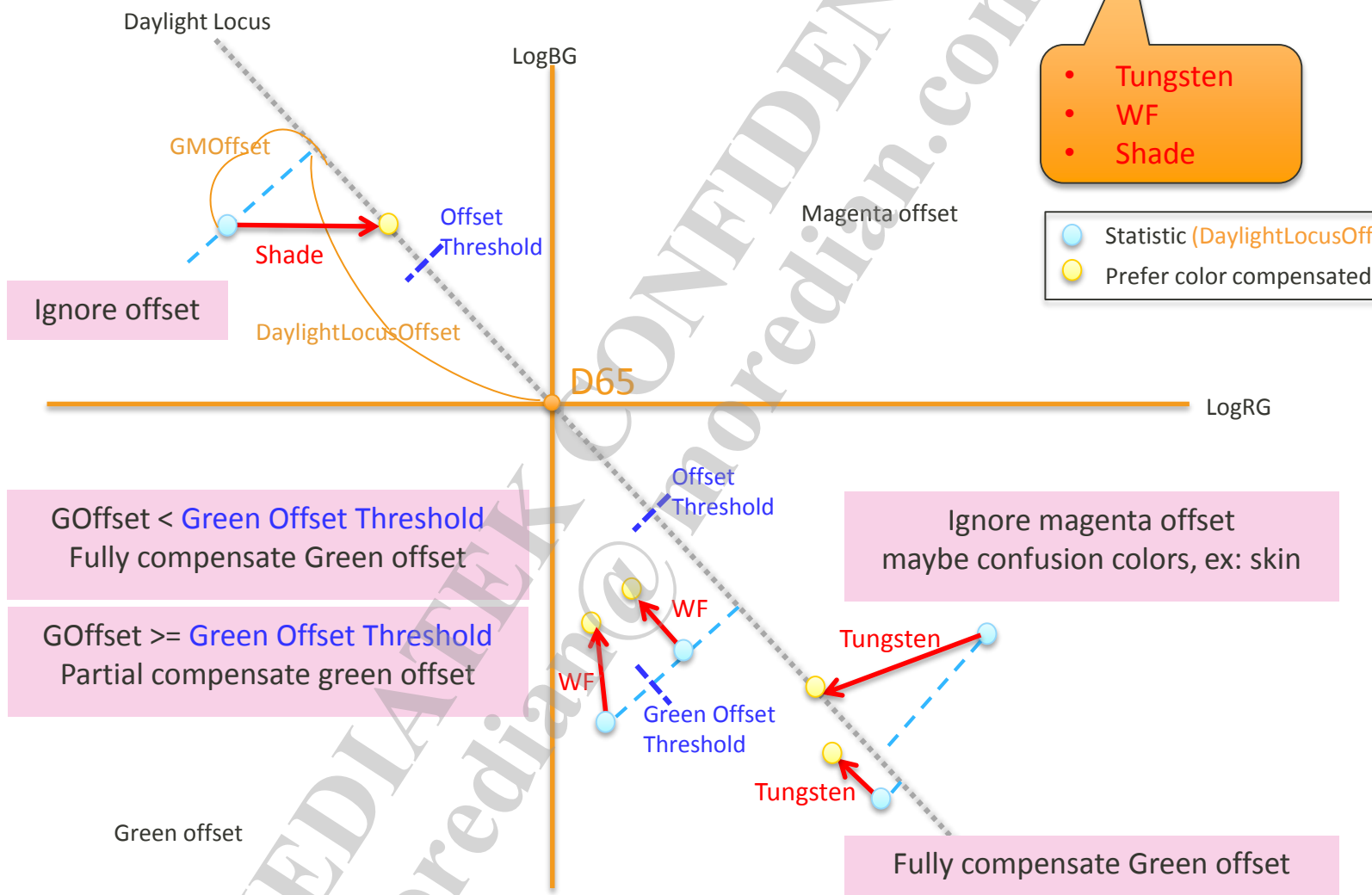
- **Method :**

  Do full compensation when current distance from the origin to the white point(log(R/G), log(B/G)) is under Offset threshold , and do partial compensation when the distance is above Offset threshold.

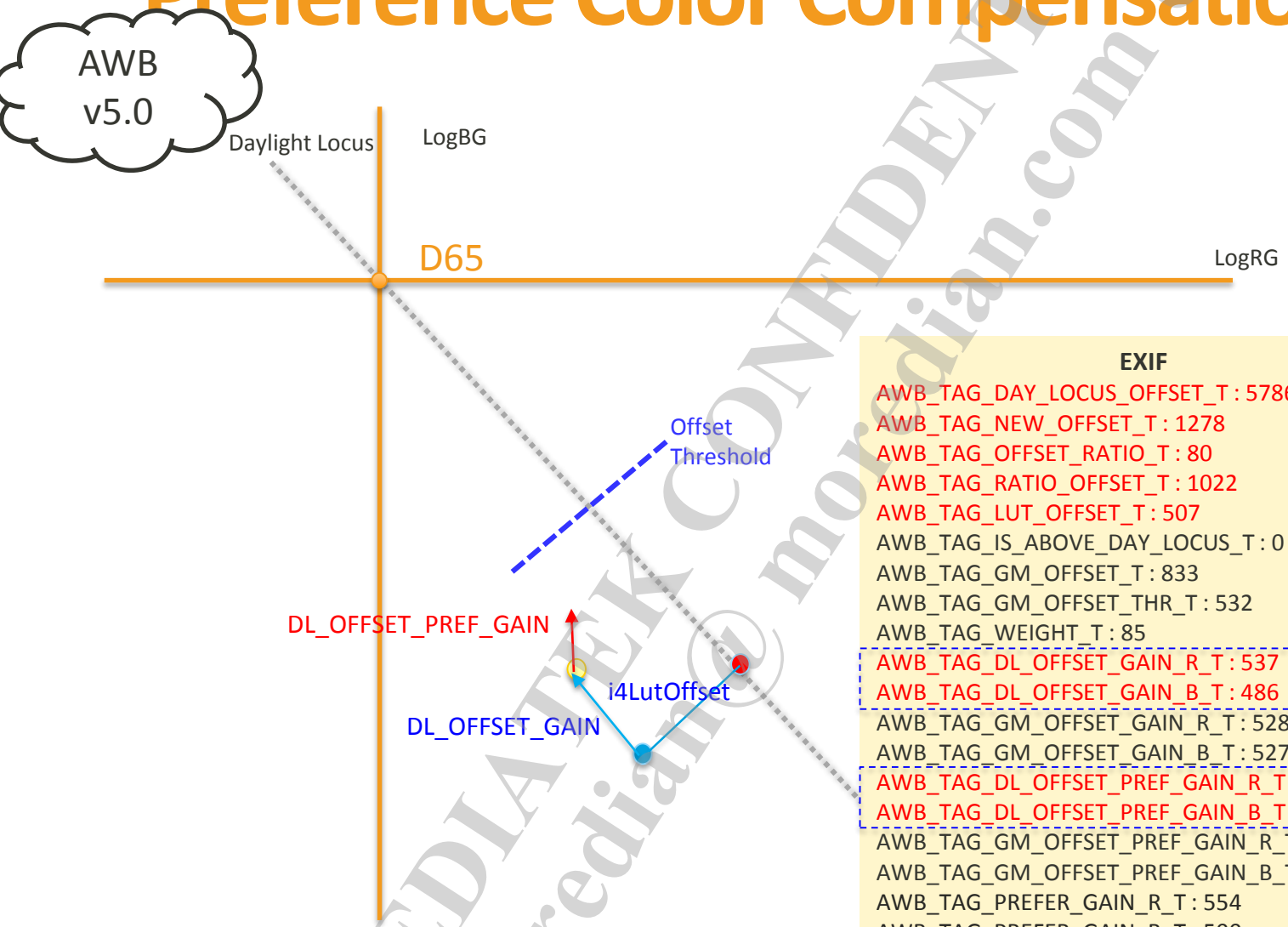
```
//Offset threshold
{
        4897, // Tungsten
        4897, // WF
        909  // Shade
},
```



Preference Color Compensation (2/11)



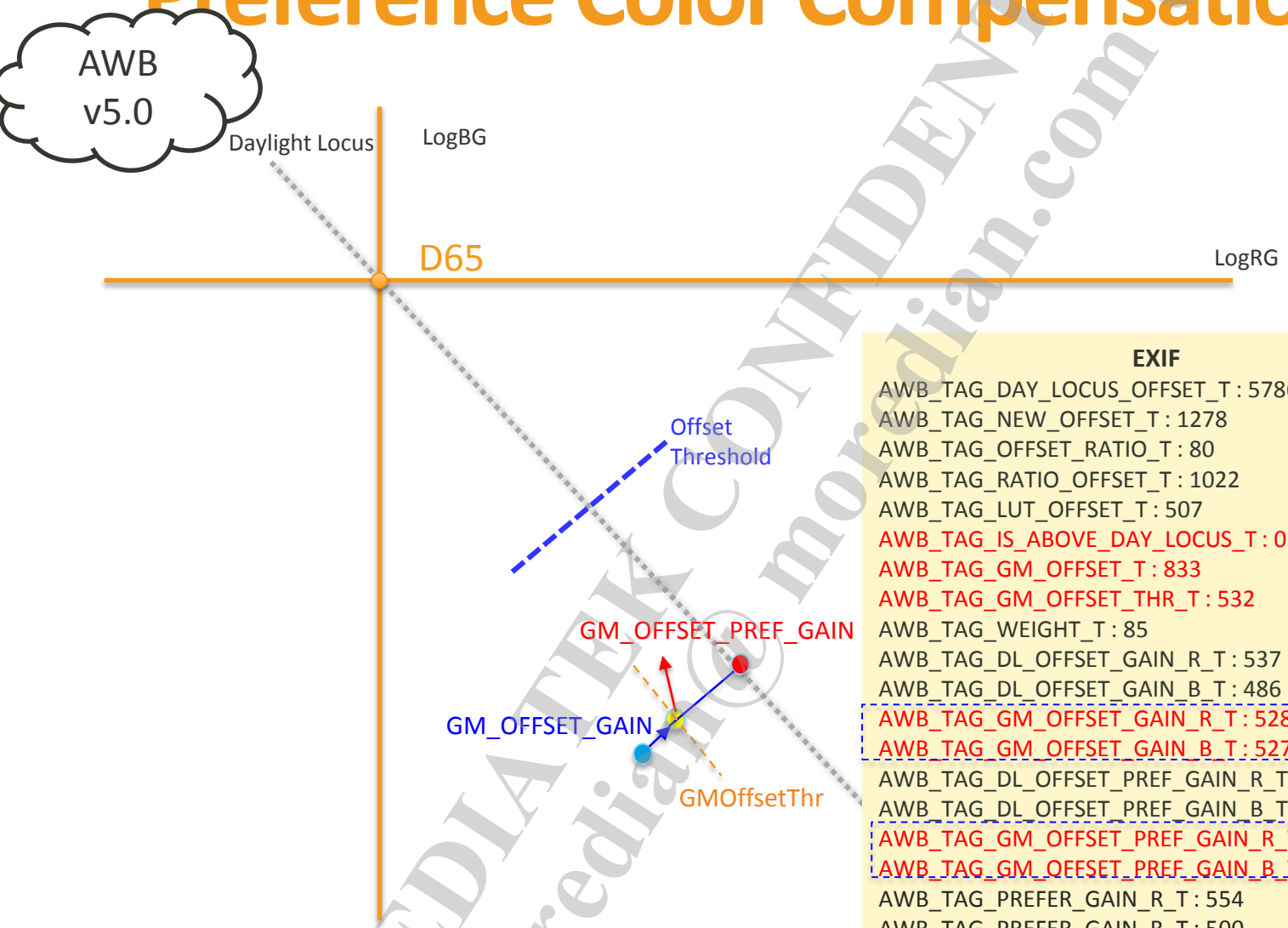
Preference Color Compensation (3/11)



EXIF

AWB_TAG_DAY_LOCUS_OFFSET_T : 5786
AWB_TAG_NEW_OFFSET_T : 1278
AWB_TAG_OFFSET_RATIO_T : 80
AWB_TAG_RATIO_OFFSET_T : 1022
AWB_TAG_LUT_OFFSET_T : 507
AWB_TAG_IS_ABOVE_DAY_LOCUS_T : 0
AWB_TAG_GM_OFFSET_T : 833
AWB_TAG_GM_OFFSET_THR_T : 532
AWB_TAG_WEIGHT_T : 85
AWB_TAG_DL_OFFSET_GAIN_R_T : 537
AWB_TAG_DL_OFFSET_GAIN_B_T : 486 — Daylight locus offset corresponding gain
AWB_TAG_GM_OFFSET_GAIN_R_T : 528
AWB_TAG_GM_OFFSET_GAIN_B_T : 527
AWB_TAG_DL_OFFSET_PREF_GAIN_R_T : 512
AWB_TAG_DL_OFFSET_PREF_GAIN_B_T : 512 — Daylight locus offset preference gain by tuning
AWB_TAG_GM_OFFSET_PREF_GAIN_R_T : 512
AWB_TAG_GM_OFFSET_PREF_GAIN_B_T : 512
AWB_TAG_PREFER_GAIN_R_T : 554
AWB_TAG_PREFER_GAIN_B_T : 500

Preference Color Compensation (4/11)



EXIF

AWB_TAG_DAY_LOCUS_OFFSET_T : 5786
AWB_TAG_NEW_OFFSET_T : 1278
AWB_TAG_OFFSET_RATIO_T : 80
AWB_TAG_RATIO_OFFSET_T : 1022
AWB_TAG_LUT_OFFSET_T : 507
AWB_TAG_IS_ABOVE_DAY_LOCUS_T : 0
AWB_TAG_GM_OFFSET_T : 833
AWB_TAG_GM_OFFSET_THR_T : 532
AWB_TAG_WEIGHT_T : 85
AWB_TAG_DL_OFFSET_GAIN_R_T : 537
AWB_TAG_DL_OFFSET_GAIN_B_T : 486
AWB_TAG_GM_OFFSET_GAIN_R_T : 528
AWB_TAG_GM_OFFSET_GAIN_B_T : 527

GM offset corresponding gain

AWB_TAG_DL_OFFSET_PREF_GAIN_R_T : 512
AWB_TAG_DL_OFFSET_PREF_GAIN_B_T : 512
AWB_TAG_GM_OFFSET_PREF_GAIN_R_T : 512
AWB_TAG_GM_OFFSET_PREF_GAIN_B_T : 512

GMoffset preference gain by tunin

AWB_TAG_PREFER_GAIN_R_T : 554
AWB_TAG_PREFER_GAIN_B_T : 500

Preference Color Compensation (5/11)

AWB v5.0

```
//Daylight locus offset LUTs
{
    { // TUNGSTEN
        {   0,  500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000, 8500, 9000, 9500, 10000},
        {   0,  350,  800, 1222, 1444, 1667, 1889, 2111, 2333, 2556, 2778, 3000, 3222, 3444, 3667, 3889, 4111, 4333, 4556, 4778, 5000}
    },
```

Daylight locus offset preference gain LUT input is same as Daylight locus offset LUT

Preference gain LUT

Daylight locus offset preference gain

```
{ // TUNGSTEN
    { // LUT: use daylight locus offset as index
      {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512},
      {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512},
      },
```

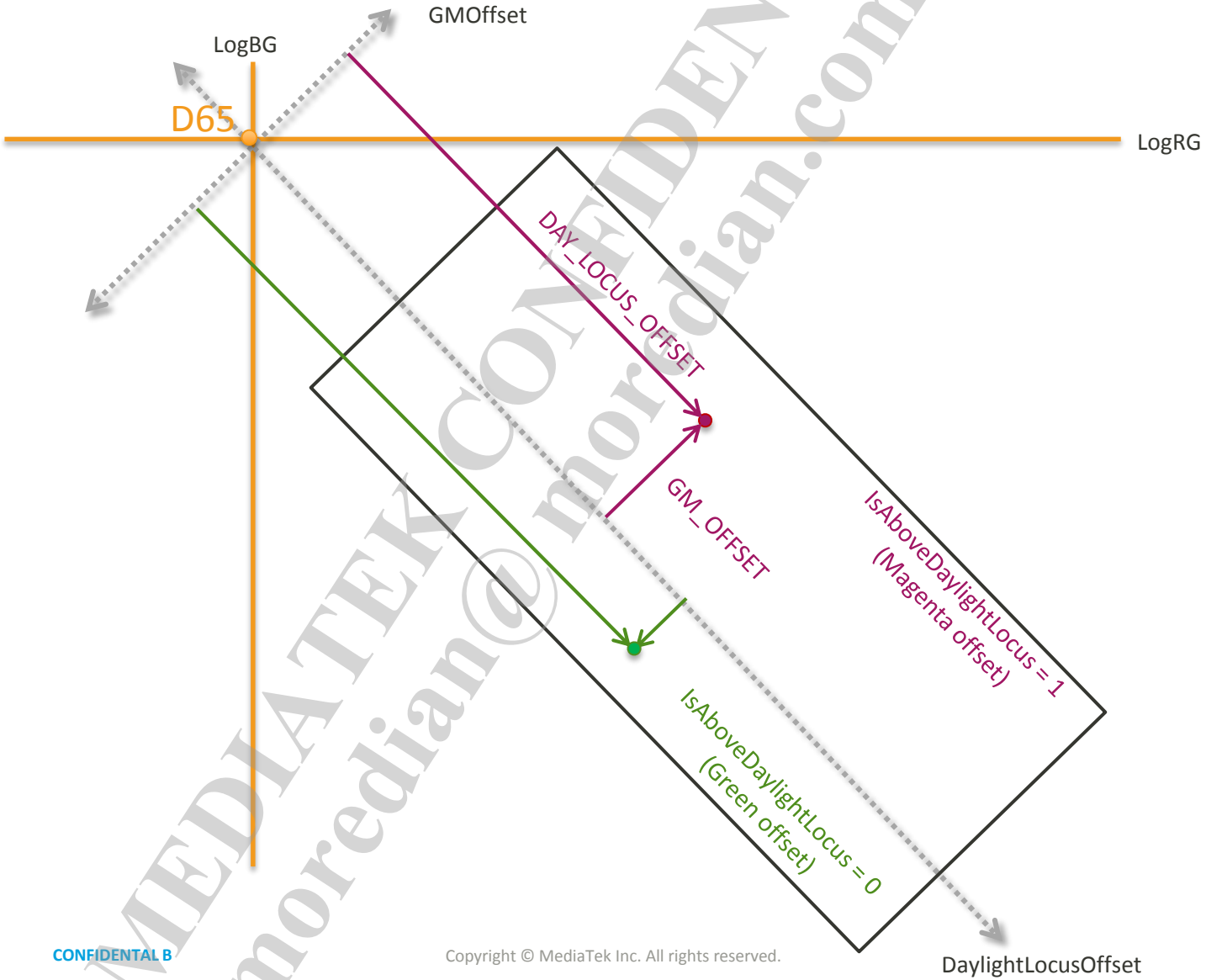
Magenta offset preference gain

```
    { // LUT: use (magenta offset - thld) as index
      //   0  100  200  300  400  500  600  700  800  900 1000
      {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
```

Green offset preference gain

```
    { // LUT: use (green offset - thld) as index
      //   0  100  200  300  400  500  600  700  800  900 1000
      {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
},
```

Preference Color Compensation (6/11)



Preference Color Compensation (7/11)

- Daylight Locus Offset

DLO : Daylight Locus Offset
PCOT : Preference Color Offset Threshold
DLNO : Daylight Locus New Offset (DLO - Offset threshold)
DLTOR : Daylight Locus Target Offset Ratio (DLNO * ratio)
DLTO : Daylight Locus Target Offset (Final result, higher means more colour cast)

AWB_TAG_ALGO_SCENE_LV : 53

AWB_NVRAM_PREF_COLOR_OFFSET_THR_WF : 4800 → Offset Threshold

AWB_TAG_DAY_LOCUS_OFFSET_WF : 8763 → DLO
AWB_TAG_NEW_OFFSET_WF : 3963 → DLNO → 8763 – 4800
AWB_TAG_OFFSET_RATIO_WF : 93 → ratio
AWB_TAG_RATIO_OFFSET_WF : 3710 → DLNO * ratio' → 3963 * 93.6%
AWB_TAG_LUT_OFFSET_WF : **1876** → **DLTO**
AWB_TAG_IS_ABOVE_DAY_LOCUS_WF : 0
AWB_TAG_GM_OFFSET_WF : 739
AWB_TAG_GM_OFFSET_THR_WF : 1095

```
//Offset ratio LV
{
        {  50, 100}, // Tungsten
        {  50,  85}, // WF
        {  80, 120}  // Shade
},
```

```
//WF
{
   {   0,  500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000, 8500, 9000, 9500, 10000},
   {   0,  550,  800,  900,  950, 1050, 1400, 1750, 2050, 2400, 2700, 3000, 3300, 3650, 4000, 4350, 4700, 5050, 5400, 5750, 6100}
```

Preference Color Compensation (8/11)

- Green/Magenta Offset

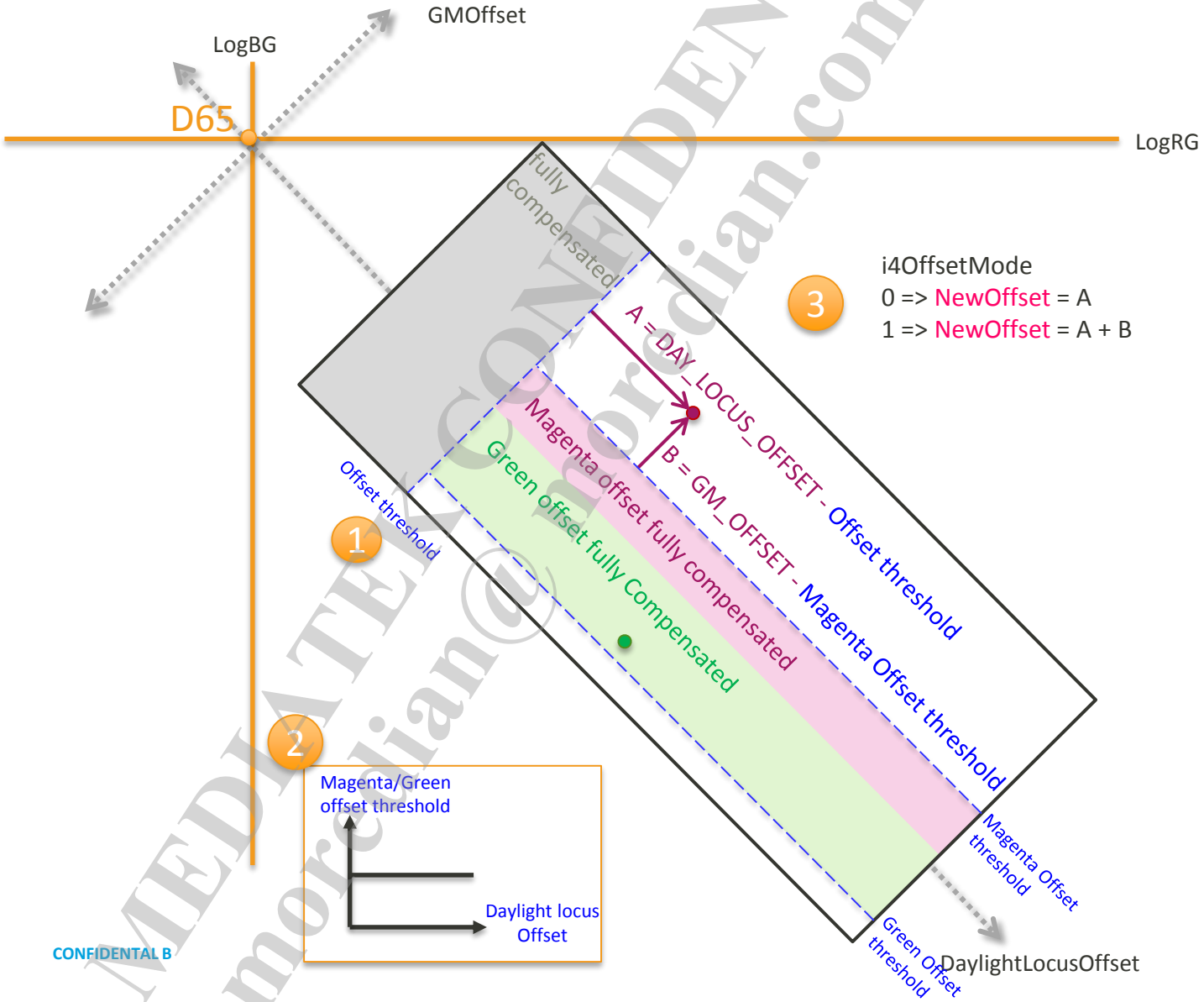
```
AWB_TAG_ALGO_SCENE_LV          : 53

AWB_NVRAM_PREF_COLOR_OFFSET_THR_WF : 4800   → Offset Threshold

AWB_TAG_DAY_LOCUS_OFFSET_WF    : 8763       → DLO
AWB_TAG_NEW_OFFSET_WF          : 3963       → DLNO → 8763 – 4800
AWB_TAG_OFFSET_RATIO_WF        : 93         → ratio
AWB_TAG_RATIO_OFFSET_WF        : 3710       → DLNO * ratio' → 3963 * 93.6%
AWB_TAG_LUT_OFFSET_WF          : 1876       → DLTO
AWB_TAG_IS_ABOVE_DAY_LOCUS_WF : 0
AWB_TAG_GM_OFFSET_WF           : 739
AWB_TAG_GM_OFFSET_THR_WF       : 1095
```

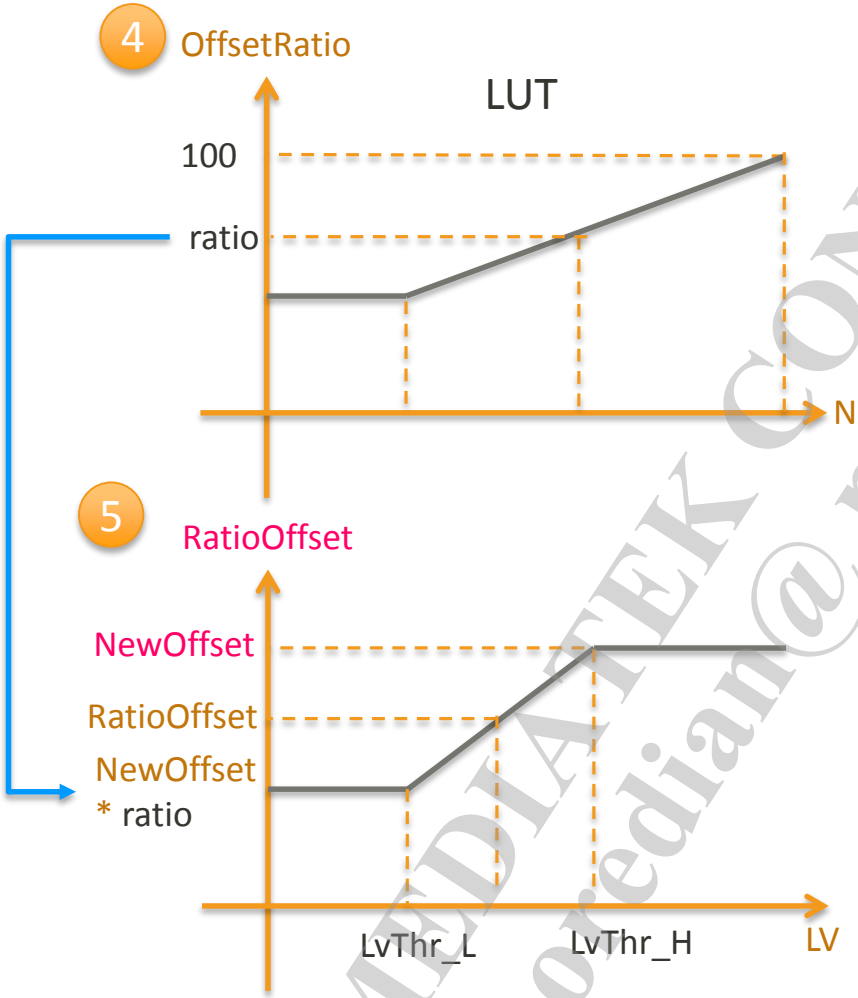
```
//Warm Fluorescent
{
    1, //0: disabe 1: enable offset when GM offset is large than threshold
    { // LUT: use daylight locus offset (0~10000) as index to get magenta offset threshold
      //  0   500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
        1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500
    },
    { // LUT: use daylight locus offset (0~10000) as index to get green offset threshold
      //  0   500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
        1500, 1500, 1500, 1500, 1500, 1500, 1500, 1400, 1300, 1200, 1100,  950, 1000, 1100, 1250, 1300, 1300, 1200, 1000, 1200, 1350
    },
},
```

Preference Color Compensation (9/11)

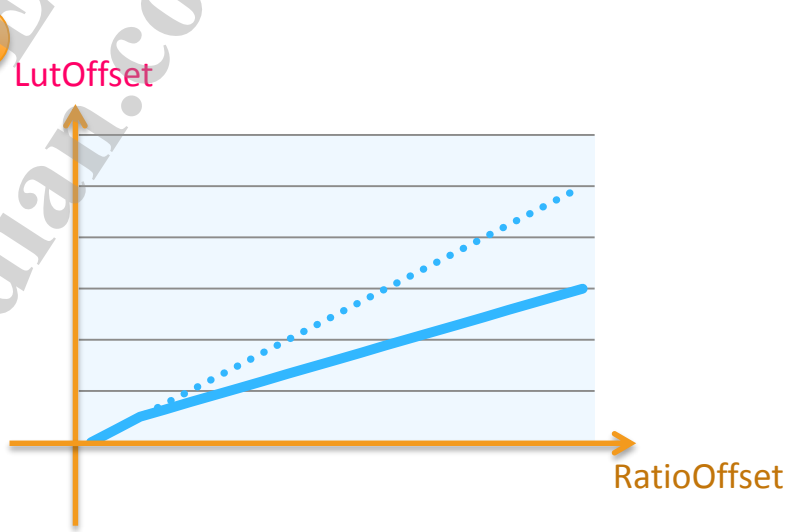


Preference Color Compensation (10/11)

Remap A+B offset by LV



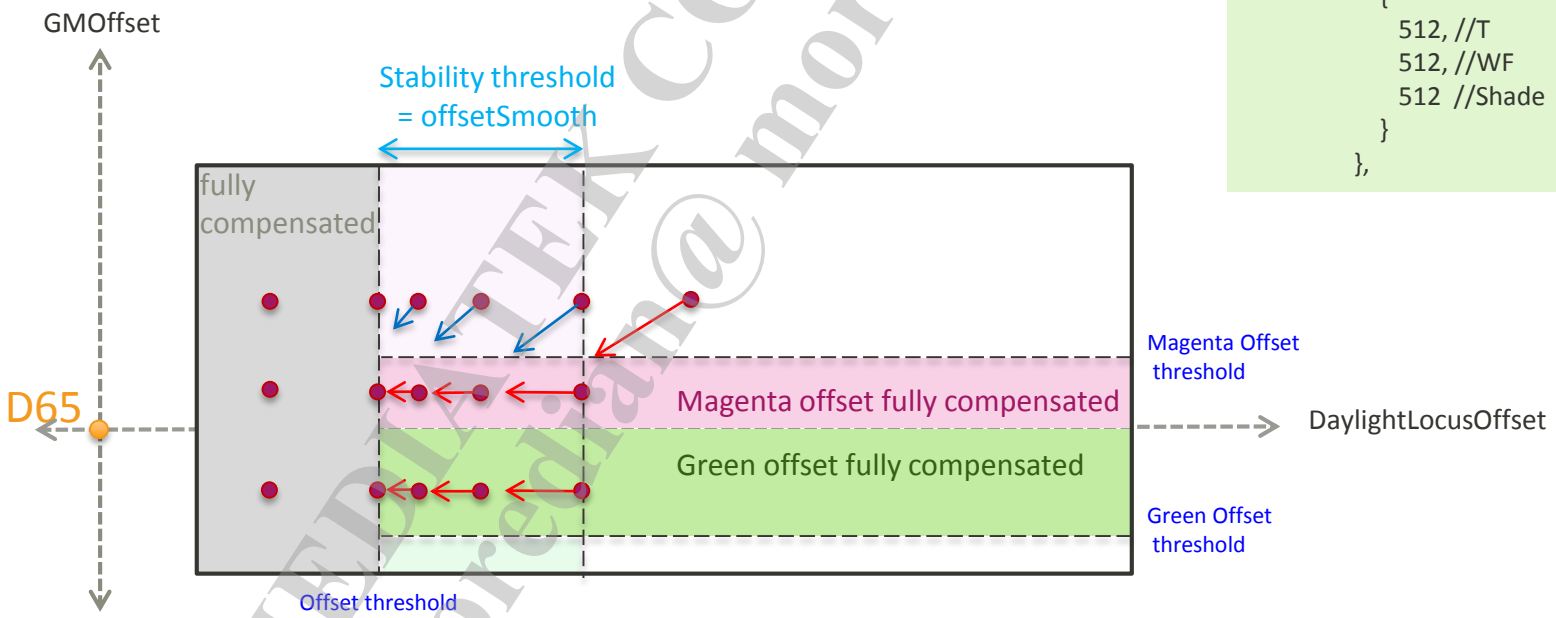
Remap A+B offset by LUT



Preference Color Compensation (11/11)

- Issue : Due to Green/Magenta Offset partial compensation, the unstable situation is obvious near the offset threshold.
- Solution : Reduce Green/Magenta Offset partial compensation effect near threshold.

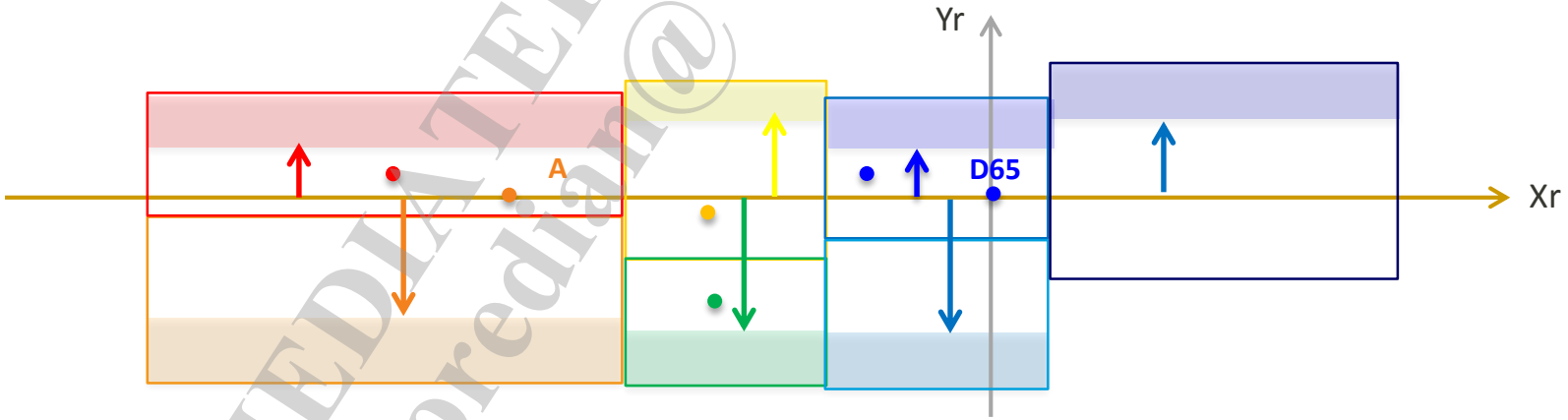
```
//Preference Color Stability
        {
          1,   //i4Enable
          {
            512, //T
            512, //WF
            512  //Shade
          }
        },
```



AWB
v5.0

Statistic Gain Constraint (1/7)

- **Purpose :**
  - Due to AWB stability issue, we need larger size for white point. And we need some restriction for these area.
    1. Project Statistic Yr toward limited boundary direction by projected weighting (影響Statistic Gain)
       - Add new mode to project by each parent block
       - Add projected weighting to control project ratio
    2. Reduce Statistic Weighting (影響Statistic Weighting)



AWB
v5.0

Statistic Gain Constraint (2/7)

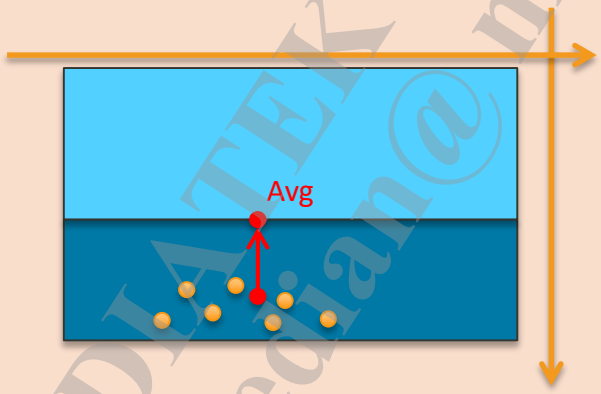
- **Tuning Parameters:**
  - Mode
    - 0: disable
    - 1: Each light sources statistics projection
    - 2: Each block statistics projection
  - Lv (usually in low LV converge to white)
    - High : reduce effect from high threshold
    - Low : no effect when LV lower than low threshold
  - Each light source Yr limitation (i4LimitY[AWB_LIGHT_NUM])
  - Each light source reduce weighting (i4WeightReduce[AWB_LIGHT_NUM])
    - Unit : 16
  - Each light source projection weighting(i4WeightReduce[AWB_LIGHT_NUM])
    - Unit : 16

```
// rStatLimit
{
    1,    //i4Enable
    { 10, 30},
    // i4LimitY[AWB_LIGHT_NUM]
    {
        0,   //Strobe
       50,   //T
      180,   //WF
       40,   //F
      138,   //CWF
       45,   //Daylight
       40,   //Shade
      115,   //DF
    },
    // i4WeightReduce[AWB_LIGHT_NUM]
    {
        0,   //Strobe
        8,   //Tungsten
        8,   //WF
        8,   //F
        8,   //CWF
        8,   //Daylight
        8,   //Shade
        8,   //DF
    },
    // i4ProjWeight[AWB_LIGHT_NUM]
    {
       16,   //Strobe
       16,   //Tungsten
       16,   //WF
       16,   //F
       16,   //CWF
       16,   //Daylight
       16,   //Shade
       16,   //DF
    },
},
```

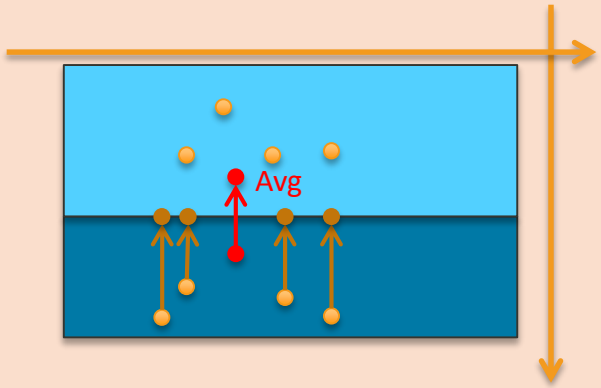
AWB
v5.0

Statistic Gain Constraint (3/7)

	Mode 1 (AWB v4.0 original)	Mode 2
Pros	Simple calculation : only use average XrYr Easy for tuning : less tuning parameters Effective for large confusion color	Complex calculation : calculate on each PB (24*18) More tuning flexibility : Also effective for evenly distributed white points



p.s. The example is for projected weighting = 16



p.s. The example is for projected weighting = 16

Statistic Gain Constraint (4/7)

- Case

Original

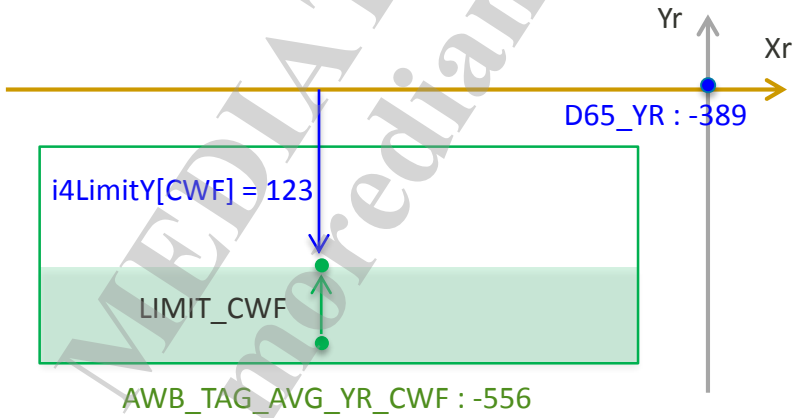

```
// rStatLimit
{
  1,   //i4Enable
  {10, 30},
  // i4LimitY[AWB_LIGHT_NUM]
  {
      0,  //Strobe
     59,  //T
    152,  //WF
     40,  //F
    123,  //CWF
     40,  //Daylight
     30,  //Shade
    123,  //DF
  },
 // i4WeightReduce[AWB_LIGHT_NUM]
  {
     0,  //Strobe
     8,  //Tungsten
     8,  //WF
     8,  //F
     8,  //CWF
     8,  //Daylight
     8,  //Shade
     8,  //DF
  }
// i4ProjWeight[AWB_LIGHT_NUM]
  {
     16,  //Strobe
     16,  //Tungsten
     16,  //WF
     16,  //F
     16,  //CWF
     16,  //Daylight
     16,  //Shade
     16,  //DF
  }
},
```

Statistic Gain Constraint (5/7)

Statistic Projected

LIMIT_X = Statistic Yr - i4LimitY[X]

Example : LIMIT_CWF =abs (-556-(-389)) – 123 = 43



EXIF

AWB_NVRAM_D65_XR : 171
AWB_NVRAM_D65_YR : -389
AWB_TAG_AVG_XR_CWF : -47
AWB_TAG_AVG_YR_CWF : -556
AWB_TAG_AVG_XR_DF : 87
AWB_TAG_AVG_YR_DF : -530

EXIF

AWB_TAG_LIMIT_STB : 0
AWB_TAG_LIMIT_T : 0
AWB_TAG_LIMIT_WF : 0
AWB_TAG_LIMIT_F : 0
AWB_TAG_LIMIT_CWF : 43
AWB_TAG_LIMIT_D : 0
AWB_TAG_LIMIT_S : 0
AWB_TAG_LIMIT_DF : 18

Statistic Gain Constraint (6/7)

Statistic Projected and reduce statistic weighting

LIMIT_X = Statistic Yr - i4LimitY[X]
Example : LIMIT_CWF =abs (-556-(-389)) – 123 = 43

Reduce Daylight Prob = LIMIT_X * i4WeightReduce[X] / 16
Example : EQV_DAYLIGHT_PROB_CWF = 87 – (43* 8 /16) = 66

EXIF

AWB_NVRAM_D65_XR : 171
AWB_NVRAM_D65_YR : -389
AWB_TAG_AVG_XR_CWF : -47
AWB_TAG_AVG_YR_CWF : -556
AWB_TAG_AVG_XR_DF : 87
AWB_TAG_AVG_YR_DF : -530

EXIF

AWB_TAG_LIMIT_STB : 0
AWB_TAG_LIMIT_T : 0
AWB_TAG_LIMIT_WF : 0
AWB_TAG_LIMIT_F : 0
AWB_TAG_LIMIT_CWF : 43
AWB_TAG_LIMIT_D : 0
AWB_TAG_LIMIT_S : 0
AWB_TAG_LIMIT_DF : 18

EXIF

AWB_TAG_DAYLIGHT_PROB_CWF : 90
AWB_TAG_EQV_DAYLIGHT_PROB_CWF : 87

Statistic Gain Constraint (7/7)

Original

EXIF

AWB_TAG_STA_GAIN_R_CWF : 491
AWB_TAG_STA_GAIN_G_CWF : 512
AWB_TAG_STA_GAIN_B_CWF : 863
AWB_TAG_SPAT_GAIN_R_CWF : 397
AWB_TAG_SPAT_GAIN_G_CWF : 512
AWB_TAG_SPAT_GAIN_B_CWF : 672
AWB_TAG_HIT_NR_CWF : 0
AWB_TAG_DAYLIGHT_PROB_CWF : 90
AWB_TAG_EQV_DAYLIGHT_PROB_CWF : 87
AWB_TAG_EQV_GAIN_R_CWF : 477
AWB_TAG_EQV_GAIN_G_CWF : 512
AWB_TAG_EQV_GAIN_B_CWF : 833

Projected

EXIF

AWB_TAG_STA_GAIN_R_CWF : 461
AWB_TAG_STA_GAIN_G_CWF : 512
AWB_TAG_STA_GAIN_B_CWF : 813
AWB_TAG_SPAT_GAIN_R_CWF : 397
AWB_TAG_SPAT_GAIN_G_CWF : 512
AWB_TAG_SPAT_GAIN_B_CWF : 672
AWB_TAG_HIT_NR_CWF : 0
AWB_TAG_DAYLIGHT_PROB_CWF : 90
AWB_TAG_EQV_DAYLIGHT_PROB_CWF : 87
AWB_TAG_EQV_GAIN_R_CWF : 452
AWB_TAG_EQV_GAIN_G_CWF : 512
AWB_TAG_EQV_GAIN_B_CWF : 794

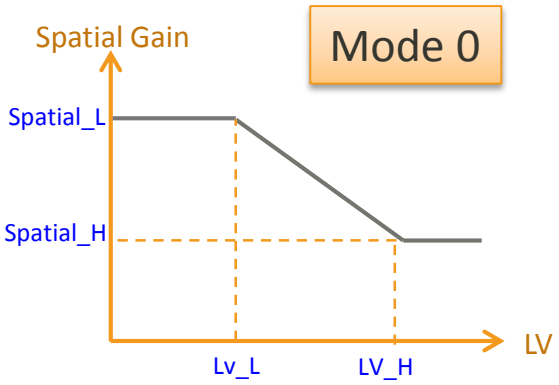
Projected & Reduce weighting

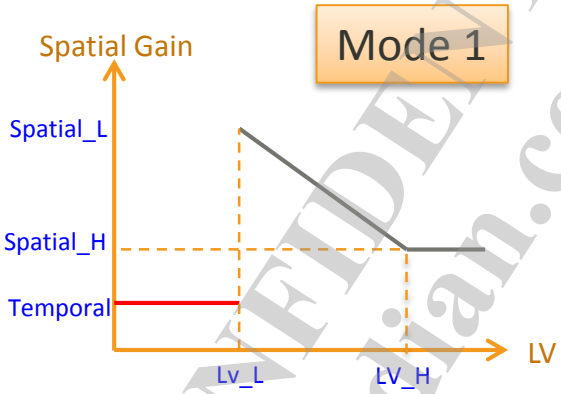
EXIF

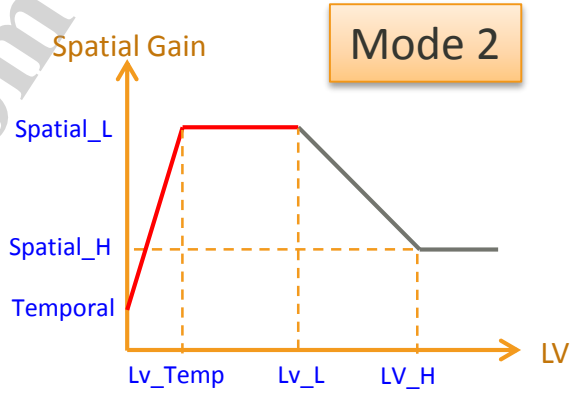
AWB_TAG_STA_GAIN_R_CWF : 461
AWB_TAG_STA_GAIN_G_CWF : 512
AWB_TAG_STA_GAIN_B_CWF : 813
AWB_TAG_SPAT_GAIN_R_CWF : 397
AWB_TAG_SPAT_GAIN_G_CWF : 512
AWB_TAG_SPAT_GAIN_B_CWF : 672
AWB_TAG_HIT_NR_CWF : 0
AWB_TAG_DAYLIGHT_PROB_CWF : 90
AWB_TAG_EQV_DAYLIGHT_PROB_CWF : 66
AWB_TAG_EQV_GAIN_R_CWF : 439
AWB_TAG_EQV_GAIN_G_CWF : 512
AWB_TAG_EQV_GAIN_B_CWF : 762

Spatial Predictor & Sunset Compensation

Spatial Predictor v4.0







// Predictor gain

```
{
   { // rSpatial_L
      991, // i4R
      512, // i4G
      697  // i4B
   },
   {// rSpatial_H
      905, // i4R
      512, // i4G
      862  // i4B
   },
   {// rTemporal
      896, // i4R
      512, // i4G
      920  // i4B
   }},
```

// AWB LV threshold for predictor

```
{
   // Non-Reliable
      { 115, 155},

   // Reliable
   {
      {  80,  135}, // Strobe
      { 110, 160}, // Tungsten
      { 110, 160}, // WF
      { 110, 160}, // Fluorescent
      { 110, 160}, // CWF
      {  80,  135}, // Daylight
      {  80,  135}, // Shade
      {  80,  135}, // DF
   },
},
```

// Spatial gain

```
{
    2, //i4GeneralMode
    2, //i4NonReliableMode
    2, //i4TempInitMode
   40, //i4ThrTemp
   {
      40, //Strobe
      40, //T
      40, //WF
      40, //F
      40, //CWF
      40, //D
      40, //Shade
      40, //DF
   }
},
```

Reliable mode

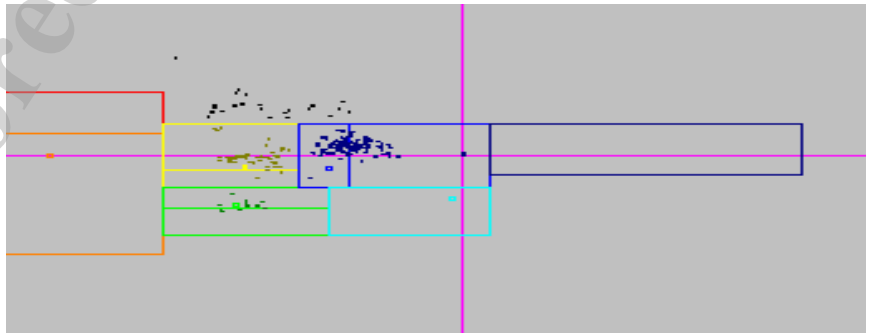
Non-Reliable mode

Sunset Compensation (1/2)

- Since the sunset scenario is high LV and using more spatial gain. In this condition, white balance preserve too much color.
- **Condition:**
  - Block in daylight prefer area and LV is high.
  - Sunset block count and ratio is high

```
// Sunset Prop
{
    1, // i4Enable
    120, // i4LVThr_L
    130, // i4LVThr_H
    10, // i4SunsetCountThr
    0, // i4SunsetCountRatio_L
    171, // i4SunsetCountRatio_H
    {
        75, // i4Sunset_BoundXr_Thr
        -415  // i4Sunset_BoundYr_Thr
    }
},
```

