CONFIDENTIAL B

MT6771 AE Tuning Introduction

有奖问答

- Q1： AE tuning之前需要注意什么？

有奖问答

- Q1： AE tuning之前需要注意什么？

1.确定shutter/gain线性度pass.

2.确定shading有calibration pass.

3.确认EV offset正确性.

4.确认最小帧率和最大ISO.

Outline

- Introduction
- AE
  - Statistics
  - Decide Target Brightness
  - Touch/Face
- LCE/DCE

What is AE

- Brightness of environment can be adaptive by human eye so brightness of human vision is always suitable.
- Brightness of camera vision is based on exposure setting. Camera brightness isn't suitable unless exposure changes to suitable exposure setting. Auto exposure, AE, plays a role in suitable exposure setting modification automatically.

Human vision

Exposure setting A

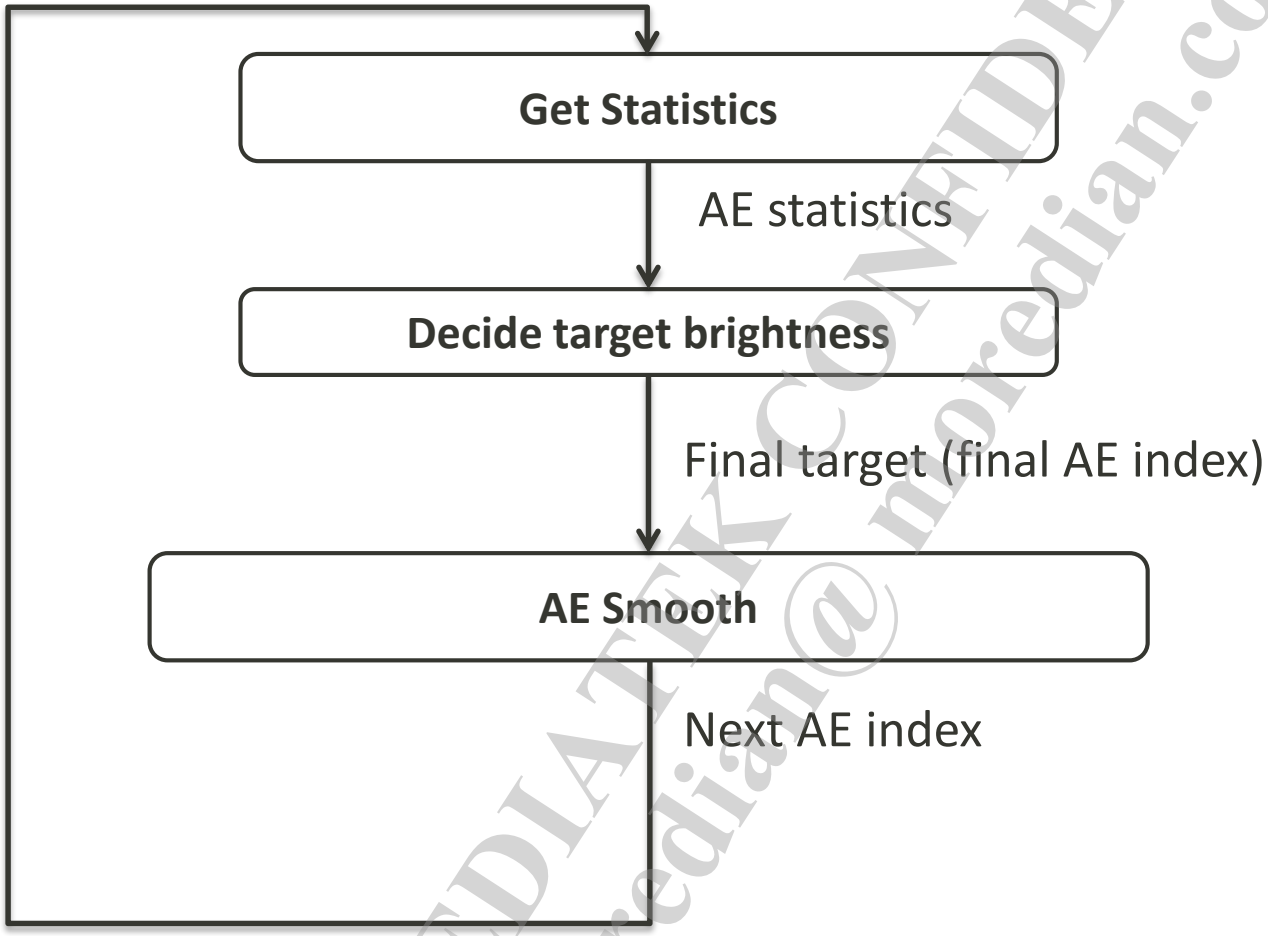
image too bright or too dark

Auto Exposure

As Human vision

Algorithm

Block Diagram



Get Statistics
1. Get info. from HW
2. Calculate brightness info. like CWV

Decide target brightness
1. Calculate final target brightness by several mechanisms to handle different scenes
2. Calculate brightness difference between CWV and target brightness, and whether needs convergence
3. Converse brightness difference to Final AE index (after 3 frame convergence)

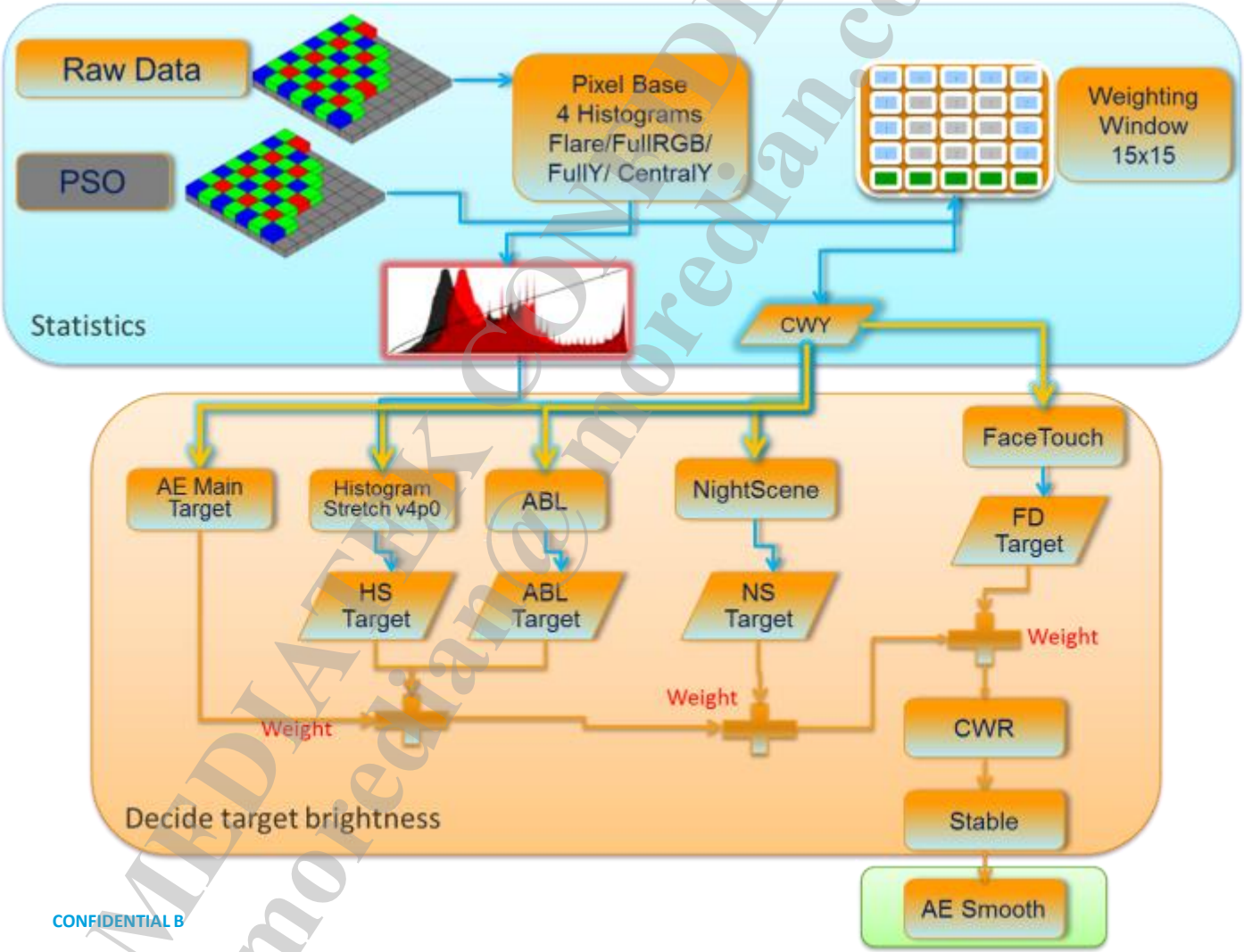
AE Smooth
1. Decide AE index for each of following several frames

Design Formulas

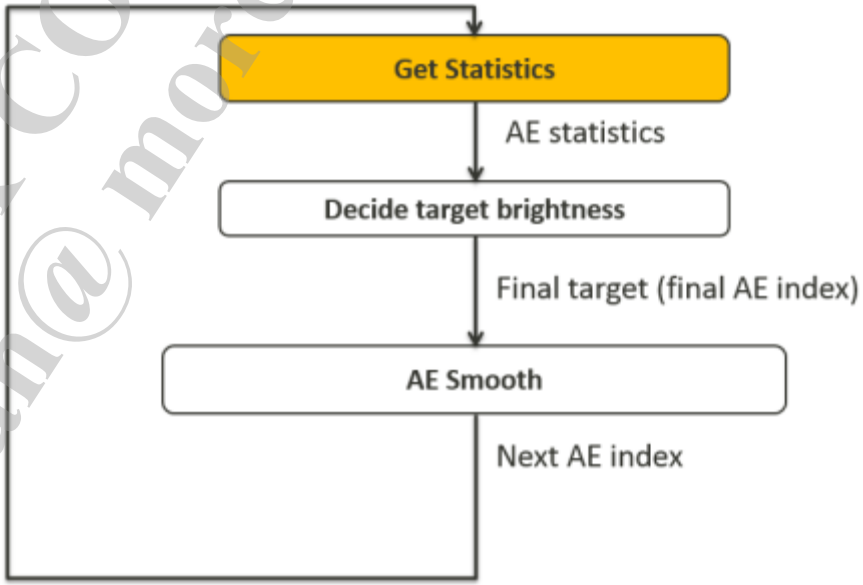
$$target = \frac{THD}{Y} * \mathrm{CWV}$$

1. THD: from parameter.
2. Y: from N% of histogram brightness value.
3. CWV: Centre Weighting Value.

Block Diagram (Detail)

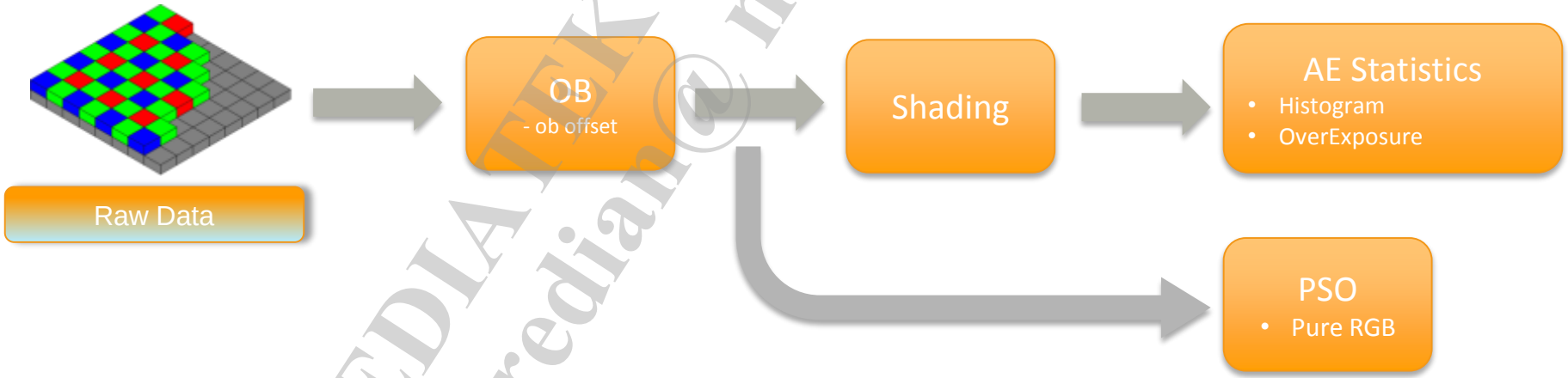


Statistics



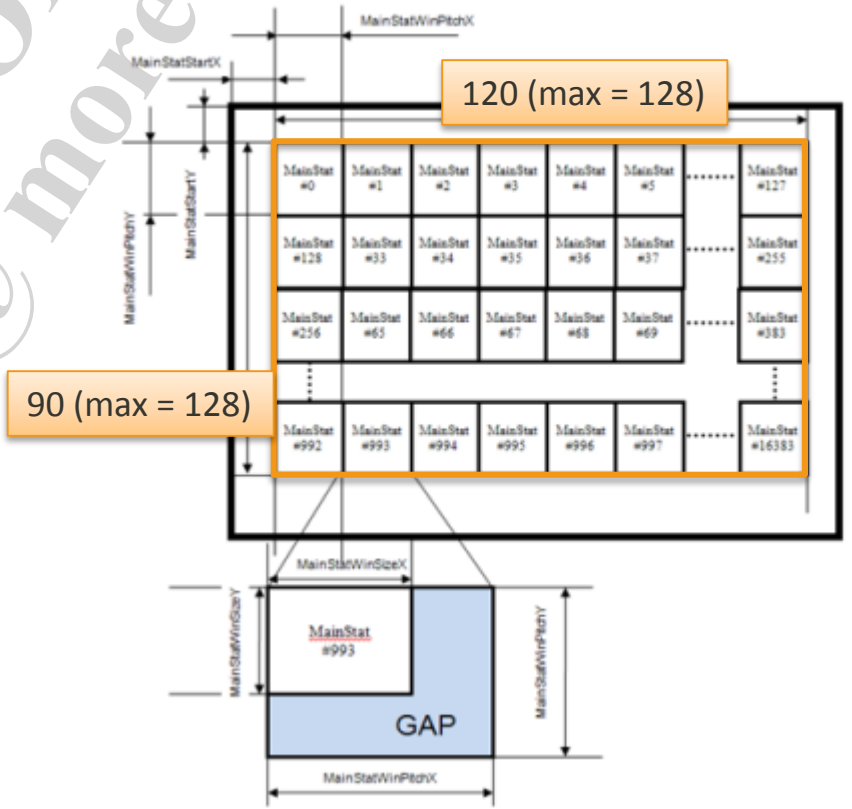
AE Statistics

- Purpose
  - To get brightness information for AE algorithm
- AE statistics
  - Including 4 Histograms and Y info. for normal AE metering
- PSO
  - Mechanism to get pure statistics, more precise at over-exposure info.



