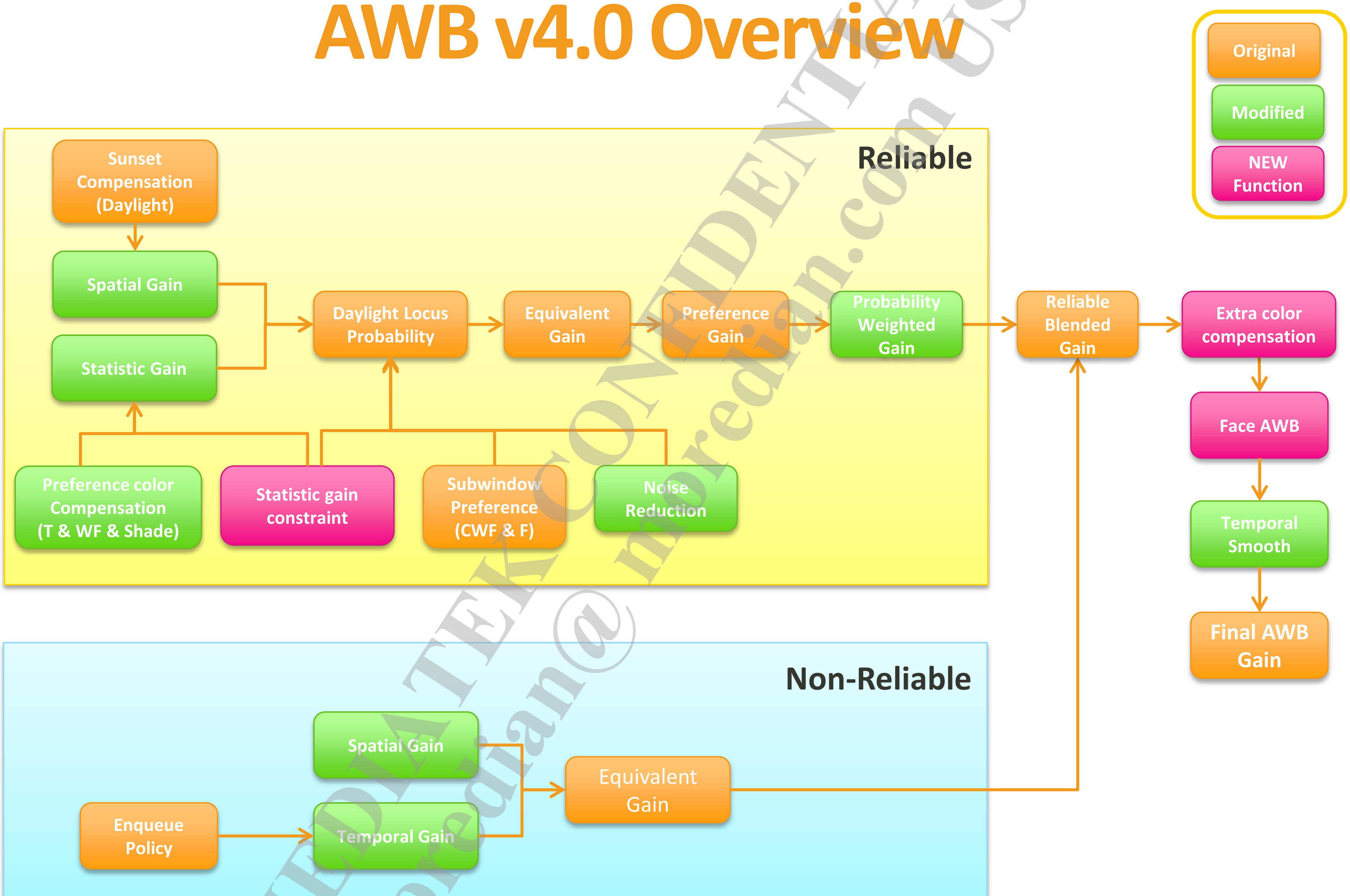
MT6771 AWB Introduction
MT6771 CCM Introduction
MT6771 ColorEngine Introduction

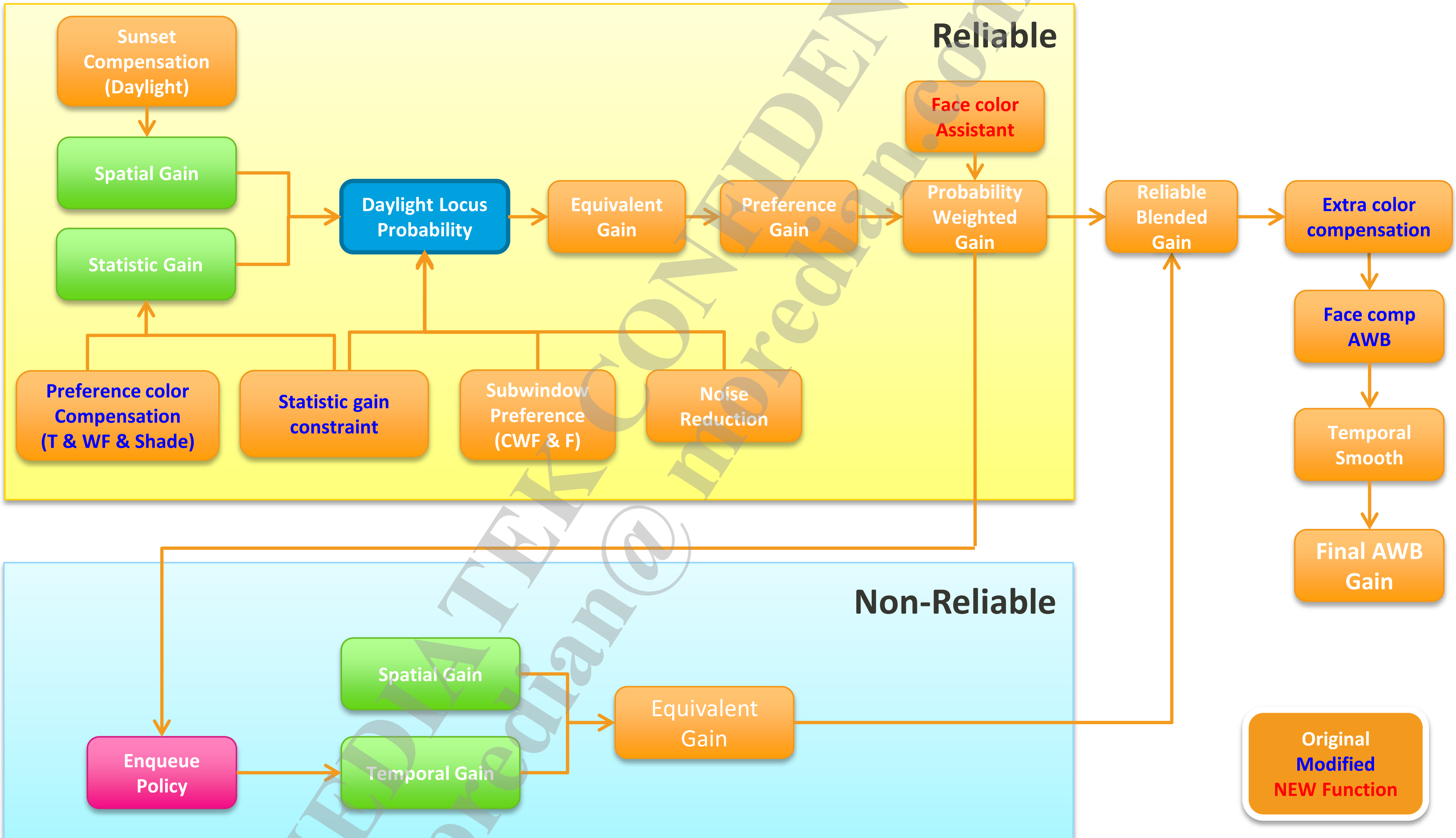
CONFIDENTAL B

MT6771
AWB Introduction

AWB v4.0 Overview



AWB v5.0 Overview



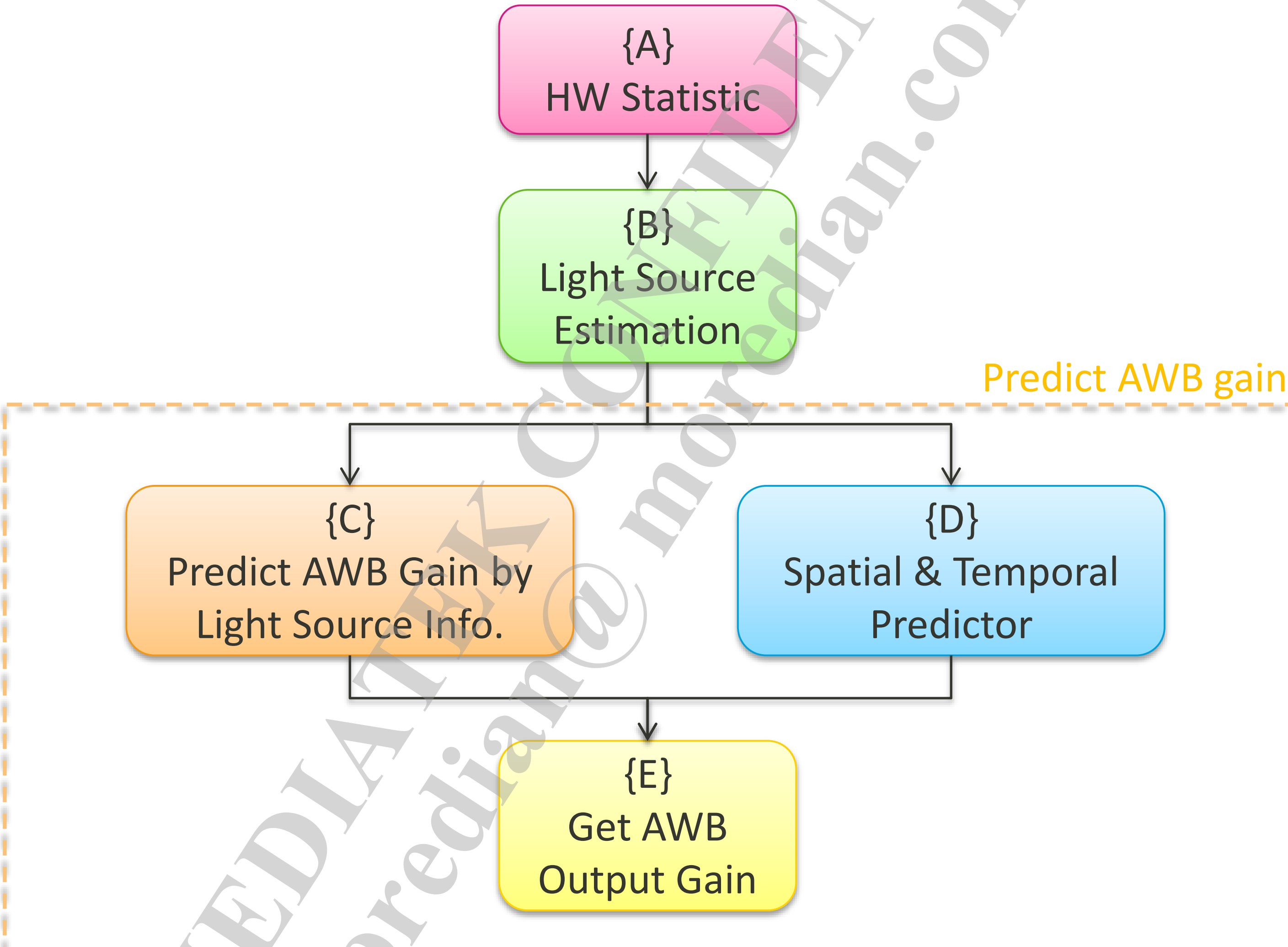
Outline

- 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

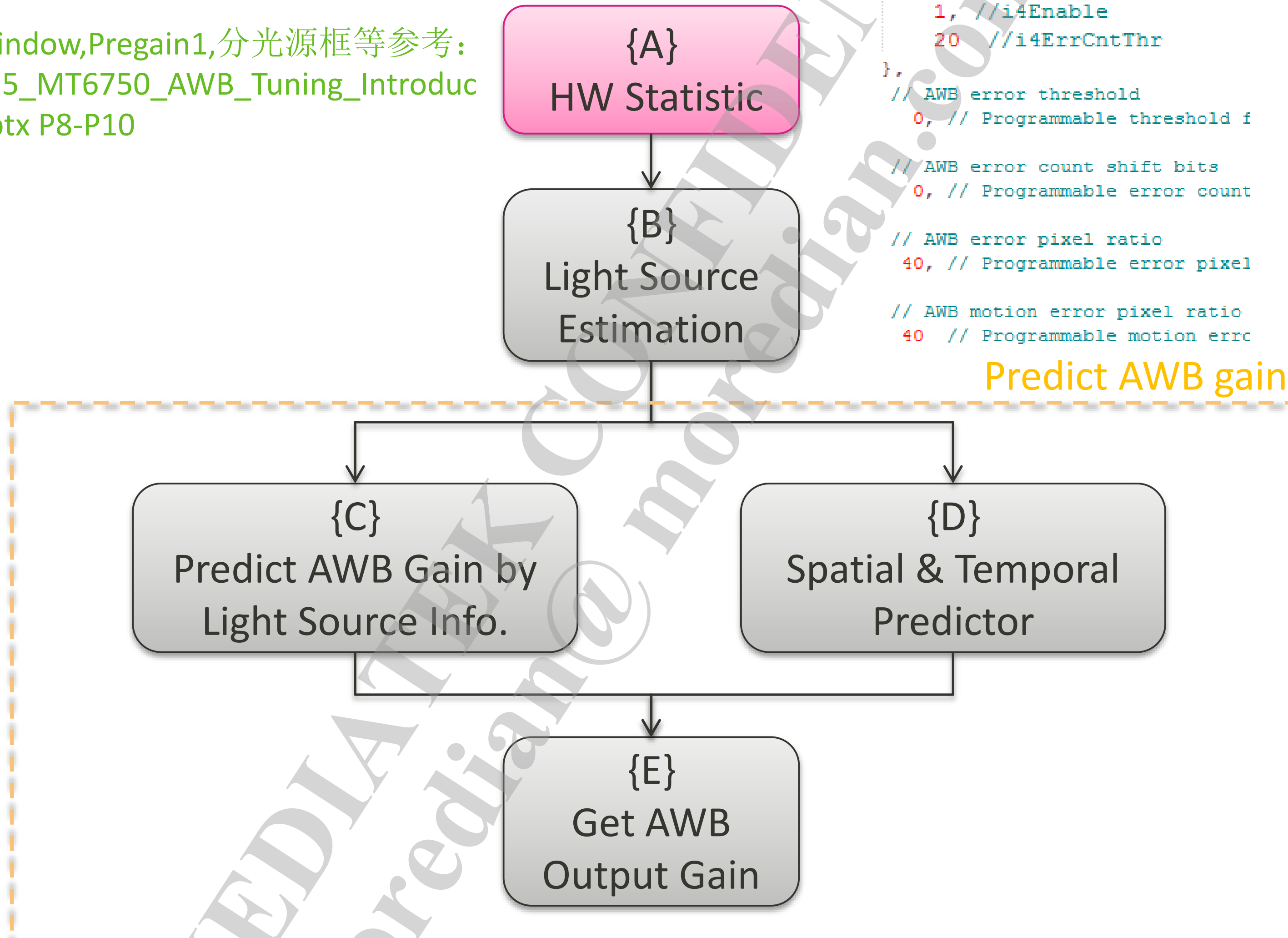
参考文档：MT6757_Camera_AWB_Tuning_Introduction.pdf中P5-P9
主要介绍awb的基本概念以及MTK的awb basic flow 和方法。

AWB Flow


{A}
HW Statistic
{B}
Light Source
Estimation
Predict AWB gain
{C}
Predict AWB Gain by
Light Source Info.
{D}
Spatial & Temporal
Predictor
{E}
Get AWB
Output Gain

{A} HW Statistic

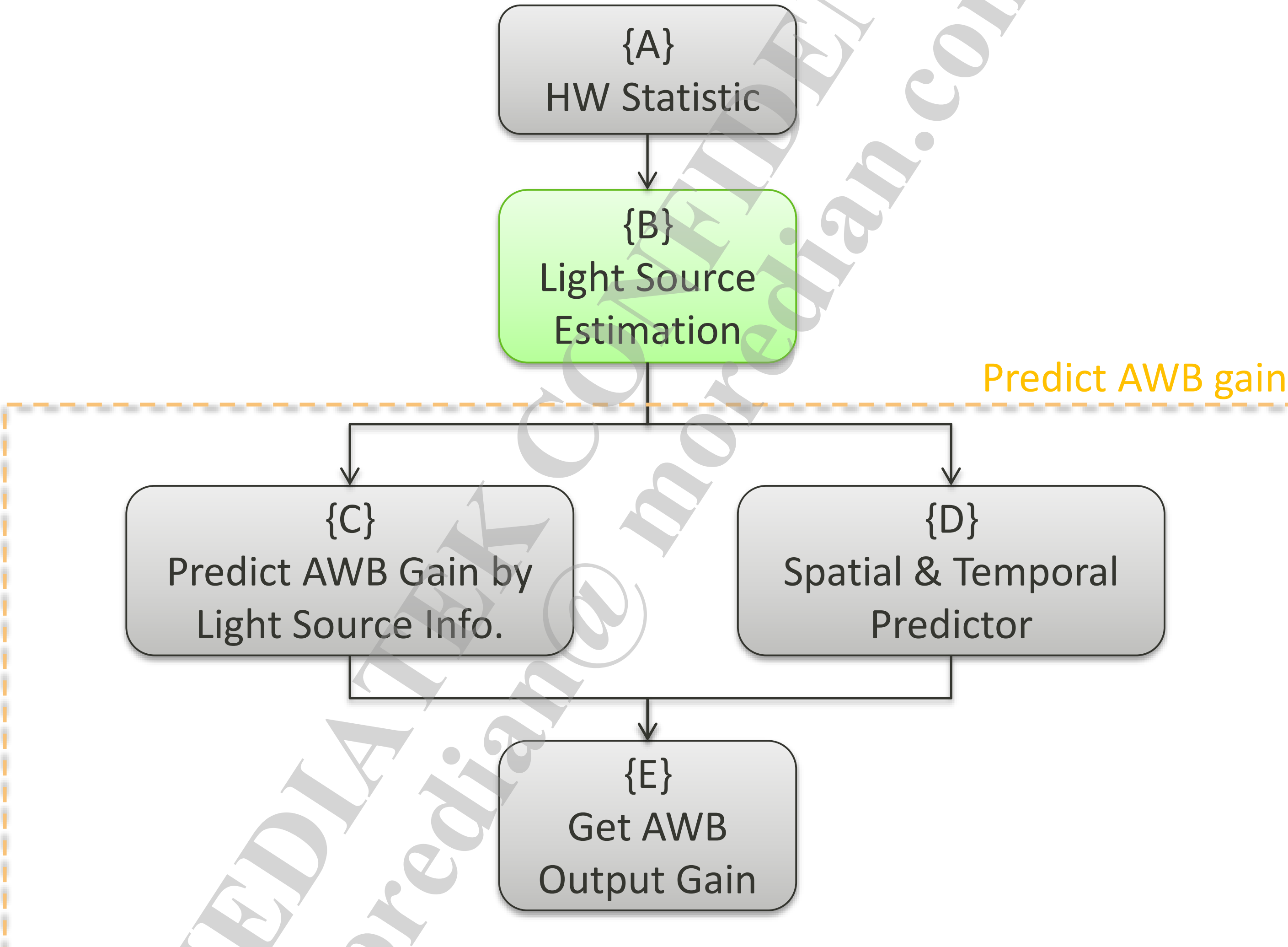
AWBwindow,Pregain1,分光源框等参考：MT6755_MT6750_AWB_Tuning_Introduction.pptx P8-P10




```
// Linear AAO parameter
{
    1,  //i4Enable
    20   //i4ErrCntThr
},
 // AWB error threshold
   0, // Programmable threshold f

 // AWB error count shift bits
   0, // Programmable error count

 // AWB error pixel ratio
  40, // Programmable error pixel

 // AWB motion error pixel ratio
  40  // Programmable motion errc
```


HW statistic 有改动，统计信息会更准确，容错率会更高。

{B} Light Source Estimation



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

120x90个childblock分成24x18个parentblock，判断每一个parent block属于哪一个光源。然后会计算各光源所有parent block的weighted R/G/B sum. Parent Block 的weight有如下面三种Mode可选。

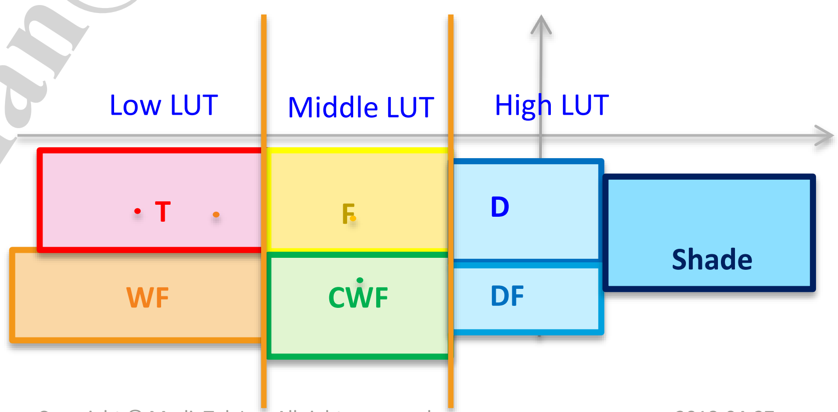
- Parent Block Weight (影响statistic gain) High luminance dominate
  - Mode 1 (Original)

Scaling Factor值越大，亮度对权重的影响越小，值愈小，则亮度对权重的影响越大。Factor固定时，block越亮，则该block权重越大。影响最终的statistic gain越多

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

↓

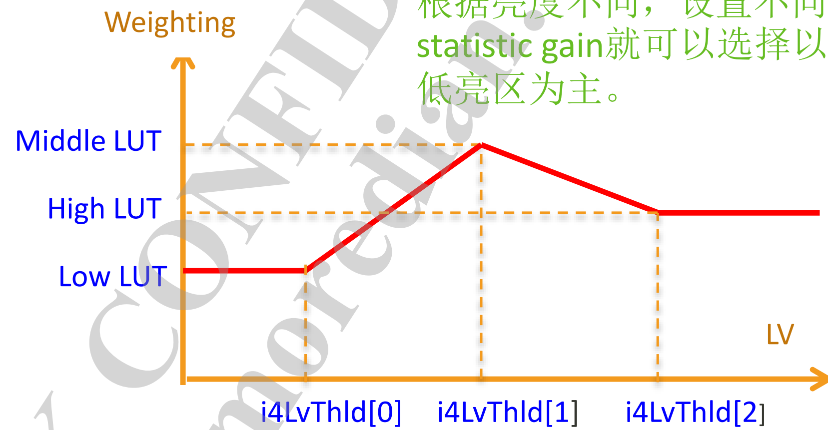
Light Source Statistics

↓

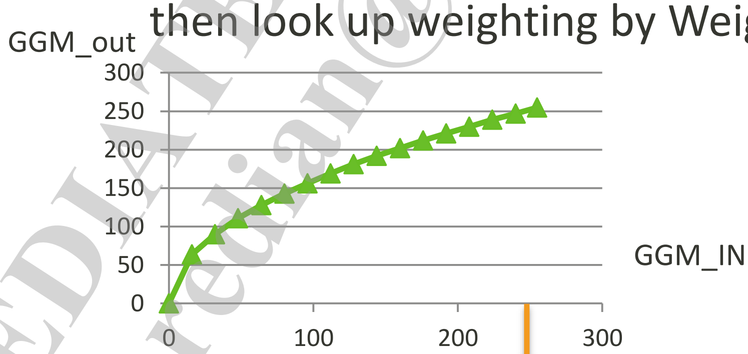
Light Source Estimation P0, P1, P2

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

不同亮度的block落到同一个光源，则可以根据亮度不同，设置不同的权重。这样statistic gain就可以选择以高亮区为主还是低亮区为主。

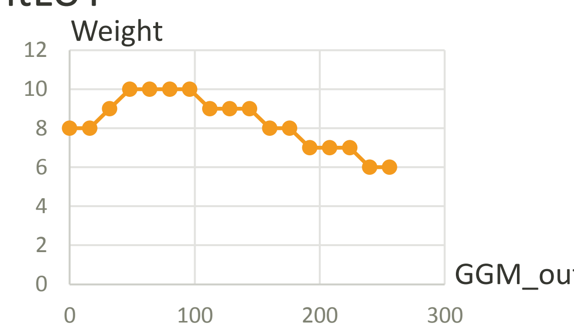


- mode2 & 3 use Gamma LUT transfer to vision linear domain and then look up weighting by WeightLUT



i4GammaLut

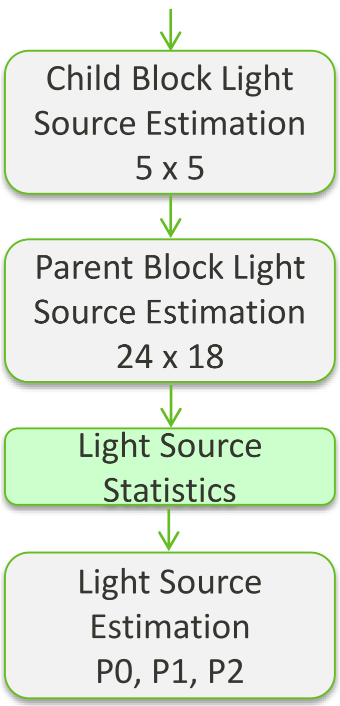
$$\frac{(R_{AVG} + G_{AVG} + B_{AVG})}{3}$$



i4WeightLut

AWB v5.0

Light Source Statistics



- 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

4	8	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240

GMA input

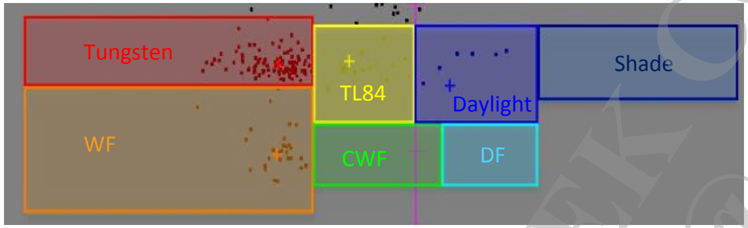
$$\frac{(R_{AVG} + G_{AVG} + B_{AVG})}{3}$$

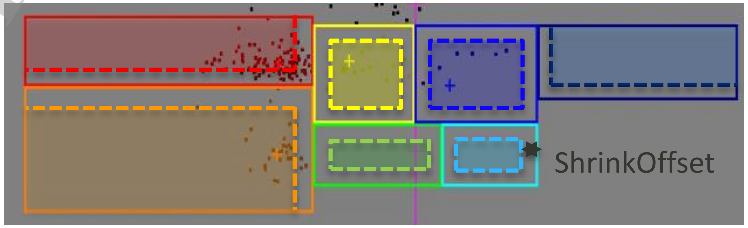
例： Mode2,当前parent block判为D65光源， 则weighttable用High那组， gamma output是130， 则确定weight是10.

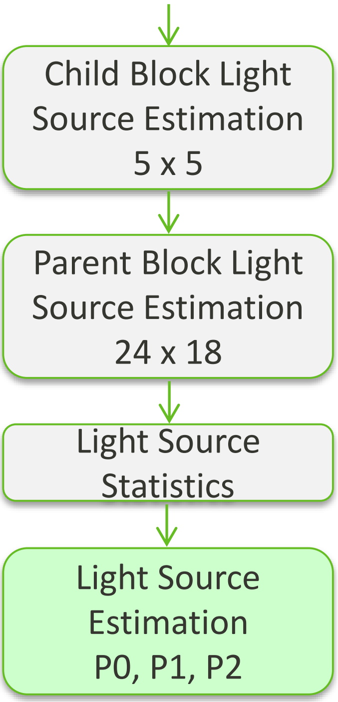
例： Mode3,当前画面： LV115（介于i4LvThld2和i4LvThld3之间）， gamma output是240， 则weighttable用mid和High插值后再用gamma output去index， weight是6.875(等于7).