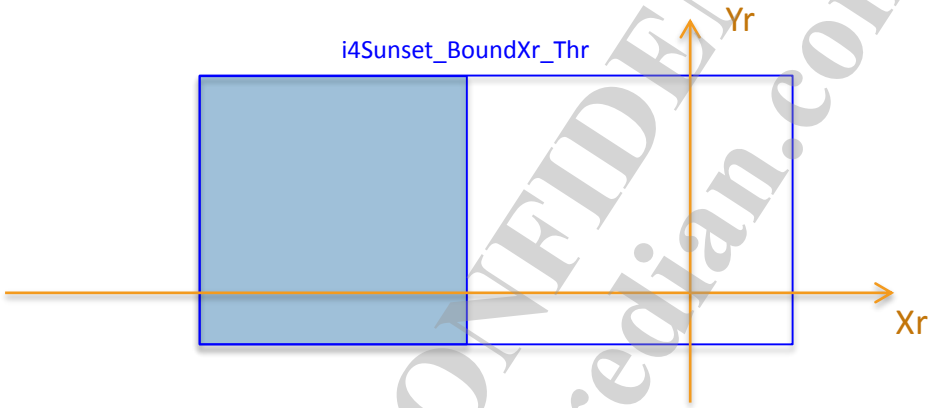
EXIF:
AWB_TAG_HIT_SUNSET_D : 1

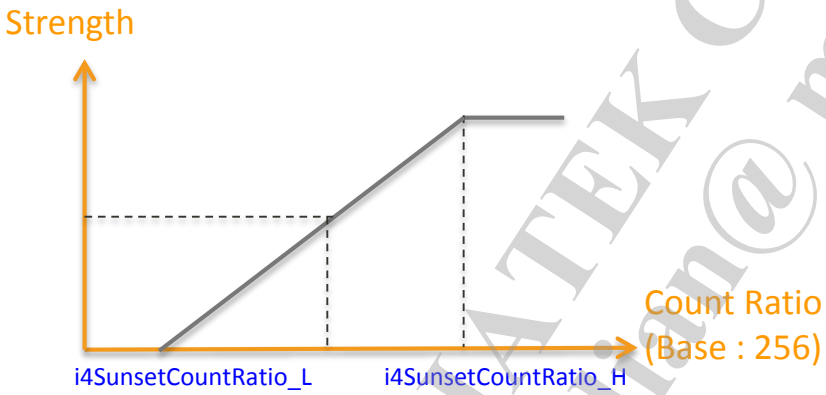


Effect Weak

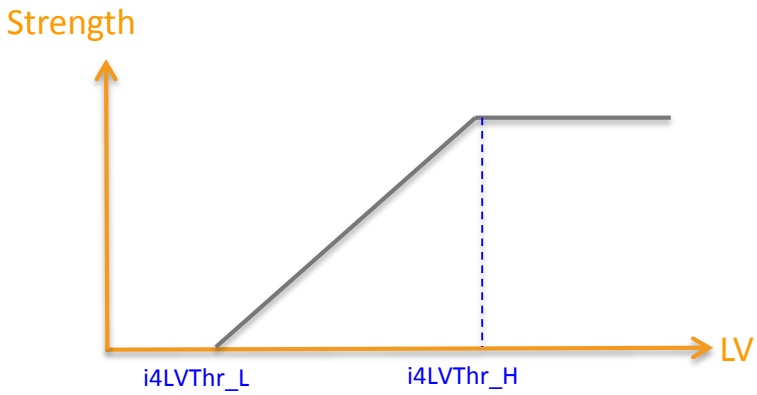
Effect Strong

Sunset Compensation (2/2)





Higher Ratio means stronger effect



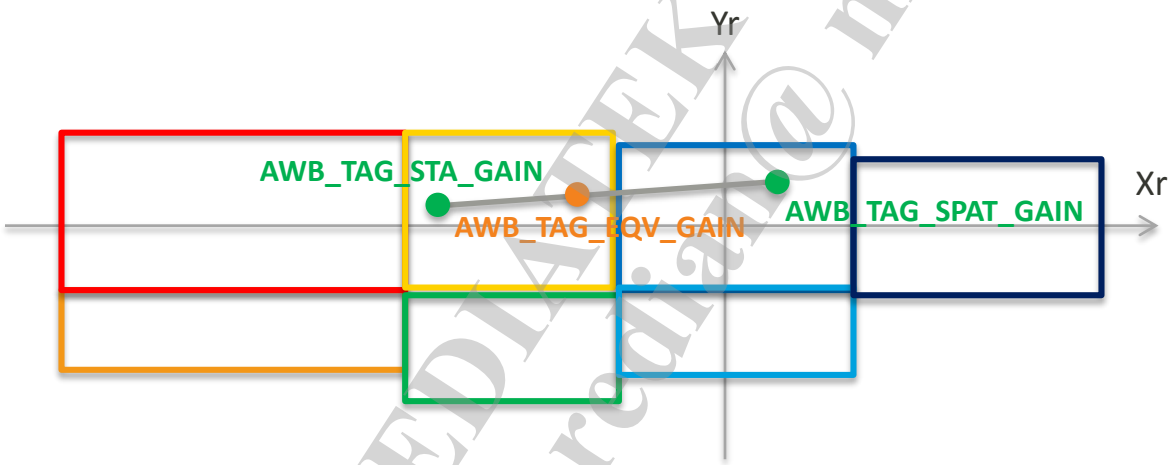
Higher Scene LV means stronger effect

Mix Statistic Gain & Spatial Gain

Daylight Locus Probability (1/2)

- Each Light source have three gains and daylight locus probability.
  - **Statistic WB Gain** : Calculated by statistic data
  - **Spatial WB Gain** : Default WB gain for each light source. Defined in tuning parameters and calculate by scene LV.
  - **Equivalent WB Gain** : WB gain blending from statistic and spatial gain
  - **Daylight locus probability** : Statistic gain and Spatial gain blending weight

EqvGain = (Statistic Gain*DaylightLocusProb + Spatial Gain*(100-DaylightLocusProb))/100



EXIF

AWB_TAG_STA_GAIN_R_X : 445
AWB_TAG_STA_GAIN_G_X : 512
AWB_TAG_STA_GAIN_B_X : 591
AWB_TAG_SPAT_GAIN_R_X : 483
AWB_TAG_SPAT_GAIN_G_X : 512
AWB_TAG_SPAT_GAIN_B_X : 510
AWB_TAG_HIT_NR_X : 0
AWB_TAG_DAYLIGHT_PROB_X : 100
AWB_TAG_EQV_DAYLIGHT_PROB_X : 100
AWB_TAG_EQV_GAIN_R_X : 445
AWB_TAG_EQV_GAIN_G_X : 512
AWB_TAG_EQV_GAIN_B_X : 589

Note: X means each light source

Daylight Locus Probability (2/2)

- **Daylight Probability**
  - High daylight locus probability means the statistic WB gain is more reliable.
  - Low daylight locus probability means the spatial WB gain is more reliable.

```
// AWB daylight locus probability (Statistic Weighting) look-up table (Max: 100; Min: 0)
{
    //LV0   1     2     3     4     5     6     7     8     9     10    11    12    13    14    15    16    17    18
    { 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  50,   25,   0,    0,    0,    0}, // Strobe
    { 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  95,   75,   75,   75,   75,   75,   75,   75,   0}, // Tungsten
    { 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  95,   75,   75,   75,   75,   75,   75,   75,   75}, // Warm fluorescent
    { 100,  100,  90,   90,   100,  100,  100,  95,   95,   100,  95,   80,   80,   80,   55,   35,   10,   0,    0}, // Fluorescent
    { 100,  100,  85,   90,   90,   90,   100,  100,  100,  95,   85,   80,   46,   37,   30,   30,   0,    0,    0}, // CWF
    { 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  95,   80,   85,   85,   70,   70,   70,   50,   50}, // Daylight
    { 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  75,   50,   50,   70,   48,   25,   0,    0}, // Shade
    { 90,   90,   90,   90,   90,   90,   90,   90,   95,   80,   80,   70,   82,   57,   30,   0,    0,    0,    0}  // Daylight fluorescent
},
```

If noisy pattern is more concerned, use more spatial WB is more stable

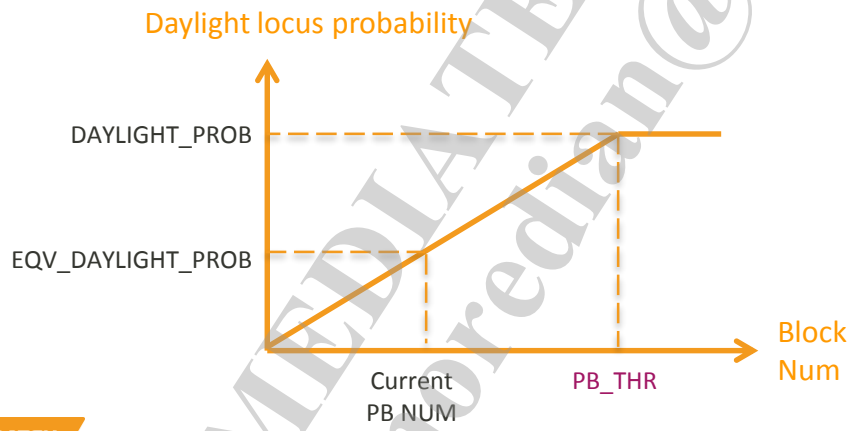
Use more statistic WB gain

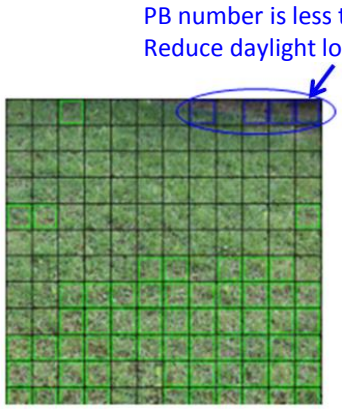
Use more spatial WB gain

AWB Noise Reduction (1/2)

- Reduce the daylight locus probability when light source count is less than threshold
  - PB_THR = Total Block Num * Ratio / 100

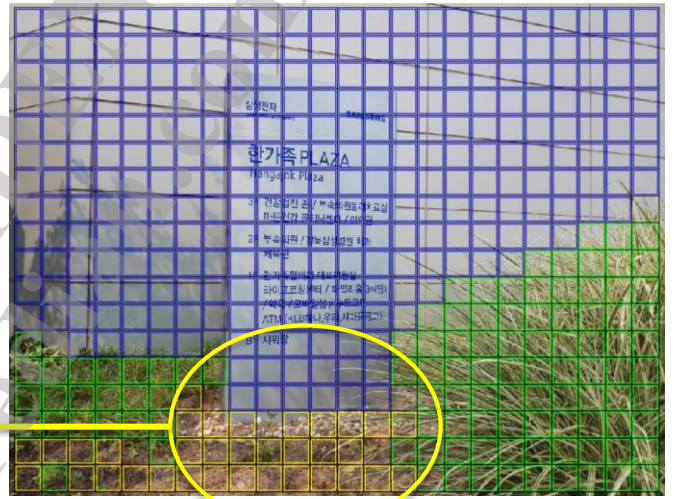
```
// Noise Reduction
{
    //LV 0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0}, //Strobe
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0}, //Tungsten
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0}, //WF
    {    0,   0,   0,   0,   0,   3,   5,   5,   5,   5,   5,  10,  10,  10,  10,  10,  10,  10,  10}, //Fluorescent
    {    0,   0,   0,   0,   0,   3,   5,   5,   5,   5,   5,  10,  10,  10,  10,  10,  10,  10,  10}, //CWF
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   2,   2,   2,   2,   2,   2,   2,   2}, //Daylight
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0}, //Shade
    {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   5,  10,  10,  10,  10,  10,  10,  10,  10}, //DF
},
```





AWB Noise Reduction (2/2)

- Case
  - Outdoor soil scene with Noise Reduction to reduce Daylight Locus Probability



EXIF

AWB_TAG_PB_NUM_THR_NONNEUTRAL	:	43
AWB_TAG_PB_NUM_THR_STB	:	0
AWB_TAG_PB_NUM_THR_T	:	0
AWB_TAG_PB_NUM_THR_WF	:	0
AWB_TAG_PB_NUM_THR_F	:	43
AWB_TAG_PB_NUM_THR_CWF	:	43
AWB_TAG_PB_NUM_THR_D	:	9
AWB_TAG_PB_NUM_THR_S	:	0
AWB_TAG_PB_NUM_THR_DF	:	43
AWB_TAG_NEUTRAL_PB_NUM_STB	:	0
AWB_TAG_NEUTRAL_PB_NUM_T	:	0
AWB_TAG_NEUTRAL_PB_NUM_WF	:	0
AWB_TAG_NEUTRAL_PB_NUM_F	:	34
AWB_TAG_NEUTRAL_PB_NUM_CWF	:	105
AWB_TAG_NEUTRAL_PB_NUM_D	:	285
AWB_TAG_NEUTRAL_PB_NUM_S	:	0
AWB_TAG_NEUTRAL_PB_NUM_DF	:	0
AWB_TAG_NEUTRAL_PB_NUM	:	424

EXIF

AWB_TAG_STA_GAIN_R_F	:	379
AWB_TAG_STA_GAIN_G_F	:	512
AWB_TAG_STA_GAIN_B_F	:	732
AWB_TAG_SPAT_GAIN_R_F	:	450
AWB_TAG_SPAT_GAIN_G_F	:	512
AWB_TAG_SPAT_GAIN_B_F	:	555
AWB_TAG_HIT_NR_F	:	1
AWB_TAG_DAYLIGHT_PROB_F	:	55
AWB_TAG_EQV_DAYLIGHT_PROB_F	:	43
AWB_TAG_EQV_GAIN_R_F	:	429
AWB_TAG_EQV_GAIN_G_F	:	512
AWB_TAG_EQV_GAIN_B_F	:	599

Sub Window: F Preference

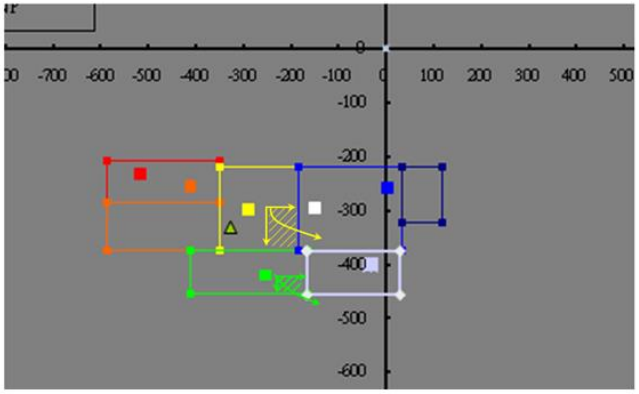
- Reduce the F daylight locus probability to keep more saturation.
- Condition
  - Parent block in F prefer region
  - LV is higher than active LV threshold

// SubWindow F Detection

```
{
    1,  // i4Enable
    1,  // i4Method
   50,  // i4LVThr_L
   90,  // i4LVThr_H
  128,  // i4DaylightProb
  {
    -177,  // Area X
    -437,  // Area Y
  },
  {
     -85,  // Vertex X
    -453,  // Vertex Y
  },
},
```

EXIF

AWB_TAG_STA_GAIN_R_F : 386
AWB_TAG_STA_GAIN_G_F : 512
AWB_TAG_STA_GAIN_B_F : 707
AWB_TAG_SPAT_GAIN_R_F : 483
AWB_TAG_SPAT_GAIN_G_F : 512
AWB_TAG_SPAT_GAIN_B_F : 510
AWB_TAG_HIT_NR_F : 0
AWB_TAG_DAYLIGHT_PROB_F : 62
AWB_TAG_EQV_DAYLIGHT_PROB_F : 61
AWB_TAG_EQV_GAIN_R_F : 422
AWB_TAG_EQV_GAIN_G_F : 512
AWB_TAG_EQV_GAIN_B_F : 621
AWB_TAG_SHADE_XR_F : -4
AWB_TAG_SHADE_YR_F : -445
AWB_TAG_SHADE_COUNT_F : 7
AWB_TAG_SHADE_TARGET_RPOB_F : 61



Weak — Strong

x 1.0 (ratio = 256)

x 0.5 (ratio = 128)

x 0 (ratio = 0)

Sub Window: CWF Preference

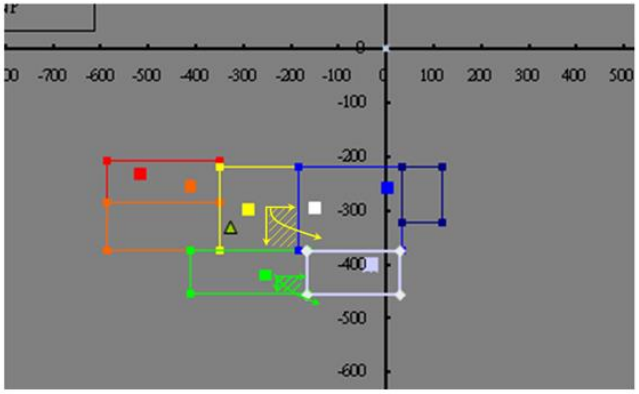
- Reduce the CWF daylight locus probability to keep more saturation.
- Condition
  - Parent block in CWF prefer region
  - LV is higher than active LV threshold

// SubWindow CWF Detection

```
{
    1,   // i4Enable
    1,   // i4Method
   50,   // i4LVThr_L
   90,   // i4LVThr_H
  192,   // i4DaylightProb
   {
     -177,  // Area X
     -507,  // Area Y
   },
   {
      -82,  // Vertex X
     -543,  // Vertex Y
   },
},
```

EXIF

AWB_TAG_STA_GAIN_R_CWF : 433
AWB_TAG_STA_GAIN_G_CWF : 512
AWB_TAG_STA_GAIN_B_CWF : 813
AWB_TAG_SPAT_GAIN_R_CWF : 483
AWB_TAG_SPAT_GAIN_G_CWF : 512
AWB_TAG_SPAT_GAIN_B_CWF : 510
AWB_TAG_HIT_NR_CWF : 1
AWB_TAG_DAYLIGHT_PROB_CWF : 42
AWB_TAG_EQV_DAYLIGHT_PROB_CWF : 18
AWB_TAG_EQV_GAIN_R_CWF : 473
AWB_TAG_EQV_GAIN_G_CWF : 512
AWB_TAG_EQV_GAIN_B_CWF : 554
AWB_TAG_SHADE_XR_CWF : -37
AWB_TAG_SHADE_YR_CWF : -512
AWB_TAG_SHADE_COUNT_CWF : 4
AWB_TAG_SHADE_TARGET_PROB_CWF : 18

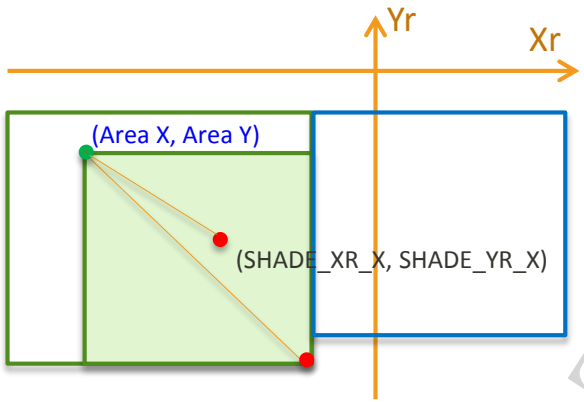


Weak — Strong

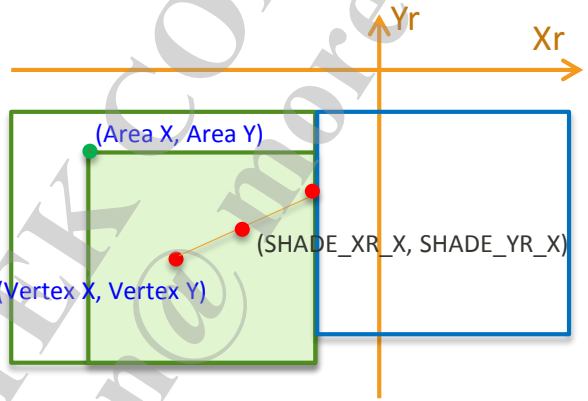
x 1.0 (ratio = 256) | x 0.5 (ratio = 128) | x 0 (ratio = 0)

Sub Window: F/CWF Preference

- Method
  - i4Method = 0, 越靠近右下角,強度越強
  - i4Method = 1, 越靠近中心點越高,強度越強



i4Method 0: Closer to the lower right corner, the stronger the intensity



i4Method 1: The closer the center point is, the stronger the strength

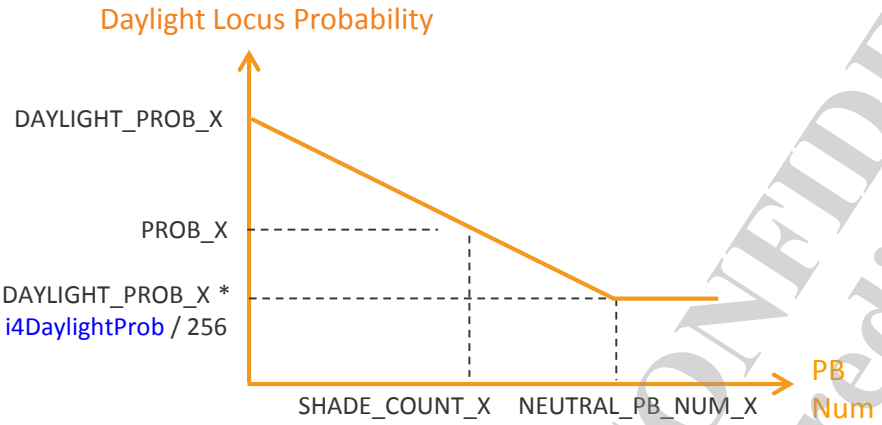
// SubWindow F Detection

```
{
    1,   // i4Enable
    1,   // i4Method
   50,   // i4LVThr_L
   90,   // i4LVThr_H
  128,   // i4DaylightProb
  {
    -177,  // Area X
    -437,  // Area Y
  },
  {
     -85,  // Vertex X
    -453,  // Vertex Y
  },
},
```

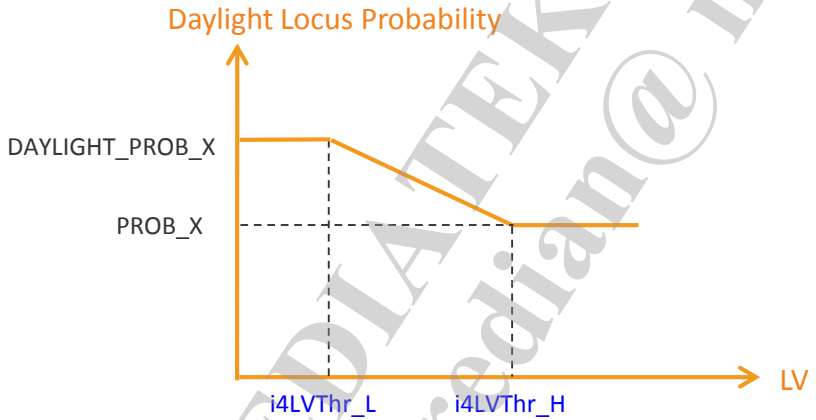
EXIF

```
AWB_TAG_NEUTRAL_PB_NUM_F : 99
AWB_TAG_NEUTRAL_PB_NUM_CWF : 18
AWB_TAG_SHADE_XR_F : -4
AWB_TAG_SHADE_YR_F : -445
AWB_TAG_SHADE_COUNT_F : 7
AWB_TAG_SHADE_TARGET_RPOB_F : 61
AWB_TAG_SHADE_XR_CWF : -37
AWB_TAG_SHADE_YR_CWF : -512
AWB_TAG_SHADE_COUNT_CWF : 4
AWB_TAG_SHADE_TARGET_PROB_CWF : 18
```

Sub Window: F/CWF Preference



Subwindow數量比例越高，spatial gain強度越強



亮度越高， spatial gain強度越強

```
// SubWindow F Detection
{
     1,   // i4Enable
     1,   // i4Method
    50,   // i4LVThr_L
    90,   // i4LVThr_H
   128,  // i4DaylightProb
   {
      -177,  // Area X
      -437,  // Area Y
   },
   {
       -85,  // Vertex X
      -453,  // Vertex Y
   },
},
```

EXIF

```
AWB_TAG_NEUTRAL_PB_NUM_F : 99
AWB_TAG_NEUTRAL_PB_NUM_CWF : 18
AWB_TAG_SHADE_XR_F : -4
AWB_TAG_SHADE_YR_F : -445
AWB_TAG_SHADE_COUNT_F : 7
AWB_TAG_SHADE_TARGET_RPOB_F : 61
AWB_TAG_SHADE_XR_CWF : -37
AWB_TAG_SHADE_YR_CWF : -512
AWB_TAG_SHADE_COUNT_CWF : 4
AWB_TAG_SHADE_TARGET_PROB_CWF : 18
```

Preference Gain

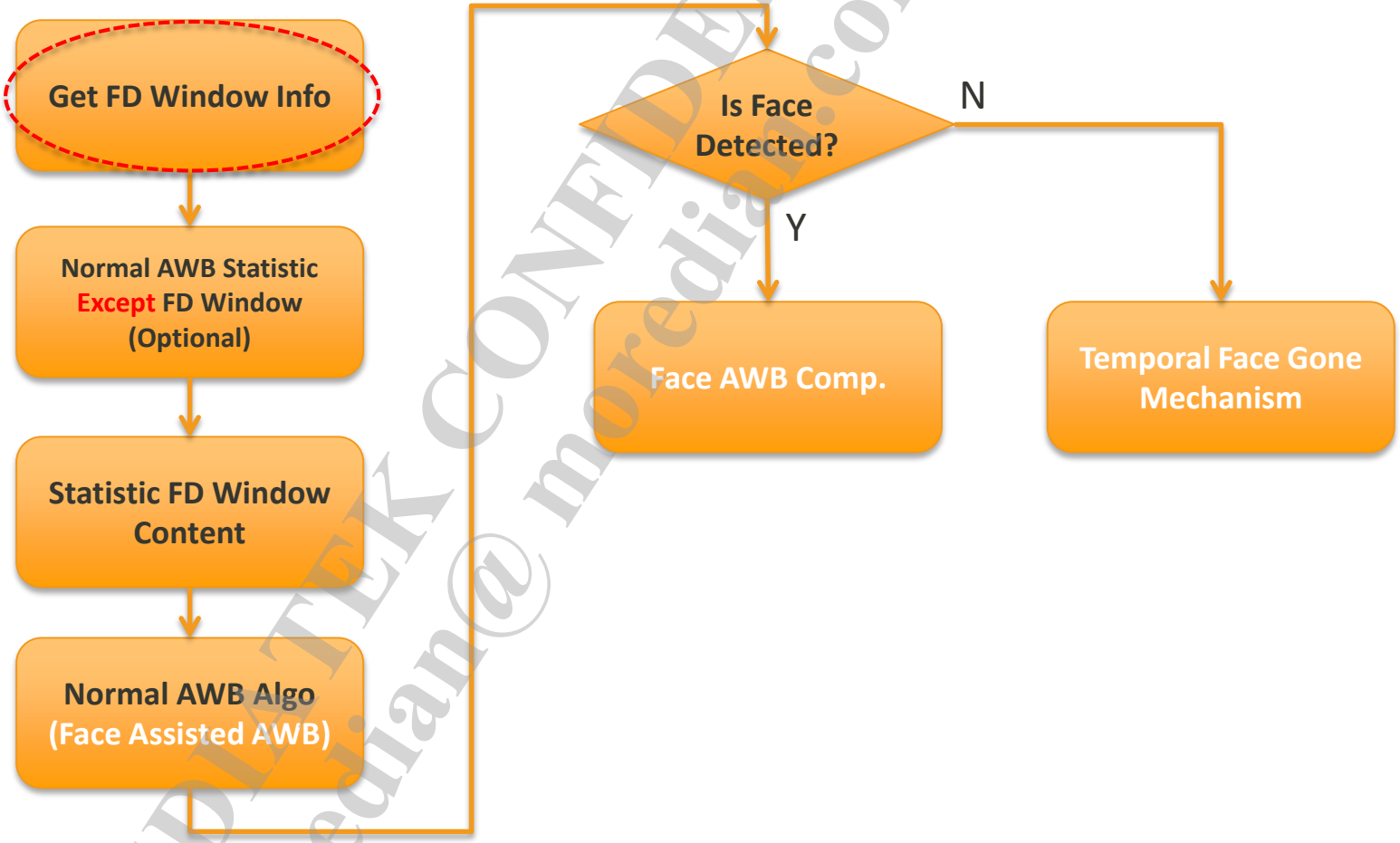
Preference Gain

- Adjust the AWB gain for different light source and LV

```
// Preference gain for each light source
{
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // STROBE
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {532, 512, 522}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // TUNGSTEN
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {516, 512, 516}, {546, 512, 528}, {568, 512, 536}, {548, 512, 520}, {524, 512, 518}, {534, 512, 524}, {534, 512, 514}, {535, 512, 516}, {522, 512, 504},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // WARM F
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {520, 512, 540}, {518, 512, 520}, {514, 512, 502}, {526, 512, 522}, {522, 512, 502}, {512, 512, 502}, {512, 512, 502}, {512, 512, 512}, {512, 512, 514}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {508, 512, 512}, {502, 512, 518}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}
        }, // F
```

Face Assisted AWB

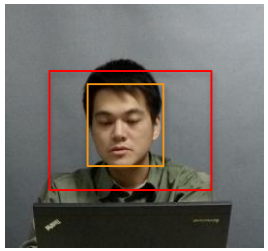
AWB Algo Flow



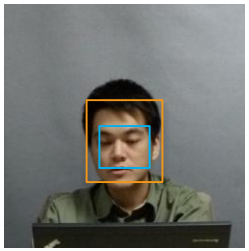
Get FD Window Info

- FD window can be scaled by tuning parameter.

Scaling up 200%



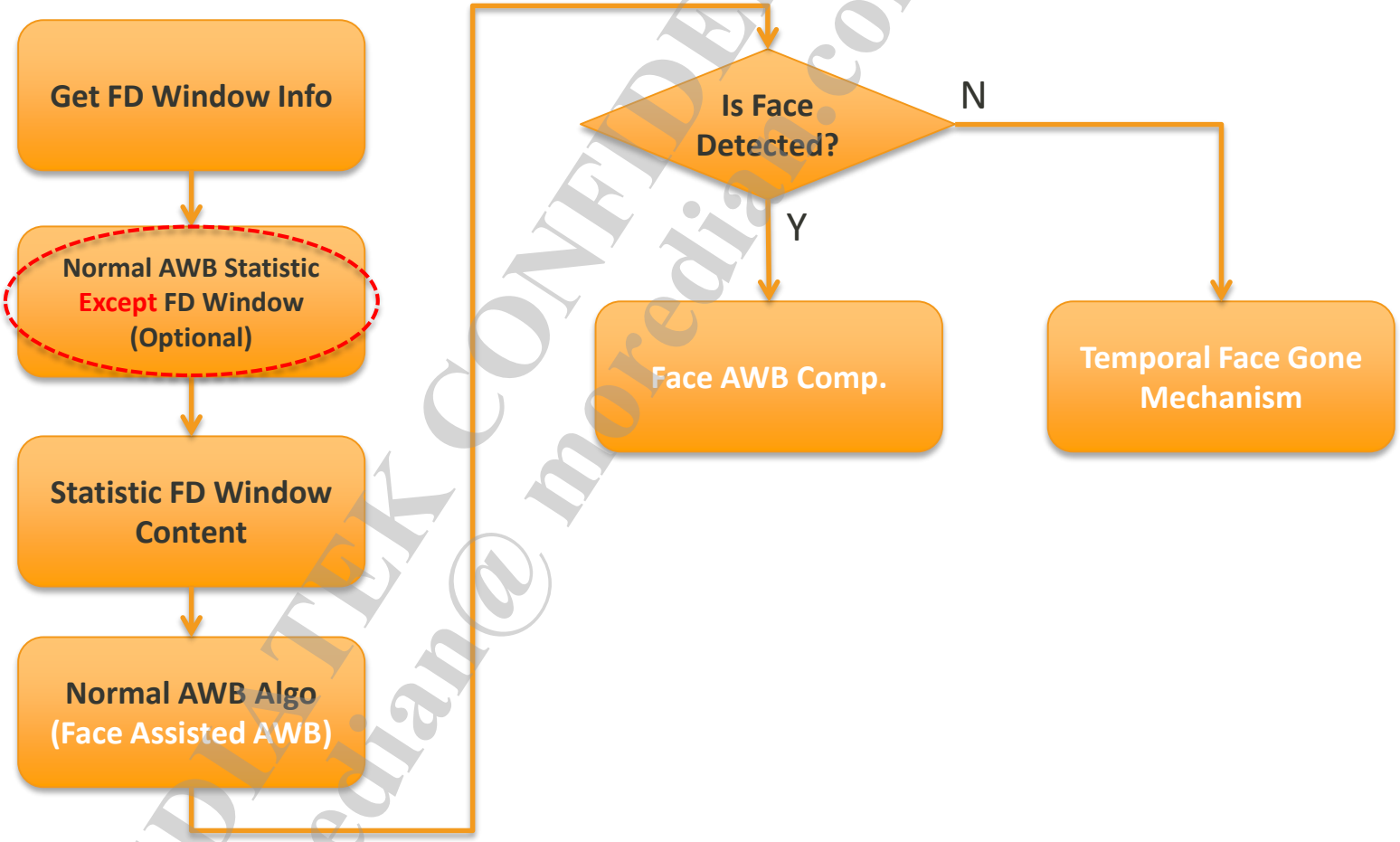
Scaling down 50%



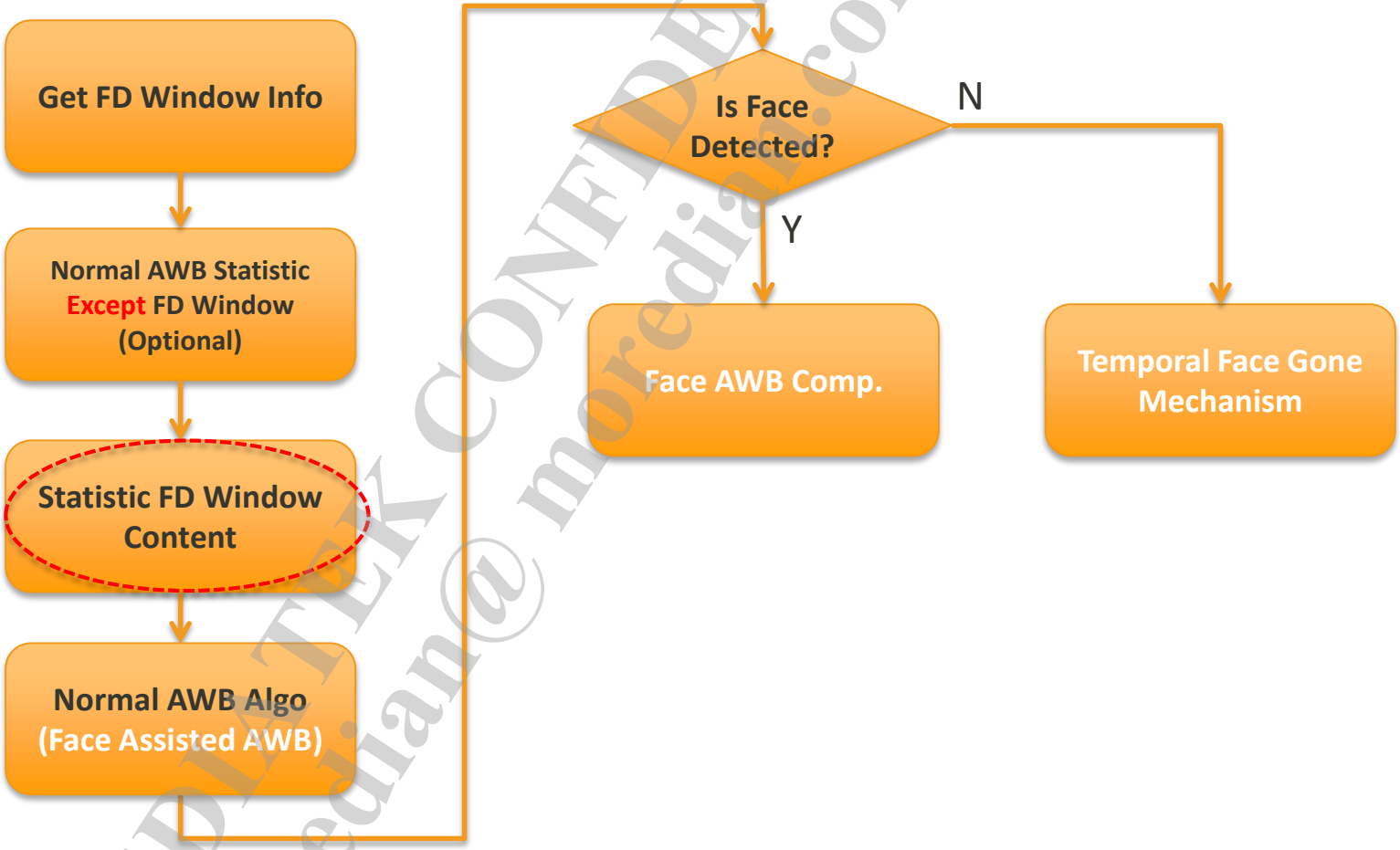
Scaling Form	Pros	Cons
Scaling up	Smooth/Consistence	Worse accuracy
Scaling down	Accuracy	Damping

```
1,      //i4StatAvoidFaceArea
100,    //i4FaceWinRatio  50~400
1,      //u4FaceCentralWeight  0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
// rStatisticNR
{
    1,      //i4Enable
    200     //i4DiffThr
},
```

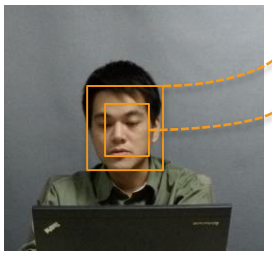
AWB Algo Flow



AWB Algo Flow



Statistic FD Window Content



$$FaceAvg\_R = \frac{(Central\_R \times Central\_Weight) + (1 \times ScaledWindow\_R)}{(Central\_Weight + 1)}$$

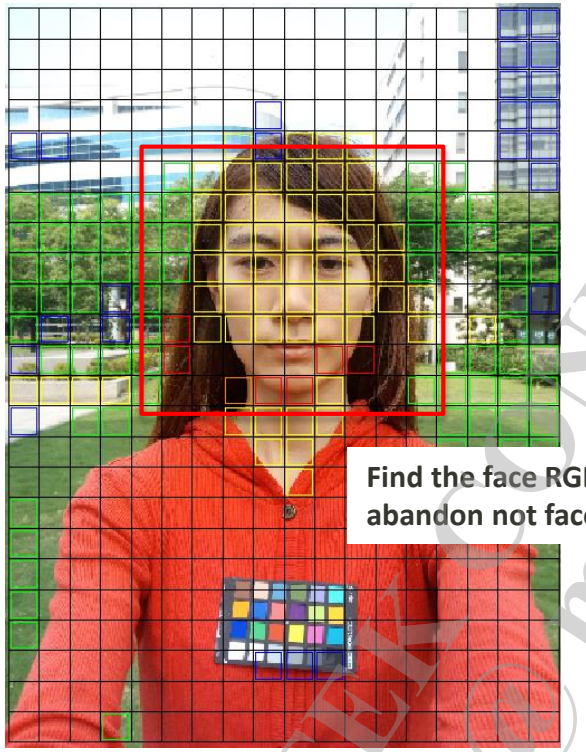
Central Weight	Behavior
small	Focus on default size
large	Focus on skin, more accuracy, but might be more unstable

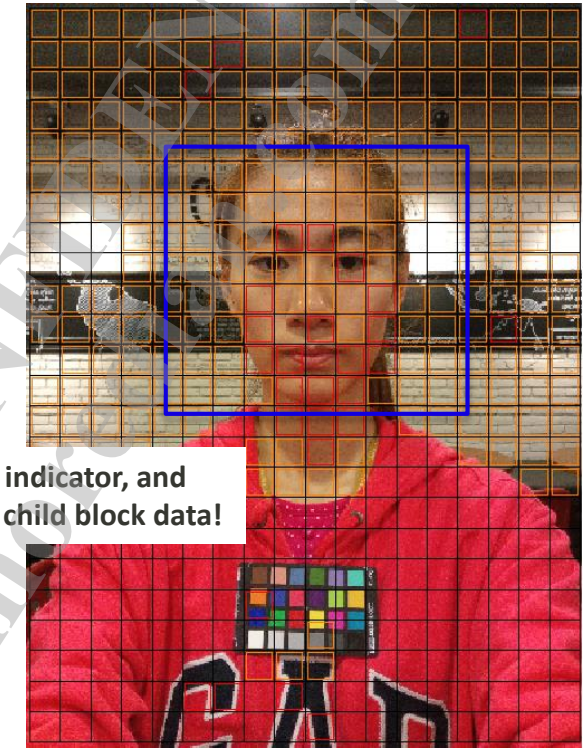
Final output:
Weighted average Face RGB.
(Weighed by face area size)

```
1,      //u4FaceCentralWeight  0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
// rStatisticNR
{
    1,      //i4Enable
    200     //i4DiffThr
},
// rFD_RGB_Bound
{
      1,    //u4LowBound
    254     //u4HiBound
},
```

If one channel of face RGB is out of range, then abandon this face info

Face Statistic NR Process





Find the face RGB indicator, and abandon not face child block data!

```
1,      //u4FaceCentralWeight  0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
// rStatisticNR
{
    1,      //i4Enable
    200     //i4DiffThr
},
// rFD_RGB_Bound
{
    1,    //u4LowBound
    254   //u4HiBound
},
```

If diff is larger than thr, then abandon this child block for avoiding face statistic.

AWB v5.0

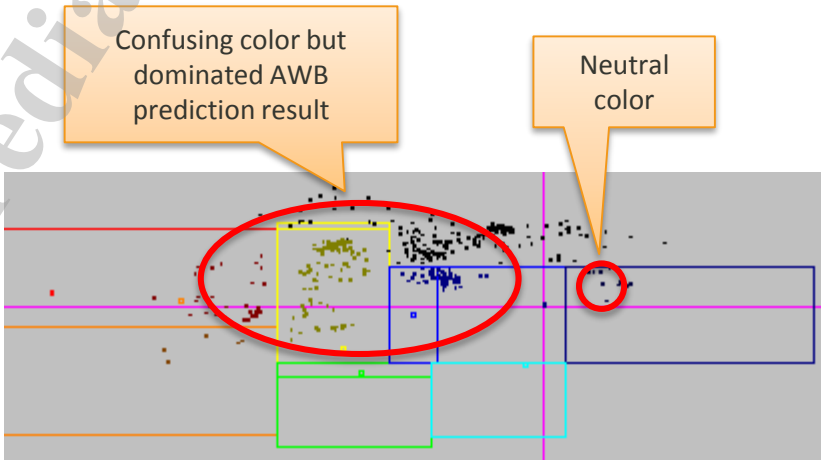
Face Assisted AWB

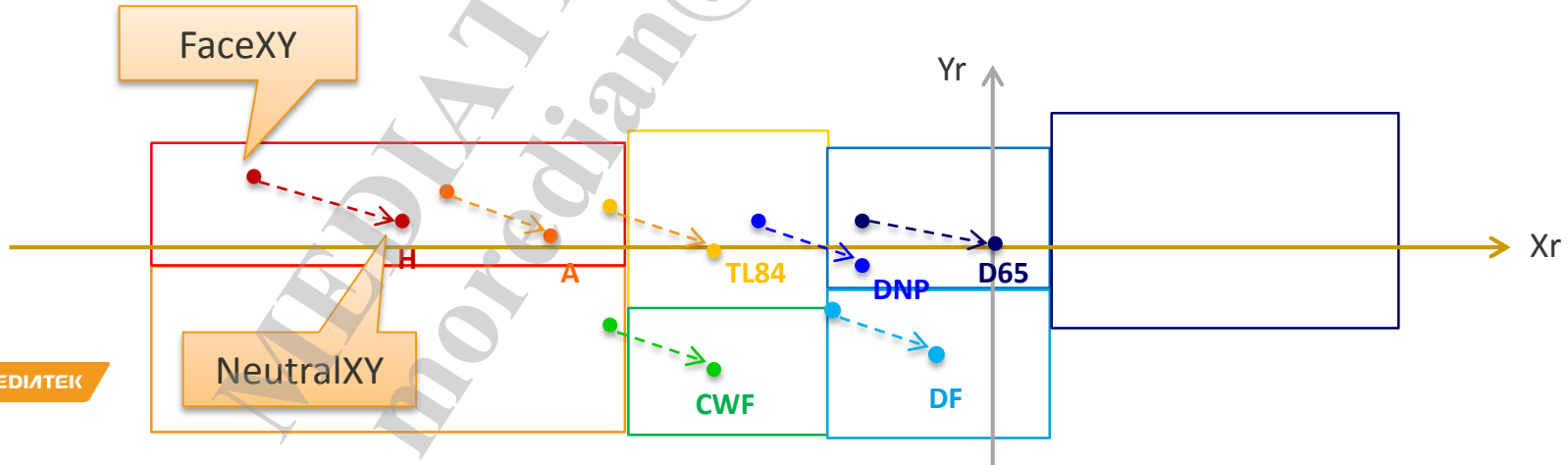
```
// Face Assisted AWB
{
    1,   //i4Enable
},
```

- Purpose
  - Using face color to predict light source/gain to assist normal AWB prediction
  - To improve accuracy even in confusing-color scenes

Correct AWB result

Incorrect AWB Result

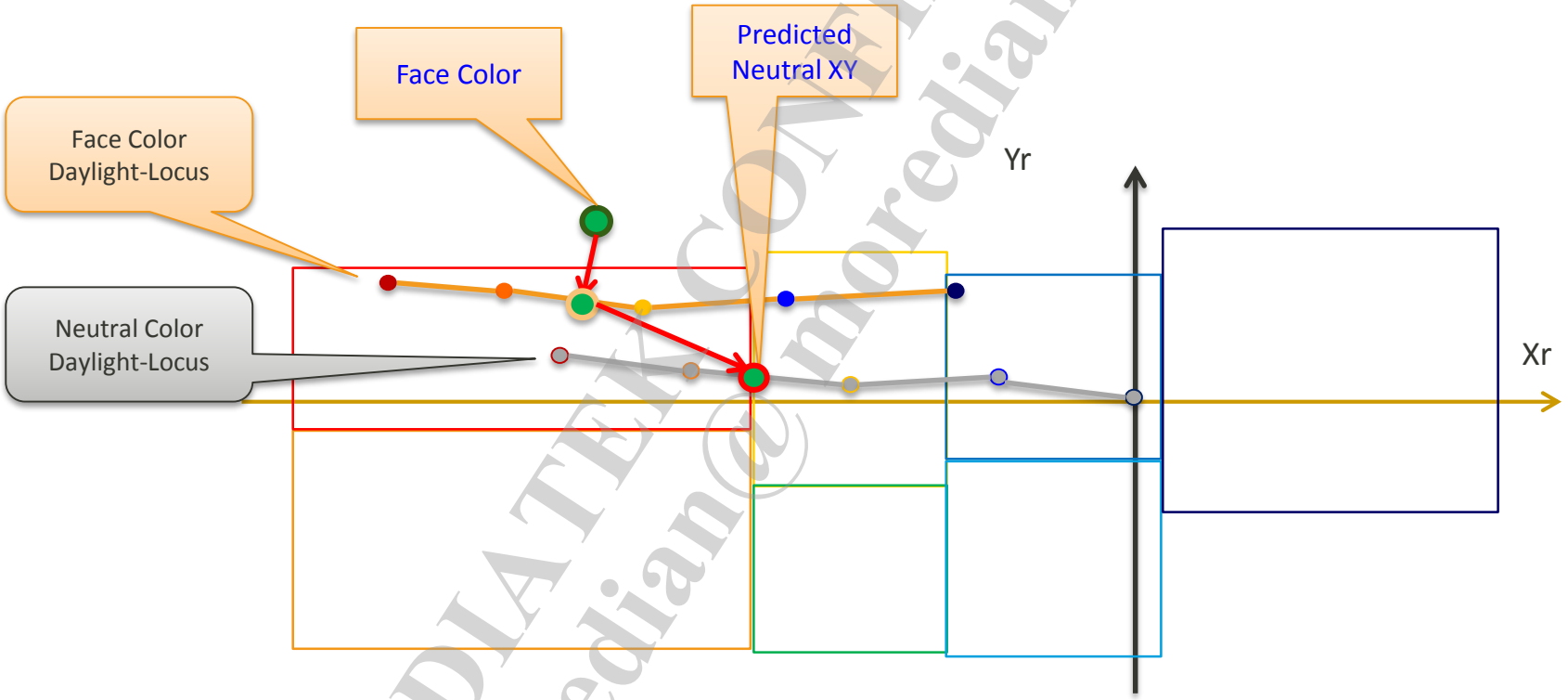




AWB
v5.0

Face Assisted AWB

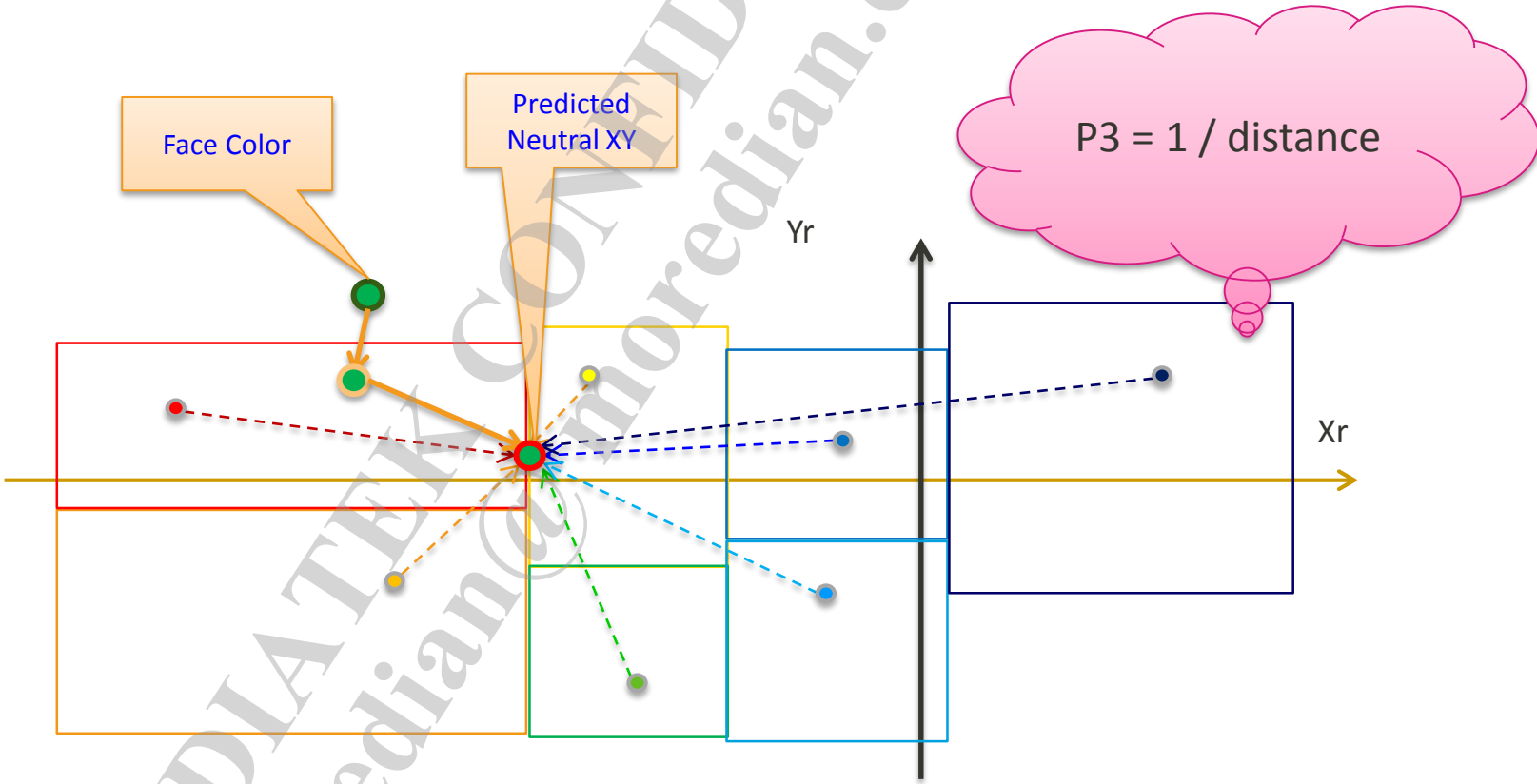
- P3 is the probability according face skin color
  - The XrYr difference between light source average and face predictor.



AWB
v5.0

Face Assisted AWB

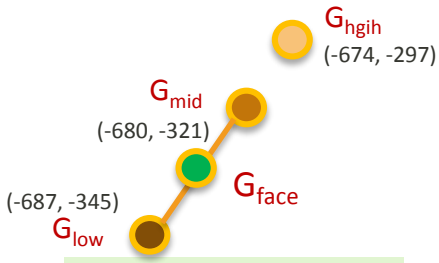
- P3 is the probability according face skin color
  - The XrYr difference between light source average and face predictor.



