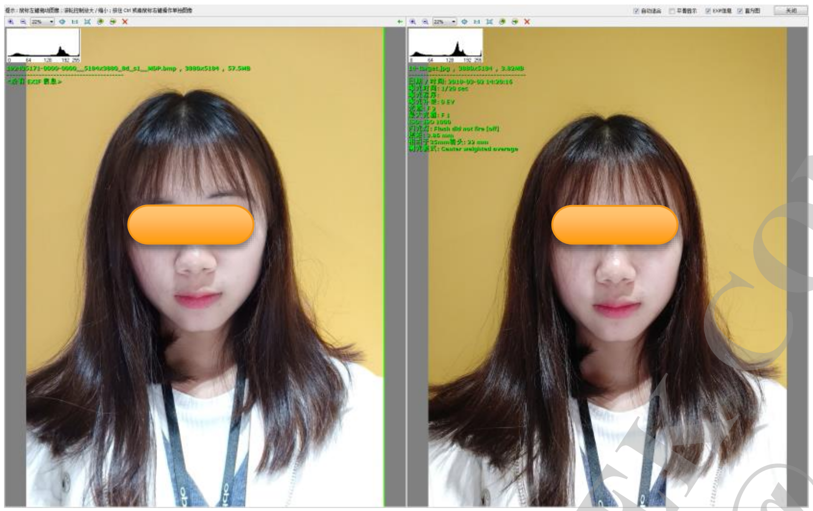
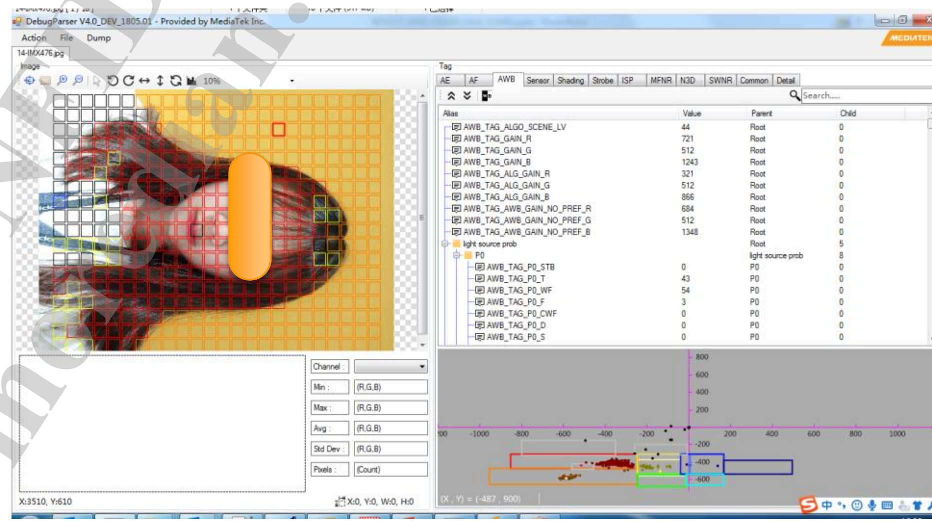
CONFIDENTIAL B

MT6771 AWB Color Case Study

Face Assist 相关参数Tuning方法

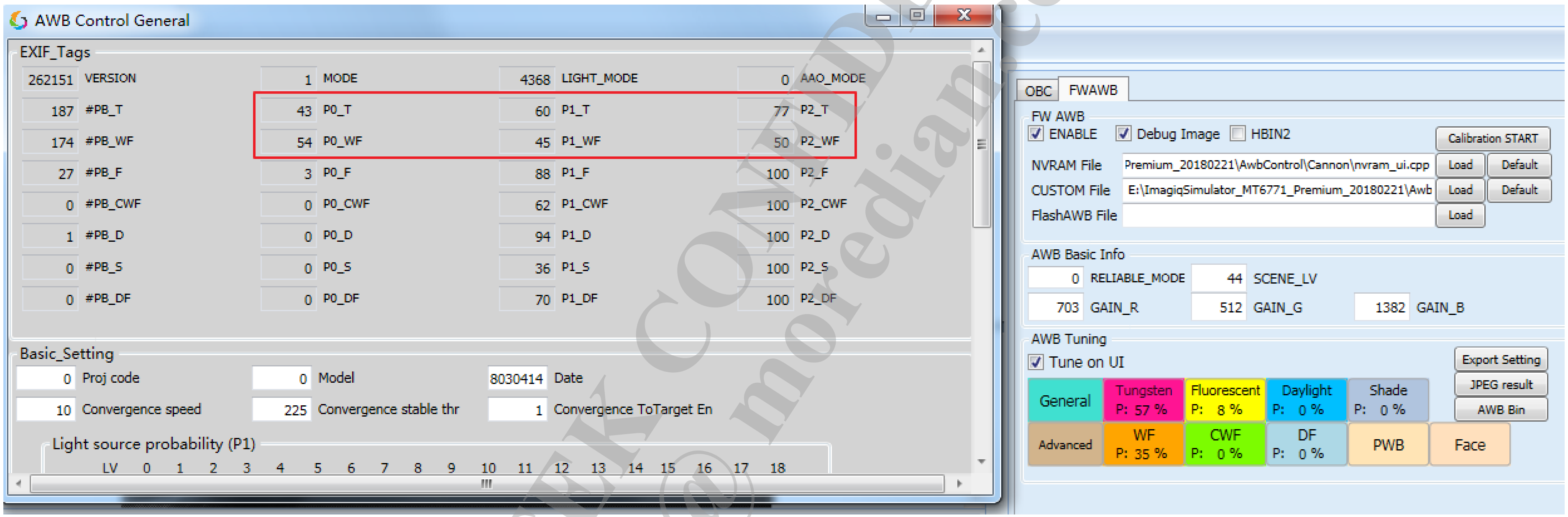
original　　　　　　　　target





分析：从EXIF信息中看落点信息，橙色完全落入WF，人脸区域主要落入Tungsten。如果把橙色收白最终效果就好比较紫红，目前的图片看起来有点蓝紫，因此可以先降低WF光比例，打开face assistant计算face reference gain是否有加R减B 的趋势。