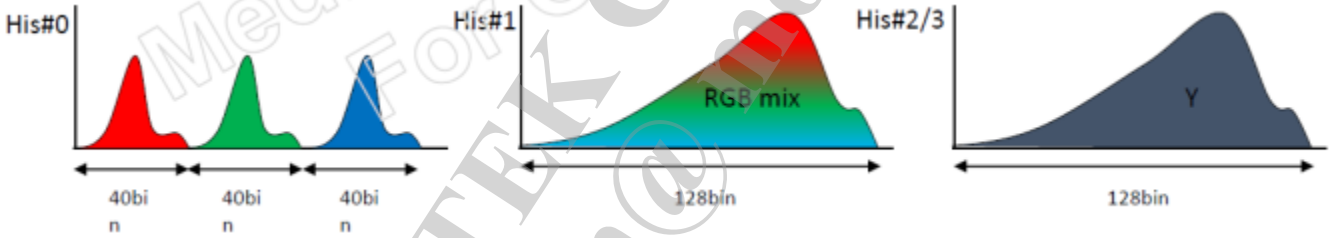
AE Statistics

- AE Statistics Window Configuration
  - Maximum window number: 128 x 128 windows
  - Current FW configuration: 120 x 90 windows
  - Average of R, G, B and error pixel count for each window
    - Error pixel count : Over-exposed or under-exposed pixels and is used in Error Handling



AE Statistics

- AE Histogram
  - 4 128-bin histograms
    - Pixel-based histogram
    - The ROI of each histogram can be configured separately
    - Histogram#0(Flare): 40 bins for R, G, and B each, cover 0~39 in 8bit level
    - Histogram#1(FullRGB): for RGB mixing (shared bins), cover 0~255 in 8bit level
    - Histogram#2(CentralY): for Y of ¼-central image, cover 0~255 in 8bit level
    - Histogram#3(FullY): for Y of full image, cover 0~255 in 8bit level

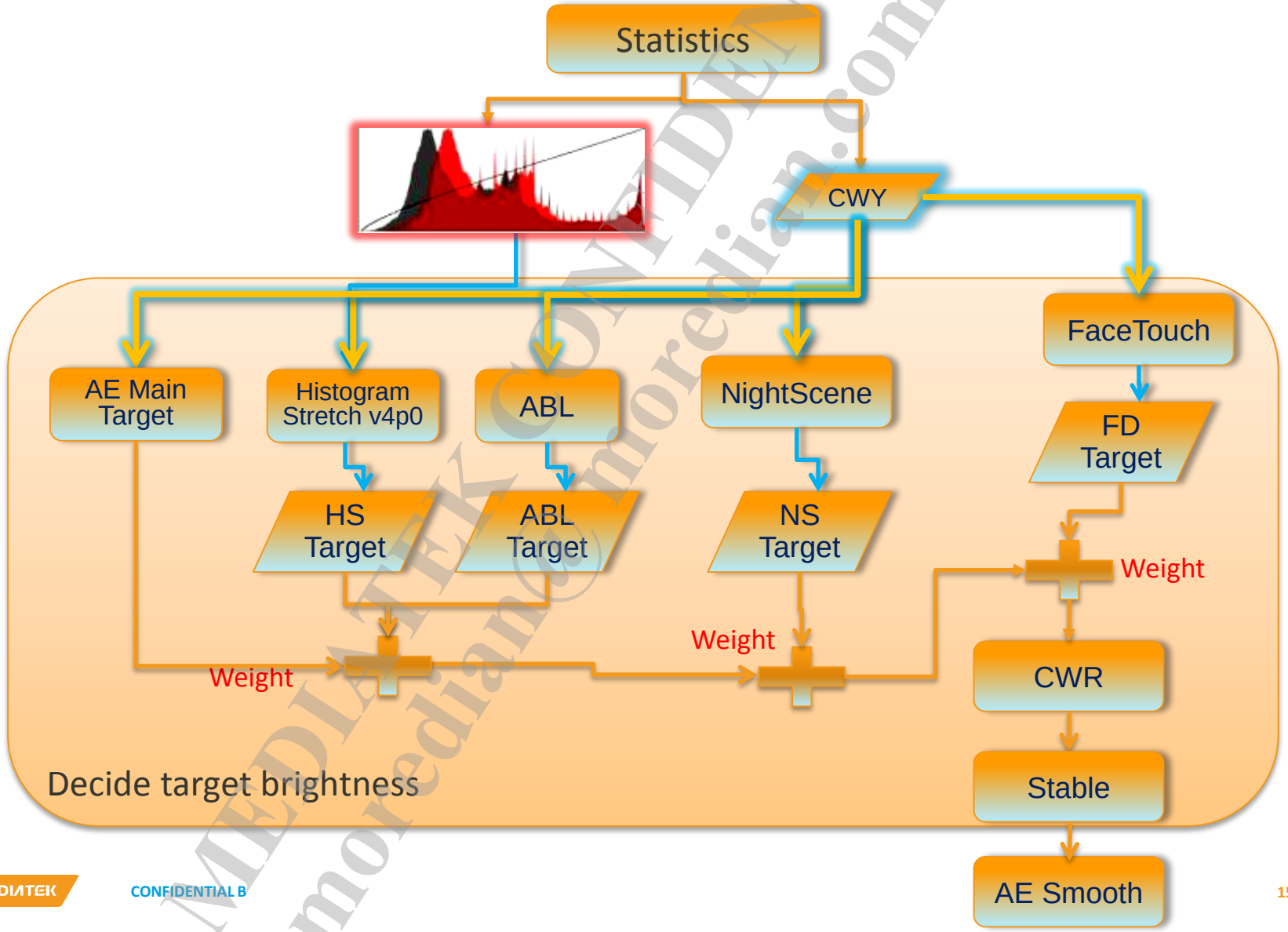


Decide Target Brightness

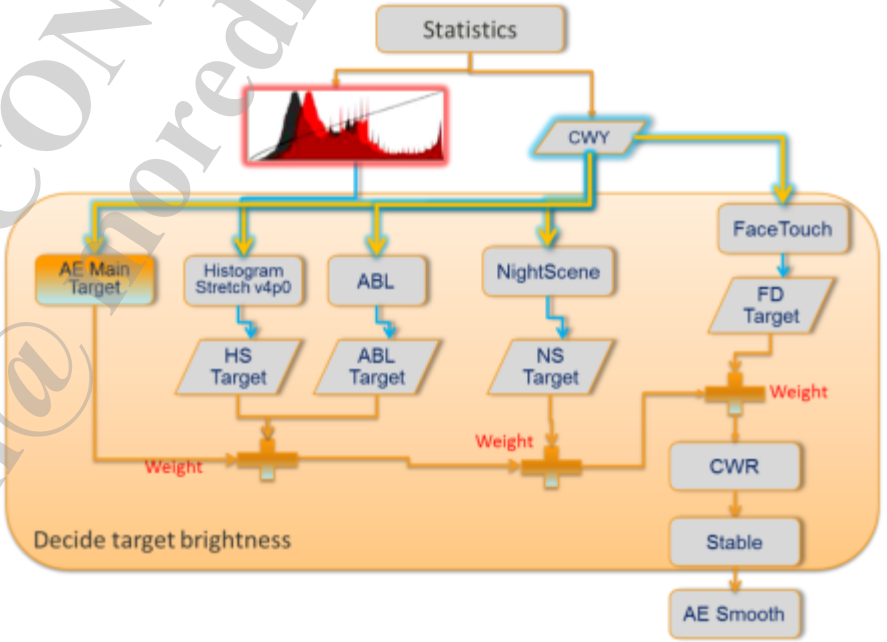
Target brightness decision mechanism

- MTK AE algorithm uses several mechanisms to decide final target
- Different scenes needs different target brightness
- If using the same target, you may meet some issues
  - Too dark at white scene
  - Too bright at dark scene
  - Too dark at front object on backlight environment
  - Too bright in the night
  - Face brightness is not well if background is bright enough

AE v4.0 Target Decision



Main Target



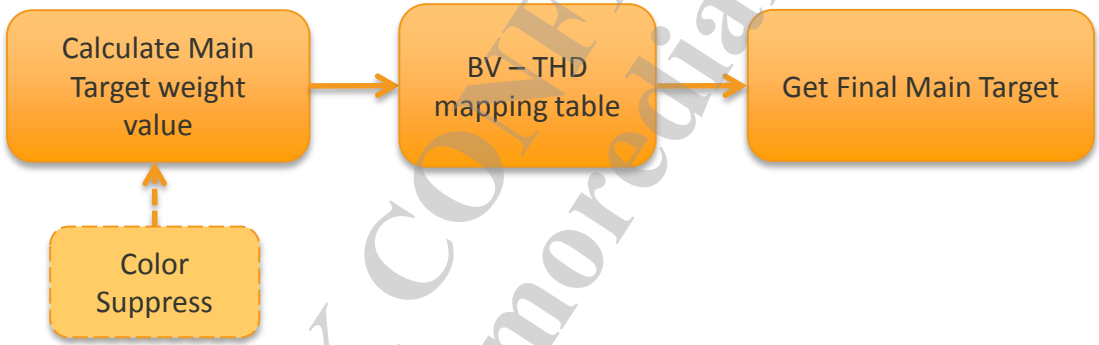
Scenes for Main Target

- Basic metering mechanism to control brightness for most scenes
  - Except pure Face scene and pure night scene

- Key features
  - Decide target based on AE_Target and BV ratio
  - Suppress brightness for scenes with large area single color, to avoid color distortion

Main Target

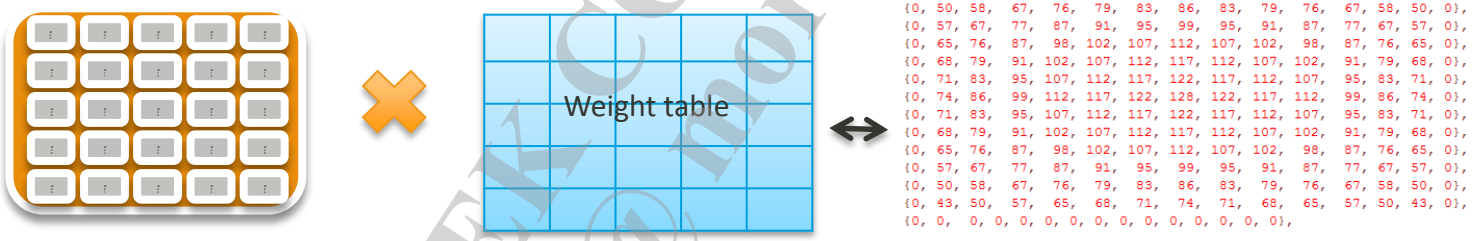
- Main target flow



Main Target

- Calculate weighting value (MTWV)
  - 15 X 15 weighting table
  - The distribution is like Gaussian Distribution

MTWV = (Block[1].Y * Weight[1] + Block[2].Y * Weight[2] + … + Block[N].Y * Weight[N]) / (Weight[1] + … + Weight[N])

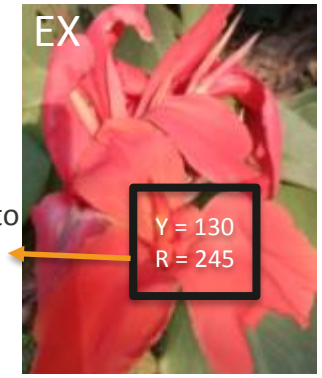



```
{0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 43, 50,  57,  65,  68,  71,  74,  71,  68,  65,  57, 50, 43, 0},
{0, 50, 58,  67,  76,  79,  83,  86,  83,  79,  76,  67, 58, 50, 0},
{0, 57, 67,  77,  87,  91,  95,  99,  95,  91,  87,  77, 67, 57, 0},
{0, 65, 76,  87,  98, 102, 107, 112, 107, 102,  98,  87, 76, 65, 0},
{0, 68, 79,  91, 102, 107, 112, 117, 112, 107, 102,  91, 79, 68, 0},
{0, 71, 83,  95, 107, 112, 117, 122, 117, 112, 107,  95, 83, 71, 0},
{0, 74, 86,  99, 112, 117, 122, 128, 122, 117, 112,  99, 86, 74, 0},
{0, 71, 83,  95, 107, 112, 117, 122, 117, 112, 107,  95, 83, 71, 0},
{0, 68, 79,  91, 102, 107, 112, 117, 112, 107, 102,  91, 79, 68, 0},
{0, 65, 76,  87,  98, 102, 107, 112, 107, 102,  98,  87, 76, 65, 0},
{0, 57, 67,  77,  87,  91,  95,  99,  95,  91,  87,  77, 67, 57, 0},
{0, 50, 58,  67,  76,  79,  83,  86,  83,  79,  76,  67, 58, 50, 0},
{0, 43, 50,  57,  65,  68,  71,  74,  71,  68,  65,  57, 50, 43, 0},
{0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
```

Tuning custom

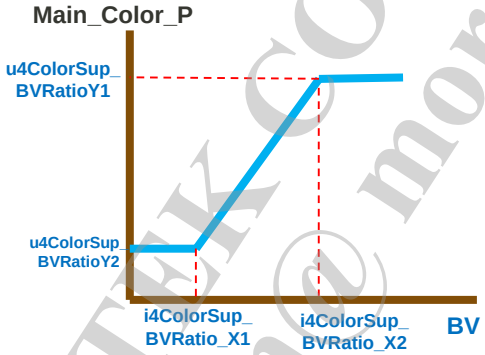
```
static strMainTargetCFG g_MainTargetCFG =
{
    TRUE,               //bEnableAEMainTarget
    1024,               //u4MainTargetWeight
    {                       //u4MainTargetWeightTbl[15][15] : Gau weight table
        {0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
        {0, 43, 50,  57,  65,  68,  71,  74,  71,  68,  65,  57, 50, 43, 0},
        {0, 50, 58,  67,  76,  79,  83,  86,  83,  79,  76,  67, 58, 50, 0},
        {0, 57, 67,  77,  87,  91,  95,  99,  95,  91,  87,  77, 67, 57, 0},
        {0, 65, 76,  87,  98, 102, 107, 112, 107, 102,  98,  87, 76, 65, 0},
        {0, 68, 79,  91, 102, 107, 112, 117, 112, 107, 102,  91, 79, 68, 0},
        {0, 71, 83,  95, 107, 112, 117, 122, 117, 112, 107,  95, 83, 71, 0},
        {0, 74, 86,  99, 112, 117, 122, 128, 122, 117, 112,  99, 86, 74, 0},
        {0, 71, 83,  95, 107, 112, 117, 122, 117, 112, 107,  95, 83, 71, 0},
        {0, 68, 79,  91, 102, 107, 112, 117, 112, 107, 102,  91, 79, 68, 0},
        {0, 65, 76,  87,  98, 102, 107, 112, 107, 102,  98,  87, 76, 65, 0},
        {0, 57, 67,  77,  87,  91,  95,  99,  95,  91,  87,  77, 67, 57, 0},
        {0, 50, 58,  67,  76,  79,  83,  86,  83,  79,  76,  67, 58, 50, 0},
        {0, 43, 50,  57,  65,  68,  71,  74,  71,  68,  65,  57, 50, 43, 0},
        {0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    },
```

Main Target



Use R instead of Y to calculate MTWV

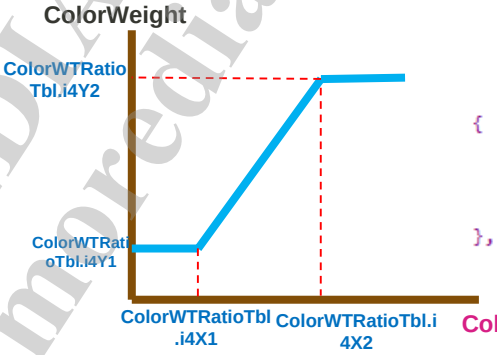
- Calculate weighting value (MTWV)
  - When **Color Suppression** enabled, modify the method to calculate MTWV
  - **Color Suppression** – block base (Optional)
    - To avoid color distortion because some color channel is over 255
    - Using R / G / B per-block info to calculate accuracy target instead of using Y for high Main_Color_P
    - Main_Color_P



Tuning custom

```
{   //ColorSupBVRatioTbl
    4000,                    // ColorSupBVRatioTbl.i4X1 :BV
    0,                  // ColorSupBVRatioTbl.i4Y1 :Ratio
    6100,                // ColorSupBVRatioTbl.i4X2  :BV
    1024                  // ColorSupBVRatioTbl.i4Y2 :Ratio
},
```

    - ColorWeight (Optional)



Tuning custom

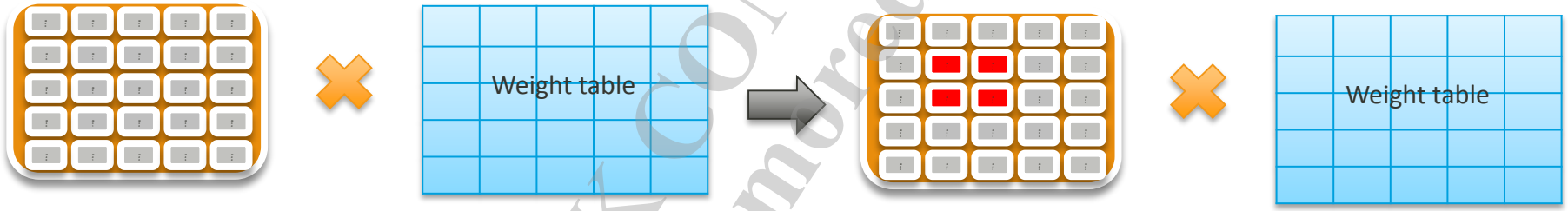
```
{   //ColorWTRatioTbl
    130,                    // ColorWTRatioTbl.i4X1 :ColorDiff
    768,                 // ColorWTRatioTbl.i4Y1 :u4MainTargetWeight
    300,                 // ColorWTRatioTbl.i4X2  :ColorDiff
    1024                  // ColorWTRatioTbl.i4Y2 :u4MainTargetWeight / 2
},
```

Main Target

- Calculate weighting value (MTWV)
  - **Color Suppression** – block base (Optional)
    - Calculate MTWV

If Main_Color_P * ColorWeight * uColorSuppressRatioR/G/B * R/G/B > Y
→ Use Max R/G/B instead of Y for the block to calculate MTWV

Ex.