AWB v5.0

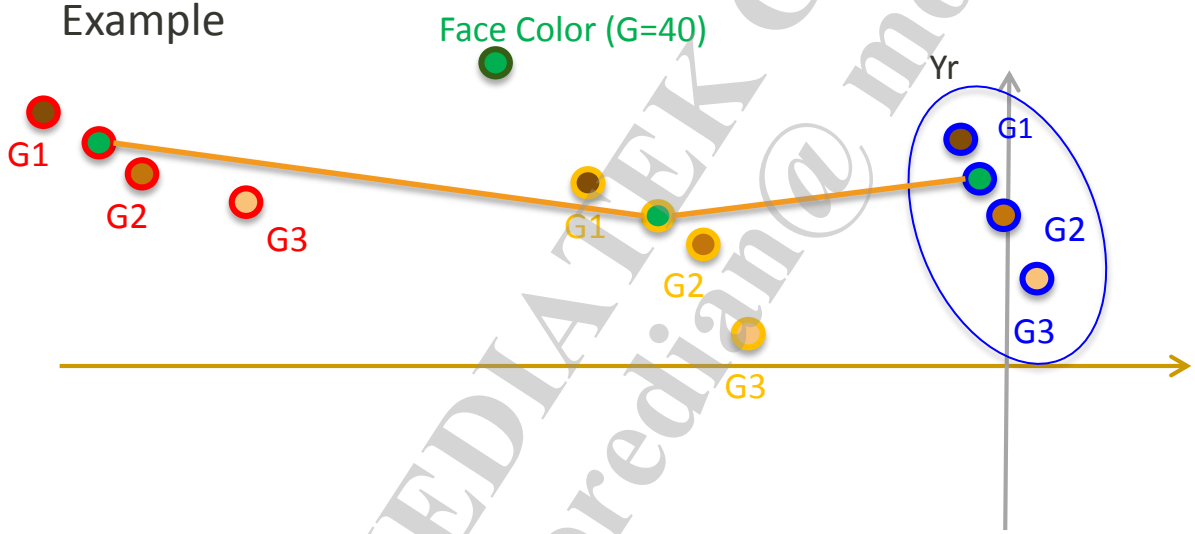
Face Assisted AWB - Predictor Estimation

- Reference Target Selection
  - Use face color “G” channel to define different face color under different light sources
  - Get reference target XY for each light sources
  - Blending all reference targets by its weighting




```
// i4RefTargetThr
{
   32,   //ThrLow
   64,   //ThrMid
   96,   //ThrHigh
},
```

Example



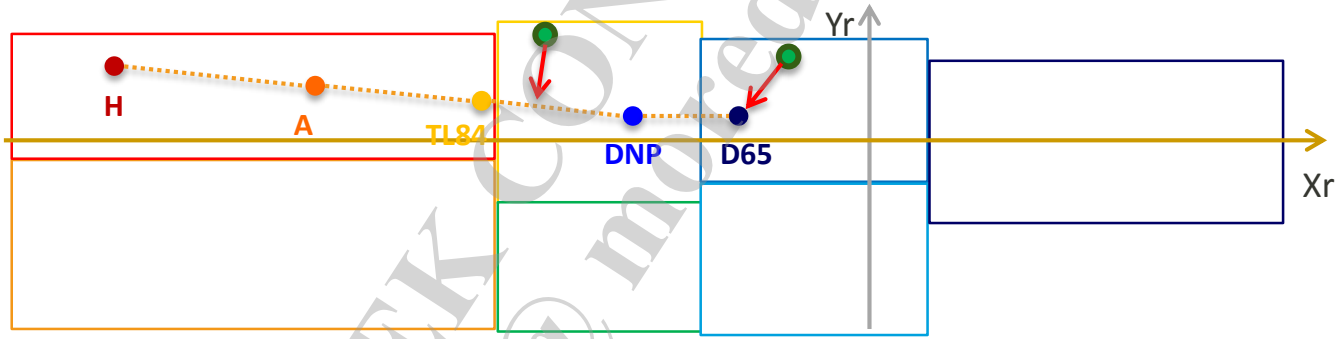

```
// i4RefTargetXY
{
   // ThrLow          ThrMid          ThrHigh
   {{    0,      0}, {    0,      0}, {    0,      0}},  // STB
   {{-665, -406}, {-665, -406}, {-665, -406}},  // T
   {{-505, -415}, {-505, -415}, {-505, -415}},  // WF
   {{-284, -450}, {-284, -450}, {-284, -450}},  // F
   {{-276, -592}, {-276, -592}, {-276, -592}},  // CWF
   {{-163, -414}, {-163, -414}, {-163, -414}},  // D
   {{   -7, -414}, {   -7, -414}, {   -7, -414}},  // S
   {{ -24, -552}, {  -24, -552}, {  -24, -552}},  // DF
},
```

AWB v5.0

Face Assisted AWB - Predictor Estimation

- Weighting function for reference target estimation
  - **Mode 0**
    - Only reference those light source located on ideal daylight locus
      - D65, DNP, TL84, A, HOR light source
    - Projection on line of daylight-locus simply

```
// rRefTarget
{
    0,   //i4Mode
  100,   //i4WeightCoef_a
  100,   //i4WeightCoef_b
},
```



EXIF

AWB_TAG_FACEAST_DIST_WEI_H : 0
AWB_TAG_FACEAST_DIST_WEI_A : 0
AWB_TAG_FACEAST_DIST_WEI_TL84 : 73
AWB_TAG_FACEAST_DIST_WEI_CWF : 0
AWB_TAG_FACEAST_DIST_WEI_DNP : 27
AWB_TAG_FACEAST_DIST_WEI_D65 : 0
AWB_TAG_FACEAST_DIST_WEI_DF : 0

EXIF

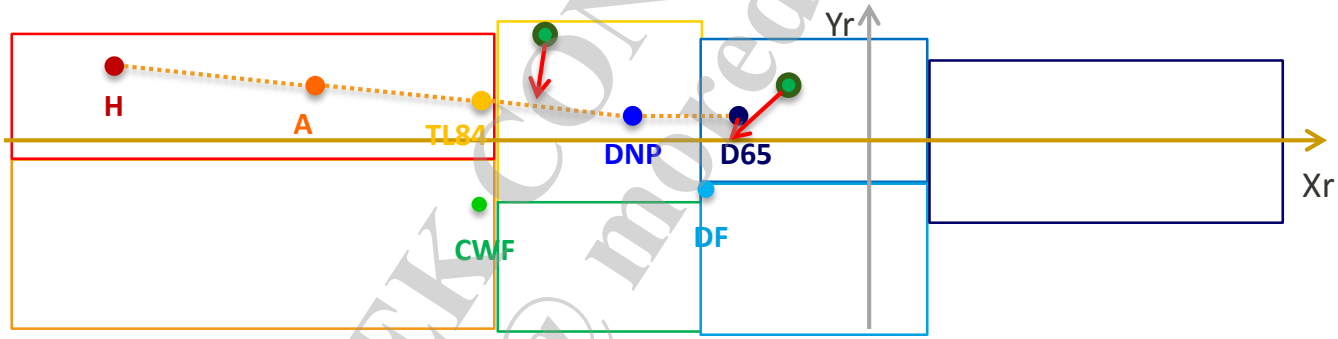
AWB_TAG_FACEAST_DIST_WEI_H : 0
AWB_TAG_FACEAST_DIST_WEI_A : 0
AWB_TAG_FACEAST_DIST_WEI_TL84 : 0
AWB_TAG_FACEAST_DIST_WEI_CWF : 0
AWB_TAG_FACEAST_DIST_WEI_DNP : 0
AWB_TAG_FACEAST_DIST_WEI_D65 : 100
AWB_TAG_FACEAST_DIST_WEI_DF : 0

AWB v5.0

Face Assisted AWB - Predictor Estimation

- Weighting function for reference target estimation
  - **Mode 1**
    - Reference all light sources and calculated by distance by
      - D65, DNP, TL84, A, Horizon, CWF and DF light sources
      - Can be controlled by two coefs (coef_a / coef_b)

```
// rRefTarget
{
    1,   //i4Mode
  100,   //i4WeightCoef_a
  100,   //i4WeightCoef_b
},
```



EXIF

AWB_TAG_FACEAST_DIST_WEI_H : 0
AWB_TAG_FACEAST_DIST_WEI_A : 0
AWB_TAG_FACEAST_DIST_WEI_TL84 : 78
AWB_TAG_FACEAST_DIST_WEI_CWF : 0
AWB_TAG_FACEAST_DIST_WEI_DNP : 22
AWB_TAG_FACEAST_DIST_WEI_D65 : 0
AWB_TAG_FACEAST_DIST_WEI_DF : 0

EXIF

AWB_TAG_FACEAST_DIST_WEI_H : 0
AWB_TAG_FACEAST_DIST_WEI_A : 0
AWB_TAG_FACEAST_DIST_WEI_FTL84: 0
AWB_TAG_FACEAST_DIST_WEI_CWF : 0
AWB_TAG_FACEAST_DIST_WEI_DNP : 4
AWB_TAG_FACEAST_DIST_WEI_D65 : 66
AWB_TAG_FACEAST_DIST_WEI_DF : 29

AWB v5.0

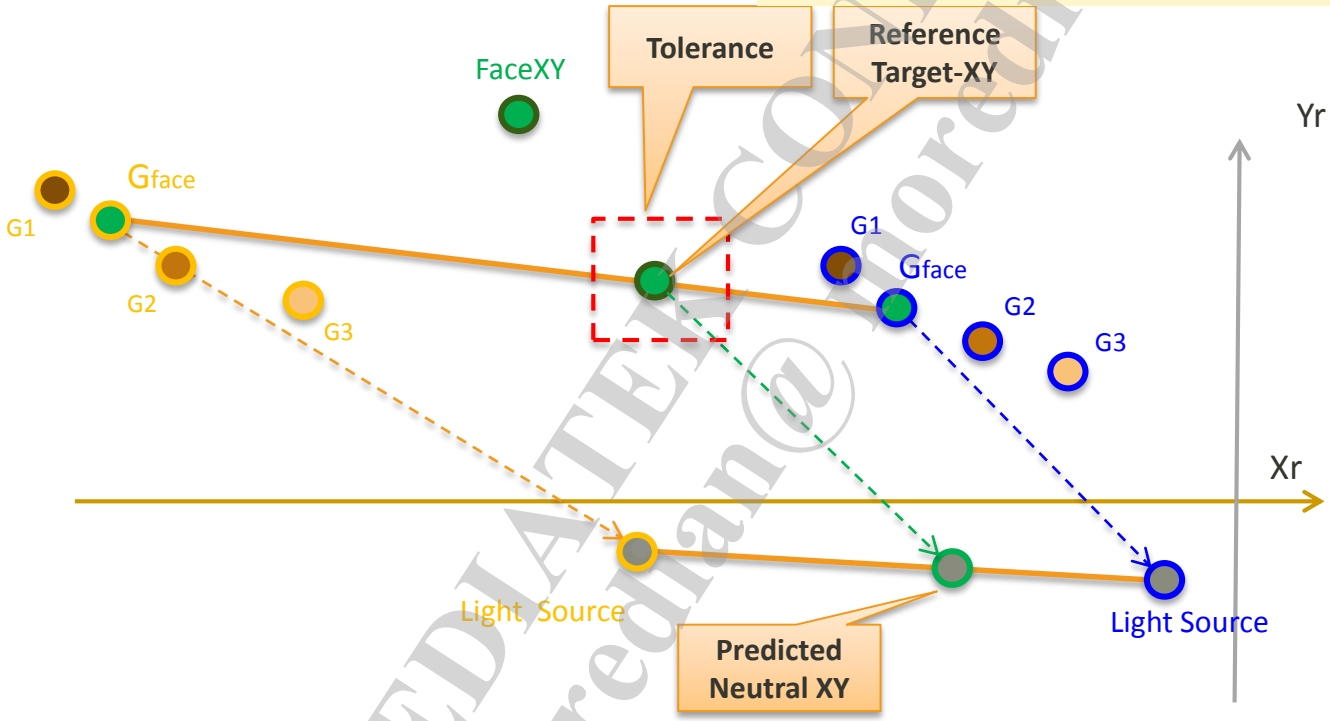
Face Assisted AWB - Predictor Estimation

- Face prediction result by weighting function
  - (1) Reference target-XY
  - (2) Tolerance
  - (3) Predicted neutral-XY

EXIF

AWB_TAG_FACEAST_REF_TARGET_XR : -313
AWB_TAG_FACEAST_REF_TARGET_YR : -392
AWB_TAG_FACEAST_TARGET_TOL : 31
AWB_TAG_FACEAST_PREDICT_XR : -107
AWB_TAG_FACEAST_PREDICT_YR : -501

```
//i4TOL
{
   32,   //STB
   36,   //H
   36,   //A
   32,   //TL84
   30,   //CWF
   28,   //DNP
   24,   //D65
   24,   //DF
},
```



EXIF

AWB_NVRAM_HORIZON_XR : -516
AWB_NVRAM_HORIZON_YR : -428
AWB_NVRAM_A_XR : -344
AWB_NVRAM_A_YR : -440
AWB_NVRAM_TL84_XR : -127
AWB_NVRAM_TL84_YR : -512
AWB_NVRAM_CWF_XR : -103
AWB_NVRAM_CWF_YR : -547
AWB_NVRAM_DNP_XR : -35
AWB_NVRAM_DNP_YR : -460
AWB_NVRAM_D65_XR : 140
AWB_NVRAM_D65_YR : -445
AWB_NVRAM_DF_XR : 114
AWB_NVRAM_DF_YR : -535

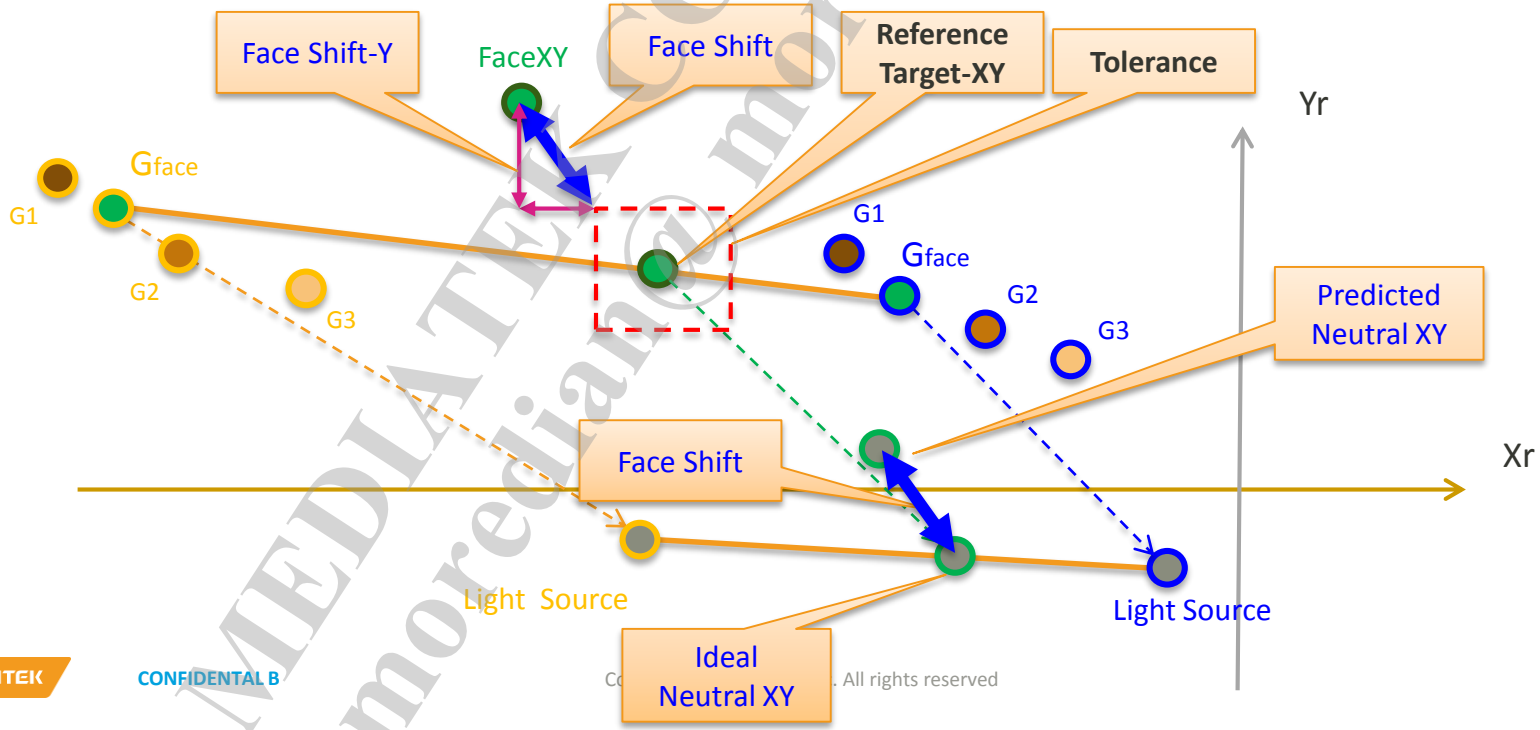
AWB v5.0

Face Assisted AWB - Predictor Estimation

- Face prediction result by weighting function
  - (4) Predicted-Neutral-XY
    - Ideal-Neutral-XY + Face-Shift-XY for color preservation
    - **Face-Shift XY**
      - Face color with bigger Face-Shift-Y value for confusing face color detection
      - have higher probability face is under abnormal illuminates (away from daylight-locus)
      - wrong statistic data on face

EXIF

AWB_TAG_FACEAST_FACE_XR : -308
AWB_TAG_FACEAST_FACE_YR : -363
AWB_TAG_FACEAST_REF_TARGET_XR : -313
AWB_TAG_FACEAST_REF_TARGET_YR : -392
AWB_TAG_FACEAST_TARGET_TOL : 31
AWB_TAG_FACEAST_FACE_SHIFT : 0
AWB_TAG_FACEAST_PREDICT_XR : -107
AWB_TAG_FACEAST_PREDICT_YR : -501



AWB v5.0

Face Assisted AWB - Predictor Estimation

- Face prediction result by weighting function
  - (5) Preference Gain - Preference gain provided flexible tuning on face assisted predicted gain
    - Where predicted gain is converted from Predicted-Neutral-XY by domain convert

EXIF

AWB_TAG_FACEAST_PREDICT_GAIN_R : 407
AWB_TAG_FACEAST_PREDICT_GAIN_B : 784
AWB_TAG_FACEAST_PREDICT_PREFER_GAIN_R : 512
AWB_TAG_FACEAST_PREDICT_PREFER_GAIN_B : 512

```
//rPrefGain
{
  { 512,  512}, // STB
  { 512,  512}, // H
  { 512,  512}, // A
  { 512,  512}, // TL84
  { 512,  512}, // CWF
  { 512,  512}, // DNP
  { 512,  512}, // D65
  { 512,  512}, // DF
},
```

[Tips]

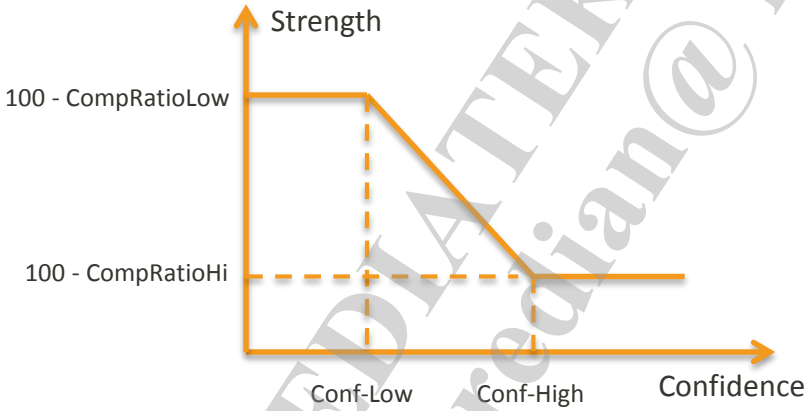
1. 單獨要調某個亮度或FaceG，且不希望影響其他二組reference target
   → 修改Reference Target XY (i4RefTargetXY)
2. 調整某一個光源下的整體色調或Preference
   → 修改Preference Gain (rPrefGain)
   請勿直接修改”Rotated XY coordinate of AWB light source”，此為calibration result

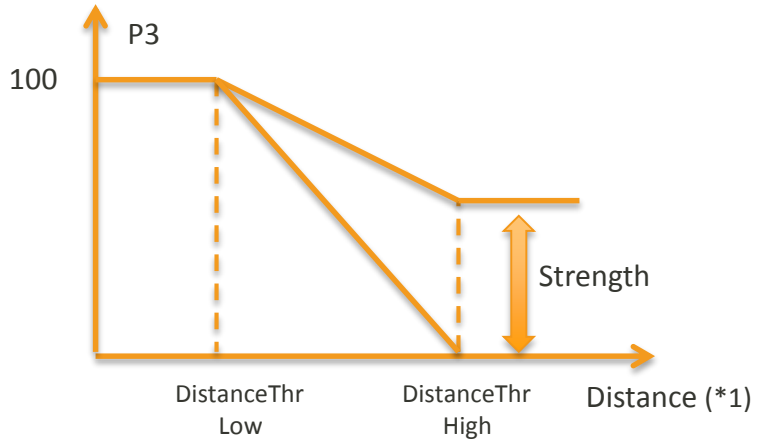
AWB v5.0

Face Assisted AWB - Predictor Probability

- P3 is the probability according
  - The strength is determined by following factors
    - Conf_P0: face size ratio
    - Conf_P1: Scene LV (environment brightness)
    - Conf_P2: Face shift range (confusing)
    - Conf_P3: Face temporal information (stable confidence)
    - Conf_P4: Face color ratio if statistic-NR enable
  - Confidence = Conf_P0 * Conf_P1 * Conf_P2 * Conf_P3 * Conf_P4
  - P3 = LUT(Distance)

```
// rCompSetting
{
    // Low  High
    {   0,  100 }, //rConfThr
    {  50,  200 }, //rDistanceThr
    {   0,   60 }, //rCompRatio
},
```



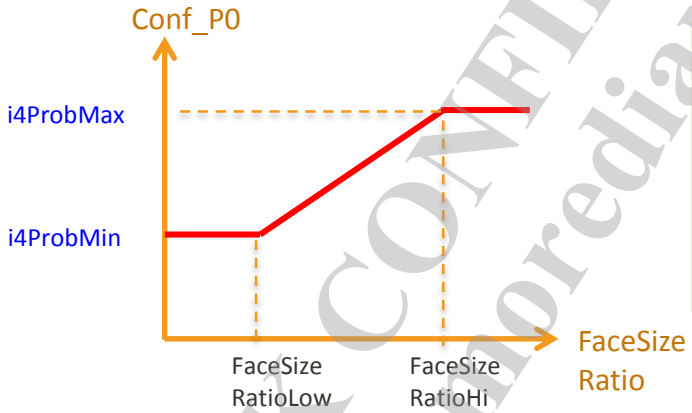


***1. Distance: distance between EqvXY of light-src and NeutralXY of Face-Predictor**

AWB v5.0

Face Assisted AWB - Predictor Probability

- Conf_P0 - Probability of face size ratio
  - Leverage face-comp v1.5 design of face size calculation




```
// rProb0 - FaceSize
{
    3200,  // i4FaceSizeRatioLow
    6000,  // i4FaceSizeRatioHi
       0,   //i4ProbMin
     100,   //i4ProbMax
},
```

EXIF

AWB_TAG_FACEAST_SIZE_RATIO : 18750
AWB_TAG_FACEAST_CONF_PROB0 : 100

AWB v5.0

Face Assisted AWB - Predictor Probability

- Conf_P1 - Probability of scene luminance level
  - Similar to P1 probability of normal AWB prediction, but control P3 strength

```
// rProb1 - LV
{
  //LV0  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15  16  17  18
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // STROBE
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // H
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // A
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // TL84
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40 }, //CWF
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // DNP
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // D65
  { 50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40 }, //DF
},
```

EXIF

AWB_TAG_ALGO_SCENE_LV : 61
AWB_TAG_FACEAST_CONF_PROB1_STB : 0
AWB_TAG_FACEAST_CONF_PROB1_T : 100
AWB_TAG_FACEAST_CONF_PROB1_WF : 100
AWB_TAG_FACEAST_CONF_PROB1_F : 100
AWB_TAG_FACEAST_CONF_PROB1_CWF : 100
AWB_TAG_FACEAST_CONF_PROB1_D : 100
AWB_TAG_FACEAST_CONF_PROB1_S : 100
AWB_TAG_FACEAST_CONF_PROB1_DF : 100

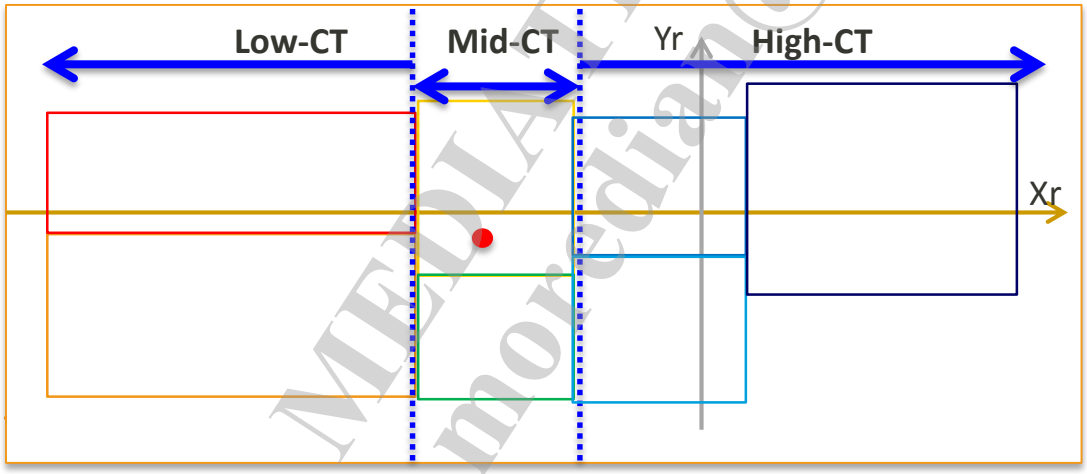
AWB v5.0

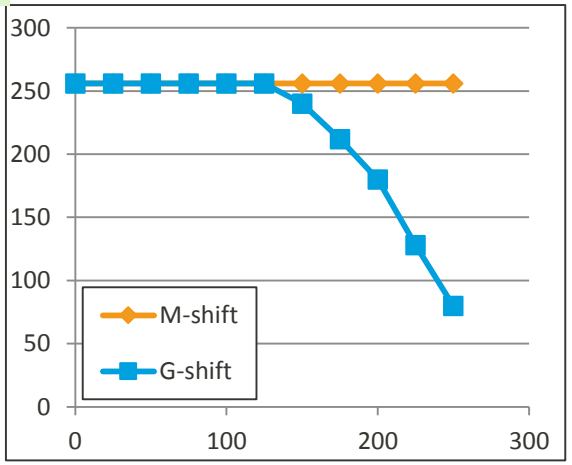
Face Assisted AWB - Predictor Probability

- Conf_P2 - Probability of confusing scene by face-shift-Y
  - Similar to green/magenta offset of daylight-locus offset model, but simple
  - “Face-Shift” value is distance in Y direction only
  - Bigger “FaceShift” value means FaceXY is unreliable, b’cz it is far-away from daylight locus

```
//rProb2
{
    // 0   25   50   75  100  125  150  175  200  225  250
   {{ 256, 256, 256, 256, 256, 256, 256, 256, 240, 160, 128 },  // Hi-CT Magenta
    { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,   80 }}, // Hi-CT Green
   {{ 256, 256, 256, 256, 256, 256, 256, 256, 240, 160, 128 },  // Mid-CT Magenta
    { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,   80 }}, // Mid-CT Green
   {{ 256, 256, 256, 256, 240, 212, 180, 128,  96,  64,   40 },  // Low-CT Magenta
    { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,   80 }}, // Low-CT Green
},
```

EXIF
AWB_TAG_FACEAST_CONF_PROB2 : 100

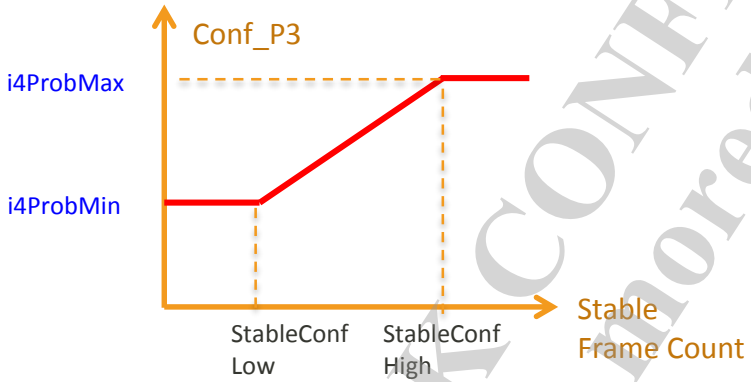




AWB v5.0

Face Assisted AWB - Predictor Probability

- Conf_P3 - Probability of stability confidence
  - If frame with face is detected continuously and stable, face assisted predictor have higher confidence value to get corrective P3 strength




```
//rProb3
{
    5,  // i4StableConfLow
   60,  // i4StableConfHi
    0,  // i4ProbMin
  100,  // i4ProbMax
},
```

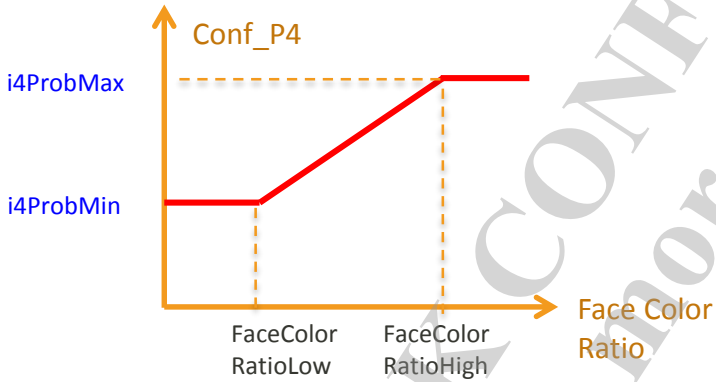
EXIF

AWB_TAG_FACEAST_CONF_PROB3 : 100

AWB v5.0

Face Assisted AWB - Predictor Probability

- Conf_P4 - Probability of valid face color ratio
  - Only if statistic-NR is enable

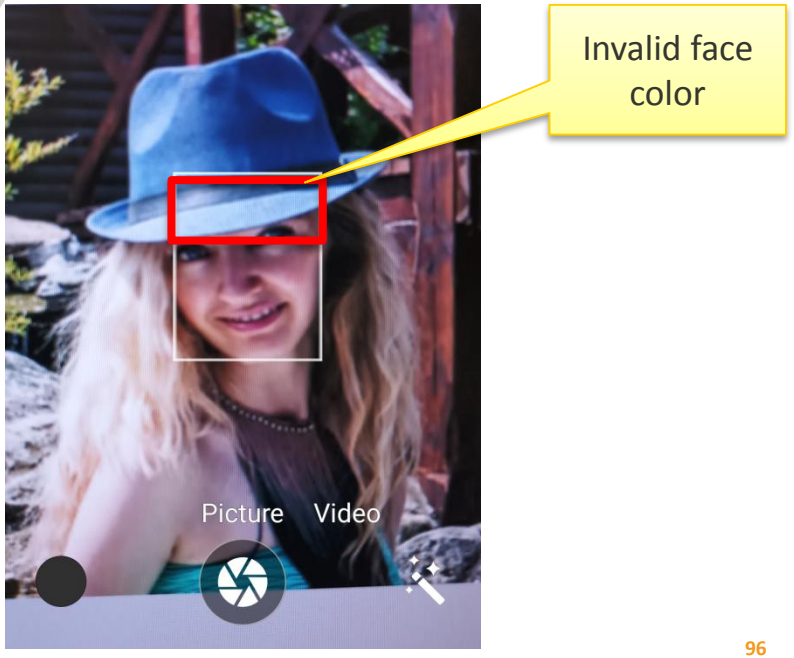



```
//rProb4
{
   200,  // i4FaceColorRatioLow
   800,  // i4FaceColorRatioHi
     0,  // i4ProbMin
   100,  // i4ProbMax
},
```

EXIF

AWB_TAG_FACEAST_FACECOLOR_RATIO : 833

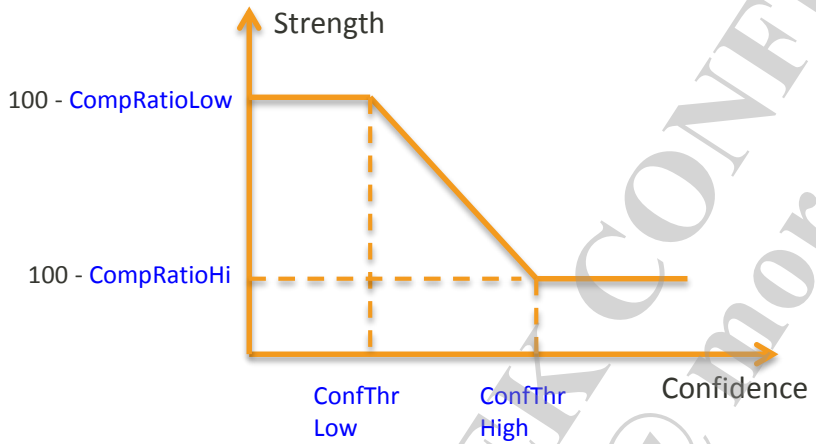
AWB_TAG_FACEAST_CONF_PROB4 : 100



AWB v5.0

Face Assisted AWB - Predictor Probability

- P3-Stregnth of each light source
  - Confidence(light) = Conf_P0 * Conf_P1(light) * Conf_P2 * Conf_P3 * Conf_P4
  - Get strength of each light source by its confidence value




```
// rCompSetting
{
  {  0, 100 }, //rConfThr
  { 50, 200 }, //rDistanceThr
  {  0,  40 },//rCompRatio
},
```

EXIF

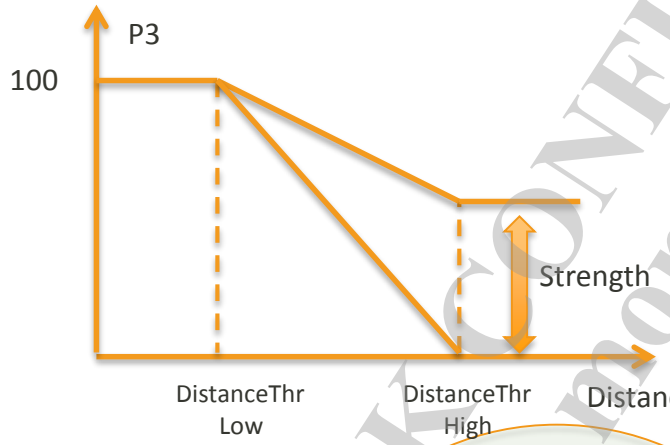
AWB_TAG_FACEAST_STRENGTH_STB : 0
AWB_TAG_FACEAST_STRENGTH_T : 60
AWB_TAG_FACEAST_STRENGTH_WF : 60
AWB_TAG_FACEAST_STRENGTH_F : 60
AWB_TAG_FACEAST_STRENGTH_CWF : 0
AWB_TAG_FACEAST_STRENGTH_D : 0
AWB_TAG_FACEAST_STRENGTH_DF : 0
AWB_TAG_FACEAST_STRENGTH_S : 0

AWB v5.0

Face Assisted AWB - Predictor Probability

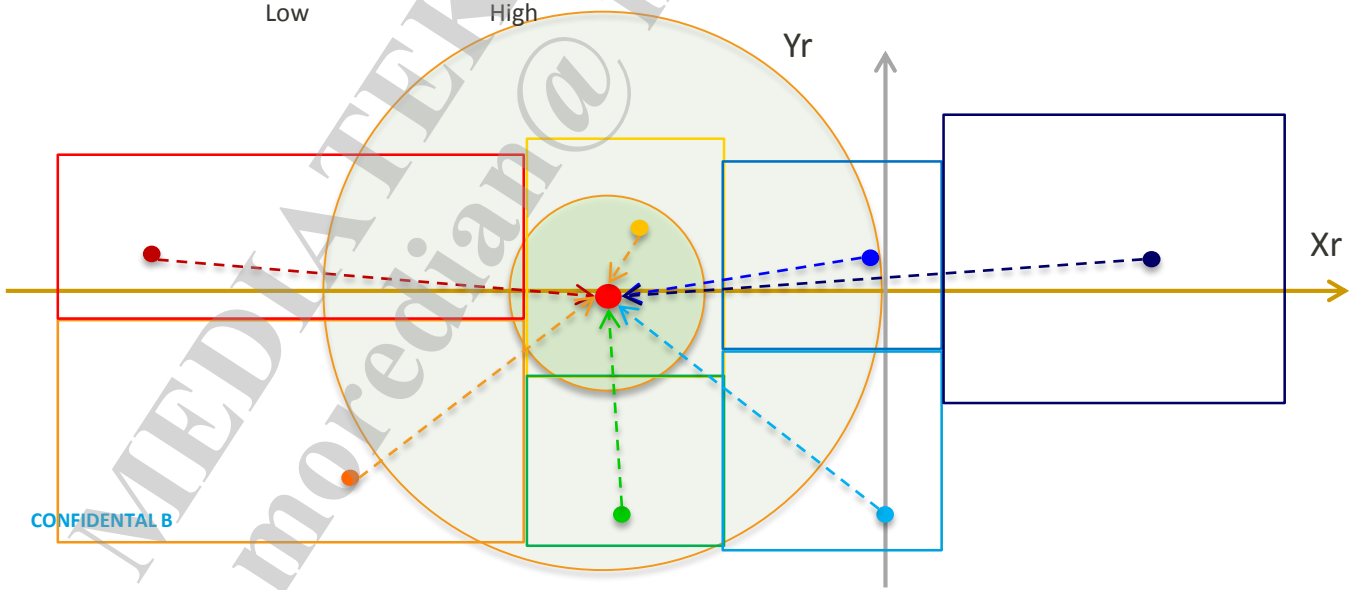
- P3 probability of each light source is calculated by
  - Strength of each light source
  - Distance between light source XY and face assisted predicted-neutral-XY

```
// rCompSetting
{
  { 0, 100 }, //rConfThr
  { 50, 200 }, //rDistanceThr
  { 0,  40 },//rCompRatio
},
```



EXIF

AWB_TAG_FACEAST_P3_STB : 100
AWB_TAG_FACEAST_P3_T : 60
AWB_TAG_FACEAST_P3_WF : 60
AWB_TAG_FACEAST_P3_F : 60
AWB_TAG_FACEAST_P3_CWF : 100
AWB_TAG_FACEAST_P3_D : 71
AWB_TAG_FACEAST_P3_S : 99
AWB_TAG_FACEAST_P3_DF : 100



AWB v5.0

Face Assisted AWB - Final AWB Gain

- P[i] = P0[i] x P1[i] x P2[i] and normalized to 100, i is light source index
- $P_{Final}[i] = (P[i] \times P3[i])/100$: Not normalized, base :100
- $P_{FacePredictor} = \sum P[i] - \sum P_{Final}[i]$
- Final Gain $= [(\sum(P_{Final}[i] \times Gain[i] \times PreferGain[i]) + (P_{FacePredictor} \times FacePredictor)] / \sum(P[i]$
- Example :

	STB	T	WF	F	CWF	D	S	DF	∑
P	0	3	1	33	0	62	1	0	100
P3	100	60	60	60	100	71	99	100	-
P_{Final}	0	2	1	20	0	44	1	0	68
$P_{FacePredictor}$	0	1	0	13	0	18	0	0	32

EXIF

AWB_TAG_P_STB : 0
AWB_TAG_P_T : 3
AWB_TAG_P_WF : 1
AWB_TAG_P_F : 33
AWB_TAG_P_CWF : 0
AWB_TAG_P_D : 62
AWB_TAG_P_S : 1
AWB_TAG_P_DF : 0

AWB_TAG_FACEAST_GAINPROB_STB : 0
AWB_TAG_FACEAST_GAINPROB_T : 1
AWB_TAG_FACEAST_GAINPROB_WF : 0
AWB_TAG_FACEAST_GAINPROB_F : 13
AWB_TAG_FACEAST_GAINPROB_CWF : 0
AWB_TAG_FACEAST_GAINPROB_D : 18
AWB_TAG_FACEAST_GAINPROB_S : 0
AWB_TAG_FACEAST_GAINPROB_DF : 0

AWB_TAG_FINAL_GAINPROB_STB : 0
AWB_TAG_FINAL_GAINPROB_T : 2
AWB_TAG_FINAL_GAINPROB_WF : 1
AWB_TAG_FINAL_GAINPROB_F : 20
AWB_TAG_FINAL_GAINPROB_CWF : 0
AWB_TAG_FINAL_GAINPROB_D : 44
AWB_TAG_FINAL_GAINPROB_S : 1
AWB_TAG_FINAL_GAINPROB_DF : 0

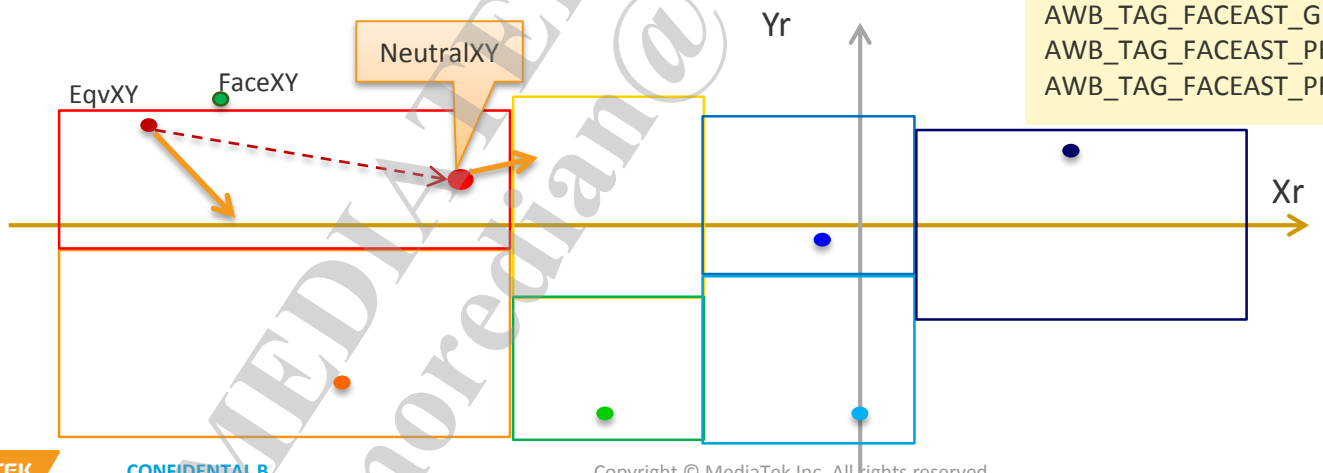
AWB v5.0

Face Assisted AWB + Preference Color (More Flexible & Precise Gain Control)

- Motivation
  - For T/WF/S light sources, confusing color will indicated the XY location of preference light source
  - But preference color is dominated by this inaccurate DLocusOffset / GMOffset ($PrefGain_{orig}$)
- Improvement
  - If NeutralXY is located in preference light area, using NeutralXY to re-process preference color prediction to get new preference gain ($PrefGain_{new}$)
  - Tuning parameter are same as “Preference Color” of each T/WF/S light source (Link)

EXIF

AWB_TAG_FACEAST_DAY_LOCUS_OFFSET : 6153
AWB_TAG_FACEAST_NEW_OFFSET : 1353
AWB_TAG_FACEAST_OFFSET_RATIO : 80
AWB_TAG_FACEAST_RATIO_OFFSET : 1304
AWB_TAG_FACEAST_LUT_OFFSET : 682
AWB_TAG_FACEAST_IS_ABOVE_DAY_LOCUS : 0
AWB_TAG_FACEAST_GM_OFFSET : 444
AWB_TAG_FACEAST_GM_OFFSET_THR : 1500
AWB_TAG_FACEAST_WEIGHT : 0
AWB_TAG_FACEAST_DL_OFFSET_GAIN_R : 545
AWB_TAG_FACEAST_DL_OFFSET_GAIN_B : 476
AWB_TAG_FACEAST_GM_OFFSET_GAIN_R : 512
AWB_TAG_FACEAST_GM_OFFSET_GAIN_B : 512
AWB_TAG_FACEAST_DL_OFFSET_PREF_GAIN_R : 512
AWB_TAG_FACEAST_DL_OFFSET_PREF_GAIN_B : 512
AWB_TAG_FACEAST_GM_OFFSET_PREF_GAIN_R : 512
AWB_TAG_FACEAST_GM_OFFSET_PREF_GAIN_B : 512
AWB_TAG_FACEAST_PREFER_GAIN_R : 545
AWB_TAG_FACEAST_PREFER_GAIN_B : 476



AWB v5.0

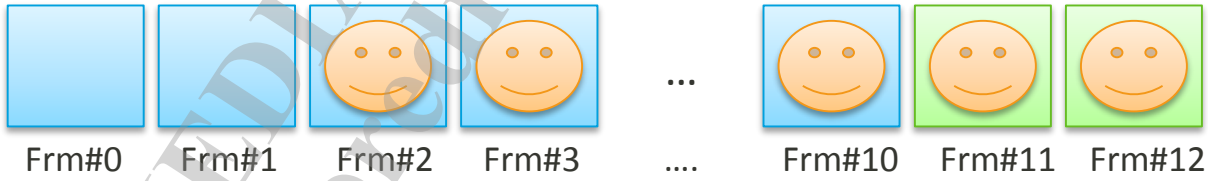
Face Assisted AWB – Stability

- Temporal smooth (*2)
  - Leverage Face Comp. AWB by convergence speed control
  - Applied on face-predicted gain before blending
  - For more details please reference next page
- Face gone process (*2)
  - Similar to face-comp. AWB face-gone mechanism
    - Scene change detection but only estimation by LV change
    - Temporal probability reduce by time
    - For more details
- Stability control (*2)
  - face skin color of current frame will apply temporal smooth by indicated weighting with previous one
  - When start-up camera, delay some frame for stability purpose to avoid face-detection unstable.

```
// rTempoSmooth (*1)
{
  10,   //i4Speed: Converge Speed 1~100
   2,   //i4MinStep: Minimum converge step
   1    //i4ProbReduceStep
},
```

```
// rSceneChange (*1)
{
    30,   //i4LVChangeTh
 20000    //i4AWBGainChangeTh
},
```

```
// rStableSetting
{
   7,  // i4TempoWeight
   8,  // i4DelayFrm
},
```



*1 – These parameters are common setting for Face-Assisted & Face-Comp. AWB

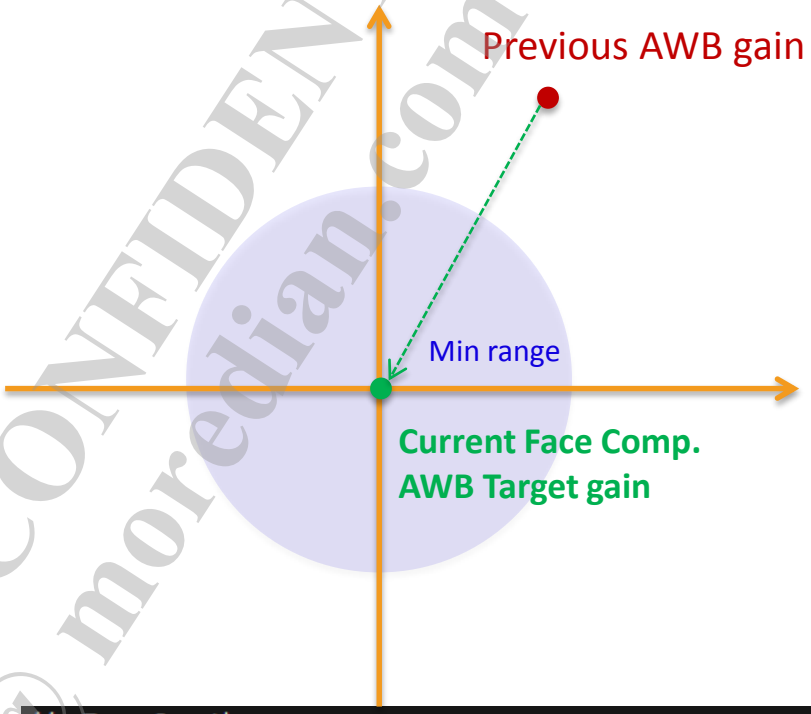
*2 – These parameters can not be simulated by ImagiqSimulator

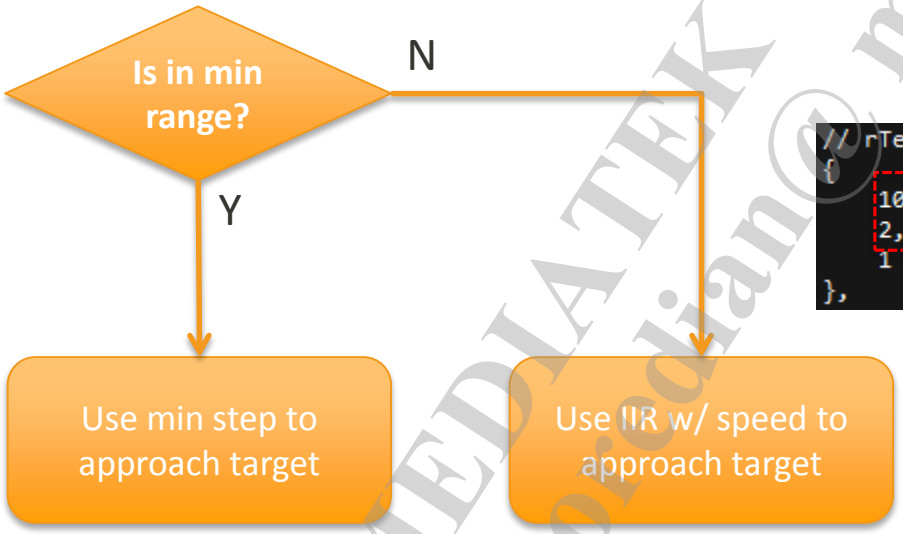
Temporal Face AWB Gain Mechanism

IIR approaching target：

1. Converge speed = 1~100
2. Minimum step

Minimum range = MinStep*100/speed





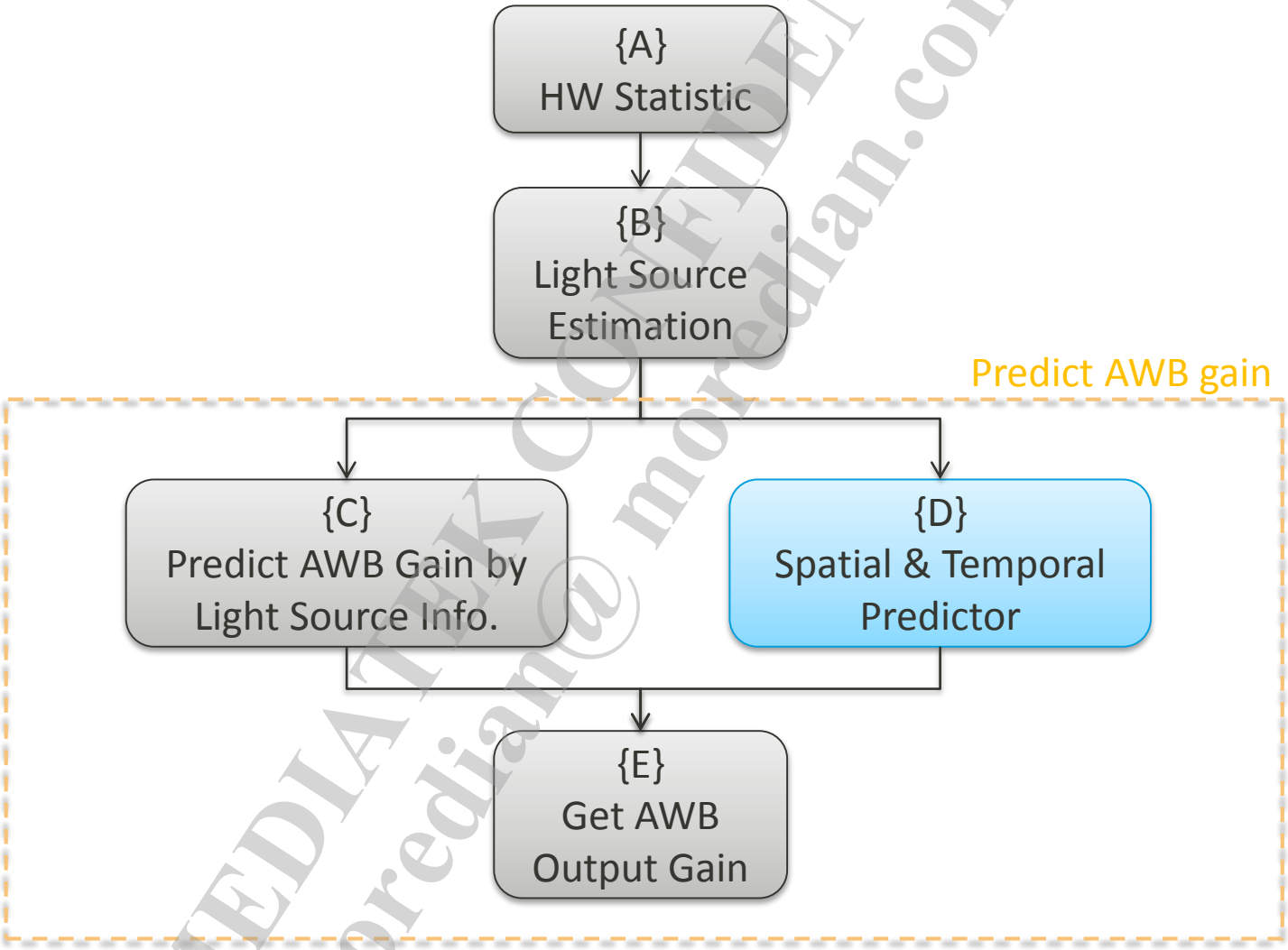

```
// rTempoSmooth
{
    10,    //i4Speed: Converge Speed 1~100
    2,    //i4MinStep: Minimum converge step
    1     //i4ProbReduceStep: If face gone, the face prob reduce step 1~100
},
```