1、在Tuning on UI 界面导入拍照时的参数，从模拟EXIF结果看，2个色温的P2都有降低，可再继续降低WF光的P2。



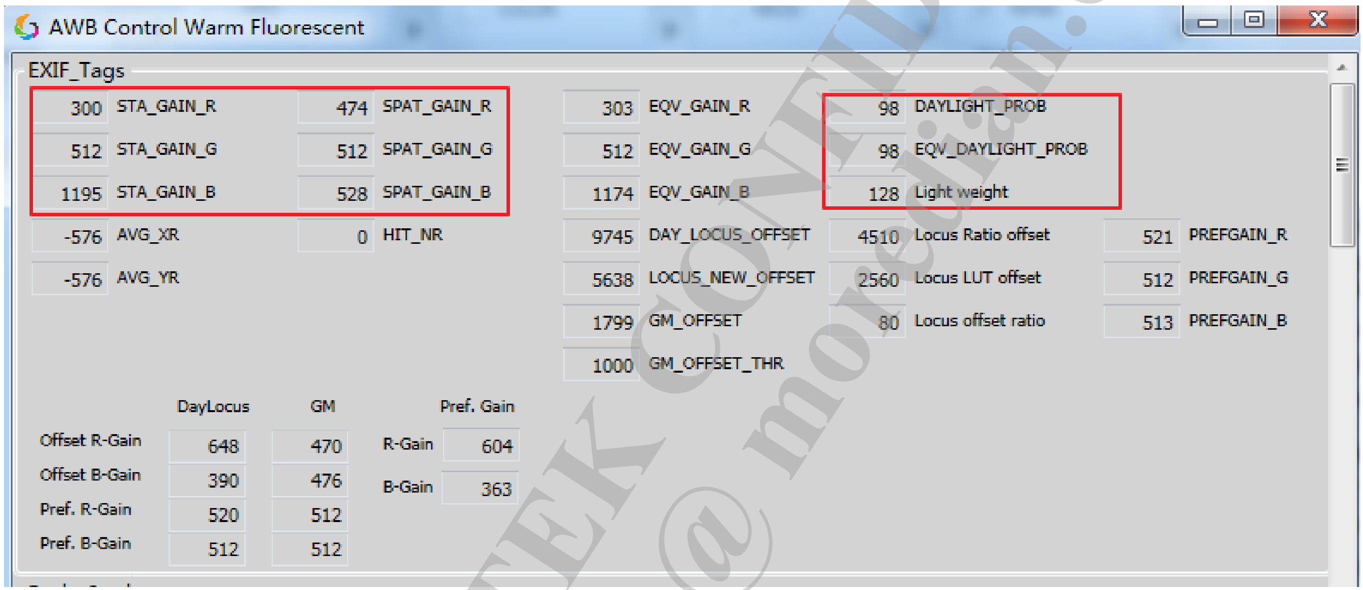
Tungsten

294	EQV_GAIN_R
512	EQV_GAIN_G
873	EQV_GAIN_B

WF

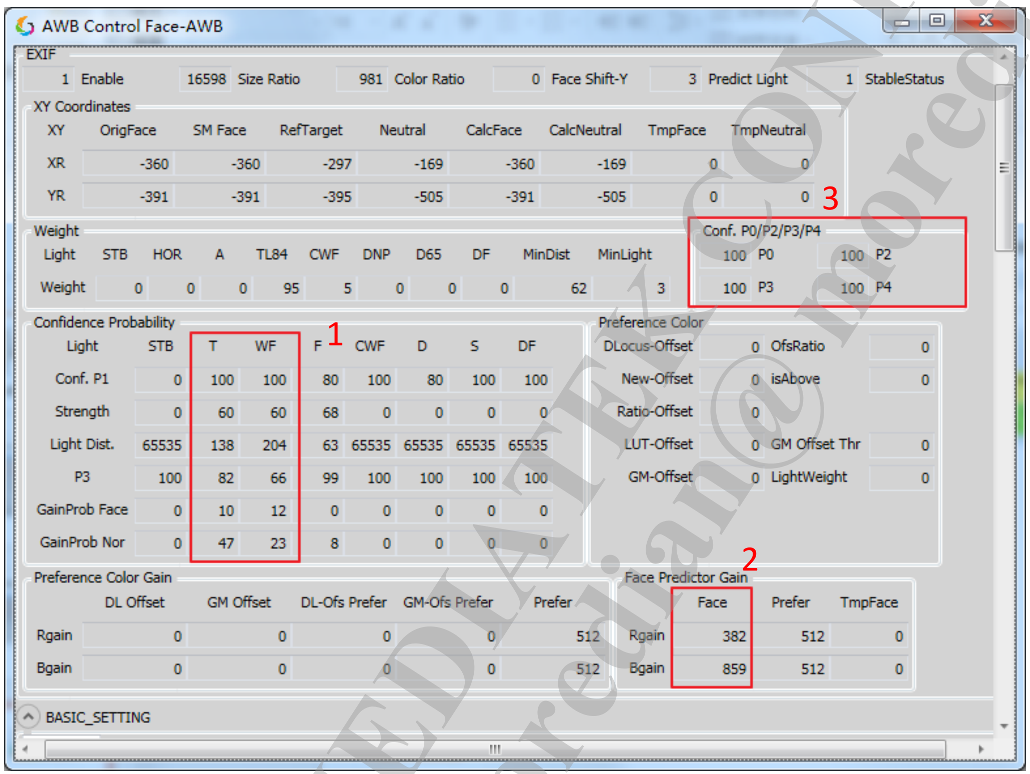
303	EQV_GAIN_R
512	EQV_GAIN_G
1174	EQV_GAIN_B

2、WF 光statistics gain比较蓝，可以通过多混spatial gain来降低，但是会导致同LV下的落入WF光的低色温太黄，对比T和WF的EQV Gain，T光的要正常一些，因位橙色很大概率是混淆色，因此首先考虑降低WF的P2来降低WF比例。

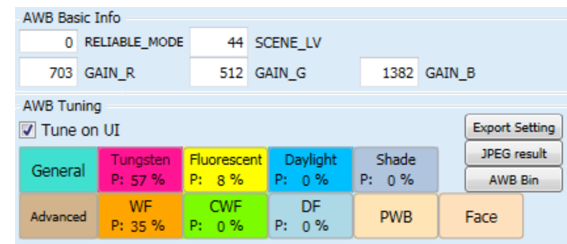


3、face assistant EXIF 信息查看

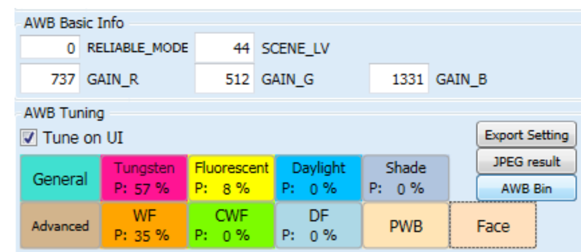
AWB Tuning on UI 界面可查看Face assistant 模拟的结果，可先看confidence probability的GainProb Face，如果为0，表示face assistant对这个色温没有作用，如果有作用再查看face predictor gain，可以用这个gain与各色温的EQV GAIN*preference gain做对比，看多混face predictor gain是否会对效果有改善，如果没改善并且作用是相反的，可以调整face assistant相关参数降低混合比例，最常调的是根据LV调各色温的P1来降低混合比例。



Face assist off



Face assist on



4、face assistant 常调参数

Prob1 (1)

LV	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Tungsten	50	75	100	100	100	100	90	90	100	100	100	100	100	100	100	100	80	60	40
WF	50	75	100	100	100	100	100	100	100	100	100	100	100	100	100	100	80	60	40
F	50	75	75	75	80	80	80	80	80	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	80	80	80	80	80	80	80	80	80	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	50	75	100	100	100	100	100	100	100	100	100	100	100	100	100	100	80	60	40

Prob2 (2)