MTWV = (Block[1].Y * Weight[1] + … + Block[7].R * Weight[7] + … + Block[N].Y * Weight[N]) / (Weight[1] + … + Weight[N])

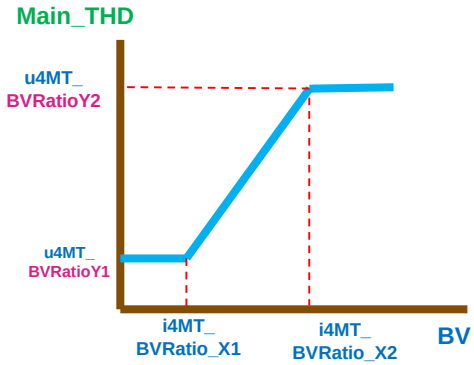
Tuning custom

```
TRUE,                   //bEnableColorWTRatio
{   //ColorWTRatioTbl
    0,                      // ColorWTRatioTbl.i4X1 :ColorDiff
    600,                    // ColorWTRatioTbl.i4Y1 :u4MainTargetWeight
    100,                  // ColorWTRatioTbl.i4X2  :ColorDiff
    1024                    // ColorWTRatioTbl.i4Y2 :u4MainTargetWeight / 2
},
100,                    //uColorSupRatioR : 0~100
100,                      //uColorSupRatioB : 0~100
0,                      //uColorSupRatioB : 0~100
{   //ColorSupBVRatioTbl
    4000,                   // ColorSupBVRatioTbl.i4X1 :BV
    0,                    // ColorSupBVRatioTbl.i4Y1 :Ratio
    6000,                   // ColorSupBVRatioTbl.i4X2  :BV
    1024                    // ColorSupBVRatioTbl.i4Y2 :Ratio
},
};
```

Adjust ratio to do color suppression for each color channel

Main Target

- BV-THD mapping table
  - Map Main_THD by BV, for calculating Final Main_Target



Tuning custom

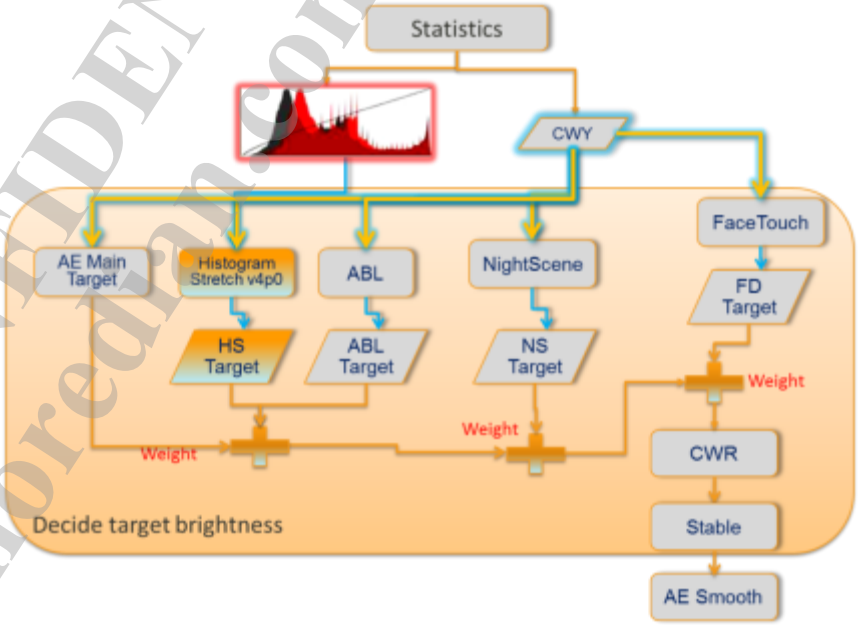
```
{   // TargetBVRatioTbl
    1000,
    49,
    8500,
    72
},
```

```
// TargetBVRatioTbl.i4X1 :BV
    // TargetBVRatioTbl.i4Y1 :Target
// TargetBVRatioTbl.i4X2  :BV
    // TargetBVRatioTbl.i4Y2 :Target
```

- Final Main Target

Final Main_Target = CWV * Main_THD / MTWV

HISTOGRAM STRETCH V4.0



Scenes for Histogram Stretch(1) Raise Brightness

If only AE target is considered, the brightness is lower.

With HS

After HS, the final target is stretched, and the brightness is suitable.

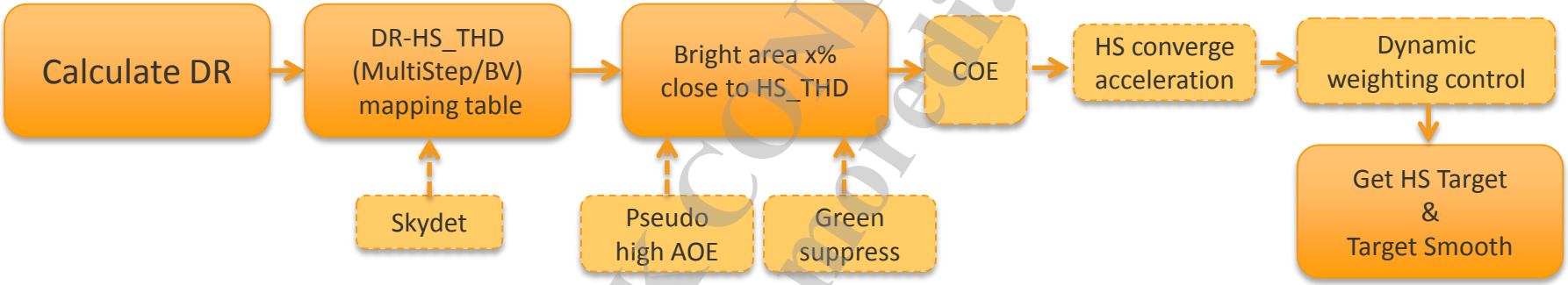
The brightness of a highly bright area in an image is raised to a certain level.

Scenes for Histogram Stretch(2) Reduce Brightness

- For a over-exposed area, HS control the brightness of highly bright area to a certain level to avoid over exposure.

Histogram Stretch v4p0

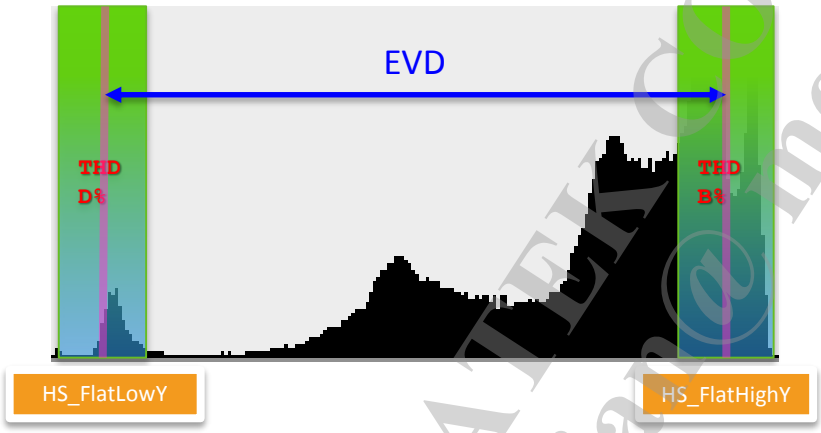
- HS v4p0 flow



Histogram Stretch v4p0

- HS dynamic range (EV Diff or EVD)
  - Brightness difference between bright part and dark part means HS dynamic range
  - Used to determine whether the histogram distributes flatly
  - HS_FlatHighY - HS_FlatLowY (by u4FlatBrightPcent & u4FlatDarkPcent)

不建議調

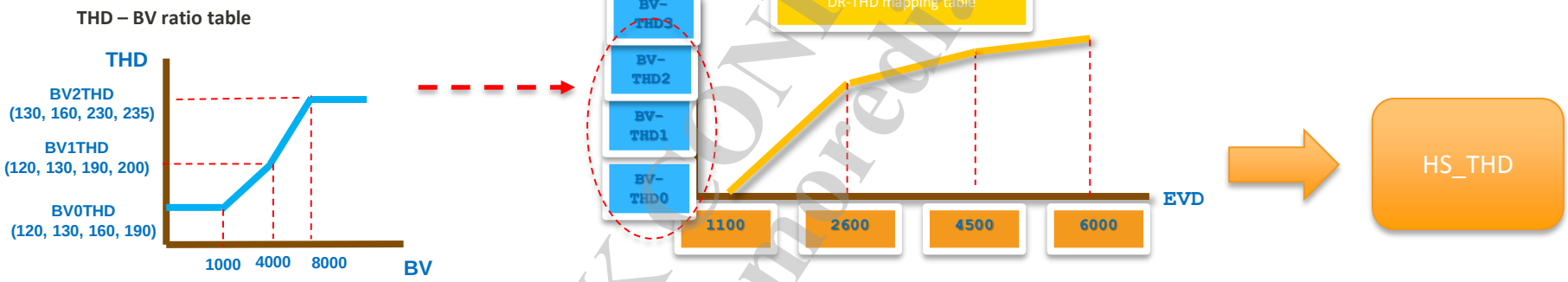


NVRAM

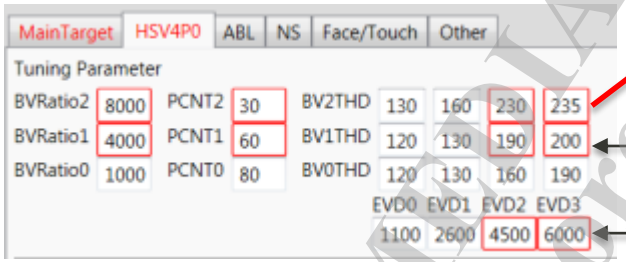
```
//rMeteringSpec
{
    //rHS_Spec
    {
        TRUE,//bEnableHistStretch
        1024,//u4HistStretchWeight
        60, //50, //20,//u4Pcent
        160, //166,//176,//u4Thd
        75, //54, //74,//u4FlatThd

        120,//u4FlatBrightPcent
        120,//u4FlatDarkPcent
```

Histogram Stretch v4p0

- DR-HS_THD (MultiStep/BV) mapping table
  - Larger contrast(larger dynamic range) uses larger threshold in principle.
  - HS_THD is interpolated depending on different EVD and BV



Ex.
BV: 5200
EVD: 4600



Interpolate HS_THD

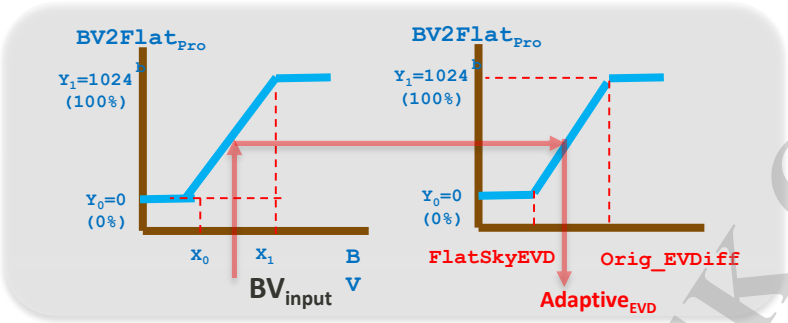
Tuning custom

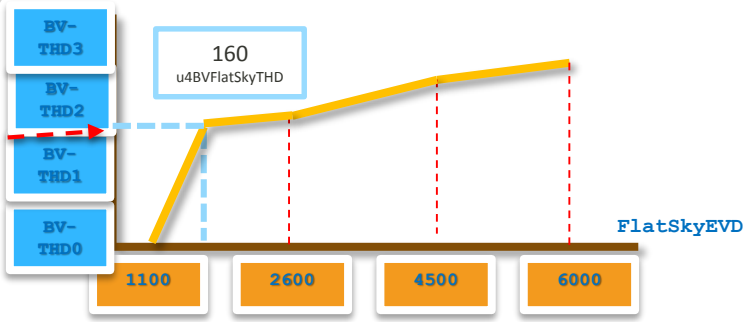
```
static strHS4p0CFG g_HS4p0CFG =
{
    TRUE,                       //bEnableHS4p0
    1024,                       //u4HS4p0Weight
    3,                              //u4BVSize ,max16
    4,                              //u4EVDSize ,max16
    {1000, 4000, 8000},         //i4HS_BVRatio, length = u4BVSize
    {80, 60, 30},       //u4HS_PcntRatio, length = u4BVSize, 80 equal 8%
    {1100, 2600, 4500, 6000},    //u4HS_EVDRatio , length = u4EVDSize
                                                //u4HSTHDRatioTbl[u4BVSize][u4EVDSize] , HS THD
    //EVD-THD
    {120, 130, 160, 190},
    //BV1-THD
    {120, 130, 190, 200},
    //BV2-THD
    {130, 160, 230, 235},
```

Histogram Stretch v4p0

- Sky detect
  - Adaptive fine tune mapping table HS_THD
    - For scenes that sky is not bright enough due to lower HS_THD by flat sky scene

建議調整BV ratio控制落入區間
亮度不足則調大BVFlatTHD





Tuning custom

```
static strHSFlatSkyCFG g_HSbyRealBV =
{
    TRUE,                       //bEnableFlatSkyEnhance
    2,                            //uSkyDetInThd  : 2%
    5,                            //uSkyDetOutThd : 5%
    160,                    //u4FlatSkyTHD
    1500,                   //u4FlatSkyEVD
    {8000,  11000}, //u4BVRatio_X : BV
    {0, 1024}               //u4BVRatio_Y : Ratio
};
```

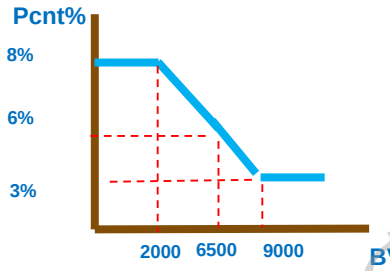
Larger uSkyDetInThd raises the probability of being detected as sky

Larger uSkyDetOutThd raises the probability of being detected as non-sky state from sky state

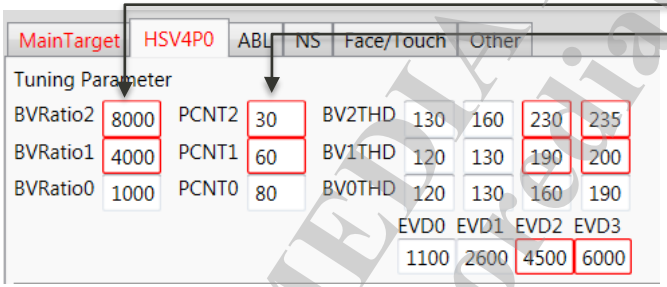
Histogram Stretch v4p0

- Bright area x% close to HS_THD
  - Determine HS_THD by interpolating using BVRatio

Pcnt % – BV ratio table



Tuning custom

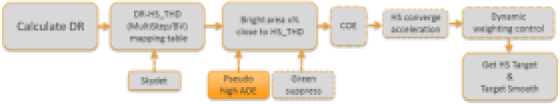



```
static strHSv4p0CFG g_HSv4p0CFG =
{
    TRUE,                   //bEnableHSv4p0
    1024,                   //u4HSv4p0Weight
    3,                          //u4BVSize ,max16
    4,                          //u4EVDSize ,max16
    {1000, 4000, 8000},     //i4HS_BVRatio, length = u4BVSize
    {80, 60, 30},       //u4HS_PcntRatio, length = u4BVSize, 80 equal 8%
    {1100, 2600, 4500, 6000},     //u4HS_EVDRatio , length = u4EVDSize
    {                                   //u4HSTHDRatioTbl[u4BVSize][u4EVDSize] , HS THD
        //BV0-THD
        {120, 130, 160, 190},
        //BV1-THD
        {120, 130, 190, 200},
        //BV2-THD
        {130, 160, 230, 235},
    },
```