Same with normal AWB temporal smooth mechanism

Back (Face-Assist)

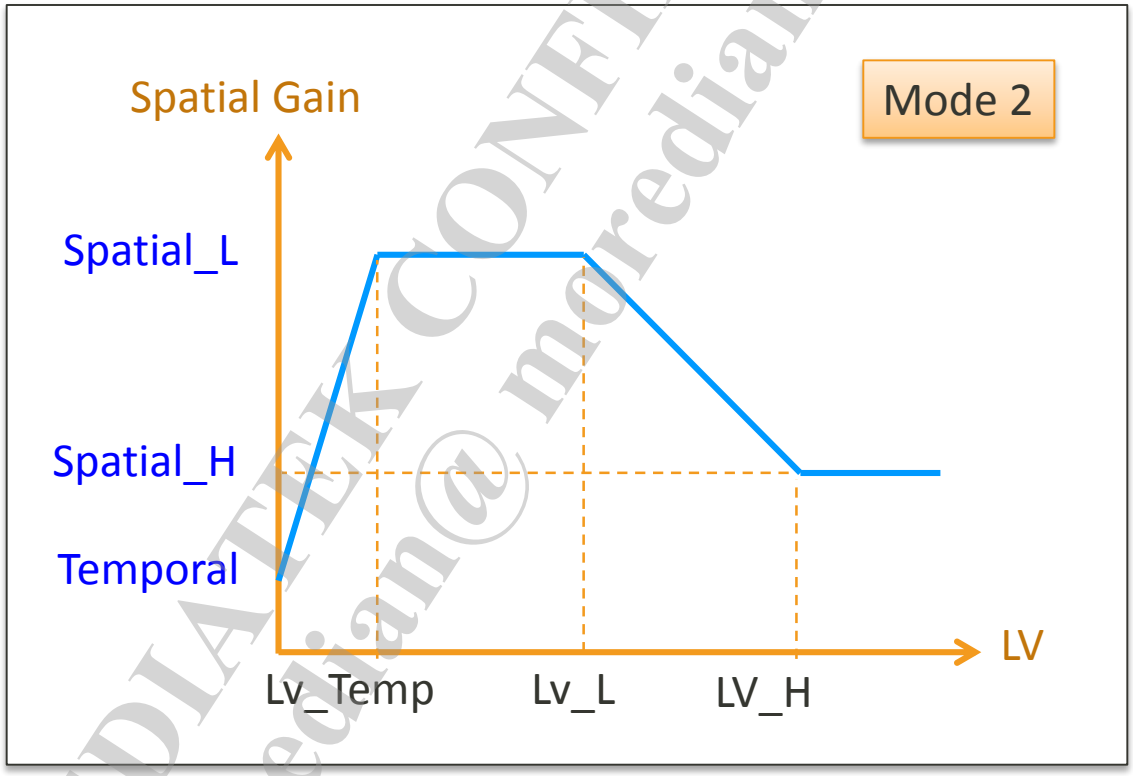
{D} Spatial & Temporal Predictor

AWB Flow



Spatial Predictor

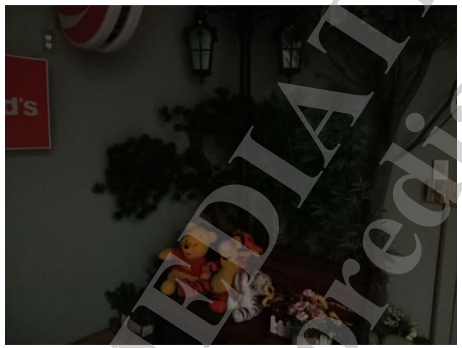
- Spatial Gain for non-reliable mode



Temporal Predictor (1/2)

- Temporal queue will keep enqueue the frame with enough neutral block number while normal AWB processing.
- **Enqueue Policy**
  - Total Neutral Blocks > i4Neutral_ParentBlk_Thr
  - Neutral Blocks in CWF < i4CWFDF_LUTThr * 24 * 18 /100
  - Neutral Blocks in DF < i4CWFDF_LUTThr * 24 * 18 /100

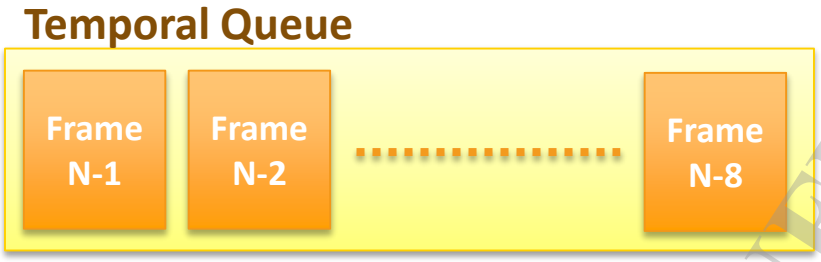
```
// AWB number threshold for temporal predictor
{
  65,  // i4Neutral_ParentBlk_Thr
  //LV 0    1    2    3    4    5    6    7    8    9    10   11   12  13  14  15  16  17  18
  {  100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  50,  25,   2,   2,  2,   2,   2,   2,  2} // i4CWFDF_LUTThr
},
```



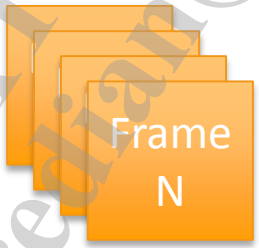
AWB_TAG_ALGO_SCENE_LV	-14
AWB_TAG_NEUTRAL_PB_NUM_STB	0
AWB_TAG_NEUTRAL_PB_NUM_T	2
AWB_TAG_NEUTRAL_PB_NUM_WF	43
AWB_TAG_NEUTRAL_PB_NUM_F	92
AWB_TAG_NEUTRAL_PB_NUM_CWF	10
AWB_TAG_NEUTRAL_PB_NUM_D	1
AWB_TAG_NEUTRAL_PB_NUM_S	0
AWB_TAG_NEUTRAL_PB_NUM_DF	0
AWB_TAG_NEUTRAL_PB_NUM	148

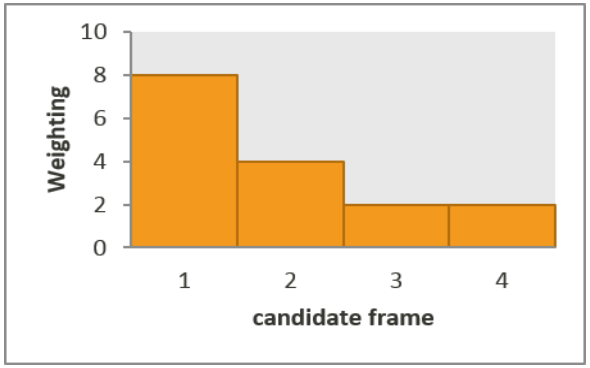
Fit Enqueue Policy

Temporal Predictor (2/2)



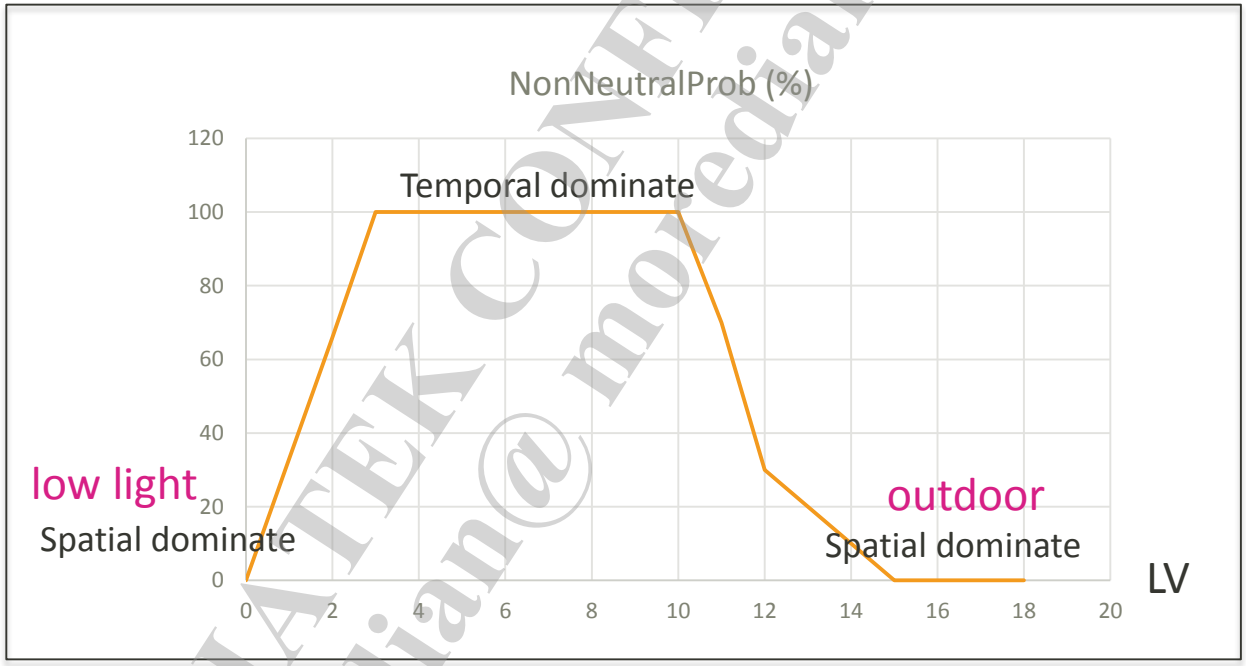
1. Find 4 candidates witch LV is closed to current frame.
2. Sort 4 candidates by using total neutral PB number.
3. Give corresponding weighting. (XrYr)





Mix Spatial & Temporal Predictor

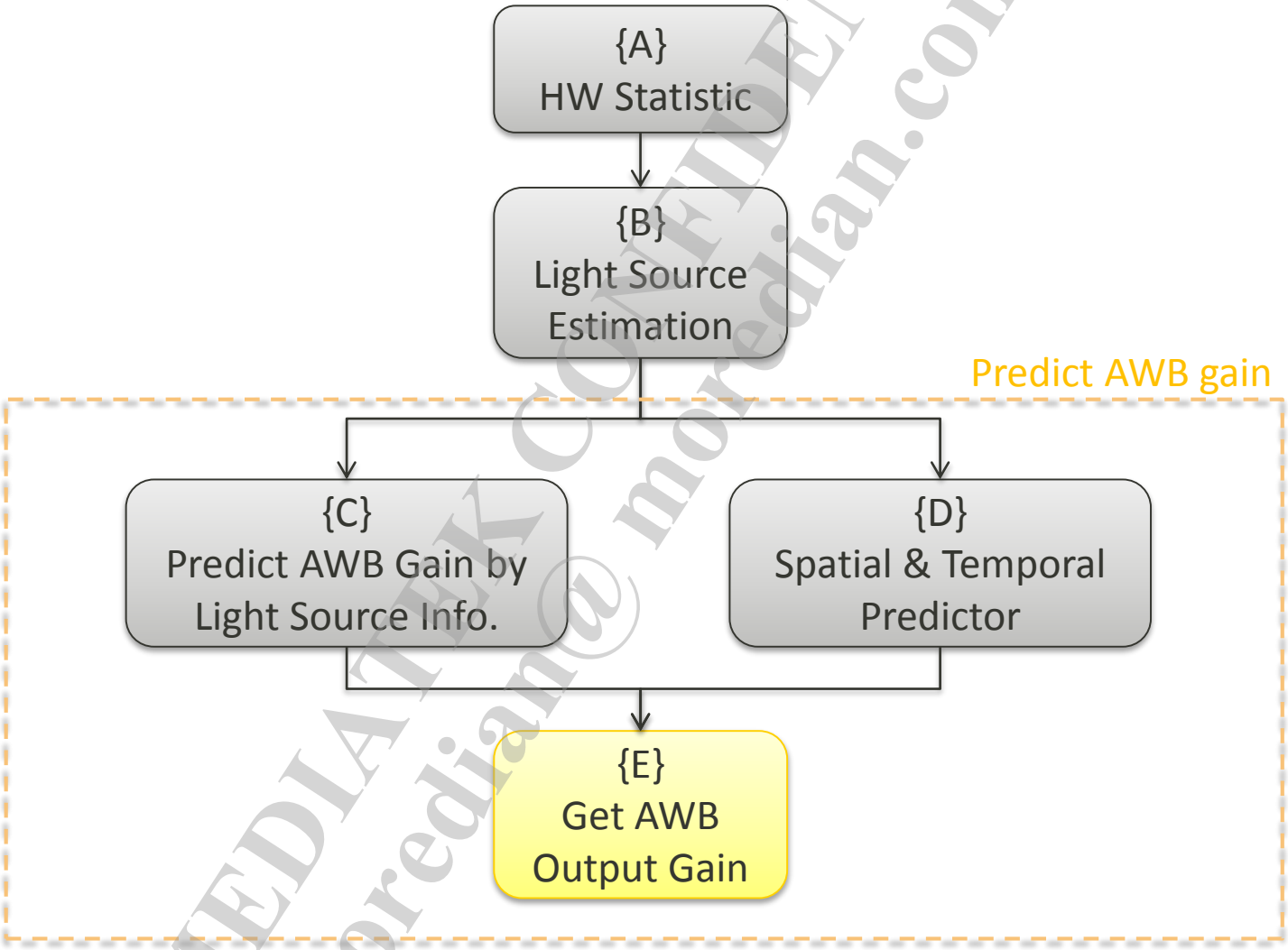
Spatial Predictor + Temporal Predictor




```
// AWB non-neutral probability for spatial and temporal weighting look-up table (Max: 100; Min: 0)
{
   //LV0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18
     { 0,  33,  66, 100, 100, 100, 100, 100, 100, 100, 70,  30,  20,  10,  0,  0,  0,  0}
},
```

{E} Get AWB Output Gain

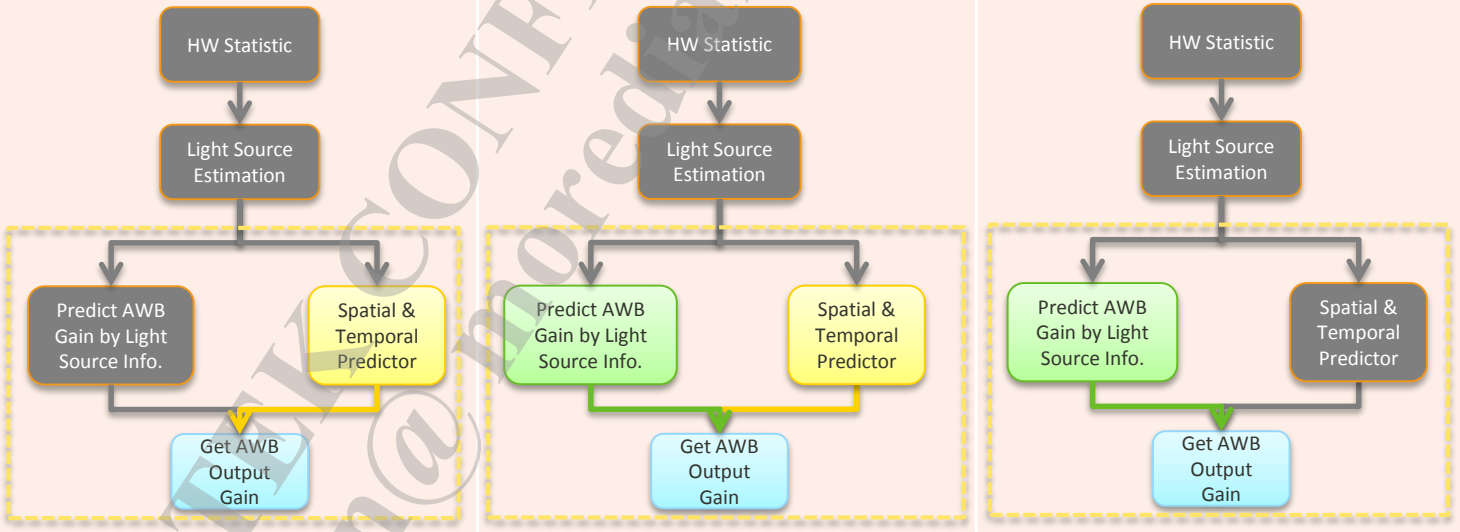
AWB Flow

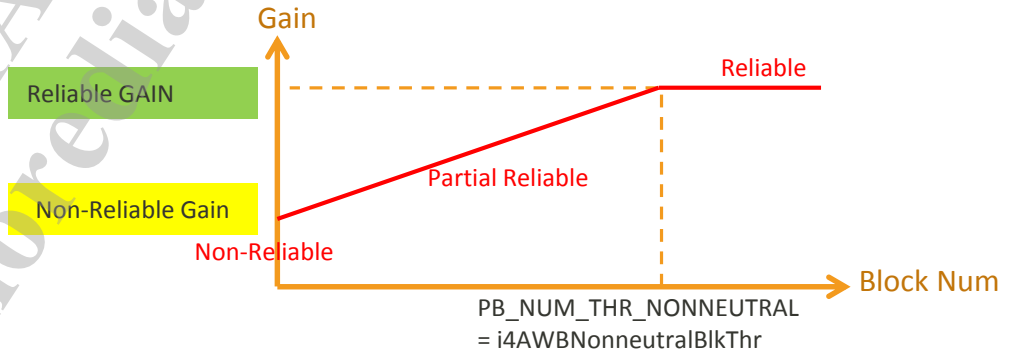


Reliable mode

- Reliable/ Partial Reliable
- Extra Color
- Face Comp.
- Speed Control

AWB_NO_ANY_RELIABLE_MODE	AWB_PARTIAL_RELIABLE_MODE	AWB_FULL_RELIABLE_MODE
NPB= 0	NPB < i4AWBNonneutralBlkThr	NPB >= i4AWBNonneutralBlkThr





AWB
v5.0

Extra Color Compensation (1/5)

Reliable/
Partial Reliable

Extra Color

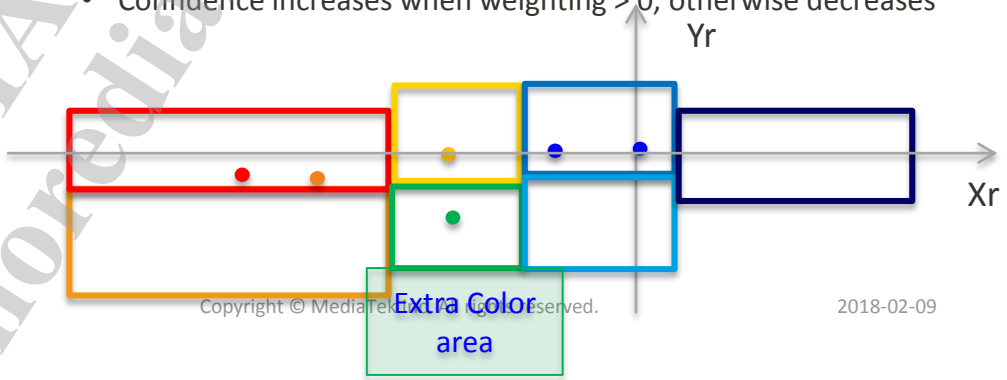
Face Comp.

Speed Control

Purpose :

Detect special color region and blend with specified gain. Usually used to green grass and other daylight locus confusing color.

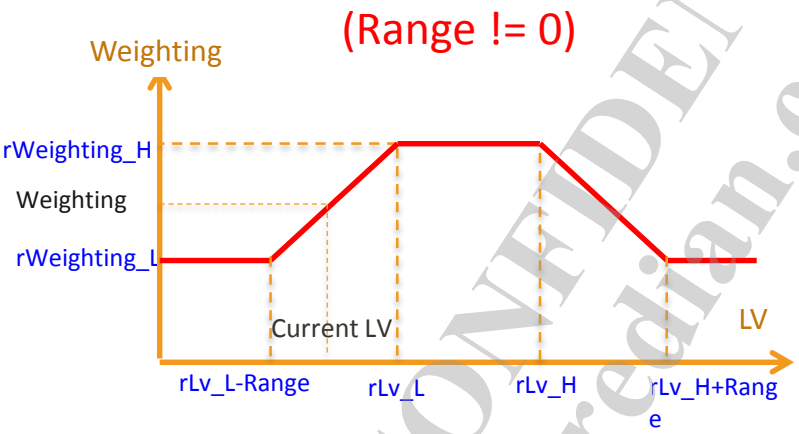
- 3 effects (modes) with different weighting at the same time
  - 1 : blending predefined gain **(Effect is strong)**
  - 2 : reduce P2, must select light sources **(Must used in multi light source, no effect when only single light source)**
  - 3 : reduce statistic weighting and increase spatial gain by reducing daylight locus probability, must select light sources **(Spatial gain is related to LV)**
- Support 8 sets (suggest used on confusing color only)
- More flexible LV setting
- G detection range (0~255)
- Temporal smooth
  - Confidence increases when weighting > 0, otherwise decreases

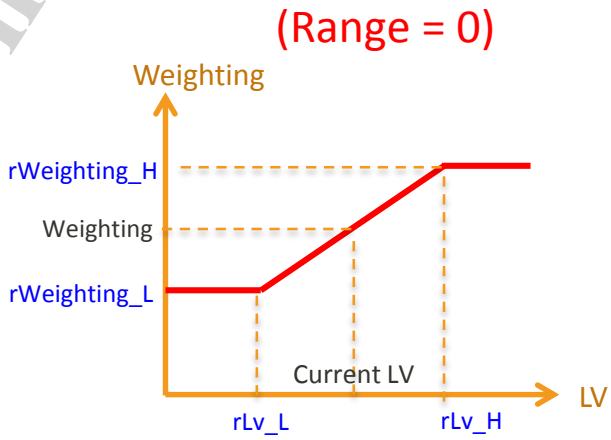


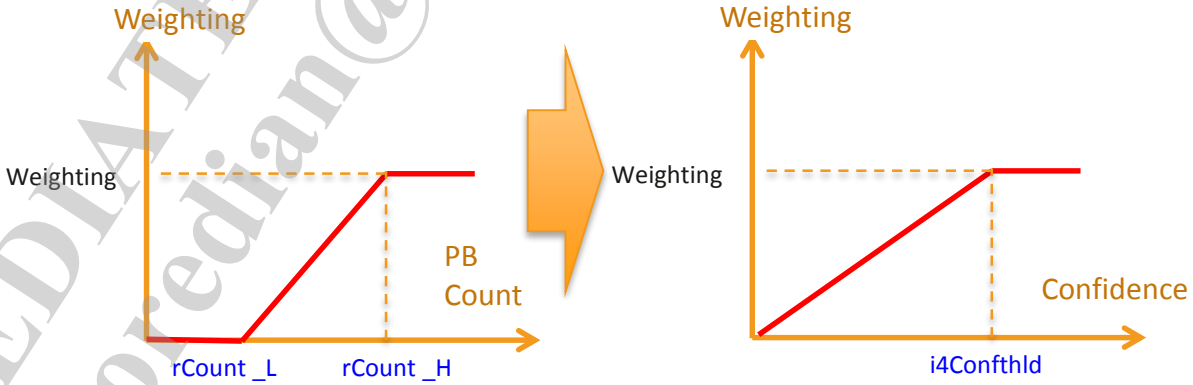
AWB
v5.0

Extra Color Compensation (2/5)

- Reliable/ Partial Reliable
- Extra Color
- Face Comp.
- Speed Control



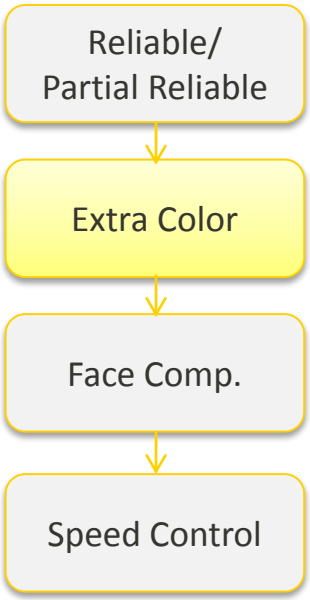




ExtraColor 數量比例越高,強度越強

AWB v5.0

Extra Color Compensation (3/5)




```
// Extra color Detection
{ //set 1
  1, // i4Enable
  {256, 0, 0}, //i4ModeWeight
  16, // i4ConfThld
  // Select light source
  {0, 1, 0, 0, 0, 0, 0,0},
  0, //LV Range
  // Extra Color AWB gain
  {
    1082, // GainR
    512, // GainG
    666, // GainB
  },
  // Extra Color area
  {
     71, // i4RightBound
    -223, // i4LeftBound
    -536, // i4UpperBound
    -636, // i4LowerBound
  },
  { 40,  70}, // Green Level
  { 30, 100}, // rLv
  { 30, 200}, // rCount
  { 10,  20}, // rWeighting
},
{ //set 2
.....
```

Extra window number# enable

Mode 1,2,3 weighting , 0~256 (Unit: 256)

Temporal smooth frame threshold. When extra window occurs continuously more than the threshold, it applies full compensation, else apply a ratio (= continuously frame no. / i4ConfThld)

Compensated light source for mode 2 & 3

Determine weighting curve, if it is 0, weighting keeps at rWeighting_H when LVis higher than rLv_H, else reduce to rWeighting_L from r_Lv_H to rLv_H + LV Range

Predefined gain used in mode 1

Extra window boundary in Xr Yr domain, inside the range will be check other conditions for extra color confirmation

Extra color G value condition, inside this range will be counted

rLv_L and rLv_H to describe the weighting curve

rCount_L and rCount_H to describe weighting curve

rWeighting_L and rWeighting_H to describe weighting curve

AWB v5.0

Extra Color Compensation (4/5)

- Reliable/ Partial Reliable
- Extra Color
- Face Comp.
- Speed Control

AWB v4.0	AWB v5.0
	

AWB v5.0

Extra Color Compensation (5/5)

Example

EXIF (Disable Extra Color)

```
AWB_TAG_ALGO_SCENE_LV : 49
AWB_TAG_P2_T : 100

AWB_TAG_DAYLIGHT_PROB_T : 94
AWB_TAG_EQV_DAYLIGHT_PROB_T : 94
AWB_TAG_EQV_GAIN_R_T : 240
AWB_TAG_EQV_GAIN_G_T : 512
AWB_TAG_EQV_GAIN_B_T : 987

AWB_TAG_EXTRACOLOR_5_INFO_COUNT : 174
AWB_TAG_EXTRACOLOR_5_INFO_WEI_GAIN : 15
AWB_TAG_EXTRACOLOR_5_INFO_WEI_P2 : 13
AWB_TAG_EXTRACOLOR_5_INFO_WEI_DL_PROB : 15
AWB_TAG_EXTRACOLOR_5_INFO_CONF : 16
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_R : 0
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_G : 0
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_B : 0
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_R : 0
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_G : 0
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_B : 0
```

EXIF (Enable Extra Color)

```
AWB_TAG_ALGO_SCENE_LV : 49
AWB_TAG_P2_T : 87

AWB_TAG_DAYLIGHT_PROB_T : 94
AWB_TAG_EQV_DAYLIGHT_PROB_T : 80
AWB_TAG_EQV_GAIN_R_T : 265
AWB_TAG_EQV_GAIN_G_T : 512
AWB_TAG_EQV_GAIN_B_T : 889

AWB_TAG_EXTRACOLOR_5_INFO_COUNT : 174
AWB_TAG_EXTRACOLOR_5_INFO_WEI_GAIN : 15
AWB_TAG_EXTRACOLOR_5_INFO_WEI_P2 : 13
AWB_TAG_EXTRACOLOR_5_INFO_WEI_DL_PROB : 15
AWB_TAG_EXTRACOLOR_5_INFO_CONF : 16
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_R : 266
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_G : 512
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_IN_B : 869
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_R : 290
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_G : 512
AWB_TAG_EXTRACOLOR_5_INFO_GAIN_OUT_B : 835
```

Extra Color Tuning Parameter

```
1, // i4Enable
{ 96, 80, 96}, // i4ModeWeight
16, // i4ConfThr
{0, 1, 0, 0, 0, 0, 0, 0}, // i
20, // i4LvRange
// Extra Color AWB gain
{
    1124, // GainR
     512, // GainG
     808, // GainB
},
// Extra Color area
{
       -340, // i4RightBound
       -400, // i4LeftBound
       -380, // i4UpperBound
       -430, // i4LowerBound
},
{  25,  50}, // rGlevel
{  30, 80}, // rLv
{  50, 200}, // rCount
{   0,  50}, // rWeighting
```

Face Comp. AWB

Reliable/
Partial Reliable

↓

Extra Color

↓

Face Comp.

↓

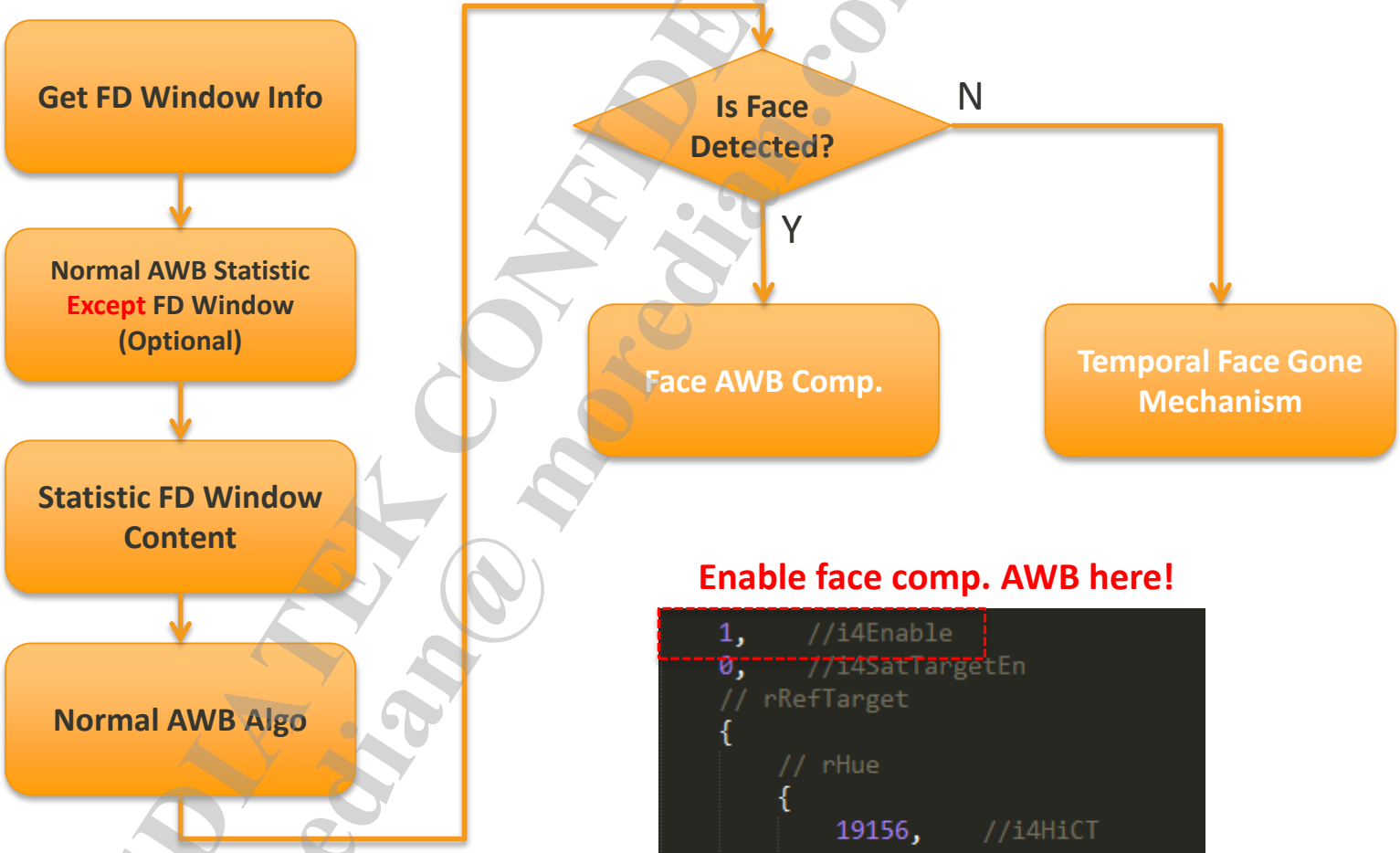
Speed Control

Purpose :

In order to keep the skin color more stable under different illuminant condition. Adjust AWB gain toward to user-defined skin color slightly and usually used in front camera. (AWB assist Face)

w/o face comp. AWB

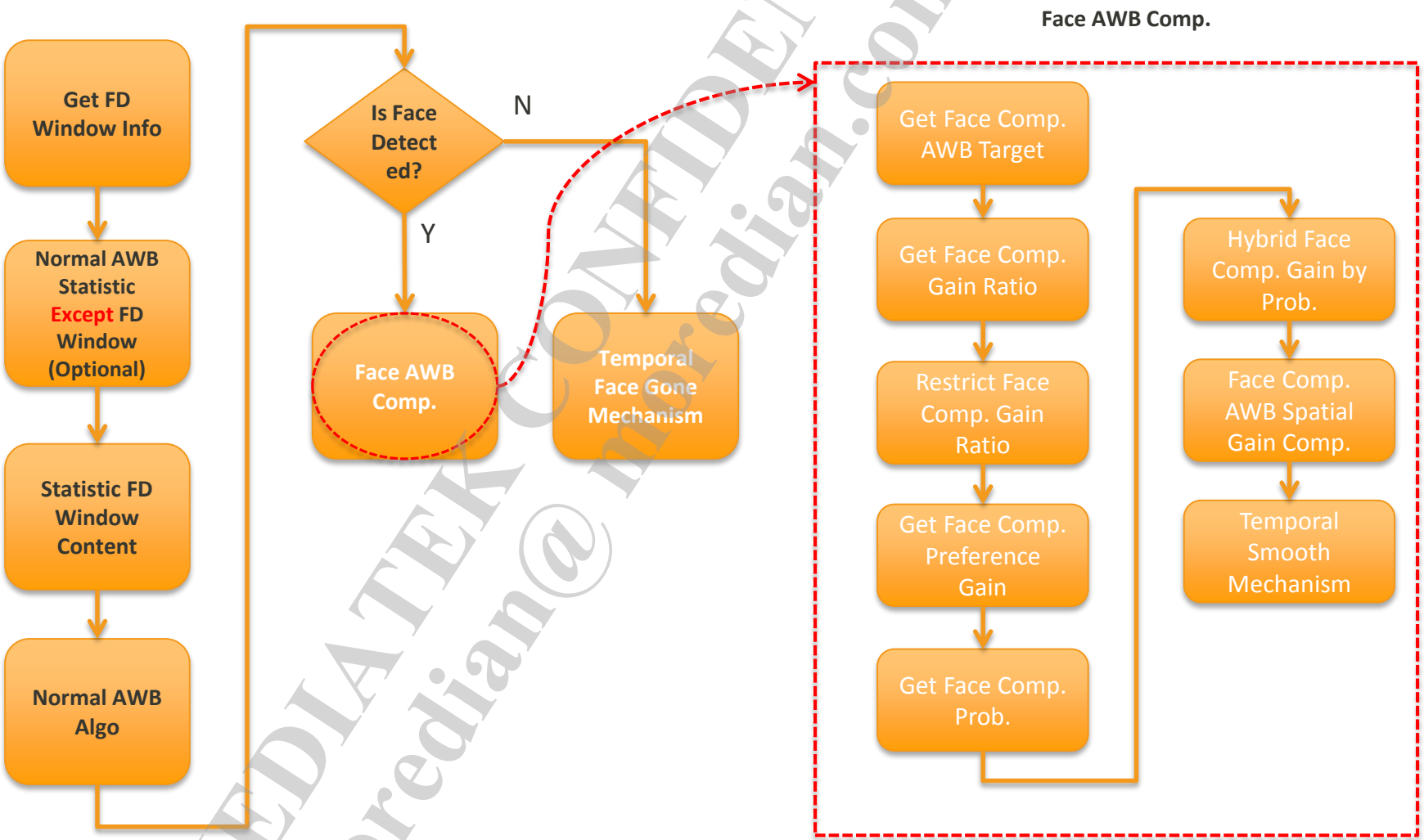
AWB Algo Flow



Enable face comp. AWB here!

```
1,    //i4Enable
0,    //i4SatTargetEn
// rRefTarget
{
    // rHue
    {
        19156,    //i4HiCT
        16684,    //i4MidCT
        23143     //i4LowCT
    },
```

Face Comp. AWB v1.5 Algo Flow



Face Comp. AWB Concept

Proc RAW domain

{R, G, B} = {30, 40, 18}
{R/G, B/G} = {0.75, 0.45}

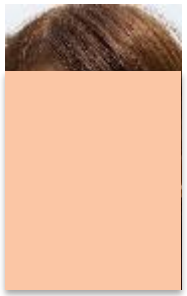
Normal AWB

After Normal AWB

AWB Gain = {1062, 512, 786}
{R/G, B/G} = {1.556, 0.691}

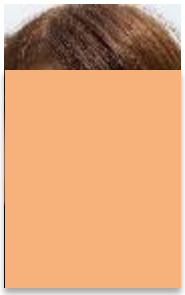
Face Comp. Ratio

Target by Tuning



Face Comp. Ratio = {0.843, 1.236}
{R/G, B/G} = {1.312, 0.854}

Restrict Ratio



Face Comp. Restrict Ratio (5%) = {0.95, 1.05}
{R/G, B/G} = {1.478, 0.726}

Face Comp. Gain

$$FaceAWBGain = NormalAWBGain \times RestrcitRatio$$

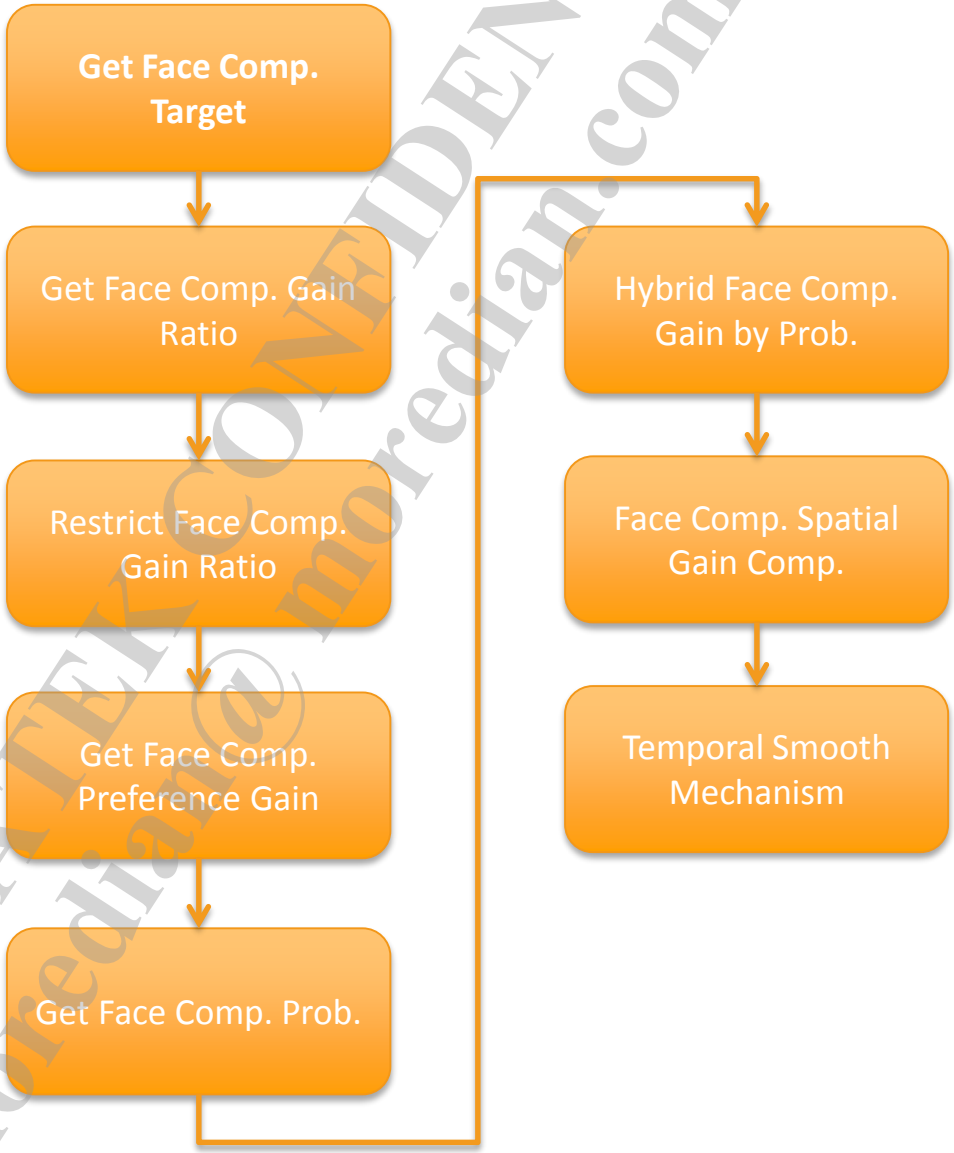
Face Comp. Preference Gain

$$FaceAWBGain = FaceAWBGain \times PreferenceGain$$

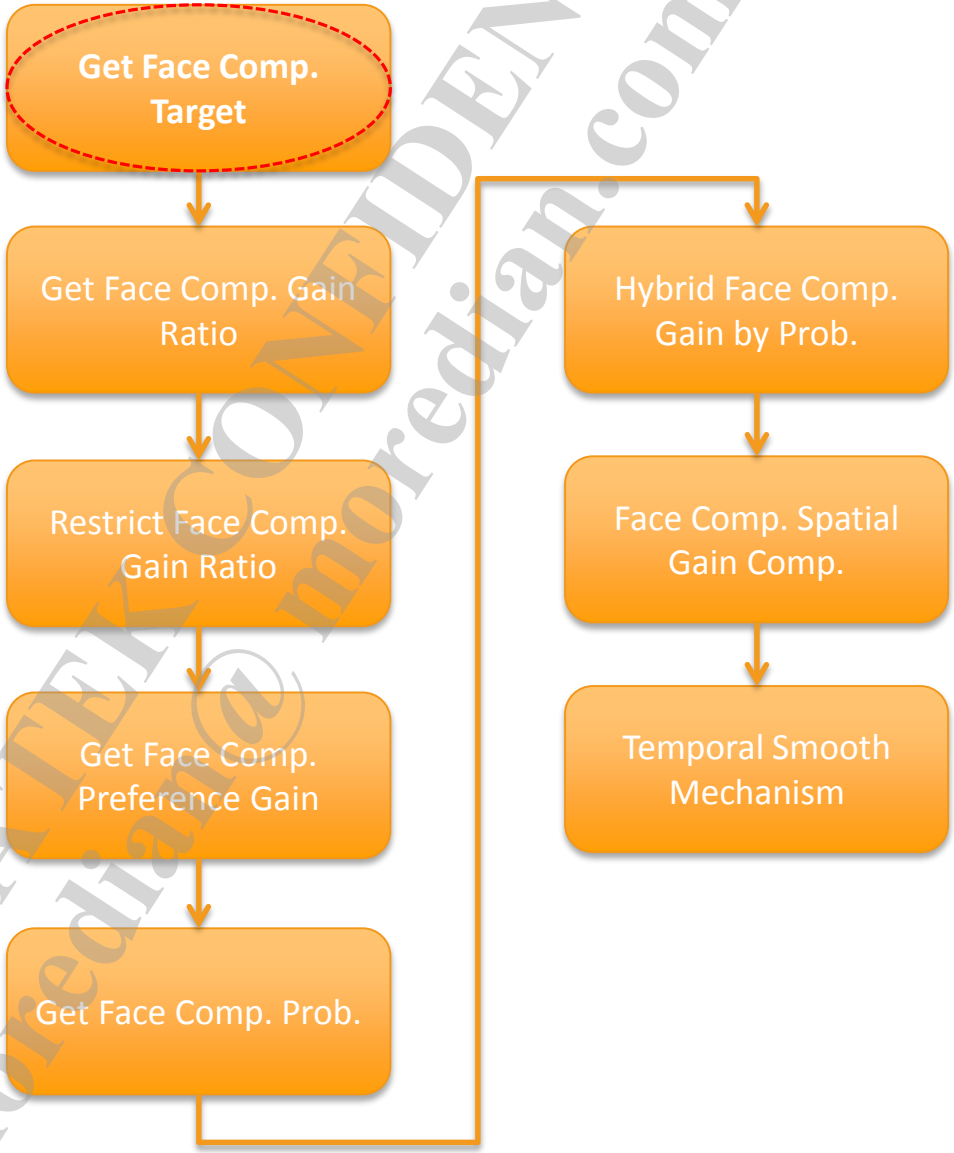
Face Comp. Prob

Hybrid Face Comp. Gain & Normal AWB Gain

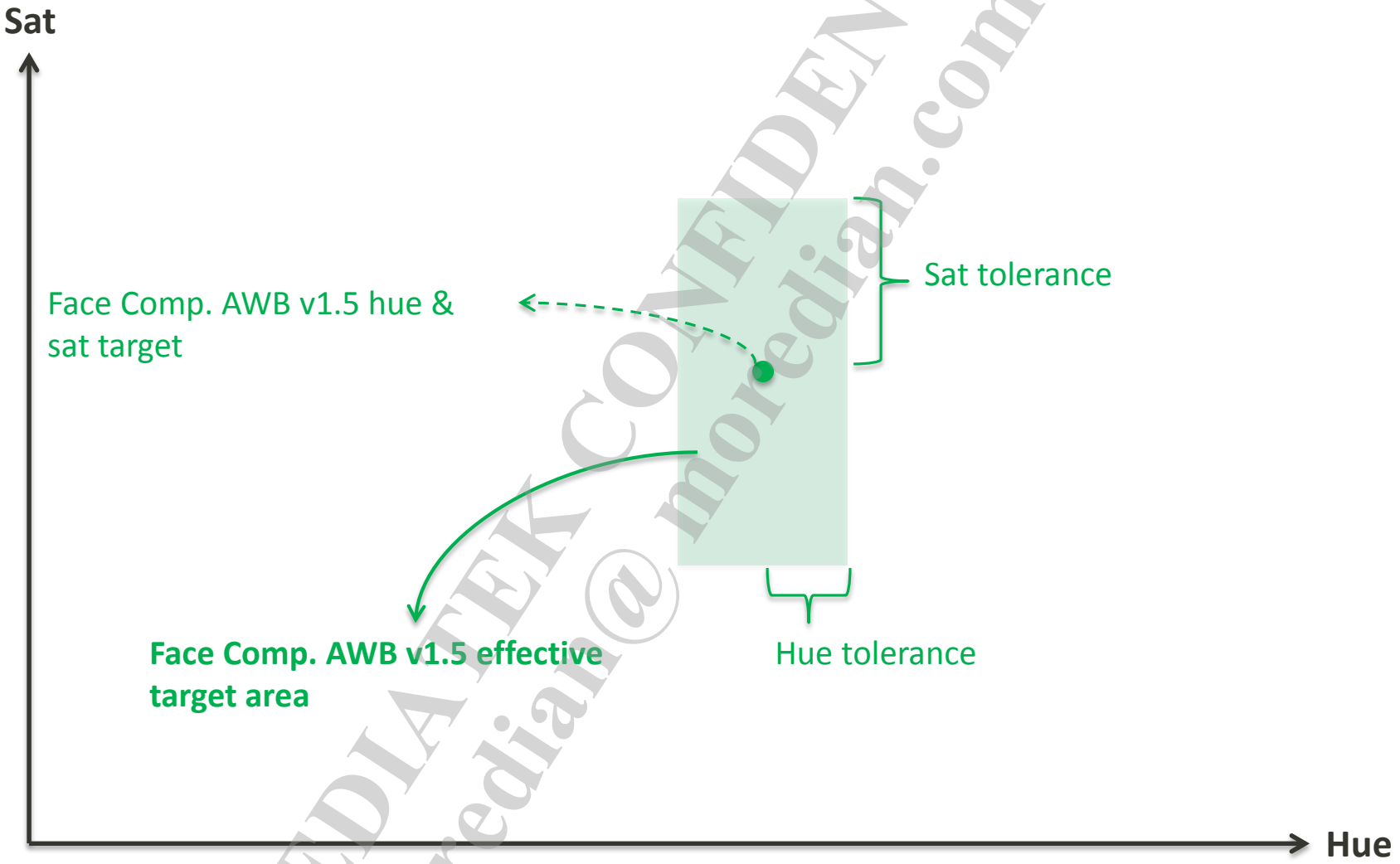
Face Comp. AWB v1.5 Comp.