	0	25	50	75	100	125	150	175	200	225	250
High-CT Magenta	256	256	256	256	256	256	232	212	200	160	128
High-CT Green	256	256	256	256	256	256	240	212	180	128	80
Mid-CT Magenta	256	256	256	256	256	256	256	256	200	160	128
Mid-CT Green	256	256	256	256	256	256	240	212	180	128	80
Low-CT Magenta	256	256	256	256	240	212	180	128	96	64	40
Low-CT Green	256	256	256	256	256	256	240	212	180	128	80

Prob3 / Prob4 (4)

	Prob3 (Stable)	Prob4 (Color)
THR Low	5	200
THR High	30	600
MinProb	0	0
MaxProb	100	100

Compensation Setting (3)

	Confidence	Distance	Comp. Ratio
Thr. Low	0	60	0
Thr. Low	100	230	40

Stable Setting

TempoWeight	7
Delay Frame	8

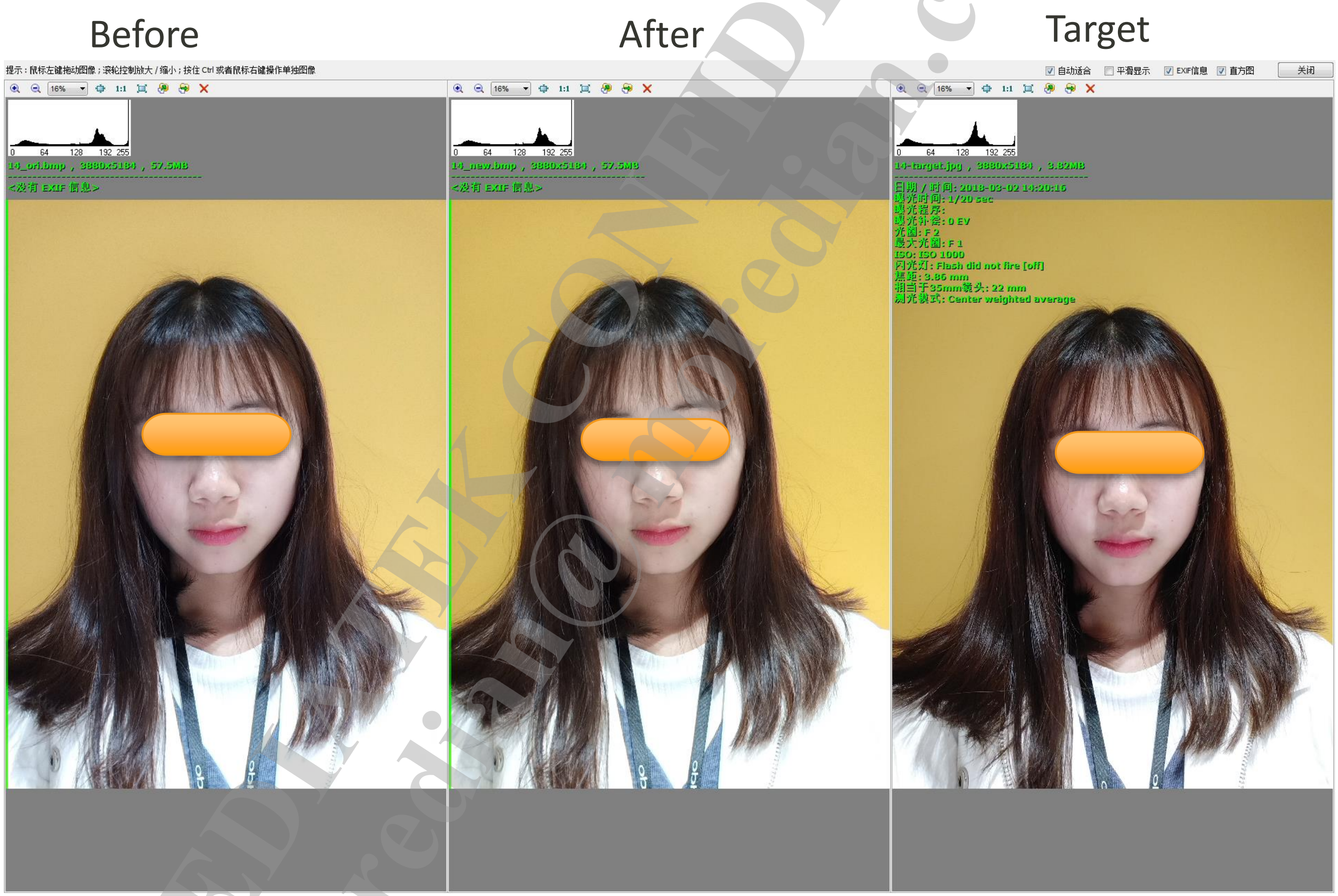
1、如果是单一场景问题，有些根据LV和色温来调整P1和P2，但是P2不太容易打到，所以常调的还是P1。

2、如果想整体加大face assistant强度，可直接调整3的comp ratio，前期要先确定好

3、Distance的Thr Low和High调整可影响最终P3的结果，需前期确定好范围。

4、Prob3和Prob4在拍前摄正常大小人脸时，调整的参数应让最终稳定后的P3和P4能达到100

5、通过调整WF的P2和打开face assistant模拟，最终结果如下图中间效果，较之前有很大改善。
Face assist和P2的作用主要是用来降低混淆色比例，因此遇到混淆色优先考虑这2个feature。



Face assist calibration

Face assist reference target Calibration 方法：
1、用AWB golden去灯箱拍各个色温下的人像卡jpg。

2、将EXIF信息中ORIG_FACE_XR/YR值填入参数档，如下图红框部分。各色温的Low/mid/high分别对应不同亮度的target，可对应不同肤色的target，默认我们可以设一样的。