Light Source Statistics

- **Statistic Smooth**
  - 参考MT6757_Camera_AWB_Tuning_Introduction.pdf p14-P18







Light Source Probability

For each light source L_i, define a probability $P(L_i)$ calculate b
3 type probability.

$$P_{(i)} = P0_{(i)} * P1_{(i)} * P2_{(i)}$$

P_0 : Probability of weighted parent block number
P_1 : Probability of scene luminance level
P_2 : Probability of confusion color (for T, WF, Shade)

EXIF
AWB_TAG_P0_STB : 0
AWB_TAG_P0_T : 0
AWB_TAG_P0_WF : 0
AWB_TAG_P0_F : 1
AWB_TAG_P0_CWF : 92
AWB_TAG_P0_D : 7
AWB_TAG_P0_S : 0
AWB_TAG_P0_DF : 0

EXIF
AWB_TAG_P1_STB : 100
AWB_TAG_P1_T : 100
AWB_TAG_P1_WF : 100
AWB_TAG_P1_F : 100
AWB_TAG_P1_CWF : 100
AWB_TAG_P1_D : 100
AWB_TAG_P1_S : 100
AWB_TAG_P1_DF : 100

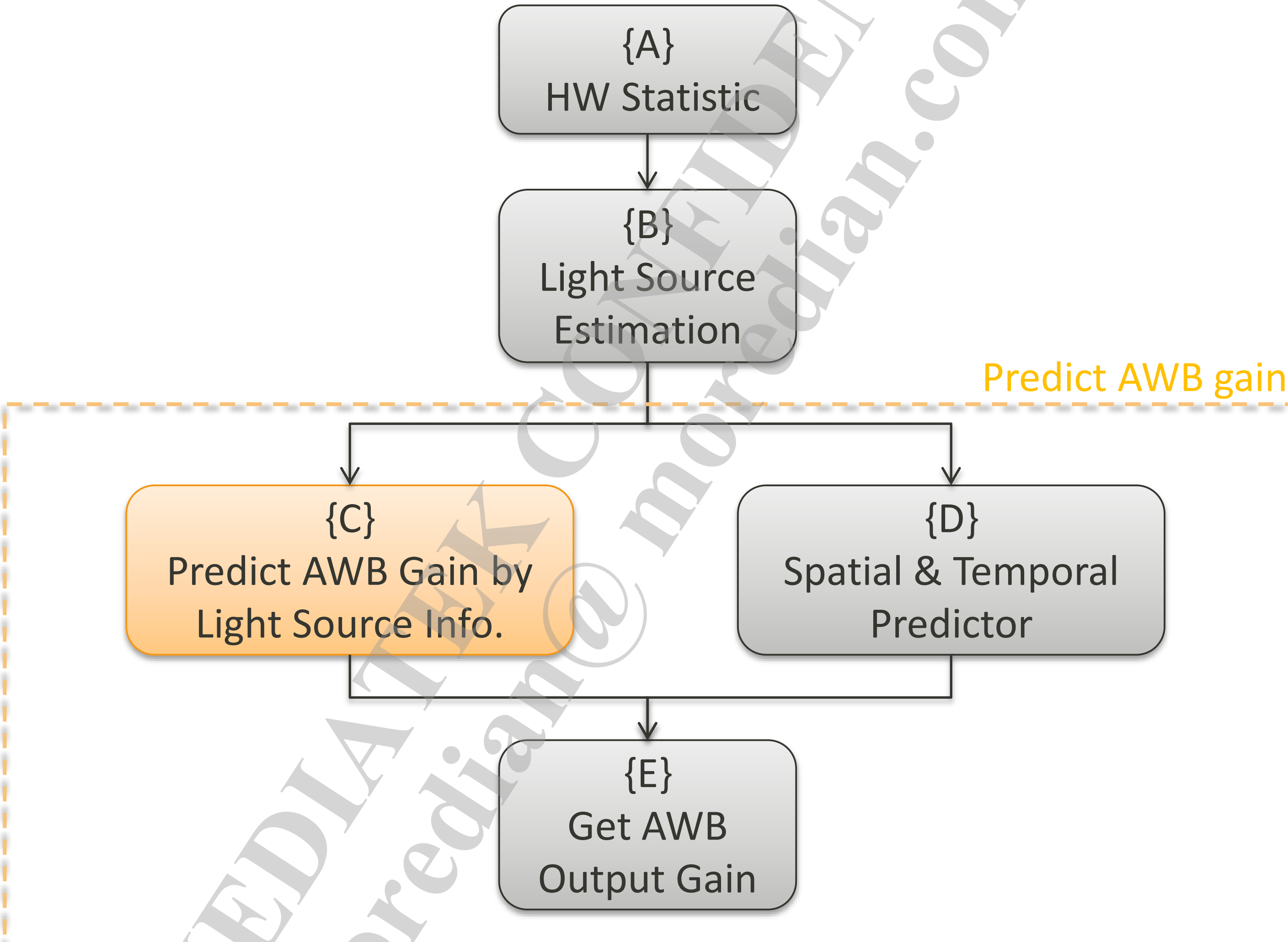
EXIF
AWB_TAG_P2_STB : 100
AWB_TAG_P2_T : 100
AWB_TAG_P2_WF : 100
AWB_TAG_P2_F : 100
AWB_TAG_P2_CWF : 100
AWB_TAG_P2_D : 100
AWB_TAG_P2_S : 100
AWB_TAG_P2_DF : 100

EXIF
AWB_TAG_P_STB : 0
AWB_TAG_P_T : 0
AWB_TAG_P_WF : 0
AWB_TAG_P_F : 1
AWB_TAG_P_CWF : 92
AWB_TAG_P_D : 7
AWB_TAG_P_S : 0
AWB_TAG_P_DF : 0

Light Source Probability

P0,P1,P2 ,P0 Stability,P2 stability :
参考MT6757_Camera_AWB_Tuning_Introduction.pdf P51-P57

{C} Predict AWB Gain by Light Source Info.



{C} Predict AWB Gain by Light Source Info.

- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- Preference Gain
- Face Assisted AWB

Preference Color Compensation

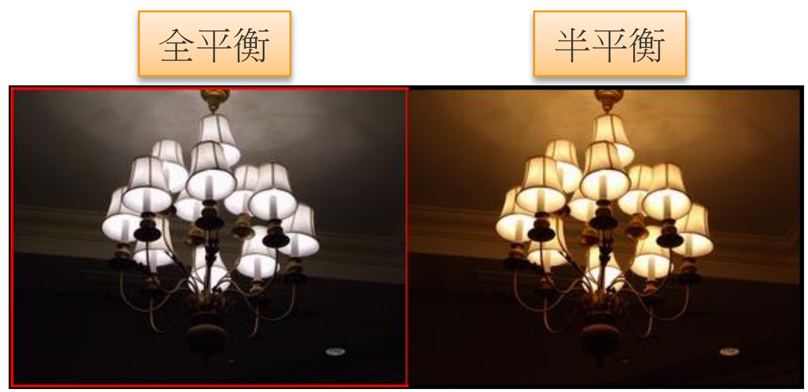
- **Purpose :**

  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

Daylight Locus offset和magenta offset以及green offset对gain值和颜色的影响参考：
MT6757_Camera_AWB_Tuning_Introduction.pdf P25-P32

Preference Color Compensation

AWB
v5.0

Exif多打印出如下信息：

1、Daylight locus 方向：AWB_TAG_LUT_OFFSET_T 这一段距离所代表的gain值。

2、 Daylight locus 垂直方向：AWB_TAG_GM_OFFSET_T - AWB_TAG_GM_OFFSET_THR_T这一段距离所代表的gain值。

3、AWB_TAG_PREFER_GAIN_R_T 和 AWB_TAG_PREFER_GAIN_B_T 是a.Daylight locus 方向,b.Daylight locus 垂直方向,c.DL方向preference gain,d.DL垂直方向preference gain， 四个部分（即1,2,4,5）组成的总gain值即T光源总共保留未收白的gain。

新增部分：

4、Daylight locus 方向增加DL preference gain.

5、Gmoffset 方向增加 GM preference gain.

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 — 1、Daylight locus offset corresponding gain
AWB_TAG_GM_OFFSET_GAIN_R_T : 528
AWB_TAG_GM_OFFSET_GAIN_B_T : 527 — 2、GM offset corresponding gain
AWB_TAG_DL_OFFSET_PREF_GAIN_R_T : 512
AWB_TAG_DL_OFFSET_PREF_GAIN_B_T : 512 — 4、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 — 5、GMoffset preference gain by tuning
AWB_TAG_PREFER_GAIN_R_T : 554
AWB_TAG_PREFER_GAIN_B_T : 500 — 3、

Preference Color Compensation

AWB
v5.0

Preference Color compensation影响的是statistic gain

```
//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； Input 为EXIF中AWB_TAG_RATIO_OFFSET_T

Preference gain LUT

Daylight locus offset preference gain

Magenta offset preference gain

Green offset preference gain

```
{ // TUNGSTEN
    { // LUT: use daylight Locus ratio offset as index
        {512, 512}, {512, 512}, {512, 512}, {500, 512}, {500, 516}, {505, 518}, {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},
    },
    { // LUT: use (magenta offset - thld) as index
      //  0   100   200   300   400   500   600   700   800   900  1000
        {512, 512}, {520, 520}, {515, 518}, {512, 512}, {512, 512}, {500, 516}, {500, 520}, {512, 512}, {512, 512}, {512, 512}, {512, 512}
    },
    { // 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}
    },
},
```

1、Magenta_offset or Green_offset 小于thr的时候，则Magenta和Green preference gain就不会作用。
2、Daylight locus offset preference gain和Gmoffset preference gain主要作用是在AWB_TAG_LUT_OFFSET_T 和AWB_TAG_GM_OFFSET_T -AWB_TAG_GM_OFFSET_THR_T的基础上再分别自由乘以一个gain值，使得statistic gain往希望的方向调整。
Note： statistic gain是Daylight locus offset preference gain和GMoffset preference gain后的结果。

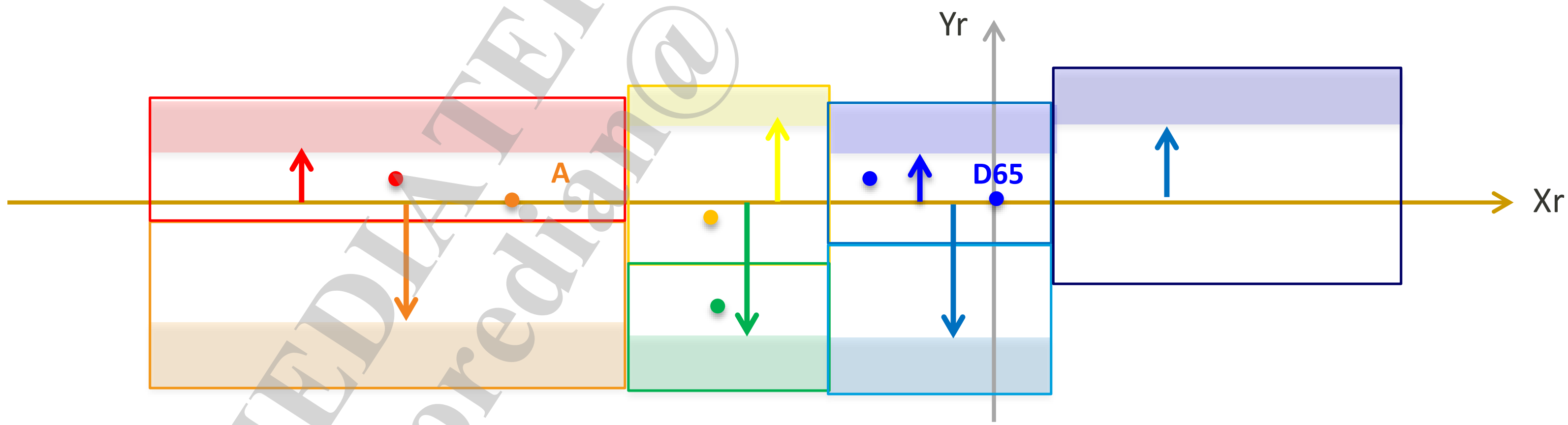
{C} Predict AWB Gain by Light Source Info.

- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- Preference Gain
- Face Assisted AWB

AWB v5.0

Statistic Gain Constraint (1/3)

- **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 (影响Statistic Gain)
       - Add new mode to project by each parent block
       - Add projected weighting to control project ratio
    2. Reduce Statistic Weighting (影响Statistic Weighting)



AWB v5.0

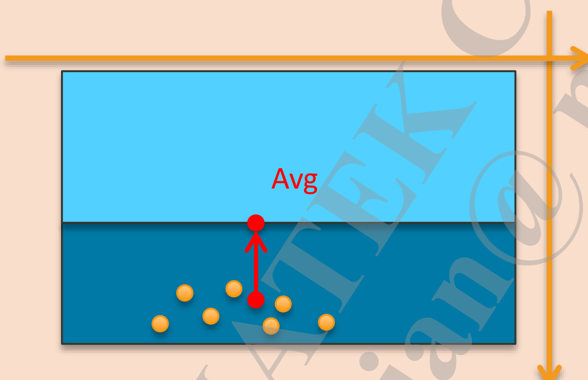
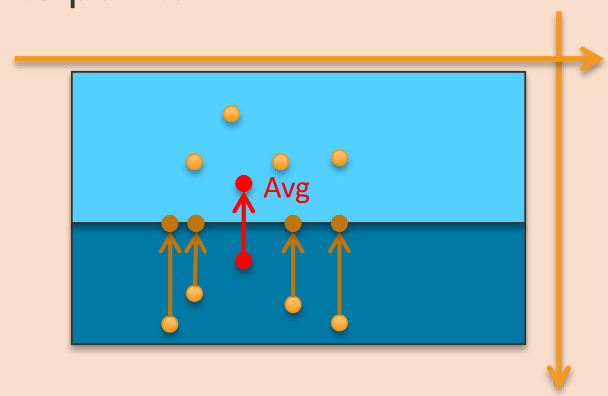
Statistic Gain Constraint (2/3)

- **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(i4ProjectReduce[AWB_LIGHT_NUM])
    - Unit : 16
    - Mode1和mode2可选择limit的比例。

```
// 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 (3/3)

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

Mode1:每个光源中All PB average在limitY下方才会把average投影到limitY的位置。

Mode2:
1、每个光源中判断each PB位置，如 在limitY下方则把该PB投影到limitY，然后再结合limitY上方的PB计算average。作用更强。
2、可以通过i4ProjWeight控制投影距离。

Case部分参考：MT6757_Camera_AWB_Tuning_Introduction.pdf P37-P39

{C} Predict AWB Gain by Light Source Info.

- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- Preference Gain
- Face Assisted AWB

- Spatial predictor： 参考MT6757_Camera_AWB_Tuning_Introduction.pdf P20-P21
- Sunset compensation：参考MT6757_Camera_AWB_Tuning_Introduction.pdf P23-P24

{C} Predict AWB Gain by Light Source Info.

- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- Preference Gain
- Face Assisted AWB

Mix Statistic Gain & Spatial Gain

- 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
    - AWB daylight locus probability (Statistic Weighting) look-up table
    - AWB NR(H,A,CWF,F,D65,Shade,DF)
    - SUB window(F,CWF,D)

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

参考：MT6757_Camera_AWB_Tuning_Introduction.pdf P40-P47

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

{C} Predict AWB Gain by Light Source Info.

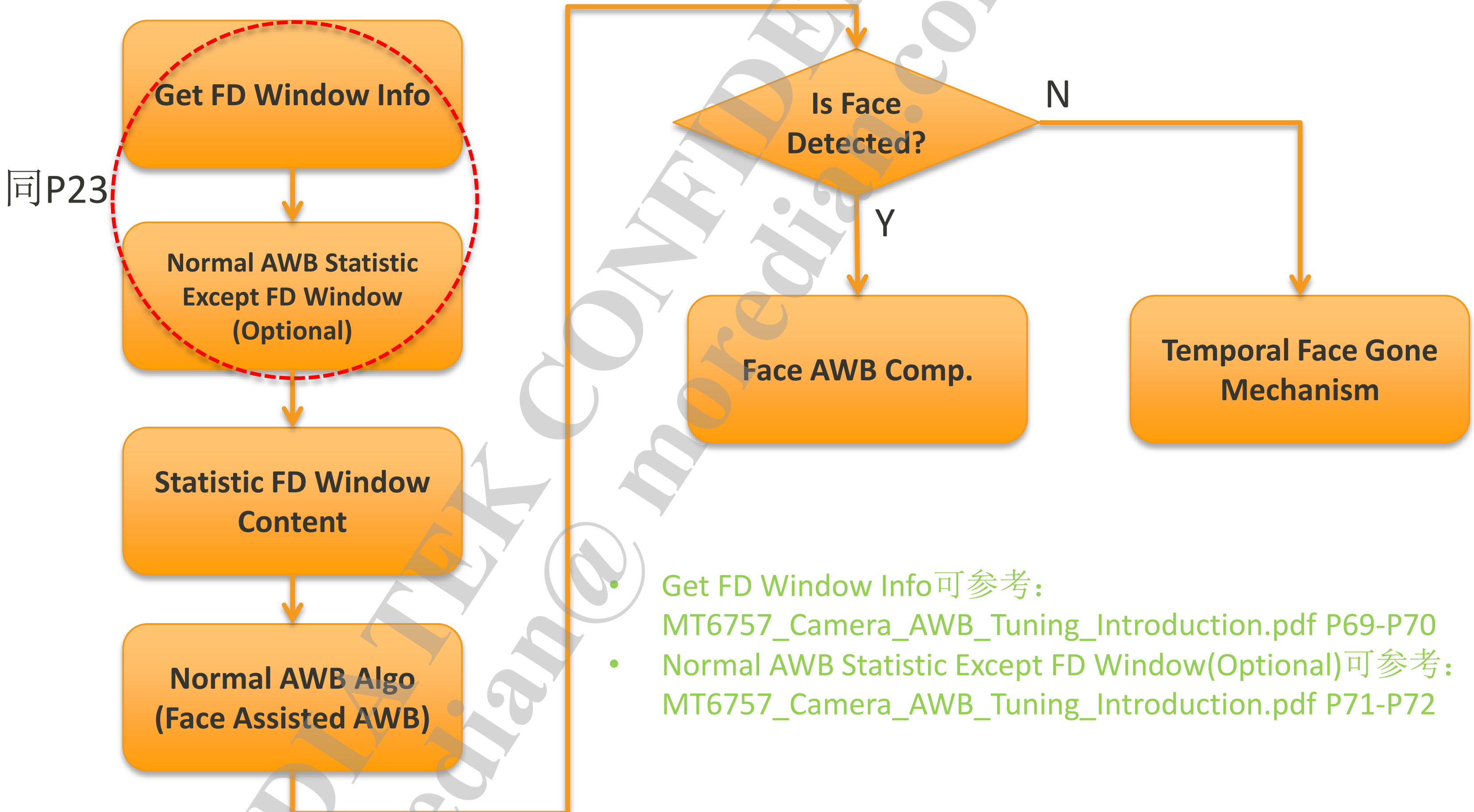
- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- **Preference Gain**
- Face Assisted AWB

Preferrence gain同之前平台即每个光源每个LV有一组preference gain，可参考：
MT6757_Camera_AWB_Tuning_Introduction.pdf P49

{C} Predict AWB Gain by Light Source Info.

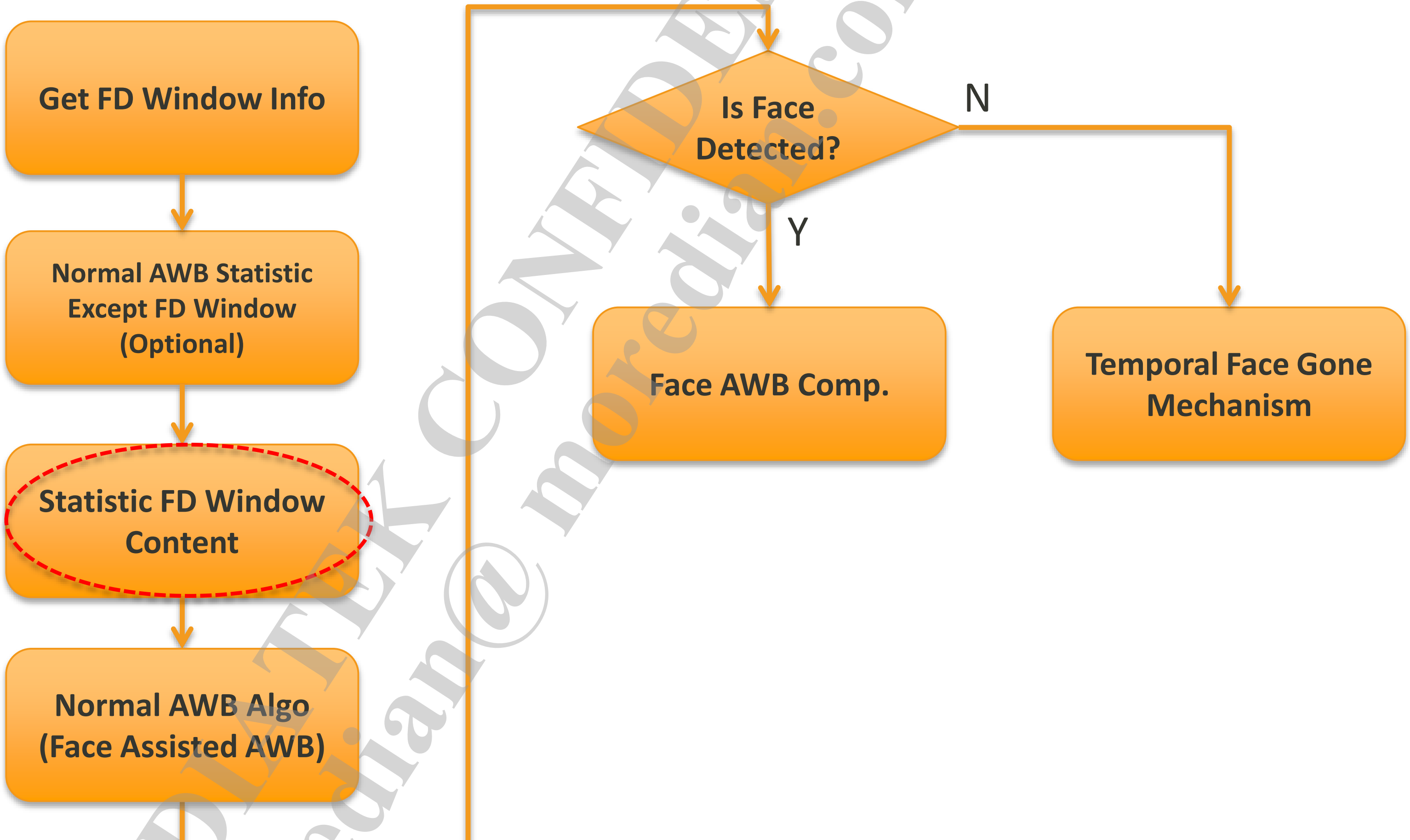
- Preference Color Compensation
- Statistic Gain Constraint
- Spatial Predictor & Sunset Compensation
- Mix Statistic Gain & Spatial Gain
- Preference Gain
- Face Assisted AWB

AWB Algo Flow



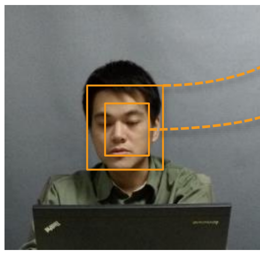
- Get FD Window Info可参考：
  MT6757_Camera_AWB_Tuning_Introduction.pdf P69-P70
- Normal AWB Statistic Except FD Window(Optional)可参考：
  MT6757_Camera_AWB_Tuning_Introduction.pdf P71-P72

AWB Algo Flow



Statistic FD Window Content

Central weight



Scaled FD Window

Central Window (1/4 of original FD window)

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

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

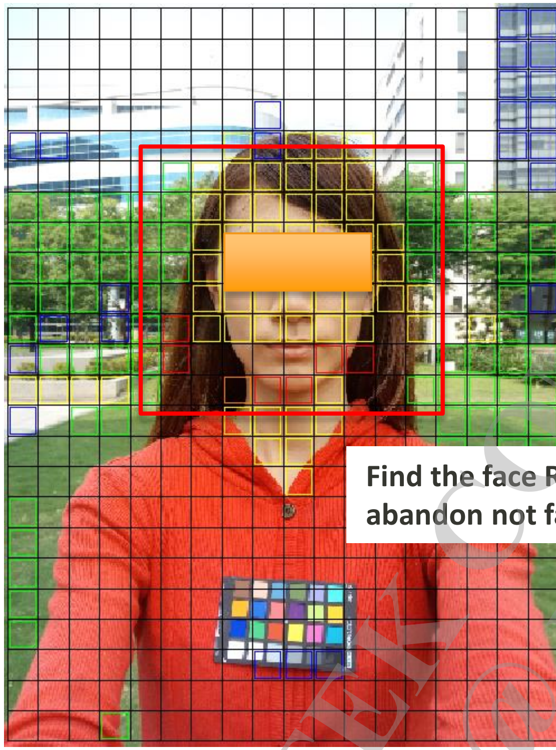
Final output:
Weighted average Face RGB.
(Weighed by face area size)

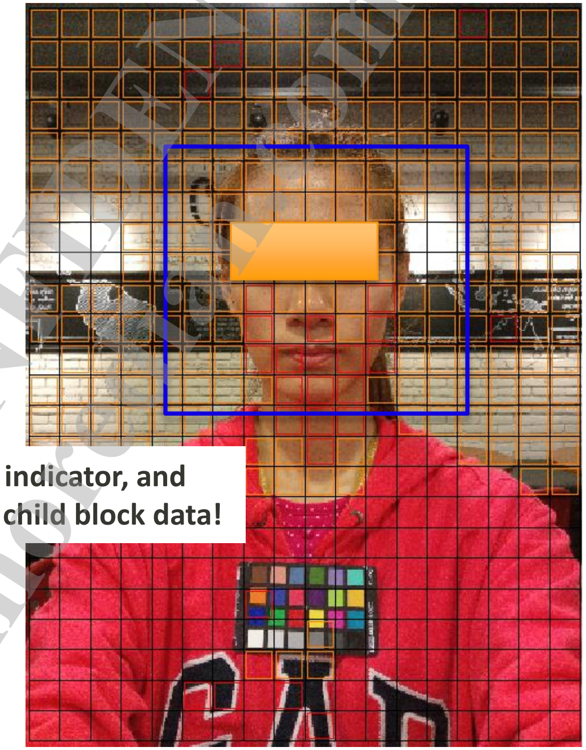
```
1,      //u4FaceCentralWeight  0=>0, 1=>1/2, 2=>2/3, 3=>3/4 ...
// rStatisticNR
{
    1,      //i4Enable
    200     //i4DiffThr
},
// rFD_RGB_Bound
{
     1,     //u4LowBound
    254     //u4HiBound
},
```