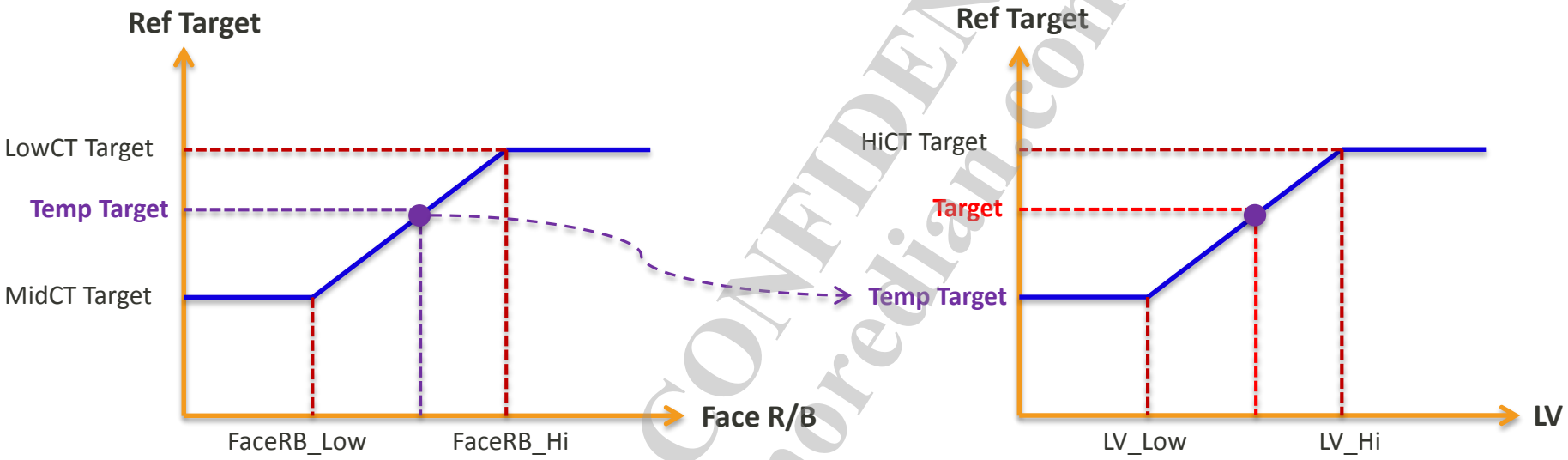
Face Comp. AWB v1.5 Comp.



Face Comp. AWB Target Definition



Dynamic Face Skin Tone Target




```
1,    // i4SatTargetEn
// rRefTarget
{
    //i4Hue
    {
        19000,    // i4HiCT target hue, Unit = 1000
        17000,    // iMidCT target hue, Unit = 1000
        23000     // i4LowCT target hue, Unit = 1000
    },
    //i4Sat
    {
        40000,    // i4Sat target saturation, Unit = 1000
        35000,    // i4Sat target saturation, Unit = 1000
        50000     // i4Sat target saturation, Unit = 1000
    },
    7     // i4TempoWeight for Temporal Target weighting: 0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
},
// rSceneJudge
{
    90,    //i4LVLow for Indoor & Outdoor dynamic
    110,    //i4LVHi for Indoor & Outdoor dynamic
    2300,    //i4FaceRB_Low for Indoor Mid CT & Low CT dynamic, Unit = 1000
    3000    //i4FaceRB_Hi for Indoor Mid CT & Low CT dynamic, Unit = 1000
},
```

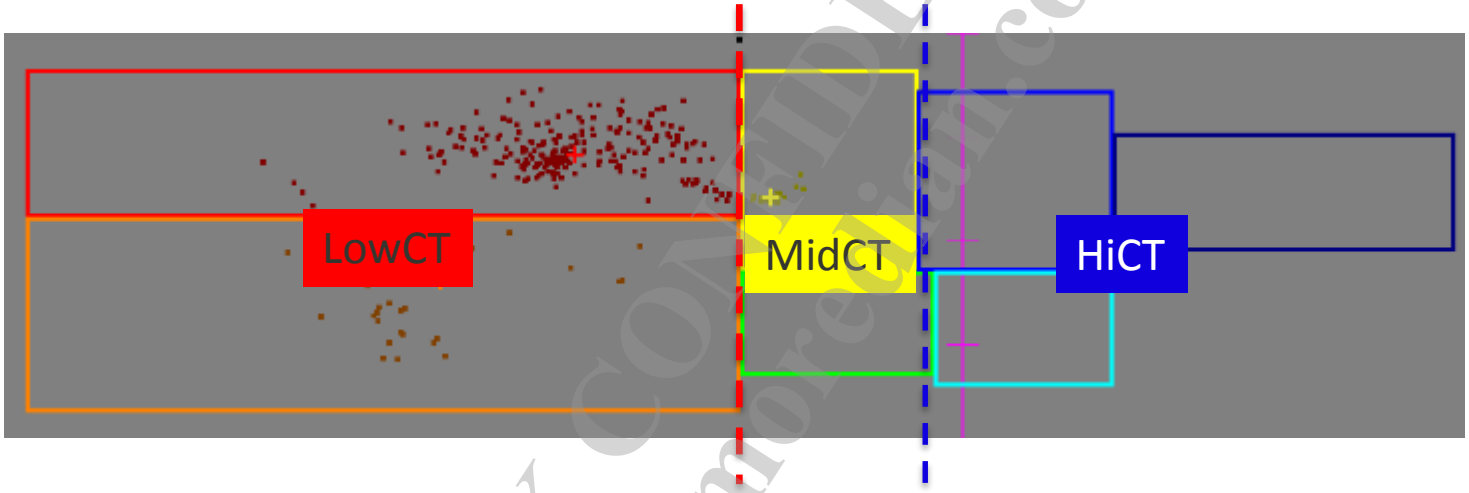
Use saturation target

Target hue

Target saturation

Dynamic switch threshold

Face Comp. AWB v1.5 Target Classify – Normal AWB Info



HiCT_P = Daylight_Prob + DF_Prob + Shade_Prob
MidCT_P = Fluorescent_Prob + CWF_Prob
LowCT_P = Tungsten_Prob + WF_Prob

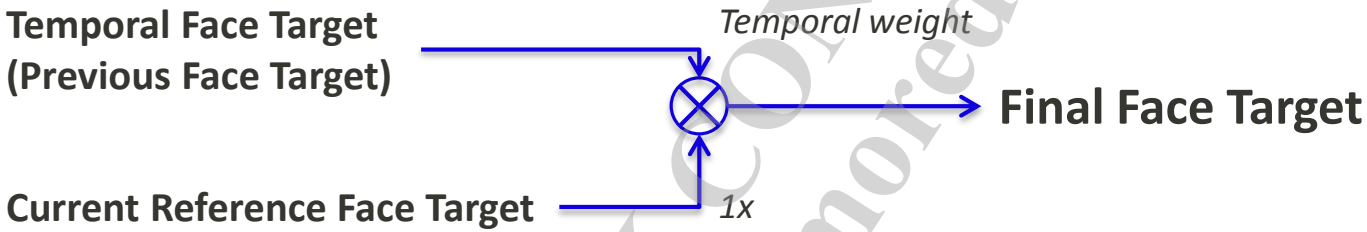
$$TempTarget = \frac{LowTarget \times LowCT\_P + MidTarget \times MidCT\_P}{LowCT\_P + MidCT\_P}$$

$$FaceTarget = \frac{TempTarget \times (100 - HiCT\_P) + HiTarget \times HiCT\_P}{100}$$

When face R/B thr or LV thr is set 0, algo will auto change to use normal AWB info.

Temporal Face Skin Tone Target Mechanism

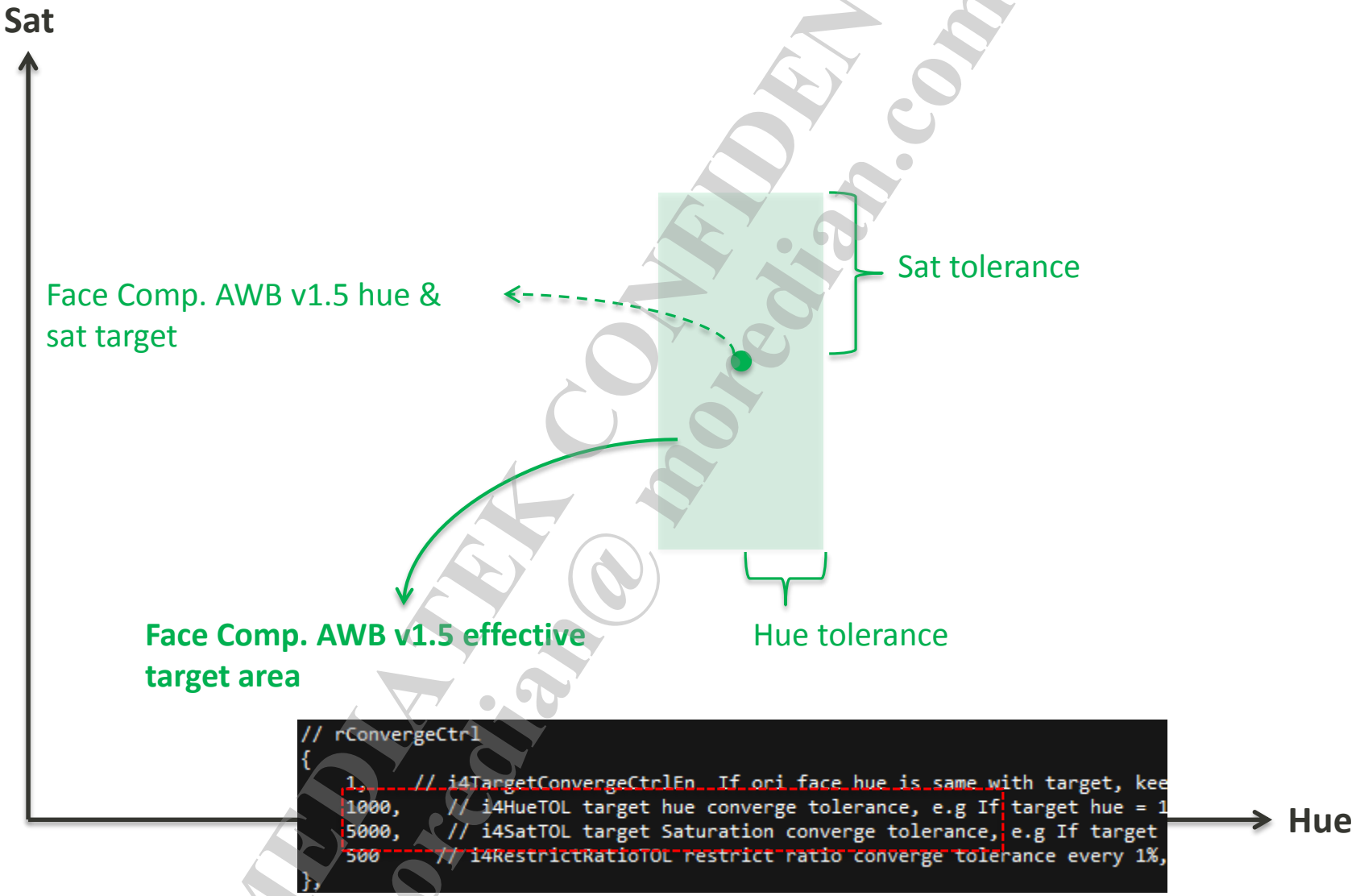
$$FinalFaceTarget = \frac{(TemporalFaceTarget \times Temporal\_Weight) + (1 \times CurrentFaceTarget)}{(Temporal\_Weight + 1)}$$




```
1,    // i4SatTargetEn
// rRefTarget
{
    //i4Hue
    {
        19000,    // i4HiCT target hue, Unit = 1000
        17000,    // iMidCT target hue, Unit = 1000
        23000     // i4LowCT target hue, Unit = 1000
    },
    //i4Sat
    {
        40000,    // i4Sat target saturation, Unit = 1000
        35000,    // i4Sat target saturation, Unit = 1000
        50000     // i4Sat target saturation, Unit = 1000
    },
    7     // i4TempoWeight for Temporal Target weighting: 0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
},
```

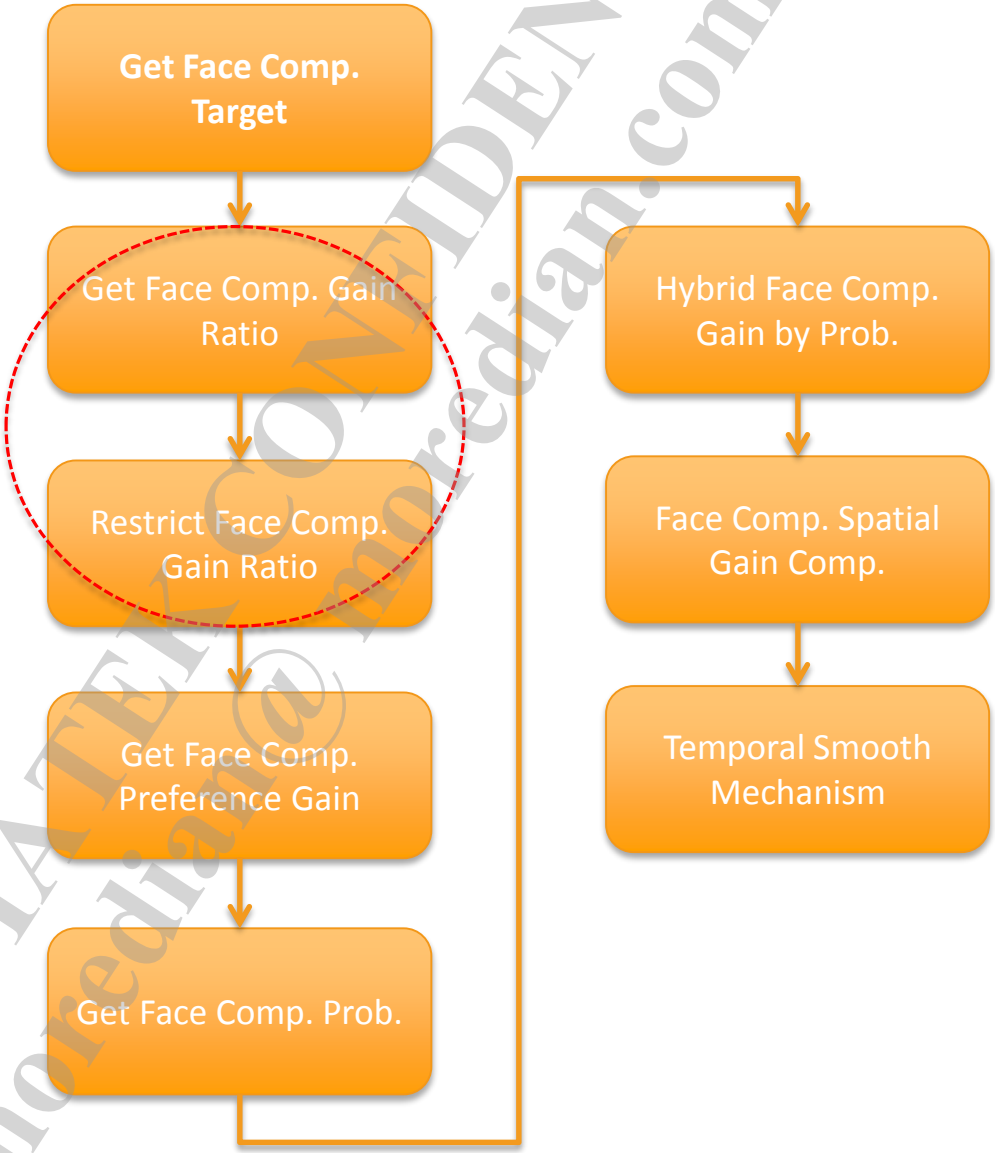
If using simulation, this mechanism won't work!

Face Comp. AWB Target Definition

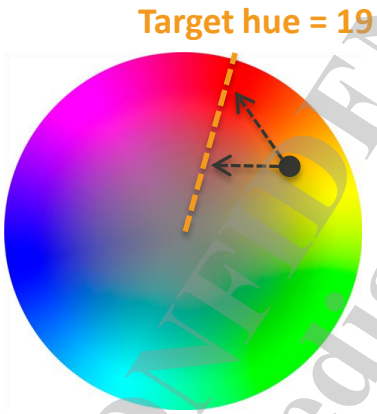



```
// rConvergeCtrl
{
    1,      // i4TargetConvergeCtrlEn  If ori face hue is same with target, kee
    1000,   // i4HueTOL target hue converge tolerance, e.g If target hue = 1
    5000,   // i4SatTOL target Saturation converge tolerance, e.g If target
    500     // i4RestrictRatioTOL restrict ratio converge tolerance every 1%,
},
```

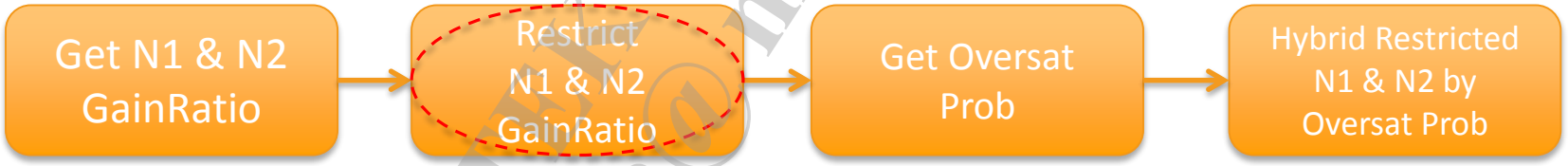
Face Comp . AWB v1.5 Comp.



Get Face Comp. AWB Gain Ratio



Algo will automatically calculate 2 ratio (**yellowish** & **bluish**), no need to tuning.



Face Comp. AWB Gain Ratio Restrict Mechanism

Restrict ratio is a LUT based on LV as tuning parameters

```
// i4GainRatioRestrictLUT
//LV0  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
{  0, 10, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 10,  0,  0},
```

Increase calculation & tuning precision, unit = 0.1%

Face Comp. AWB v1.5 Algorithm Target Converge Control

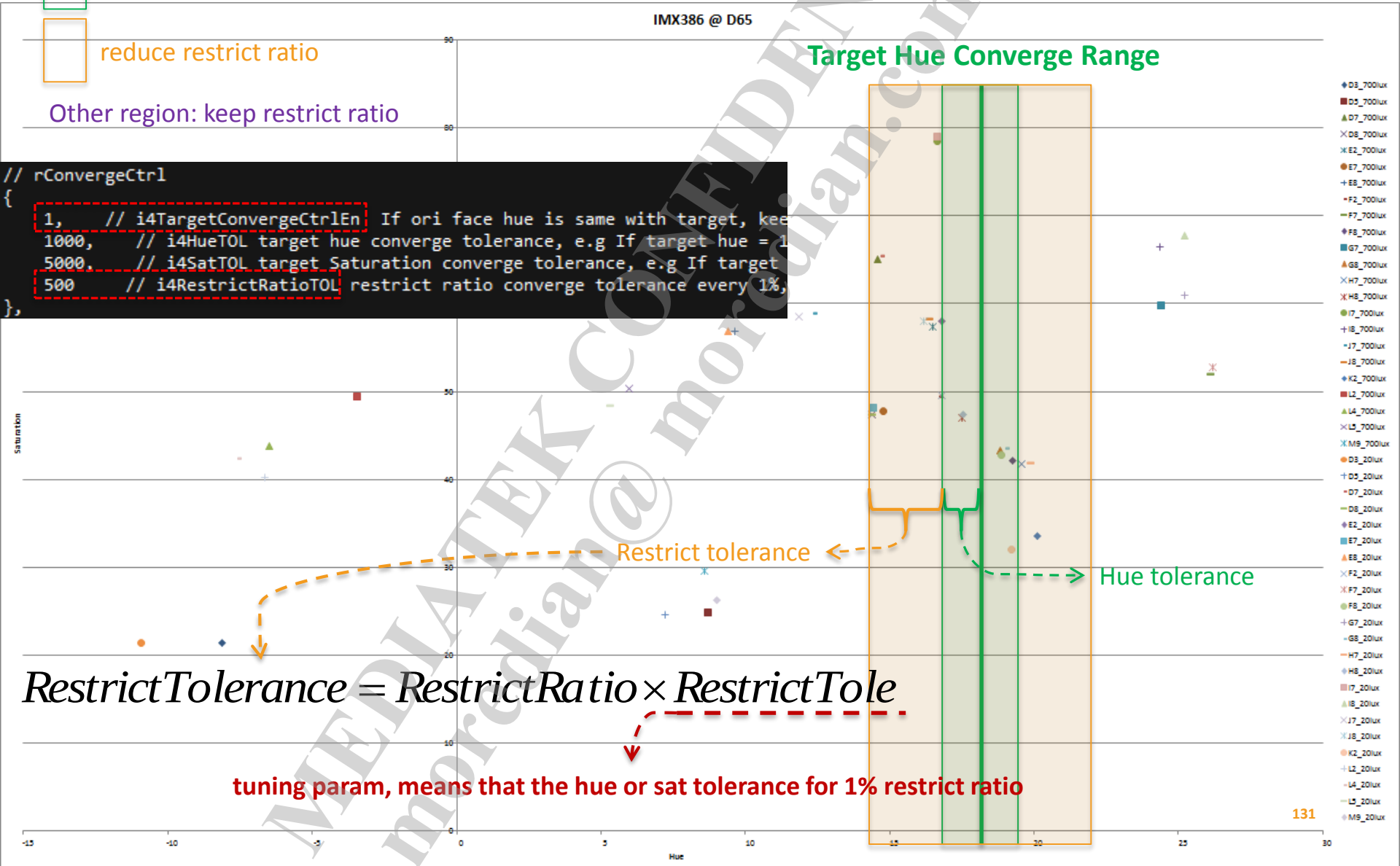
restrict ratio = 0

reduce restrict ratio

Other region: keep restrict ratio

Sat target disable!

```
// rConvergeCtrl
{
    1,    // i4TargetConvergeCtrlEn If ori face hue is same with target, kee
    1000,    // i4HueTOL target hue converge tolerance, e.g If target hue = 1
    5000,    // i4SatTOL target Saturation converge tolerance, e.g If target
    500    // i4RestrictRatioTOL restrict ratio converge tolerance every 1%,
},
```



$$RestrictTolerance = RestrictRatio \times RestrictTole$$

tuning param, means that the hue or sat tolerance for 1% restrict ratio

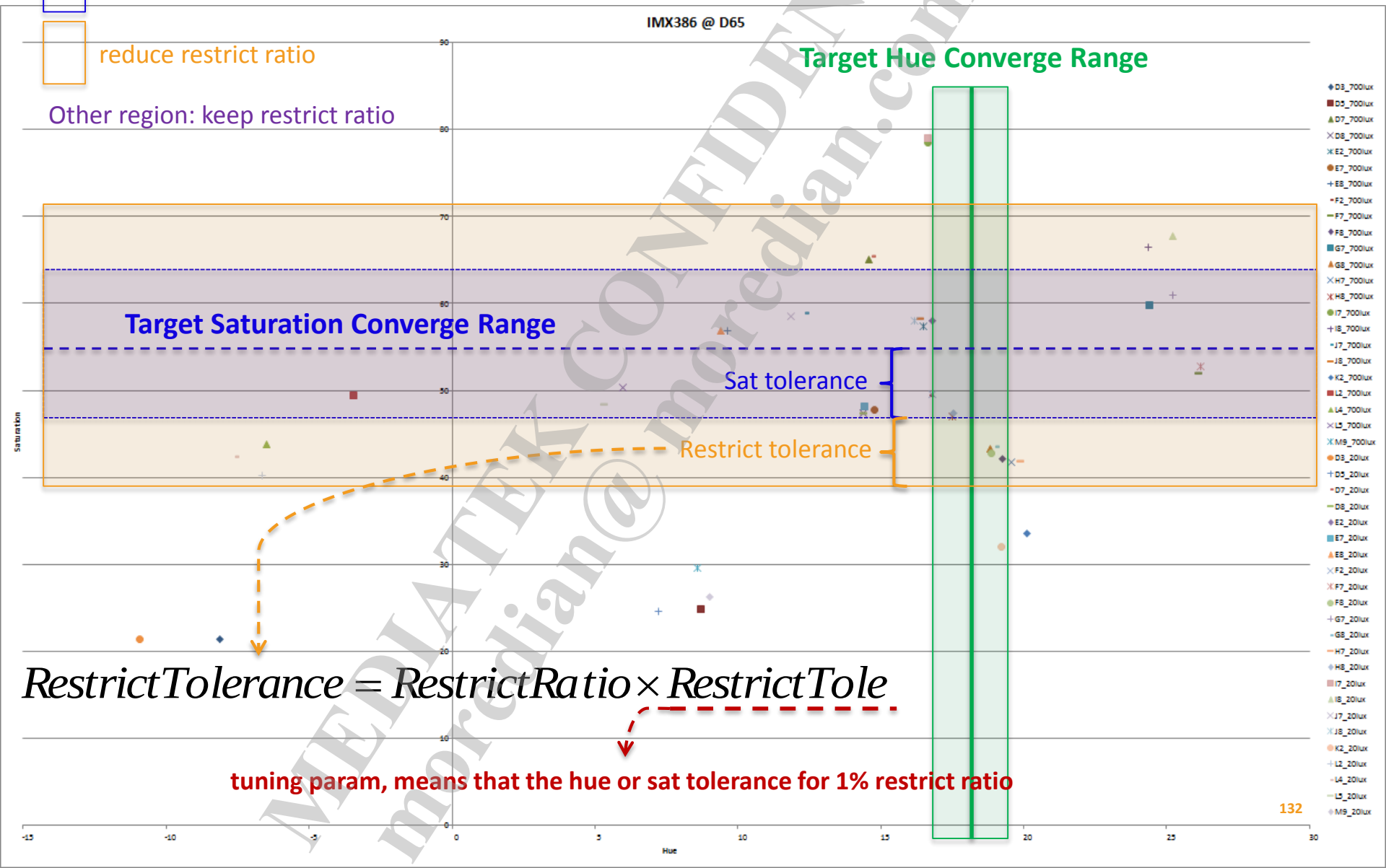
Face Comp. AWB v1.5 Algorithm Target Converge Control

restrict ratio = 0

reduce restrict ratio

Other region: keep restrict ratio

Sat target enable!



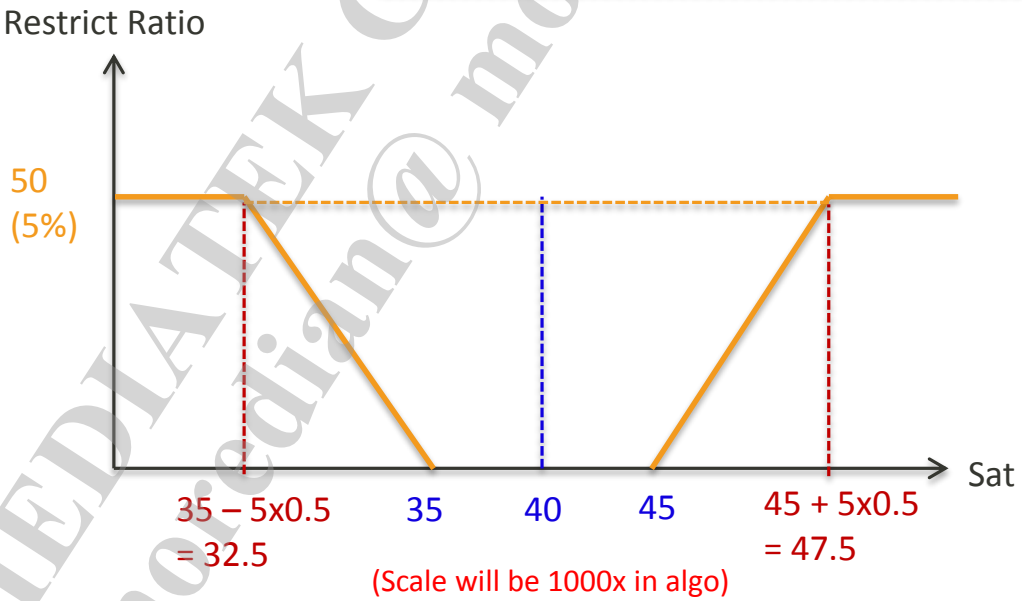
$$RestrictTolerance = RestrictRatio \times RestrictTole$$

tuning param, means that the hue or sat tolerance for 1% restrict ratio

Face Comp. AWB v1.5 Algorithm Target Converge Control Example

e.g.
Target sat = 40
Sat TOL = 5
Restrict Ratio = 50 (means 5%)
Restrict TOL = 500

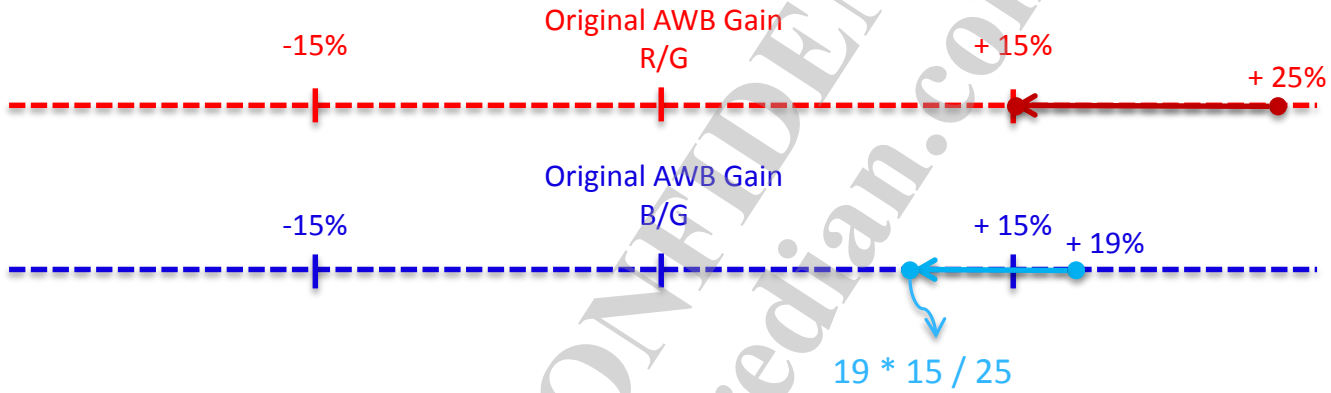
```
//i4Sat
{
    40000,     // i4Sat target saturation, Unit = 1000
    35000,     // i4Sat target saturation, Unit = 1000
    50000      // i4Sat target saturation, Unit = 1000
},
// rConvergeCtrl
{
    1,    // i4TargetConvergeCtrlEn  If ori face hue is same with target, kee
    1000,     // i4HueTOL target hue converge tolerance, e.g If target hue = 1
    5000,     // i4SatTOL target Saturation converge tolerance, e.g If target
    500     // i4RestrictRatioTOL restrict ratio converge tolerance every 1%,
},
// i4GainRatioRestrictLUT
//LV0  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
{  0, 10, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 10,  0,  0},
```



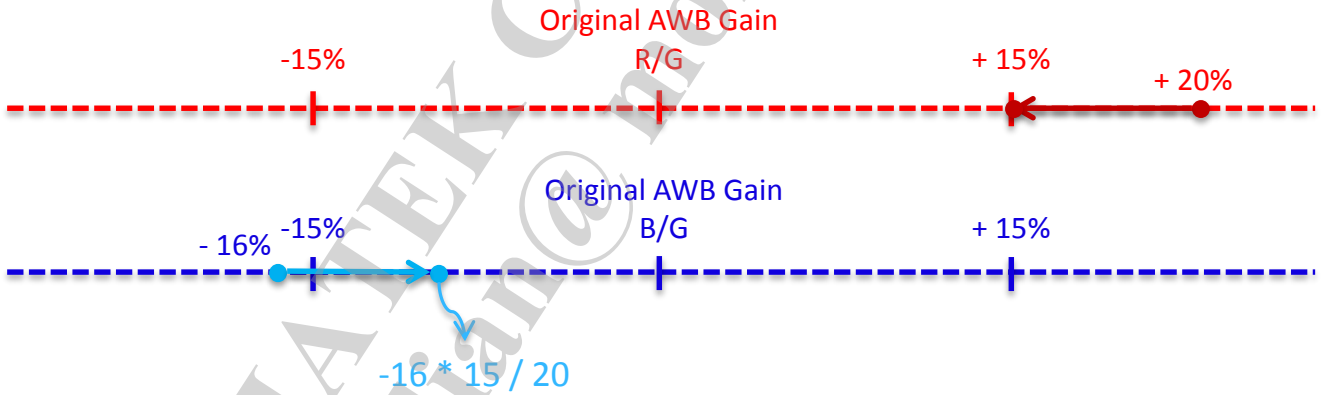
(Scale will be 1000x in algo)

Face Comp. AWB Gain Ratio Restrict Mechanism

Case I:



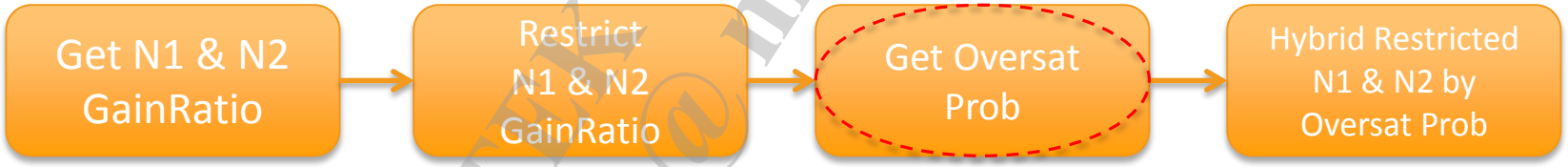
Case II:



Restrict ratio will have a temporal smooth for avoiding AWB jumping or unstable.

Get Face Comp. AWB Gain Ratio

Algo will automatically calculate 2 ratio (**yellowish** & **bluish**), no need to tuning.



Get Face Comp. AWB Gain Ratio

$$\begin{cases} AddSatGainRatio\_R = n_1 \times PreGainRatio\_R \\ AddSatGainRatio\_B = \dfrac{1}{hn_1} \times PreGainRatio\_B \end{cases}$$

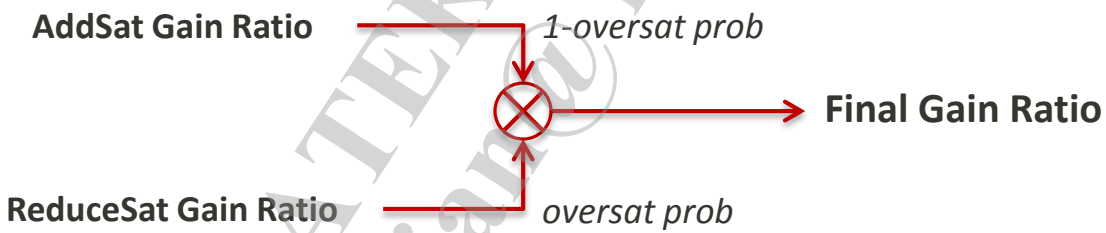
{R+, B-}

How to get oversat prob?

$$\begin{cases} ReduceSatGainRatio\_R = n_2 \times PreGainRatio\_R \\ ReduceSatGainRatio\_B = \dfrac{1}{hn_2} \times PreGainRatio\_B \end{cases}$$

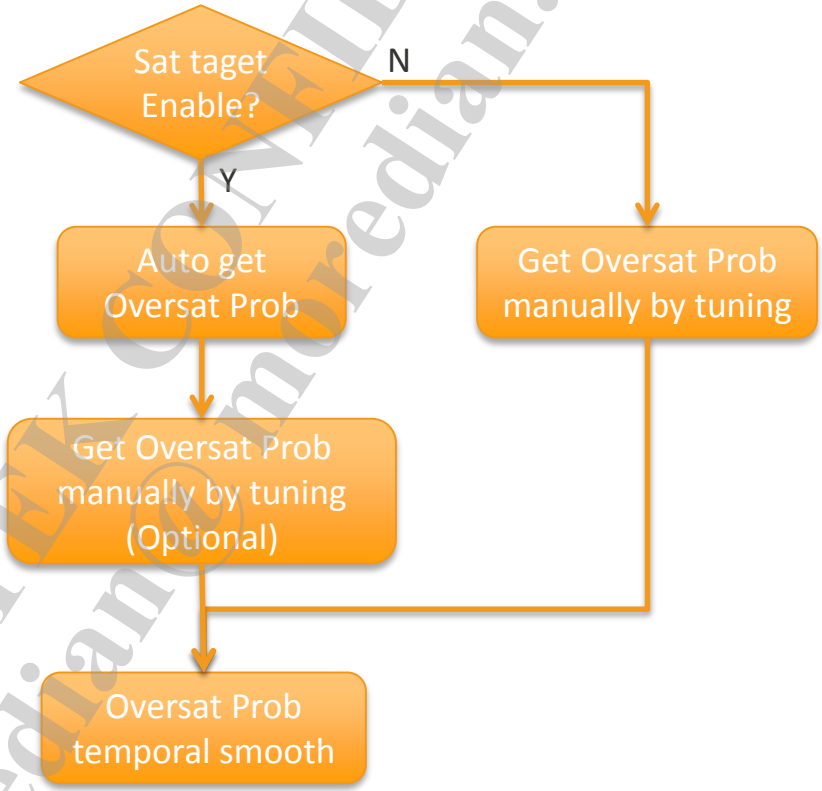
{R-, B+}

Oversat prob will also have a temporal smooth.



$$FinalGainRatio = \frac{(AddSatGainRatio \times (100 - Oversat\_Prob)) + (\mathrm{Re}\, duceSatGainRatio \times OversatProb)}{100}$$

Face Comp. AWB v1.5 Algorithm Saturation Control Design – Oversat Prob



Manual Get Oversat Prob

Sat target disable!

Yellowish AWB Gain for **general indoor scene.**

Bluish AWB Gain for **avoiding over-saturate face, includes outdoor or some indoor scene.**

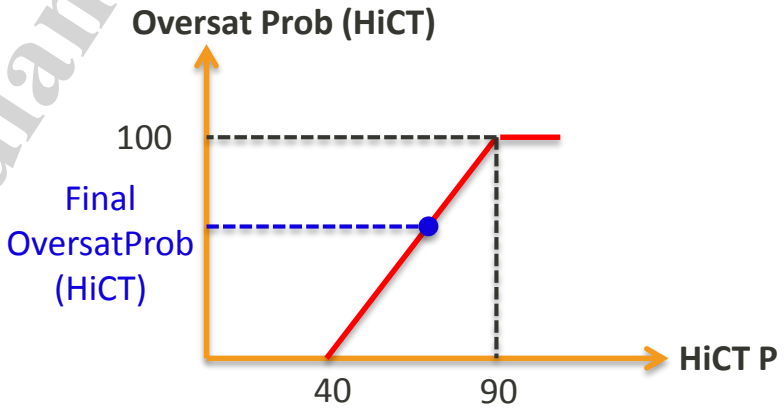
```
// rOversat
{
    // rLV
    {
        90,     //i4LVLow
        120,     //i4LVHi
        0,     //i4OverSatProb_Low
        100     //i4OverSatProb_Hi
    },
    // rHiCT
    {
        40,     //i4CT_P_Low
        90,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rMidCT
    {
        40,     //i4CT_P_Low
        90,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rLowCT
    {
        0,     //i4CT_P_Low
        0,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        0     //i4OverSatProb_Hi
    }
},
```

used for indoor & outdoor judgment

used for indoor high CT scene

used for indoor middle CT scene

used for indoor low CT scene



$$Oversat\_Prob = \max(LV\_Prob, HiCT\_Prob, MidCT\_Prob, LowCT\_Prob)$$

Manual Get Oversat Prob

Sat target enable!

Yellowish AWB Gain for **general indoor scene.**

Bluish AWB Gain for **avoiding over-saturate face, includes outdoor or some indoor scene.**

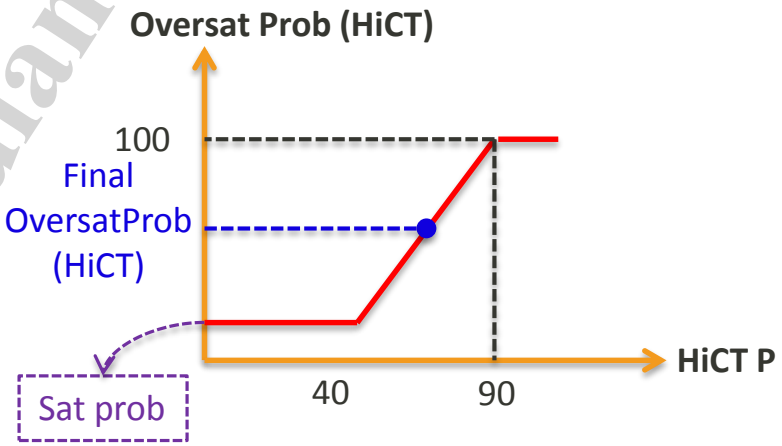
```
// rOversat
{
    // rLV
    {
        90,     //i4LVLow
        120,     //i4LVHi
        0,     //i4OverSatProb_Low
        100     //i4OverSatProb_Hi
    },
    // rHiCT
    {
        40,     //i4CT_P_Low
        90,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rMidCT
    {
        40,     //i4CT_P_Low
        90,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rLowCT
    {
        0,     //i4CT_P_Low
        0,     //i4CT_P_Hi
        0,     //i4OverSatProb_Low
        0     //i4OverSatProb_Hi
    }
},
```

used for indoor & outdoor judgment

used for indoor high CT scene

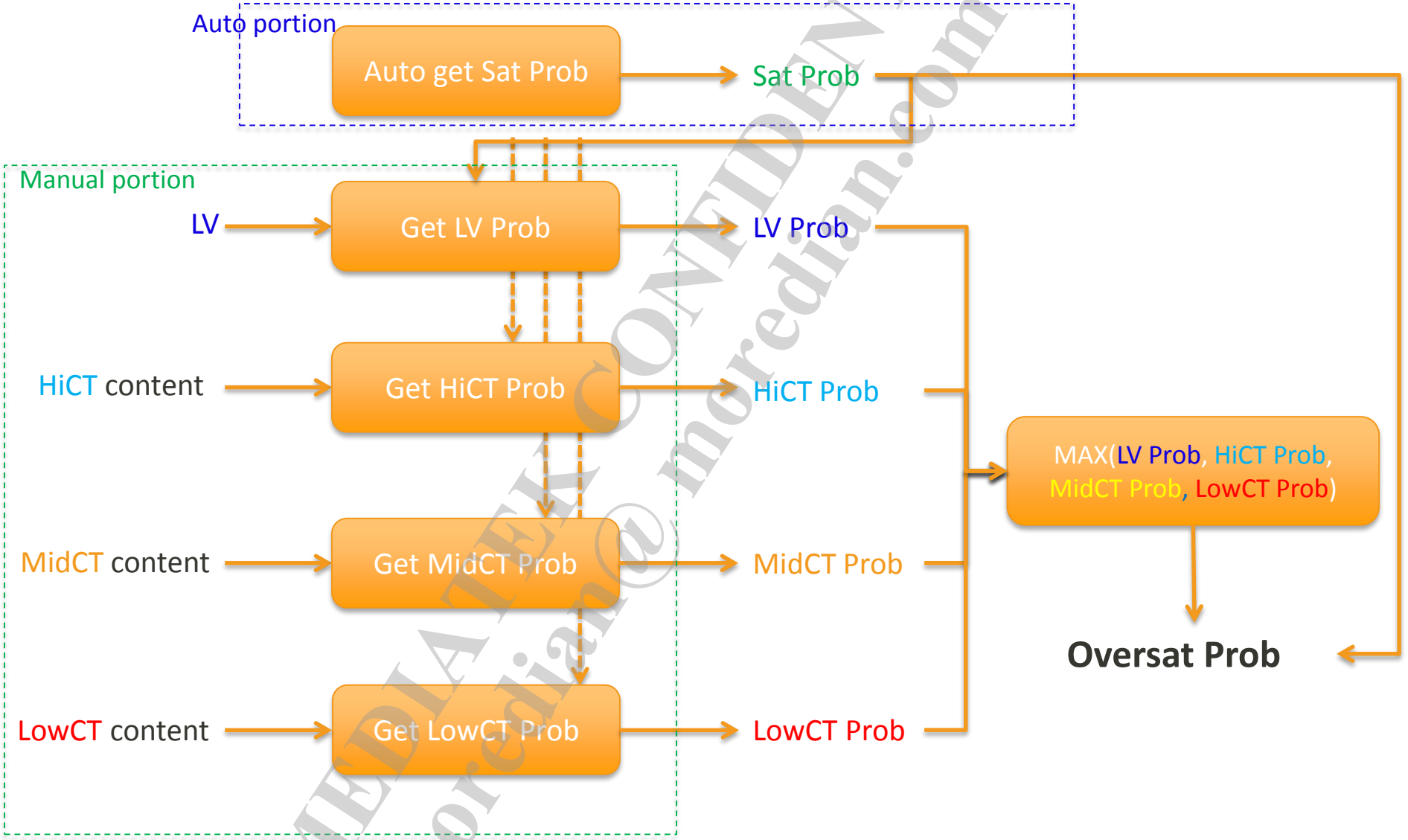
used for indoor middle CT scene

used for indoor low CT scene

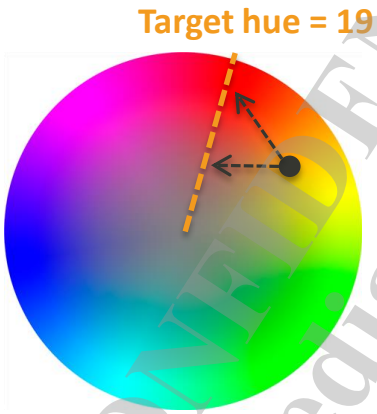


$$Oversat\_Prob = \max(LV\_Prob, HiCT\_Prob, MidCT\_Prob, LowCT\_Prob)$$

Get Oversat Prob



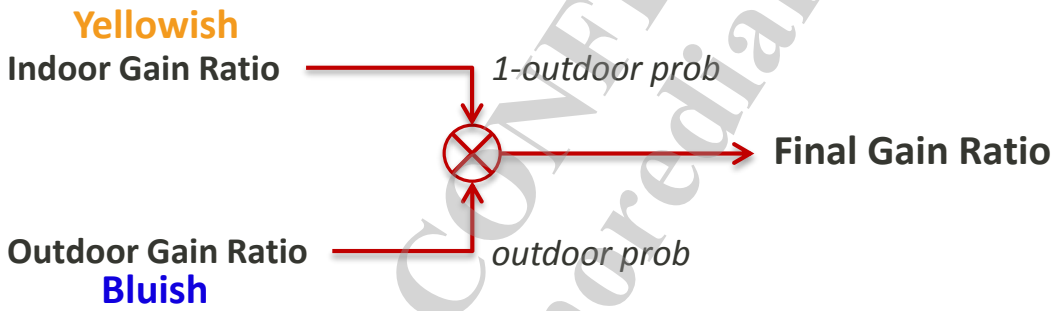
Get Face Comp. AWB Gain Ratio



Algo will automatically calculate 2 ratio (**yellowish** & **bluish**), no need to tuning.

Get Face Comp. AWB Gain Ratio

Outdoor prob will also have a temporal smooth.
Simulation won't work!

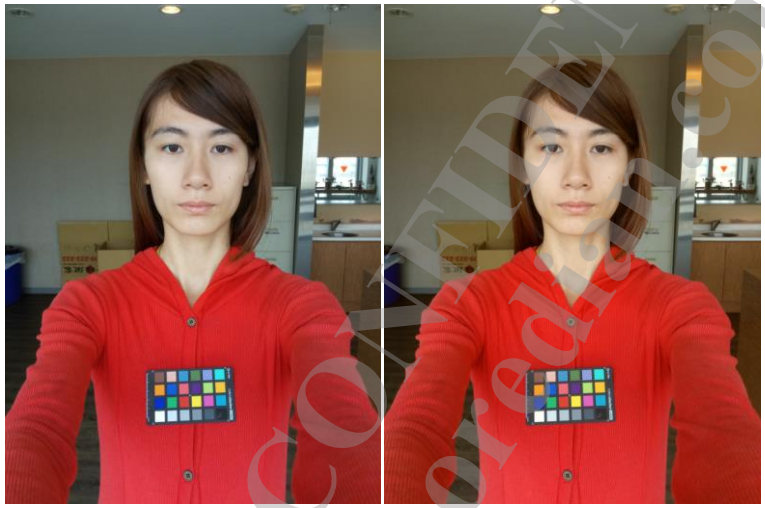


$$FinalGainRatio = \frac{\left(IndoorGainRatio \times \left(100 - Outdoor\_Prob\right)\right) + \left(OutdoorGainRatio \times OutdoorProb\right)}{100}$$

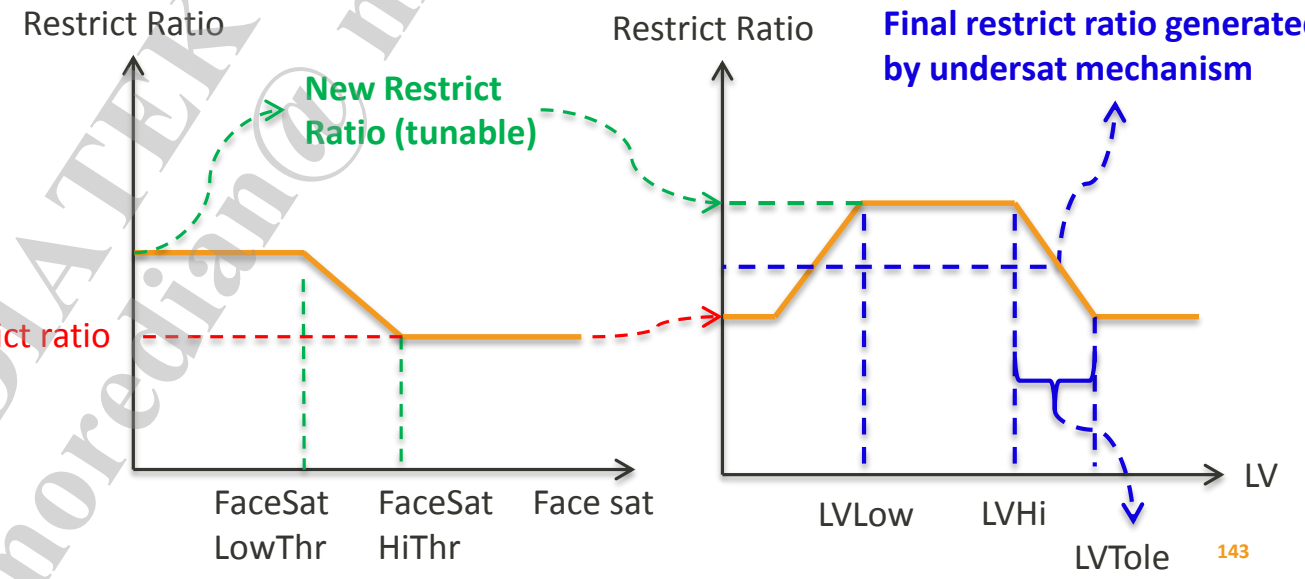
Undersat Prevention Mechanism (1/2)

Original

Modify



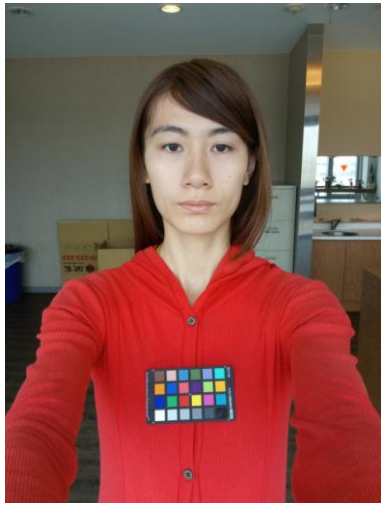
Face Comp. AWB v1.5 can force increasing restrict ratio when face is undersat.