AWB_TAG_FACEAST_ORIG_FACE_XR	-662	none
AWB_TAG_FACEAST_ORIG_FACE_YR	-323	none

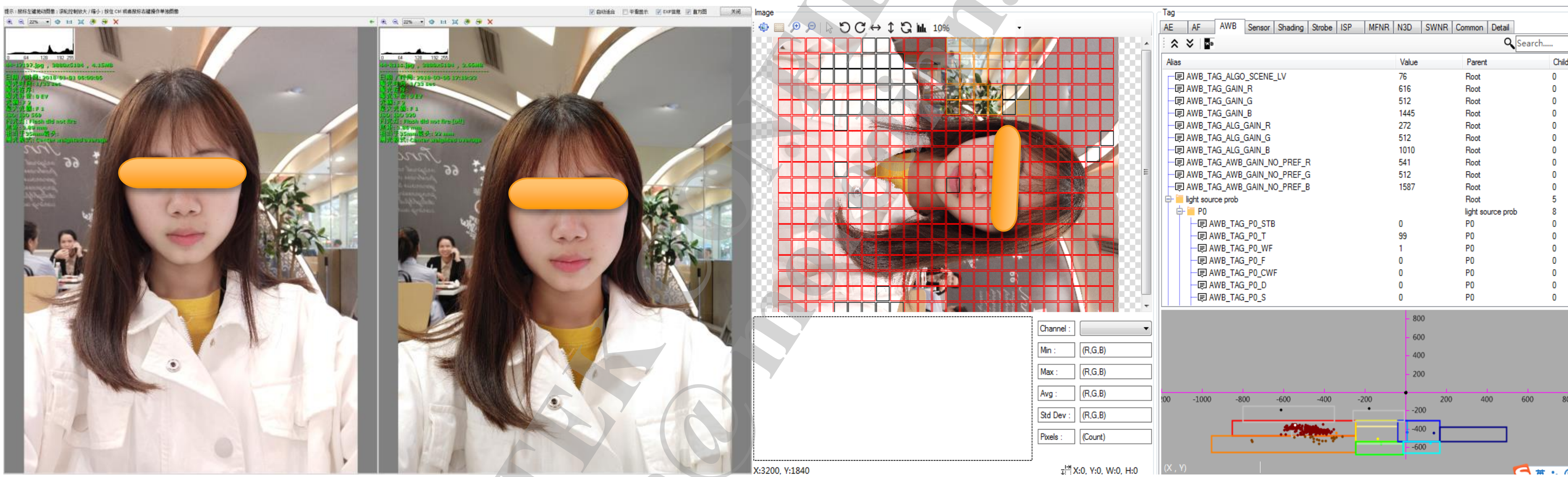
Reference Target

1	Mode	100	Coef_A	100	Coef_B	32	ThrLow	64	ThrMid	96	ThrHigh

	Strobe		Horizon		A		TL84		CWF		DNP		D65		DF	
Tolerance	32		36		36		32		30		28		24		24	
Face Low	4095	4095	-662	-323	-512	-341	-298	-389	-274	-513	-190	-334	-65	-341	-88	-473
Face Mid	4095	4095	-662	-323	-512	-341	-298	-389	-274	-513	-190	-334	-65	-341	-88	-473
Face High	4095	4095	-662	-323	-512	-341	-298	-389	-274	-513	-190	-334	-65	-341	-88	-473
Pref. Gain	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512

Face Comp 相关参数Tuning方法

original　　　　　　　　target



分析：从落点和EXIF信息看，这个景基本都是Tungsten 光源，比target偏红，可以调整低色温相关的参数来改善，如果低色温相关参数调整与其他景冲突，可以考虑face comp是否有机会调整。

face comp 相关EXIF分析：

1、查看OrigTarget 与Target gain的差异，如果2个gain一致表示face comp对这个景没有作用。如果没有作用，可以先查看2和3部分的条件是否满足

2、2和3的exif信息表示当前Hue和saturation与参数设置的target Hue和sat，如果Hue或者Sat的current与target差异超过Tolerance，才会用face comp来补偿normal awb 的结果。因此要让一个景face comp有作用，首先调整参数满足Hue或者Sat的条件

3、如果满足face comp的作用条件，则可以调整作用强度。从调试经验看，一般face comp作用的景都是SAT满足条件，作用的方向主要就是加饱和度和降饱和度，这个景的current SAT为42775，大于target，则会apply一组reduceSat 的gain，reduce的趋势是减R加B，加B的最大强度是3.5%，R做适当的计算。

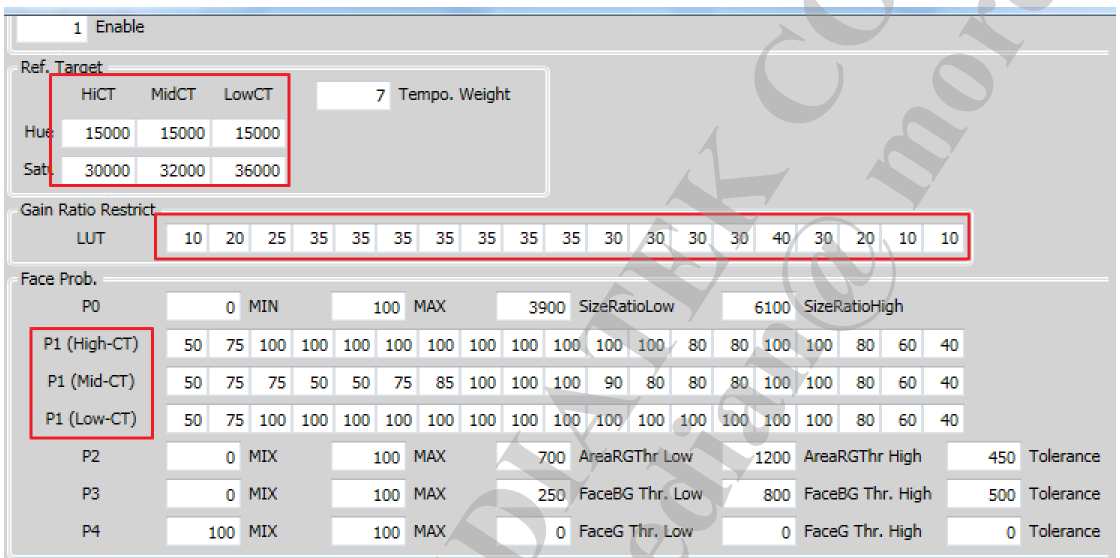
Face comp主要调整参数：

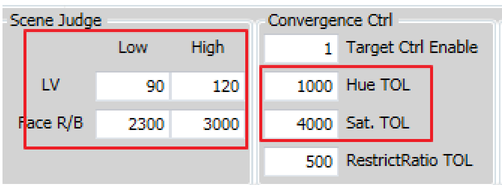
1、ref target，可调整低中高色调的target Hue和saturation，搭配TOL一起调整。这部分参数前期先大致调整到一个范围，后期做大的改动影响范围较大；

2、根据LV调整gain ratio，如果部分LV需要加大或者降低face comp的强度，可调整gain ratio；

3、同一个LV下，不同色温下可能需要apply不同强度的face comp，可根据P1来分别调整。当scene Judge参数为0，则P1的低中高色温由normal AWB的色温确定权重。如果scene judge有值，则计算face的R/B并用scene Judge参数来判断色温，ref target的低中高色温区分方法同P1。