Histogram Stretch v4p0

- HS Target
  - calculate HS_Y (by u4HS_PcntRatio)
  - Final HS target :

HS_Target = CWV * HS_THD / HS_Y

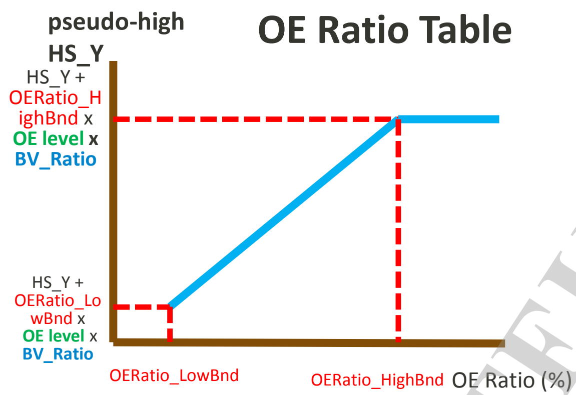


Histogram Stretch v4p0

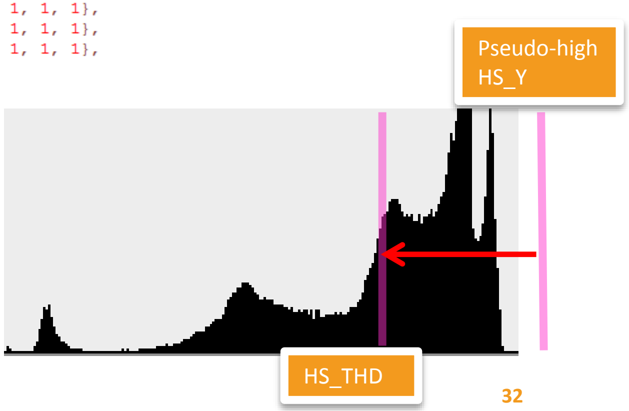
- Pseudo-high AOE
  - Effective when enough overexposure(OE) pixels, larger BV are achieved
  - Calculate a pseudo-high value(> HS_Y) used to replace HS_Y so as to reduce the final HS target
  - To prevent too much overexposure pixels

Tuning custom

```
static strOverExpoAOECFG g_strOverExpoAOECFG =
{
    TRUE,                                           // bEnableOverExpoAOE
    TRUE,                                          //bEnableHistOverExpoCnt
    20,                                              //u4OElevel : pseudo-high strength 6x
    1,                                                //u4OERatio_LowBnd : 2% over exposure area
    20,                                               //u4OERatio_HighBnd : 20% overexposure area
    {4500,   6000},                                //i4OE_BVRatio_X[2]
    {0,      1024},                                  //u4OE_BVRatio_Y[2]
    {                                                  //u4OEWeight[15][15] : Spatial OE weight (avg mode)
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
        {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1},
    }
};
```



Bright area x% close to HS_THD → Pseudo high AOE → Psuedo-high HS_Y > HS_Y



Histogram Stretch v4p0

- Green Suppress (optional)
  - Used to reduce brightness of scenes with large green area
  - Green ratio

G/R > u4HsGreenGRRatio &
G/B > u4HsGreenGBRatio → G block

GreenRatio = G_block_number / total_block_number

Tuning custom

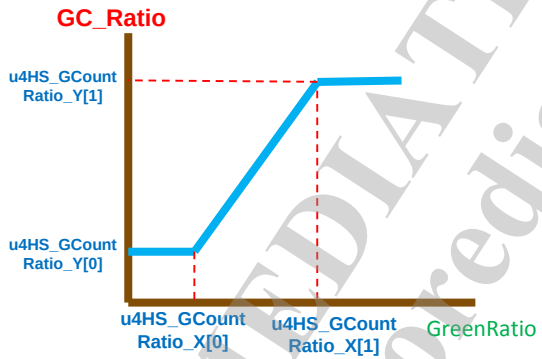
g_HSv4p0CFG

```
FALSE,          //bEnableGreenSupress;
120,            //uGreenGRRatio;
130,            //uGreenGBRatio;
70,             //uSupRatio;
{256,512},      //u4HS_GCountRatio_X
{0,1024},       //u4HS_GCountRatio_Y
```

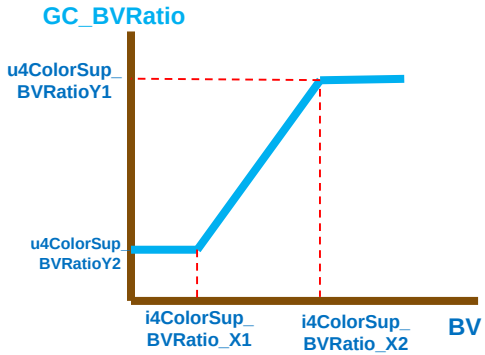
g_MainTargetCFG

```
{   //ColorSupBVRatioTbl
    5000,                   // ColorSupBVRatioTbl.i4X1 :BV
    0,                  // ColorSupBVRatioTbl.i4Y1 :Ratio
    6500,                // ColorSupBVRatioTbl.i4X2  :BV
    1024                  // ColorSupBVRatioTbl.i4Y2 :Ratio
},
```

- GC_Ratio



- GC_BVRatio



Histogram Stretch v4p0

- Green Suppress (optional)
  - GSupTHD

GSupTHD = HS_THD * uSupRatio / 100

GSupTHD = [HS_THD * (1024 - GC_Ratio) + GSupTHD * GC_Ratio] / 1024

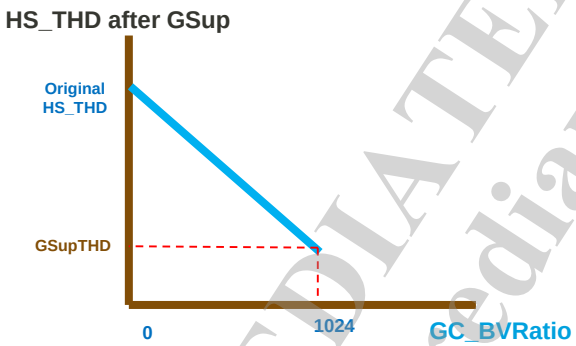
  - HS_THD after Green Suppress

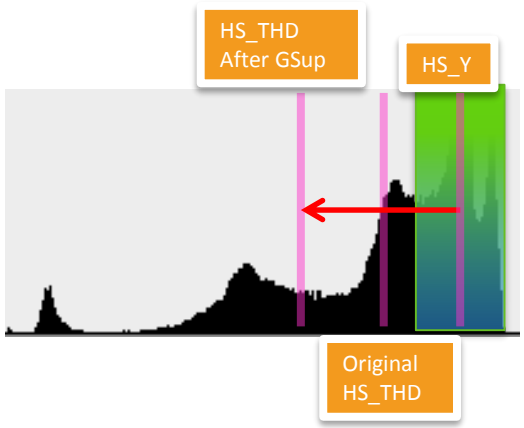
HS_THD aft GSup = HS_THD * (1024 - GC_BVRatio) + GC_BVRatio * GSupTHD

Tuning custom

g_HSv4p0CFG

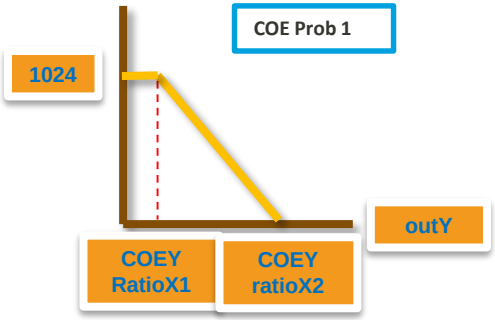
```
FALSE,          //bEnableGreenSupress;
120,            //uGreenGRRatio;
130,            //uGreenGBRatio;
70,             //uSupRatio;
{256,512},      //u4HS_GCountRatio_X
{0,1024},       //u4HS_GCountRatio_Y
```

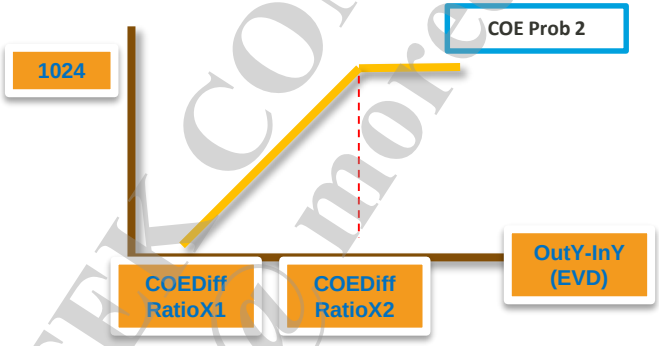




Histogram Stretch v4p0

- COE (Central Over Exposure)
  - Reduce the HS target to avoid central over exposure
- COEP
  - COEP = Prob 1 * Prob2
  - COEP Prob 1
    - OutY = 5x5 block without M22(center)
  - COEP Prob 2
    - InY = M22 (center block)





- COEP Target
  - Using CentralYHist to calculate COE_Y (by u4COEPcent)

COE_Target = CWV * u4COE_THD / COE_Y

- HS Target after COE

HS_Target_aftCOE = (1024 – COEP) * HS_Target + COEP * COE_Target

NVRAM

```
//rAOE_Spec
{
    TRUE, //bEnableAntiOverExposure
    1024, //u4AntiOverExpWeight
    10,     //u4Pcent
  220,//  200,  //u4Thd

    TRUE, //bEnableCOEP
    1,     //u4COEPcent
    115,  //u4COEThd
    0,  // u4BVCompRatio
    //sCOEYRatio;      // the outer y ratio
    {
        23,    //i4X1
        1024,  //i4Y1
        47,    //i4X2
        0      //i4Y2
    },
    //sCOEDiffRatio;  // inner/outer y difference ratio
    {
        1000, //i4X1
        0,    //i4Y1
        2500, //i4X2
        1024   //i4Y2
    }
```

InY & OutY :

Histogram Stretch v4p0

- HS Converge Acceleration (optional)
  - 成立條件

亮部夠亮: HS_FlatHighY >= u4HS_FHY_Hbound

整體夠亮: CWV >= u4CWV_HBound

HS Target在可調整範圍內: u4HS_ACCTarget < HS Target <= u4HS_TargetHBound

  - 以u4HS_ACCTarget取代原本HS target, 藉由降低HS target來加速由亮至目標的收斂速度

Tuning custom

```
static strNonCWRAcc g_AENonCWRAcc =
{
    FALSE,          //bNonCWRAccEnable enable
    255,
    160,
    160,            //it should  be <=hs_THD
    58,             //it should keep in target~hs_flat
};
```

```
typedef struct {
    MBOOL   bNonCWRAccEnable;
    MUINT32 u4HS_FHY_HBound;
    MUINT32 u4CWV_HBound;
    MUINT32 u4HS_TargetHBound;
    MUINT32 u4HS_ACCTarget;
} strNonCWRAcc;
```

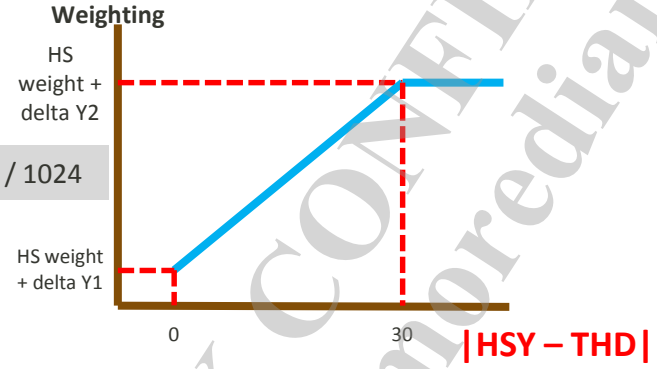
Histogram Stretch v4p0

- Dynamic weighting control (optional)
  - Much far away THD → larger weighting
  - Large OE_Ratio (Pseudo-high) → larger weighting

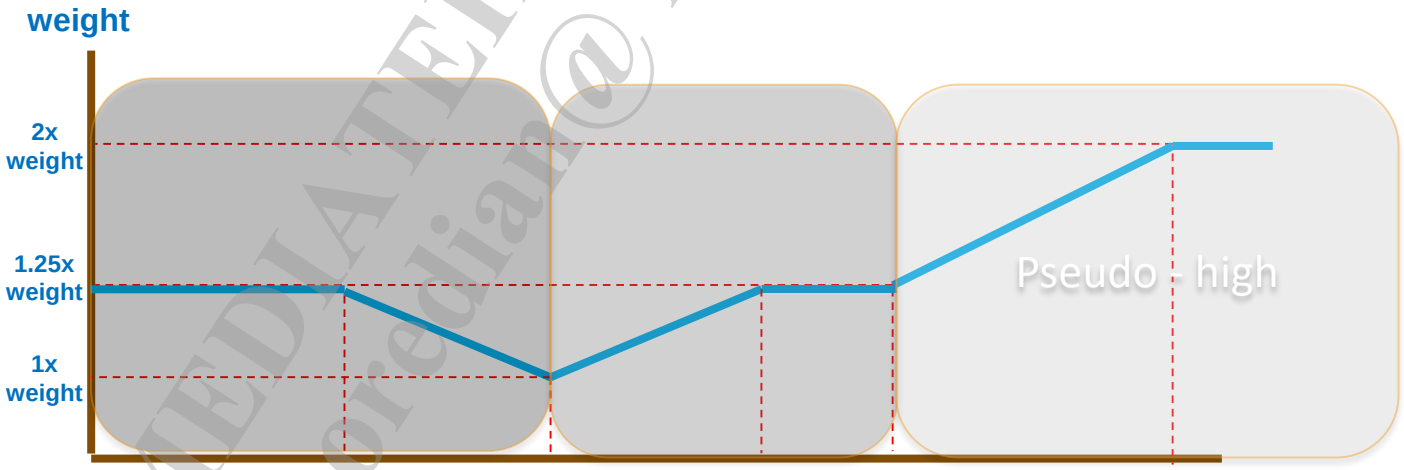
Tuning custom

```
{//HSv4p0CFG
    TRUE,                       //bEnableHSv4p0
    1024,                       //u4HSv4p0Weight
    6,                              //u4BVSize ,max16
    7,                              //u4EVDSize ,max16
    {-500, 1000, 3500, 6500, 8000, 11000},      //i4HS_BVRatio,
    {50, 50, 35, 25, 20, 10},     //u4HS_PcntRatio, length = u4BV
    {500, 1100, 2500, 3500, 5000, 6500, 8000},     //u4HS_EVDR
    {                                                //u4HSTHDRatioTbl[
        //BV0-THD
        {120, 130, 140, 140, 150, 165, 160},
        //BV1-THD
        {120, 140, 155, 160, 165, 175, 185},
        //BV2-THD
        {130, 140, 160, 170, 170, 190, 205},
        //BV3-THD
        {130, 160, 175, 170, 170, 200, 210},
        //BV4-THD
        {140, 150, 175, 175, 200, 205, 215},
        //BV5-THD
        {145, 150, 185, 205, 210, 210, 220},
    },
    TRUE,                       //bEnableDynWTRatio
    {       //DynWTRatioTbl
        0,                                  // DynWTRatioTbl.i4X1 : m
        0,                                    // DynWTRatioTbl.i4Y1 :
        30,                                 // DynWTRatioTbl.i4X2  :max
        384,                                  // DynWTRatioTbl.i4Y2 : a
    }
```