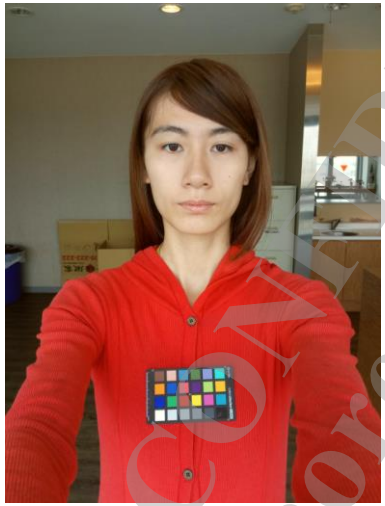

```
// rUndersat
{
    10000,     // i4SatHi
     5000,     // i4SatLow
       75,     // i4NewRestrict
       25,       // i4LVLow
       95,       // i4LVHi
       10          // i4LVTOL
},
```

Undersat Prevention Mechanism (2/2)

Original

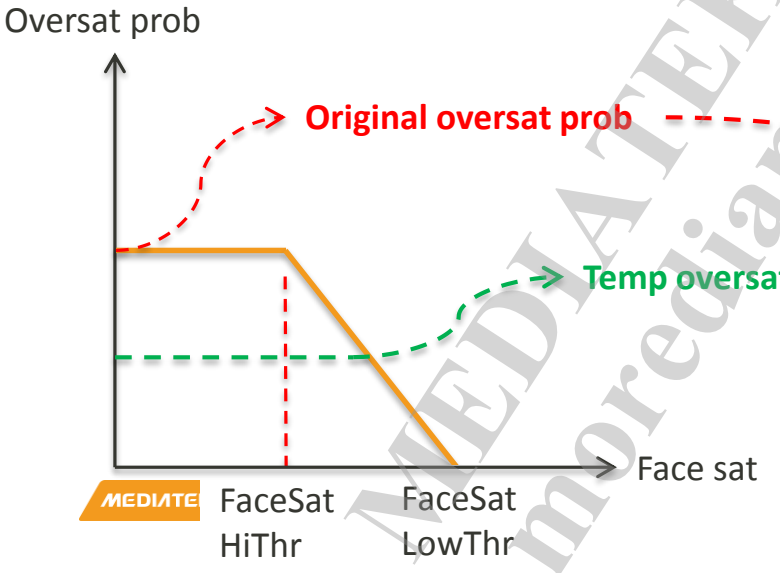


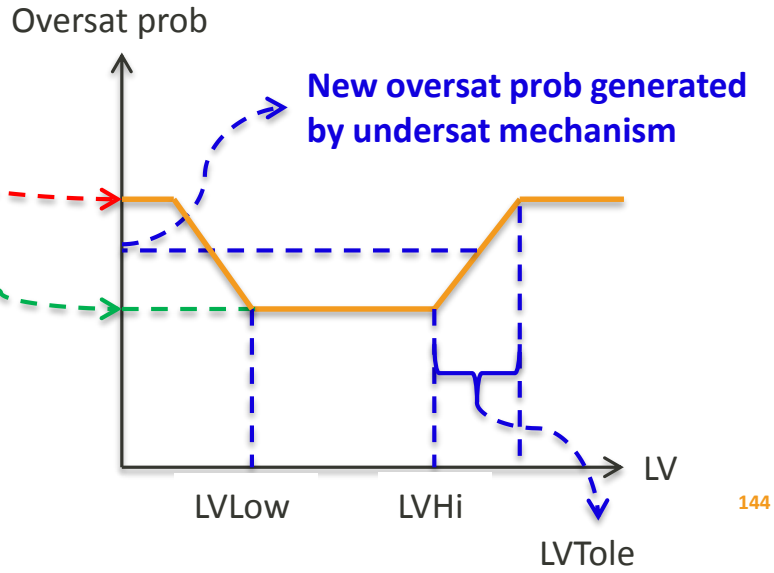
Modify



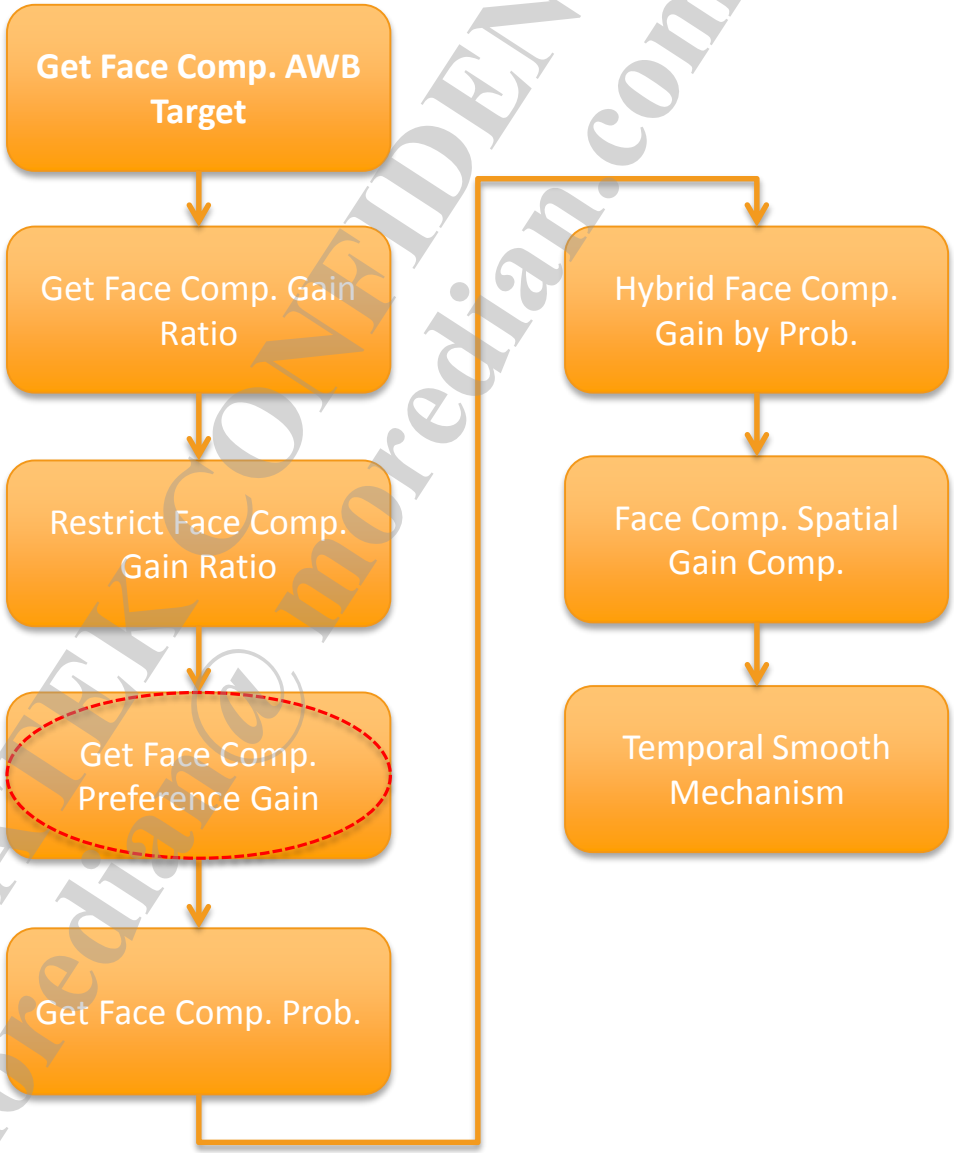
Face Comp. AWB v1.5 will force decreasing oversat prob when face is undersat.

```
// rUndersat
{
    10000,    // i4SatHi
     5000,    // i4SatLow
       75,    // i4NewRestrict
       25,      // i4LVLow
       95,      // i4LVHi
       10         // i4LVTOL
},
```





Face Comp. AWB v1.5 Comp.



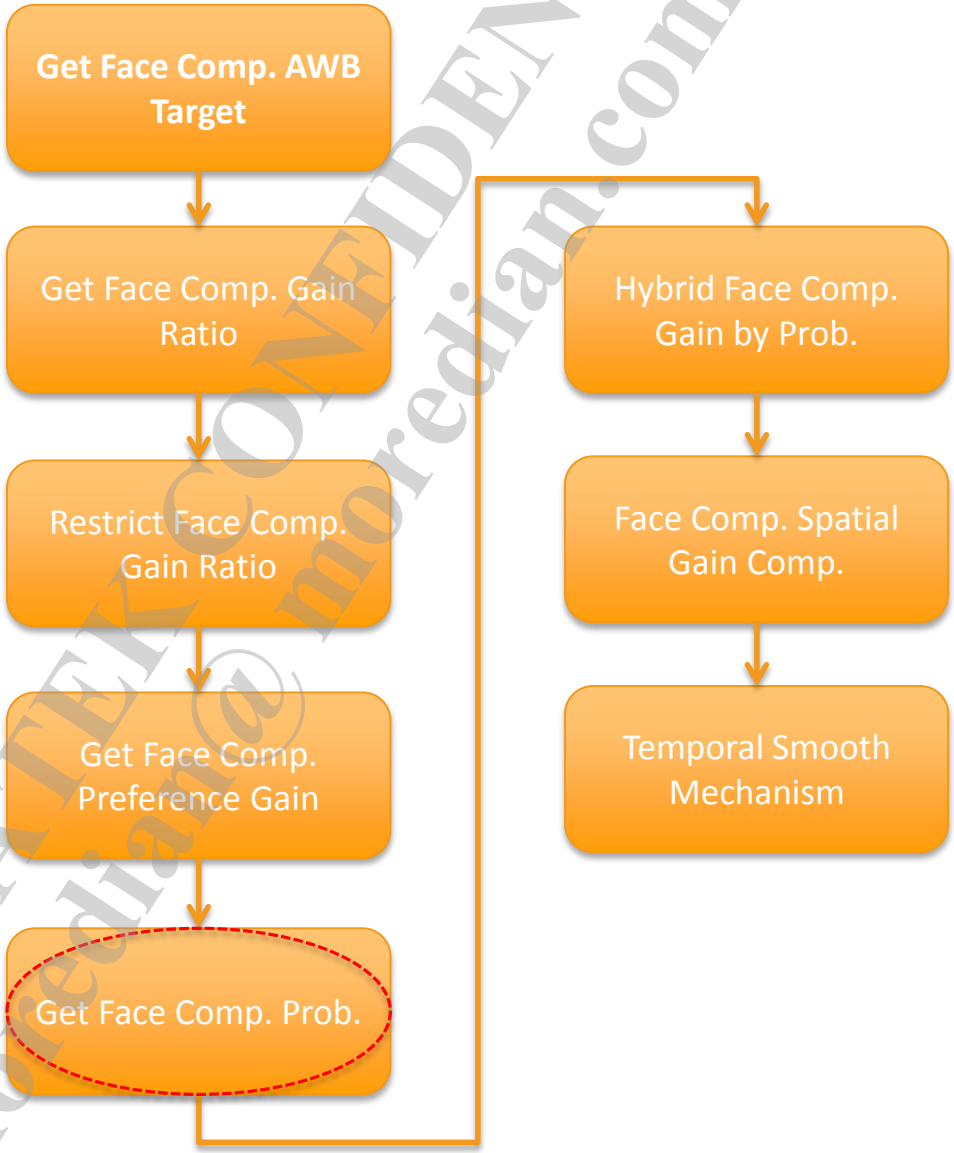
Get Face Comp. AWB Preference Gain

```
// rPrefGain
{
        //   LV0        LV1        LV2        LV3        LV4        LV5        LV6        LV7        LV8        LV9
        {512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},
        //         LV10          LV11        LV12        LV13        LV14        LV15        LV16        LV17        LV18
        {512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},
},
```

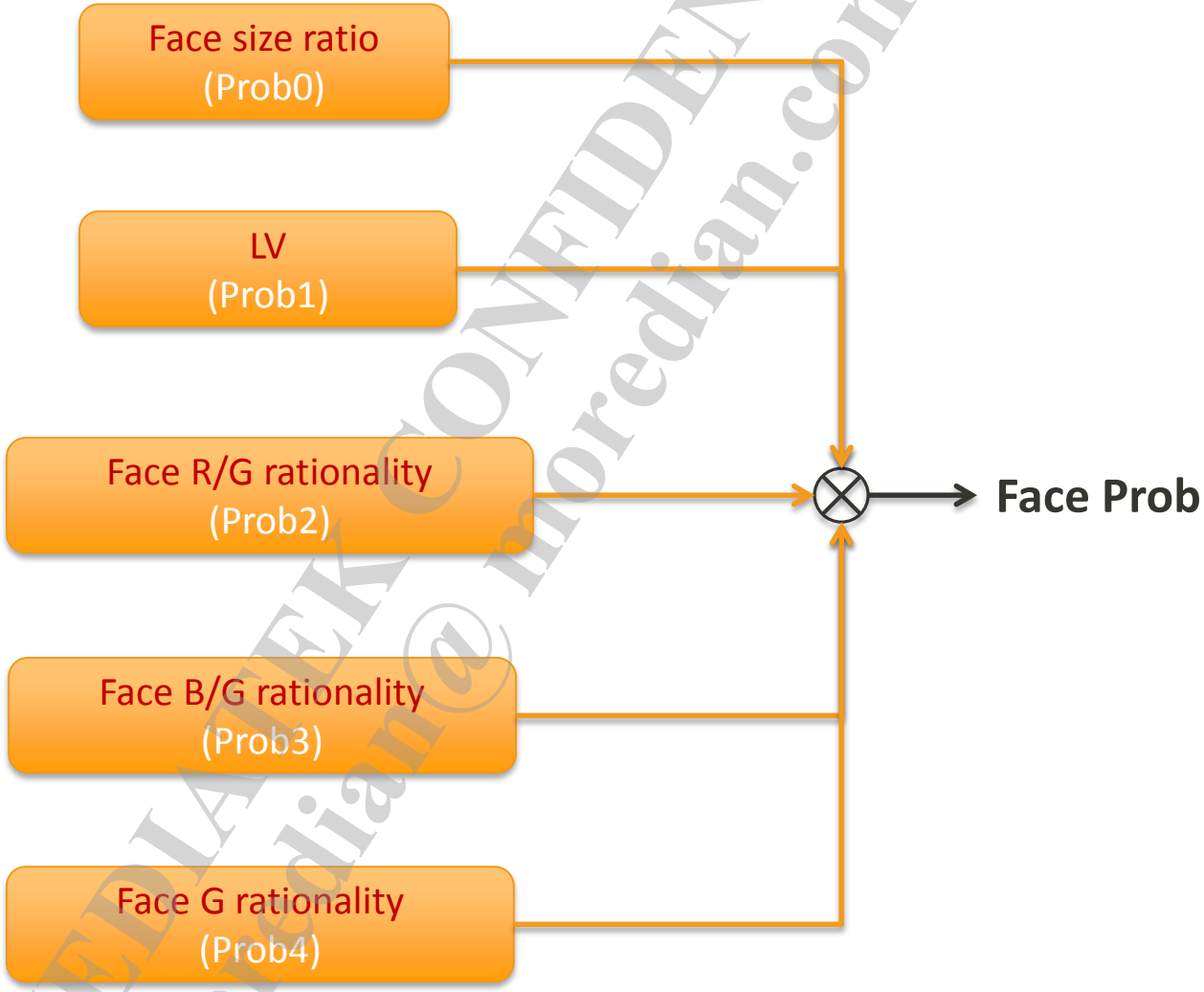
$$FaceAWBGain = FaceAWBGain \times PreferenceGain$$

Same with normal AWB preference gain

Face Comp. AWB v1.5 Comp.



Face Comp. AWB Probability

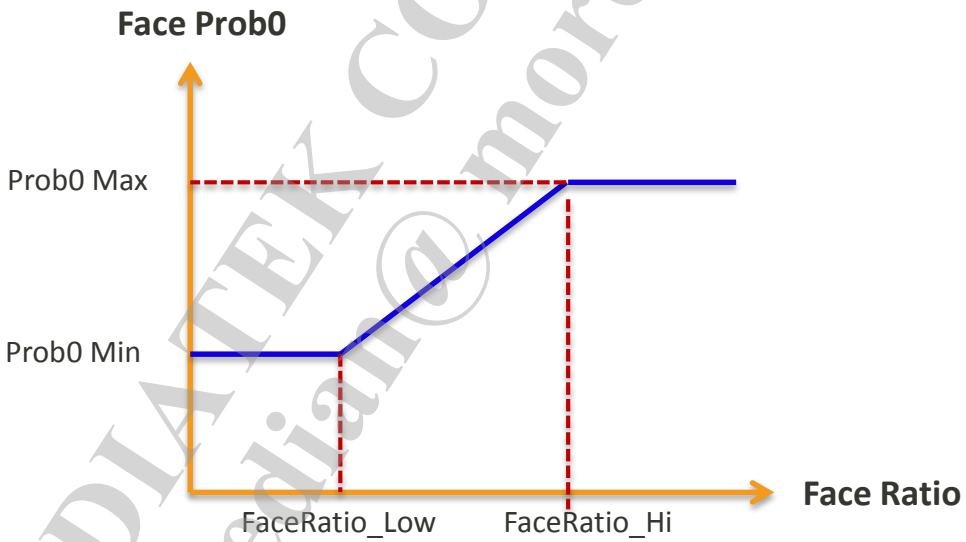


Get Face Comp. AWB Prob0

$$Face\_Prob0 = INTERPOLATE(Face\_Ratio, Face\_Min\_Ratio, Face\_Max\_Ratio, Min\_Prob0, Max\_Prob0)$$

$$Face\_Ratio = \frac{Face\_Area \times 1000}{120 * 90} \times 100$$

```
// rProb0
{
    3900,    //i4FaceSizeRatioLow
    6100,    //i4FaceSizeRatioHi
    0,       //i4Prob0_Min
    100      //i4Prob0_Max
},
```



Get Face Comp. AWB Prob1

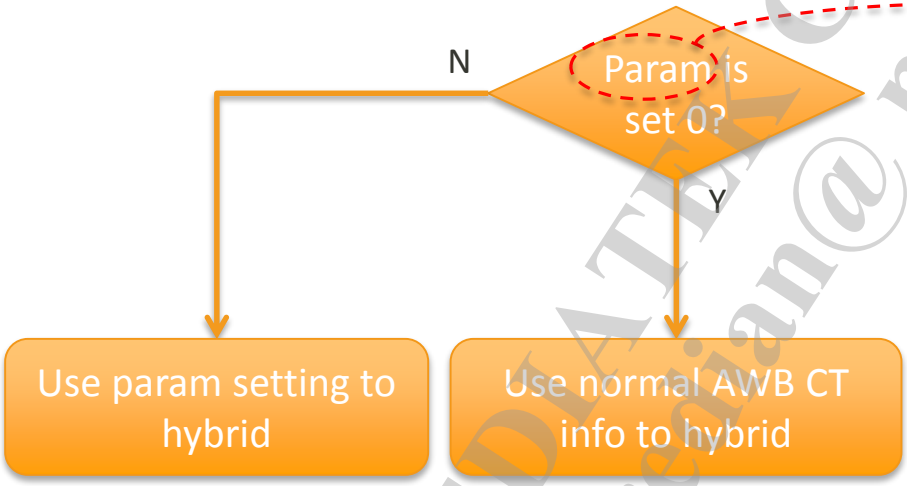
$$Face\_Prob1 = LV\_LUT$$

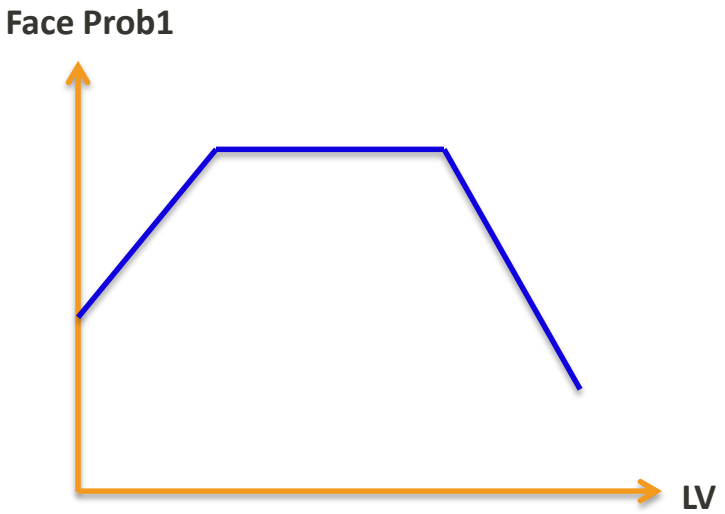
Same with normal AWB P1 LUT

```
// rProb1
{
    //LV0   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }, //i4HiCT_LUT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }, //i4MidCT_LUT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }  //i4LowCT_LUT
},
```

How to hybrid 3 set of P1?

```
// rSceneJudge
{
    90,    //i4LVLow for Indoor & Outdoor dynamic
    110,    //i4LVHi for Indoor & Outdoor dynamic
    2300,    //i4FaceRB_Low for Indoor Mid CT & Low CT dynamic, Unit = 1000
    3000     //i4FaceRB_Hi for Indoor Mid CT & Low CT dynamic, Unit = 1000
},
```





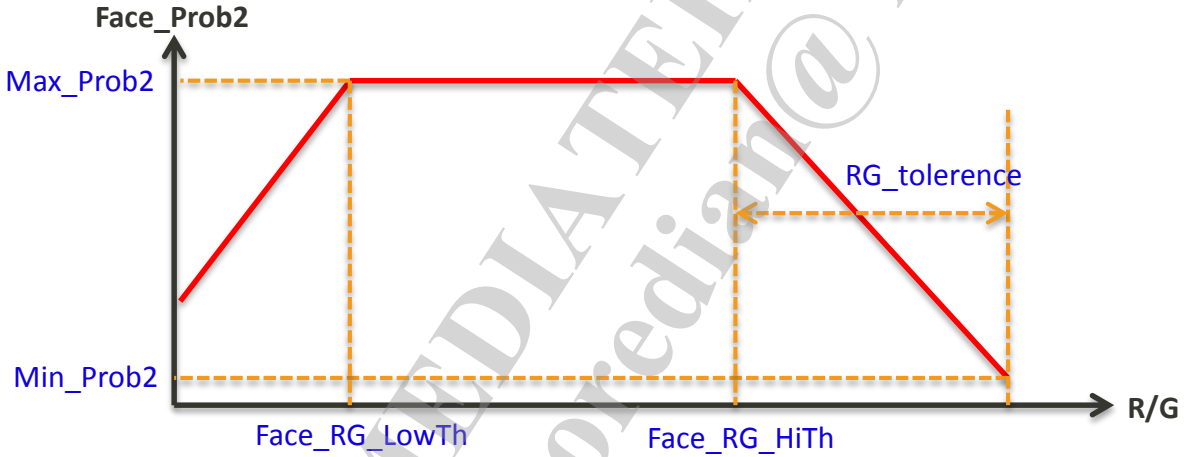
Get Face Comp. AWB Prob2 & Prob3

Face Prob2 is associated with face area R/G correctness.

Face Prob3 is associated with face area B/G correctness.

$$Face\_Prob2 = INTERPOLATE(Face\_RG, Face\_RG\_LowTh, Face\_LowMin, Max\_Prob2, Min\_Prob2)$$

$$Face\_Prob2 = INTERPOLATE(Face\_RG, Face\_RG\_HiTh, Face\_HiMax, Max\_Prob2, Min\_Prob2)$$




```
// rProb2
{
     400,    //i4FaceRG_Low Unit=1000
    1200,    //i4FaceRG_Hi Unit=1000
     500,    //i4FaceRG_TOL Unit=1000
       0,    //i4Prob2_Min
     100     //i4Prob2_Max
},
// rProb3
{
     250,    //i4FaceBG_Low Unit=1000
     800,    //i4FaceBG_Hi Unit=1000
     500,    //i4FaceBG_TOL Unit=1000
       0,    //i4Prob3_Min
     100     //i4Prob3_Max
},
```

Get Face Comp. AWB Prob4

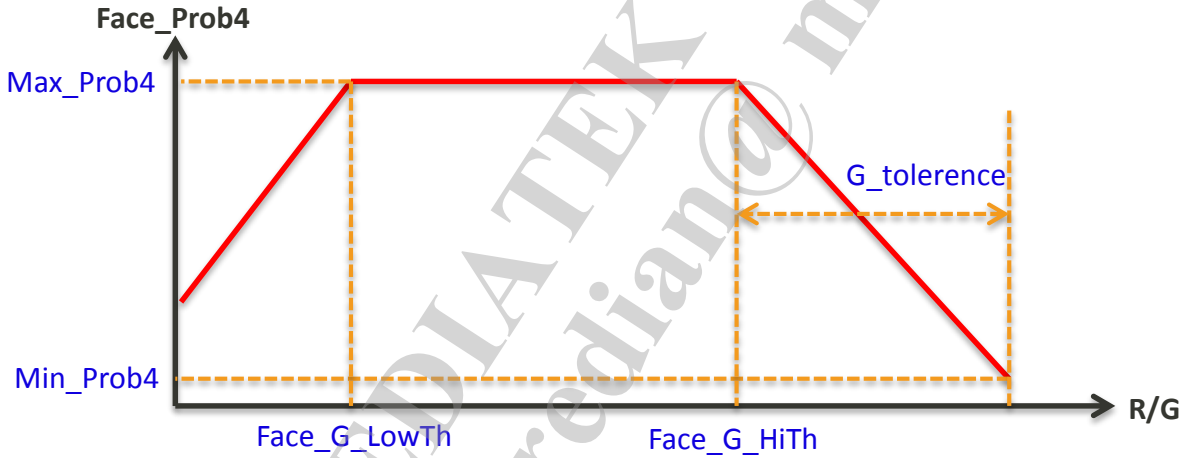
Why we need P4?

1. When face is dark or saturated, the R/G or B/G is not accuracy, and also not stable.

2. In some condition we cannot tune it by P1. e.g. the LV is high but face is dark in left scenes.

3. For some special application, face AWB only want to focus on one specific kind of skin tone.

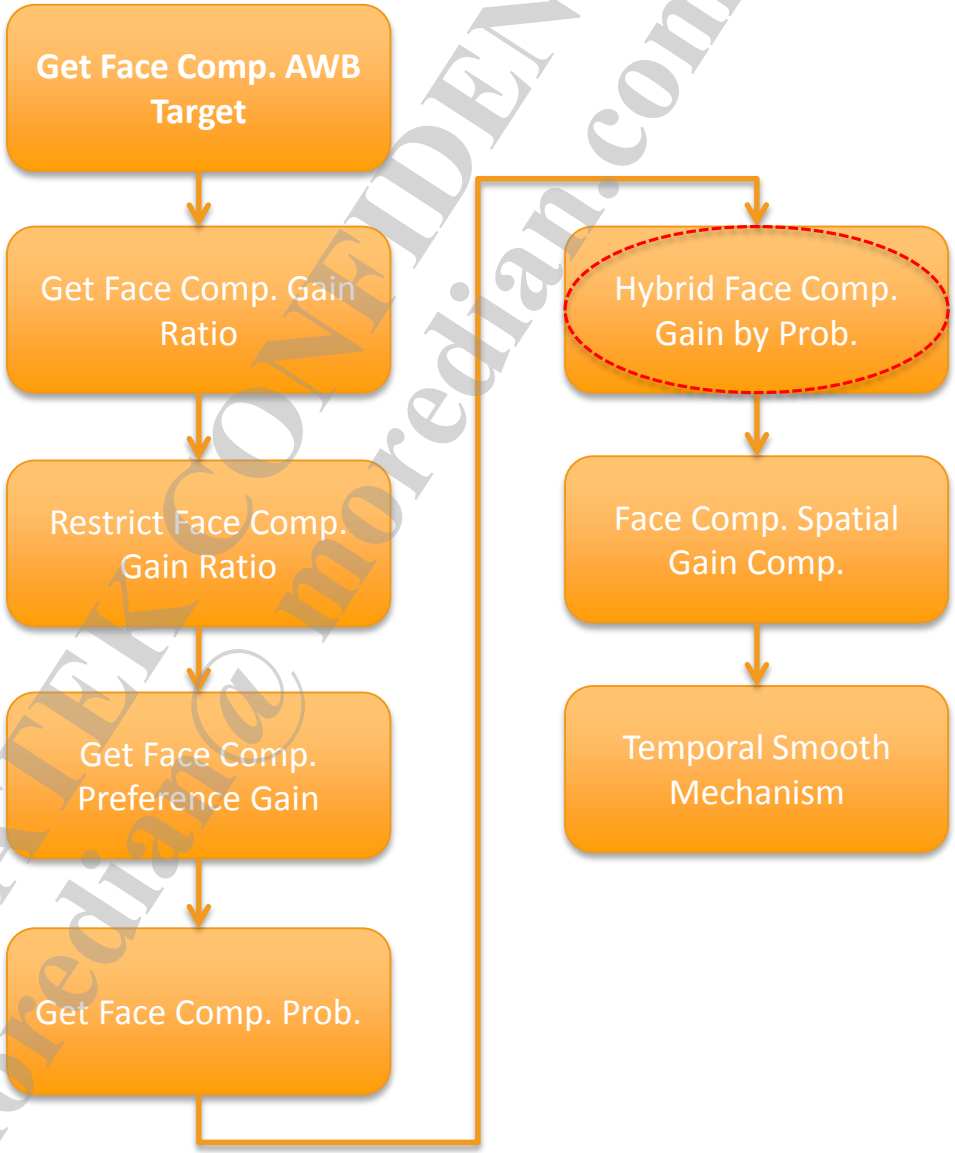
$$Face\_Prob4 = INTERPOLATE(Face\_G, Face\_G\_LowTh, Face\_G\_LowMin, Max\_Prob4, Min\_Prob4)$$




```
// rProb4
{
        0,    //i4FaceG_Low Unit=1000
        0,    //i4FaceG_Hi Unit=1000
        0,    //i4FaceG_TOL Unit=1000
      100,    //i4Prob4_Min
      100     //i4Prob4_Max
},
```

$$Face\_Prob4 = INTERPOLATE(Face\_G, Face\_G\_HiTh, Face\_G\_HiMax, Max\_Prob4, Min\_Prob4)$$

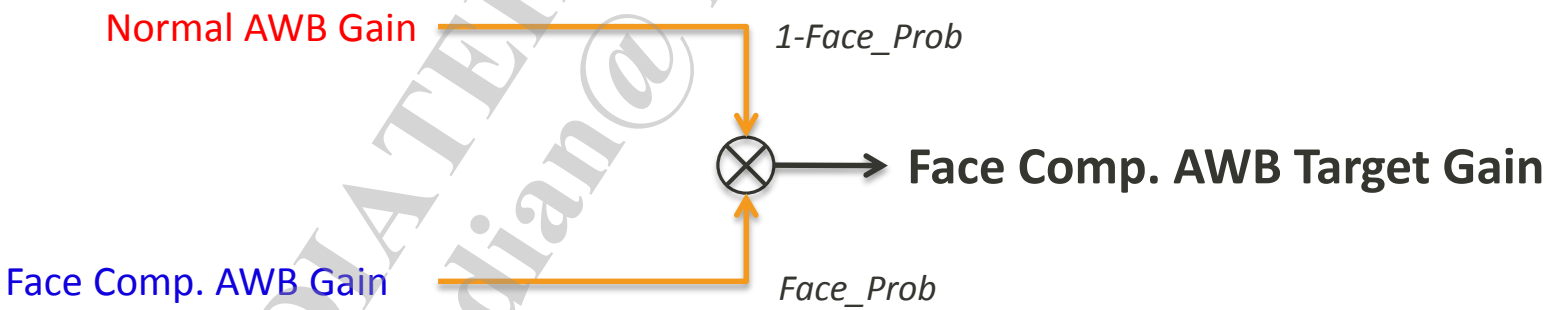
Face Comp. AWB v1.5 Comp.



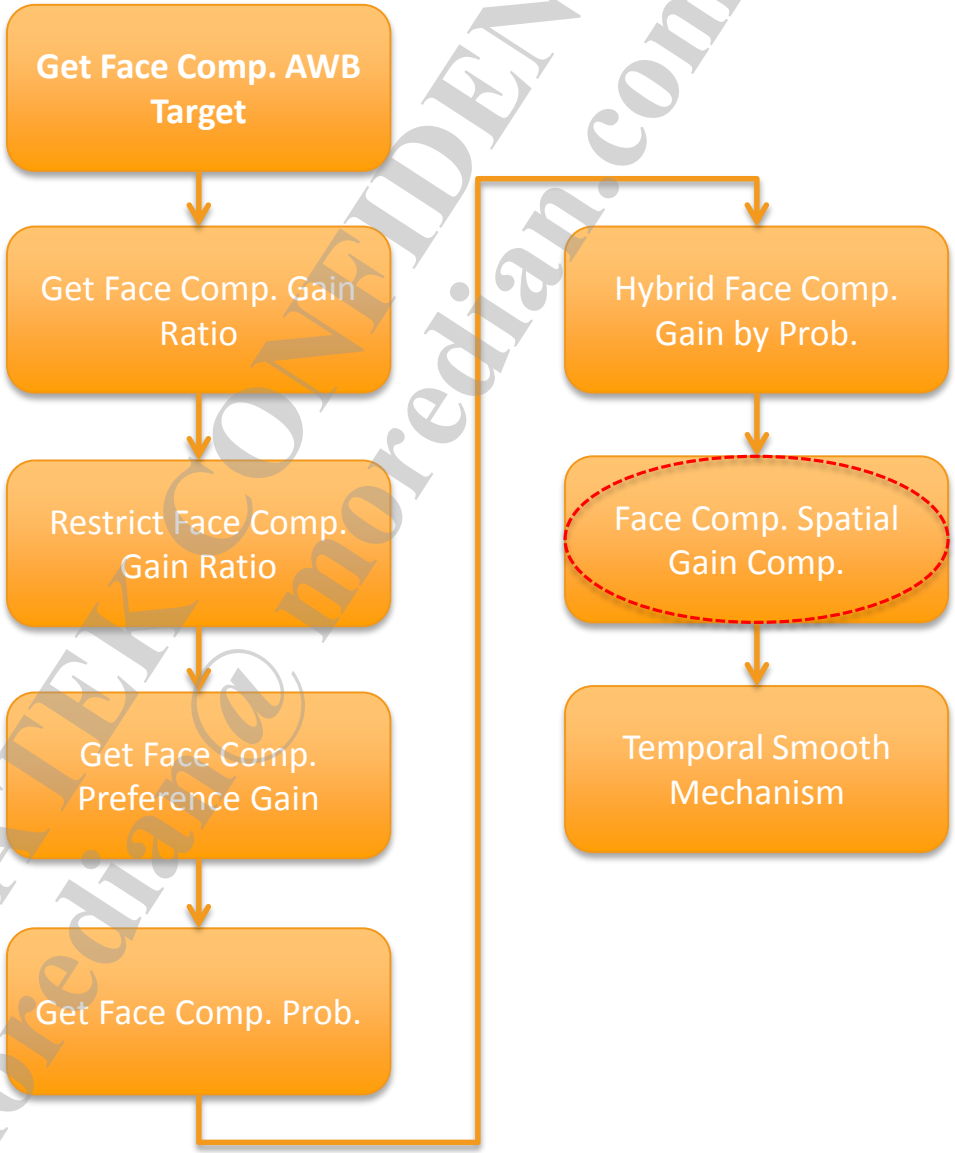
Get Face Comp. AWB Prob & Hybrid Face Comp. AWB Gain

$$Face\_Prob = Face\_Prob0 \times Face\_Prob1 \times Face\_Prob2 \times Face\_Prob3 \times Face\_Prob4$$

$$FaceAWBTargetGain = \frac{(Normal\_AWB\_Gain \times (100 - Face\_Prob)) + (Face\_AWB\_Gain \times Face\_Prob)}{100}$$



Face Comp. AWB v1.5 Comp.



Face Comp. AWB Advanced Compensation

Face AWB compensation coverage (focus on large face size scenario)

Sometimes face size is small, and neutral statistic data is not enough.

In this case, statistic data is not correct and not enough, normal AWB might not accuracy.

For this scenario, there is an optional advanced compensation to hybrid spatial gain.

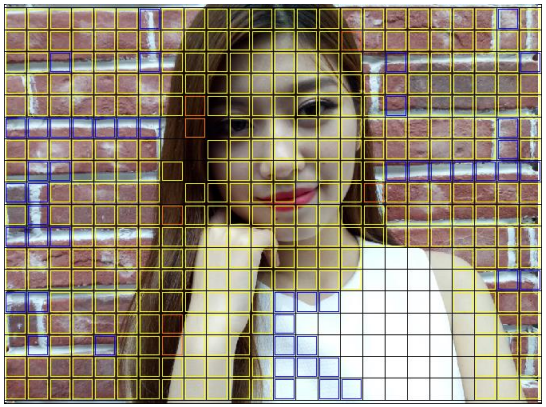
Face Comp. AWB Spatial Gain

Face Comp. AWB Spatial Prob0

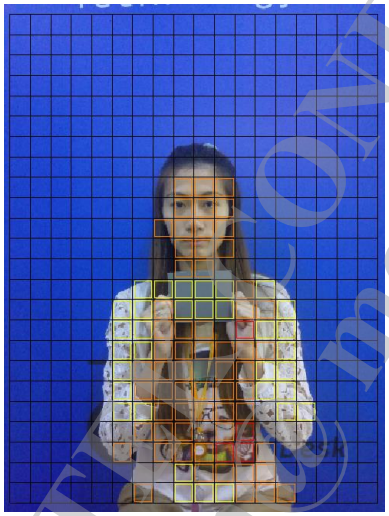
Spatial Prob0: Consider face size ratio

$$Face\_Spatial\_Prob0 = INTERPOLATE(Face\_Ratio, Face\_Max\_Ratio, Face\_Min\_Ratio, Min\_Prob0, Max\_Prob0)$$

No need to use spatial gain

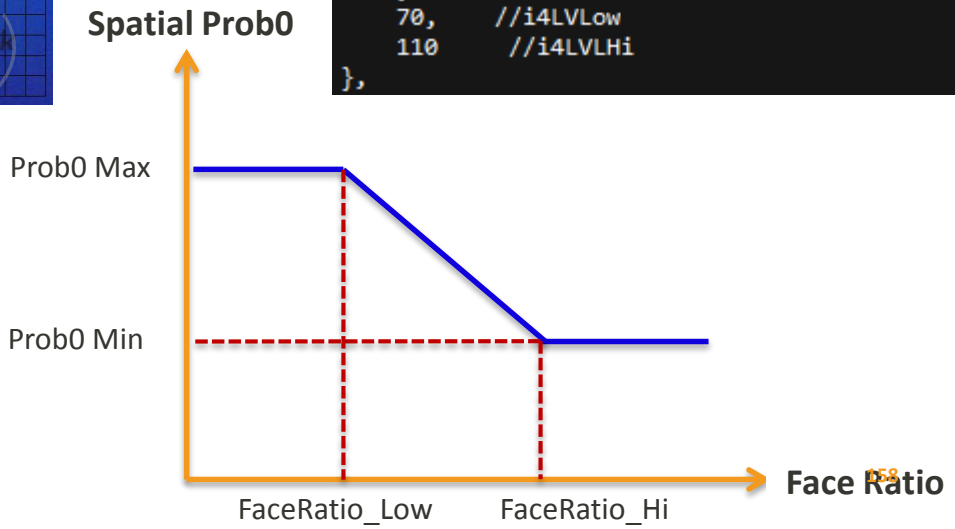


Need to use spatial gain



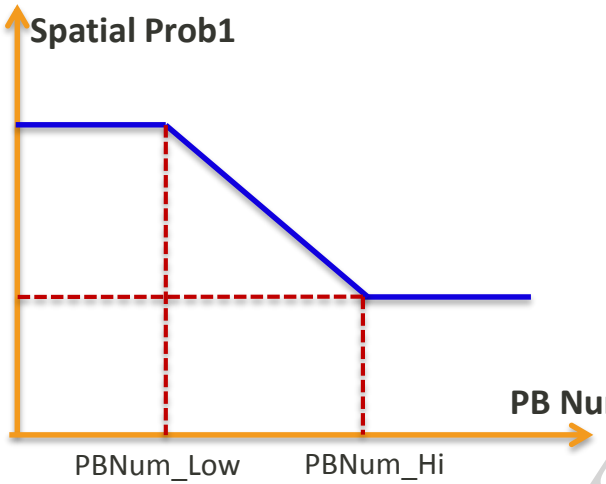
```
// rSpatial
{
    // rP0
    {
        3900,    //i4FaceSizeRatioHi  Unit=1000
        500,    //i4FaceSizeRatioLow Unit=1000
        0,    //i4Prob0_Min
        50     //i4Prob0_Max
    },
    // P1
    {
        75,    //i4NeutralParentBlkNum_Hi
        15,    //i4NeutralParentBlkNum_Low
        0,    //i4Prob1_Min
        50     //i4Prob1_Max
    },
    70,    //i4LVLow
    110     //i4LVLHi
},
```

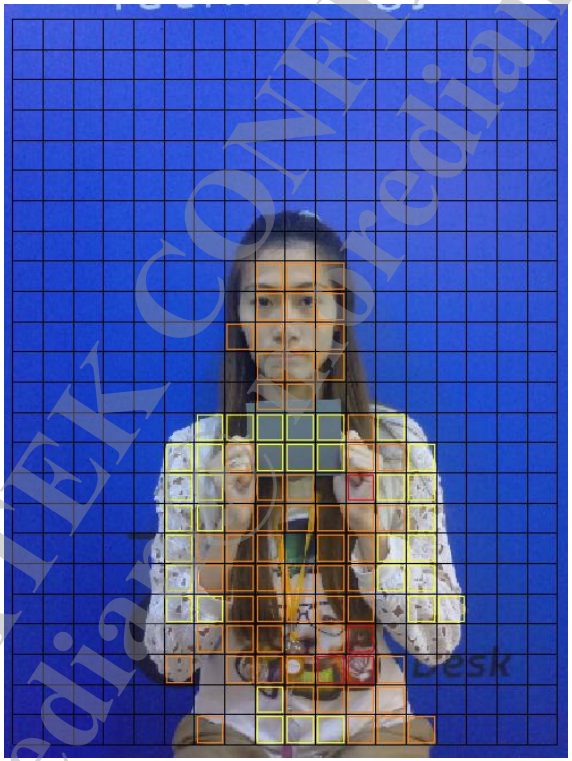
Different with Face Prob0, Spatial Prob0 will be smaller when face size ratio is higher.



Face Comp. AWB Spatial Prob1

Spatial Prob1: Consider neutral parent block number which exclude face area



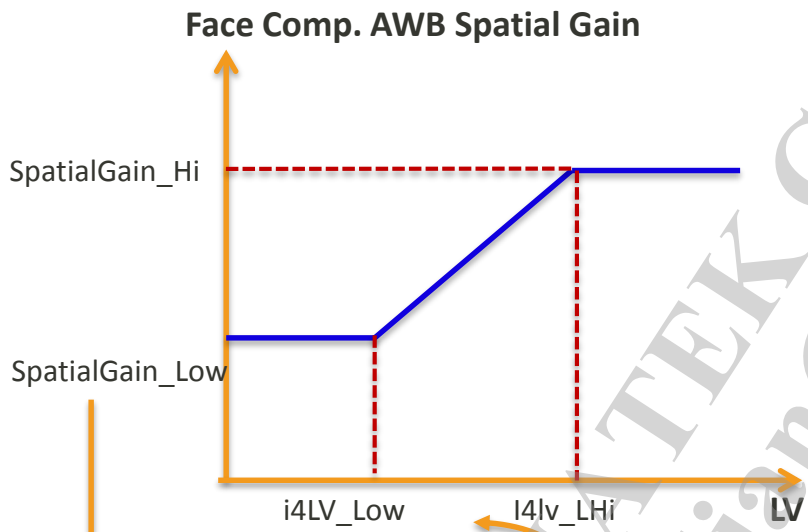


```
// rSpatial
{
    // rP0
    {
        3900,    //i4FaceSizeRatioHi  Unit=1000
        500,    //i4FaceSizeRatioLow Unit=1000
        0,    //i4Prob0_Min
        50     //i4Prob0_Max
    },
    // P1
    {
        75,    //i4NeutralParentBlkNum_Hi
        15,    //i4NeutralParentBlkNum_Low
        0,    //i4Prob1_Min
        50     //i4Prob1_Max
    },
    70,    //i4LVLow
    110    //i4LVLHi
},
```

Face Comp. AWB Spatial Gain Compensation

$$Face\_Spatial\_Prob = Face\_Spatial\_Prob0 \times Face\_Spatial\_Prob1$$

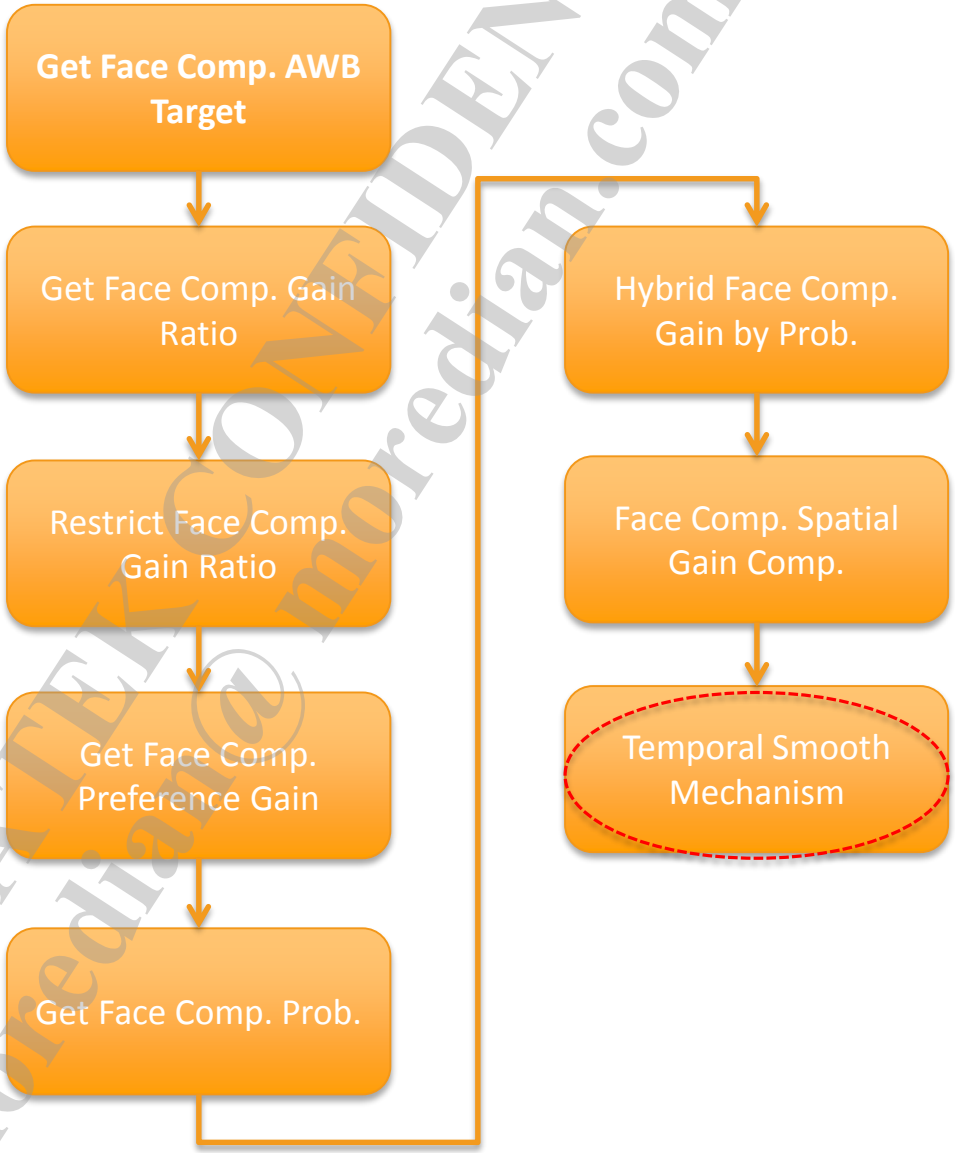
$$FaceAWBTargetGain = \frac{(FaceAWBTargetGain \times (100 - Face\_Spatial\_Prob)) + (FaceAWBSpatialGain \times Face\_Spatial\_Prob)}{100}$$




```
// rSpatial
{
    // rP0
    {
        3900,    //i4FaceSizeRatioHi  Unit=1000
        500,    //i4FaceSizeRatioLow Unit=1000
        0,    //i4Prob0_Min
        50     //i4Prob0_Max
    },
    // P1
    {
        75,    //i4NeutralParentBlkNum_Hi
        15,    //i4NeutralParentBlkNum_Low
        0,    //i4Prob1_Min
        50     //i4Prob1_Max
    },
    70,    //i4LVLow
    110     //i4LVLHi
};
```

Spatial gain採用normal awb 的spatial L和spatial H

Face Comp. AWB v1.5 Comp.