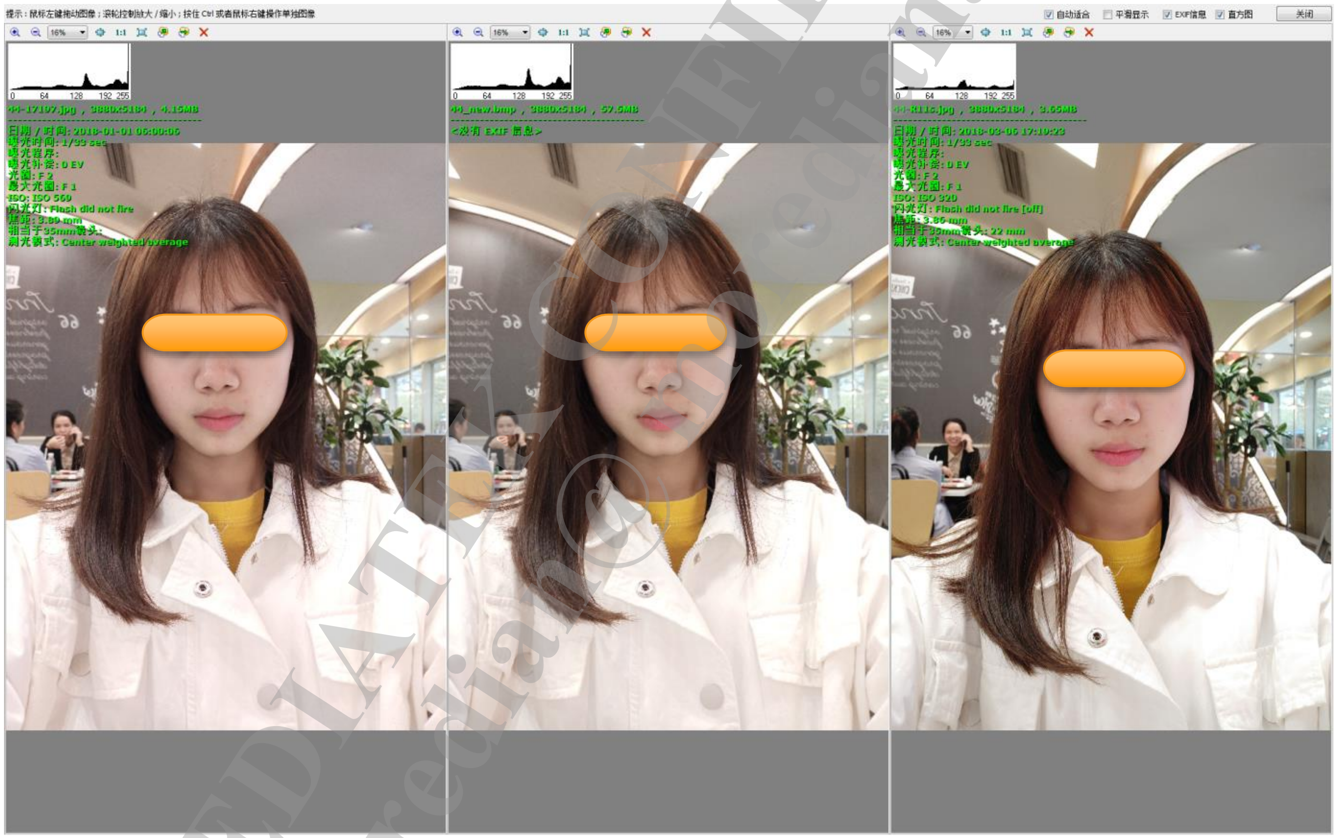
4、P2和P3不常调，除非有极端景才需要做相应调整。P2主要低色温能打到，P3主要是中高色温能打到





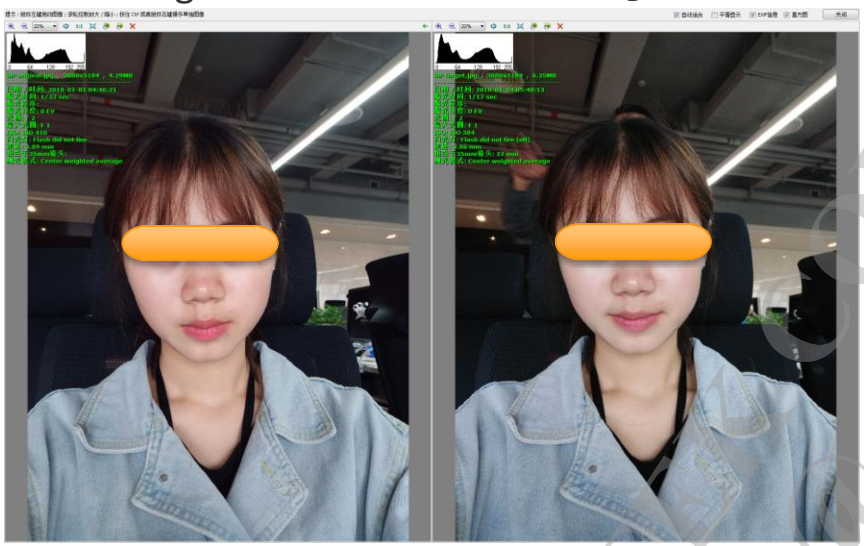
5、face comp可以用来改善人脸饱和度非常低或非常高时的效果，可在不影响正常饱和度场景下，将饱和度差异大的景补偿到一定范围，face comp可作为针对特殊景做差异化调整。