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

AWB v5.0

Face Statistic NR Process



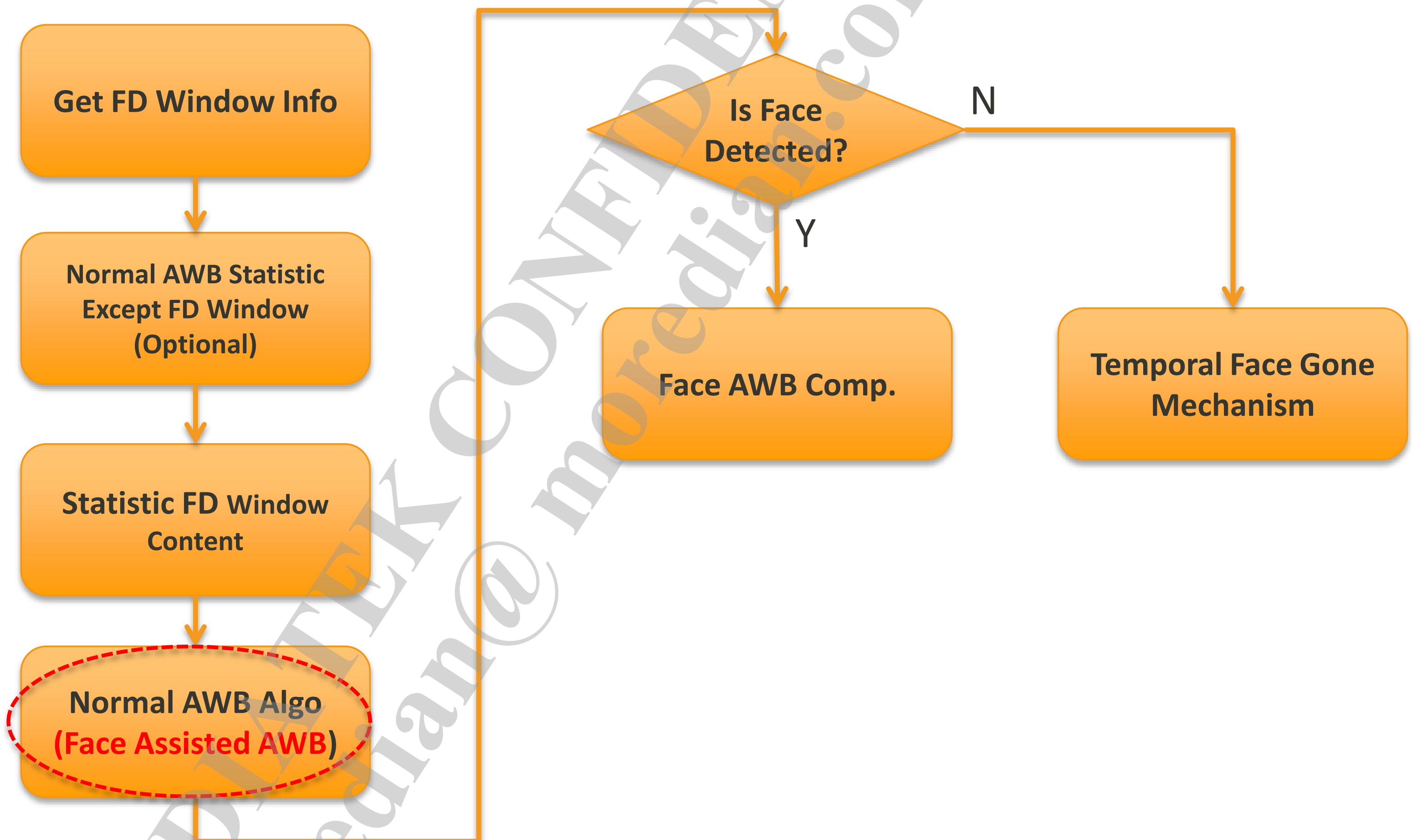


Find the face RGB indicator, and abandon not face child block data!

```
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 Algo Flow



AWB
v5.0

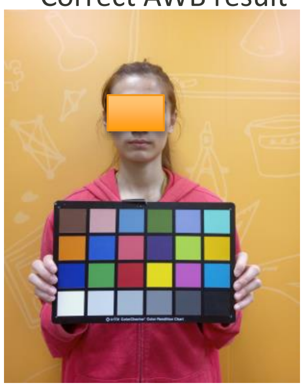
Face Assisted AWB

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

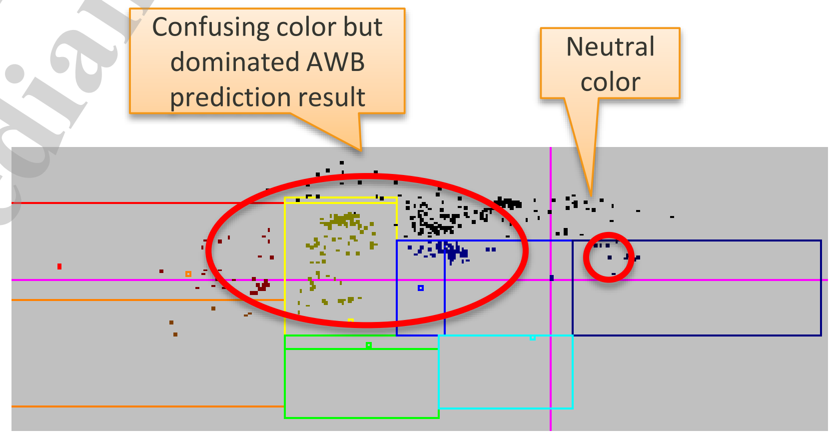
- Purpose
  - Using face color to predict light source/gain to assist normal AWB prediction
  - To improve accuracy even in confusing-color scenes

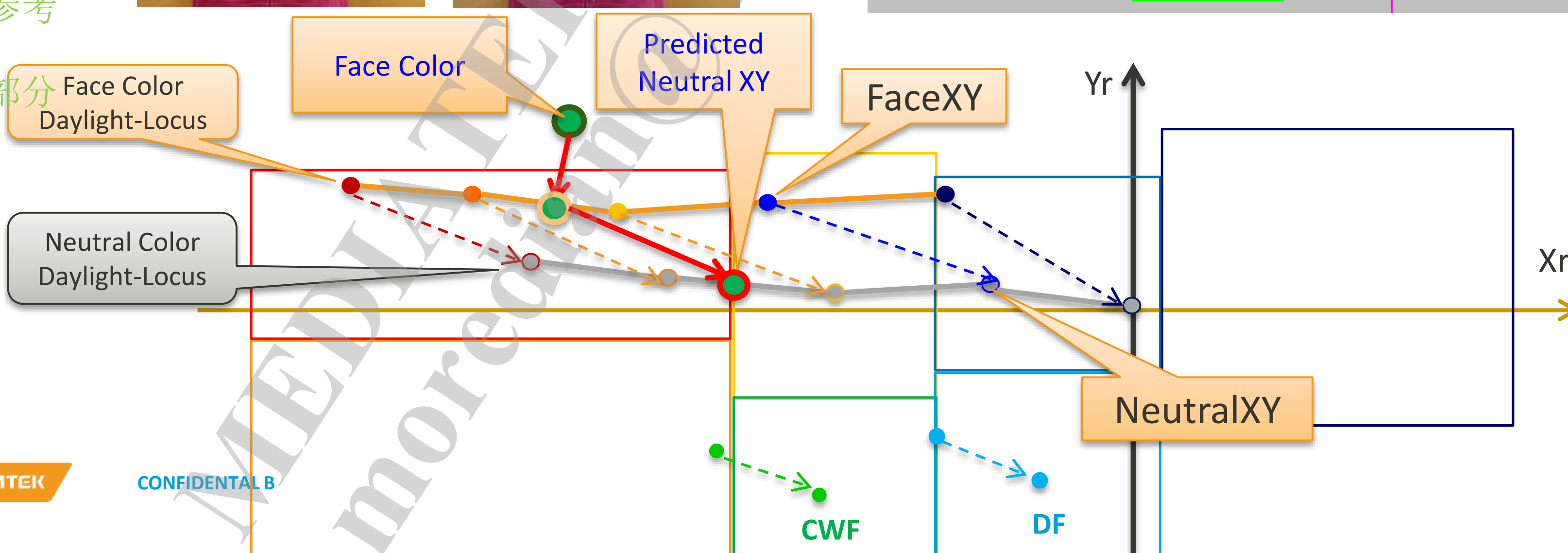
Note：各光源下Face color的位置需要calibration，具体方法，参考MT6771 caseshare部分

Correct AWB result



Incorrect AWB Result

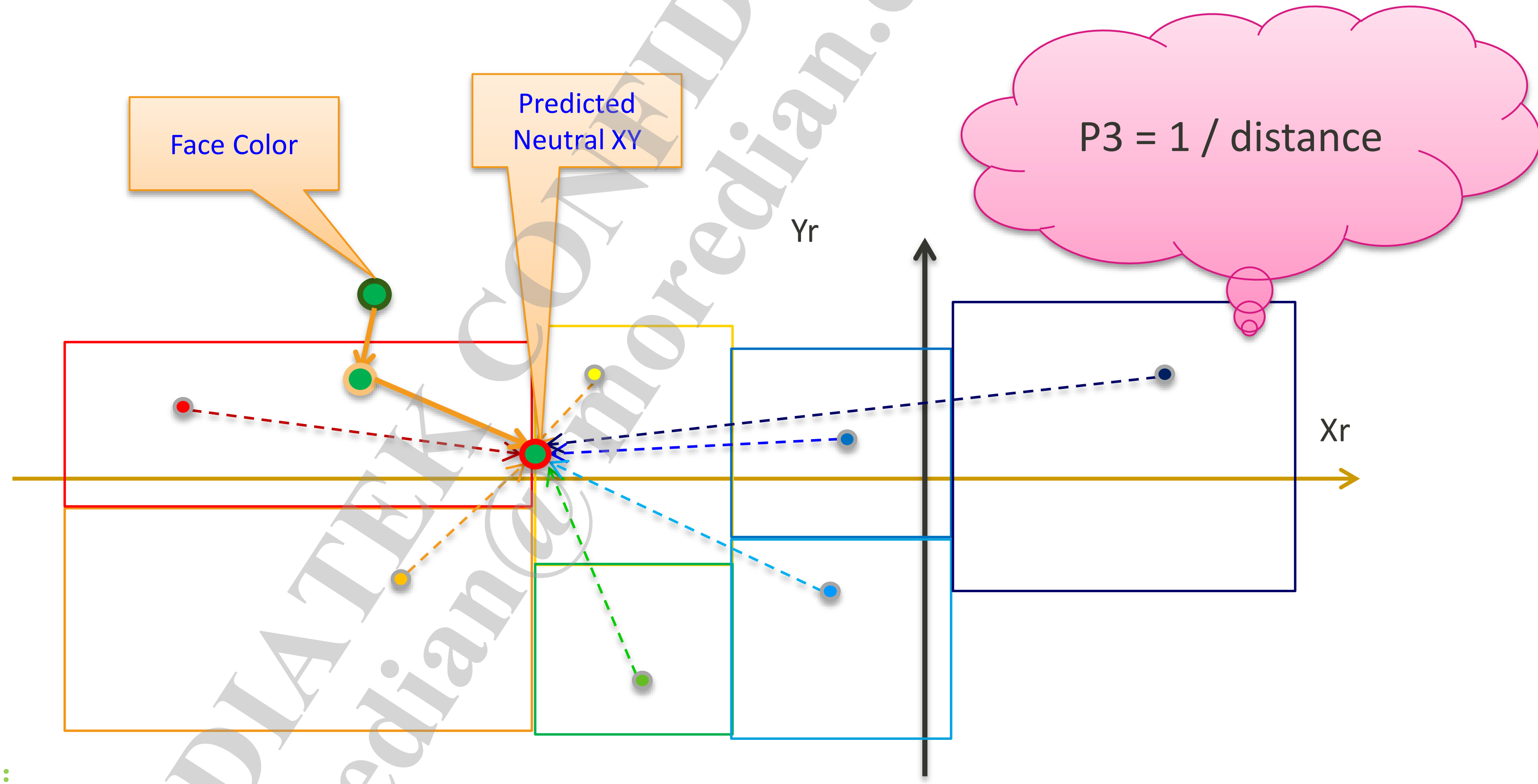




AWB v5.0

Face Assisted AWB

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



Note：
1、距离越近则该光源P3越大，最大为100,表示该光源gain的占比(P)不会被减弱。
2、距离越远则该光源P3越小，小于100,表示该光源gain的占比(P)会被减弱，减弱的部分会用face predictor gain补充。

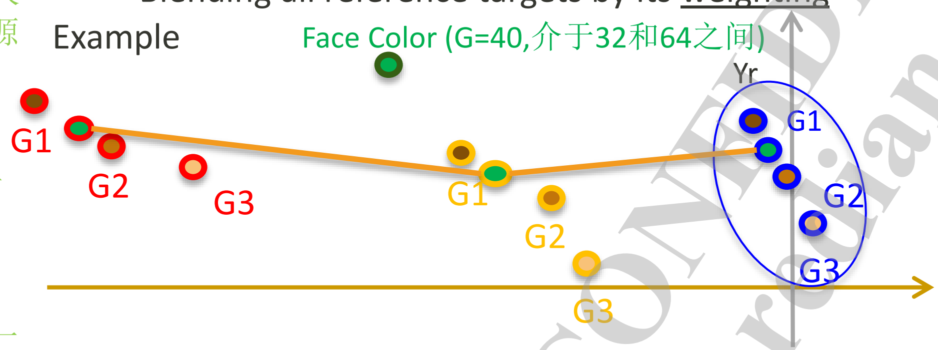
AWB v5.0

Face Assisted AWB - Predictor Estimation

Note：face predictor gain的计算过程：
1、先根据当前face Gvalue与i4RefTargetTHr的关系，给每一个光源分别插值出一个face reference targetXY。
2、再根据距离计算每个光源的weighting。
3、然后根据weighting得到出一个face reference target XY，在Tolerance范围内。
4、 face reference target XY会映射出predicted ideal neutral target XY。
5、再加上face shift xy得到predicted neutral target XY。
6、最后乘以face assistant Preference Gain最终得到face predictor gain。
7、如果是T/A/S光源，还会经过prefer color机制才会得到最终的face Predictor gain。

- 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

Example

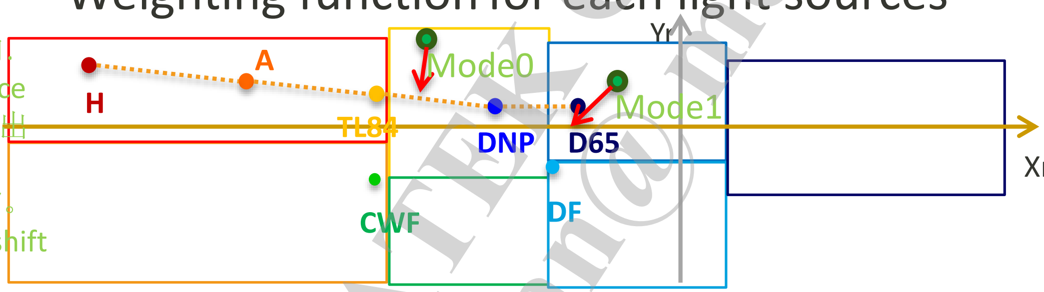



```
// i4RefTargetXY
{
  // ThrLow         ThrMid          ThrHigh
  {{    0,    0}, {    0,    0}, {    0,    0}},  // STB
  {{-665, -406}, {-665, -406}, {-665, -406}},  // T
  {{-505, -415}, {-505, -415}, {-505, -415}},  // WF
  {{-284, -450}, {-284, -450}, {-284, -450}},  // F
  {{-276, -592}, {-276, -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
},
```

Xr

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

- Weighting function for each light sources




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

- Mode0：CWF和DF不参加计算，直接将当前face xy投影到daylight locus线上，距离某个光源ideal 落点越近则那个光源weighting越大。 i4WeightCoef_a, i4WeightCoef_b不起作用
- Mode1:CWF和DF参与计算， weighting受距离和 i4WeightCoef_a, i4WeightCoef_b 控制。如：i4WeightCoef_a值越大距离差异影响越大，值越小距离差异影响越小。

EXIF
AWB_TAG_FACEAST_DIST_WEI_XX

AWB v5.0

Face Assisted AWB - Predictor Estimation

Face prediction result by weighting function

- (1) Reference target-XY
- (2) Tolerance
- (3) Predicted ideal neutral-XY
- (4) Predicted-Neutral-XY
  - Ideal-Neutral-XY + Face-Shift-XY for color preservation
  - **<u>Face-Shift XY</u>**主要是Yr方向的
    - 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

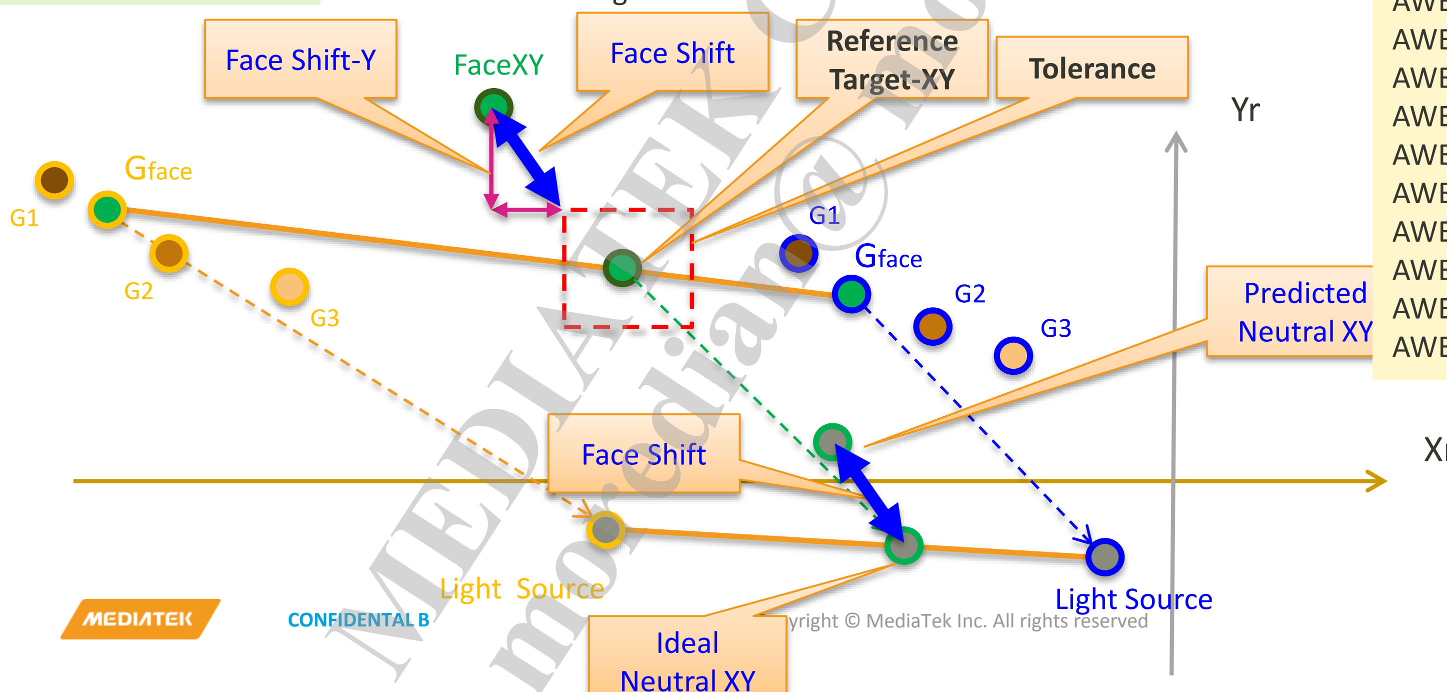
```
//i4TOL
{
   32,   //STB
   36,   //H
   36,   //A
   32,   //TL84
   30,   //CWF
   28,   //DNP
   24,   //D65
   24,   //DF
},
```

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

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
  - (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
},
```

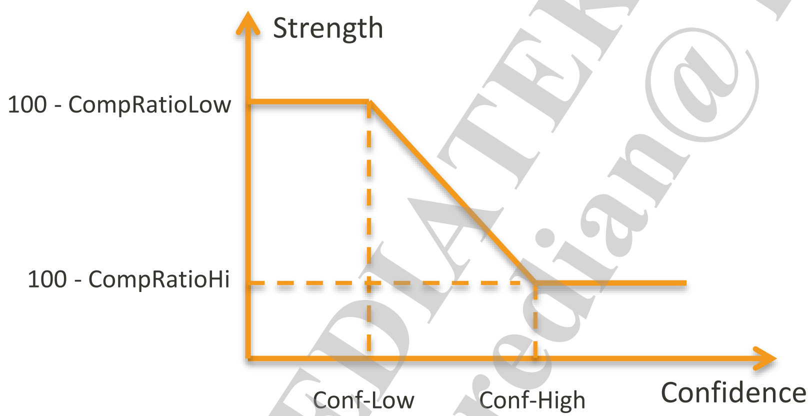
[Tips]
1. 单要调某个亮度或FaceG，且不希望影响其他二组reference target
   → 修改Reference Target XY (i4RefTargetXY)
2. 调整某一个光源下的整体色调或Preference
   → 修改Preference Gain (rPrefGain)
   请勿直接修改"Rotated XY coordinate of AWB light source"，此为calibration result

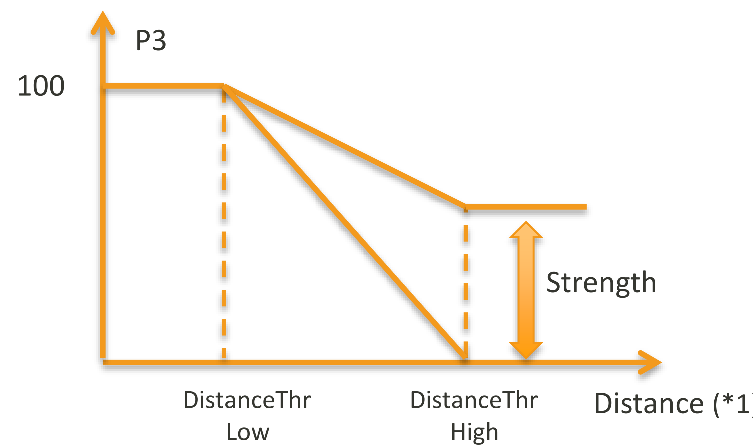
AWB v5.0

Face Assisted AWB – Face Predictor Probability

- P3 is the probability according
  - The face predictor 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
},
```



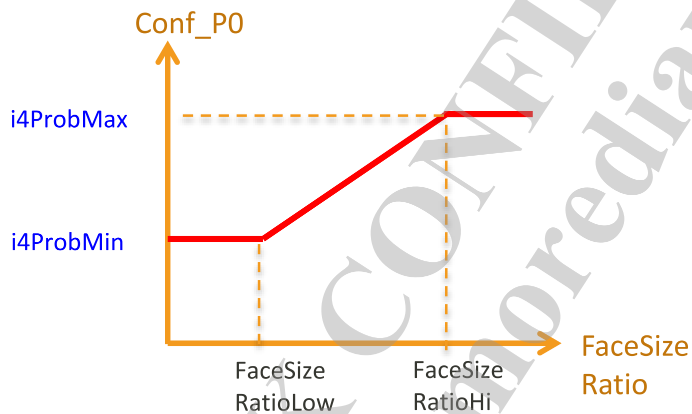


***1. Distance: distance between EqvXY of light-src and NeutralXY of Face-Predictor**

AWB v5.0

Face Assisted AWB – Face 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 – Face 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

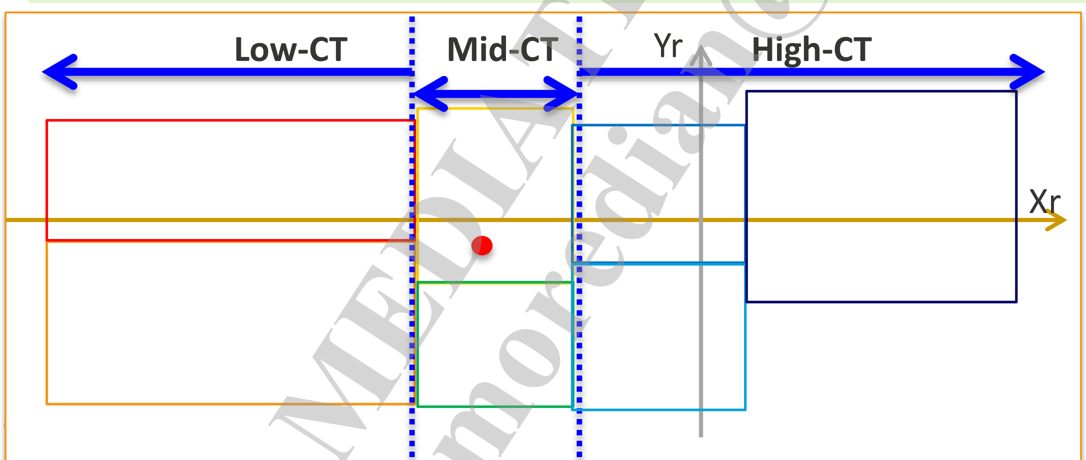
AWB v5.0

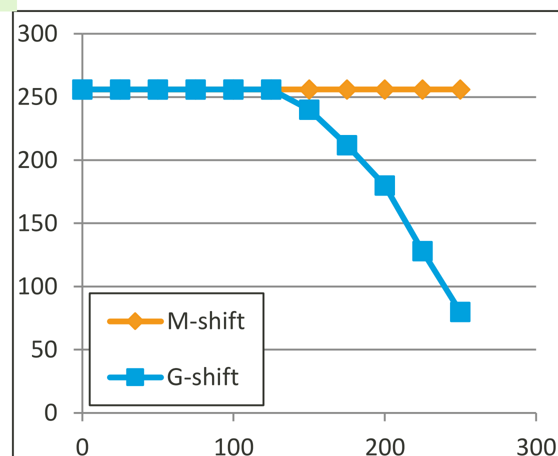
Face Assisted AWB – Face Predictor Probability

- 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
},
```

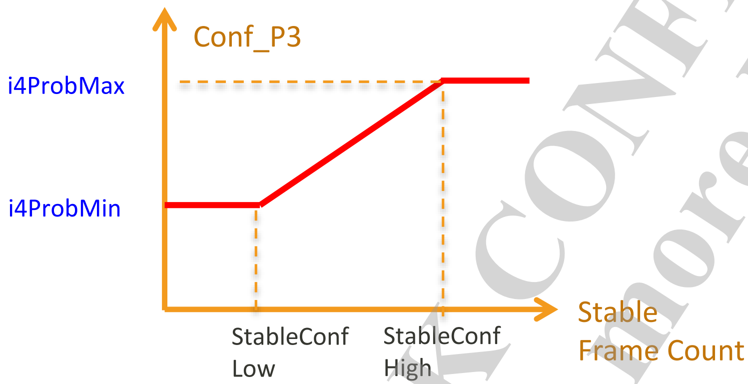
EXIF
AWB_TAG_FACEAST_CONF_PROB2 : 100





Face Assisted AWB – Face 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
},
```

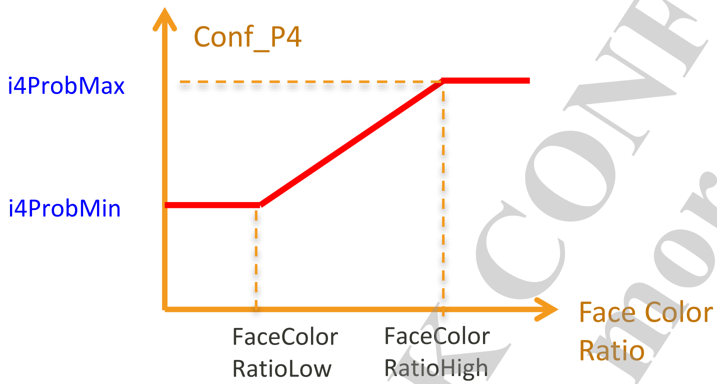
EXIF

AWB_TAG_FACEAST_CONF_PROB3 : 100

AWB
v5.0

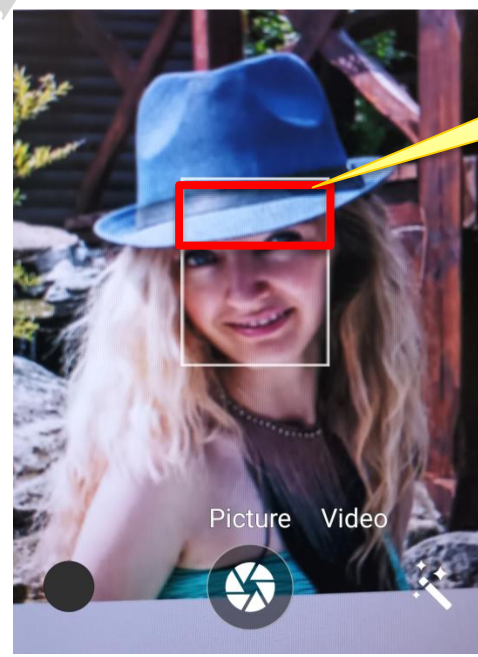
Face Assisted AWB – Face Predictor Probability

- 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

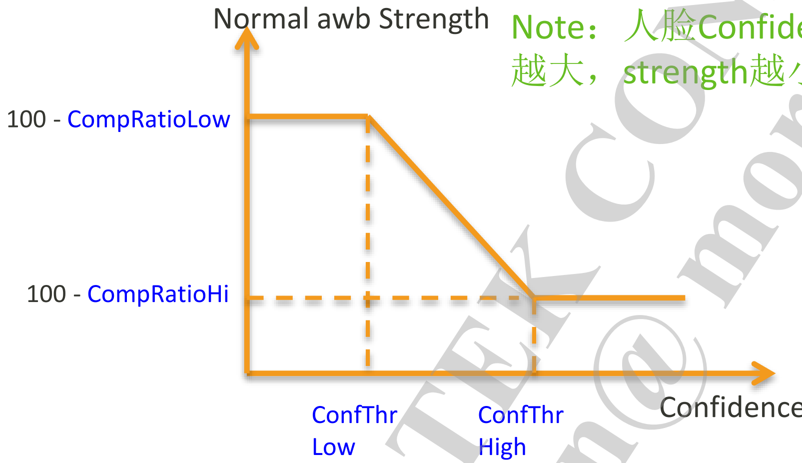


Invalid face color

AWB v5.0

Face Assisted AWB – Face 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



Note：人脸Confidence越大，strength越小。

```
// 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

Note：每个光源都有一个P3，P3表示此光源normal awb gain相对face predictorgain的可信度，也就表示某光源混normal awb gain的比例。
例如：normal awb的T光源P=60，P3=90，则F光源normal awb gain prob=60*0.9=54，face predictor gain prob=6
即F光源会混54%的normal awb gain，混6%的face predictor gain。

AWB v5.0

Face Assisted AWB – Face Predictor Probability

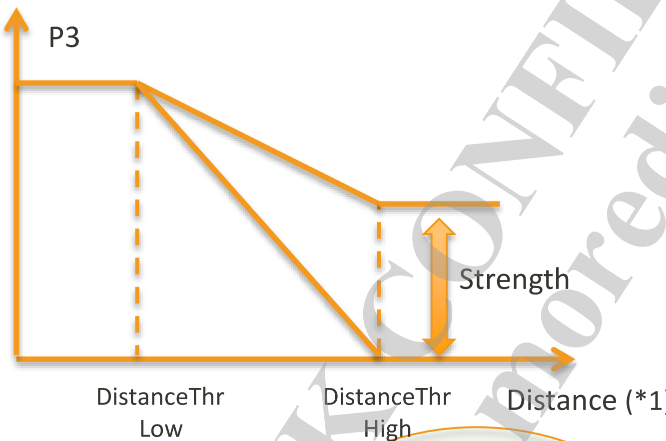
- P3 probability of each light source is calculated by
  - Strength of each light source
  - Distance between light source XY and face assisted predicted-neutral-XY

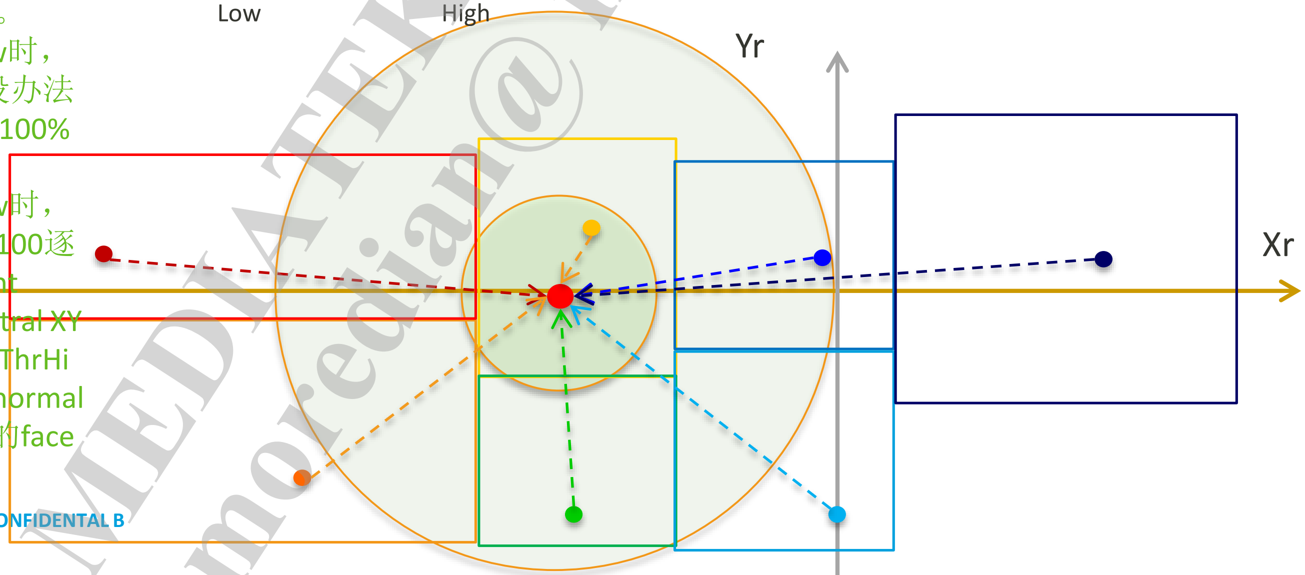
```
// rCompSetting
{
   { 0, 100 }, //rConfThr
   { 50, 200 }, //rDistanceThr
   { 0,  40 },//rCompRatio
},
```

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

Note：1、light source XY与predicted neutral XY之间distance<=DistanceThr Low的光源一直可信，P3不受confidence (strength)的影响保持为100，即100%用normal awb 。
2、light source XY与predicted neutral XY之间 distance > DistanceThr Low的光源，光源可信度下降，P3会受confidence (strength)影响，distance越大受影响越多。
a. confidence<ConfThrLow时，strength等于100，人脸没办法参考，P3保持为100，即100%用normal awb。
b. confidence>ConfThrLow时，strength逐渐减小，P3从100逐渐减小到strength，即light source XY与predicted neutral XY之间 distance > =DistanceThrHi的光源，用strength%的 normal awb，用(100-strength)%的face assist awb。





AWB v5.0

Face Assisted AWB - Final AWB Gain

- Example：
- $P[i] = P0[i] \times P1[i] \times P2[i]$ and normalized to 100, i is light sc
- $P_{Final}[i] = (P[i] \times P3[i])/100$: 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

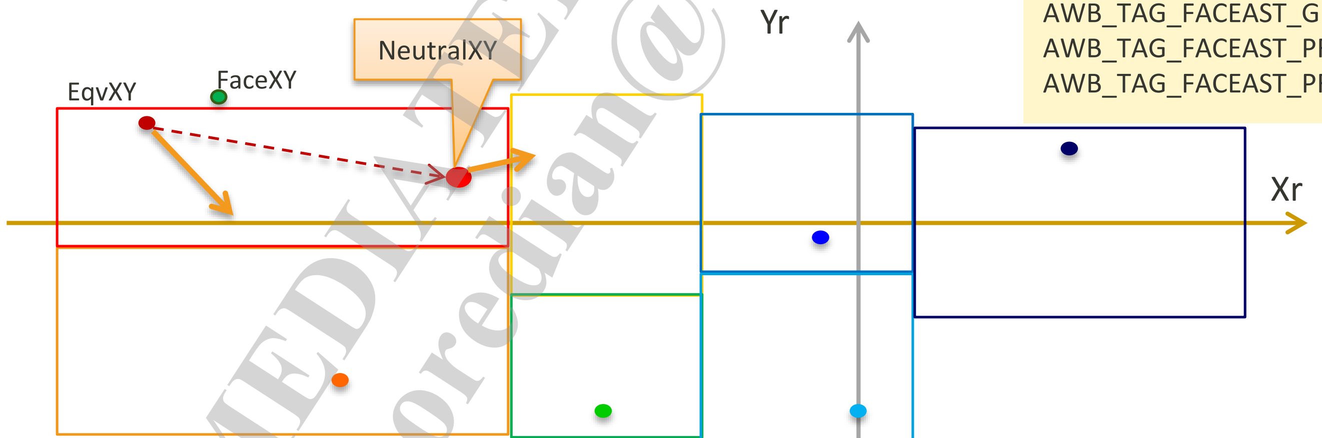
AWB_TAG_FINAL_GAINPROB_STB : 0
AWB_TAG_FINAL_GAINPROB_T : 2
AWB_TAG_FINAL_GAINPROB_WF : 1
AWB_TAG_FINAL_GAINPROB_F : 20
AWB_TAG_FINAL_GAINPROB_CWF : 0
AWB_TAG_FINAL_GAINPROB_D : 44
AWB_TAG_FINAL_GAINPROB_S : 1
AWB_TAG_FINAL_GAINPROB_DF : 0