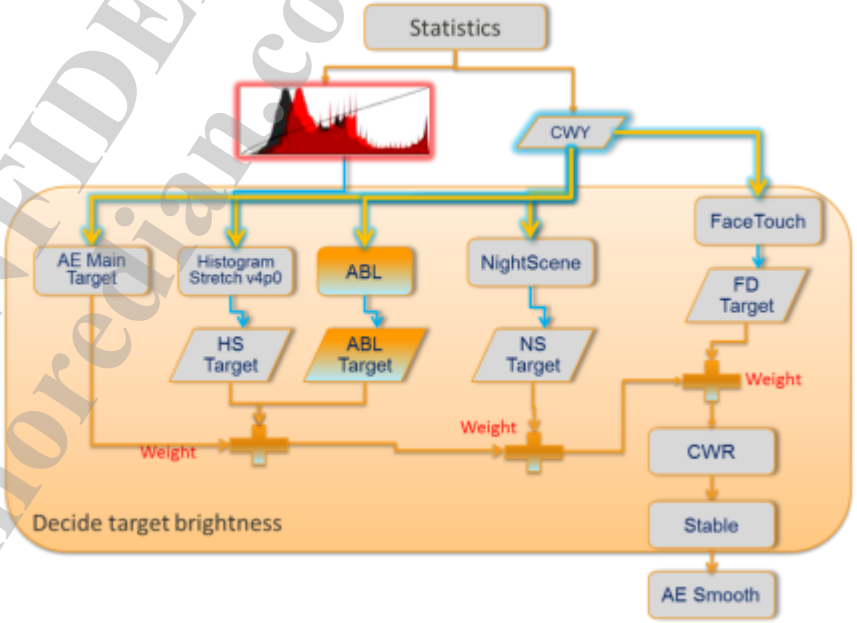
Delta Y = HS weighing *DynWTRatioTbl.i4Y / 1024



Weighting control table

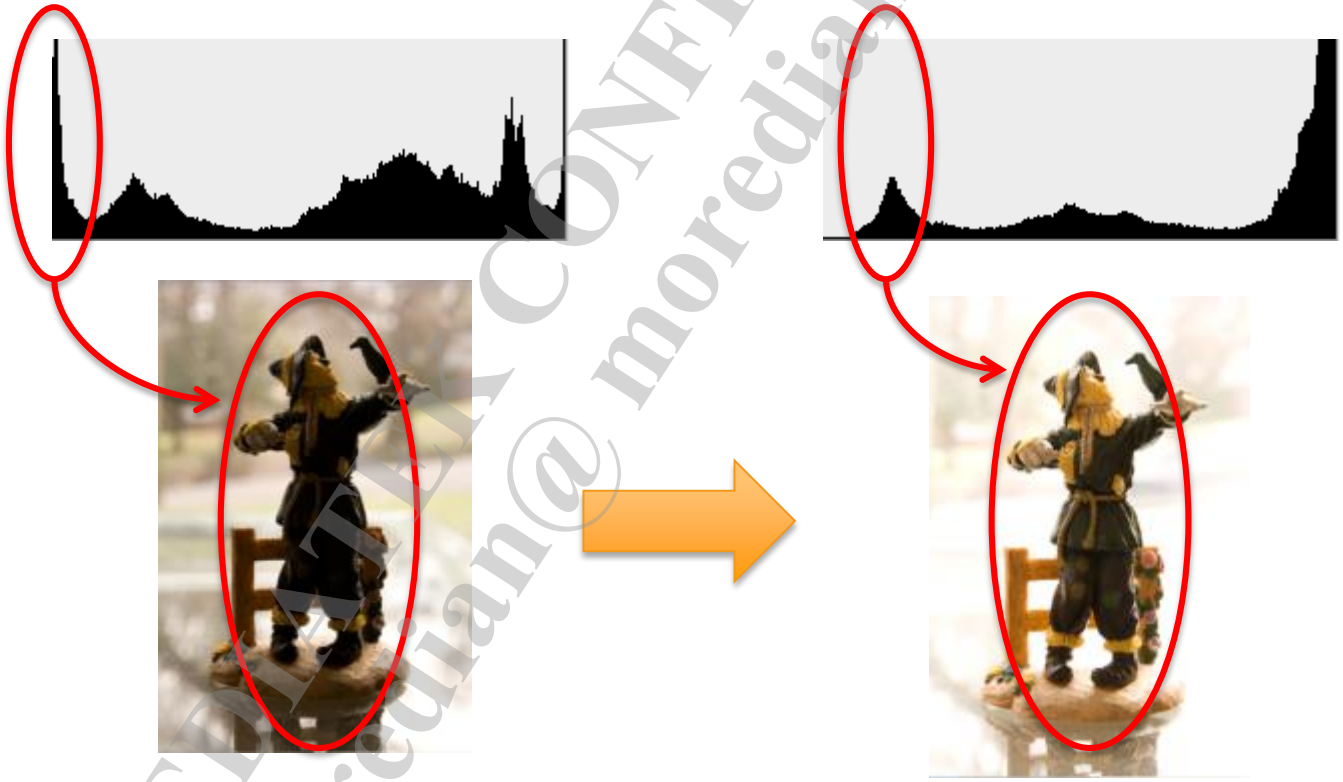


ABL (ANTI-BACK LIGHT)



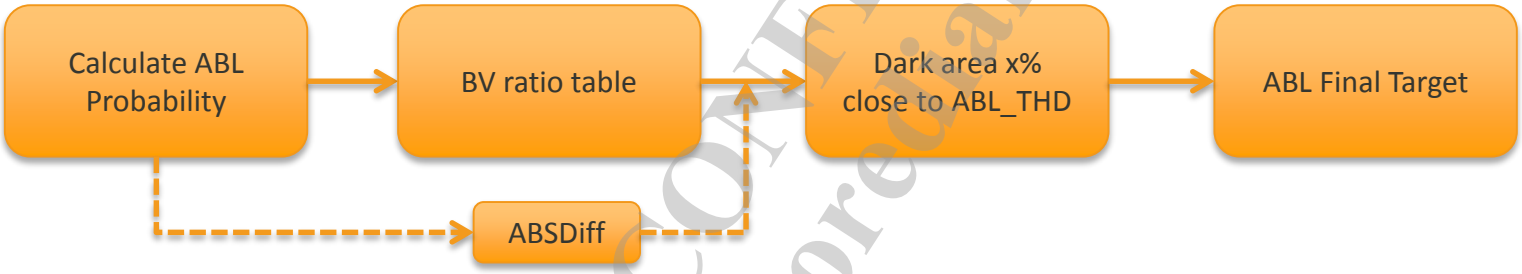
ABL

- Raises the brightness of a dark part to a certain level



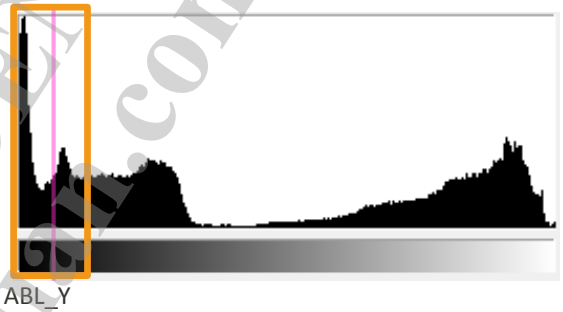
Anti – Back Light

- ABL 3.0 Target Flow



Anti – Back Light

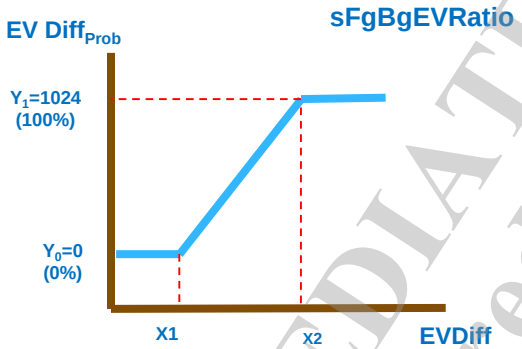
- ABL_Y
  - Using CentralYHist to calculate ABL_Y (by u4Pcent)



- ABL Probability

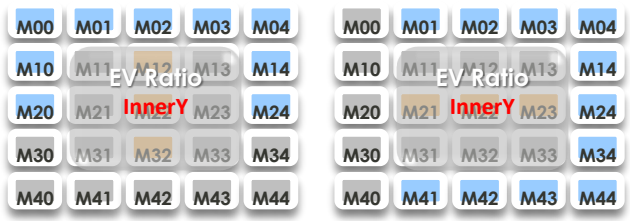
ABL Prob = EVD Prob x BV Prob

- EV Diff Prob
  - **OutY** has two different type (C & Corner) in blue area
  - **InY** = **ABL_Y** + orange area
  - EV Diff = **OutY** – **InY**

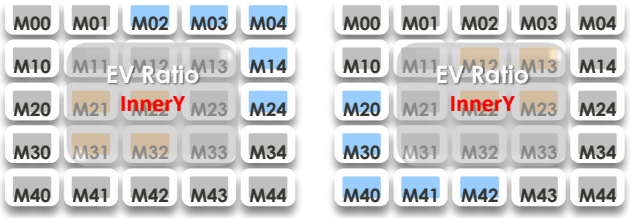


調整EVD影響 ABL_Prob

C type

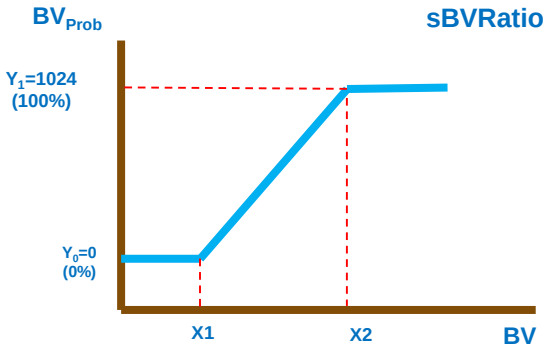


Corner type



Anti – Back Light

- BV Prob



根據場景調整BV ratio影響ABL_Prob

NVRAM

Adjust ABL_Y

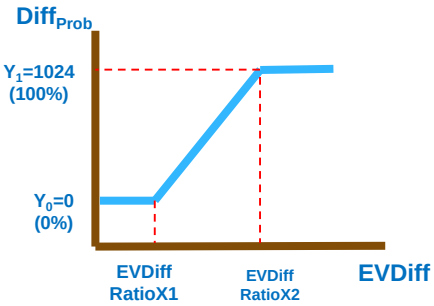
Adjust EVD Prob

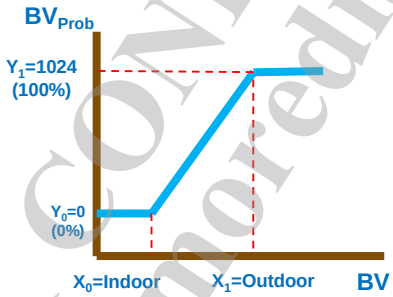
Adjust BV Prob

```
//rABL_Spec
{
    TRUE,//FALSE, //TRUE,//bEnableBlackLigh 11/30
    1024,//u4BackLightWeigh
    500,//550,//350,//400,//u4Pcent
    20,//18,//22,//u4Thd,
    255, // center luminance
    180,//160,//190,//200,//256, // final target li
    //sFgBgEVRatio
    {
        3600,//1800,//1900,//2100,//2200, //i4X1
        0,    //i4Y1
        8000, //3900,//4000, //i4X2 11/31
        1024   //i4Y2
    },
    //sBVRatio
    {
        4800,//3800,//i4X1
        0,   //i4Y1
        9000,//5000,//i4X2  //11/31
        1024  //i4Y2
    }
},
```

Anti – Back Light

- ABL absDiff
  - Add **ABS function** for EV Diff
  - Independent BV ratio table
  - Solve Corner building back light





反向ABL有獨立的EVD&BV ratio table來決定ABL_Prob

- Abs ABL Prob

ABL Prob = abs EVD Prob x abs BV Prob

- Final ABL Prob

Final ABL Prob = Max(ABL Prob, abs ABL Prob)

Tuning custom

```
static strABL_absDiff g_ABLabsDiff =
{
    TRUE,                              //enable enhance ABL(absDiff)
    {2500,  5000},              //u4EVDiffRatio_X
    {0, 1024},                      //u4EVDiffRatio_Y
    {9000,  11000},           //u4BVRatio_X
    {0, 1024}                       //u4BVRatio_Y
};
```

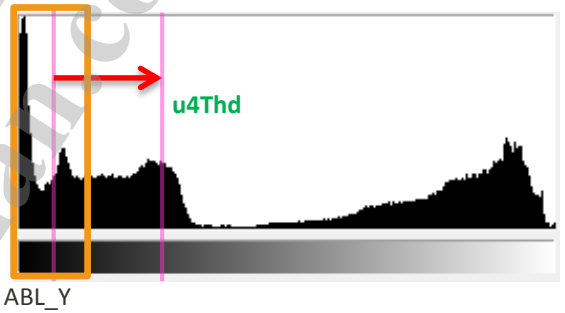
Anti – Back Light

- Dynamic ABL Weighting

  ABL Weighting = u4BackLightWeight x ABL_Prob

- ABL Target

  ABL Target = CWV * u4Thd / ABL_Y



- ABL Final Target
  - ABL Target formula :

    ABL Final Target = AE_Target + (ABL_Target – AE_Target) x ABL_Prob

  - Condition 1 : central block needs to be darker than threshold

    Central Y(M22) x AE_Target < CWV x u4CenterHighBnd

  - Condition 2 : ABL_Target needs to be larger than HS Target

    ABL Final Target >= HS_Target

  - Condition 3 : limit ABL target strength

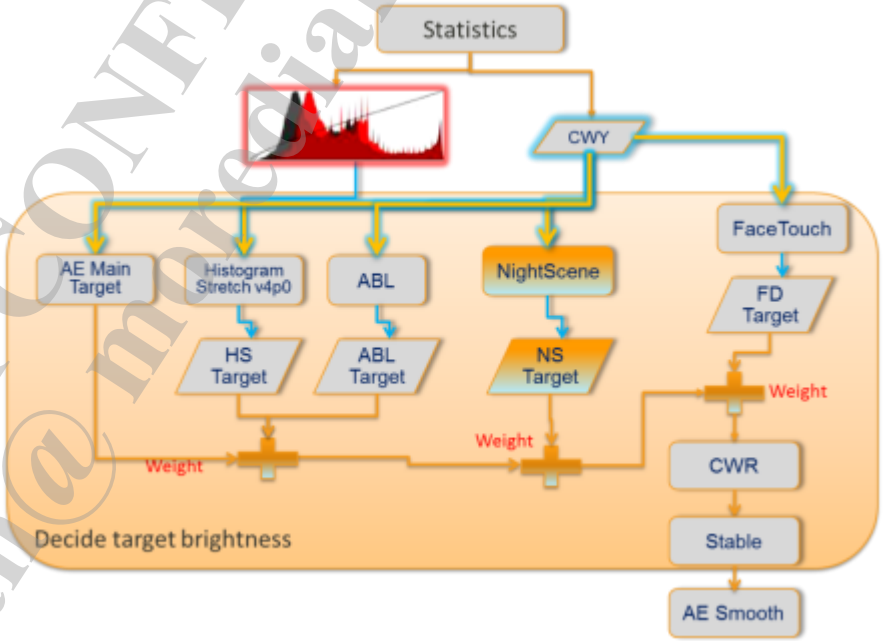
    ABL Final Target <= u4TargetStrength x AE_Target

NVRAM

```
//rABL_Spec
{
    TRUE,//FALSE, //TRUE,//bEnableBlackLigh 11/30
    1024,//u4BackLightWeigh
    500,//550,//350,//400,//u4Pcent
    20,//18,//22,//u4Thd,
    255, // center luminance
    180,//160,//190,//200,//256, // final target li
```

u4TargetStrength有效控制
ABL final target的強度

NS

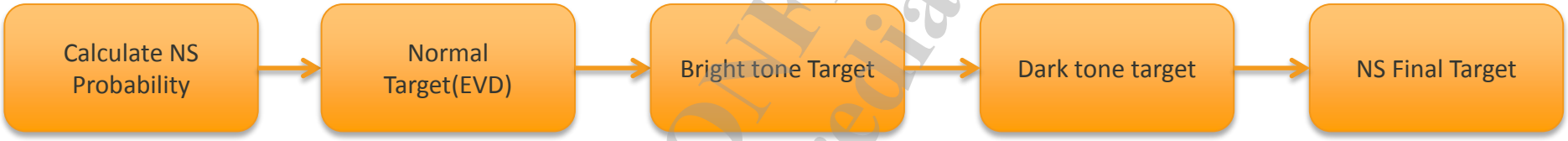


NS

- What to take care in night scene?
  - To avoid brightness of whole image is raised too bright → NS normal target
  - To keep bright part bright enough → bright tone target
  - To keep details in dark tone → low bound target
  - Max of the 3 targets as Final NS target