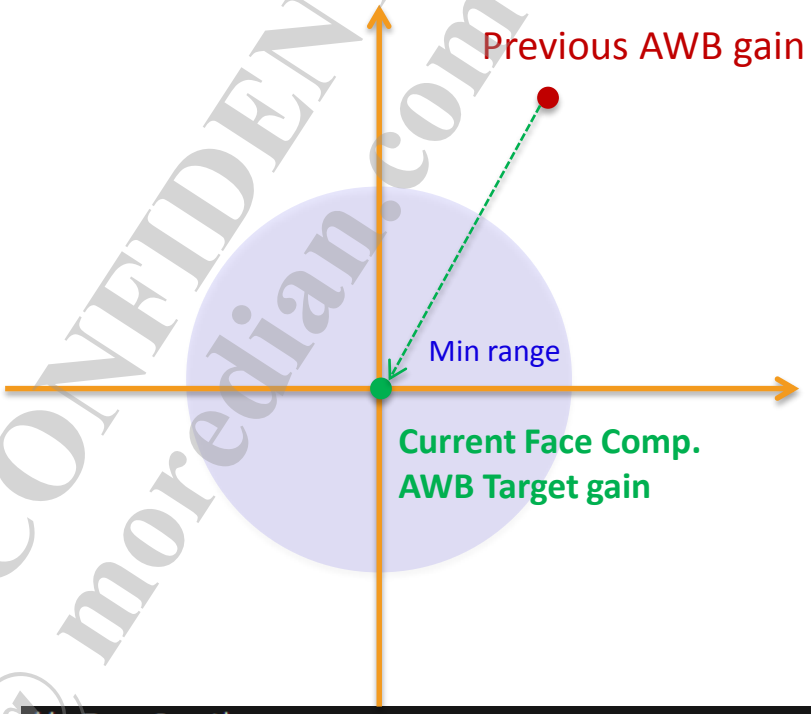


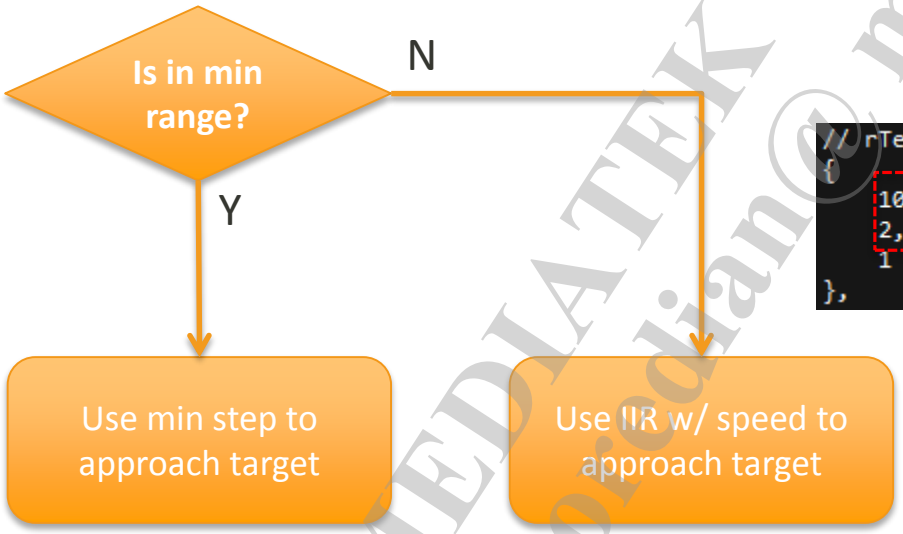
Temporal Face Comp. AWB Gain Mechanism

IIR approaching target：

1. Converge speed = 1~100
2. Minimum step

Minimum range = MinStep*100/speed



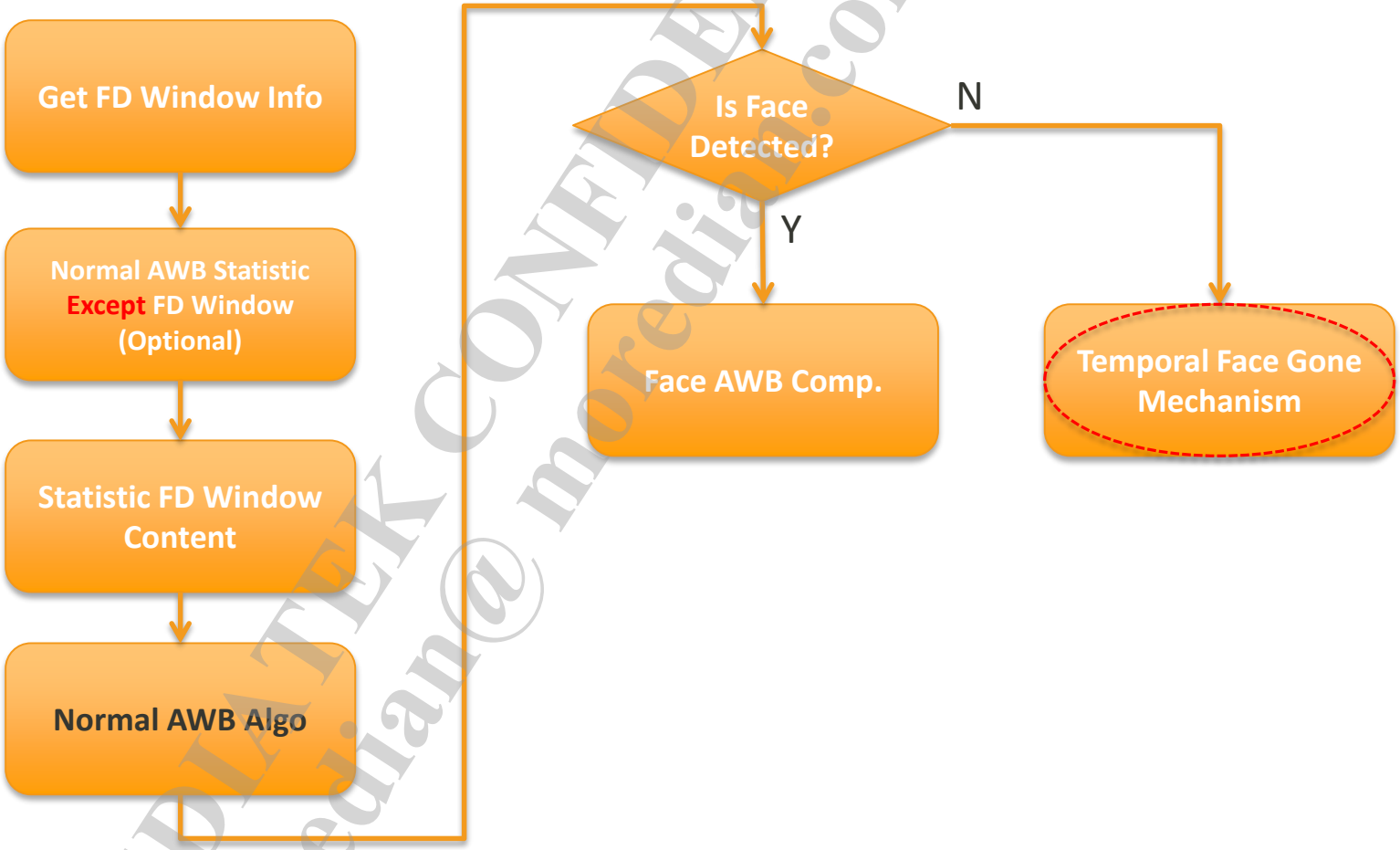



```
// rTempoSmooth
{
    10,    //i4Speed: Converge Speed 1~100
    2,    //i4MinStep: Minimum converge step
    1     //i4ProbReduceStep: If face gone, the face prob reduce step 1~100
},
```

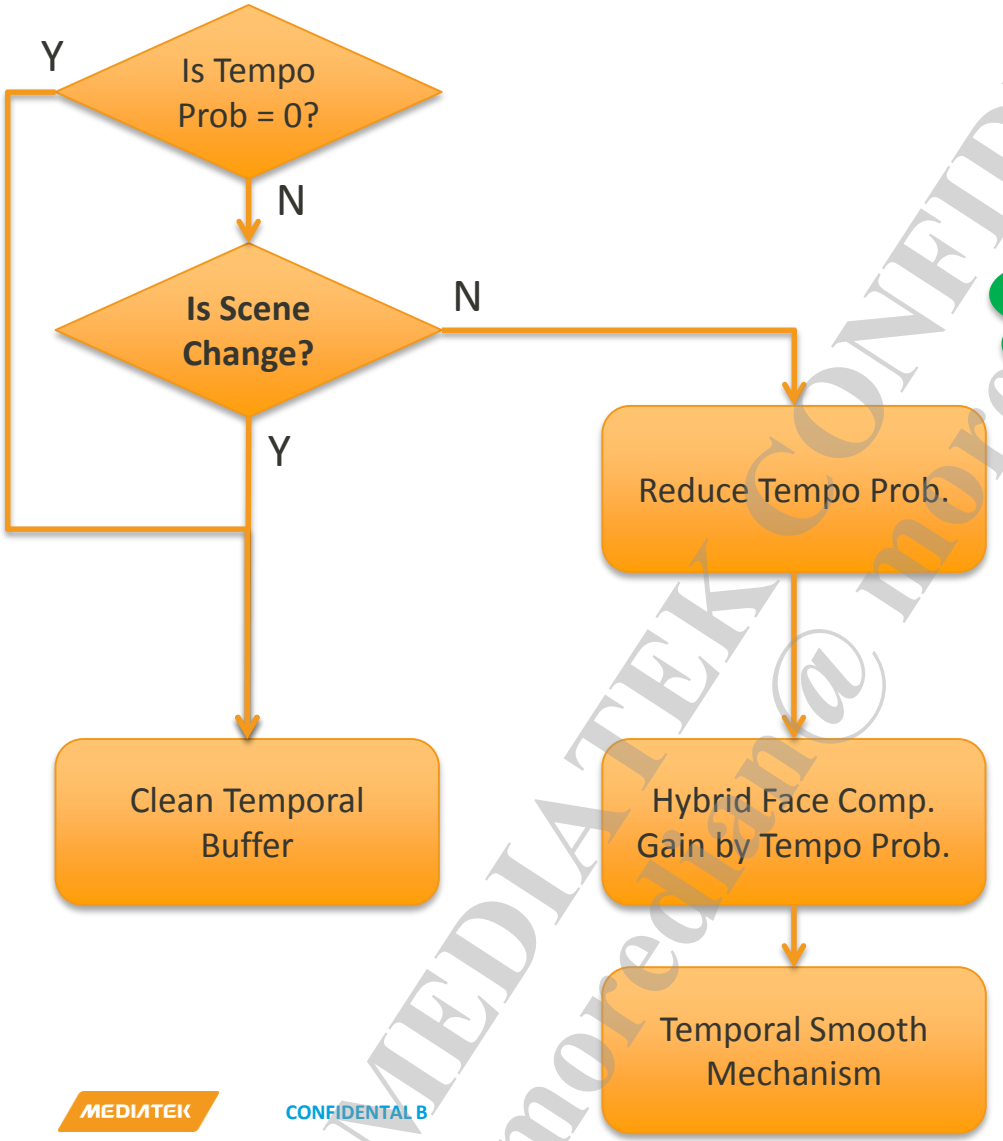
Same with normal AWB temporal smooth mechanism

Back (Face-Assist)

Currently Face Comp. AWB Algo Flow



Temporal Face Gone Mechanism




```
// rTempoSmooth
{
    10,     //i4Speed: Converge Speed 1~100
    2,      //i4MinStep: Minimum converge step
    1       //i4ProbReduceStep: If face gone, the face prob reduce step 1~100
},
```

Tempo Prob will be set 100 when the face is detected.

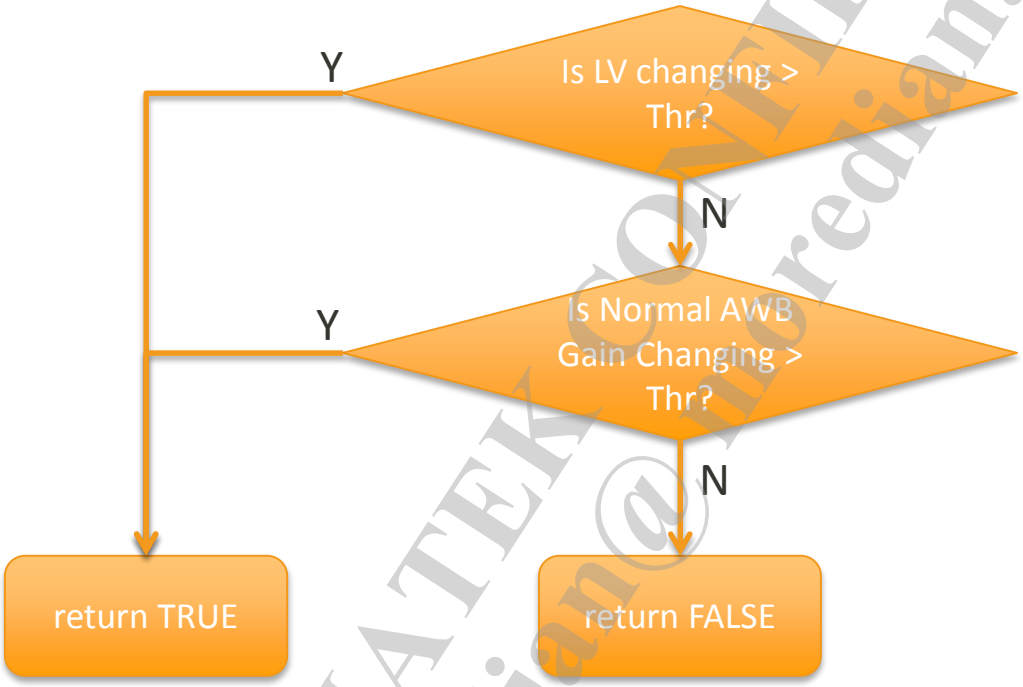
Reduce Step is Tunable

Using the latest Face comp AWB Target gain to hybrid with current normal AWB gain

Same as face awb comp using mechanism.

Back (Face-Assist)

Simply Scene Change Detection

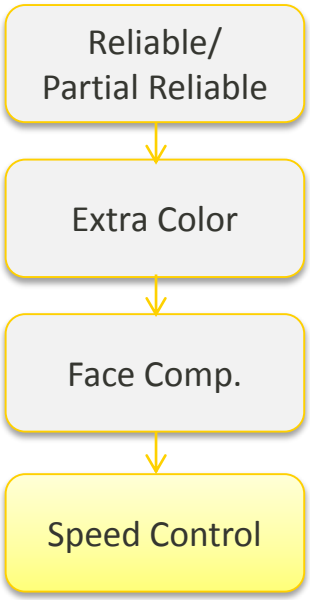


Compare current LV & last Face Comp. AWB temporal LV

Compare current normal AWB gain & last Face Comp. AWB temporal normal AWB gain

```
// rSceneChange
{
    30,   //i4LVChangeTh
    20000    //i4AWBGainChangeTh
},
```

Speed Control




```
// AWB convergence parameter
    {
        10, // i4Speed: Convergence speed: (0 ~ 100)
       225, // i4StableThr: Stable threshold ((currentRgain - targetRgain)^2 +
                                              (currentBgain - targetBgain)^2), WB gain format: 4.9
        1  // i4ToTargetEnable: Preview converge to target enable.
    },
```

Stable	Unstable
SmoothGain = TargetGain	smoothGain = 10% * TargetGain + 90% * PrevGain

Tuning Parameters

AWB v5.0

Parameters

- Parameter files path
  - **Path: vendor\mediatek\proprietary\custom\$project\hal\imgsensor\ver1\$sensor\$scenario**
    - $sensor_$scenario_AWB.cpp

      Usage: AWB parameters that relate to module's characteristics and preference (NVRAM)

      AWB parameter that has less to do with the module's characteristics (Hard Code)

      Move from awb_tuning_custom_xxx.cpp in AWB v5.0

  - **Path: vendor\mediatek\proprietary\custom\$project\hal\camera_3a**
    - awb_tuning_custom_main.cpp
    - awb_tuning_custom_main2.cpp
    - awb_tuning_custom_sub.cpp

    Be empty and move to camera_tuning_para_$scenario_$sensor.cpp in AWB v5.0

Parameters

- **Light Source Estimation**
  1. Child Block -> Parent Block Light Estimation
  2. Light Source Estimation
     P0, P1, P2

- **Predict AWB Gain by Light Source Info.**
  1. Preference Color Compensation
  2. Statistic Color Constraint
  3. Spatial Predictor
  4. Sunset compensation
  5. Daylight Locus Probability
  6. Noise Reduction
  7. Sub-window
  8. Preference Gain
  9. Face Assisted AWB

- **Spatial & Temporal Predictor**
  1. Spatial Predictor
  2. Temporal Predictor

- **Get AWB Output Gain**
  1. Reliable mode
  2. Extra Color
  3. Face Comp. AWB

Parameters

- **Light Source Estimation**
  1. Child Block -> Parent Block Light Estimation
  2. Light Source Estimation
     P0, P1, P2

- **Predict AWB Gain by Light Source Info.**
  1. Preference Color Compensation
  2. Statistic Color Constraint
  3. Spatial Predictor
  4. Sunset compensation
  5. Daylight Locus Probability
  6. Noise Reduction
  7. Sub-window
  8. Preference Gain
  9. Face Assisted AWB

- **Spatial & Temporal Predictor**
  1. Spatial Predictor
  2. Temporal Predictor

- **Get AWB Output Gain**
  1. Reliable mode
  2. Extra Color
  3. Face Comp. AWB

Parent Block Light Estimation

```
2,    //i4CbMinThr
```

Feature name:
Property for low light stability
variable name: i4CbMinThr
data range: 0 ~ 25
The minimum value for valid statistic block
T0' = Max (T0, i4CbMinThr)

Light Source Statistics

```
// Smooth Statistic
{
    1,    //i4Enable
    {0, 23, 23, 23, 23, 23, 23, 23},  //i4StatWinShrinkOffset
    0,    //Reserved1
    0,    //Reserved2
    0,    //Reserved3
    0,    //Reserved4
    0,    //Reserved5
},
```

Feature name: Statistic Smooth

Variable name: i4Enable
Data range: 0 ~ 1

Variable name: i4StatWinShrinkOffset
Data range: N/A

AWB
v5.0

Light Source Statistics

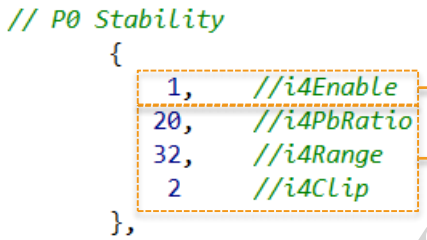
```
// Linear AAO parameter
{
    1, //i4Enable
    1  //i4ErrCntThr
},
```

Feature name: Linear AAO statistics
Variable name: i4ErrCntThr
Data range: 0 ~ 15
When CB error count is larger than i4ErrCntThr, the CB statistics will be replaced by the linear AAO

Feature name: Parent block weighting
Variable name: i4Mode
Data range: 0 ~3
Mode 1: use i4Scaling Factor as weighting
Mode 2: use PB light source color temperature to determine weighting LUT
Mode 3: use LV look up i4LvThld to determine weighting LUT

```
// Parent block weight parameter
{
    1, // i4Mode : 0:Disable 1:Linear weighting 2:Weighting LUT 3:Weighting LUT
    8,  // i4ScalingFactor: [6] 1~12, [7] 1~6, [8] 1~3, [9] 1~2, [>=10]: 1
    {20, 70, 120}, // i4LvThld[3]
    // Gamma LUT
    {0, 64, 90, 111, 128, 143, 156, 169, 181, 192, 202, 212, 221, 230, 239, 247, 256},
    // Weighting LUT for High Mid Low color temperature
    {{ 4, 12, 14, 13, 10, 7, 6, 5, 5, 4, 4, 4, 3, 3, 3, 3, 2}, // Low
    { 4, 12, 14, 13, 10, 7, 6, 5, 5, 4, 4, 4, 3, 3, 3, 3, 2}, // Middle
    { 4, 12, 14, 13, 10, 7, 6, 5, 5, 4, 4, 4, 3, 3, 3, 3, 2}} // High
},
```

Light Source Estimation P0



Feature name: Property of P0 stability

Variable name: i4Enable
Data range: 0 ~ 1
0 means disable P0 stabilized function
1 means enable P0 stabilized function

Variable name: i4PbRatio
Data range: 0 ~ 100

Variable name: i4Range
Data range: 0 ~ 32

Variable name: i4Clip
Data range: 0 ~ 10

Light Source Estimation P1

Feature name: Light source probability (P1) look-up table according to current LV for each light source
Data range: 0 ~ 100 // 數值越高, P1越高

```
// AWB Light source probability (P1) look-up table (Max: 100; Min: 0)
{
   //LV0   1    2    3    4    5    6    7    8    9    10   11   12   13   14   15   16   17   18
    {100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100}, // Strobe
    {100,  90,  85,  90, 100, 100,  85,  65,  36,  15,  15,  15,   8,   8,   1,   1,   1,   1}, // Tungsten
    {100,  90,  85,  85,  80,  80,  82,  72,  80,  55,  35,  15,   8,   8,   1,   1,   1,   1}, // Warm fluorescent
    {100, 100, 100, 100, 100, 100, 100, 100,  75,  80,  80,  80,  75,  55,  45,  10,   1,   1}, // Fluorescent
    {100, 100,  95, 100, 100, 100,  90, 100,  95,  90,  95,  80,  65,  43,  25,  30,   1,   1}, // CWF
    {100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100}, // Daylight
    {100, 100, 100, 100, 100, 100,  90, 100, 100, 100, 100, 100,  75,  75,  65,  45,  33,   1}, // Shade
    {100, 100, 100, 100, 100, 100, 100,  75,  55,  65,  80,  75,  50,  57,  33,   1,   1,   1}  // Daylight fluorescent
},
```

Light Source Estimation P2

Feature name: Light source probability (P2) look-up table according to green or magenta offset of corresponding light source white point for tungsten, warm fluorescent and shade

Data range: 0 ~ 256 (256 is 100 percent)

```
// AWB light source weight LUT
{
    //Tungsten
    {
        { // LUT: use magenta offset (0~1000) as index to get tungsten weight (x/256)
          // 0  100  200  300  400  500  600  700  800  900 1000
            256, 256, 256, 256, 256, 256, 256, 160,  96,  80,  32
        },
        { // LUT: use green offset (0~1000) as index to get tungsten weight (x/256)
          // 0  100  200  300  400  500  600  700  800  900 1000
            256, 256, 256, 256, 256, 256, 256, 128,  64,  32,  16
        },
    },
    //Warm Fluorescent
    {
        { // LUT: use magenta offset (0~2000) as index to get Warm Fluorescent weight (x//256)
          //  0  200  400  600  800 1000 1200 1400 1600 1800 2000
            256, 256, 256, 256, 256, 256, 256, 192,  96,  40,  16
        },
        { // LUT: use green offset (0~2000) as index to get Warm Fluorescent weight (x//256)
          //  0  200  400  600  800 1000 1200 1400 1600 1800 2000
            256, 256, 192,  80,  64,  64,  32,  16,  16,  16,  16
        }
    },
    //Shade
    {
        { // LUT: use magenta offset (0~1000) as index to get Shade weight (x/256)
          // 0  100  200  300  400  500  600  700  800  900 1000
            256, 256, 144,  80,  48,  32,  16,  16,  16,  16,  16
        },
        { // LUT: use green offset (0~1000) as index to get Shade weight (x/256)
          // 0  100  200  300  400  500  600  700  800  900 1000
            256, 256, 192, 112,  32,  32,  16,  16,  16,  16,  16
        },
    },
},
```

Magenta offset vs. weight

Green offset vs. weight

Light Source Estimation P2

```
// P2 Stability
        {
            1,  //i4Enable
           20,  //i4PbRatio
           20,  //i4LvThr
           80,  //i4P0Thr1
          110   //i4P0Thr2
        },
```

Feature name: Property of P2 stability

Variable name: i4Enable
Data range: 0 ~ 1
0 means disable P2 stabilized function
1 means enable P2 stabilized function

Variable name: i4PbRatio
Data range: 0 ~ 100
The P2 is effective when valid neutral block less than this value

Variable name: i4LvThr
Data range: 0 ~ 180
The P2 is effective when LV is less than this value

Variable name: i4P0Thr1
Data range: 0 ~ 150
Variable name: i4P0Thr2
Data range: 0 ~ 150
The P2 is fully effective when P0 > i4P0Thr1.
The P2 is not effective if P0 > i4P0Thr2
The P2 is partial effective when i4P0Thr2 > P0 > i4P0Thr1.

Parameters

- **Light Source Estimation**
  1. Child Block -> Parent Block Light Estimation
  2. Light Source Estimation
     P0, P1, P2

- **Predict AWB Gain by Light Source Info.**
  1. Preference Color Compensation
  2. Statistic Color Constraint
  3. Spatial Predictor
  4. Sunset compensation
  5. Daylight Locus Probability
  6. Noise Reduction
  7. Sub-window
  8. Preference Gain
  9. Face Assisted AWB

- **Spatial & Temporal Predictor**
  1. Spatial Predictor
  2. Temporal Predictor

- **Get AWB Output Gain**
  1. Reliable mode
  2. Extra Color
  3. Face Comp. AWB

Preference Color Compensation

Feature name: Offset threshold of Preference color.
Do full compensation when current distance from the origin to the white point(log(R/G), log(B/G)) is under Offset threshold , and do partial compensation when the distance is above Offset threshold
Data range: 0 ~ 10000

```
//Offset threshold
{
        4468, // Tungsten
        4468, // WF
        909   // Shade
},
```

Feature name: LV threshold for daylight locus partial compensation new offset calculation.
Daylight locus target offset = interpolate(current LV, i4ThrL, i4ThrH, Daylight locus temp offset*(ratio look-up in i4LUT for daylight locus offset partial compensation), Daylight locus temp offset)
Data range: 0 ~ 10000

```
//Offset ratio LV
{       {35,  70}, // Tungsten
        {50, 100}, // WF
        {35,  75}  // Shade
},
```

Preference Color Compensation

Feature name: Use daylight locus new offset (0~10000) as index to get daylight locus target offset ratio (0~100)
Data range: 0 ~ 100

```
//AWB daylight locus target Offset ratio LUT
{
    //Tungsten
    {
        {// LUT: use daylight locus new offset (0~10000) as index to get daylight locus target offset ratio (0~100)
            // 0  500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000 7500 8000 8500 9000 9500 10000
             100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  100
        },
    },
    //Warm Flurorescent
    {
        {// LUT: use daylight locus new offset (0~10000) as index to get daylight locus target offset ratio (0~100)
            // 0  500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000 7500 8000 8500 9000 9500 10000
             100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  100
        },
    },
    //Shade
    {
        {// LUT: use daylight locus new offset (0~10000) as index to get daylight locus target offset ratio (0~100)
            // 0  500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000 7500 8000 8500 9000 9500 10000
              50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50,  50
        },
    },
},
```

Preference Color Compensation

Feature name: New target offset remapping LUT table.

It is used to adjust white balance partial compensation degree when the white point distance in log(R/G)-log(B/G) domain is larger than threshold (i4OffsetThr) for tungsten, warm fluorescent and shade

```
//Daylight locus offset LUTs
{
    //Tungsten
    {
        {0,  500,  1000,  1500,  2000,  2500,  3000,  3500,  4000,  4500,  5000,  5500,  6000,  6500,  7000,  7500,  8000,  8500,  9000,  9500,  10000},
        {0,  250,   500,   750,  1000,  1250,  1300,  1350,  2000,  2700,  2778,  3200,  3222,  3444,  3667,  3667,  3889,  4333,  4556,  4778,  5000} // i4LUTOut
    },
    //WF
    {
        {0,  500,  1000,  1500,  2000,  2500,  3000,  3500,  4000,  4500,  5000,  5500,  6000,  6500,  7000,  7500,  8000,  8500,  9000,  9500,  10000},
        {0,  400,  900,   1000,  1250,  1200,  1500,  1950,  1900,  2400,  3000,  3000,  3200,  3467,  3889,  4111,  4333,  4556,  4778,   5000} // i4LUTOut
    },
    //Shade
    {
        {0,  500,  1000,  1500,  2000,  2500,  3000,  3500,  4000,  4500,  5000,  5500,  6000,  6500,  7000,  7500,  8000,  8500,  9000,  9500,  10000},
        {0,  300,   500,  1900,  2200,  2500,  3000,  4000,  4500,  4750,  5000,  5500,  6000,  6500,  7000,  7500,  8000,  8500,  9000,  9500,  10000} // i4LUTOut
    },
},
```

Preference Color Compensation

Feature name: Green and Magenta offset remapping look-up table for partial compensation offset for tungsten, warm fluorescent and shade

```
//AWB Green/Magenta offset threshold
{
    //Tungsten
    {
        1, //0: disabe 1: enable offset when GM offset is large than threshold
        { // LUT: use daylight locus offset (0~10000) as index to get magenta offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
           400, 400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,  400,   400 },
        { // LUT: use daylight locus offset (0~10000) as index to get green offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
          2000,2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000,  2000 },
    },
    //Warm Fluorescent
    {
        1, //0: disabe 1: enable offset when GM offset is large than threshold
        { // LUT: use daylight locus offset (0~10000) as index to get magenta offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
          1500,1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500,  1500 },
        { // LUT: use daylight locus offset (0~10000) as index to get green offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
          1500,1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1500,  1500 },
    },
    //Shade
    {
        1, //0: disabe 1: enable offset when GM offset is large than threshold
        { // LUT: use daylight locus offset (0~10000) as index to get magenta offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
           100, 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,   100 },
        { // LUT: use daylight locus offset (0~10000) as index to get green offset threshold
          // 0  500  1000  1500  2000  2500  3000  3500  4000  4500  5000  5500  6000  6500  7000  7500  8000  8500  9000  9500  10000
           100, 100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,  100,   100 },
    },
},
```

AWB v5.0

Preference Color Compensation

Feature name: Daylight locus offset gain and Green/Magenta offset gain to control preference compensation extra preference gain

```
// Offset Gain
{
    //Tungsten
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512},
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Warm Fluorescent
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Shade
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
},
// Magenta offset Gain
{
    //Tungsten
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512},
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Warm Fluorescent
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Shade
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    }
},
// Green offset Gain
{
    //Tungsten
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512},
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Warm Fluorescent
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    //Shade
    {
        {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    }
}
```

AWB v5.0

Statistic Color Constraint

- Property of statistic gain constraint for each light source

```
// rStatLimit
{
    1,     //i4Mode
    { 10, 30}, //LV
    // i4LimitY[AWB_LIGHT_NUM]
    {
          0,   //Strobe
         50,   //T
        180,   //WF
         40,   //F
        138,   //CWF
         45,   //Daylight
         40,   //Shade
        115,   //DF
    },
    // i4WeightReduce[AWB_LIGHT_NUM]
    {
        0,   //Strobe
        8,   //Tungsten
        8,   //WF
        8,   //F
        8,   //CWF
        8,   //Daylight
        8,   //Shade
        8,   //DF
    },
    // i4ProjWeight[AWB_LIGHT_NUM]
    {
       16,   //Strobe
       16,   //Tungsten
       16,   //WF
       16,   //F
       16,   //CWF
       16,   //Daylight
       16,   //Shade
       16,   //DF
    },
},
```

Variable name: i4Mode
Mode1 use each light source average location to projection.
Mode2 use each PB average location to projection.

Variable name: LV
Low: no effect when LV lower than low threshold
High: reduce effect from high threshold

Variable name: i4LimitY
These value define the reasonable distance from daylight locus. The values of Tungsten, Fluorescent, Daylight and shade are the upper boundary. The values of Warm Fluorescent, CWF and DF are the lower boundary

Variable name: i4WeightReduce
Data range: 0 ~ 16
If the statistic location is far away from daylight locus.
It will reduce the statistic gain weighting

Variable name: i4ProjWeight
Data range: 0 ~ 16
Mode mode 2, if the PB location is outside boundary, it can project to boundary by ratio defined in i4ProjWeight

Spatial Predictor

- Modes of spatial gain for each light source, non-reliable and temporal initial mode

```
// Spatial gain
{
    2,    //i4GeneralMode
    2,    //i4NonReliableMode
    2,    //i4TempInitMode
   40,    //i4ThrTemp

   // i4TempLv[AWB_LIGHT_NUM]
   {
     40, //Strobe
     40, //T
     40, //WF
     40, //F
     40, //CWF
     40, //D
     40, //Shade
     40, //DF
   }
},
```

Variable name: i4GeneralMode
Data range: 0 ~ 2
Variable name: i4NonReliableMode
Data range: 0 ~ 2
Variable name: i4TempInitMode
Data range: 0 ~ 2

Variable name: i4ThrTemp
Data range: 0 ~ 180
This value is used when non-reliable and temporal initial mode using mode 2

Variable name: i4TempLv
Data range: 0 ~ 180
These value are used in normal mode when using mode 2

Spatial Predictor

- Predefined white balance gain for different LV condition

```
// Predictor gain
{
    // rSpatial_L
    {
       1034, // i4R
        512, // i4G
        706  // i4B
    },
    // rSpatial_H
    {
        852, // i4R
        512, // i4G
        863  // i4B
    },
    // rTemporal_General
    {
        855, // i4R
        512, // i4G
        910  // i4B
    }
},
```

rSpatial_L gain is used for low LV condition

rSpatial_H gain is used for high LV condition

rTemporal gain is used for extra low LV or non-reliable condition

Spatial Predictor

- A set of LV threshold for spatial gain determination for each illuminant light source including no neutral light

```
// AWB LV threshold for predictor
{
        { 115, 155},   // NonReliable
    {
        {  80, 135},   // Strobe
        { 110, 160},   // Tungsten
        { 110, 160},   // WF
        { 130, 160},   // Fluorescent
        { 114, 146},   // CWF
        {  80, 135},   // Daylight
        {  80, 135},   // Shade
        {  80, 135},   // DF
    },
},
```

LV low threshold and high threshold for spatial gain determination in no reliable light source environment

LV low threshold and high threshold for spatial gain determination for each light source

Sunset Compensation

- Property description for sunset feature and daylight locus estimation for daylight and also parameters used to define possible shade area (sunset area) inside daylight window

```
// Sunset Prop
{
      1, // i4Enable
    120, // i4LVThr_L
    130, // i4LVThr_H
     10, // i4SunsetCountThr
      0, // i4SunsetCountRatio_L
    192, // i4SunsetCountRatio_H
     {
         151, // i4Sunset_BoundXr_Thr
        -381, // i4Sunset_BoundYr_Thr
     },
},
```

LV low threshold and high threshold for sunset target Gain weighting

Variable name: i4SunsetCountThr
Data range: 0~432
Sunset feature exists when parent block number > i4SunsetCountThr

Variable name: i4SunsetCountRatio_L & i4SunsetCountRatio_H
Data range: 0~256 (unit: 256, 256 means 100 percent)
Sunset count low threshold and high threshold for sunset target gain weighting

Right & upper bound of sunset shade area

Daylight Locus Probability

Feature name: Look-up table according to current LV for each light source to determine the percentage of statistic gain usage

Data range: 0 ~ 100

e.g. EQ gain of A = (i4LUT[A][LV] * statistics gain of A + (100 - i4LUT[A][LV]) * spatial gain) / 100

```
// AWB daylight locus probability (Statistic Weighting) look-up table (Max: 100; Min: 0)
{
    //LV0   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    { 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  50,  25,   0,   0,   0,   0}, // Strobe
    { 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  95,  75,  75,  75,  75,  75,  75,  75,   0}, // Tungsten
    { 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  95,  75,  75,  75,  75,  75,  75,  75,  75}, // Warm fluorescent
    { 100, 100,  90,  90, 100, 100, 100,  95,  95, 100,  95,  80,  80,  80,  55,  35,  10,   0,   0}, // Fluorescent
    { 100, 100,  85,  90,  90,  90, 100, 100, 100,  95,  85,  80,  46,  37,  30,  30,   0,   0,   0}, // CWF
    { 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  95,  80,  85,  85,  70,  70,  70,  50,  50}, // Daylight
    { 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  75,  50,  50,  70,  48,  25,   0,   0}, // Shade
    {  90,  90,  90,  90,  90,  90,  90,  90,  90,  95,  80,  80,  70,  82,  57,  30,   0,   0,   0}  // Daylight fluorescent
},
```

Noise Reduction

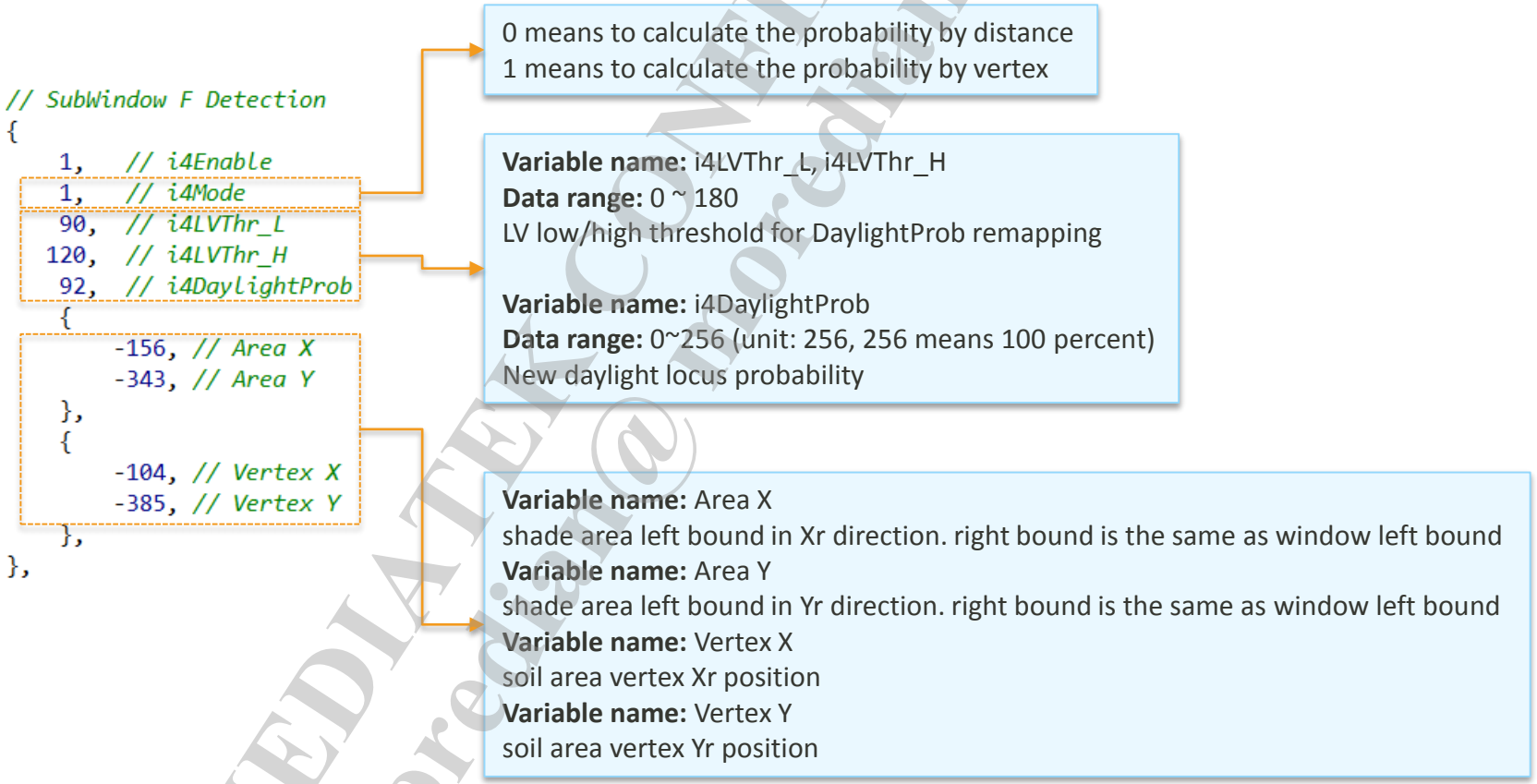
Feature name: Parent block threshold for noise reduction
Data range: 0 ~ 100 for each element in the table

These look-up table defines threshold ratio based on LV for all light source. When the neutral block of these light source is below threshold, it is not accumulated for statistics
e.g. Neutral block number[CWF] >= Parent block number * i4NRThr[CWF][LV]

```
// Noise Reduction
{
    //LV 0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, //Strobe
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, //Tungsten
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, //WF
    {   0,  0,  0,  0,  0,  3,  5,  5,  5,  5,  5,  10, 10, 10, 10, 10, 10, 10, 10}, //Fluorescent
    {   0,  0,  0,  0,  0,  3,  5,  5,  5,  5,  5,  10, 10, 10, 10, 10, 10, 10, 10}, //CWF
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  2,  2,  2,  2,  2,  2,  2,  2}, //Daylight
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, //Shade
    {   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  5,  10, 10, 10, 10, 10, 10, 10, 10}, //DF
},
```

Sub-window

- Property description for CWF / F feature and the daylight locus probability estimation for CWF / F parameters used to define possible shade area (soil area) inside CWF / F window



Preference Gain

Feature name: A set of gain applied after WB algorithm calculation output gain based on light source and LV.
It is used to optimize and fine tune for preference

Variable name: {i4R, i4G, i4B} gain table from LV0 to LV18 for each light source
(Strobe, Tungsten, Warm Fluorescent, Fluorescent, CWF, Daylight, Daylight Fluorescent, Shade)

```
// Preference gain for each light source
{
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // STROBE
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {532, 512, 522}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // TUNGSTEN
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {512, 512, 512}, {516, 512, 516}, {546, 512, 528}, {568, 512, 536}, {548, 512, 520}, {524, 512, 518}, {534, 512, 524}, {534, 512, 514}, {535, 512, 516}, {522, 512, 504},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}, {512, 512, 512}
        }, // WARM F
        {
            //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
            {520, 512, 540}, {518, 512, 520}, {514, 512, 502}, {526, 512, 522}, {522, 512, 502}, {512, 512, 502}, {512, 512, 502}, {512, 512, 512}, {512, 512, 514}, {512, 512, 512},
            //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
            {512, 512, 512}, {512, 512, 512}, {508, 512, 512}, {502, 512, 518}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}, {502, 512, 512}
        }, // F
```

AWB v5.0

Face AWB Common Setting

```
1,      //i4StatAvoidFaceArea
100,    //i4FaceWinRatio  50~400
1,      //u4FaceCentralWeight  0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
// rStatisticNR
{
    1,      //i4Enable
    200     //i4DiffThr
},
// rFD_RGB_Bound
{
      1,    //u4LowBound
    254     //u4HiBound
},
```

Variable name: i4StatAvoidFaceArea
Data range: 0 and 1
0 means normal AWB statistic
1 means normal AWB statistic will exclude face area

Variable name: i4FaceWinRatio
This parameter can change the original FD window size
Unit: percentage
Data range: 50 ~ 400

Variable name: u4FaceCentralWeight
This parameter can enhance FD window central weighting, the formula is (Original FD info + Central FD info * weighting) / (1+ weighting)
Data range: 0 ~

Variable name: rStatisticNR
0 means disable statistic NR
1 means enable statistic NR
Data range: 0 and 1

Variable name: i4DiffThr
Unit: N/A
Data range: 0~
This parameter is used to judge the child block belongs to face or not

Variable name: rFD_RGB_Bound
Data range: 0 ~ 255
If face R or G or B is lower than Low Bound, or higher than High bound, this face will be ignored for using

AWB v5.0

Face AWB Common Setting

- Face AWB temporal smooth mechanism
  - For Face-Comp. AWB & Face-Assisted AWB

```
// rTempoSmooth
{
    10,    //i4Speed: Converge Speed 1~100
    2,     //i4MinStep: Minimum converge step
    1      //i4ProbReduceStep: If face gone, the face prob reduce step 1~100
},
```

Variable name: rTempoSmooth

i4Speed

Control the temporal IIR step. If it sets to 10, the temporal smooth will separate the target distance into 10 steps and forward 1 step

Unit: N/A

Data range: 0 ~ 100

i4MinStep

Minimum AWB gain step. If one step, which is calculated by i4Speed, is smaller than i4MinStep, then AWB gain will choose to forward 1 i4MinStep

Unit: percentage

Data range: 0 ~

i4ProbReduceStep

Used for face gone. If face gone or disappear, the face probability will start to reduce by i4ProbReduceStep

Unit: percentage

Data range: 0 ~

AWB
v5.0

Face AWB Common Setting

- Face AWB temporal smooth mechanism
  - For Face-Comp. AWB & Face-Assisted AWB

```
// rSceneChange
{
    30,   //i4LVChangeTh
    20000    //i4AWBGainChangeTh
},
```

Variable name: rSceneChange used when face gone.

i4LVChange_Th

Used when face gone or disappear, and the LV changes more than this threshold in 1 frame, then directly make face comp. awb probability to 0

Unit: 10 = 1x

Data range: 0 ~ 180

i4AWBGainChange_Th

Used when face gone or disappear, and the AWB gain changes more than this threshold in 1 frame, then directly make face comp. awb probability to 0

Unit: 10 = N/A

Data range: 0 ~

AWB changes distance = (previous normal AWB R gain - current AWB R gain)^2 + (previous normal AWB B gain - current AWB B gain)^2

AWB v5.0

Face Assisted AWB

```
1,    //i4Enable
{ //i4TOL
      32,    //STB
      32,    //T
      32,    //WF
      32,    //F
      32,    //CWF
      32,    //D
      32,    //SHADE
      32,    //DF
},
//rRefTarget - Reference Target Setting
{
    1,    //i4Mode
  100,    //i4WeightCoef_a
  100,    //i4WeightCoef_b
   // i4RefTargetThr
   {
      32,    //ThrLow
      64,    //ThrMid
      96,    //ThrHigh
   },
   // i4RefTargetThr
   {
      // Ref_Low       Ref_Mid       Ref_High
      {{4095, 4095}, {4095, 4095}, {4095, 4095}}, // STROBE
      {{-552, -346}, {-552, -346}, {-552, -346}}, // TUNGSTEN
      {{-389, -352}, {-389, -352}, {-389, -352}}, // WARM FLUORESCENT
      {{-173, -390}, {-173, -390}, {-173, -390}}, // FLUORESCENT
      {{-141, -504}, {-141, -504}, {-141, -504}}, // CWF
      {{ -38, -399}, { -38, -399}, { -38, -399}}, // DAYLIGHT
      {{  97, -356}, {  97, -356}, {  97, -356}}, // SHADE
      {{  95, -468}, {  95, -468}, {  95, -468}}, // DF
   },
},
```

Variable name: i4Enable
Enable flag for face assisted AWB, 1: Enable, 0: Disable

Variable name: i4TOL[light]
Tolerance range for each light source.
Unit: coordinate in XY domain
Data range: 0 ~

Variable name: i4Mode
Weight function mode selection for reference target calculation.
0: Only reference those light source which located on daylight locus
1: Reference all light source (default)

Variable name: i4WeightCoef_a / i4WeightCoef_b
Weight coefficients for weight function of Mode-1.
WeightFunc(x) = coef_a * (x^2) + coef_b * x, where x is reciprocal of distance.
Data range: 0 ~

Variable name: i4RefTargetThr[3]
Threshold for reference target selection, indicated by Face-G value.
Unit: Pixel value in 8 bits resolution
Data range: 0 ~ 255

Variable name: rRefTargetXY
For every light source, there are three reference target coordinates can be set.
Coordinate domain is (Xr, Yr).
Unit: coordinate in XY domain
Data range: n/a

AWB v5.0

Face Assisted AWB

```
//rPrefGain for Predictor Gain
{
    { 512,  512}, // STROBE
    { 512,  512}, // TUNGSTEN
    { 512,  512}, // WARM FLUORESCENT
    { 512,  512}, // FLUORESCENT
    { 512,  512}, // CWF
    { 512,  512}, // DAYLIGHT
    { 512,  512}, // SHADE
    { 512,  512}, // DF
},
```

Variable name: rPrefGain[i]
{i4R, i4B} gain table from LV0 to LV18 for each light sources, these pref. gain will blending by weighting value and applied on prediction gain of face assisted AWB.
Unit: 1x = 512
Data range: 1 ~8191

```
//rProb0 - Face Size
{
    3200,  // i4FaceSizeRatioLow
    6000,  // i4FaceSizeRatioHi
       0,  // i4ProbMin
     100,  // i4ProbMax
},
```

Variable name: rProb0
Face size ratio for probability calculation of face assisted AWB.
Unit: Size Ratio = percentage*1000, Prob0 = percentage
Data range: size ratio 0 ~ 100000, Prob0 0 ~ 100

Variable name: rProb1
Light source probability LUT according to current LV for each light source
Unit: Percentage
Data range: 0 ~ 100

```
//rProb1 - LV
{{
    //LV 0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,   1}, // STROBE
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // TUNGSTEN
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // WARM FLUORESCENT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // FLUORESCENT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // CWF
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // DAYLIGHT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // SHADE
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40}, // DF
}},
```

AWB v5.0

Face Assisted AWB

```
//rProb2
{
    { //High-CT
      //   0    25   50   75  100  125  150  175  200  225  250
        { 256, 256, 256, 256, 256, 256, 256, 256, 240, 160, 128 }, // Magenta
        { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,  80 }, // Green
    },
    { //Mid-CT
      //   0    25   50   75  100  125  150  175  200  225  250
        { 256, 256, 256, 256, 256, 256, 256, 256, 240, 160, 128 }, // Magenta
        { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,  80 }, // Green
    },
    { //Low-CT
      //   0    25   50   75  100  125  150  175  200  225  250
        { 256, 256, 256, 256, 240, 212, 180, 128,  96,  64,  40 }, // Magenta
        { 256, 256, 256, 256, 256, 256, 240, 212, 180, 128,  80 }, // Green
    }
},
```

Variable name: rProb2
Probability for invalid face color prevention according to the magenta/green shift level (Y direction of FaceShift).
Currently we categorized all light sources into 3 group (High/Middle/Low color temperature).
If FaceShift-Y is negative, use “Green” LUT.
Otherwise, use “Magenta” LUT.
Unit: percentage*256
Data range: 0 ~ 256

AWB v5.0

Face Assisted AWB

```
//rProb3
{
      5,  // i4StableConfLow
     60,  // i4StableConfHi
      0,  // i4ProbMin
    100,  // i4ProbMax
},
//rProb4
{
    200,  // i4FaceColorRatioLow
    800,  // i4FaceColorRatioHi
      0,  // i4ProbMin
    100,  // i4ProbMax
},
```

Variable name: rProb3
Probability for face consistence. When face existed continuous frames, higher P3 strength will be used.
Unit: StableConf = frame count, Prob. = percentage * 100
Data range: StableConf - 0 ~ , Prob - 0 ~ 100

Variable name: rProb4
Face color ratio for probability estimation if statistic NR is enable to avoid face color unstable.
Unit: face color ratio = percentage * 1000 , Prob. = percentage * 100
Data range: face color - 0 ~ 1000, Prob - 0 ~ 100

AWB v5.0

Face Assisted AWB

```
//rCompSetting
{
    //rConfThr
    {
          0,  // i4ThrL
        100,  // i4ThrH
    },
    //rDistanceThr
    {
         50,  // i4ThrL
        200,  // i4ThrH
    },
    //rCompRatio
    {
          0,  // i4ThrL
         10,  // i4ThrH
    },
},
```

Feature Name: rCompSetting
Compensation setting for face assisted AWB.