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

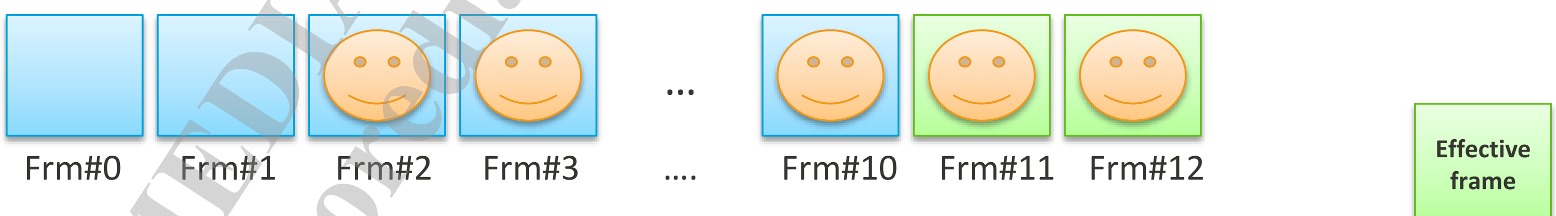
Face Assisted AWB – Stability

- 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
- 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 purpo
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
},
```

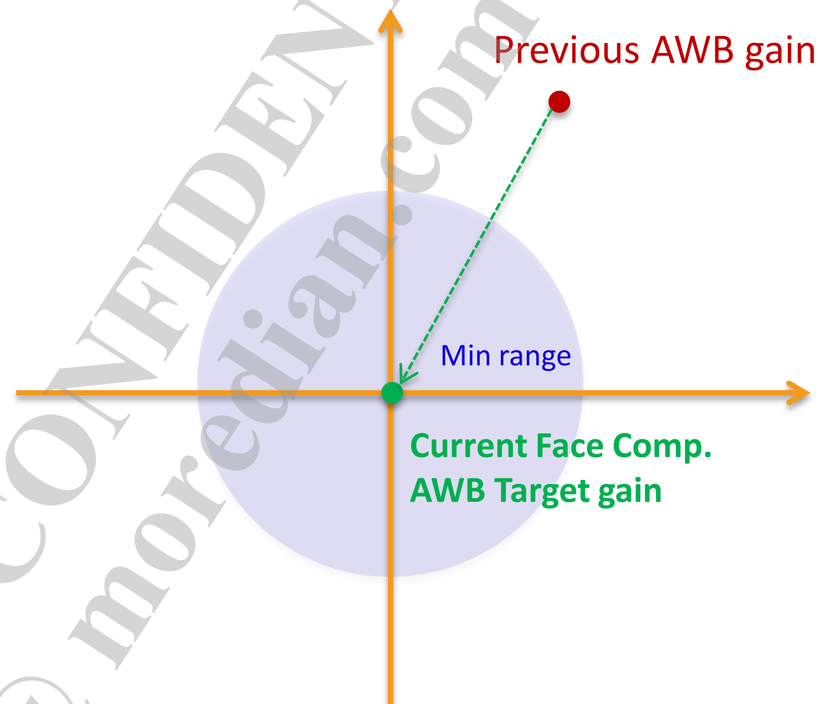


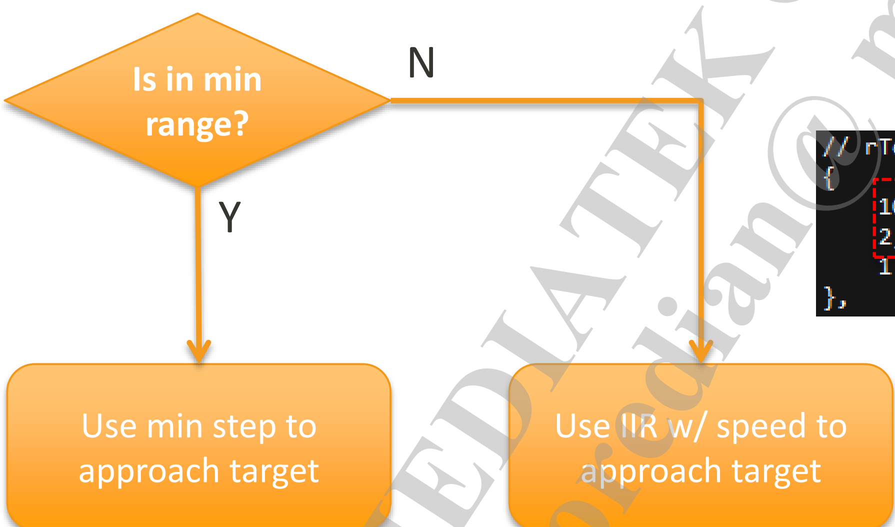
*1 – These parameters are common setting for Face-Assisted & Face-Comp. AWB
*2 – These parameters can not be simulated by ImagiqSimulator

Temporal Face AWB Gain Mechanism

IIR approaching target：
1. Converge speed = 1~100
2. Minimum step

Minimum range = MinStep*100/speed



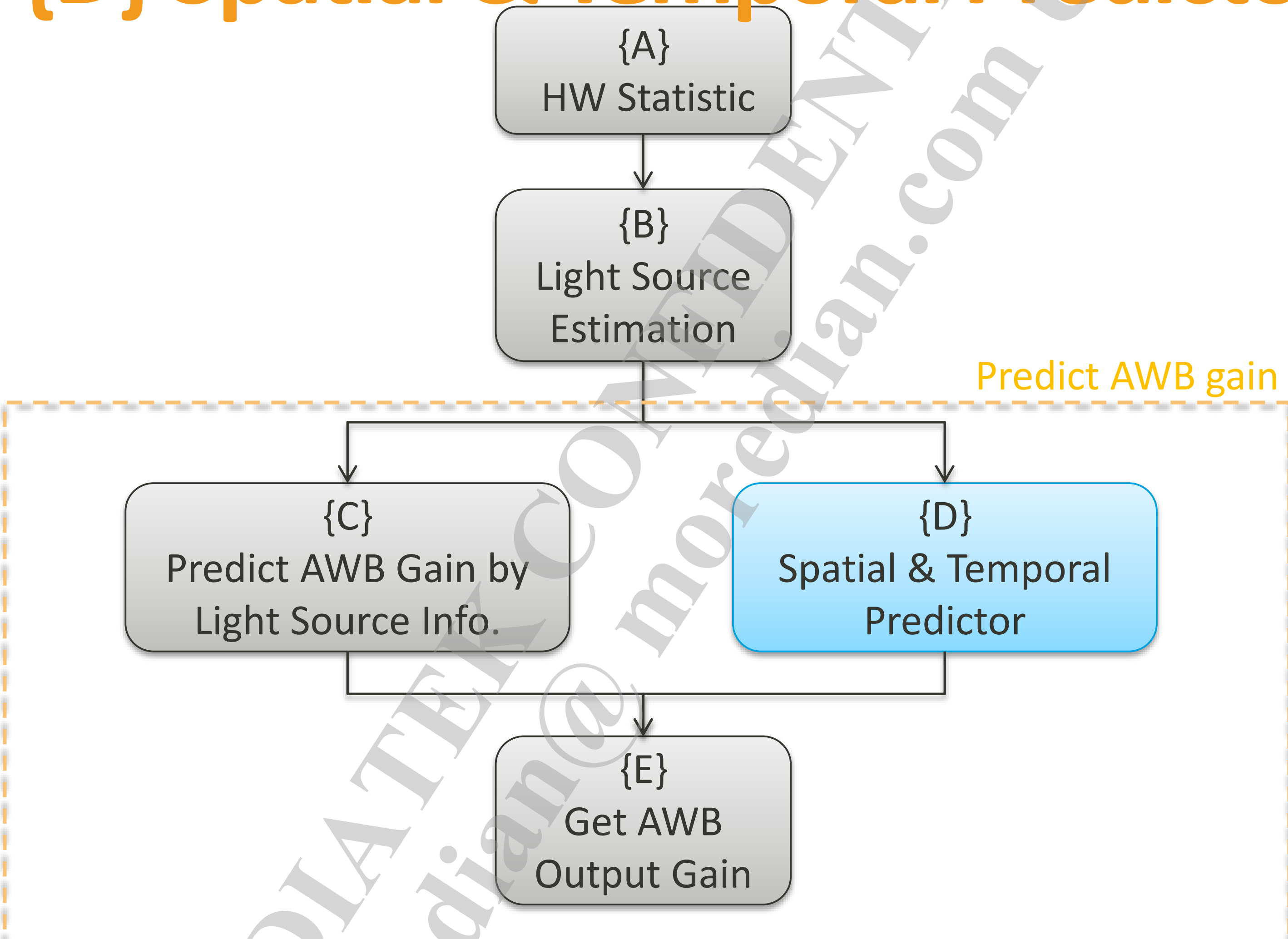



```
// rTempoSmooth
{
    10,    //i4Speed: Converge Speed 1~100
    2,    //i4MinStep: Minimum converge step
    1     //i4ProbReduceStep: If face gone, the face prob reduce step 1~100
},
```

Same with normal AWB temporal smooth mechanism

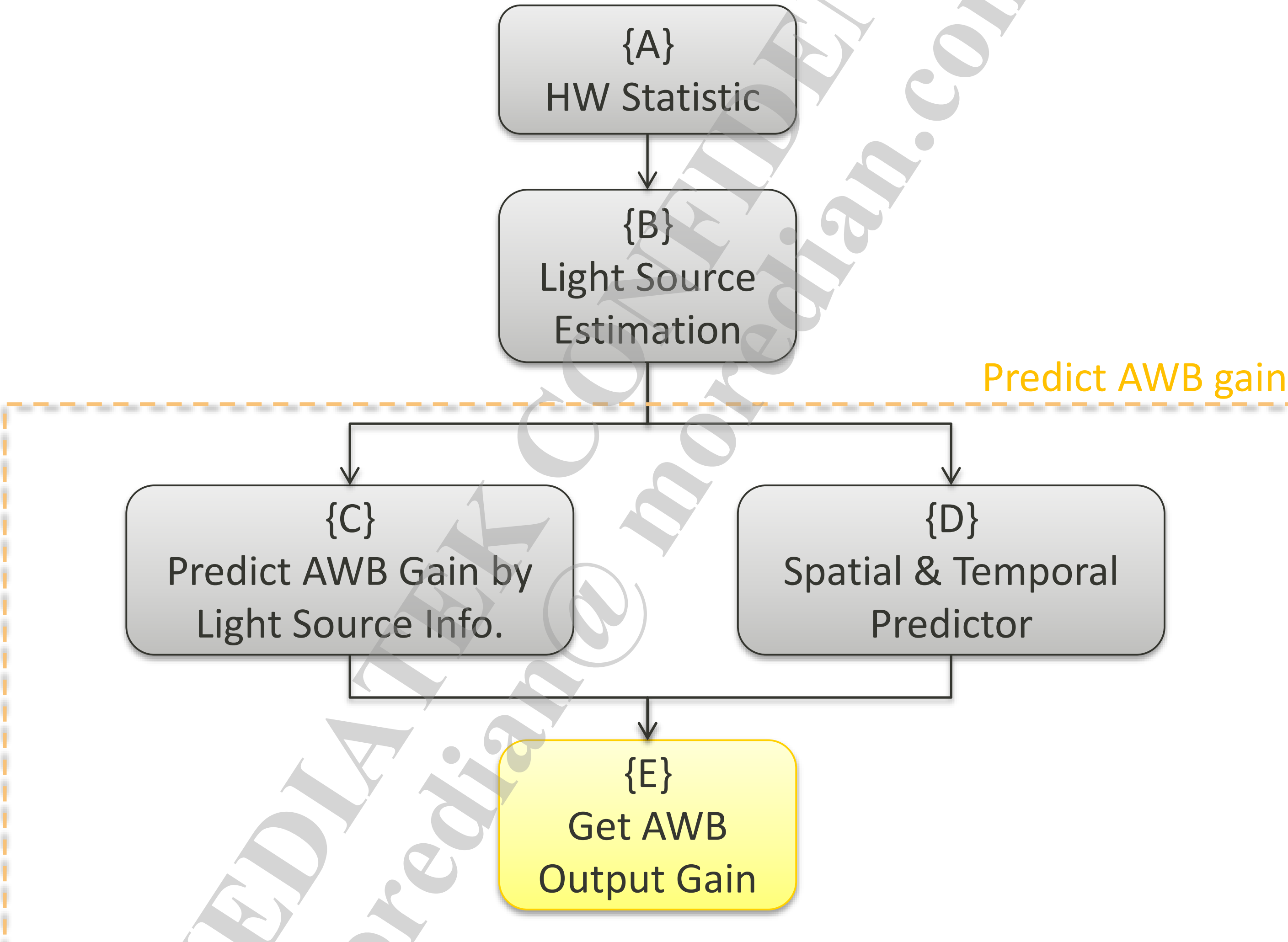
Back (Face-Assist)

{D} Spatial & Temporal Predictor

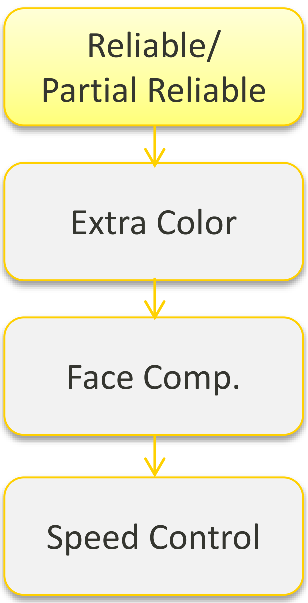


Spatial Predictor,Temporal Predictor,Mix Spatial & Temporal Predictor内容可参考：
MT6757_Camera_AWB_Tuning_Introduction.pdf P58-P61

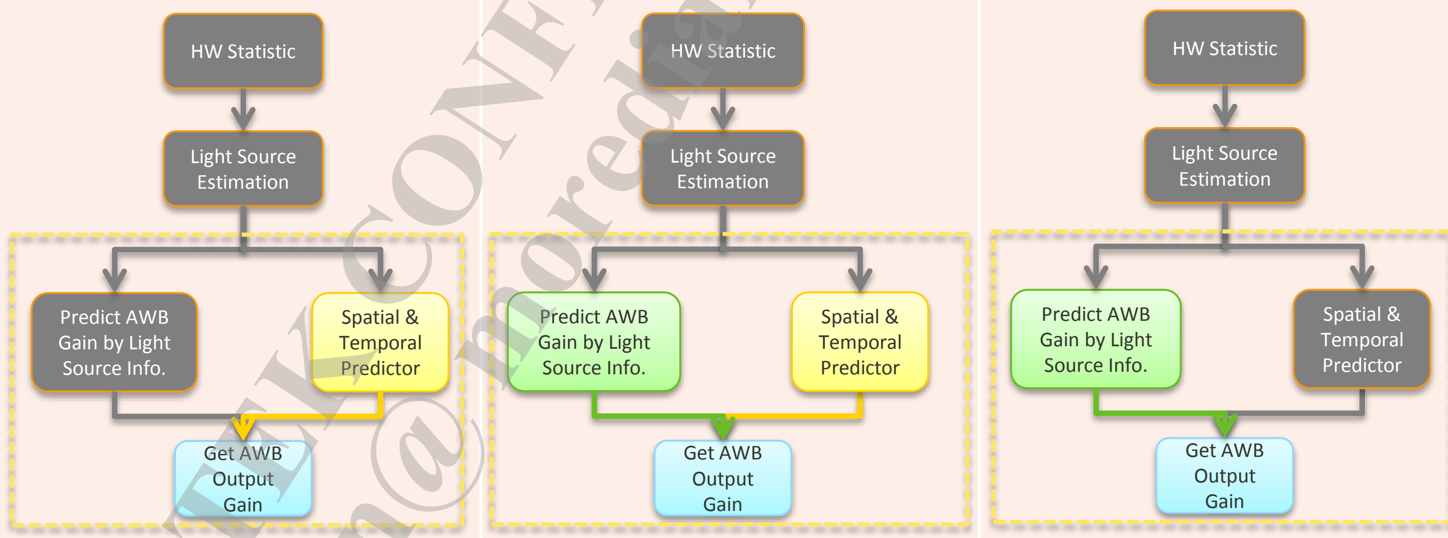
{E} Get AWB Output Gain

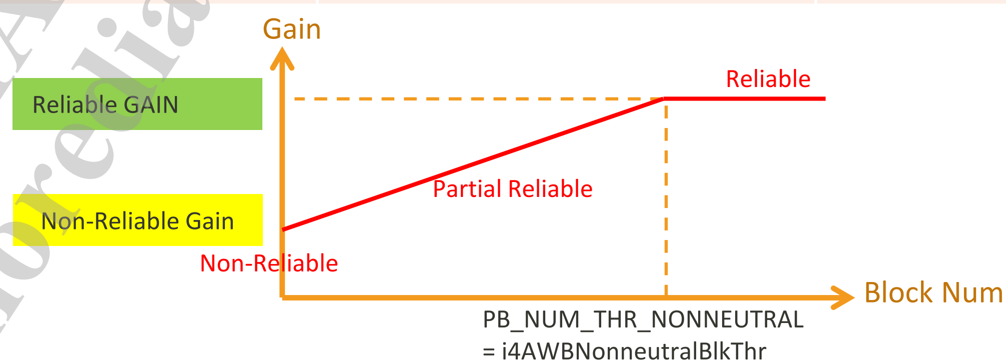


Reliable mode



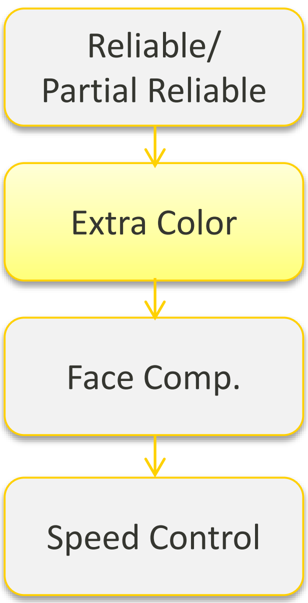
AWB_NO_ANY_RELIABLE_MODE	AWB_PARTIAL_RELIABLE_MODE	AWB_FULL_RELIABLE_MODE
NPB= 0	NPB < i4AWBNonneutralBlkThr	NPB >= i4AWBNonneutralBlkThr





AWB v5.0

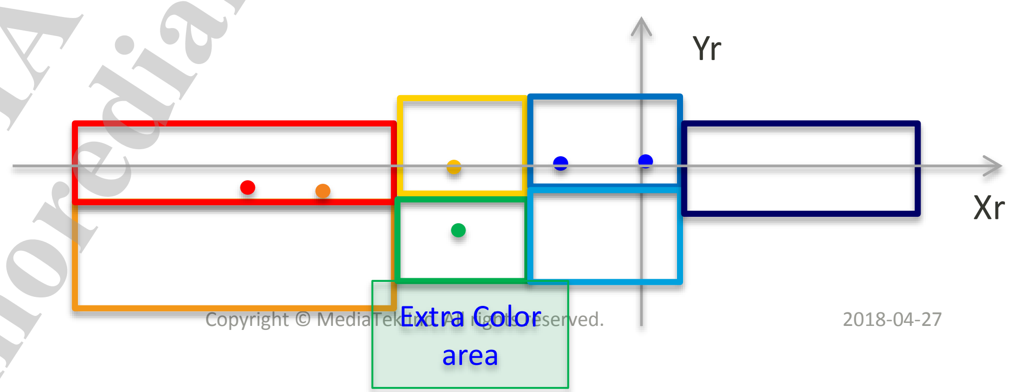
Extra Color Compensation (1/5)



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

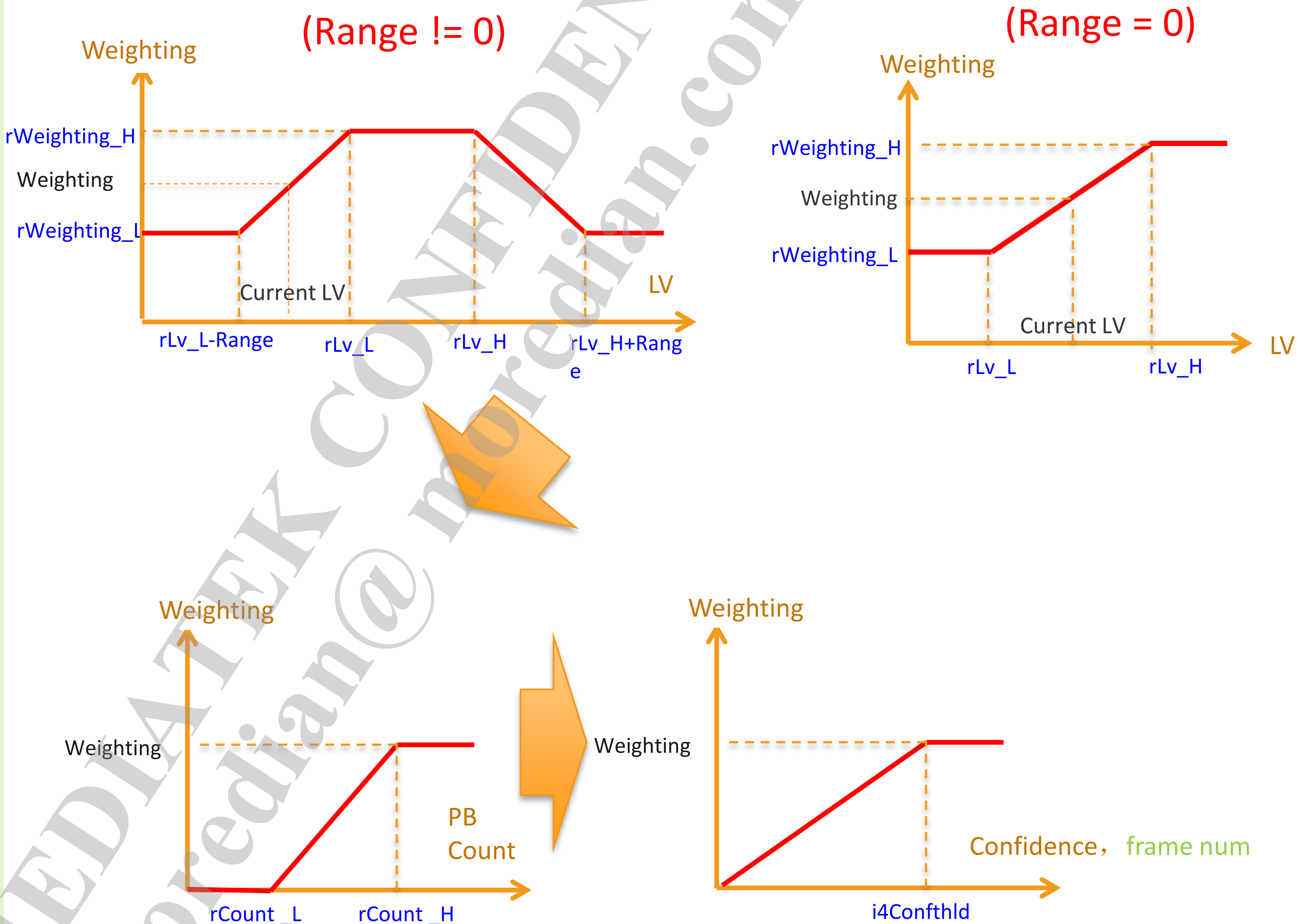


AWB v5.0

Extra Color Compensation (2/5)

```
// 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 color weight:



ExtraColor 數量比例越高,強度越強

AWB v5.0

Extra Color Compensation (3/5)

Param:

Reliable/ Partial Reliable

Extra Color

Face Comp.