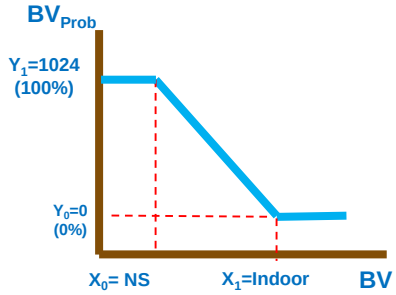
Night Scene

- NS Target Flow



Night Scene

- NS Probability
  - NS_BV Prob by BV ratio table

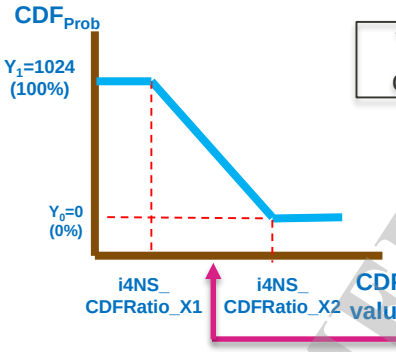


NVRAM

```
//sBVRatio
{
-50,//  -2000,   //i4X1
 1024,   //i4Y1
 2000,//3000, //i4X2
 0       //i4Y2
},
```

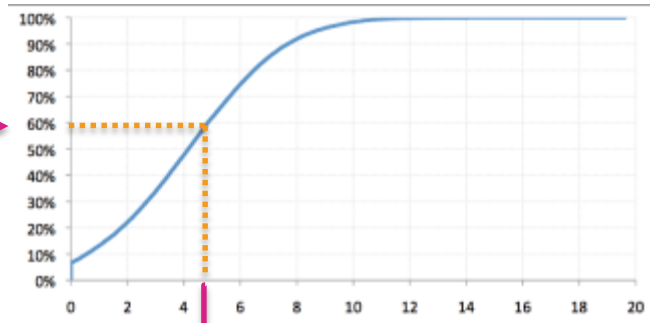
  - CDF (Cumulative Distribution Function) mapping table
    - It will be confused between dark indoor & NS if only BV Prob
    - Need to set a percentage of CDF for more accurate probability

調整BV ratio用來區分In door & NS , 再以CDF Prob區分過暗室內景與過亮夜景



Tuning custom

```
static strNS_CDF g_strNSCDF =
{
    TRUE,              // bEnable
    600,               // u4Pcent
    {-1000,  -300},    // i4NS_CDFRatio_X
    {1024,   0},       // u4NS_CDFRatio_Y
```



Night Scene

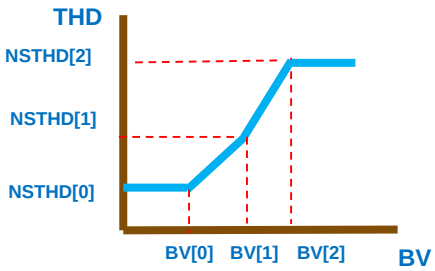
- NS Final Probability

NS Prob = NS_BV Prob x NS_CDF Prob

Night Scene

- NS Normal Target
  - NS_THD
    - THD-BV ratio table

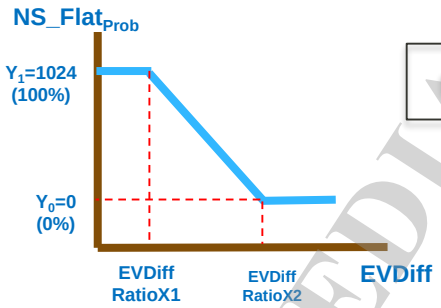
NS Normal Target 機制同 normal scene - HS



Tuning custom

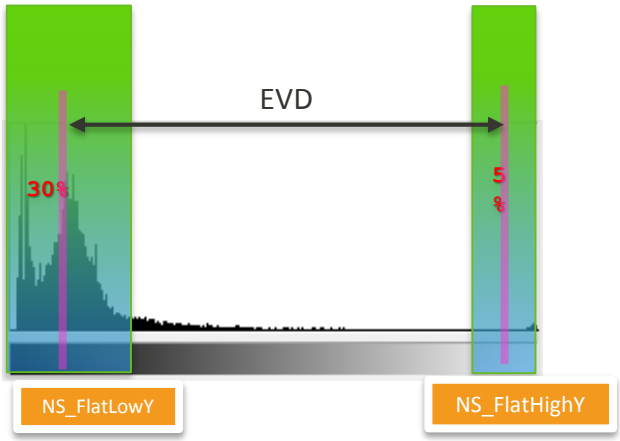
```
static strNSBVCFG g_NSBVCFG =
{
    TRUE,    //bEnableNSBVCFG
    3,       //u4TblLength
  {-4500, -3000, 1000},        //i4BV[16]
   {55, 80, 90},    //u4BTTHD[16]
    {140, 160, 170},    //u4NSTHD[16]
};
```

  - Calculate NS_Flat_Prob
    - Flat night scene (like sky) uses lower THD to avoid too large target
    - EVD using FullRGBHist (by **u4FlatBrightPcent** & **u4FlatDarkPcent**)
    - EVD mapping table



NVRAM

```
50,  //u4FlatBrightPcent;
300,   //u4FlatDarkPcent;
//sFlatRatio
{
    1200, //i4X1
    1024, //i4Y1
  2800,//  2400, //i4X2
    0    //i4Y2
},
```



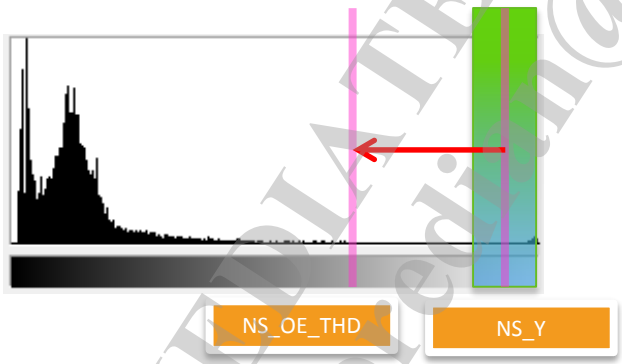
Night Scene

- NS Normal Target
  - Calculate NS_OE_THD

NS_OE_THD
= NS_THD – (NS_THD - u4FlatThd)x NS_Flat_Prob

  - NS Normal Target
    - NS_Y using FullRGBHist (by **u4Pcent**)

NS_Normal_Target
= CWV * NS_OE_THD / NS_Y



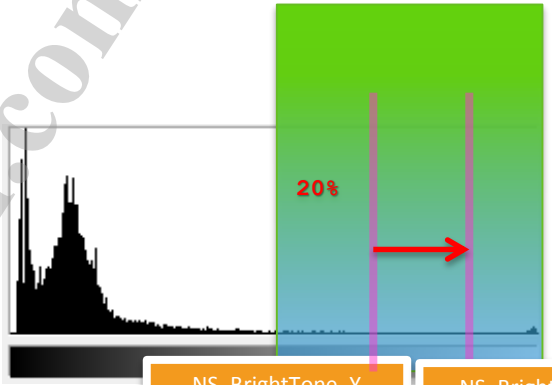
NVRAM

```
//rNS_Spec
{
    TRUE, // bEnableNightScene
     10, //5,     //u4Pcent
    175, // 170,   //u4Thd
     55,//60,,//72, //52,    //u4FlatThd

    180, //  200,  //u4BrightTonePcent
    55,//60,//75, //90, //85,//80,//u4B

    500,  //u4LowBndPcent
    5,    //u4LowBndThd
    22,//26,    //u4LowBndThdLimit

    50,  //u4FlatBrightPcent;
    300,   //u4FlatDarkPcent;
    //sFlatRatio
    {
        1200, //i4X1
        1024, //i4Y1
      2800,//  2400, //i4X2
        0    //i4Y2
    },
```

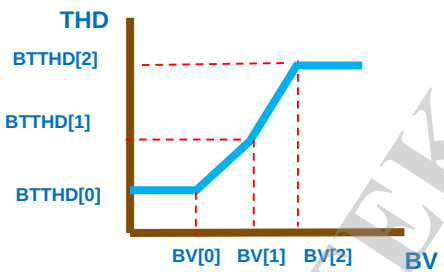
Night Scene

- Bright Tone Target
  - NS_BrightTone_Y using FullRGBHist (by **u4BrightTonePcent**)

Bright tone THD 取代 NS_OE_THD



  - BTTHD
    - THD-BV ratio table



NVRAM

```
static strNSBVCFG g_NSBVCFG =
{
    TRUE,    //bEnableNSBVCFG
    3,       //u4TblLength
   {-4500, -3000, 1000},        //i4BV[16]
   {55, 80, 90},    //u4BTTHD[16]
    {140, 160, 170},    //u4NSTHD[16]
};
```

  - Calculate NS_BrightTone_THD
    - Consider NS_Flat_Prob like NS normal target

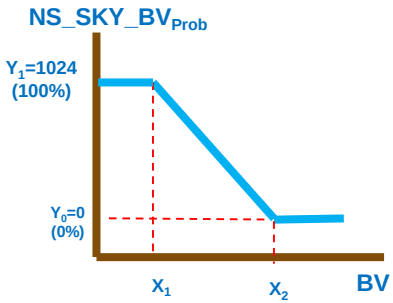
NS_BrightTone_THD
= BTTHD – (BTTHD - u4FlatThd)x NS_Flat_Prob

NVRAM

```
//rNS_Spec
{
    TRUE, // bEnableNightScene
    10, //5,     //u4Pcent
   175, // 170,  //u4Thd
    55,//60,,//72, //52,    //u4FlatThd
```

Night Scene

- **bEnableNightSkySuppresion**
  - To avoid brightness of NS sky is raised too high
  - NS_Sky Prob by BV ratio table



根據NS_Sky_Prob修改
bright tone THD

NVRAM

```
TRUE, // bEnableNightSkySuppresion
//sSkyBVRatio
{
    -4000, //i4X1
    1024, //i4X2
    -2000,  //i4Y1
    0     //i4Y2
}
```

  - Calculate new NS_BrightTone_THD

NS_BrightTone_THD ’
= NS_BrightTone_THD– (NS_BrightTone_THD - u4FlatThd)x NS_Sky_Prob

- NS Bright Tone Target

NS_Bright_Tone_Target
= CWV * NS_BrightTone_THD ’ / NS_Y

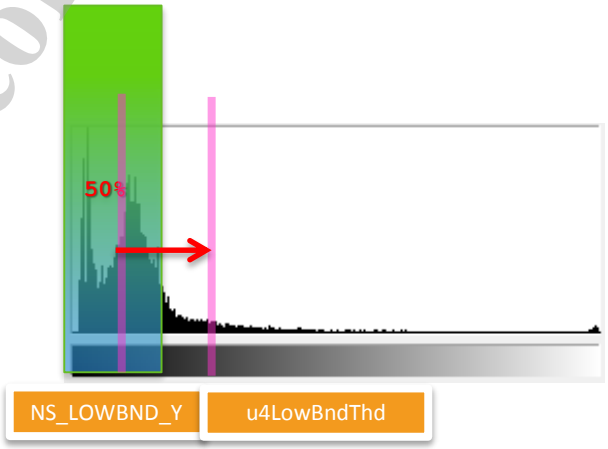
Night Scene

- Dark Tone Target
  - NS_LOWBND_Y using FullRGBHist(by **u4LowBndPcent**)
  - Calculate NS_LOWBND_Target

NS_LOWBND_Target
= CWV * u4LowBndThd / NS_LOWBND_Y

  - Dark Tone Target limit

u4LowBndThd <= NS_LOWBND_Target <= u4LowBndThdLimit



常用u4LowBndThdLimit控制Dark tone的強度

NVRAM

```
500,  //u4LowBndPcent
5,    //u4LowBndThd
22,//26,    //u4LowBndThdLimit
```

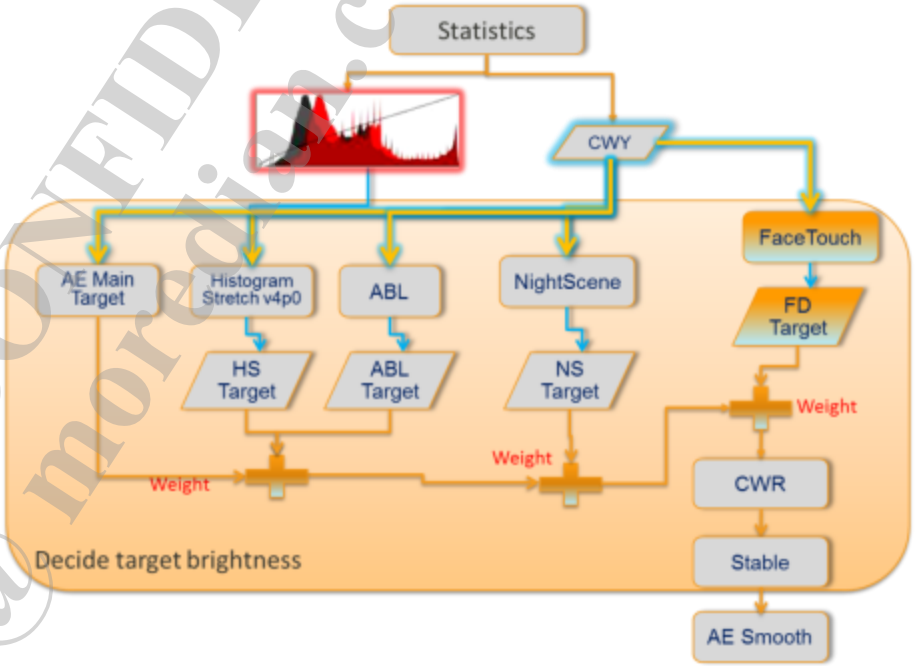
Night Scene

- NS Final Target

```
NS Final Target =
      MAX(NS_Normal_Target , NS_BrightTone_Target, NS_LOWBND_Target)
```

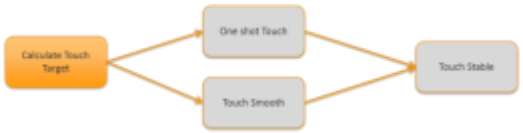
三種機制下選擇target最大者為NS final target

TOUCH / FACE



Touch

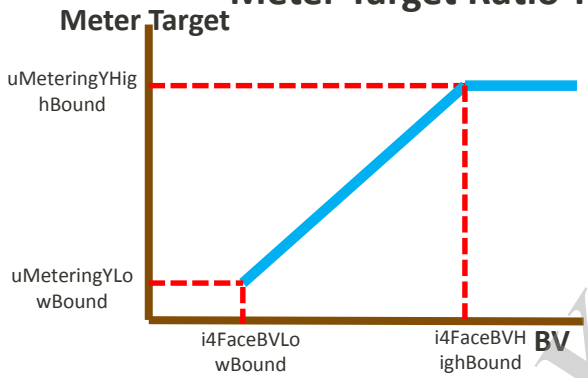
- Touch Target Flow



Touch

- Calculate Touch Target (Meter Target)
  - According to Touch area, get **MeterY**(avg) from 120x90 Y block
  - Meter Target BV ratio table (use the same BV parameters as face AE)

Meter Target Ratio Table



Tuning custom

```
static strFaceSmooth g_AEFaceSmooth = {
    10,                 // u4MinYLowBound;
   360,                 // u4MaxYHighBound;
     0,                 // i4FaceBVLowBound;
  4000,                 // i4FaceBVHighBound;
   235,                 // tempFDY HighBound
     1,                 // tempFDY LowBound
    20,                 // face lost max count
   100,                 // face window scale ratio
    35,                 //face NS lowbound
```

NVRAM

```
// rTOUCHFD_Spec
{
    40, //uMeteringYLowBound;
    50, //uMeteringYHighBound;
    50,//46,//50,//45, //uFaceY
    58,//56,//60,//55, //uFaceY
    3,  //uFaceCentralWeight;
    120,//u4MeteringStableMax;
    80, //u4MeteringStableMin;
}
```

- Touch Target

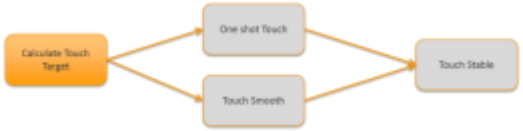
  Touch_Target = CWV x Meter Target / MeterY