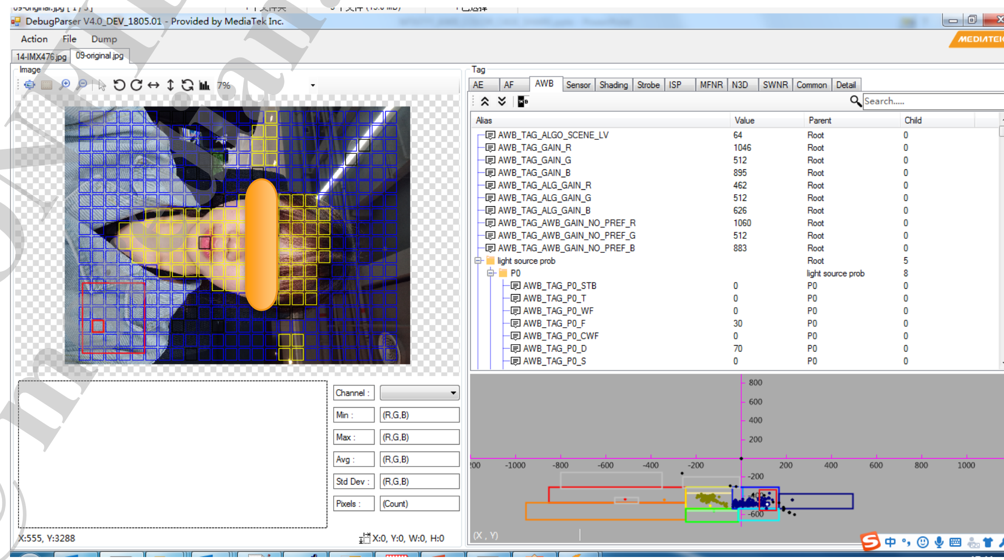
Before After Target



Extra Color 相关参数Tuning方法

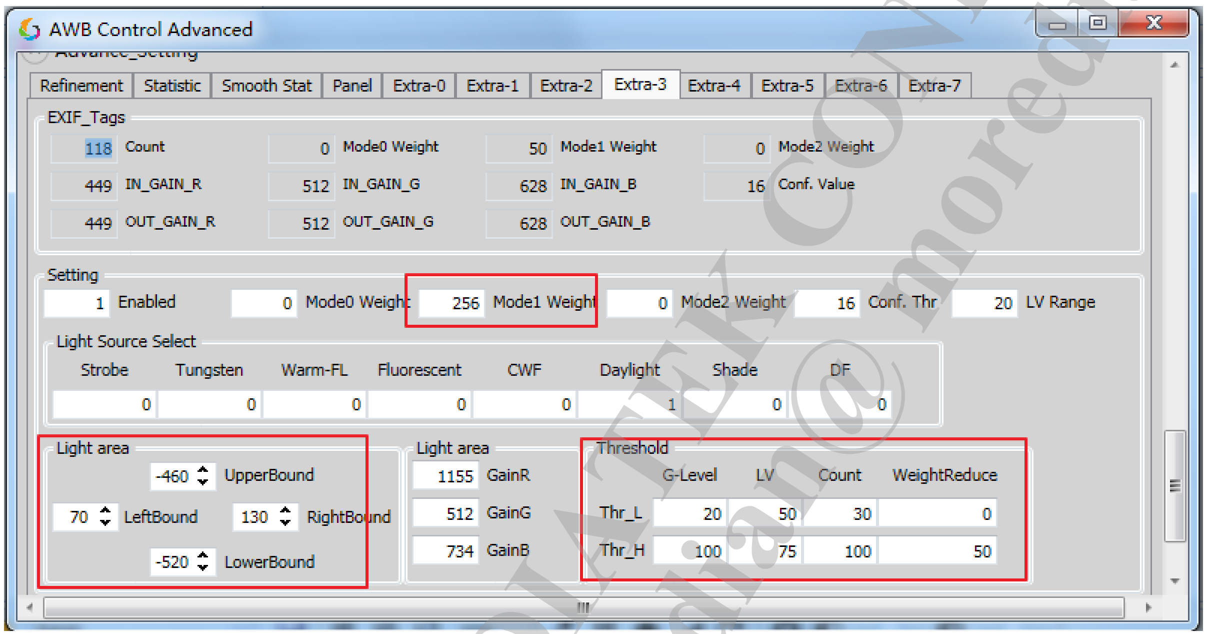
original　　target



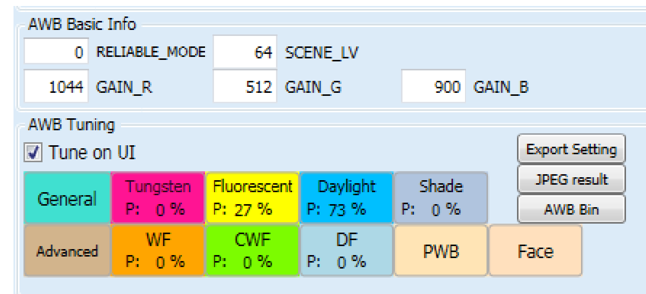


分析：与对比机相比有点偏红，从EXIF信息中看落点信息，主要是牛仔色衣服落入Daylight并靠近右边，如红色框位置，这样就会导致最新daylight 平均落点会在真实光源的基础上右移，导致计算的统计gain会偏红。Daylight可调整的参数比较少，衣服是混淆色，不建议直接调整preference gain或混spatial gain比例，落点比较靠下，也没办法调整limit Y来改善。这种景基本只能通过调整光源比例来改善，MT6771有多开出8个extra color框，可以改变光源的P2和daylight locus prob，因此这个景可以尝试用extra color 来做特殊调整。

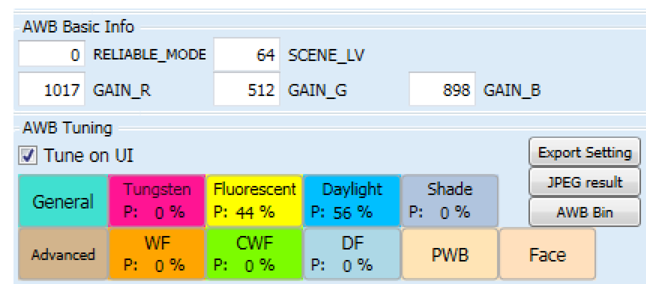
1、用DP看衣服落点位置，用Light area把衣服的落点框进来，然后再模拟输出process raw，量测衣服区域的G value，设置G-Level范围卡落入extra color的落点，如果只是针对这个景，还需要用LV来卡范围，以免影响其他景，模拟可以从exif信息看到满足落点的count和各mode的weight。Mode0为混入extra color的spatial gain，Mode1位降低P2，mode2为降低daylight locus prob。此场景可以通过降低daylight的P2来改善，因此light source select选daylight，mode1的weight设最大256，最终模拟可以让Daylight的P2降低50%。



Before:

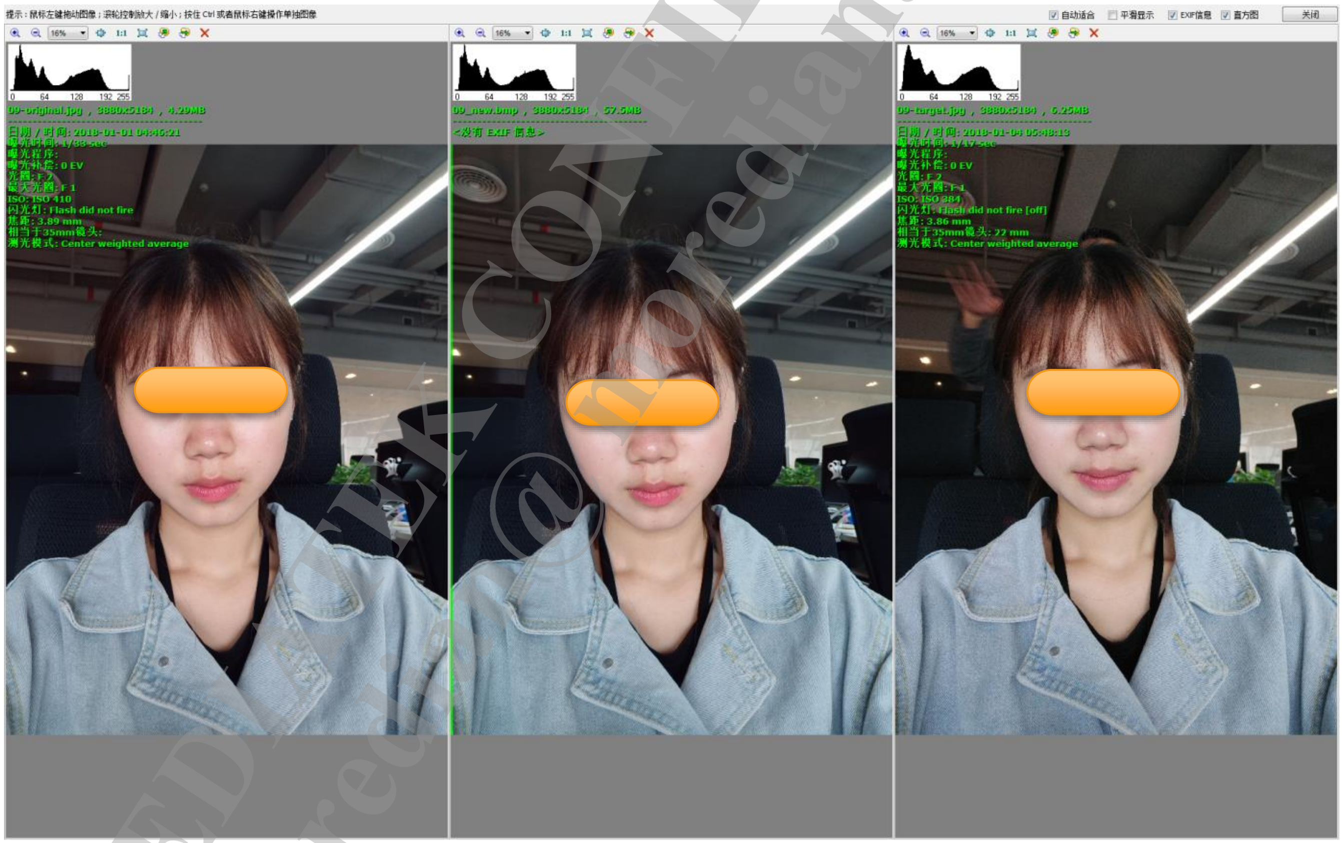


After:



2、最终效果有改善，MT6771在之前平台基础上，增加了更多的调试弹性。能更好的解决特殊景的一些问题，但用extra color调试时需多考虑不同场景是不是会有冲突，如果是单个景问题，可将条件设置严一些，这样影响范围比较小。

Before After Target



Color engine default value reset 导致闪烁

项目调试过程中有用color engine中的Y engine 模块加大face 区域的亮度，给参数过程中部分index的default value不小心被reset掉，导致概率性出现闪烁的问题，下面蓝框的值为被reset的值。

Default value：

```
.COLOR_TBL = {
.cfg_main   ={.bits ={.COLOR_C2P_BYPASS=0, .COLOR_P2C_BYPASS=0, .COLOR_YENG_BYPASS=0, .COLOR_SENG_BYPASS=0, .COLOR_HENG_BYPASS=0, .rsv_5=0, .COLOR_ALL_BYPASS
.c_boost_main={.bits ={.COLOR_C_BOOST_GAIN=128, .rsv_8=0, .COLOR_C_NEW_BOOST_EN=0, .rsv_14=0, .COLOR_C_NEW_BOOST_LMT_L=64, .COLOR_C_NEW_BOOST_LMT_U=255}},
.c_boost_main_2={.bits ={.COLOR_COLOR_CBOOST_YOFFSET=0, .rsv_8=0, .COLOR_COLOR_CBOOST_YOFFSET_SEL=0, .rsv_18=0, .COLOR_COLOR_CBOOST_YCONST=128}},
.luma_adj   ={.bits ={.COLOR_Y_LEVEL_ADJ=64, .rsv_7=0, .COLOR_Y_SLOPE_LMT=255, .rsv_16=0}},
.g_pic_adj_main_1={.bits ={.COLOR_G_CONTRAST=128, .rsv_10=0, .COLOR_G_BRIGHTNESS=1024, .rsv_27=0}},
```

被 reset value

```
.COLOR_TBL = {
.cfg_main   ={.bits ={.COLOR_C2P_BYPASS=0, .COLOR_P2C_BYPASS=0, .COLOR_YENG_BYPASS=0, .COLOR_SENG_BYPASS=0, .COLOR_HENG_BYPASS=0, .rsv_5=0, .COLOR_ALL_BYPASS
.c_boost_main={.bits ={.COLOR_C_BOOST_GAIN=0, .rsv_8=0, .COLOR_C_NEW_BOOST_EN=0, .rsv_14=0, .COLOR_C_NEW_BOOST_LMT_L=0, .COLOR_C_NEW_BOOST_LMT_U=0}},
.c_boost_main_2={.bits ={.COLOR_COLOR_CBOOST_YOFFSET=0, .rsv_8=0, .COLOR_COLOR_CBOOST_YOFFSET_SEL=0, .rsv_18=0, .COLOR_COLOR_CBOOST_YCONST=0}},
.luma_adj   ={.bits ={.COLOR_Y_LEVEL_ADJ=0, .rsv_7=0, .COLOR_Y_SLOPE_LMT=0, .rsv_16=0}},
.g_pic_adj_main_1={.bits ={.COLOR_G_CONTRAST=128, .rsv_10=0, .COLOR_G_BRIGHTNESS=1024, .rsv_27=0}},
```

Color engine default value reset 导致闪烁

- Color调试请注意以下事项：

1、各index COLOR_TBL （如前几行），control bit 若无特殊需求，尽可能维持 default value, or 各 index align

2、因为这些control bit不会被内差 → 若相邻 index, control bit 开关有差异，会造成不预期的跳变

3、以memory color 为例子，同一组 memory color,全部最好都定位为同一用途，control bit要开就全部index一起开，input range 也设为一样的window，效果控制可靠output offset控制，不想有作用就設定为 0 (slope = 128)

4、提交参数前，需要compare diff code,看是否都是预期内的修改（因为参数有可能被tool 改掉一些 control bit 设定）

AWB Log分析

1、adb cmd:
adb shell setprop debug.mapping_mgr.enable 1
adb shell setprop debug.awb_log.enable 1
adb shell setprop debug.awb_mgr.enable 1
adb shell setprop debug.awb.enable 1

2、通过log确认不同模式切换时AWB 参数index是否正确。如下Idx后的数字为AWB NVRAM对应的index。Index主要由Profile,sensor mode，APP和face来确定，不同组合跑哪个index有参数档中的MT6771_NVRAM_IF_xxx.xlsx确定。
PF: profile   SM： sensor mode FD: face detect

```
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 2) (PF Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP MTKCam, FD 1, ZOOM 1, LV 0, CT 0, ISO 0, CUSTOM 0)
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 0) (PF Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP MTKCam, FD 0, ZOOM 0, LV 0, CT 0, ISO 0, CUSTOM 0)
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 3) (PF mHDR Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP HDR, FD 0, ZOOM 1, LV 0, CT 0, ISO 0, CUSTOM 0)
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 3) (PF mHDR Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP HDR, FD 1, ZOOM 1, LV 0, CT 0, ISO 0, CUSTOM 0)
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 0) (PF Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP MTKCam, FD 0, ZOOM 0, LV 0, CT 0, ISO 0, CUSTOM 0)
MtkCam/MappingMgr: [query] [Dev:2-Mod:AWB(25)] (Idx 1) (PF Preview, SM Preview, Bin 0, P2 0, FLASH 0, APP HDR, FD 1, ZOOM 1, LV 0, CT 0, ISO 0, CUSTOM 0)
```

3、OTP Gain: pregain1

GainSetNum为3，表示有读到3组OTP gain，并且enable了三色温 OTP功能。

三色温OTP enable

```
D awb_algo: [2] AWB Mode = 1
D awb_algo: [2] Stat Config W = 1296  H = 1940
D awb_algo: [CalIllum] GainSetNum 3  CalIllum En 1 CalIllumDbg = 0
D awb_algo: [CalIllum][updatePreGain1Param][GainSetNum=3] =================AWB_3Illum_CompGain_v1p5=============
D awb_algo: [2][Pregain1Param] H value : Unit   0    0    0    0  Golden   0    0    0    0
D awb_algo: [2][Pregain1Param] M value : Unit   0    0    0    0  Golden   0    0    0    0
D awb_algo: [2][Pregain1Param] L value : Unit   0    0    0    0  Golden   0    0    0    0
D awb_algo: [2][Pregain1Param] H Gain  : Unit  928  512  823  Golden  921  512  830  PreGain1   516  512  508
D awb_algo: [2][Pregain1Param] M Gain  : Unit  807  512 1007  Golden  796  512  998  PreGain1M  519  512  517
D awb_algo: [2][Pregain1Param] L Gain  : Unit  689  512 1621  Golden  698  512 1562  PreGain1L  505  512  531
D awb_algo: [2][updateAWBParam] Gain D65 1155 512 734  StatCal 1155 512 734  OutputCal 1164 512 728
D awb_algo: [2] handleAWB AWB Stat Buf 0xd905a000  WH 1296 x 1940  Size 120 x 90  LineSize 2760
```