Speed Control

```
// 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),可作用同时起

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
选择mode2和3 作用在Strobe,T,WF,F,CWF,D,Shade,DF中哪些光源上。

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
一般可通过量raw的感兴趣区域的G值来确定G Level范围。

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)

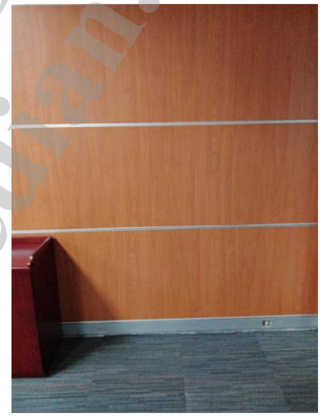
Example

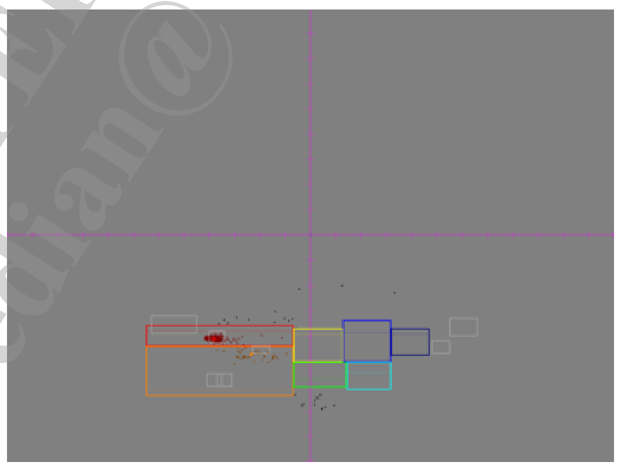
Reliable/
Partial Reliable

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

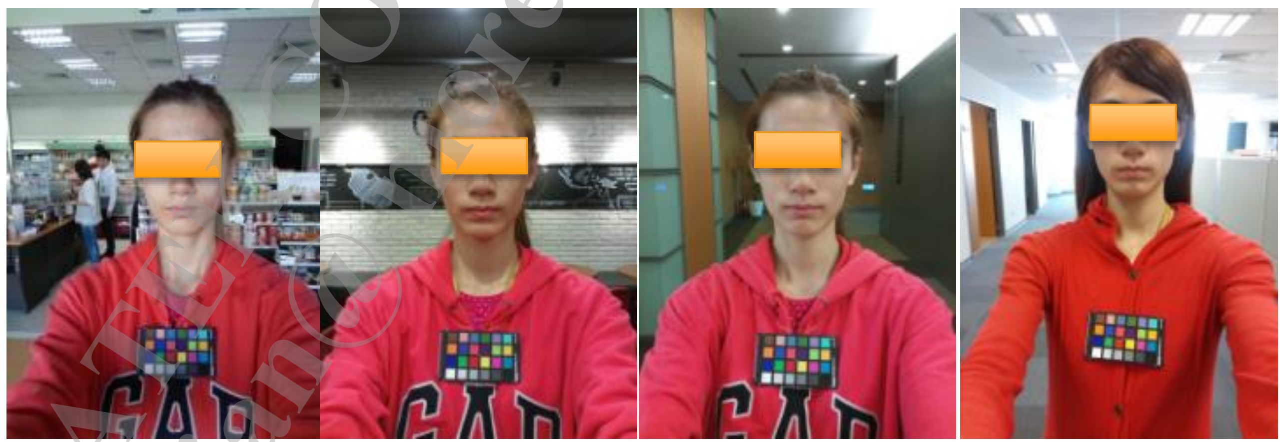
Extra Color

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)

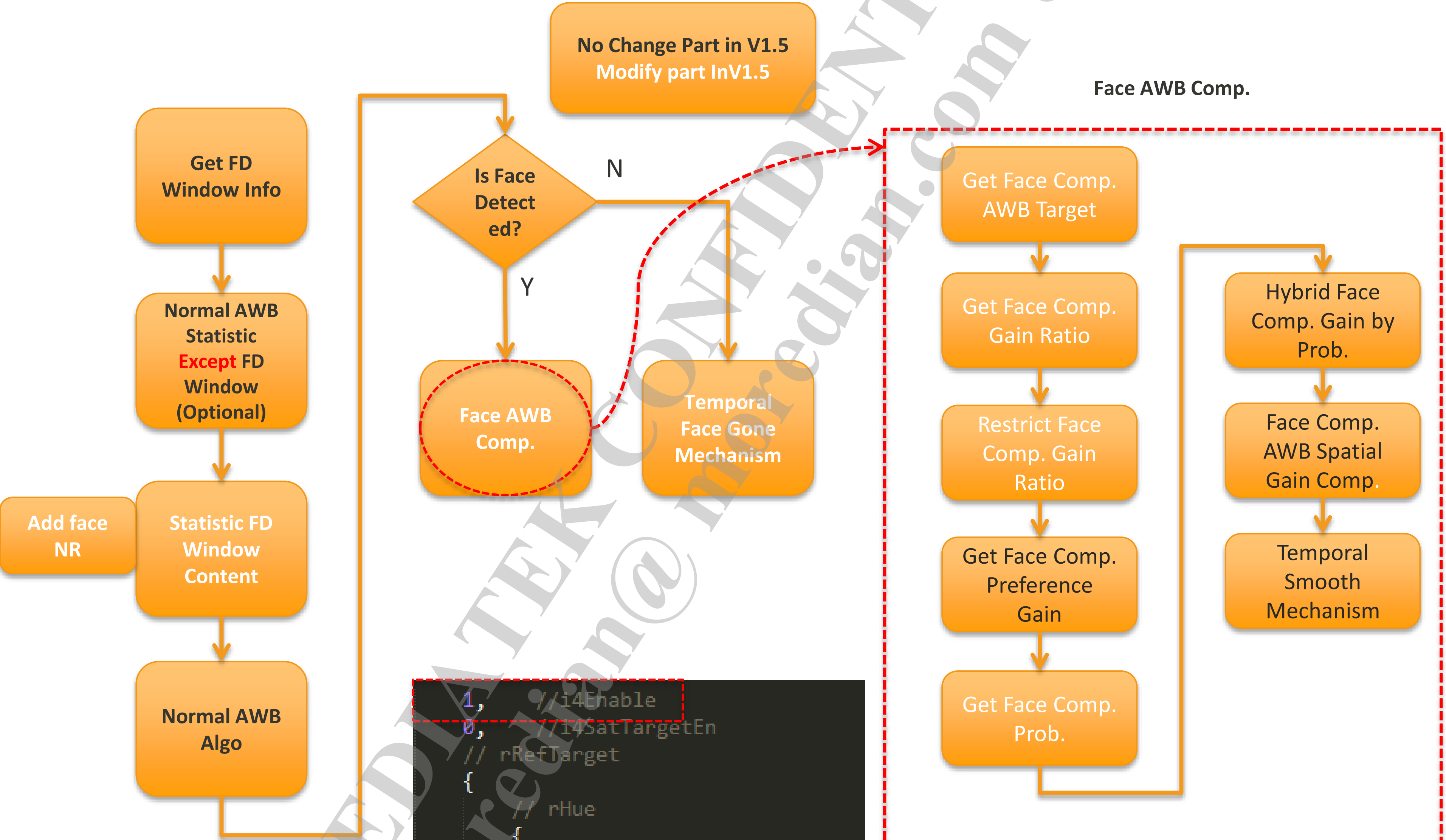


w/o face comp. AWB

Face comp. AWB可参考：

MT6757_Camera_AWB_Tuning_Introduction.pdf P67-P119

Face Comp. AWB v1.5 Algo Flow




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

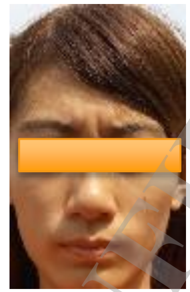
Face Comp. AWB Concept

Proc RAW domain

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

Normal AWB

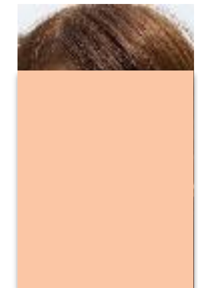
After Normal AWB



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

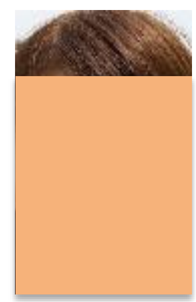
Face Comp. Ratio

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

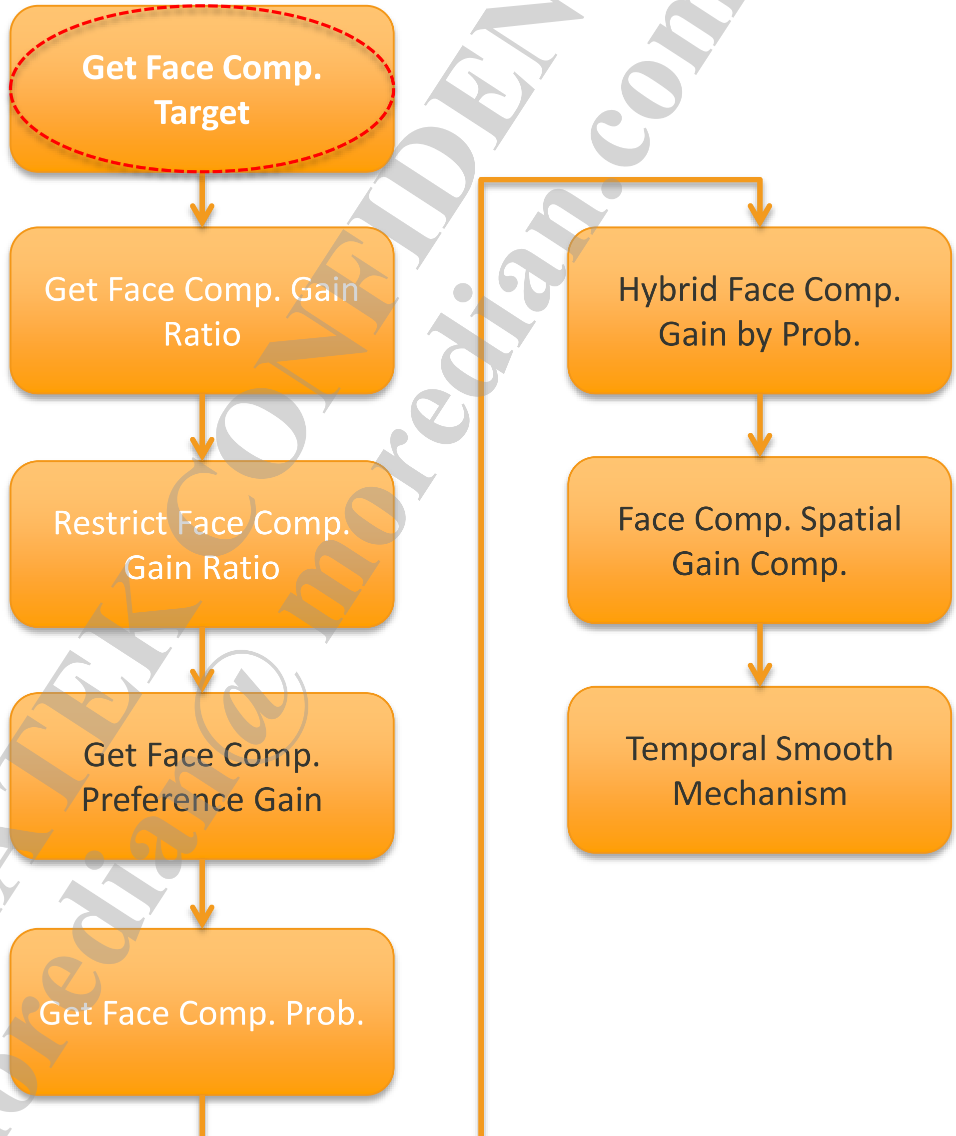
Face Comp. AWB v1.0 Comp.

Assign {R/G,B/G},algo will change to hue

Use outdoor Prob for saturation control

Do restriction after outdoor prob hybrid 2 gain ratio

P1 Only has LV index



Face Comp. AWB v1.5 Comp.

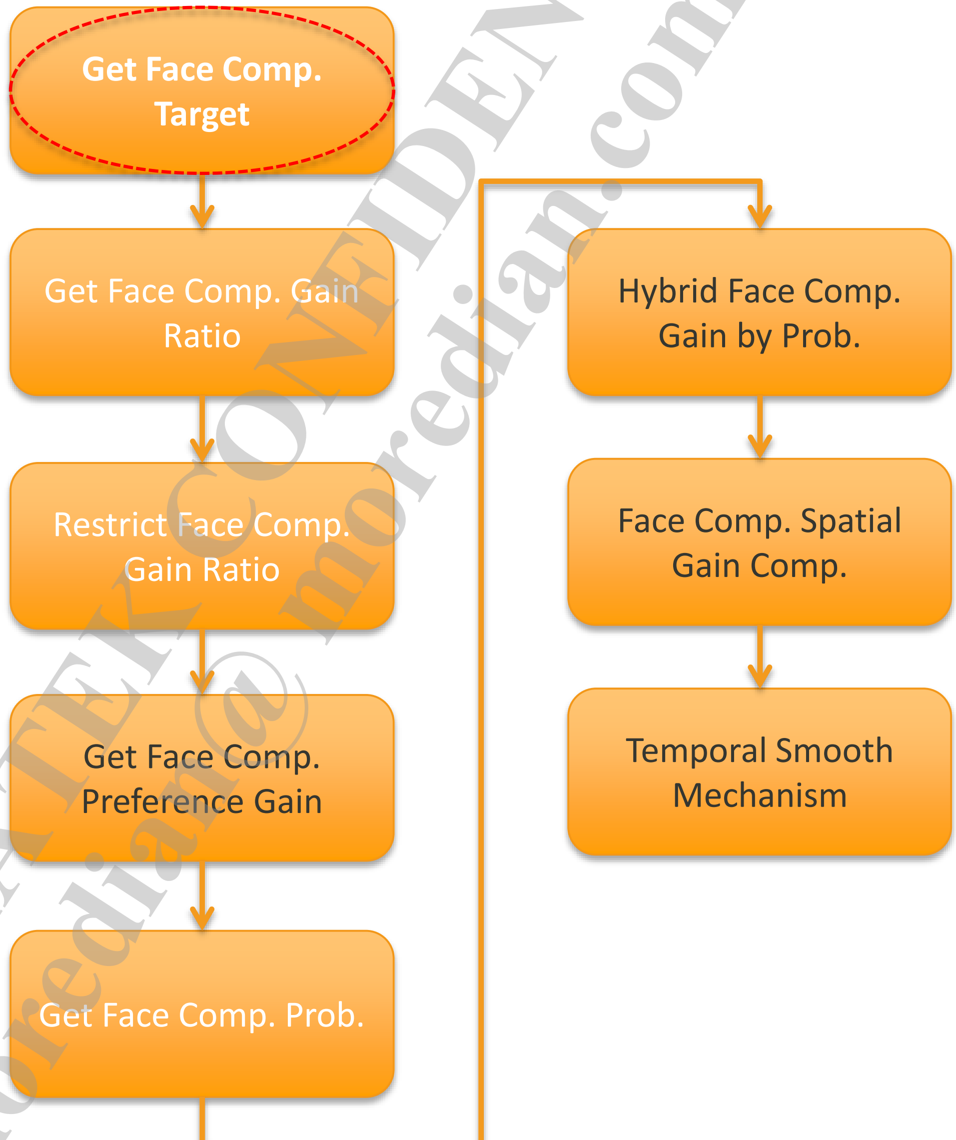
Algo will do source face data temporal smooth

Assign {Hue,Saturation,tolerance}, can refer normal awb Info for choosing target

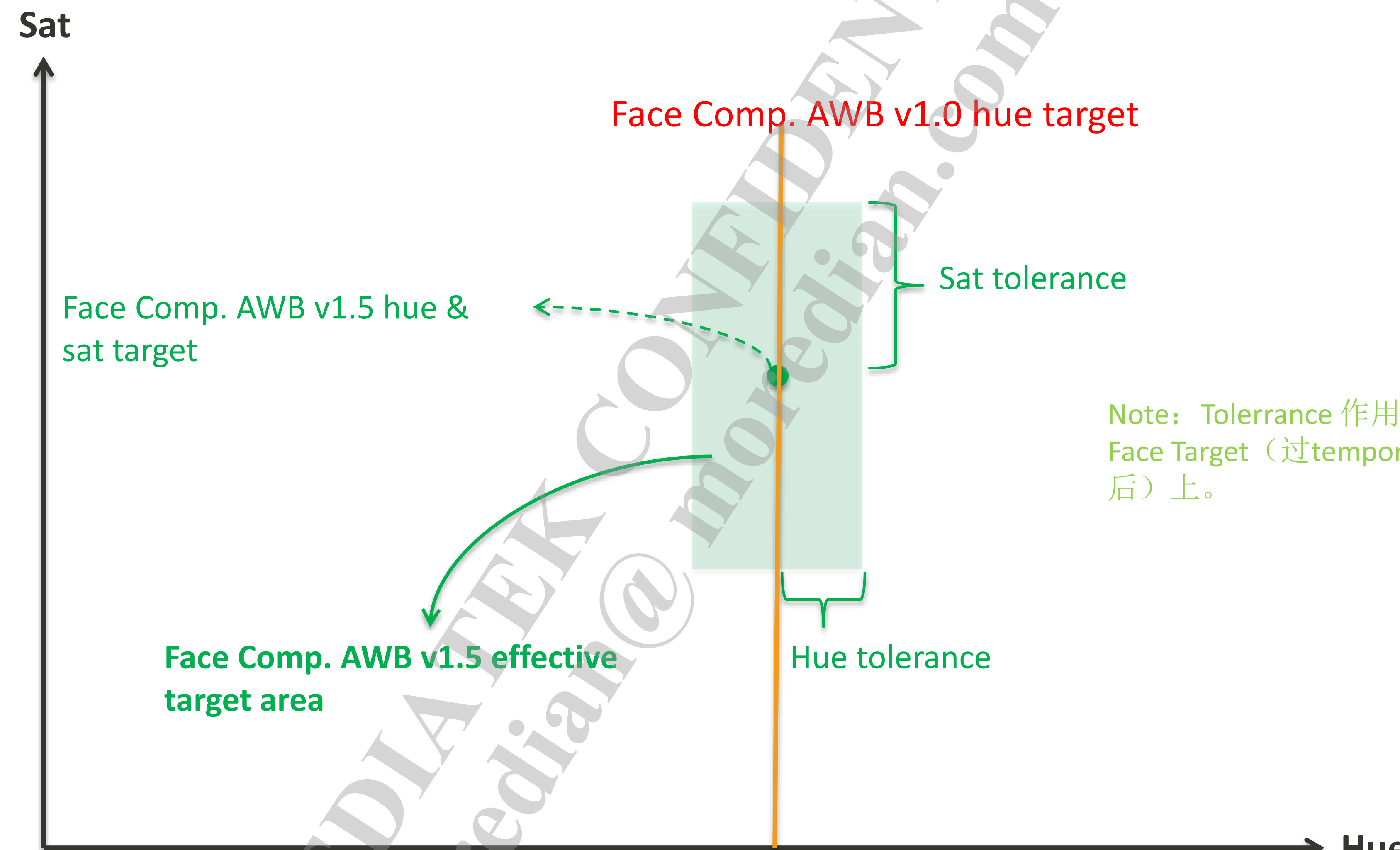
Use oversat Prob for saturation control

Do restriction before oversat prob hybrid 2 gain ratio

P1 has LV&CT info as index
Add P4 for face luma index



Face Comp. AWB Target Definition

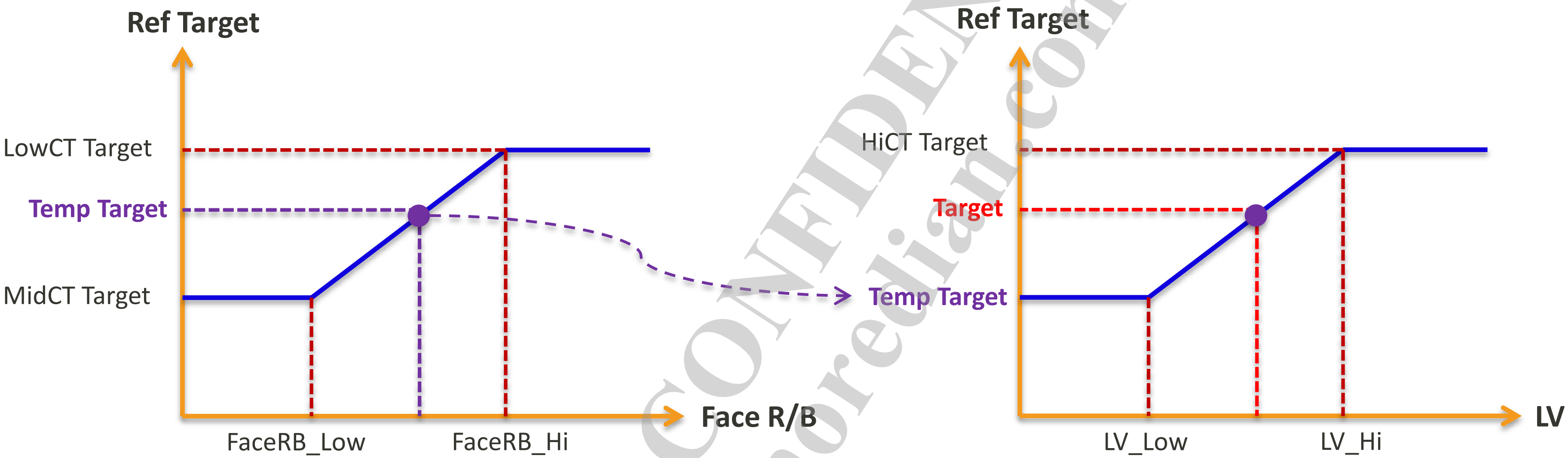


Note：Tolerrance 作用在Final Face Target（过temporal机制后）上。

Tolerance param

```
// 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%,
},
```

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
},
```

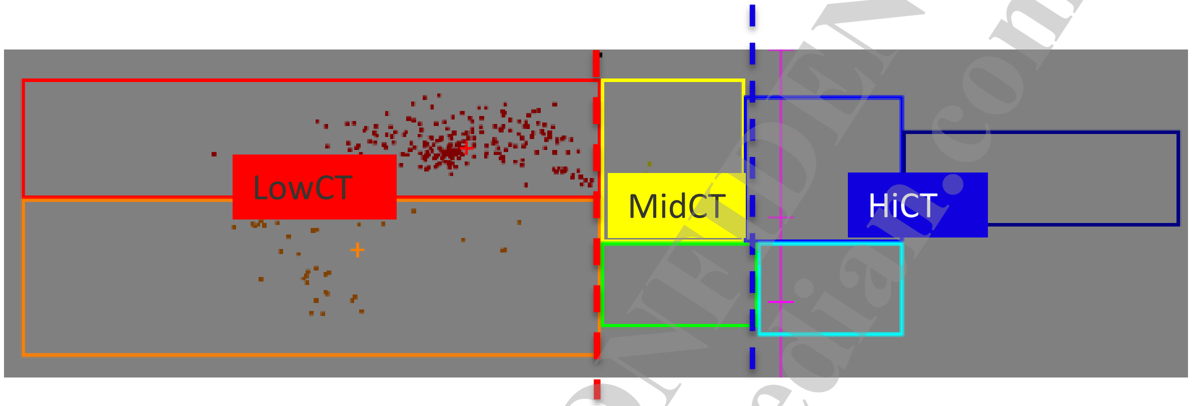
Use saturation target

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

根据HiCT_P, MidCT_P, LowCT_P的值去混设定的三个hue/sat target.

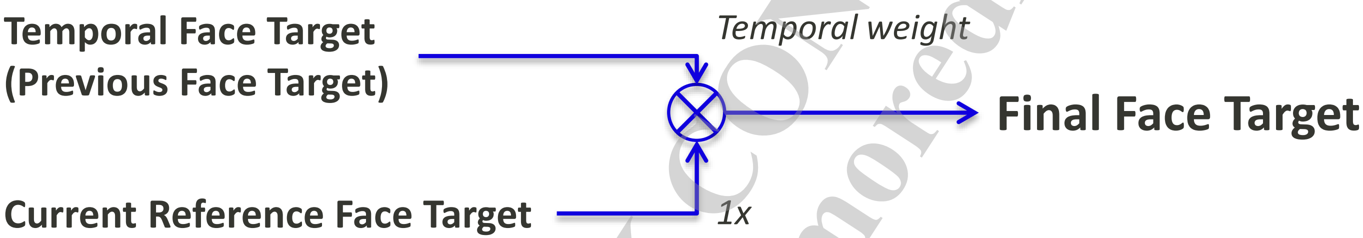
$$TempTarget = \frac{LowTarget \times LowCT\_P + MidTarget \times MidCT\_P}{LowCT\_P + MidCT\_P}$$

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

When face R/B thr or LV thr of scene judge 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 v1.5 Comp.

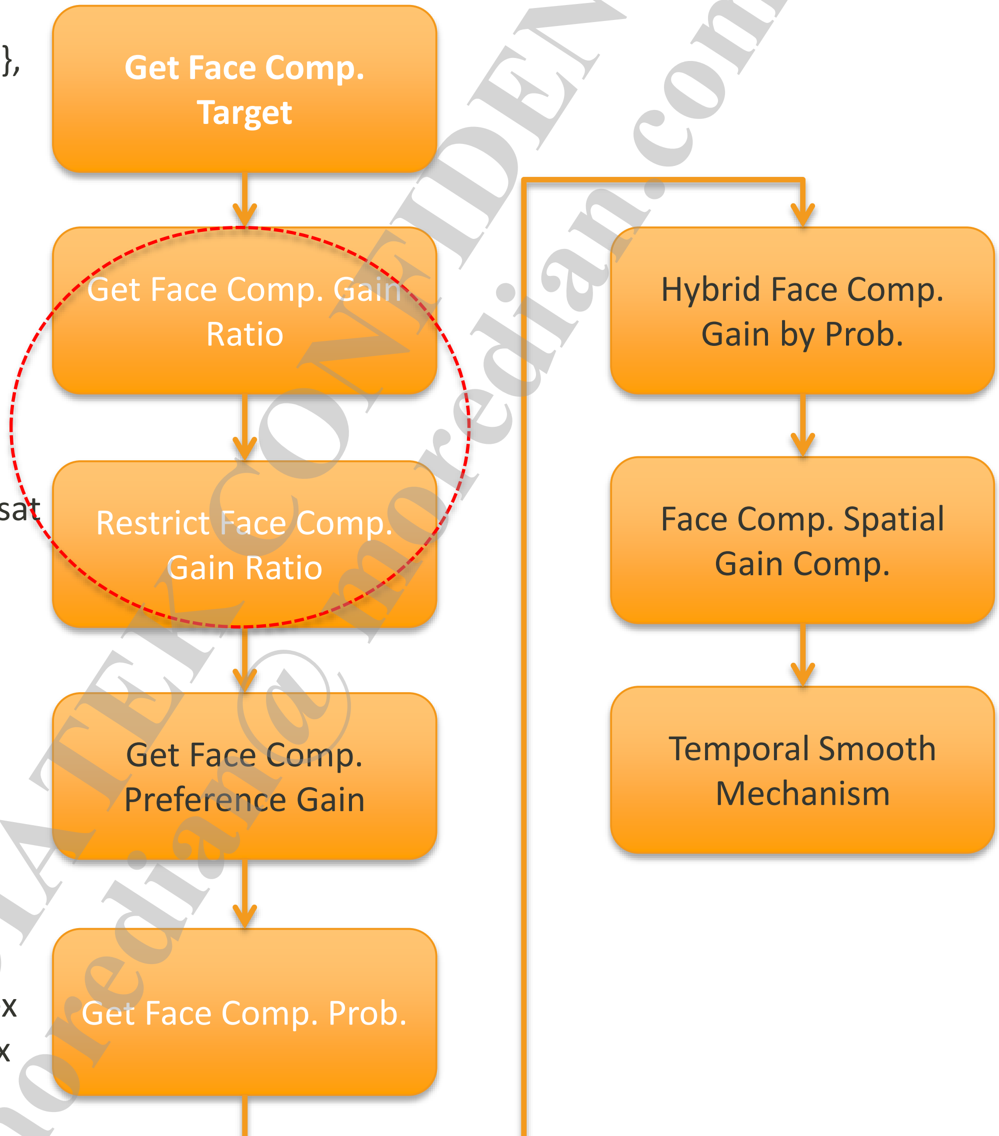
Algo will do source face data temporal smooth

Assign {Hue,Saturation,tolerance}, can refer normal awb Info for choosing target

Use oversat Prob for saturation control

Do restriction before oversat prob hybrid 2 gain ratio

P1 has LV&CT info as index
Add P4 for face luma index



Get Face Comp. AWB Gain Ratio

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

Face Comp. AWB v1.0

Get N1 & N2 GainRatio → Get outdoor Prob → Hybrid N1 & N2 by outdoor Prob → Restrict GainRatio

Face Comp. AWB v1.5

Get N1 & N2 GainRatio → Restrict N1 & N2 GainRatio → Get Oversat Prob → Hybrid Restricted N1 & N2 by Oversat Prob

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, 50, 25, 10,  0,  0},
```

Increase calculation & tuning precision, unit = 0.1%

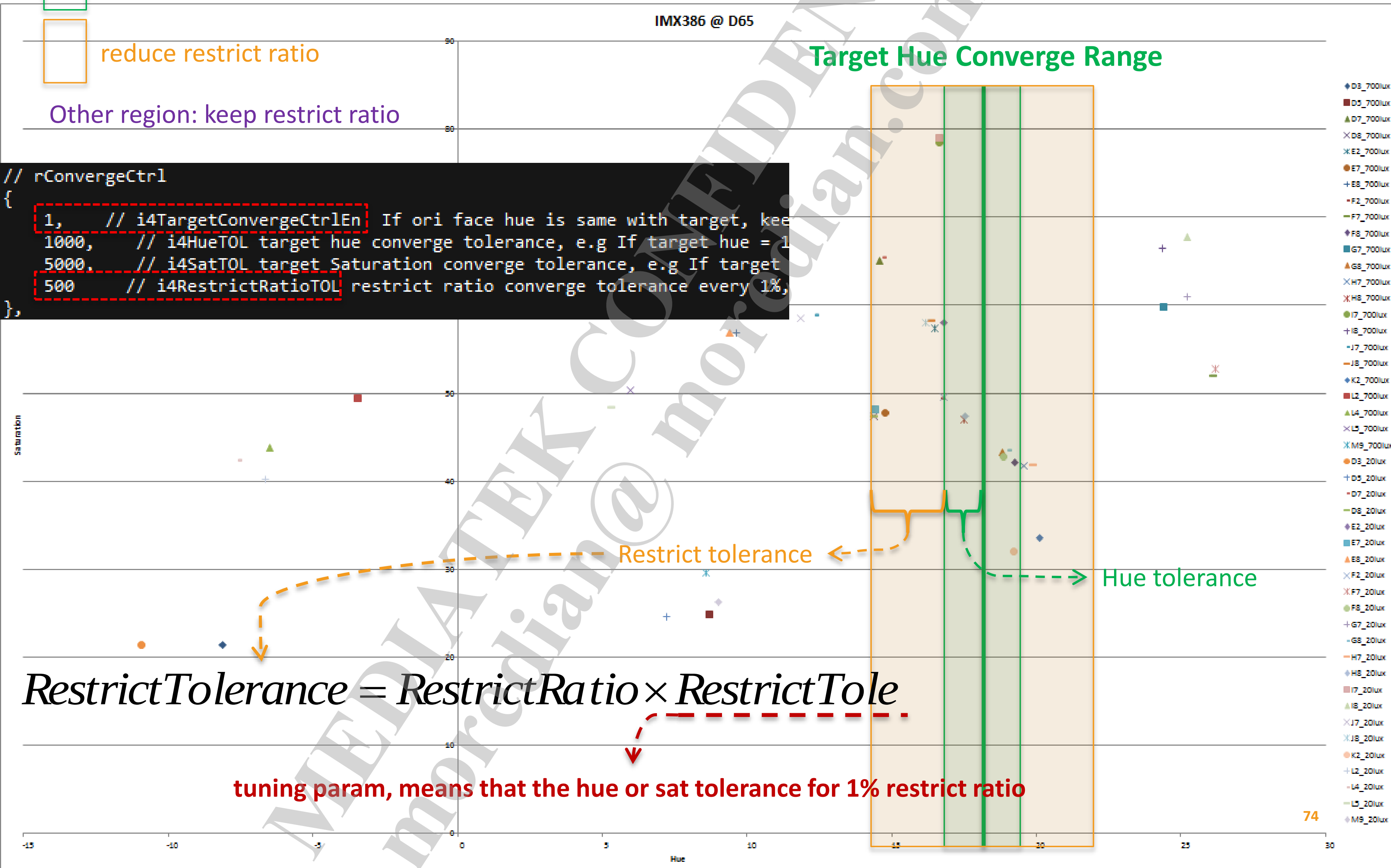
Face Comp. AWB v1.5 Algorithm Target Converge Control

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

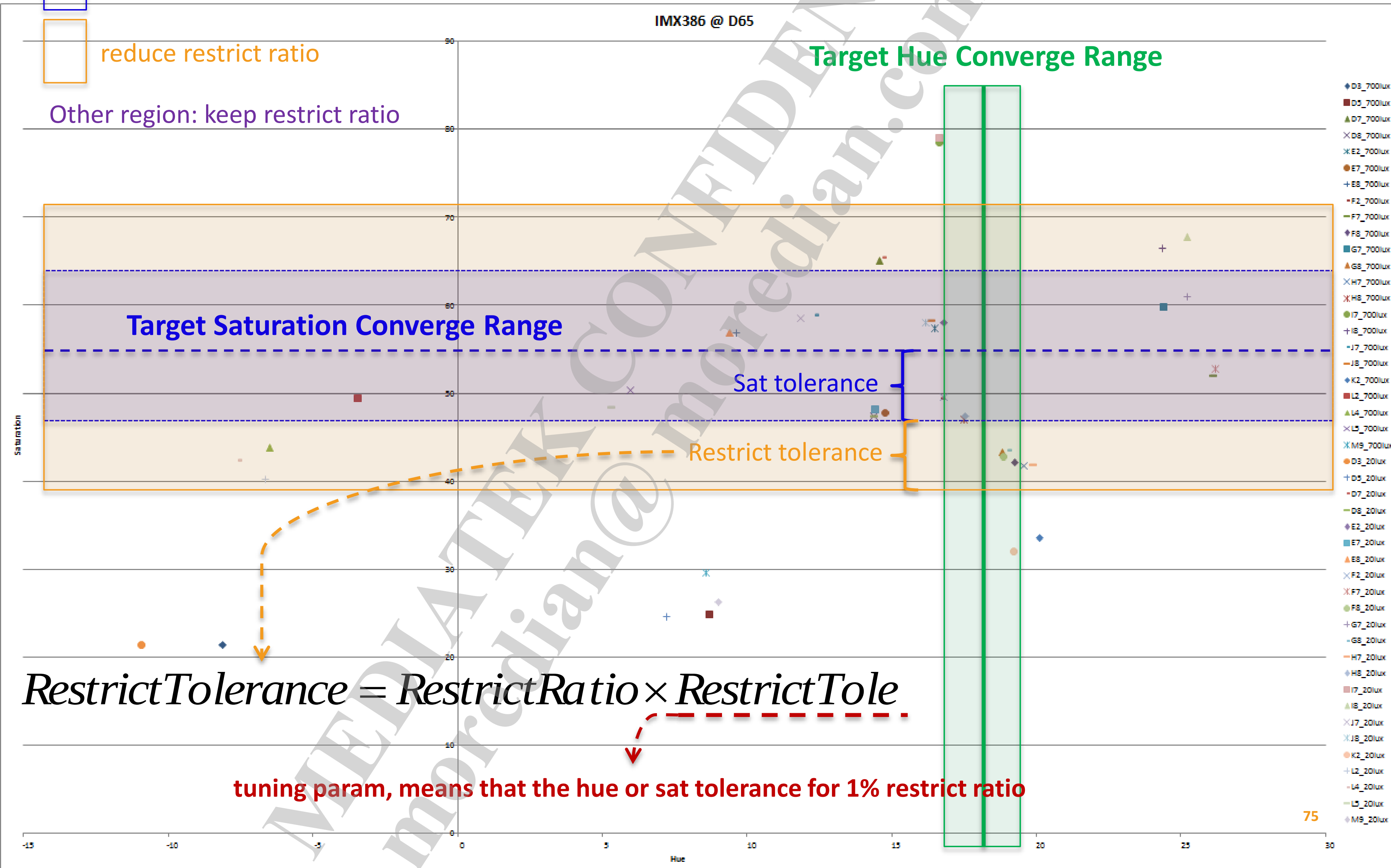
Face Comp. AWB v1.5 Algorithm Target Converge Control

Sat target enable!

restrict ratio = 0

reduce restrict ratio

Other region: keep restrict ratio



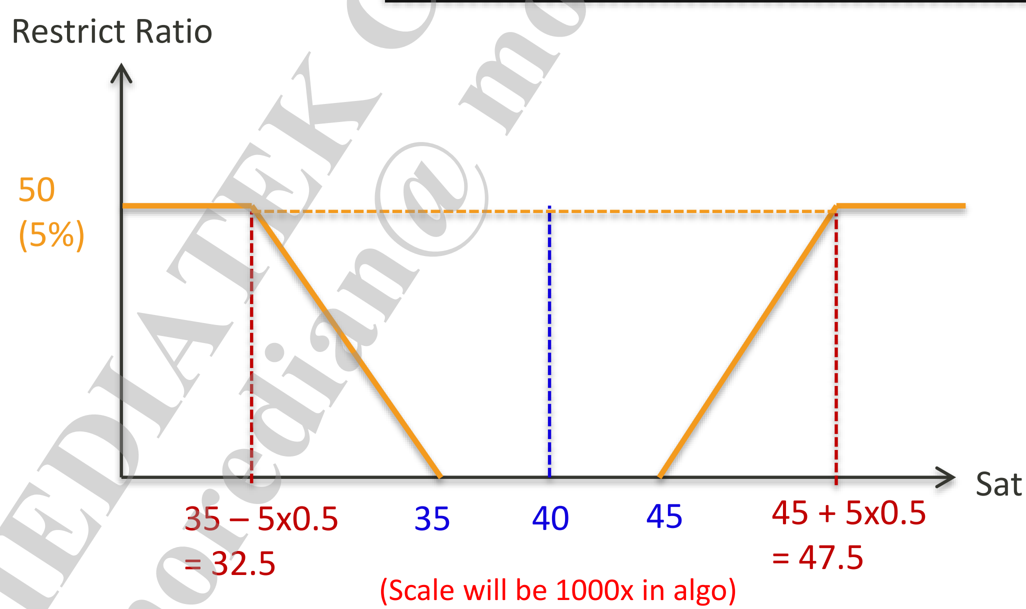
$$RestrictTolerance = RestrictRatio \times RestrictTole$$

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

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

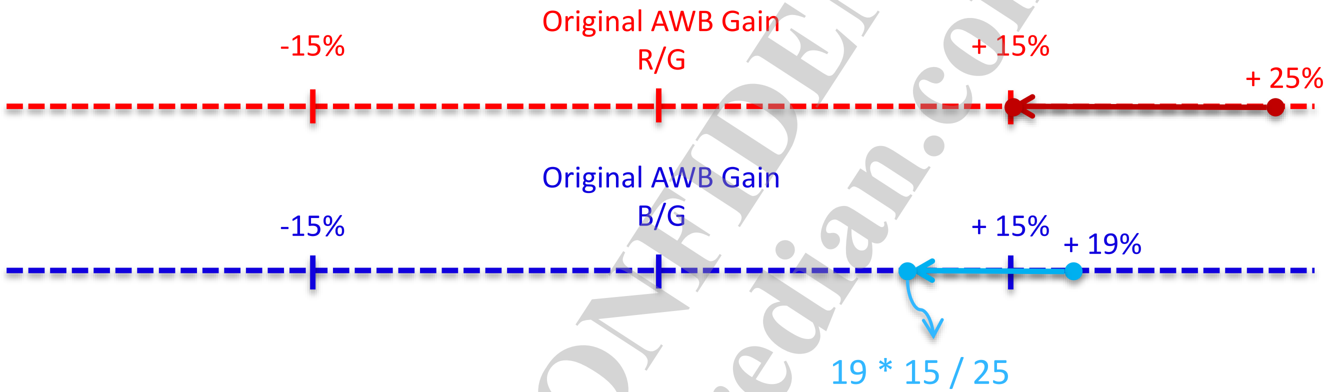
e.g.
Target sat = 40000(unit1000,mean 40)
Sat TOL = 5000(unit 1000.mean 5)
Restrict Ratio = 50 (means 5%)
Restrict TOL = 500(unit1000,mean 0.5)