- Restrict Target with High/Low Bound

Tuning custom

```
 10,                // u4MinYLowBound;       // metering and face boundary min Y value
360,                // u4MaxYHighBound;      // metering and face boundary max Y value
```

Touch Target	Final Target restrict in High/Low Bound
Low Bound (10)	> = AE Target * LowBound / 128
High Bound (360)	<= AE Target * HighBound / 128



Touch

- Blending Touch Target and Normal Target
  - When Touch hit, the final target is blended using Touch_Target and Normal Target, no Face AE

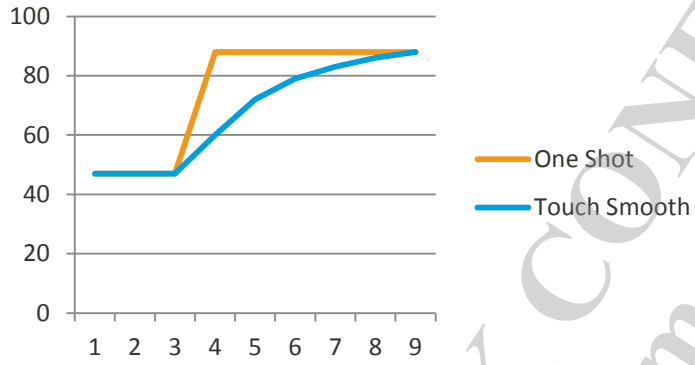
Final Target = (1024 - u4MeterWeight) * CWV + u4MeterWeight * Touch_Target

Tuning custom

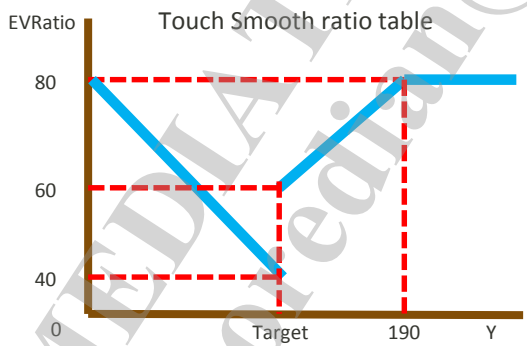
```
  10,                    // u4MinYLowBound;        // metering and face boundary min Y value
 360,                    // u4MaxYHighBound;       // metering and face boundary max Y value
1024,                    // u4MeterWeight 1024 base.100%=>1024
```

Touch

- Converge Mode
  - One Shot Touch: directly use touch AE target
  - Touch Smooth: blend touch AE target and normal AE target for better smooth

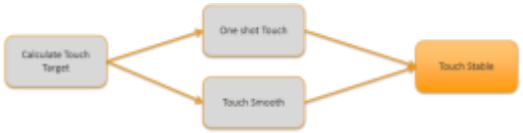


  - Touch Smooth Parameter
    - Moving Ratio Table to control the smooth step and speed (refer to AE Smooth chapter)



Tuning custom

```
static strAEMovingRatio g_AETouchMovingRatio =
{
    100, //u4SpeedUpRatio
    100, //u4GlobalRatio
    190, //u4Bright2TargetEnd
    20,   //u4Dark2TargetStart
    80, //u4B2TEnd
    60,  //u4B2TStart
    40,  //u4D2TEnd
    80,  //u4D2TStart
};
```



Touch

- Touch Stable
  - After AE converging with touch area, brightness of the touch area may change with time. When the brightness of touch area changes a lot, AE needs to give up the touch area and use normal AE totally.
  - Metering Stable Value
    - When Touch AE just converges to target, the current **MeterY** is set as Metering Stable Value
  - Metering Ratio
    - Used to determine the brightness changing threshold to give up the touch area
    - When Metering Ratio is out of the range between low bound and high bound, the touch area is given up, and **normal target** is used as the final AE target.

Metering Ratio = 100 x **cur_MeterY** / Metering Stable Value

- Restrict Target with High/Low Bound

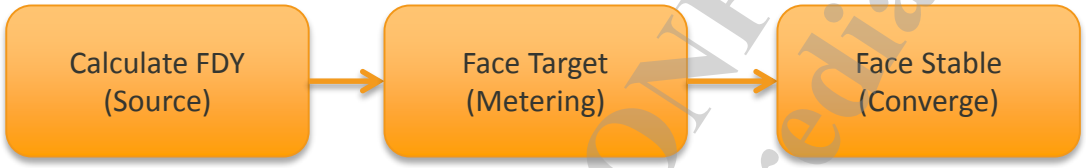
Touch Stable	Converge to Normal Target
Low Bound (80)	Metering Ratio < u4MeteringStableMin
High Bound (120)	Metering Ratio > u4MeteringStableMax

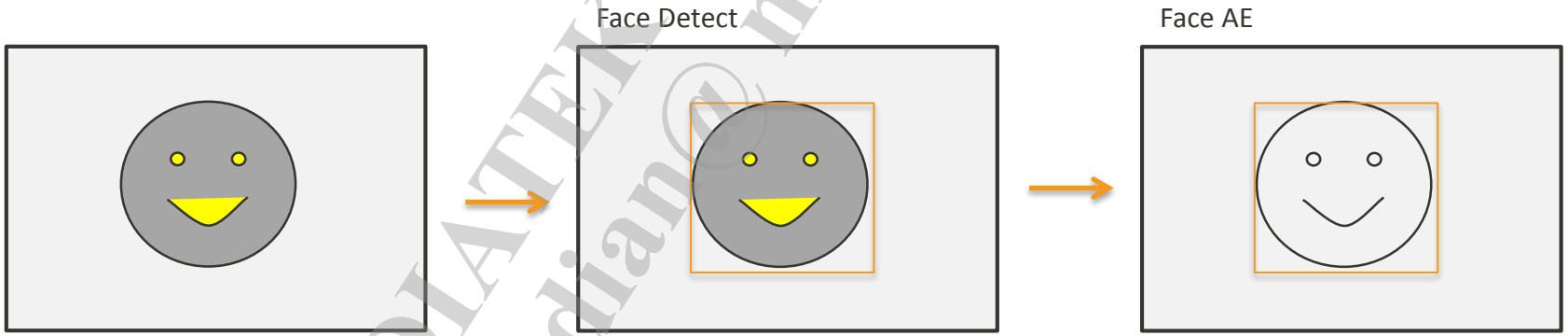
NVRAM

```
// rTOUCHFD_Spec
{
    40, //uMeteringYLowBound;
    50, //uMeteringYHighBound;
    50,//46,//50,//45, //uFaceY
    58,//56,//60,//55, //uFaceY
    3,  //uFaceCentralWeight;
    120,//u4MeteringStableMax;
    80, //u4MeteringStableMin;
}
```

Face

- Face Target Flow

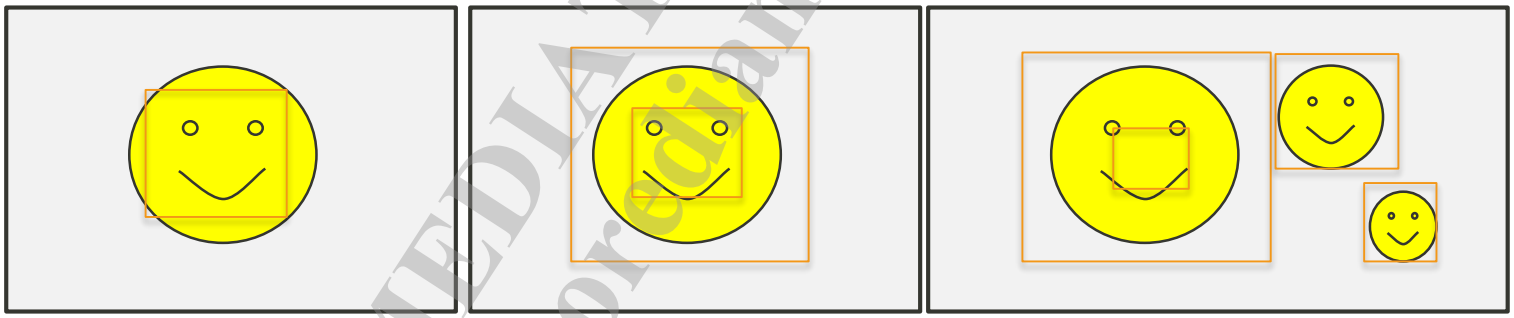




Source FDY

- 在Source部分, AeAlgo 會根據上層所提供的Face框, 進一步算出一個FDY來當做此幀Face的亮度

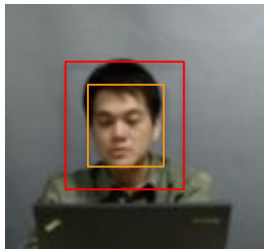
- Source part 的 Tuning method
  1. Face size scaling (大小框)
  2. Face Landmark (眼睛嘴巴位置)
  3. Multi-Face



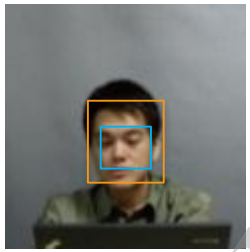
Face size scaling

- To adjust the final face size according to received face area

Height, Width scaling up 200%



Height, Width scaling down 50%



- Tuning Parameter

Tuning custom

```
static strFaceSmooth g_AEFaceSmooth = {
    10,                 // u4MinYLowBound;        // metering and face boundary min Y value
   360,                 // u4MaxYHighBound;       // metering and face boundary max Y value
     0,                 // i4FaceBVLowBound;      // face boundary min BV value
  4000,                 // i4FaceBVHighBound;     // face boundary max BV value
   235,                 // tempFDY HighBound
     1,                 // tempFDY LowBound
    20,                 // face lost max count
   100,                 // face window scale ratio range 1~400% default 90%
    35,                //face NS lowbound
     5,                 // u4FD_Lock_MaxCnt
     5,                 // u4FD_MixCWR_MaxCnt
     0,                 // u4FD_TemporalSmoothY
    80,                 // u4FD_FaceWidthCropRat
    80,                 // u4FD_FaceHeightCropRat
```

No use

- Tuning preference

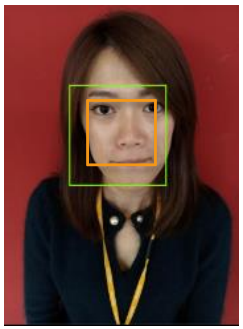
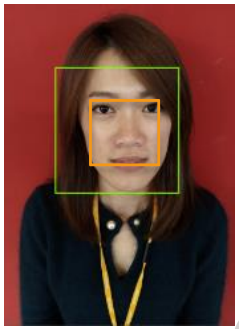
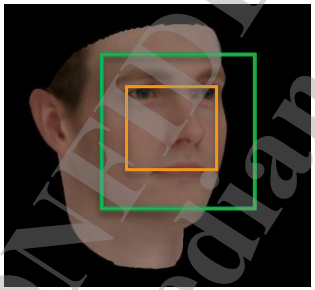
	Pros	Cons
Scaling up	Smooth/Consistence	Worse accuracy
Scaling down	Accuracy	Damping

Face Landmark

- Has better consistence and less influenced by background bias – landmark can focus on face region

landmark

face

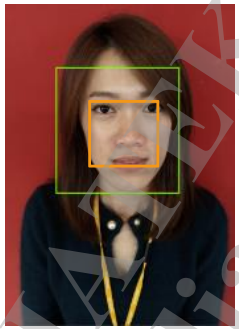
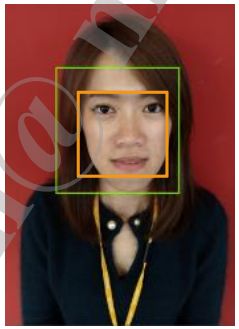




Tuning custom

```
static strFaceLandMarkCtrl g_AEFaceLandmark = {
    TRUE,                    //bLandmarkCtrlFlag
     0,                      //u4LandmarkCV_Thd
     3,                      //u4LandmarkWeight
   100,                      //i4LandmarkExtRatW;
   100,                      //i4LandmarkExtRatH;
    90,                      //i4LandmarkTrustRopDegree;
     5,                      //i4RoundXYPercent;
     3,                      //i4RoundSZPercent;
     9,                      //i4THOverlap;
     3,                      //i4BUFLEN; //must not exce
     8,                      //i4SMOOTH_LEVEL;
     0,                      //i4MOMENTUM;
    60                       //u4LandmarkWeightPercent;
```

- Scaled by LandmarkExtRatW, LandmarkExtRatH

u4LandmarkExtRatW = 100
u4LandmarkExtRatH = 100

u4LandmarkExtRatW = 130
u4LandmarkExtRatH = 130

Face Landmark

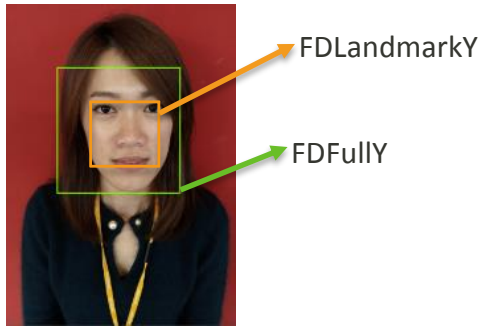
- FDY

FDY = (FDLandmarkY * u4LandmarkWeightPercent + FDFullY(100 - u4LandmarkWeightPercent)) / 100

- Tuning preference

Tuning custom

```
static strFaceLandMarkCtrl g_AEFaceLandmark = {
    TRUE,                  //bLandmarkCtrlFlag
     0,                     //u4LandmarkCV_Thd
     3,                     //u4LandmarkWeight
   100,                     //i4LandmarkExtRatW;
   100,                     //i4LandmarkExtRatH;
    90,                     //i4LandmarkTrustRopDegree;
     5,                     //i4RoundXYPercent;
     3,                     //i4RoundSZPercent;
     9,                     //i4THOverlap;
     3,                     //i4BUFLEN; //must not exce
     8,                     //i4SMOOTH_LEVEL;
     0,                     //i4MOMENTUM;
    60                      //u4LandmarkWeightPercent;
```



Landmark Weight	
small	Focus on default size(FDLandmarkY has smaller weighting)
large	More focus on skin, more accuracy (FDLandmarkY has larger weighting)

Multi-Face Weight

- Multi-Face Weight policy
  - Sorting detected faces and giving each face weighting for mixing Final FDY
  - Landmark would be FDY[0] if existing valid landmark (Landmark only support 1 face)

- Sorting Priority
  - Determine the order to multiply FaceWeight

Sorting Priority	
FACE_PRIOR_TIME	According to detected time
FACE_PRIOR_SIZE	From large size to small size
FACE_PRIOR_LIGHT	From bright face to dark face
FACE_PRIOR_DARK	From dark face to bright face

Tuning custom

```
typedef struct {
    MBOOL bEnableMultiFaceWeight;
    FACE_PRIOR_ENUM Prior;
    MUINT32 u4FaceWeight[MAX_AE_METER_AREAS];

}strFaceWeight;
static strFaceWeight g_AEFaceWeight = {
    MTRUE,
    FACE_PRIOR_TIME,
    {
        1,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    }
};
```

Sorting priority

FaceWeight

- Final FDY

Final FDY = (FDY[0]*FaceWeight[0] + … + FDY[N]*FaceWeight[N]) / (FaceWeight[0] + … + FaceWeight[N])

Face type and FDY classification

- Face types
  - To handle different face types
  - Classify different face types and give them different FDY

Frontal Face (Robust Face)	• Use current FDY value • Record robust FDY value

Side Face	• If previous frame has no face or frontal face never occurs
	  • Treat as side face, use current FDY value
	• If previous frame has face
	  • Treat as OT face, use previous frame FDY value

Object Tracking face (OT face) • Motion Face • Impulse Y Diff Face	• Use previous frame FDY value
No face	• Clear all FDY