三色温OTP gain

D65 Gain*PreGain1

4、RGB sum：可通过RGB Sum计算G/R,G/B来反应画面R,G,B的一个比例

faceAWB：打印face 框位置，统计的avgFaceXrYr，并计算face的color ratio。

```
D awb_algo: [FaceAWB][getFDAWBWindowInfo] Idx:0 FaceWin L:39 R:92 Low:20 Hi:53 FaceWinScale L:39 R:92 Low:20 Hi:53
D awb_algo: [2] RGB sum 492825 643242 259627  Error 4  ErrMo 0  Mo 0
D awb_algo: [FaceAWB][getFDAWBWindowContentInfo] ChildBlkNum:1717 WinRatio:100 CentralWeight:1
D awb_algo: [FaceAWB] StatNR En:1 Thr:273
D awb_algo: [FaceAWB] SizeSum:680823 FinalSizeSum:680823 MaxSize:680823 MaxSizeIdx:0
D awb_algo: [FaceAWB] MaxFaceR:42 G:45 B:15
D awb_algo: [FaceAWB] AvgFaceR:42 G:45 B:15 FaceArea_RG:934 BG:327 RB:2853
D awb_algo: [FaceAWB] MgrFaceNum:1 FinalFaceNum:1 FaceDetect:1
D awb_algo: [FaceAWB] AvgFaceXrYr: -6645, -6528, ColorRatio: 935
```

5、Face Assist相关log，打印计算出的P和faceGain。
FaceGain Prev:前一帧 face gain, curr :当前face gain, Target: smooth 之后的face gain.
FaceXY 表示face平均落点的XY，如果出现face assistant异常，可查看face XY是否异常，如果异常可能是face框统计值问题，neutralXY表示根据faceXY 估计当前环境色温的XY位置。

```
awb_algo: [FaceAst-Conf] P0(100), P2(100), P3(100), P4(100)
awb_algo: [FaceAst-Conf]        STB   T   WF   F   CWF   D    S    DF
awb_algo: [FaceAst-Conf] P1 - 0, 93, 100, 80, 100, 80, 100, 100
awb_algo: [FaceAst-Conf] P  - 0, 93, 100, 80, 100, 80, 100, 100
awb_algo: [FaceAst-FaceGain] Prev(323, 512, 967), Curr(326, 512, 954), FaceGain(325, 512, 965)
awb_algo: [FaceAst-EnqChk] i4StableStatus,1, bRet,1
awb_algo: [FaceAst-Enq] TmpBufNum(8), LV(73), XY(-455, -414), TOL(34)
awb_algo: [FaceAst-Record] Case-1: Stable or EnqBufferNotEnough
awb_algo: [FaceAst-Record][TempoInfo] FaceXY(-455, -414), NeutralXY(-266, -478), FaceGain(325, 512, 965)
```

6、打印Normal AWB各色温的统计值，PB：parent Block，WPB: weight parent Block。RGB sum表示该色温下所有落点的统计值累加，XrYr表示RGB转换到XrYr domain的坐标值，每帧各色温的P0，P1，P2和最终的P

```
awb_algo: L1  PB 248  WPB 317346  RGB Sum 12254128 14732219  5397233  Avg  39  46  17  XrYr  -332  -415
awb_algo: L2  PB   9  WPB  10386  RGB Sum   386100   483256   153906  Avg  37  47  15  XrYr  -374  -480
awb_algo: L3  PB 136  WPB 137200  RGB Sum  5930778  8353537  3673340  Avg  43  61  27  XrYr  -208  -416
awb_algo: L5  PB  17  WPB  24358  RGB Sum   419006   774929   502679  Avg  17  32  21  XrYr    39  -390
```

```
awb_algo: P0  STB   0  H  62  WF   0  F  38  CWF   0  D   0  S   0  DF   0
awb_algo: P1  STB 100  H  80  WF  80  F  90  CWF  56  D  95  S  55  DF  60
awb_algo: P2  STB 100  H  89  WF 100  F 100  CWF 100  D 100  S 100  DF 100
awb_algo: P   STB   0  H  56  WF   0  F  44  CWF   0  D   0  S   0  DF   0
```

7、face comp的相关log，
OriAWBGain :当前帧经过face comp之前的gain
FaceAWBTargetGain: face comp计算之后的gain
FinalAWBGain: temporal smooth之后的gain。

```
awb_algo: [FaceAWB][awb_face_comp] OriAWBGain R:307 G:512 B:916 FaceAWBTargetGain R:302 G:512 B:945 FinalAWBGain R:302 G:512 B:928
awb_algo: [FaceAWB] FinalFace RG:1296 BG:821 CurrHue:23653 Sat:42582 TargetHue:15000 Sat:36000 TOLHue:1000 Sat:4000 FinalHue:22640 Sa
```

8、SMOOTH mode：
Target: 当前帧计算的AWB Gain, Prev: 前一帧的final AWB Gain Final: 当前帧经过smooth后的final AWB Gain，这些gain都是没有乘过pregain1和D65 gain的。最终apply到ISP的gain为final gain*outputCal(此log为1164,512,728)

```
[2][SMOOTH][0] AwbState 0  AwbStable 1  Diff 4  StableThr 225  Speed 10  Target 302 512 928  Prev 302 512 926  Final 302 512 928
[2][PV Gain][0] Mode 1  LV 64, Rgain = 686, Ggain = 512, Bgain = 1320
[2] CoreState 0 CCT 3067  F_Idx 100  DF_Idx 100  Target  686, 512, 1320  Output 686, 512, 1320  FullWB 632, 512, 1445
```

9、打印每帧计算的最终AWB Gain（final gain乘上D65 Gain和pregain1）和当前LV，可定位AWB跳变的帧

```
06:07:40.044  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 698, Ggain = 512, Bgain = 1249
06:07:40.069  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 702, Ggain = 512, Bgain = 1283
06:07:40.102  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 702, Ggain = 512, Bgain = 1283
06:07:40.131  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 714, Ggain = 512, Bgain = 1266
06:07:40.165  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 714, Ggain = 512, Bgain = 1262
06:07:40.197  8686 14833 D awb_algo: [2][PV Gain][0] Mode 1  LV 73, Rgain = 714, Ggain = 512, Bgain = 1262
```

normal AWB/ColorEngine调试部分请参考6763 workshop case share文档

文档路径：

DCC>HW >Common Design Notes> Camera >Smart Phone>MT6763**>**
CameraWorkshop>**/MT6757_MT6763_AWB_COLOR_CASE_SHARE.pdf**

everyday genius