```
//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, 50, 25, 10,  0,  0},
```

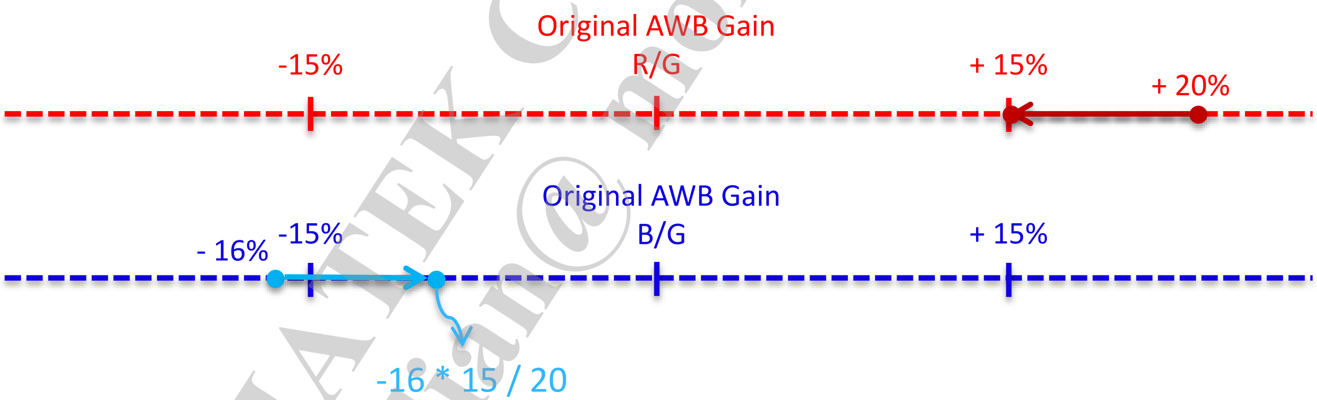


Face Comp. AWB Gain Ratio Restrict Mechanism

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.

Face Comp. AWB v1.0

Get N1 & N2 GainRatio → Get outdoor Prob → Hybrid N1 & N2 by outdoor Prob → Restrict GainRatio

Face Comp. AWB v1.5

Get N1 & N2 GainRatio → Restrict N1 & N2 GainRatio → Get Oversat Prob → Hybrid Restricted N1 & N2 by Oversat Prob

Get Face Comp. AWB Gain Ratio

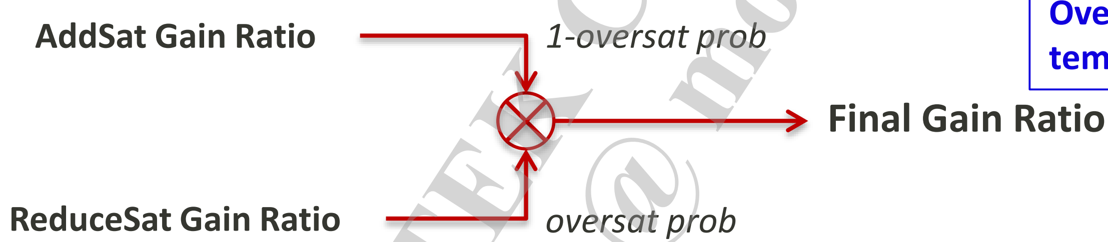
$$\begin{cases} AddSatGainRatio\_R \\ AddSatGainRatio\_B \end{cases}$$

{R+, B-}

$$\begin{cases} ReduceSatGainRatio\_R \\ ReduceSatGainRatio\_B \end{cases}$$

{R-, B+}

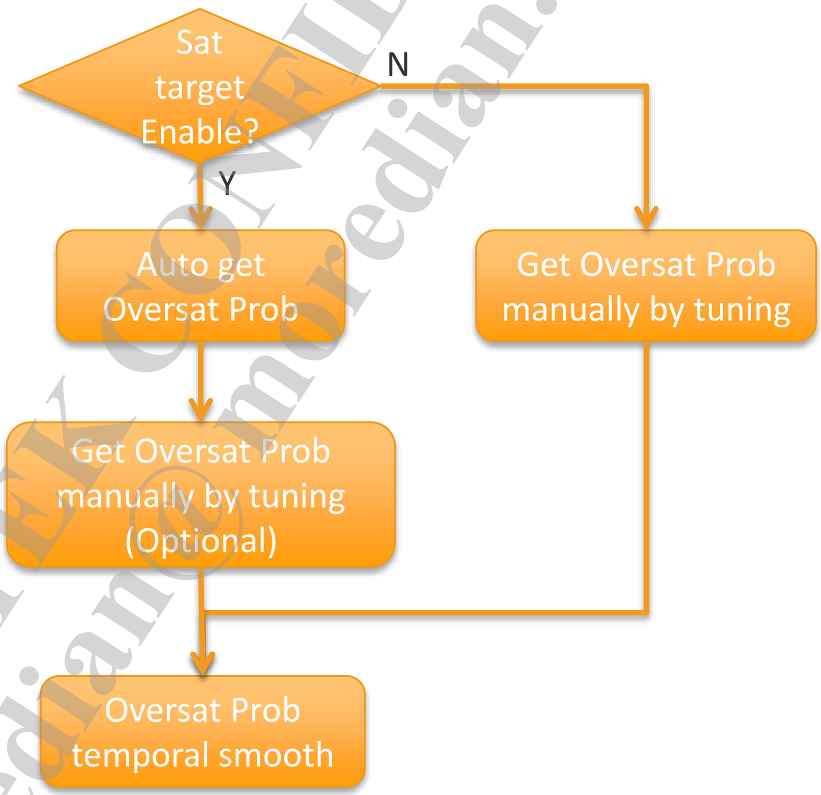
How to get oversat prob?



Oversat prob will also have a temporal smooth.

$$FinalGainRatio = \frac{\left(AddSatGainRatio \times \left(100 - Oversat\_Prob\right)\right) + \left(ReduceSatGainRatio \times OversatProb\right)}{100}$$

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.**

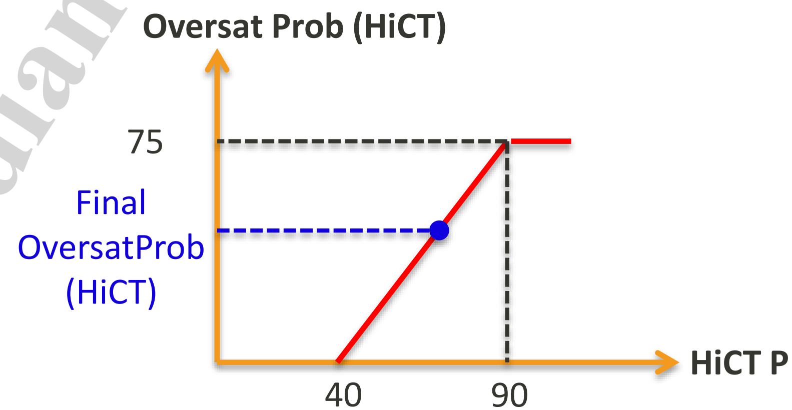
```
// rOversat
{
    // rLV
    {
        90,    //i4LVLow
        120,    //i4LVHi
        0,    //i4OverSatProb_Low
        100     //i4OverSatProb_Hi
    },
    // rHiCT
    {
        40,    //i4CT_P_Low
        90,    //i4CT_P_Hi
        0,    //i4OverSatProb_Low
        75     //i4OverSatProb_Hi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



HiCT_P = Daylight_Prob + DF_Prob + Shade_Prob

MidCT_P = Fluorescent_Prob + CWF_Prob

LowCT_P = Tungsten_Prob + WF_Prob

Sat target disable时，会根据左边设定参数计算出四个prob，取最大的那个作为Oversat_Prob。

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

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
    }
},
```

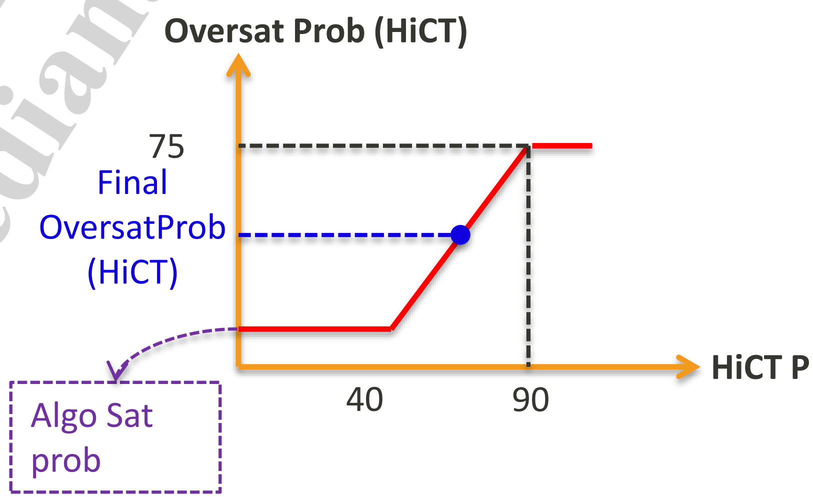
used for indoor & outdoor judgment

used for indoor high CT scene

No use if sat target enable

used for indoor middle CT scene

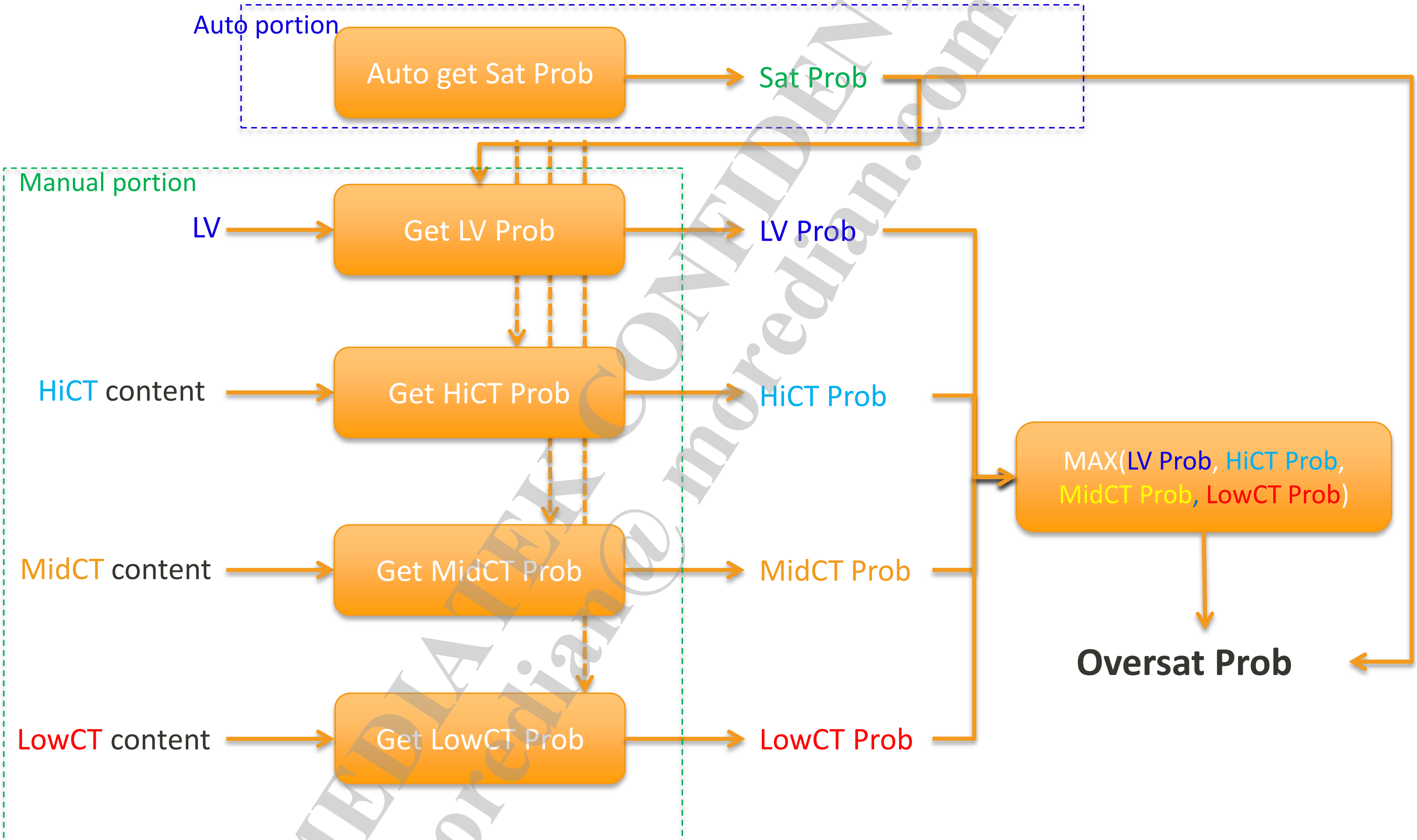
used for indoor low CT scene



Sat target enable时，左边参数全部设为0就会用algo算出来的SatProb作为Oversat_Prob.如果i4OverSatProb_Hi高于Algo Sat prob时，则会按上图插值出finalOverSatProb。

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

Get Oversat Prob



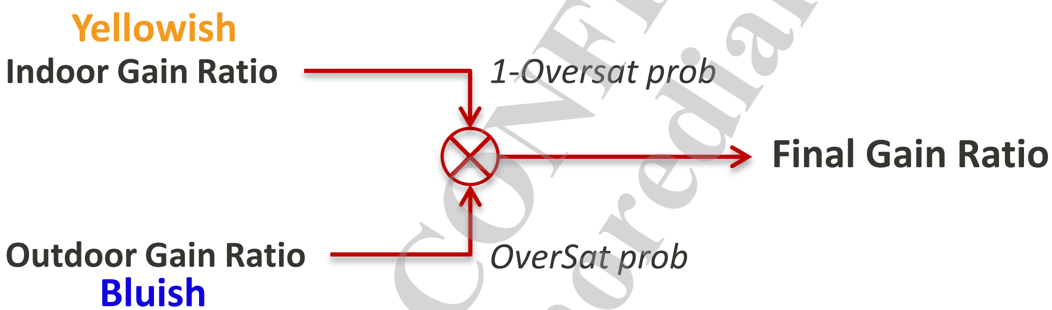
Get Face Comp. AWB Gain Ratio

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

Get N1 & N2 GainRatio → Restrict N1 & N2 GainRatio → Get Oversat Prob → Hybrid Restricted N1 & N2 by Oversat Prob

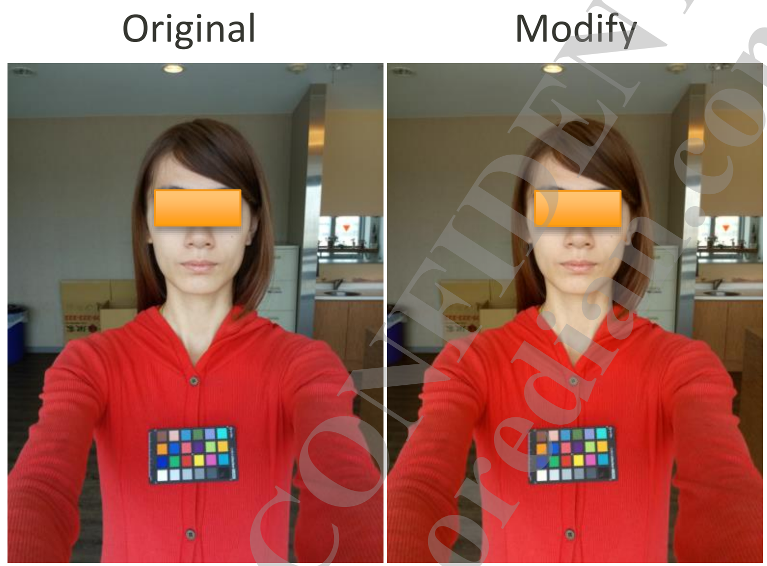
Get Face Comp. AWB Gain Ratio

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

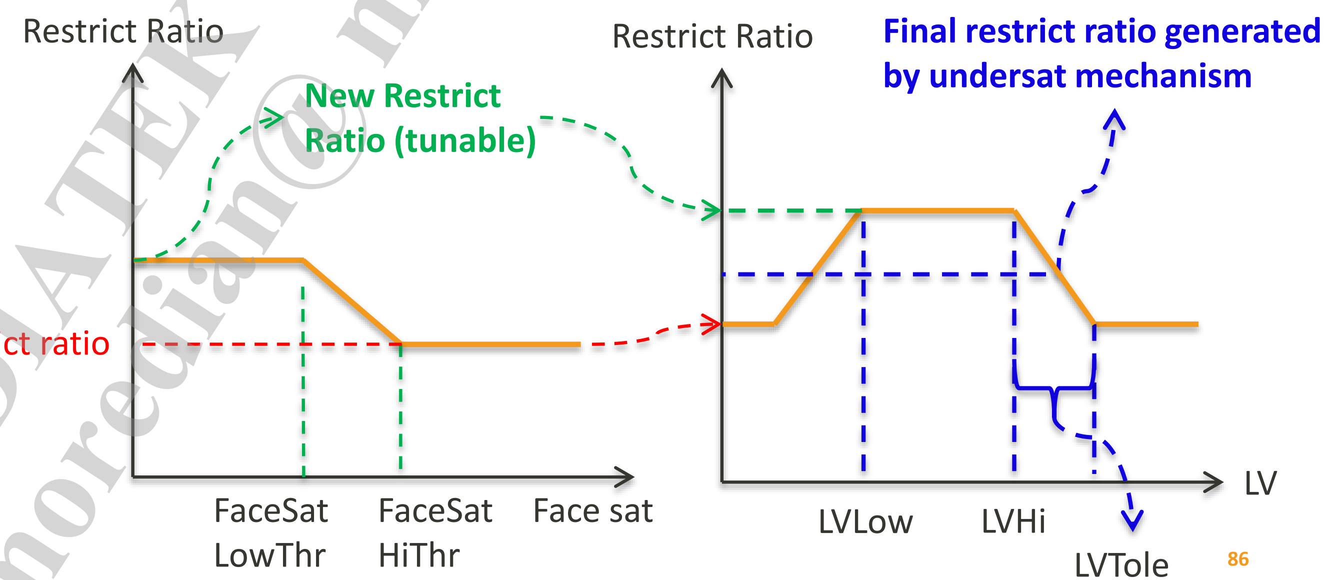


$$FinalGainRatio = \frac{\left(IndoorGainRatio \times \left(100 - Oversat\_Prob\right)\right) + \left(OutdoorGainRatio \times Over\right.}{100}$$

Undersat Prevention Mechanism (1/2)



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

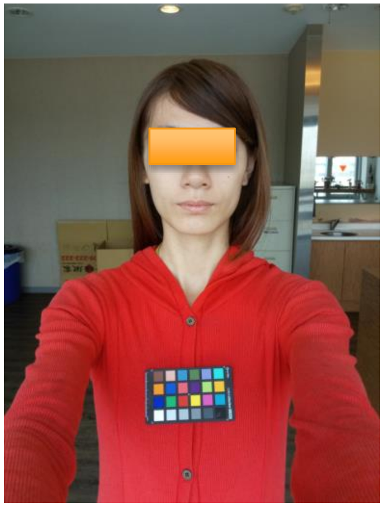


1、增加Restrict Ratio

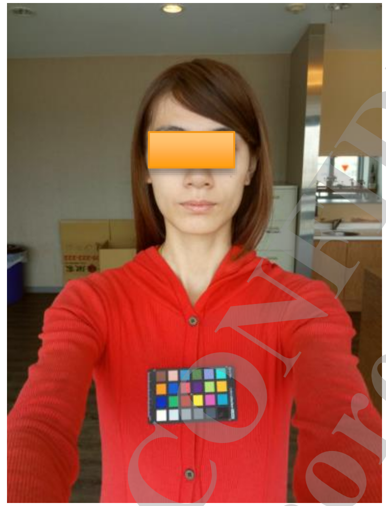
```
// rUndersat
{
    10000,    // i4SatHi
     5000,    // i4SatLow
       75,    // i4NewRestrict
       25,      // i4LVLow
       95,      // i4LVHi
       10         // i4LVTOL
},
```

Undersat Prevention Mechanism (2/2)

Original

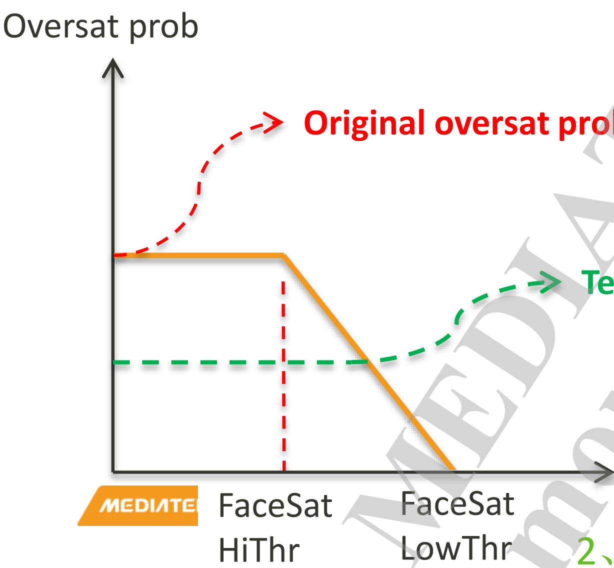


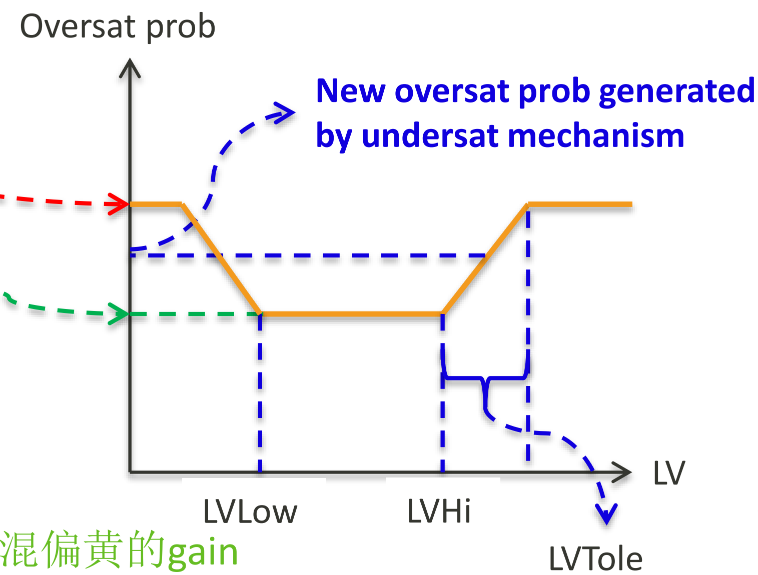
Modify



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
},
```





2、降低 OverSatProb,多混偏黄的gain

Face Comp. AWB v1.5 Comp.

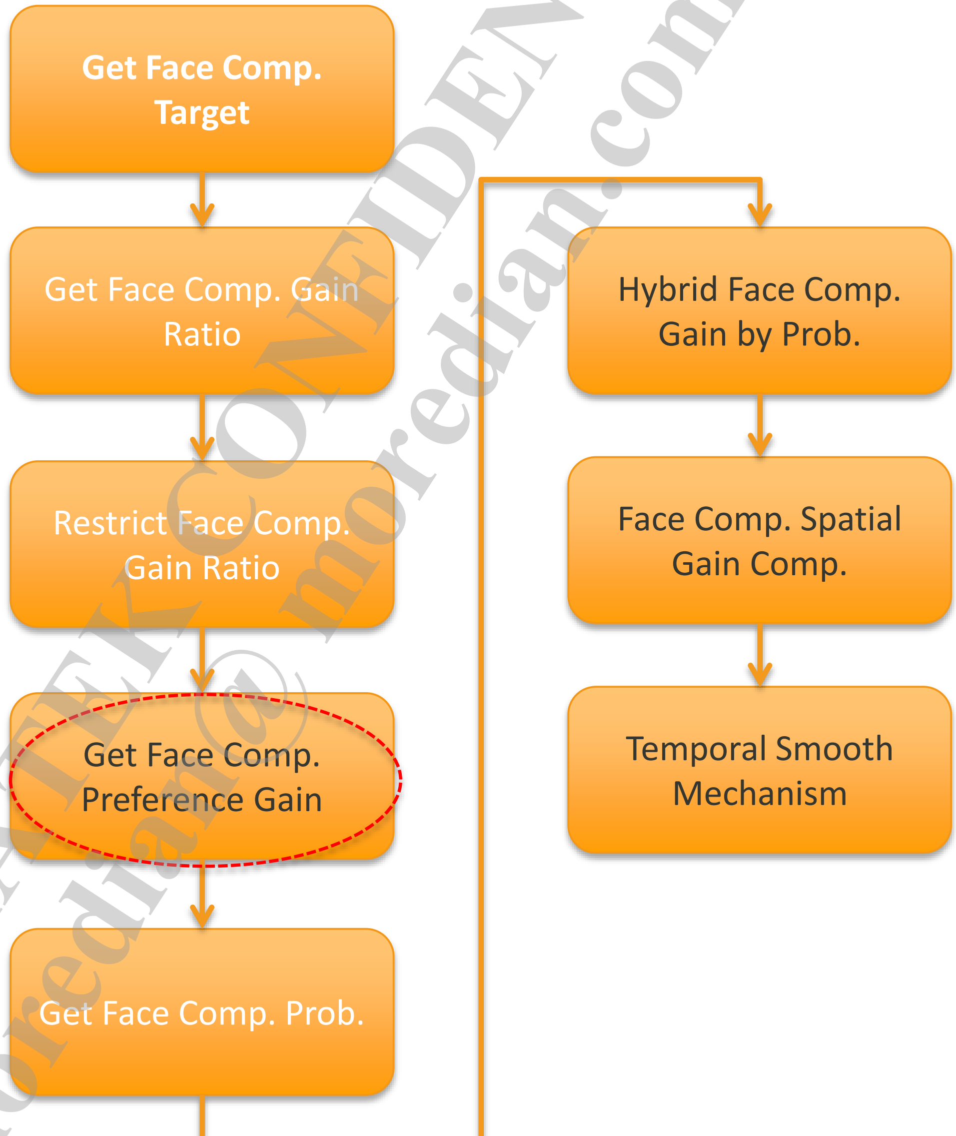
Algo will do source face data temporal smooth

Assign {Hue,Saturation,tolerance}, can refer normal awb Info for choosing target

Use oversat Prob for saturation control

Do restriction before oversat prob hybrid 2 gain ratio

P1 has LV&CT info as index
Add P4 for face luma index



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.