Variable name: rConfThrLow (Low/High)
Confidence(P0*P1*p2*P3*P4) threshold for calculating P3 strength.
Unit: StableConf = frame count, Prob. = percentage * 100
Data range: StableConf - 0 ~ , Prob - 0 ~ 100

Variable name: rDistanceThr(Low/High)
High/Low distance threshold for compensation ratio.
Unit: distance in XY domain
Data range: 0 ~

Variable name: rCompRatio(Low/High)
Min/Max compensation ratio of face assisted AWB.
Unit: percentage * 100
Data range: 0 ~ 100

```
// rStableSetting
{
    7,  // i4TempoWeight
    8,  // i4DelayFrm
},
```

Feature Name: rStableSetting
Stable control setting for face assisted AWB

Variable name: i4TempoWeight
Weighted factor for face XY temporal smooth. FaceXY of current frame always blending with calculated result of previous frame.
Unit: weighting value
Data range: 0 ~

Variable name: i4DelayFrm
Predefined delay frame number for stability if face is unstable.
Unit: frame number
Data range: 0 ~

Parameters

- **Light Source Estimation**
  1. Child Block -> Parent Block Light Estimation
  2. Light Source Estimation
     P0, P1, P2

- **Predict AWB Gain by Light Source Info.**
  1. Preference Color Compensation
  2. Statistic Color Constraint
  3. Spatial Predictor
  4. Sunset compensation
  5. Daylight Locus Probability
  6. Noise Reduction
  7. Sub-window
  8. Preference Gain
  9. Face Assisted AWB

- **Spatial & Temporal Predictor**
  1. Spatial Predictor
  2. Temporal Predictor

- **Get AWB Output Gain**
  1. Reliable mode
  2. Extra Color
  3. Face Comp. AWB

Temporal Predictor

Feature name: Neutral parent block number threshold (ratio) for CWF/DF temporal enqueuer

Variable name: i4Neutral_ParentBlk_Thr
Data range: 0 ~ 432
One condition for current frame be enqueued to temporal when neutral parent block number > i4Neutral_ParentBlk_Thr

Variable name: i4CWFDF_LUTThr
Look-up table of neutral block threshold ratio based on LV for CWF and DF light source

```
// AWB number threshold for temporal predictor
{
    65,  // i4Neutral_ParentBlk_Thr
    //LV 0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {  100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  50,  25,   2,   2,   2,   2,   2,   2,   2}  // i4CWFDF_LUTThr
},
```

Feature name: non-neutral probability for spatial and temporal weighting look-up table
Variable name: i4LUT
Data range: 0 ~ 100
The look-up table defines the weighting to use temporal gain (High weighting means use higher temporal gain)

```
// AWB non-neutral probability for spatial and temporal weighting look-up table (Max: 100; Min: 0)
{
   //LV0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
     {  0,  33,  66,  100,  100,  100,  100,  100,  100,  100,  70,  30,  20,  10,   0,   0,   0,   0}
},
```

Parameters

- **Light Source Estimation**
  1. Child Block -> Parent Block Light Estimation
  2. Light Source Estimation
     P0, P1, P2

- **Predict AWB Gain by Light Source Info.**
  1. Preference Color Compensation
  2. Statistic Color Constraint
  3. Spatial Predictor
  4. Sunset compensation
  5. Daylight Locus Probability
  6. Noise Reduction
  7. Sub-window
  8. Preference Gain
  9. Face Assisted AWB

- **Spatial & Temporal Predictor**
  1. Spatial Predictor
  2. Temporal Predictor

- **Get AWB Output Gain**
  1. Reliable mode
  2. Extra Color
  3. Face Comp. AWB

Reliable Mode

Feature name: Parent block threshold for reliable mode
Data range: 0 ~ 100

These look-up table defines threshold ratio based on LV for no neutral light.
When the total Neutral Parent Block Number < i4NonNeutral * 432/100, AWB gain will blended with EQV Gain in non-reliable mode

```
// Reliable parent block threshold
{
    //LV 0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {    6,   6,   6,   6,   6,   6,   6,   6,   6,   6,   6,  10,  10,  10,  10,  10,  10,  10,  10},
    // Noise Reduction
    {
        //LV 0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0},  //Strobe
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0},  //Tungsten
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0},  //WF
        {    0,   0,   0,   0,   0,   3,   5,   5,   5,   5,   5,  10,  10,  10,  10,  10,  10,  10,  10},  //Fluorescent
        {    0,   0,   0,   0,   0,   3,   5,   5,   5,   5,   5,  10,  10,  10,  10,  10,  10,  10,  10},  //CWF
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   2,   2,   2,   2,   2,   2,   2,   2},  //Daylight
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0},  //Shade
        {    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   5,  10,  10,  10,  10,  10,  10,  10,  10},  //DF
    },
},
```

AWB v5.0

Extra Color

- Property description for extra color and compensated with 3 different modes

```
// Extra Color 0 yellow
{
    1, // i4Enable
    { 192, 128, 128}, // i4ModeWeight
    16, // i4ConfThr
    {0, 1, 1, 0, 0, 0, 0, 0}, // i4SelLightSrc : Strobe T WF F CWF D Shade DF
    20, // i4LvRange
    // Extra Color AWB gain
    {
        1124, // GainR
         512, // GainG
         808, // GainB
    },
    // Extra Color area
    {
        -350, // i4RightBound
        -410, // i4LeftBound
        -550, // i4UpperBound
        -600, // i4LowerBound
    },
    {  40,  70}, // rGlevel
    {  20,  80}, // rLv
    {  40, 200}, // rCount
    {   0,  35}, // rWeighting
},
```

Variable name: i4ModeWeight
Mode1, 2 and 3 weighting

Variable name: i4ConfThr
Temporal smooth threshold, when extra color existed continuous frames the weighting is maximum

Variable name: i4SelLightSrc
Mode2 and 3 effect on which light source

Variable name: i4LvRange
A value to control weighting curve by LV

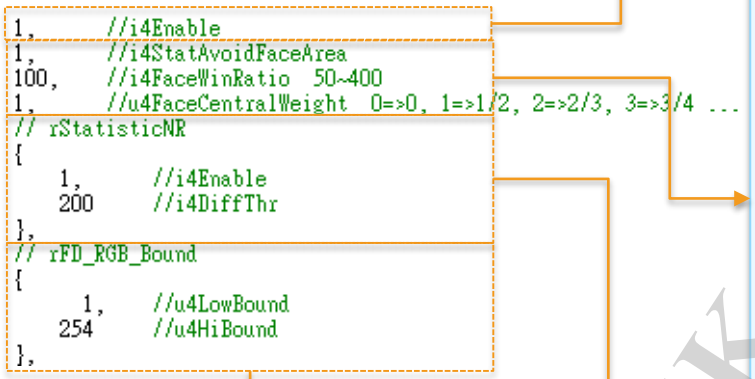
Predefined blending AWB gain

Extra color window in Xr Yr domain

Variable name: rGlevel
Data range: 0 ~ 255
Only the CB average G value inside this range will be taken into consideration
Variable name: rLv
Data range: 0 ~ 180
Blending weight for different LV
Variable name: rCount
Data range: 0 ~ 432
Start to compensate when detected count > rCount.i4ThrL
Fully compensated when detected count > rCount.i4ThrH
Variable name: rWeighting
Data range: 0 ~ 100
Blending weight for rCount, rLV, continuous frame

Face Comp. AWB

- Face Comp. AWB basic control & basic statistic tuning



0 means disable face comp. AWB
1 means enable face comp. AWB

Variable name: i4StatAvoidFaceArea
Data range: 0 and 1
0 means normal AWB statistic
1 means normal AWB statistic will exclude face area

Variable name: i4FaceWinRatio
Unit: percentage
Data range: 50 ~ 400
This parameter can change the original FD window size

Variable name: u4FaceCentralWeight
Data range: 0 ~
This parameter can enhance FD window central weighting, the formula is (Original FD info + Central FD info * weighting) / (1+ weighting)

Variable name: rStatisticNR
Data range: 0 and 1
0 means disable statistic NR
1 means enable statistic NR

Variable name: i4DiffThr
Unit: N/A
Data range: 0~
This parameter is used to judge the child block belongs to face or not

Variable name: rFD_RGB_Bound
Data range: 0 ~ 255
If face R or G or B is lower than Low Bound, or higher than High bound, this face will be ignored for using

Face Comp. AWB

- Face Comp. AWB preference skin tone target

```
1,    // i4SatTargetEn
// rRefTarget
{
    //i4Hue
    {
        19000,    // i4HiCT target hue, Unit = 1000
        17000,    // iMidCT target hue, Unit = 1000
        23000     // i4LowCT target hue, Unit = 1000
    },
    //i4Sat
    {
        40000,    // i4Sat target saturation, Unit = 1000
        35000,    // i4Sat target saturation, Unit = 1000
        50000     // i4Sat target saturation, Unit = 1000
    },
    7     // i4TempoWeight for Temporal Target weighting: 0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
},
// rSceneJudge
{
    90,    //i4LVLow for Indoor & Outdoor dynamic
    110,    //i4LVHi for Indoor & Outdoor dynamic
    2300,    //i4FaceRB_Low for Indoor Mid CT & Low CT dynamic, Unit = 1000
    3000     //i4FaceRB_Hi for Indoor Mid CT & Low CT dynamic, Unit = 1000
},
```

Variable name: i4SatTargetEn
Data range: 0 and 1
0 means not to use sat target, only use hue target
1 means use both hue & sat target

Variable name: rRefTarget
Default value: calibrated from calibration library
Unit: 1000 = 1x
Data range: Hue: 0 ~ 60000 Sat: 0 ~ 100000
This parameter is the preference skin tone of face {H, S}, which is after applying AWB gain.

Variable name: rSceneJudre
(a)i4LV_LowThr & i4LV_HiThr are LV threshold used for Mid & Hi CT ref target interpolation
(when LV is higher, the hi CT probability is higher)
Unit: 10 = 1x
(b)i4FaceRB_LowThr & i4FaceRB_HiThr are face processed raw R/B threshold,
which are used for Mid & Low CT ref target interpolation
(when face R/B is higher, the low CT probability is higher)
Unit: 1000 = 1x
If these parameters are set to 0, algo will use normal AWB light source prob for scene judgment.

Face Comp. AWB

- Get Face Comp. AWB Gain Ratio

```
// rConvergeCtrl
{
    1,    // i4TargetConvergeCtrlEn  If or:
    1000,    // i4HueTOL target hue converg
    5000,    // i4SatTOL target Saturation
    500     // i4RestrictRatioTOL restrict
},
// rOversat
{
    // rLV
    {
        90,    //i4LVLow
        120,    //i4LVHi
        0,    //i4OverSatProb_Low
        100     //i4OverSatProb_Hi
    },
    // rHiCT
    {
        40,    //i4CT_P_Low
        90,    //i4CT_P_Hi
        0,    //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rMidCT
    {
        40,    //i4CT_P_Low
        90,    //i4CT_P_Hi
        0,    //i4OverSatProb_Low
        75     //i4OverSatProb_Hi
    },
    // rLowCT
    {
        0,    //i4CT_P_Low
        0,    //i4CT_P_Hi
        0,    //i4OverSatProb_Low
        0     //i4OverSatProb_Hi
    }
},
```

rConvergeCtrl

Variable name: i4TargetConvergeCtrlEn
Data range: 0 and 1
0 means not to use target converge ctrl mechanism
1 means enable target converge ctrl mechanism

Variable name: i4HueTOL, i4SatTOL
Unit: 1000 = 1x
The hue & sat target tolerance

Variable name: i4RestricRatioTOL
Unit: 1000 = 1x
When enable target converge ctrl, the restrict ratio dynamic changing tolerance in hue or sat domain

rLV

Use LV to interpolate from i4LV_Low to i4LV_Hi to get Probability from i4Prob_Low to i4Prob_Hi
Unit: LV 10 = 1x, Prob = percentage
Data range: LV = 0 ~ 180, Prob = 0 ~ 100

rHiCT rMidCT rLowCT

Use Hi/Mid/Low Prob to interpolate from i4CT_P_Low to i4CT_P_Hi to get oversat probability from i4OverSatProb_Low to i4OverSatProb_Hi
Unit: percentage
Data range: CT_P = 0 ~ 100, Prob = 0 ~ 100

Face Comp. AWB

```
// rUndersat
{
    10000,    // i4SatHi
     5000,    // i4SatLow
       75,    // i4NewRestrict
       25,      // i4LVLow
       95,      // i4LVHi
       10         // i4LVTOL
},
```

rUndersat

Use face sat to interpolate from i4SatHi to i4SatLow to get undersat strength

Unit: LV 10 = 1x, Prob = percentage

Data range: LV = 0 ~ 180, Prob = 0 ~ 100

Variable name: i4HueTOL, i4SatTOL

Unit: 1000 = 1x

The hue & sat target tolerance

Variable name: i4RestricRatioTOL

Unit: 1000 = 1x

When enable target converge ctrl, the tolerance in hue or sat domain for restrict ratio dynamic changing

Variable name: i4GainRatioRestrictLUT

Unit: 1000 = 1x (50 = 5%)

Data range: 0 ~ 1000

This LV LUT will restrict face comp. AWB gain can only change the normal AWB gain in specific percentage range.

Face Comp. AWB gain = Original AWB gain * Restrict Gain Ratio

```
// i4GainRatioRestrictLUT
//LV0  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
{  0, 10, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 10,  0,  0},
```

Variable name: {i4R, i4G, i4B} gain table from LV0 to LV18.

Unit: 512 = 1x

Data range: 0 ~ 4096

A set of gain applied after face comp. AWB algorithm calculation output gain based on LV, just like normal AWB preference gain LUT.

```
// rPrefGain
{
  //    LV0             LV1             LV2             LV3             LV4             LV5             LV6             LV7             LV8             LV9
  {512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},
  //    LV10            LV11            LV12            LV13            LV14            LV15            LV16            LV17            LV18
  {512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},{512, 512, 512},
},
```

Face Comp. AWB

- Get Face Probability (P0 & P1)

```
// rProb0
{
    3900,    //i4FaceSizeRatioLow
    6100,    //i4FaceSizeRatioHi
    0,       //i4Prob0_Min
    100      //i4Prob0_Max
},
```

rProb0
Referring face size compare to 120x90. It uses face size ratio to interpolate from i4FaceSizeRatio_Low to i4FaceSizeRatio_Hi to get Probability0 from i4Prob0_Min to i4Prob0_Max.
Unit: size ratio = percentage*1000, Prob0 = percentage
Data range: size ratio 0 ~ 100000, Prob0 0 ~ 100

```
// rProb1
{
    //LV0   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }, //i4HiCT_LUT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }, //i4MidCT_LUT
    {  50,  75, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  80,  60,  40  }  //i4LowCT_LUT
},
```

rProb1 is a LV LUT for face probability1
Unit: percentage
Data range: 0 ~ 100

Face Comp. AWB

- Get Face Probability (P2 & P3)

```
// rProb2
{
    400,    //i4FaceRG_Low Unit=1000
   1200,    //i4FaceRG_Hi Unit=1000
    500,    //i4FaceRG_TOL Unit=1000
      0,    //i4Prob2_Min
    100     //i4Prob2_Max
},
// rProb3
{
    250,    //i4FaceBG_Low Unit=1000
    800,    //i4FaceBG_Hi Unit=1000
    500,    //i4FaceBG_TOL Unit=1000
      0,    //i4Prob3_Min
    100     //i4Prob3_Max
},
// rProb4
{
      0,    //i4FaceG_Low Unit=1000
      0,    //i4FaceG_Hi Unit=1000
      0,    //i4FaceG_TOL Unit=1000
    100,    //i4Prob4_Min
    100     //i4Prob4_Max
},
```

rFaceAWBProb2

It is referring to face processed raw R/G information. If the R/G is out of range, the face prob2 must be reduced. It interpolate from

i4FaceAreaRG_LowTh to (i4FaceAreaRG_LowTh -i4FaceAreaRG_Tolerance) or
i4FaceAreaRG_HiTh to (i4FaceAreaRG_HiTh + i4FaceAreaRG_Tolerance)

to get probability2 from i4Prob2_Max to i4Prob2_Min

Unit: R/G 1000 = 1x, Prob = percentage
Data range: R/G N/A, Prob 0 ~ 100

rFaceAWBProb3

Same with rFaceAWBProb2, just change to refer face processed raw B/G information.

rFaceAWBProb4

Same with rFaceAWBProb2, just change to refer face processed raw G information.

Face Comp. AWB

- Face Spatial Compensation when face size is small and neutral parent block number is very few

```
// rSpatial
{
    // rP0
    {
        3900,    //i4FaceSizeRatioHi  Unit=1000
        500,     //i4FaceSizeRatioLow Unit=1000
        0,     //i4Prob0_Min
        50      //i4Prob0_Max
    },
    // P1
    {
        75,     //i4NeutralParentBlkNum_Hi
        15,     //i4NeutralParentBlkNum_Low
        0,     //i4Prob1_Min
        50      //i4Prob1_Max
    },
    70,     //i4LVLow
    110      //i4LVLHi
},
```

rSpatial_P0

Referring face size compare to 120x90. It uses face size ratio to interpolate from i4FaceSizeRatio_Hi to i4FaceSizeRatio_Low to get Probability0 from i4Prob0_Min to i4Prob0_Max.

Unit: size ratio = percentage*1000, Prob0 = percentage

Data range: size ratio 0 ~ 100000, Prob0 0 ~ 100

rSpatial_P1

It uses neutral parent block number to interpolate from i4NeutralParentBlkNum_Hi to i4NeutralParentBlkNum_Low to get Probability1 from i4Prob1_Min to i4Prob1_Max.

Unit: neutral parent block num = N/A, Prob1 = percentage

Data range: neutral parent block num 0 ~ 120*90, Prob1 0 ~ 100

i4LV_Low & i4LV_LHi

Used for AWB spatial gain calculation.

It is interpolated from i4LV_Low to i4LV_LHi to get face comp. awb spatial gain from DaylightLocus_L to DaylightLocus_H.

Unit: 10 = 1x

Data range: 0 ~ 180

Debug Parser Tag

AWB debug parser tag (1/21)

Tag	Description
AWB_TAG_MODE	AWB or PWB
AWB_TAG_ALGO_SCENE_LV	Ambient brightness
AWB_TAG_AAO_MODE	AAO mode data format 0 : 12 bit , 1 : 14 bit
AWB_TAG_SCENARIO	AWB Scenario 0 : Preview, 1 : Video, 2 : Capture
AWB_TAG_GAIN	AWB Final Gain (already merged into Calibration Gain)
AWB_TAG_ALG_GAIN	Final Gain of AWB algorithm (not yet aligned with D65 Gain)
AWB_TAG_AWB_GAIN_NO_PREF	AWB Gain without preference features (Preference Gain, Extra Color, Face AWB, etc.)
AWB_TAG_RELIABLE_MODE	Source of final AWB Gain 0: Use completely the mixture of Statistic and Spatial Predictor 1: Mixture of 0 and 2 2: Use completely the mixture of Spatial Predictor and Temporal Predictor
AWB_TAG_TEMPO_BUFF_IDX	Frame number of Temporal Predictor buffer
AWB_TAG_NONEUTRAL_PROB	Percentage of Temporal Predictor in mixture of Spatial Predictor and Temporal Predictor
AWB_TAG_NONEUTRAL_SPAT_GAIN	AWB Final Spatial Predictor Gain
AWB_TAG_NONEUTRAL_TEMPO_GAIN	AWB Final Temporal Predictor Gain
AWB_TAG_NONEUTRAL_EQV_GAIN	Gain of AWB Final Spatial Predictor mixing with Temporal Predictor

AWB debug parser tag (2/21)

Tag	Description
AWB_TAG_P0	Probability of white point under each light source in AWB Statistic
AWB_TAG_P1	Probability of brightness under each light source in AWB Statistic
AWB_TAG_P2	Probability of white point characteristic under each light source in AWB Statistic
AWB_TAG_P	Final probability under each light source in AWB Statistic
AWB_TAG_STA_GAIN	Statistic Gain under each light source
AWB_TAG_SPAT_GAIN	Spatial Predictor Gain under each light source
AWB_TAG_HIT_NR	Conditions for AWB NR
AWB_TAG_DAYLIGHT_PROB	Percentage of Statistic in mixture of Statistic and Spatial Predictor under each light source
AWB_TAG_EQV_DAYLIGHT_PROB	Final percentage of Statistic in mixture of Statistic and Spatial Predictor under each light source (considering Feature Detect, AWB NR)
AWB_TAG_EQV_GAIN	Final Gain of mixture of Statistic and Spatial Predictor under each light source
AWB_TAG_HIT_SUNSET	Conditions for Sunset Feature Detect
AWB_TAG_SHADE_XR_F/CWF	[Sub Window] XR coordinate of PB inside sub window (shade area)
AWB_TAG_SHADE_YR_F/CWF	[Sub Window] YR coordinate of PB inside sub window (shade area)
AWB_TAG_SHADE_COUNT_F/CWF	[Sub Window] PB number inside sub window (shade area)
AWB_TAG_SHADE_TARGET_PROB_F/CWF	[Sub Window] Daylight locus probability after Sub Window PB count mechanism

AWB debug parser tag (3/21)

Tag	Description
AWB_TAG_EXTRACOLOR_INFO_COUNT	[Extra Color] Parent block number inside Extra Color detection area
AWB_TAG_EXTRACOLOR_INFO_WEI_GAIN	[Extra Color] Weighting of Extra Color to mixed with pre-defined gain
AWB_TAG_EXTRACOLOR_INFO_WEI_P2	[Extra Color] Weighting of Extra Color to reduce P2
AWB_TAG_EXTRACOLOR_INFO_WEI_DL_PROB	[Extra Color] Weighting of Extra Color to reduce Daylight locus probability
AWB_TAG_EXTRACOLOR_INFO_CONF	[Extra Color] Temporal confidence of Extra color detection
AWB_TAG_EXTRACOLOR_INFO_GAIN_IN	[Extra Color] AWB gain before mixed with Extra Color pre-defined gain
AWB_TAG_EXTRACOLOR_INFO_GAIN_OUT	[Extra Color] AWB gain after mixed with Extra Color pre-defined gain

AWB debug parser tag (4/21)

Tag	Description
AWB_TAG_PB_NUM_THR_NONNEUTRAL	THR of total number of white points that satisfies AWB NR
AWB_TAG_PB_NUM_THR	THR of number of white points that satisfies AWB NR under each light source
AWB_TAG_NEUTRAL_PB_NUM	Number of white points under each light source
AWB_TAG_LIGHT_MODE	Light source which has white points in it (in binary)
AWB_TAG_STAT_ERR_COUNT	Number of child blocks with too many dark or bright pixels (Error blocks)
AWB_TAG_STAT_MO_COUNT	Number of child blocks determined as motion blocks
AWB_TAG_STAT_ERR_MO_COUNT	Number of child blocks determined as error and also motion blocks
AWB_TAG_CHILD_BLK_NUM_THR	Child block number threshold (T0) for determine light source for parent block
AWB_TAG_ONE_SHOT_SMOOTH_ENABLE	Whether to enable ONESHOT mechanism
AWB_TAG_ONE_SHOT_SMOOTH_LV_L	If ambient brightness is lower than this value, Capture Gain will use the result of Spatial Predictor.
AWB_TAG_ONE_SHOT_SMOOTH_LV_H	If ambient brightness is higher than this value, Capture Gain will not refer to Preview Gain.
AWB_TAG_SPAT_LV_THR_L	LV low threshold for spatial gain determination for each illuminant light source
AWB_TAG_SPAT_LV_THR_H	LV high threshold for spatial gain determination for each illuminant light source
AWB_TAG_PRE_COLOR_LV_THR_L	If S/T/WF light source and ambient brightness is smaller than this value, AWB algorithm will decide the white balance.
AWB_TAG_PRE_COLOR_LV_THR_H	If S/T/WF light source and ambient brightness is bigger than this value, algorithm will allow max. color cast.

AWB debug parser tag (5/21)

Tag	Description
AWB_TAG_STATS_R	Sum of R value of all parent blocks in each light source
AWB_TAG_STATS_G	Sum of G value of all parent blocks in each light source
AWB_TAG_STATS_B	Sum of B value of all parent blocks in each light source
AWB_TAG_AVG_XR	Coordinate of AWB Gain on XY Domain under each light source
AWB_TAG_AVG_YR	Coordinate of AWB Gain on XY Domain under each light source
AWB_TAG_LIMIT	Statistic Yr - i4LimitY
AWB_TAG_DAY_LOCUS_OFFSET_T/WF/S	S/T/WF light source in DAYLIGHT LOCUS algorithm result
AWB_TAG_NEW_OFFSET_T/WF/S	S/T/WF light source has color cast or not (0: no color cast)
AWB_TAG_OFFSET_RATIO_T/WF/S	S/T/WF light source in DAYLIGHT LOCUS algorithm result
AWB_TAG_RATIO_OFFSET_T/WF/S	S/T/WF light source in DAYLIGHT LOCUS algorithm result
AWB_TAG_LUT_OFFSET_T/WF/S	S/T/WF light source in DAYLIGHT LOCUS algorithm result Final result, higher means more color cast
AWB_TAG_IS_ABOVE_DAY_LOCUS_T/WF/S	1 : Green/Magenta offset is above Daylight Locus -> Magenta Offset 0 : Green/Magenta offset is above Daylight Locus -> Green Offset
AWB_TAG_GM_OFFSET_T/WF/S	Green/Magenta offset
AWB_TAG_GM_OFFSET_THR_T/WF/S	Green/Magenta offset threshold
AWB_TAG_WEIGHT	Only for Shade/WF/Tungsten. The bigger the weight, the bigger P2
AWB_TAG_DL_OFFSET_GAIN	Daylight locus offset corresponding gain
AWB_TAG_GM_OFFSET_GAIN	Green/Magenta offset corresponding gain

AWB debug parser tag (6/21)

Tag	Description
AWB_TAG_DL_OFFSET_PREF_GAIN	Preference gain for Daylight locus offset
AWB_TAG_GM_OFFSET_PREF_GAIN	Preference gain for Green/Magenta offset
AWB_TAG_PREF_GAIN	Preference gain combined with Daylight locus offset and Green/Magenta offset
PWB_TAG_NEUTRAL_AREA_PB_NUM	Number of white points of PWB in specific light source
PWB_TAG_REFERENCE_AREA_PB_NUM	Number of white points of PWB in reference light source
PWB_TAG_PB_NUM	Total number of PWB points
PWB_TAG_DEFAULT_GAIN	PWB Default Gain
PWB_TAG_GAIN_NEUTRAL_AREA	Gain got from PWB using white points under specific light source
PWB_TAG_GAIN_REFERENCE_AREA	Gain got from PWB using white points under reference light source
PWB_TAG_LIGHT_SOURCE	Specific light source of PWB
PWB_TAG_LIGHT_REFERENCE_AREA_XOR	Coordinate of gain got from PWB referring to light source on XY Domain
PWB_TAG_LIGHT_REFERENCE_AREA_YOR	Coordinate of gain got from PWB referring to light source on XY Domain
PWB_TAG_LIGHT_REFERENCE_AREA_XPR	Coordinate of gain got from PWB referring to light source on XY Domain
PWB_TAG_LIGHT_REFERENCE_AREA_YPR	Coordinate of gain got from PWB referring to light source on XY Domain
PWB_TAG_MWB_CCT	Color temperature calculated by MWB mechanism
PWB_TAG_MWB_CENTER_XR	Xr coordinate of MWB mechanism
PWB_TAG_MWB_CENTER_YR	Yr coordinate of MWB mechanism
PWB_TAG_MWB_RANGE_X	MWB window size in Xr Yr domain (i4WindowSizeX)
PWB_TAG_MWB_RANGE_Y	MWB window size in Xr Yr domain (i4WindowSizeY)

AWB debug parser tag (7/21)

Tag	Description
AWB_TAG_STAT_CONFIG_WINDOW_SIZE_X	HW statistic related data. Horizontal window size of each MainStat
AWB_TAG_STAT_CONFIG_WINDOW_SIZE_Y	HW statistic related data. Vertical window size of each MainStat
AWB_TAG_STAT_CONFIG_WINDOW_PITCH_X	HW statistic related data. Horizontal Pitch Window size of each MainStat
AWB_TAG_STAT_CONFIG_WINDOW_PITCH_Y	HW statistic related data. Vertical Pitch Window size of each MainStat
AWB_TAG_STAT_CONFIG_WINDOW_ORIGIN_X	HW statistic related data. Original X coordinate of HW statistic
AWB_TAG_STAT_CONFIG_WINDOW_ORIGIN_Y	HW statistic related data. Original Y coordinate of HW statistic
AWB_TAG_STAT_CONFIG_WINDOW_NUM_X	HW statistic related data. Number of MainStat window in X axis
AWB_TAG_STAT_CONFIG_WINDOW_NUM_Y	HW statistic related data. Number of MainStat window in Y axis
AWB_TAG_STAT_CONFIG_LOW_THR	HW statistic related data. Low threshold for error pixel count
AWB_TAG_STAT_CONFIG_HIGH_THR	HW statistic related data. High threshold for error pixel count
AWB_TAG_STAT_CONFIG_PIXEL_COUNT	HW statistic related data. Pixel count for calculate average RGB value
AWB_TAG_STAT_CONFIG_PREGAIN_LIMIT_R/G/B	HW statistic related data. Maximum PreGain Limit value
AWB_TAG_STAT_CONFIG_PREGAIN_R/G/B	HW statistic related data. Module Pregain1 (Unit gain/Golden gain)
AWB_TAG_STAT_ERROR_SHIFT_BIT	Error count shift bits. Programmable error count shift bits : 0 ~ 7
AWB_TAG_STAT_ERROR_THR	Error count threshold for the allowed total over-exposed and under-exposed pixels in one main stat window
AWB_TAG_STAT_MO_ERROR_THR	Mo & error count threshold for pixels in one main stat window
AWB_TAG_STAT_ROTATION_MATRIX_COS	Rotation matrix parameter. Value of cosine normalized by 256
AWB_TAG_STAT_ROTATION_MATRIX_SIN	Rotation matrix parameter. Value of sine normalized by 256

AWB debug parser tag (8/21)

Tag	Description
AWB_TAG_STAT_CONFIG_AWBXY_WIN	Window range of each AWB light source
AWB_TAG_LINEAR_AAO_ENABLE	Enable linear AAO statistics
AWB_TAG_LINEAR_AAO_ERRCNT_THR	Parent block error threshold, when error count is larger than threshold, using linear AAO instead
AWB_TAG_SMOOTHSTAT_ENABLE	Enable Smooth Statistics
AWB_TAG_SMOOTHSTAT_OFFSET	Shrink window offset (i4StatWinShrinkOffset) for Smooth Statistics
AWB_TAG_SMOOTHSTAT_DIST_THR	Distance threshold for Smooth Statistics
AWB_TAG_FACE_AWB_ENABLE	Enable Face Comp. AWB from NVRAM setting
AWB_TAG_FACE_AWB_EXCUTE	Make sure to execute face comp. AWB 0: Won't execute face comp. AWB, even NVRAM set enable 1: Will execute face comp. AWB
AWB_TAG_NO_STAT_FACE_AREA_ENABLE	0 : normal AWB statistic 1 : normal AWB statistic will exclude face area
AWB_TAG_FACE_NUM	Number of face in whole image
AWB_TAG_FACE_MGR_MAX_WIN_R/L/T/B	Maximum face window location in MGR domain (Debug & Simulation using)
AWB_TAG_FACE_WIN_RATIO	Face window ratio (i4FaceWinRatio)
AWB_TAG_FACE_MAX_WIN	Maximum face window location
AWB_TAG_FACE_MAX_SACLE_WIN	Maximum face window location which is after scaling by win ratio
AWB_TAG_FACE_IS_DETECT	1 : Face is detected
AWB_TAG_FACE_STAT_NR_ENABLE	Enable face statistic NR
AWB_TAG_FACE_STAT_NR_TH	Face statistic NR threshold

AWB debug parser tag (9/21)

Tag	Description
AWB_TAG_FACE_CENTRAL_WEIGHT	Central weighting of FD window (u4FaceCentralWeight)
AWB_TAG_FACE_TOTAL_SIZE_SUM	All face size summation
AWB_TAG_FACE_MAX_SIZE	The maximum face size
AWB_TAG_FACE_MAX_INDEX	The maximum face index
AWB_TAG_FACE_WEIGHTED_AVG_R/G/B	Face RAW Avg RGB, R/G, B/G, R/B
AWB_TAG_FACE_RAW_RG/BG/RB	Face RAW Avg R/G, B/G, R/B
AWB_TAG_FACE_ORIGINAL	Face RAW apply original AWB gain R/G, B/G
AWB_TAG_FACE_CURRENT	Face RAW after temporal smooth (the source of face comp. AWB in this frame)
AWB_TAG_FACE_CT_LV_LOW/HI	Scene Judge LV threshold
AWB_TAG_FACE_CT_FACE_RB_LOW/HI	Scene Judge face raw R/B threshold
AWB_TAG_FACE_HICT_PROB	Daylight + Shade + DF Prob from normal AWB
AWB_TAG_FACE_MIDCT_PROB	F + CWF Prob from normal AWB
AWB_TAG_FACE_LOWCT_PROB	T+ WF Prob from normal AWB
AWB_TAG_FACE_REFERENCE_TARGET_HUE	Reference face hue target
AWB_TAG_FACE_REFERENCE_TARGET_SAT	Reference face saturation target
AWB_TAG_FACE_SAT_TARGET_ENABLE	0: No using saturation target, only use hue target 1: using both hue & saturation target
AWB_TAG_FACE_TARGET_CONVERGE_CTRL_ENABLE	Enable target converge control mechanism
AWB_TAG_FACE_UNDERSAT_SAT_HI/LOW	Face saturation threshold for Undersat prevention mechanism
AWB_TAG_FACE_UNDERSAT_RESTRICT	New Restrict ratio after Undersat prevention

AWB debug parser tag (10/21)

Tag	Description
AWB_TAG_FACE_UNDERSAT_LV_LOW/HI	Undersat prevention works LV filter
AWB_TAG_FACE_UNDERSAT_LV_TOL	Undersat prevention works LV filter tolerance
AWB_TAG_FACE_RESTRICT_TOL	When enable target converge ctrl, the tolerance in hue or sat domain for restrict ratio dynamic changing
AWB_TAG_FACE_GAIN_RATIO_RESTRICT	Face Comp. AWB Gain restrict ratio set by NVRAM LUT
AWB_TAG_FACE_CURRENT_GAIN_RATIO_RESTRICT	Face Comp. AWB Gain restrict ratio after temporal smooth
AWB_TAG_FACE_ADDSAT_GAIN_RATIO_RB/BG	Adding saturation (yellowish) gain ratio
AWB_TAG_FACE_REDUCESAT_GAIN_RATIO_RB/BG	Reducing saturation (bluisg) gain ratio
AWB_TAG_FACE_OVERSAT_PROB_KEEPSAT	Keepsat prob
AWB_TAG_FACE_OVERSAT_PROB_SAT	Sat prob, which is auto generated by saturation control algorithm
AWB_TAG_FACE_OVERSAT_PROB_LV	Oversat prob generated by LV condition tuning manually
AWB_TAG_FACE_OVERSAT_PROB_HI/MID/LOW_CT	Oversat prob generated by CT condition tuning manually
AWB_TAG_FACE_OVERSAT_PROB	Combine all oversat prob
AWB_TAG_FACE_OVERSAT_CURRENT_PROB	Oversat prob after temporal smooth
AWB_TAG_FACE_GAIN_RATIO_RG/BG	Final gain ratio
AWB_TAG_FACE_PREFERENCE_GAIN_R/G/B	Face Comp. AWB preference gain
AWB_TAG_FACE_GAIN_R/G/B	Face Comp. AWB Gain before hybrid with Normal AWB gain & Spatial Gain
AWB_TAG_FACE_SIZE_RATIO	Face size ratio for reference to tune P0
AWB_TAG_FACE_PROB	Face Comp. AWB Probability status
AWB_TAG_FACE_CURRENT_PROB0	Face Comp. AWB P0 after temporal smooth
AWB_TAG_FACE_NEUTRAL_PB_NUM	Neutral Parent Block Number (for Face_Spatial_Prob1)

AWB debug parser tag (11/21)

Tag	Description
AWB_TAG_FACE_SPATIAL_PROB	Face_Spatial_Prob = Face_Spatial_Prob0 x Face_Spatial_Prob1
AWB_TAG_FACE_SPATIAL_GAIN	Face Spatial Gain
AWB_TAG_FACE_ORIGINAL_TARGET_GAIN_R/G/B	Original (Normal) AWB Gain
AWB_TAG_FACE_TARGET_GAIN_R/G/B	Face Comp. AWB Gain after hybrid with Normal AWB gain & Spatial Gain
AWB_TAG_FACE_FINAL_GAIN_R/G/B	Final face Comp. AWB Gain after face temporal smooth
AWB_TAG_FACE_FINAL_RG/BG	Final Face R/G & B/G after face comp. AWB
AWB_TAG_FACE_CURRENT_HUE	Face Hue, which is get from face RAW applies original AWB gain
AWB_TAG_FACE_TARGET_HUE	Face Hue, which is get from reference face target
AWB_TAG_FACE_TOLERANCE_HUE	Target hue tolerance
AWB_TAG_FACE_FINAL_HUE	Face Hue, which is get from face RAW applies final face comp. AWB gain
AWB_TAG_FACE_CURRENT_SAT	Face Sat, which is get from face RAW applies original AWB gain
AWB_TAG_FACE_TARGET_SAT	Face Sat, which is get from reference face target
AWB_TAG_FACE_TOLERANCE_SAT	Target sat tolerance
AWB_TAG_FACE_FINAL_SAT	Face Sat, which is get from face RAW applies final face comp. AWB gain
AWB_TAG_FACE_RSV_A/B/C	Reserved
AWB_TAG_FACE_RSV_H1/H2/H3	Reserved
AWB_TAG_FACE_RSV_H/CHCTRL	Reserved
AWB_TAG_FACE_RSV_N1/N2	Reserved
AWB_TAG_GENDER_ENABLE	Enable gender mechanism
AWB_TAG_GENDER_FEMALE_EXIST	Female exist in this frame
AWB_TAG_GENDER_PROB	Gender confidence

AWB debug parser tag (12/21)

Tag	Description
AWB_TAG_GENDER_TARGET_HUEOFS	Gender offset to refine target hue
AWB_TAG_GENDER_TARGET_SATOFS	Gender offset to refine target sat
AWB_TAG_GENDER_OVERSAT_PROB_RATIO	Gender ratio to refine oversat ratio
AWB_TAG_GENDER_FINAL_HUEOFS	Gender offset (modified by confidence) to refine target hue
AWB_TAG_GENDER_FINAL_SATOFS	Gender offset (modified by confidence) to refine target sat
AWB_TAG_GENDER_FINAL_OVERSAT_RATIO	Gender ratio (modified by confidence) to refine oversat ratio
AWB_TAG_PORTRAIT_MW_ENABLE	Portrait enable by MW
AWB_TAG_PORTRAIT_ENABLE	Portrait enable by NVRAM
AWB_TAG_PORTRAIT_IS_VALUE	No use
AWB_TAG_PORTRAIT_NO_VALUE	No use
AWB_TAG_PORTRAIT_VALUE	No use
AWB_TAG_PORTRAIT_EXIST	Portrait exist in this frame
AWB_TAG_PORTRAIT_PROB	Portrait confidence
AWB_TAG_PORTRAIT_FRAME_DELAY	Portrait frame delay
AWB_TAG_CCT	Color temperature of ambient light source
AWB_TAG_F_INDEX	Possibility of ambient light source being Fluorescent
AWB_TAG_DF_INDEX	Possibility of ambient light source being Daylight Fluorescent
AWB_TAG_CCT_X/Y	Not use
AWB_TAG_CCT_XR/YR	Xr Yr coordinate of final AWB color temperature decision
AWB_TAG_CCT_MIRED_H/L	Color Temperature Index_L & Index_H in Mired units
AWB_TAG_CCT_MIRED	Interpolated Color Temperature by MIRED_H & MIRED_L

AWB debug parser tag (13/21)

Tag	Description
AWB_TAG_CCT_YR_TL84/CWF/MEAN_F/D65/ DF/MEAN_DF	Yr coordinate of each light source for estimate color temperature MEAN_F:average Yr of CWF & F, MEAN_DF:average Yr of D65 & DF
AWB_TAG_CAP_SCENE_LV	Not use. Please refer to “AWB_TAG_ALGO_SCENE_LV”
AWB_TAG_PV_SCENE_LV	Not use. Please refer to “AWB_TAG_ALGO_SCENE_LV”
AWB_TAG_RAW_PREGAIN_NUM	Number of AWB calibration gain
AWB_TAG_RAW_PREGAIN1_R/G/B	MTK AWB Calibration Gain (AWB Pre-Gain 1)
AWB_TAG_RAW_PREGAIN1_R/G/B_M	MTK AWB Calibration Gain of middle color temperature
AWB_TAG_RAW_PREGAIN1_R/G/B_L	MTK AWB Calibration Gain of low color temperature
AWB_TAG_RAW_PREGAIN2_R/G/B	AWB Pre-Gain 2
AWB_TAG_LIGHT_STAT_CAL_GAIN	Golden D65 Gain
AWB_TAG_OUTPUT_CAL_GAIN	Unit D65 Gain
AWB_TAG_PREFGAIN	Preference gain used
AWB_TAG_STAT_LIMIT_ENABLE	1 : Statistic Gain Constraint is enabled
AWB_TAG_STAT_LIMIT_Y	i4LimitY
AWB_TAG_STAT_LIMIT_W_RED	i4WeightReduce
AWB_TAG_STAT_LIMIT_PROJ_W	Parent block limit Y projection weighting by light source
AWB_TAG_CB_MIN_THR	The minimum value for valid statistic block (i4CbMinThr)
AWB_TAG_SPAT_GENERAL_MODE	Spatial Predictor mode for General
AWB_TAG_NONRELIABLE_MODE	Spatial Predictor mode for Non-reliable
AWB_TAG_TEMP_INIT_MODE	Spatial Predictor mode for Initialization
AWB_TAG_LV_TEMP	LV_Temp threshold for Spatial Predictor using mode 2

AWB debug parser tag (14/21)

Tag	Description
AWB_TAG_REFINE_P0_ENABLE	1 : P0 stability function is enabled
AWB_TAG_REFINE_P0_PB_RATIO	Variable of P0 stability : i4PbRatio
AWB_TAG_REFINE_P0_RANGE	Variable of P0 stability : i4Range
AWB_TAG_REFINE_P0_CLIP	Variable of P0 stability : i4Clip
AWB_TAG_REFINE_P2_ENABLE	1 : P2 stability function is enabled
AWB_TAG_REFINE_P2_PB_RATIO	Variable of P2 stability : i4PbRatio
AWB_TAG_REFINE_P2_LV_THR	Variable of P2 stability : i4LvThr
AWB_TAG_REFINE_P2_P0_THR1/THR2	Variable of P2 stability : i4P0Thr1 & i4P0Thr2
AWB_TAG_OFFSET_SMOOTH_ENABLE	1 : Preference Color Stability is enabled
AWB_TAG_OFFSET_SMOOTH_THR_T/WF/S	Variable of : Preference Color Stability : rOffsetSmooth
AWB_TAG_EXTRACOLOR_ENABLE	1 : Extra Color Detection is enabled
AWB_TAG_EXTRACOLOR_MODE_WEI_GAIN/P2/DL_PROB	Extra Color applied weighting for mode pre-defined gain/reduce P2/reduce Daylight probability
AWB_TAG_EXTRACOLOR_CONFTHR	Extra Color temporal confidence threshold
AWB_TAG_EXTRACOLOR_SEL_LIGHT_SRC	Extra Color mode P2/DL_PROB applied light source
AWB_TAG_EXTRACOLOR_LV_RANGE	Extra Color weighting curve LV range to keep weighting high and reverse to weighting low when it is not zero
AWB_TAG_EXTRACOLOR_GAIN_R/G/B	Extra Color pre-defined AWB gain
AWB_TAG_EXTRACOLOR_AREA_U/D/L/R	Extra Color area window
AWB_TAG_EXTRACOLOR_GAVG_L/H	Extra Color green channel average low/high boundary

AWB debug parser tag (15/21)

Tag	Description
AWB_TAG_EXTRACOLOR_LV_L/H	Extra Color LV low/high boundary
AWB_TAG_EXTRACOLOR_COUNT_L/H	Extra Color parent block count L/H, L is minimum count to start extra color and H is the count to reach maximum weighting
AWB_TAG_EXTRACOLOR_WEI_L/H	Extra Color weighting minimum and maximum value
AWB_TAG_NONEBAYER_EANBLE	Non Bayer sensor flag
AWB_TAG_NONEBAYER_COF	Non Bayer sensor tag
AWB_TAG_NONEBAYER_COF_INV	Non Bayer sensor tag
AWB_TAG_AWB_GAIN_RWB_R/G/B	R/G/B Gain for RWB sensor after 3x3 matrix conversion
AWB_TAG_FULL_GAIN_R/G/B	AWB Full Gain. Mixed with Statistic Gain * Prob. of each light source only
AWB_TAG_RCCONV_SUPPORT	1 : HW support 3x3 matrix conversion
AWB_NVRAM_UNIT_GAIN	MTK AWB Calibration Gain – Unit
AWB_NVRAM_GOLDEN_GAIN	MTK AWB Calibration Gain – Golden
AWB_NVRAM_UNIT_GAIN_R/G/B_M	MTK AWB Calibration Gain – Unit (for middle color temperature)
AWB_NVRAM_GOLDEN_GAIN_R/G/B_M	MTK AWB Calibration Gain – Golden (for middle color temperature)
AWB_NVRAM_UNIT_GAIN_R/G/B_L	MTK AWB Calibration Gain – Unit (for low color temperature)
AWB_NVRAM_GOLDEN_GAIN_R/G/B_L	MTK AWB Calibration Gain – Golden (for low color temperature)
AWB_NVRAM_TUNING_UNIT_GAIN	MTK AWB Calibration Gain
AWB_NVRAM_D65_GAIN	MTK AWB Calibration Gain – D65
AWB_NVRAM_UNIT_VALUE_R/GB/GR/B	AWB unit module Bayer value for DNP(D50)
AWB_NVRAM_GOLDEN_VALUE_R/GB/GR/B	AWB golden module Bayer value for DNP(D50)

AWB debug parser tag (16/21)

Tag	Description
AWB_NVRAM_UNIT_VALUE_R/GB/GR/B_M	AWB unit module Bayer value for middle color temperature TL84
AWB_NVRAM_GOLDEN_VALUE_R/GB/GR/B_M	AWB golden module Bayer value for middle color temperature TL84
AWB_NVRAM_UNIT_VALUE_R/GB/GR/B_L	AWB unit module Bayer value for low color temperature A light
AWB_NVRAM_GOLDEN_VALUE_R/GB/GR/B_L	AWB golden module Bayer value for low color temperature A light
AWB_NVRAM_L_XO	Original X coordinate of Gain in each AWB light source
AWB_NVRAM_L_YO	Original Y coordinate of Gain in each AWB light source
AWB_NVRAM_L_XR	Rotate X coordinate of Gain in each AWB light source
AWB_NVRAM_L_YR	Rotate Y coordinate of Gain in each AWB light source
AWB_NVRAM_L_AWB_GAIN	Gain of each AWB light source
AWB_NVRAM_ROTATION_ANGLE	Rotation Matrix of AWB
AWB_NVRAM_ROTATION_COS	Rotation Matrix of AWB
AWB_NVRAM_ROTATION_SIN	Rotation Matrix of AWB
AWB_NVRAM_SLPOE_NUMERATOR	Slope of AWB Daylight Locus
AWB_NVRAM_SLPOE_DENOMINATOR	Slope of AWB Daylight Locus
AWB_NVRAM_PREDICTOR_PREF_RATIO	Currently not for use
AWB_NVRAM_PREDICTOR_L_GAIN	Low brightness gain of Spatial Predictor
AWB_NVRAM_PREDICTOR_H_GAIN	High brightness gain of Spatial Predictor
AWB_NVRAM_PREDICTOR_G_GAIN	Default gain of Temporal Predictor

AWB debug parser tag (17/21)

Tag	Description
AWB_NVRAM_L_RIGHT/LEFT/UPPER/LOWER	Window boundary of each AWB light source
PWB_NVRAM_L_RIGHT/LEFT/UPPER/LOWER	Window boundary of each PWB light source
PWB_NVRAM_DEF_GAIN_R/G/B	PWB default gain for each light source
AWB_NVRAM_PREF_COLOR_OFFSET_THR	Color cast control on S/T/WF light source
AWB_NVRAM_SUNSET_AREA_XR/YR_THR	Variable of Sunset feature : i4Sunset_BoundXr_Thr & i4Sunset_BoundYr_Thr
AWB_NVRAM_SHADE_L_METHOD	Variable of Sub Window detection : i4Method 1 : Calculate the probability by vertex 0 : Calculate the probability by distance
AWB_NVRAM_SHADE_L_AREA_XR/YR_THR	Variable of Sub Window detection : Area X & Area Y
AWB_NVRAM_SHADE_L_VERTEX_XR/YR_THR	Variable of Sub Window detection : Vertex X & Vertex Y
AWB_NVRAM_PB_WEIGHT_ENBALE	Whether weighting of highly bright area is bigger in Statistic
AWB_NVRAM_PB_WEIGHT_SCALE_FACTOR	Whether weighting of highly bright area is bigger in Statistic
AWB_NVRAM_PB_WEIGHT_LV_L/M/H	Parent block weighting LUT L/M/H LV threshold
AWB_NVRAM_PREDICTOR_INIT_LV_THR_L	In mixture of Temporal Predictor and Spatial Predictor, the corresponding LV value of Spatial Predictor low brightness gain
AWB_NVRAM_PREDICTOR_INIT_LV_THR_H	In mixture of Temporal Predictor and Spatial Predictor, the corresponding LV value of Spatial Predictor high brightness gain
AWB_NVRAM_TEMPORAL_ENQ_NEUTRAL_BLK_THR	Lower limit of total white points that can be included in Temporal Predictor calculation
AWB_NVRAM_TEMPORAL_ENQ_CWF_DF_BLK_THR	Upper limit of white points in CWF/DF light source that can be included in Temporal Predictor calculation

AWB debug parser tag (18/21)

Tag	Description
AWB_NVRAM_SUNSET_EN	1 : Sunset feature detection is enabled
AWB_NVRAM_SUNSET_LV_THR_L/H	Variable of Sunset feature : i4LVThr_L & i4LVThr_H
AWB_NVRAM_SUNSET_COUNT_THR	Variable of Sunset feature : i4SunsetCountThr
AWB_NVRAM_SUNSET_COUNT_RATIO_L/H	Variable of Sunset feature : i4SunsetCountRatio_L & i4SunsetCountRatio_H
AWB_NVRAM_SHADE_F/CWF_EN	1 : Sub Window detection is enabled
AWB_NVRAM_SHADE_F/CWF_LV_THR_L/H	Variable of Sub Window detection : i4LVThr_L & i4LVThr_H
AWB_NVRAM_SHADE_F/CWF_DAYLIGHT_PROB	Variable of Sub Window detection : i4DaylightProb
AWB_NVRAM_NONNEUTRAL_PROB_9/10/11/12	AWB non-neutral probability for spatial and temporal weighting look-up table (only saves LV9~LV12)
AWB_NVRAM_DAYLIGHT_LOCUS_9/10/11/12	AWB daylight locus probability look-up table (only saves LV9~LV12)
AWB_NVRAM_CCT_HORIZON~	Color temperature of each light sources
AWB_NVRAM_XR_HORIZON~	XY Domain coordinate of each light source
AWB_TAG_CCT_GAIN_SEL	
AWB_TAG_STROBE_PB_RED	

AWB v5.0

AWB debug parser tag (19/21)

Tag	Description
AWB_TAG_FACEAST_ENABLE	Enable Face Comp. AWB from NVRAM setting
AWB_TAG_FACEAST_SIZE_RATIO	Face size ratio for Prob0 calculation
AWB_TAG_FACEAST_FACECOLOR_RATIO	Face color ration for Prob4 calculation
AWB_TAG_FACEAST_ORIG_FACE_XR / YR	Convert face color to XY domain, this value is original statistic value without temporal smooth
AWB_TAG_FACEAST_FACE_XR / YR	Face color in XY domain but calculating with temporal smooth
AWB_TAG_FACEAST_REF_TARGET_XR / YR	Coordinates of face reference target in XY domain
AWB_TAG_FACEAST_DIST_WEI_MIN_DIST	Minimum distance value when weight function calculation
AWB_TAG_FACEAST_DIST_WEI_MIN_LIGHT	Indicate which light source with min. distance when calc. weight function
AWB_TAG_FACEAST_DIST_WEI_<light>	Weight value for each light source for reference target estimation
AWB_TAG_FACEAST_TARGET_TOL	Tolerance value that allow face unstable
AWB_TAG_FACEAST_PREDICT_XR / YR	Predicted neutral coordinates by face assisted AWB in XY domain
AWB_TAG_FACEAST_PREDICT_GAIN_R / B	Predicted gain by face assisted AWB. This gain is applied the preference gain which blending all light source pref. gain with weighting value.
AWB_TAG_FACEAST_PREDICT_PREFER_GAIN_R / B	Final preference gain which blends all prefer. gain of every light source with their weighting value.
AWB_TAG_FACEAST_FACE_SHIFT	Y-axis of Face-Shift vector.
AWB_TAG_FACEAST_PREDICT_LIGHT	The light source that predicted by face assisted AWB.

AWB v5.0

AWB debug parser tag (20/21)

Tag	Description
AWB_TAG_FACEAST_CONF_PROB0	Face assisted AWB probability 0 for face size ratio
AWB_TAG_FACEAST_CONF_PROB1_<light>	Face assisted AWB probability 1 for environment brightness for each light source
AWB_TAG_FACEAST_CONF_PROB2	Face assisted AWB probability 2 for confusing color (corresponding to face-shift)
AWB_TAG_FACEAST_CONF_PROB3	Face assisted AWB probability 2 for stability confidence
AWB_TAG_FACEAST_CONF_PROB4	Face assisted AWB probability 2 for face color ratio (if enable statistic-NR)
AWB_TAG_FACEAST_STRENGTH_<light>	P3 strength value for each light sources in face assisted AWB
AWB_TAG_FACEAST_LIGHT_DIS_<light>	The distance between NeturalXY of face assisted AWB to light source predicted XY of normal AWB.
AWB_TAG_FACEAST_P3_<light>	Final P3 probability for each light sources in face assisted AWB.
AWB_TAG_FACEAST_GAINPROB_<light>	The gain probability for face assisted predicted gain.
AWB_TAG_FINAL_GAINPROB_<light>	The gain probability for normal AWB prediction gain.
AWB_TAG_FACEAST_DAY_LOCUS_OFFSET	Improve precise gain control for face assisted AWB, the result of face-assisted-AWB also apply “preference color” feature if predicted light source is T/WF/S. The meaning of these value are same as above description for T/WF/S light source (Link).
AWB_TAG_FACEAST_NEW_OFFSET	
AWB_TAG_FACEAST_OFFSET_RATIO	
AWB_TAG_FACEAST_RATIO_OFFSET	
AWB_TAG_FACEAST_LUT_OFFSET	
AWB_TAG_FACEAST_IS_ABOVE_DAY_LOCUS	
AWB_TAG_FACEAST_GM_OFFSET	
AWB_TAG_FACEAST_GM_OFFSET_THR	
AWB_TAG_FACEAST_WEIGHT	

AWB v5.0

AWB debug parser tag (21/21)

Tag	Description
AWB_TAG_FACEAST_DL_OFFSET_GAIN_R / B	Improve precise gain control for face assisted AWB, the result of face-assisted-AWB also apply “preference color” feature if predicted light source is T/WF/S. The meaning of these value are same as above description for T/WF/S light source (Link).
AWB_TAG_FACEAST_GM_OFFSET_GAIN_R / B	
AWB_TAG_FACEAST_DL_OFFSET_PREF_GAIN_R / B	
AWB_TAG_FACEAST_GM_OFFSET_PREF_GAIN_R / B	
AWB_TAG_FACEAST_PREFER_GAIN_R / B	
AWB_TAG_FACEAST_TEMP_BUFF_IDX	Total frame number in temporal queue of face-assisted-AWB.
AWB_TAG_FACEAST_TEMP_AVG_FACE_X / Y	Average face XY coordinate by temporal queue.
AWB_TAG_FACEAST_TEMP_AVG_TOL	Average face tolerance value by temporal queue.

everyday genius