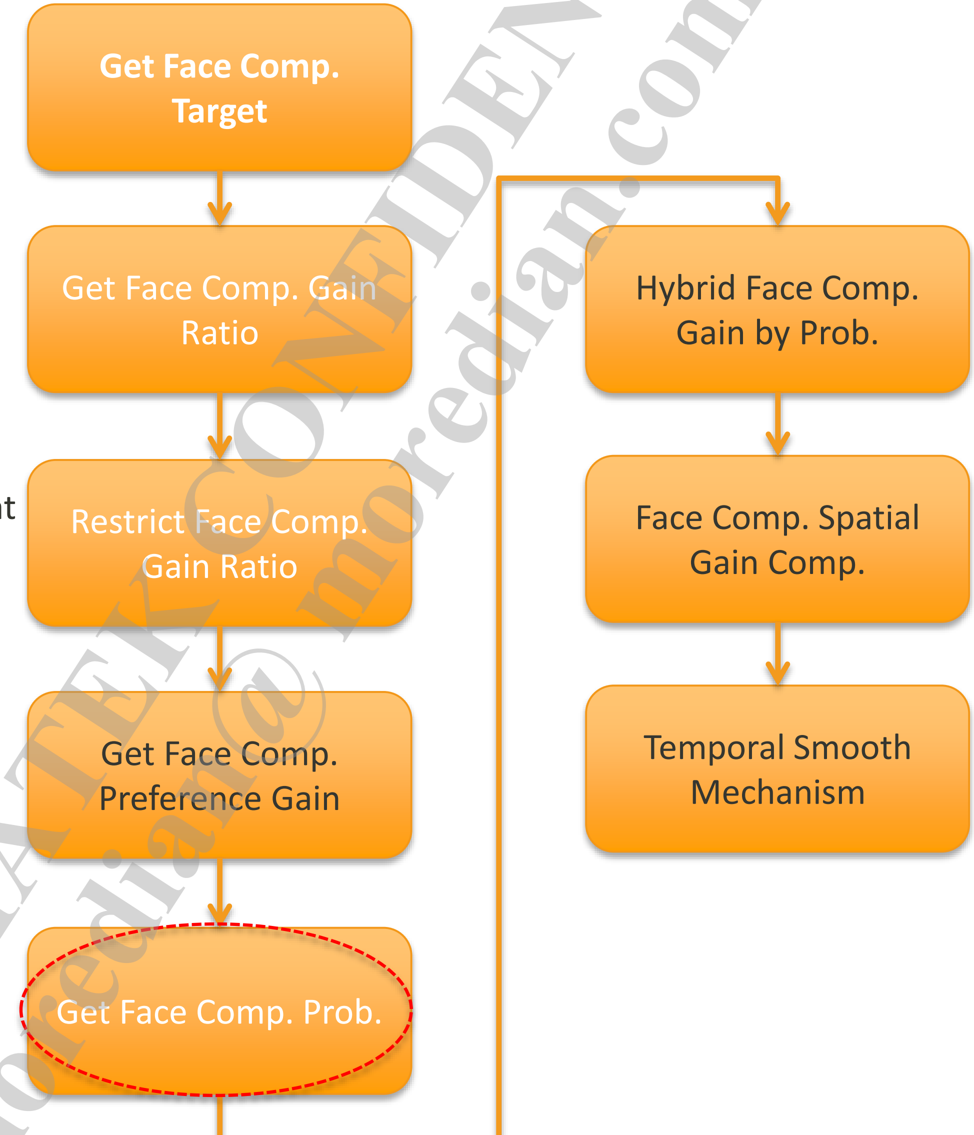
Algo will do source face data temporal smooth

Assign {Hue,Saturation,tolerance}, can refer normal awb Info for choosing target

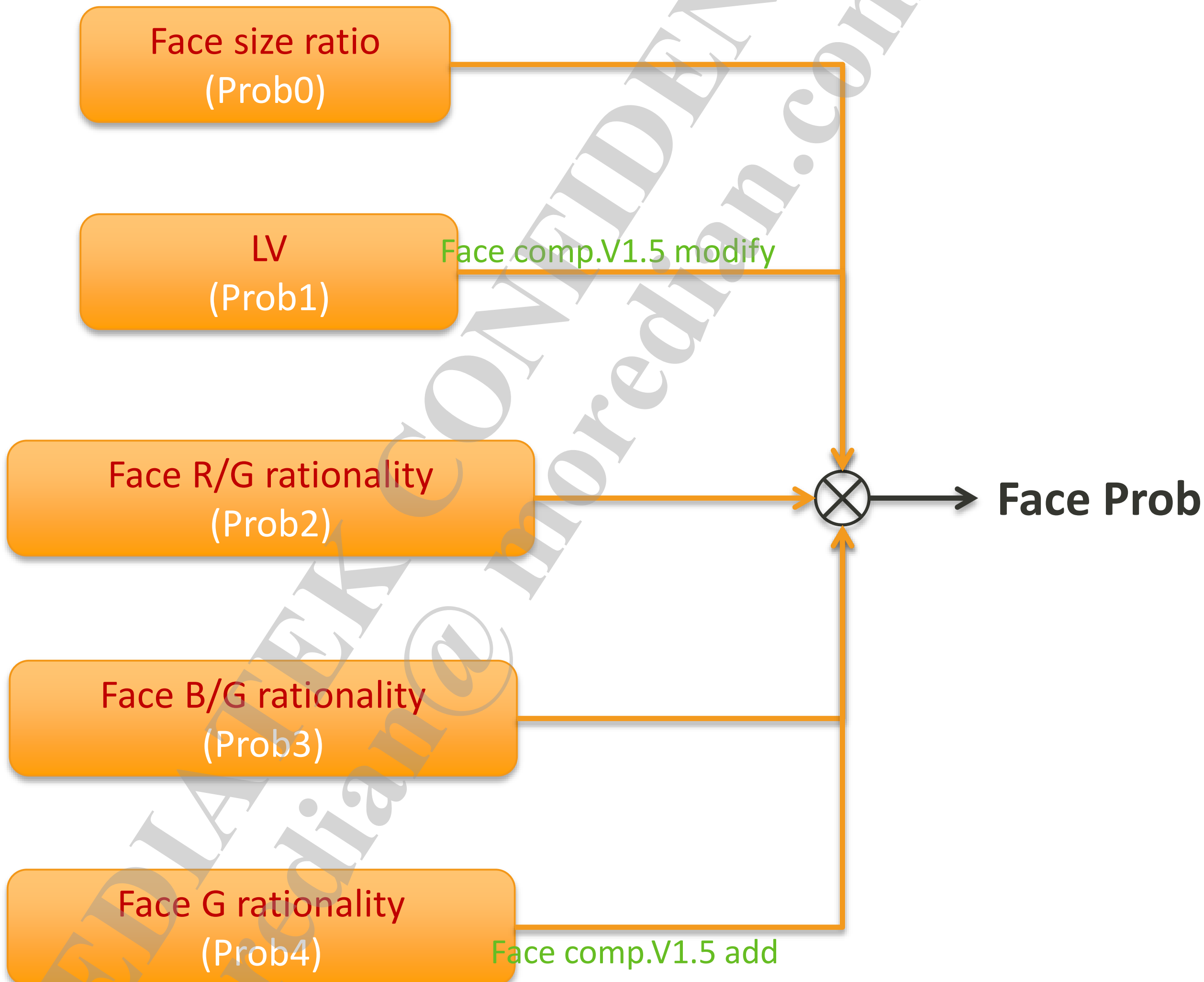
Use oversat Prob for saturation control

Do restriction before oversat prob hybrid 2 gain ratio

P1 has LV&CT info as index
Add P4 for face luma index



Face Comp. AWB Probability



Face Prob0， faceProb2， face prob3请参考：
MT6757_Camera_AWB_Tuning_Introduction.pdf P92-P96

Get Face Comp. AWB Prob1

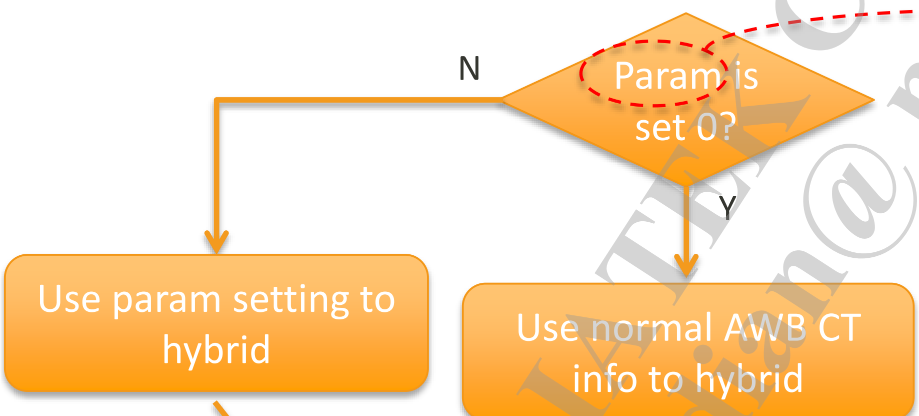
$$Face\_Prob1 = LV\_LUT$$

Same with normal AWB P1 LUT

Prob1的LUT由一组变成3组，根据色温来分

```
// 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
},
```

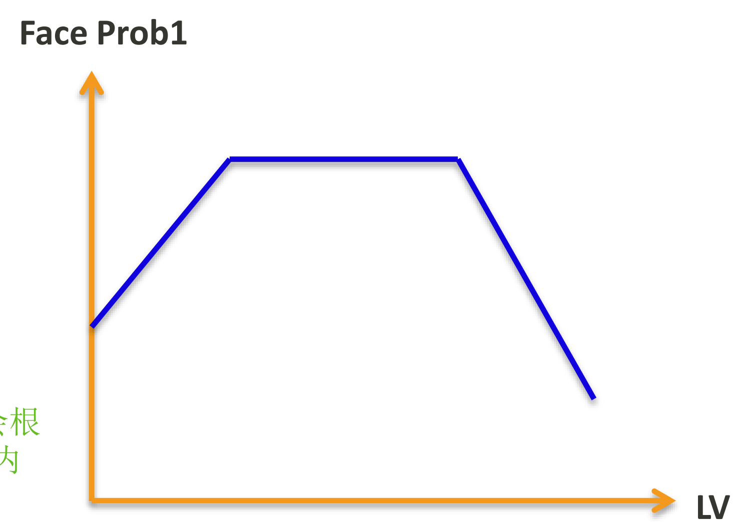
How to hybrid 3 set of P1?




```
// 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
},
```

同计算 hue/sat target一样， MidCT和LowCT之间会用FaceRBLow和FaceRBHi内插，HiCT和MidCT之间用LVLow和LVHi内插

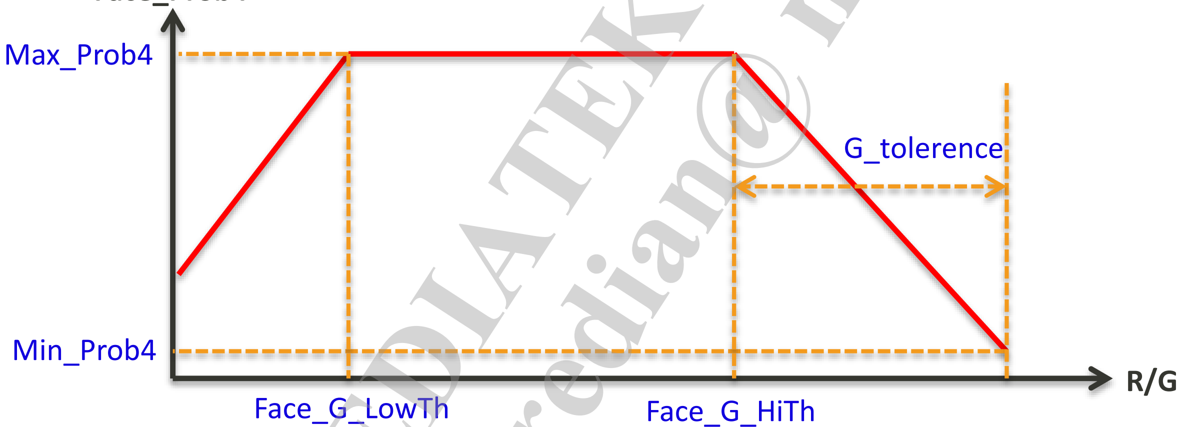
同计算target hue/sat一样，会根据HiCT_P, MidCT_P, LowCT_P 内插出Prob1。



Get Face Comp. AWB Prob4

Add 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.

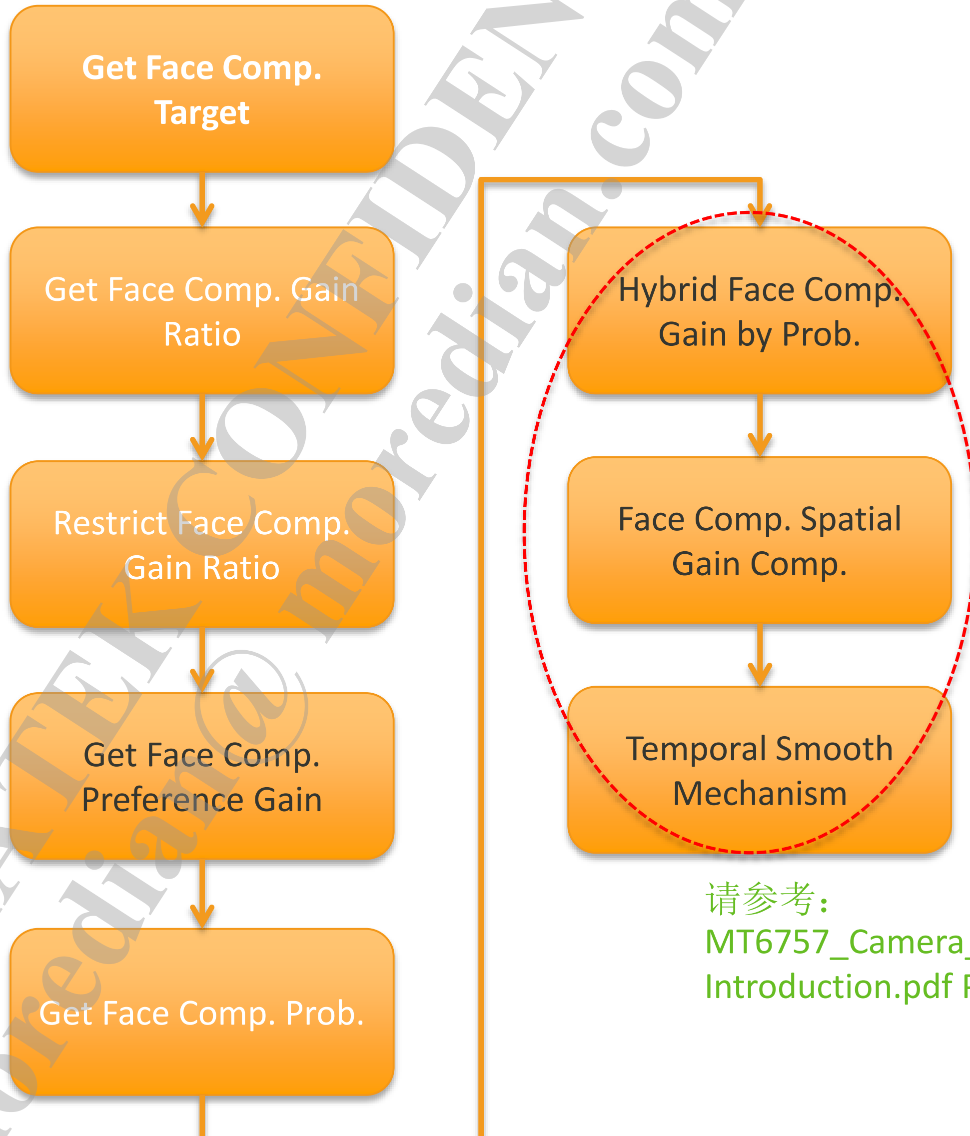
Algo will do source face data temporal smooth

Assign {Hue,Saturation,tolerance}, can refer normal awb Info for choosing target

Use oversat Prob for saturation control

Do restriction before oversat prob hybrid 2 gain ratio

P1 has LV&CT info as index
Add P4 for face luma index



请参考：
MT6757_Camera_AWB_Tuning_Introduction.pdf P96-P105

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

AWB v5.0

Light Source Statistics

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

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

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
},
```

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

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

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 ~

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 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, 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

Debug Parser Tag

AWB debug parser tag

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_LIMIT	Statistic Yr - i4LimitY
AWB_TAG_DL_OFFSET_GAIN	Daylight locus offset corresponding gain
AWB_TAG_GM_OFFSET_GAIN	Green/Magenta offset corresponding gain
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

AWB debug parser tag

Tag	Description
AWB_TAG_FACE_STAT_NR_ENABLE	Enable face statistic NR
AWB_TAG_FACE_STAT_NR_TH	Face statistic NR 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_UNDERSAT_SAT_HI/LOW	Face saturation threshold for Undersat prevention mechanism
AWB_TAG_FACE_UNDERSAT_RESTRICT	New Restrict ratio after Undersat prevention
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 debug parser tag

Tag	Description
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_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_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

Tag	Description
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_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_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

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_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 v5.0

AWB debug parser tag

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

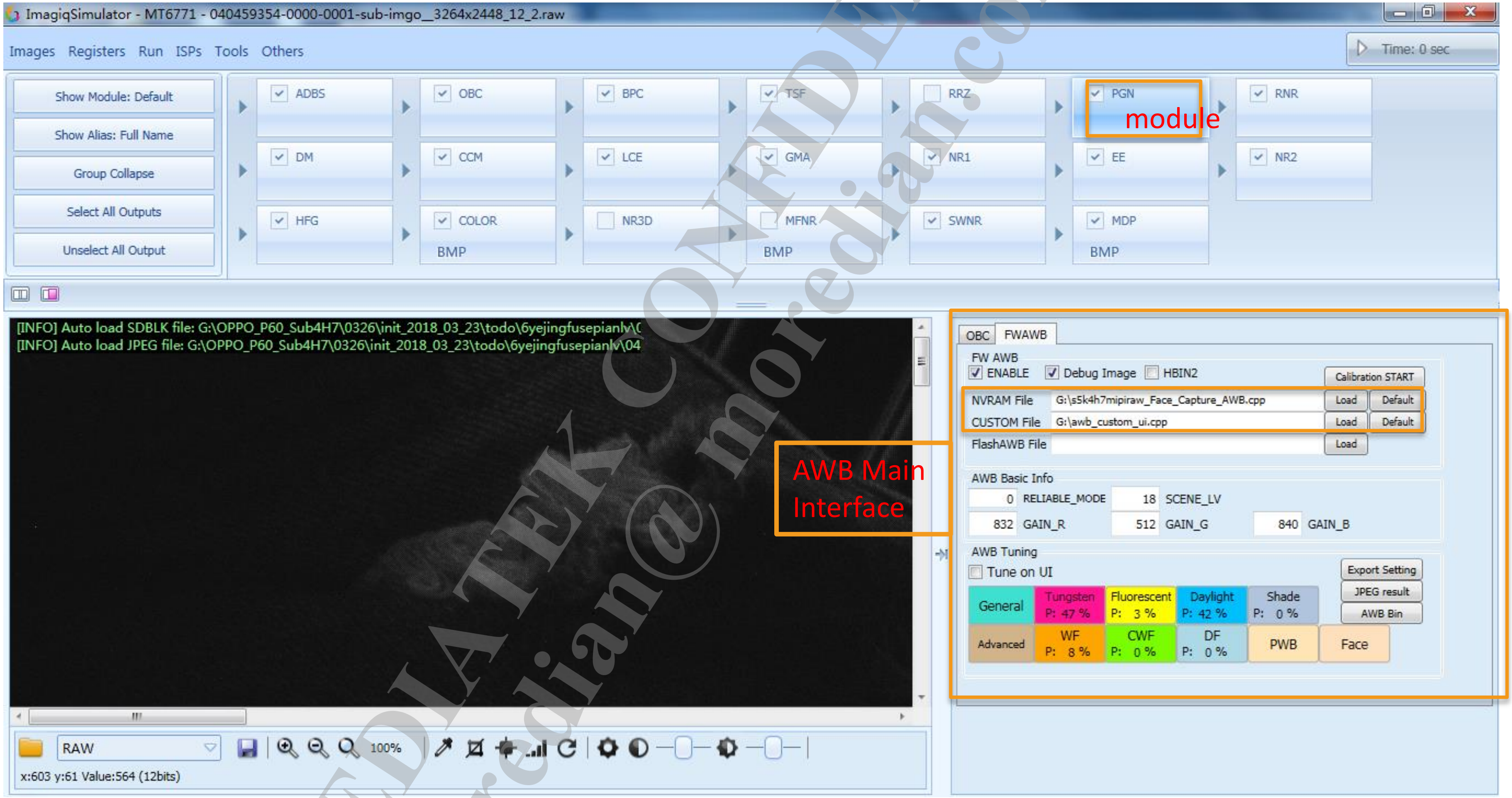
Tag	Description
AWB_TAG_FACEAST_CONF_PROB0	Face assisted AWB probability 0 for face size ratio
AWB_TAG_FACEAST_CONF_PROB1_<light>	Face assisted AWB probability 1 for environment brightness for each light source
AWB_TAG_FACEAST_CONF_PROB2	Face assisted AWB probability 2 for confusing color (corresponding 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

Tag	Description
AWB_TAG_FACEAST_DL_OFFSET_GAIN_R / B 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	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_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.

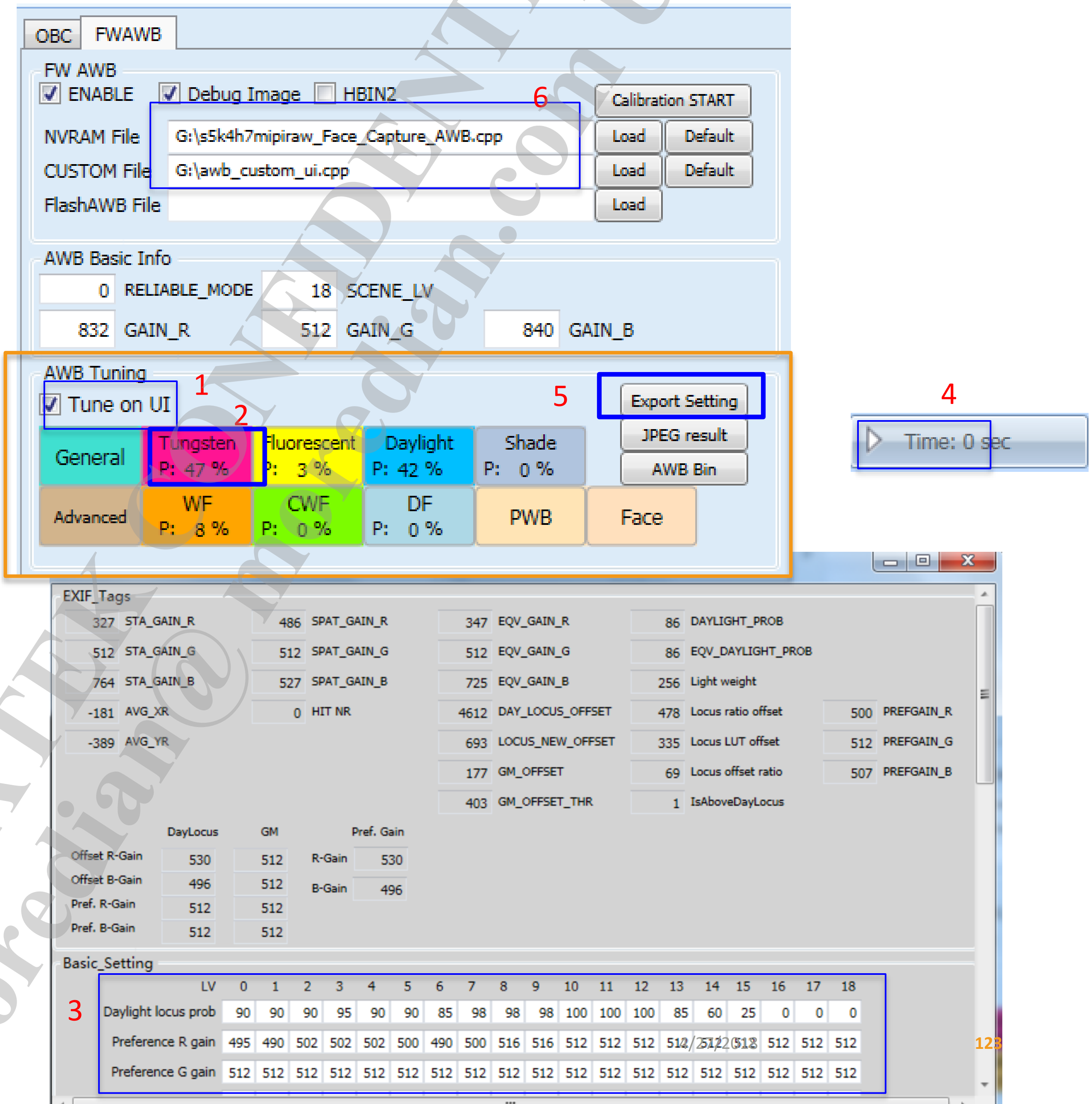
AWB Tunning Tool Oveview



模拟方法同P23，参考文档：MT6757_Camera_AWB_Tuning_Introduction.pdf P140-P145

AWB TunninTune on UI

P60 tool增加了Tune on UI 界面
Tune on UI之前的操作和P23相同。Tune on UI操作如下：
1、勾选上Tune on UI。
2、点击一个模块，则会弹出一个对话框显示模块的EXIF信息和param。
3、直接在对话框上修改参数。
4、点击右上角run 按钮，查看results结果。
5、点击Export setting按钮，先弹出nvram文件路径选择界面，输入文件名，保存参数。接着弹出custom文件路径选择界面，输入文件名，保存参数。
6、如果手动在文件中修改参数，则未勾选tune on UI，则tune on UI上param也会跟着改动。如果勾选tune on UI，则tune on UI上param不会跟着改动，以Tune On UI上参数为准。



CONFIDENTAL B

MT6771 CCM Introduction

Outline

- New Feature of P60 SCCM
- NVRAM Setting
- Debug info

New Features of P60 SCCM

- LV interpolation (6 LV partitions)
- Increased light sources support (10 CT partitions)

Steps for Calibration

For normal light condition

1. Collect processed raw in lab scene for different CT
2. Calibrate GAMMA
3. Set normal saturation target* for lab pass criterion
4. Calibrate CCM and fill in NVRAM data

* Saturation target: (High CT)D65/CWF/TL84 = 100~110
(Low CT) A/Hor = 90~100

Steps for Calibration

For low light condition

1. Dimming the light condition to the target value by using ND filter (ND16, ND32…) or controlling by lighting devices
2. Collect processed raw in lab scene for different CT
3. Set preferred saturation target (lower than normal)
4. Calibrate CCM and fill in NVRAM data

Example

Calibration

Normal LV in the lab environment is around 80~

CT_00 (Horizon)

CT_01 (A)

CT_02 (TL84)

CT_03 (CWF)

CT_04 (D65)

.IDX_Partition = {-30, -10, 20, 80, 100, 120}

Example

Calibration

Reduced LV in the lab environment is around 20~

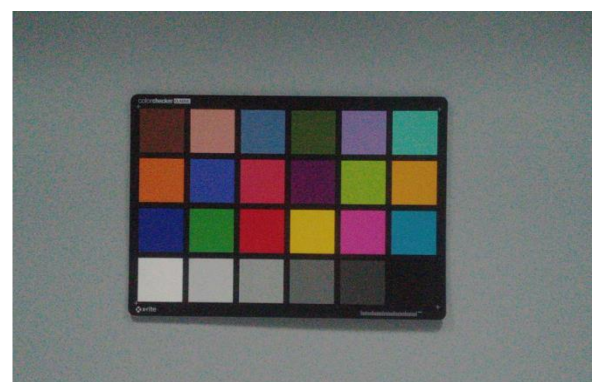
CT_00 (Horizon)

CT_01 (A)

CT_02 (TL84)

CT_03 (CWF)

CT_04 (D65)



.IDX_Partition = {-30, -10, 20, 80, 100, 120}

ISP NVRAM COLOR TABLE STRUCTURE

Ex: Tuning file : xxx_Scene_Capture_CCM.cpp

Number of calibrated light sources

.CCM_CT_valid_NUM = 5,

Weights for temporal frames

.CCM_Coef = {1, 2, 2}, //{ present frame, previous 1 frame, previous 2 frame }

ISP NVRAM CCM STRUCTURE

- Support 10 light sources or environment

```
.CCM_Reg =
  {
    {.set={//CT_00
      0x1FA8024E, 0x0000000A, 0x028D1F56, 0x0000001D, 0x1E4A0000, 0x000003B6
    }},
    {.set={//CT_01
      0x1FA8024E, 0x0000000A, 0x028D1F56, 0x0000001D, 0x1E4A0000, 0x000003B6
    }},
    {.set={//CT_02
      0x1EE20343, 0x00001FDB, 0x02BF1F95, 0x00001FAC, 0x1EAF002D, 0x00000324
    }},
    {.set={//CT_03
      0x1E3F03A2, 0x0000001F, 0x02801F78, 0x00000008, 0x1E750035, 0x00000356
    }},
    {.set={//CT_04
      0x1E950364, 0x00000007, 0x02701FE7, 0x00001FA9, 0x1DFE0061, 0x000003A1
    }},
    {.set={//CT_05
      0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
    }},
    {.set={//CT_06
      0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
    }},
    {.set={//CT_07
      0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
    }},
    {.set={//CT_08
      0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
    }},
    {.set={//CT_09
      0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000
    }}
  },
```

```
.AWBGain =
  {
    {//CT_00
      521,   // i4R
      512,   // i4G
     1930    // i4B
    },
    {//CT_01
      673,   // i4R
      512,   // i4G
     1493    // i4B
    },
    .
    .
    .
    .
    .
    .
    .
    .
    {//CT_08
      512,   // i4R
      512,   // i4G
      512    // i4B
    },
    {//CT_09
      512,   // i4R
      512,   // i4G
      512    // i4B
    }
  }
};
```

Note：需保持色温从低到高的顺序排列，如需插入其他光源同样。

LV partition setting

Ex: Tuning file : camera_ISP_common_xxxmipiraw.h

Note：做完CCM calibration之后记得将CCM参数copy到每一个LV。

```
.Lv_Env = { .u2Length = 6,
            .IDX_Partition = {-30, -10, 10, 50, 100, 120}
```

u2Length is fixed, but the points of LV partition are programmable

Debug info

ADB command :

For Smooth ccm debug info. :

adb shell setprop debug.smooth_ccm.enable 1

Information parsing from CCM NVRAM

01-01 00:08:46.528 D/ispfeature(3163): Real LV = 22
01-01 00:08:46.528 D/ispfeature(3163): Upper LV = 50
01-01 00:08:46.528 D/ispfeature(3163): Lower LV = 10
01-01 00:08:46.528 D/ispfeature(3163): Number of Upper CT = 5
01-01 00:08:46.528 D/ispfeature(3163): Number of Lower CT = 5
01-01 00:08:46.528 D/ispfeature(3163): Coef1, Coef2, Coef3 = 1, 2, 2
01-01 00:08:46.528 D/ispfeature(3163): LV_[50]_CT0_rCCM_Nvram = (590, -88, 10, -170, 653, 29, 0, -438, 950)
01-01 00:08:46.528 D/ispfeature(3163): LV_[50]_CT1_rCCM_Nvram = (590, -88, 10, -170, 653, 29, 0, -438, 950)
01-01 00:08:46.528 D/ispfeature(3163): LV_[50]_CT2_rCCM_Nvram = (835, -286, -37, -107, 703, -84, 45, -337, 804)
……..
……..

01-01 00:08:46.528 D/ispfeature(3163): LV_[10]_CT0_rCCM_Nvram = (590, -88, 10, -170, 653, 29, 0, -438, 950)
01-01 00:08:46.528 D/ispfeature(3163): LV_[10]_CT1_rCCM_Nvram = (590, -88, 10, -170, 653, 29, 0, -438, 950)
01-01 00:08:46.528 D/ispfeature(3163): LV_[10]_CT2_rCCM_Nvram = (835, -286, -37, -107, 703, -84, 45, -337, 804)
……..
……..
……..

Debug info

adb shell setprop debug.smooth_ccm.enable 1

Intermediate results

01-01 00:08:35.005 D/ispfeature(589): Initial Blending Ratio between CT0/CT2 = (0.578000)
01-01 00:08:35.005 D/ispfeature(589): Advanced Blending Ratio with estimated CT1 = (0.035804)
01-01 00:08:35.005 D/ispfeature(589): CCM interpolation results for set 0 (0:upper, 1:lower) = (688, -168, -8, -143, 671, -16, 17, -396, 891) // UpperLV set CCM result
01-01 00:08:35.005 D/ispfeature(589): Initial Blending Ratio between CT0/CT2 = (0.578000)
01-01 00:08:35.005 D/ispfeature(589): Advanced Blending Ratio with estimated CT1 = (0.035804)
01-01 00:08:35.005 D/ispfeature(589): CCM interpolation results for set 1 (0:upper, 1:lower) = (512, 0, 0, 0, 512, 0, 0, 0, 512) // LowerLV set CCM result
01-01 00:08:35.005 D/ispfeature(589): LowerLV CCM blending ratio = (0.707317)
01-01 00:08:35.005 D/ispfeature(589): Temporal CCM 0 = (581, -67, -2, -58, 573, -3, 4, -166, 674)
01-01 00:08:35.005 D/ispfeature(589): Temporal CCM 1 = (584, -70, -2, -61, 576, -3, 5, -173, 680)
01-01 00:08:35.005 D/ispfeature(589): Final CCM_Output = (578, -64, -2, -55, 570, -3, 4, -158, 666) //Final CCM blending result

CONFIDENTAL B

MT6771 ColorEngine Introduction

Outline

- ➢ P23/P60 Color Engine
- ➢ P60_ColorEngine_NewFlexibility

- P23/P60 Color Engine：
  - 原理部分请参考：MT6757_Camera_ISP_Tuning_Introduction.pdf P136-P169
  - 调试方法参考：MT6757_MT6763_AWB_CCM_Colorengine_Tuning_SOP.pdf 中 P61-P99

Outline

- P23/P60 Color Engine
- P60_ColorEngine_NewFlexibility
  - **Smooth Color Coding Guidelines**
  - Troubleshooting

Block Diagram

MT6771_NVRAM_IF_*.xlsx
Sheet : UserTable
Sheet : M! COLOR

Scenario and
CT/LV mapping table

camera_ISP_common_*.h

```
.Lv_Env={
.IDX_Partition ={-30, -10, 10, 50, 100, 120}},
.Ct_Env={
.IDX_Partition ={2000, 2500, 3000, 3500,
4000, 4500, 5000, 5500, 6000, 6500}},
```

NVRAM

***_Scenario_COLOR.cpp**

```
const ISP_NVRAM_SCOLOR_PARAM_T
Sensormipiraw_COLOR_0000 = {
   .FD_COLOR = {...}
   .COLOR_TBL = {...}
:
const ISP_NVRAM_SCOLOR_PARAM_T
Sensormipiraw_COLOR_0059 = {
   .FD_COLOR = {...}
   .COLOR_TBL = {...}
const ISP_NVRAM_COLOR_PARAM_T
Sensormipiraw_COLOR_PARAM_0000 =
{...}
```

Smooth Color
*Per frame active

Interpolation by CT → Modulation by LV → IIR → HW: Color Engine

mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/MT6771_NVRAM_IF_**Sensor**mipiraw.xlsx
mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/camera_isp_regs_**Sensor**mipiraw.h
mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/**Scenario**/**Sensor**mipiraw_**Scenario**_COLOR.cpp

MT6771_NVRAM_IF_Sensormipiraw.xlsx (Sheet : UserTable)

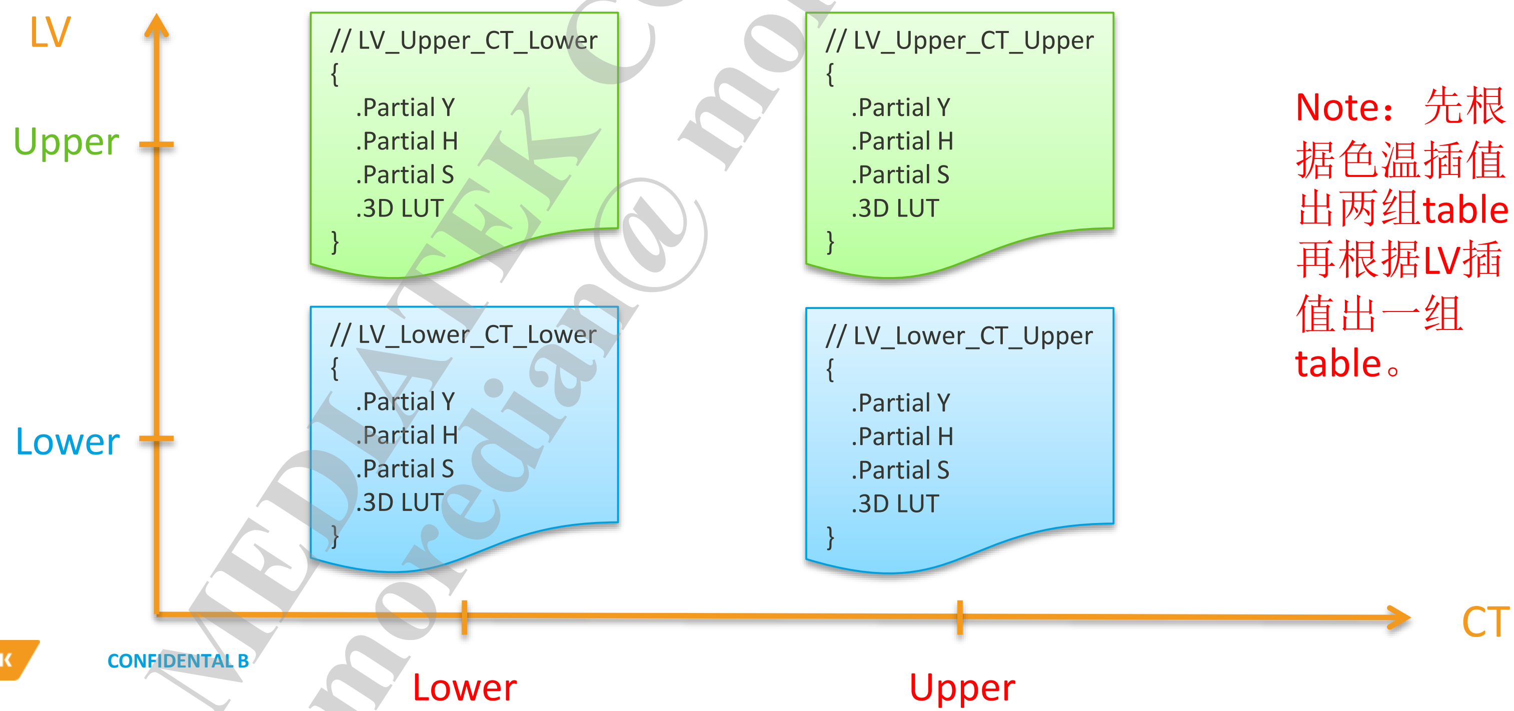
Scenario	CCM	COLOR
Scene_Capture	**Scene_Capture**	**Scene_Capture**
Face_Capture	**Face_Capture**	**Face_Capture**
Zoom_Capture	Scene_Capture	Scene_Capture
Professional_Capture	Scene_Capture	Scene_Capture
Flash_Capture	**Flash_Capture**	**Flash_Capture**
FaceBeauty_Capture	**FaceBeauty_Capture**	**FaceBeauty_Capture**
HDR_Capture	**HDR_Capture**	**HDR_Capture**
Panorama_Capture	Scene_Capture	Scene_Capture
Video_Capture	Scene_Capture	Scene_Capture
Capture_Preview	Scene_Capture	Scene_Capture
Capture_Preview_Zoom1	Scene_Capture	Scene_Capture
Capture_Preview_Zoom2	Scene_Capture	Scene_Capture
Video_Preview	Scene_Capture	Scene_Capture
Video_Preview_Zoom1	Scene_Capture	Scene_Capture
Video_Preview_Zoom2	Scene_Capture	Scene_Capture
Video_Recording	Scene_Capture	Scene_Capture
Video_Recording_Zoom1	Scene_Capture	Scene_Capture
Video_Recording_Zoom2	Scene_Capture	Scene_Capture
FaceBeauty_Preview	FaceBeauty_Capture	FaceBeauty_Capture
WeChatQQ	**WeChatQQ**	**WeChatQQ**
3rd_1080P	Scene_Capture	Scene_Capture
3rd_720P	Scene_Capture	Scene_Capture
3rd_480P	Scene_Capture	Scene_Capture

MT6771_NVRAM_IF_Sensormipiraw.xlsx (Sheet : M! COLOR)

LV	CT	ISO	Index	Folder	File	Scenario
IDX_00	IDX_00		0	Scene_Capture	COLOR	Scene_Capture
IDX_00	IDX_01		1	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_00	IDX_09		9	Scene_Capture	COLOR	Scene_Capture
IDX_01	IDX_00		10	Scene_Capture	COLOR	Scene_Capture
IDX_01	IDX_01		11	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_01	IDX_09		19	Scene_Capture	COLOR	Scene_Capture
IDX_05	IDX_00		50	Scene_Capture	COLOR	Scene_Capture
IDX_05	IDX_01		51	Scene_Capture	COLOR	Scene_Capture
⋮	⋮		⋮	⋮	⋮	⋮
IDX_05	IDX_09		59	Scene_Capture	COLOR	Scene_Capture
IDX_00	IDX_00		60	Face_Capture	COLOR	Face_Capture
IDX_00	IDX_01		61	Face_Capture	COLOR	Face_Capture
≀	≀		≀	≀	≀	≀
IDX_05	IDX_09		119	Face_Capture	COLOR	Face_Capture
IDX_00	IDX_00		0	Scene_Capture	COLOR	Zoom_Capture
IDX_00	IDX_01		1	Scene_Capture	COLOR	Zoom_Capture
≀	≀		≀	≀	≀	≀
IDX_05	IDX_09		59	Scene_Capture	COLOR	Zoom_Capture
≀ ≀	≀ ≀		≀ ≀	≀ ≀	≀ ≀	≀ ≀
IDX_00	IDX_00		120	Flash_Capture	COLOR	Flash_Capture
IDX_00	IDX_01		121	Flash_Capture	COLOR	Flash_Capture
≀	≀		≀	≀	≀	≀
IDX_05	IDX_09		179	Flash_Capture	COLOR	Flash_Capture
≀ ≀	≀ ≀		≀ ≀	≀ ≀	≀ ≀	≀ ≀

NVRAM by Scenario

- Each Scenario has 60 COLOR_TBLs, the output COLOR_TBL is bilinear interpolation result by 1/CT(Mired) and LV
  - 6 LV partitions
  - 10 CT partitions

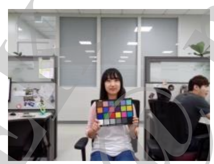


Note：先根据色温插值出两组table，再根据LV插值出一组table。

Individual Scene Tuning

- Database
  - Different CT
  - Different lighting condition
- Scene
  - Indoor





  - Outdoor




Individual Scene Tuning

↓

Decide SW Reg.

↓

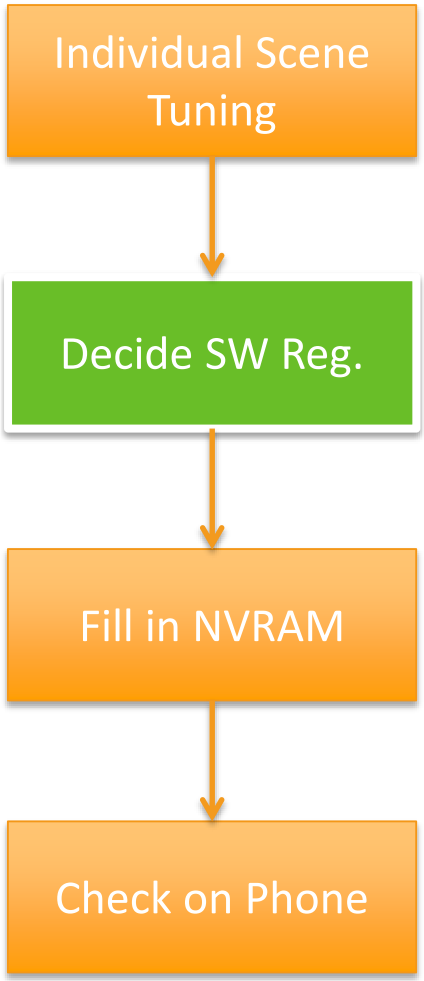
Fill in NVRAM

↓

Check on Phone

Decide SW Register

- LUM/HUE/SAT SPEED
  - Unit: 1 code/frame
  - Default: 4 code/frame
- LSP LV TH
  - LSP effective range by LV




```
mt6771/hal/imgsensor/ver1/Sensor_mipi_raw/Scenario/Sensormipiraw_Scenario_COLOR.cpp
const ISP_NVRAM_COLOR_PARAM_T Sensormipiraw_COLOR_PARAM_0000 = {
   0,  //DC OFFSET : unused
   4,  //LUM SPEED
   4,  //HUE SPEED
   4,  //SAT SPEED
   30, //LSP LV TH
   0,  //OUTDOOR EN : unused
   4,  //OUTDOOR SPEED : unused
   16  //OUTDOOR RATIO : unused
};
```

Note：
preview&video
收斂速度。

Fill in NVRAM

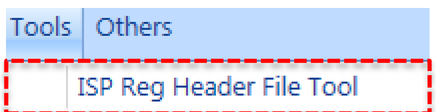
- Step

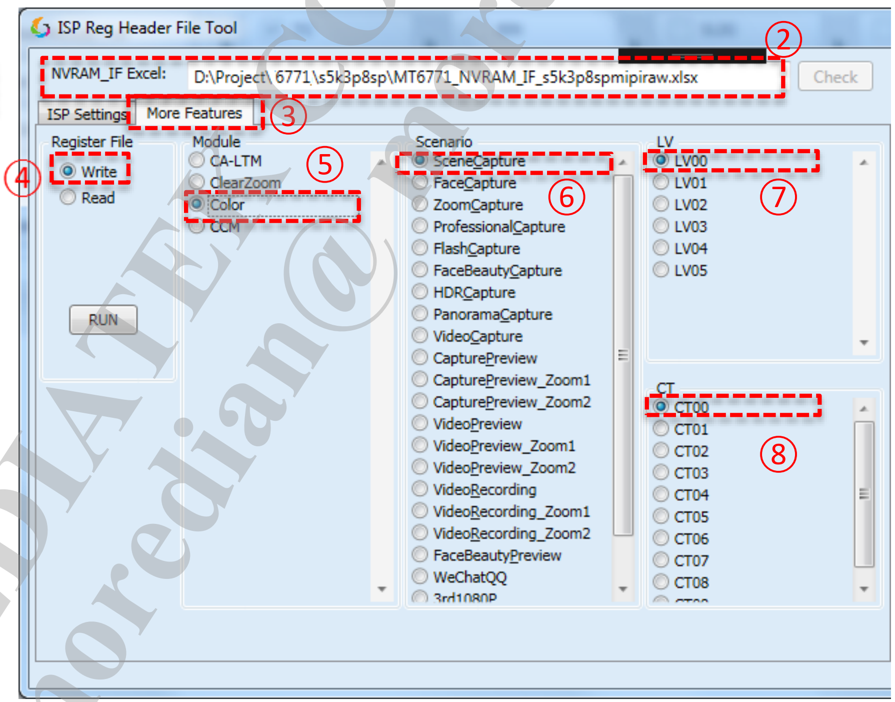
  Tool → ISP Reg Header File Tool

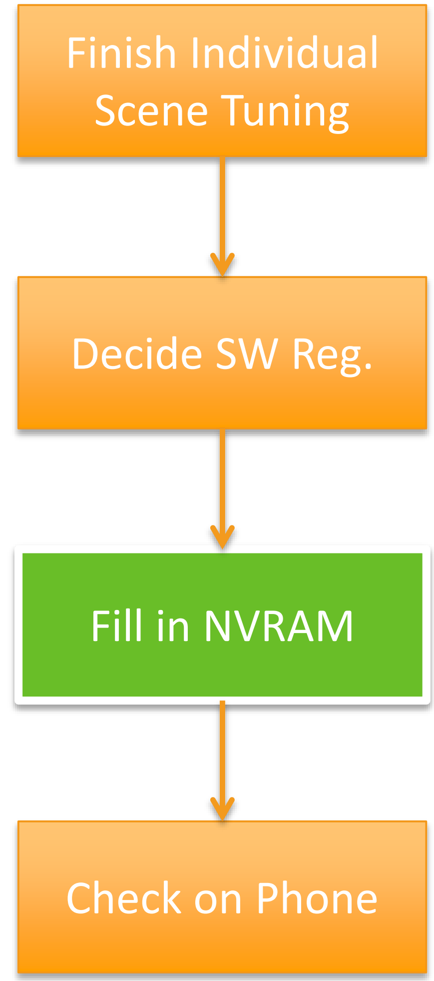
  → Open “NVRAM_IF.xlsx” → Select “More Feature”

  → Select “Write” → Select “Color” → Select “Scenario”

  → Select “LV” → Select “CT” → Run

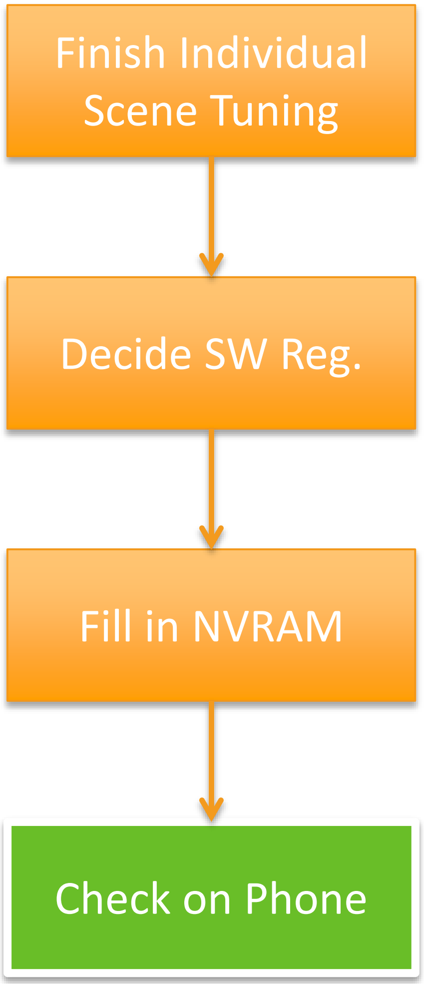






Check on Phone

- Field test
  - Check setting in EXIF
  - Check preview screen
- Troubleshooting



Outline

- P23/P60 Color Engine
- P60_ColorEngine_NewFlexibility
  - Smooth Color Coding Guidelines
  - Troubleshooting

Troubleshooting

- Bypass Y/S/H Engine and Memory Color
- Dump Input Parameters
- Dump Interpolation Results
  - Dump Global Brightness/Contrast/Saturation
  - Dump Partial Y
  - Dump Partial H
  - Dump Partial S
  - Dump S Gain by Y
  - Dump LSP
  - Dump Memory Color

Bypass Y/S/H Engine and Memory Color

Be used to debug abnormal color effect in preview or video

- Step 1: Turn on debug enable
  - adb shell setprop debug.smooth_color.enable 1
- Step 2: Bypass **Y/S/H engine and memory color**
  - adb shell setprop debug.smooth_color.bypass.yeng 1
  - adb shell setprop debug.smooth_color.bypass.heng 1
  - adb shell setprop debug.smooth_color.bypass.seng 1
  - adb shell setprop debug.smooth_color.bypass.cm 1

Dump Input Parameters

- Step 1: Turn on debug enable
  - adb shell setprop debug.smooth_color.dump 1
- Step 2: Check the debug info (=>Note)

[SmoothCOLOR()] u4RealCT = 3164, u4LowerCT = 3000, u4UpperCT = 3500, i4RealLV = 27, i4LowerLV = 10, i4UpperLV = 50
=> CT/LV info
[SmoothCOLOR()] IsCapture = 0, IsBoot = 0
=> If IsCapture = 1 or IsBoot = 1, there is no IIR
[SmoothCOLOR()] In Ram: 000, 004, 004, 004, 030, 000, 004, 016
=> See below description
[SmoothCOLOR()] LSP = (1, 1)
=> (1,1) : COLOR_LSP_EN = 1, otherwise COLOR_LSP_EN = 0
[SmoothCOLOR()] Interpolated mode !!
=> Output "Interpolated mode !!" or "Force Lock !!"

mt6771/hal/imgsensor/ver1/**Sensor**_mipi_raw/**Scenario**/**Sensor**mipiraw_Scenario_COLOR.cpp

```
const ISP_NVRAM_COLOR_PARAM_T Sensormipiraw_COLOR_PARAM_0000 = {
   0,  //DC OFFSET : unused
   4,  //LUM SPEED
   4,  //HUE SPEED
   4,  //SAT SPEED
   30, //LSP LV TH
   0,  //OUTDOOR EN : unused
   4,  //OUTDOOR SPEED : unused
   16  //OUTDOOR RATIO : unused
};
```

Dump Global Brightness/Contrast/Saturation

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.global 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamGlobal()]

```
[InterParamGlobal()]  u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamGlobal()] (LV Lower, Mired Lower)  B:128,C:128,S:128
[InterParamGlobal()] (LV Lower, Mired Upper)  B:128,C:128,S:128
[InterParamGlobal()] (LV Upper, Mired Lower) B:128,C:128,S:128
[InterParamGlobal()] (LV Upper, Mired Upper) B:128,C:128,S:128
[InterParamGlobal()] (Target)  B:128,C:128,S:128
[InterParamGlobal()] (Final )  B:128,C:128,S:128
```

Dump Partial Y

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.lum 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialY()]

```
[InterParamPartialY()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamPartialY()] PartialY LV Lower, Mired Lower : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128
[InterParamPartialY()] PartialY LV Lower, Mired Upper: 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128
[InterParamPartialY()] PartialY LV Upper, Mired Lower : 160,160,160,160,120,120,120,160,160,144,160,160,160,160,160,160
[InterParamPartialY()] PartialY LV Upper, Mired Upper: 107,104,109,107,139,138,137,106,104,116,106,109,107,104,109,107
[InterParamPartialY()] PartialY Target : 128,127,128,128,121,120,121,131,127,128,128,128,128,127,128,128
[InterParamPartialY()] PartialY Final : 128,127,128,128,121,120,121,131,127,128,128,128,128,127,128,128
```

Dump Partial H

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.hue 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialH()]

```
[InterParamPartialH()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamPartialH()] PartialH LV Lower, Mired Lower : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Lower, Mired Upper : 128,128,128,128,130,131,134,144,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialH()] PartialH LV Upper, Mired Upper : 128,128,128,128,138,154,138,140,128,128,128,128,128,128,128,128,128,144,132,128
[InterParamPartialH()] PartialH Target : 128,128,128,128,131,134,134,144,128,128,128,128,128,128,128,128,128,131,129,128
[InterParamPartialH()] PartialH Final : 128,128,128,128,131,134,134,144,128,128,128,128,128,128,128,128,128,131,129,128
```

Dump Partial $S_{1/2}$

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.sat 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamPartialS()]

```
[InterParamPartialS()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamPartialS()] PartialS point1 Lv Lower Mired Lower : 020,020,020,020,020,020,010,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Lower Mired Upper : 020,020,020,020,020,020,010,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Upper Mired Lower : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Lv Upper Mired Upper : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point2 Lv Lower Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Lower Mired Upper : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Upper Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Lv Upper Mired Upper : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS gain1 Lv Lower Mired Lower : 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Lower Mired Upper : 128,128,128,128,112,112,112,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Lv Upper Mired Upper : 128,128,128,128,120,112,116,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Lv Lower Mired Lower : 140,140,140,140,120,120,120,144,144,132,140,140,112,120,144,140,140,140,140,140
[InterParamPartialS()] PartialS gain2 Lv Lower Mired Upper : 136,144,144,144,120,120,120,144,144,132,144,144,144,144,144,144,144,144,136,128
[InterParamPartialS()] PartialS gain2 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Lv Upper Mired Upper : 138,138,138,138,120,112,116,128,128,128,138,142,148,142,138,138,128,128,128,128
[InterParamPartialS()] PartialS gain3 Lv Lower Mired Lower : 120,119,121,120,139,138,136,114,113,124,120,121,139,134,118,120,120,121,119,120
[InterParamPartialS()] PartialS gain3 Lv Lower Mired Upper : 123,116,118,118,139,138,136,114,113,124,117,118,117,116,118,118,117,119,122,128
[InterParamPartialS()] PartialS gain3 Lv Upper Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain3 Lv Upper Mired Upper : 121,121,122,121,136,143,139,128,128,128,121,120,115,118,122,121,128,128,128,128
```

Dump Partial $S_{2/2}$

```
[InterParamPartialS()] PartialS output1 Lv Lower Mired Lower : 020,020,020,020,018,018,009,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Lower Mired Upper : 020,020,020,020,018,018,009,010,010,010,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Upper Mired Lower : 020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output1 Lv Upper Mired Upper : 020,020,020,020,019,018,018,020,020,020,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output2 Lv Lower Mired Lower : 064,064,064,064,056,056,056,066,066,062,064,064,055,058,065,064,064,064,064,064
[InterParamPartialS()] PartialS output2 Lv Lower Mired Upper : 063,065,065,065,056,056,056,066,066,062,065,065,065,065,065,065,065,065,063,060
[InterParamPartialS()] PartialS output2 Lv Upper Mired Lower : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS output2 Lv Upper Mired Upper : 063,063,063,063,057,053,054,060,060,060,063,064,066,064,063,063,060,060,060,060
[InterParamPartialS()] PartialS point1 Target : 020,020,020,020,020,020,012,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point1 Final : 020,020,020,020,020,020,012,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS point2 Target : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS point2 Final : 060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060,060
[InterParamPartialS()] PartialS output1 Target : 020,020,020,020,018,018,011,012,012,012,020,020,020,020,020,020,020,020,020,020
[InterParamPartialS()] PartialS output2 Target : 063,065,065,065,056,056,056,066,066,062,065,065,063,064,065,065,065,065,063,061
[InterParamPartialS()] PartialS gain1 Target : 128,128,128,128,115,115,117,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain1 Final : 128,128,128,128,115,115,117,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamPartialS()] PartialS gain2 Target : 138,144,144,144,122,122,120,144,144,133,144,144,138,141,144,144,144,144,138,131
[InterParamPartialS()] PartialS gain2 Final : 138,144,144,144,122,122,120,144,144,133,144,144,138,141,144,144,144,144,138,131
[InterParamPartialS()] PartialS gain3 Target : 122,116,118,118,137,136,136,115,114,123,117,118,122,119,118,118,117,119,121,126
[InterParamPartialS()] PartialS gain3 Final : 122,116,118,118,137,136,136,115,114,123,117,118,122,119,118,118,117,119,121,126
```

Dump S Gain by Y

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.sgy 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamSGainByY()]

```
[InterParamSGainByY()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamSGainByY()] SGainByY000 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Lower, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 Target : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY000 Final : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 LV Lower, Mired Upper : 127,127,121,122,122,121,119,120,120,120,123,128,123,122,131,130,131,116,117,125
[InterParamSGainByY()] SGainByY064 LV Upper, Mired Lower : 127,127,123,122,131,130,131,116,117,125,121,122,122,121,119,120,120,120,123,128
[InterParamSGainByY()] SGainByY064 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY064 Target : 127,127,124,125,125,124,123,123,123,123,125,128,125,125,131,130,131,121,122,126
[InterParamSGainByY()] SGainByY064 Final : 127,127,124,125,125,124,123,123,123,123,125,128,125,125,131,130,131,121,122,126
[InterParamSGainByY()] SGainByY128 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY128 LV Lower, Mired Upper : 124,122,122,121,120,122,126,125,124,125,127,128,125,125,131,131,131,127,126,127
[InterParamSGainByY()] SGainByY128 LV Upper, Mired Lower : 124,122,125,125,131,131,131,127,126,127,122,121,120,122,126,125,124,125,127,128
[InterParamSGainByY()] SGainByY128 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY128 Target : 126,125,125,124,123,125,127,126,126,126,127,128,126,126,131,131,131,127,127,127
[InterParamSGainByY()] SGainByY128 Final : 126,125,125,124,123,125,127,126,126,126,127,128,126,126,131,131,131,127,127,127
[InterParamSGainByY()] SGainByY192 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY192 LV Lower, Mired Upper : 125,121,128,127,127,127,122,122,124,124,123,128,120,120,140,140,138,116,116,126
[InterParamSGainByY()] SGainByY192 LV Upper, Mired Lower : 125,121,120,120,140,140,138,116,116,126,128,127,127,127,122,122,124,124,123,128
[InterParamSGainByY()] SGainByY192 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY192 Target : 126,124,128,127,128,128,125,124,125,126,125,128,123,123,137,137,135,121,121,127
[InterParamSGainByY()] SGainByY192 Final : 126,124,128,127,128,128,125,124,125,126,125,128,123,123,137,137,135,121,121,127
[InterParamSGainByY()] SGainByY256 LV Lower, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Lower, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Upper, Mired Lower : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 LV Upper, Mired Upper : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 Target : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
[InterParamSGainByY()] SGainByY256 Final : 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128
```

Dump LSP

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.lvdump.lsp 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParamLSP()]

```
[InterParamLSP()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParamLSP()] LSP1 LV Lower, Mired Lower : 000,000,127,000
[InterParamLSP()] LSP1 LV Lower, Mired Upper : 000,000,127,000
[InterParamLSP()] LSP1 LV Upper, Mired Lower : 000,000,127,000
[InterParamLSP()] LSP1 LV Upper, Mired Upper : 000,000,127,000
[InterParamLSP()] LSP1 Target : 000,000,127,000
[InterParamLSP()] LSP1 Final  : 000,000,127,000
[InterParamLSP()] LSP2 LV Lower, Mired Lower : 127,000,127,127
[InterParamLSP()] LSP2 LV Lower, Mired Upper : 127,000,127,127
[InterParamLSP()] LSP2 LV Upper, Mired Lower : 127,000,127,127
[InterParamLSP()] LSP2 LV Upper, Mired Upper : 127,000,127,127
[InterParamLSP()] LSP2 Target : 127,000,127,127
[InterParamLSP()] LSP2 Final  : 127,000,127,127
```

Dump Memory Color

- Step 1: Tuning on debug log.
  - adb shell setprop debug.smooth_color.dump.cm 1
- Step 2: Check log if anything strange.
  - Keyword : [InterParam3DLUT()]

```
Only show W3 Hue

[InterParam3DLUT()]u4RealMired = 200,   u4UpperMired = 181,   u4LowerMired = 250,   i4RealLV = 60,   i4UpperLV = 100,   i4LowerLV = 50
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Lower Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Lower Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Upper Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Lv Upper Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Target : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Input Final : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Lower Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Lower Mired Upper : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Upper Mired Lower : 0806,0829,0852,0875,0898,0921,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Lv Upper Mired Upper : 0806,0847,0888,0929,0934,0939,0944
[InterParam3DLUT()] 3DLUT W3 Hue Otput Target : 0806,0832,0857,0883,0903,0000,0944
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Lower Mired Lower : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Lower Mired Upper : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Upper Mired Lower : 128,128,128,128,128,128
[InterParam3DLUT()] 3DLUT W3 Hue Slope Lv Upper Mired Upper : 230,230,230,026,026,026
[InterParam3DLUT()] 3DLUT W3 Hue Slope Target : 145,139,145,111,000,255
[InterParam3DLUT()] 3DLUT W3 Hue Slope Final : 145,139,145,111,000,255
[InterParam3DLUT()] 3DLUT W3 Hue Lv Lower Mired Lower : WGT_LSLOPE = 000, WGT_USLOPE = 000
[InterParam3DLUT()] 3DLUT W3 Hue Lv Lower Mired Upper : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Lv Upper Mired Lower : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Lv Upper Mired Upper : WGT_LSLOPE = 016, WGT_USLOPE = 016
[InterParam3DLUT()] 3DLUT W3 Hue Target : WGT_LSLOPE = 014, WGT_USLOPE = 014
[InterParam3DLUT()] 3DLUT W3 Hue Final  : WGT_LSLOPE = 014, WGT_USLOPE = 014
```

everyday genius