Face type and FDY classification

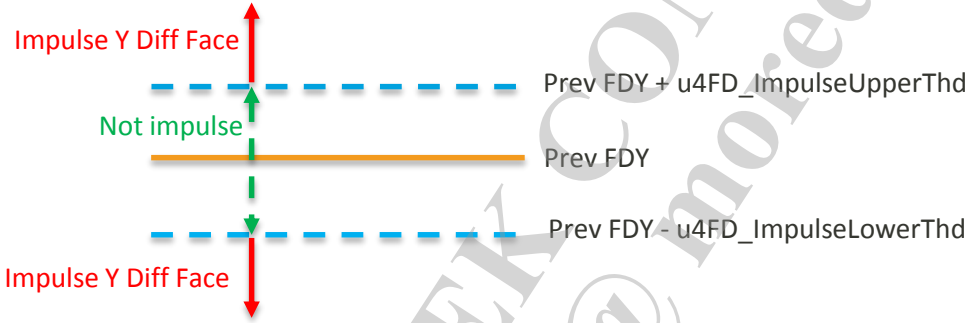
- Motion Face
  - When face motion is bigger than a specific face size ratio, the face is treated as OT Face, and previous FDY is used, so Face will keep the same
  - u4FD_FaceMotionLockRat
    - 0 means if motion occurs, Face AE will keep the same. 20 means if motion is bigger than 20% of face width, Face AE starts to keep the same

Tuning custom

```
    0,              // u4FD_PerframeAntiOverFlag
    0,              // u4FD_SideFaceLock
    1,              // u4FD_LimitStableThdLowBnd
    0,              // u4FD_LimitStableThdLowBndNum
    1,              // u4FD_ReConvergeWhenFaceChange
   15,              // u4FD_FaceMotionLockRat
    3,              // u4FD_ImpulseLockCnt
   25,              // u4FD_ImpulseUpperThd
   18,              // u4FD_ImpulseLowerThd
    2,              // u4FD_ROPTrustCnt
   30,              // u4FD_ReCovergeWhenSizeChangeRat
  120,              // u4FD_ReCovergeWhenPosChangeDist
    1,              // u4FD_ReConvergeWhenAFDone;
    1               // u4FD_OTFaceLock;
};
```

Face type and FDY classification

- Impulse Y Diff Face
  - When FDY has been out of range between Prev FDY + u4FD_ImpulseUpperThd and Prev FDY - u4FD_ImpulseLowerThd, the face is treated as OT Face for several frames
  - Larger u4FD_ImpulseLockCnt means the face is treated as OT Face for more frames, and Face AE will keep the same for more frames



Tuning custom

```
    0,          // u4FD_PerframeAntiOverFlag
    0,          // u4FD_SideFaceLock
    1,          // u4FD_LimitStableThdLowBnd
    0,          // u4FD_LimitStableThdLowBndNum
    1,          // u4FD_ReConvergeWhenFaceChange
   15,          // u4FD_FaceMotionLockRat
    3,          // u4FD_ImpulseLockCnt
   25,          // u4FD_ImpulseUpperThd
   18,          // u4FD_ImpulseLowerThd
    2,          // u4FD_ROPTrustCnt
   30,          // u4FD_ReCovergeWhenSizeChangeRat
  120,          // u4FD_ReCovergeWhenPosChangeDist
    1,          // u4FD_ReCovergeWhenAFDone;
    1           // u4FD_OTFaceLock;
};
```

Face type and FDY classification

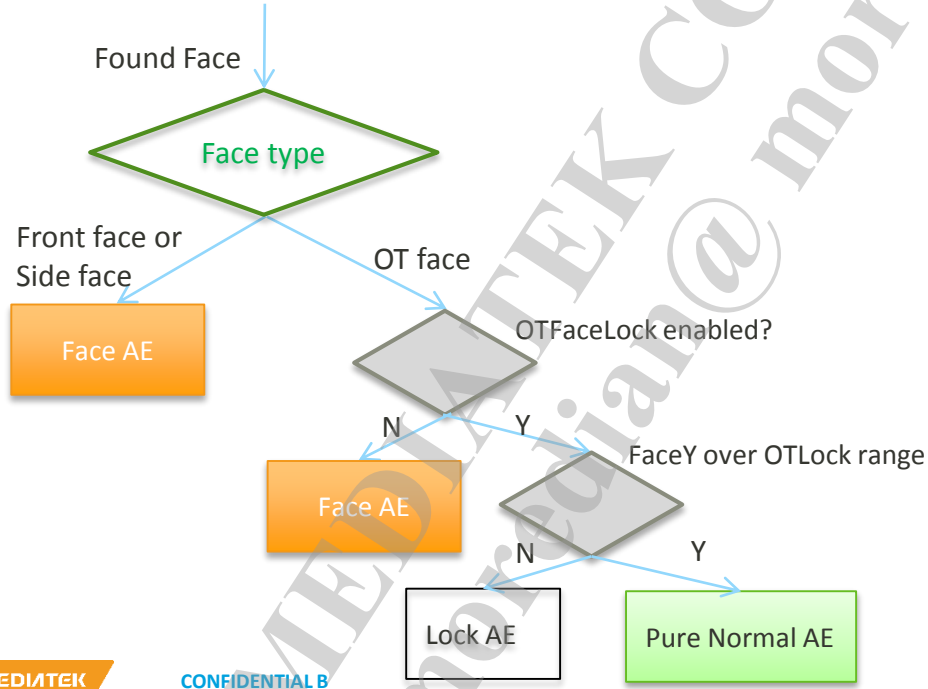
- Side Face Lock
  - Decide whether side face should be used for face AE or just keep AE (as OT face)
  - u4FD_SideFaceLock
    - Suggest 0 which can treat side face as front face instead of OT face

Tuning custom

```
    0,              // u4FD_PerframeAntiOverFlag
    0,              // u4FD_SideFaceLock
    1,              // u4FD_LimitStableThdLowBnd
    0,              // u4FD_LimitStableThdLowBndNum
    1,              // u4FD_ReConvergeWhenFaceChange
   15,              // u4FD_FaceMotionLockRat
    3,              // u4FD_ImpulseLockCnt
   25,              // u4FD_ImpulseUpperThd
   18,              // u4FD_ImpulseLowerThd
    2,              // u4FD_ROPTrustCnt
   30,              // u4FD_ReCovergeWhenSizeChangeRat
  120,              // u4FD_ReCovergeWhenPosChangeDist
    1,              // u4FD_ReCovergeWhenAFDone;
    1               // u4FD_OTFaceLock;
};
```

Metering Face Target

- 根據Source – FDY的值,來算出最後的Face Target
- Face Target Flow 又分為
  - Found Face -> FDY ! = 0
  - Not Found -> FDY = 0
    - Lost Face Process

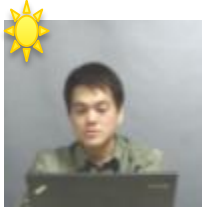


Tuning custom

```
220,                // u4FD_OTLockUpperBnd
  0,                // u4FD_OTLockLowerBnd
  2,                // u4FD_ContinueTrustCnt
  0,                // u4FD_PerframeAntiOverFlag
  0,                // u4FD_SideFaceLock
  1,                // u4FD_LimitStableThdLowBnd
  0,                // u4FD_LimitStableThdLowBndNum
  1,                // u4FD_ReConvergeWhenFaceChange
 15,                // u4FD_FaceMotionLockRat
  3,                // u4FD_ImpulseLockCnt
 25,                // u4FD_ImpulseUpperThd
 18,                // u4FD_ImpulseLowerThd
  2,                // u4FD_ROPTrustCnt
 30,                // u4FD_ReCovergeWhenSizeChangeRat
120,                // u4FD_ReCovergeWhenPosChangeDist
  1,                // u4FD_ReCovergeWhenAFDone;
  1                 // u4FD_OTFaceLock;
```

Face Target (Found Face)

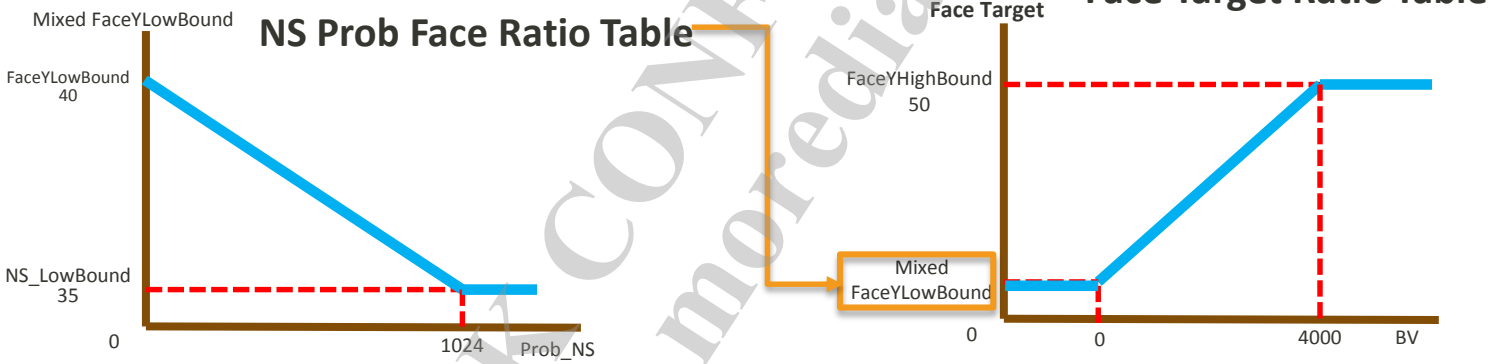
- Flow
  - Calculate Face Target
  - Calculate Face Probability
  - Blend Face Target and Normal Target by Face Probability
- Tuning Method
  - FDTargetbyBV (FD BV ratio table)
  - FDY High/Low Bound (臉的target上下限)
  - FaceLocSizeCheck (Weighting by size & location)
  - FaceOECheck (adjust weighting by over exposure percentage)
  - FCY (adjust weighting by face Y)



Calculate Face Target

- Determine Face Target by BV
  - NS Face Target with CDF NS Probability (區別室內/夜景)

Mixed FaceYLowBound = FaceNSLowBound* Prob_NS + (1024 - Prob_NS) * FaceYLowBound



- Tuning preference

BV / Face Target	
NS	NS Face Target in indoor BV
Indoor	Indoor Face Target in indoor BV
Outdoor	Outdoor Face Target in outdoor BV

Calculate Face Target

- Tuning parameters

NVRAM

```
// rTOUCHFD_Spec
{
    40, //uMeteringYLowBound;
    50, //uMeteringYHighBound;
    40, //uFaceYLowBound;
    50, //uFaceYHighBound;
    3,  //uFaceCentralWeight;
    120,//u4MeteringStableMax;
    80, //u4MeteringStableMin;
}
```

Tuning custom

```
static strFaceSmooth g_AEFaceSmooth = {
    10,                 // u4MinYLowBound;          // metering and face boundary min Y valu
    360,                // u4MaxYHighBound;         // metering and face boundary max Y value
    0,                  // i4FaceBVLowBound;        // face boundary min BV value
    4000,               // i4FaceBVHighBound;       // face boundary max BV value
    235,                // tempFDY HighBound
    1,                  // tempFDY LowBound
    20,                 // face lost max count
    100,                // face window scale ratio range 1~400% default 90%
    35,                 //face NS lowbound
```

Face Target High/Low Bound

- Restrict the range of Face Target to avoid unreasonable Face Target in extreme condition

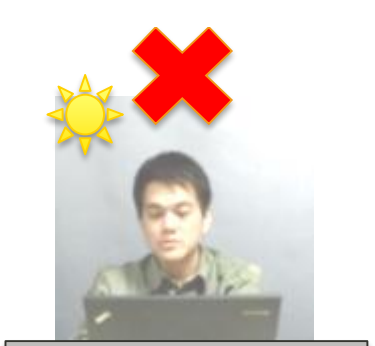
- Tuning Parameter

Tuning custom

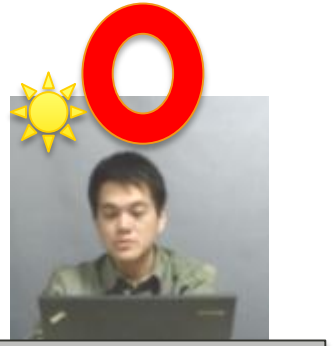
```
static strFaceSmooth g_AEFaceSmooth = {
    10,                 // u4MinYLowBound;      // metering and face boundary min Y valu
   360,                 // u4MaxYHighBound;     // metering and face boundary max Y value
     0,                 // i4FaceBVLowBound;    // face boundary min BV value
  4000,                 // i4FaceBVHighBound;   // face boundary max BV value
   235,                 // tempFDY HighBound
     1,                 // tempFDY LowBound
    20,                 // face lost max count
   100,                 // face window scale ratio range 1~400% default 90%
    35,                //face NS lowbound
```

- Tuning preference

Face Target	Final Face Target restricted between High/Low Bound
Low Bound (10)	> = AE Target * LowBound / 128
High Bound (256)	<= AE Target * HighBound / 128



FDY Target large without restrict



Restrict with High Bound

Face Probability

- Face Probability is used to calculate Final Face target (face found)

Final face target = FaceTarget * Final FaceP + (1024 - Final FaceP) * NormalTarget

- 3 mechanisms to determine face probability

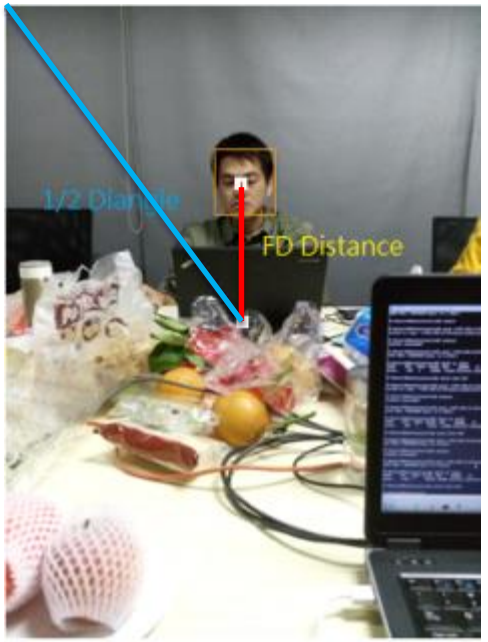
- FaceP_locSize
  - Larger Face size has larger Face_P
  - Shorter distance from image center has larger Face_P
- FaceP_OE
  - Reduce FaceP of image with larger overexposure (OE) pixels to fit environment brightness
- FCY (Face central Y)
  - Give brighter face or darker face larger face probability
  - Used to raise the face probability suppressed by OE Check mechanism, when FCY is bright enough or is dark enough

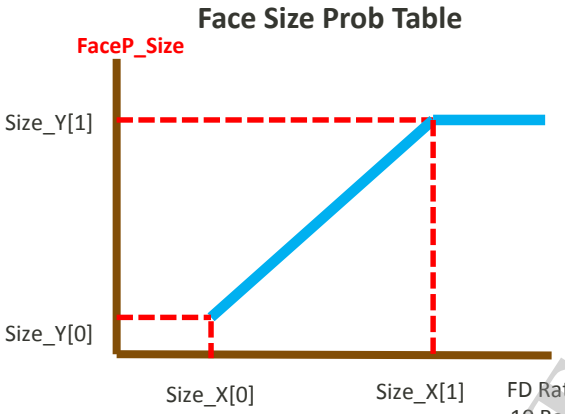
- Tuning preference

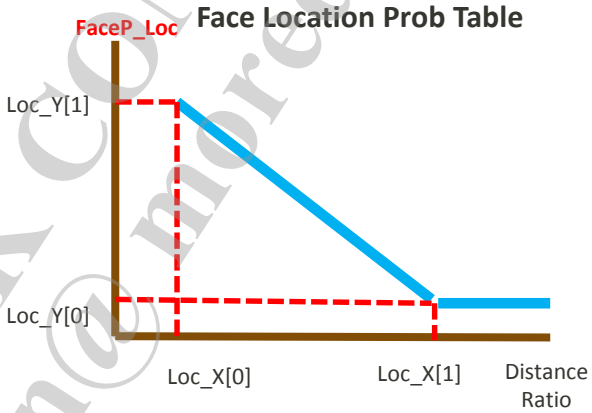
Face Target 混 Normal Target	
FaceP small	Final Target參考環境亮度反應, Smooth佳
FaceP Large	Accuracy受背景影響程度小

Face Probability

- FaceP_LocSize
  - Controlled by size of face area and distance between face and image center
- Formula
  - FaceP_locSize = FaceP_Size * FaceP_Loc





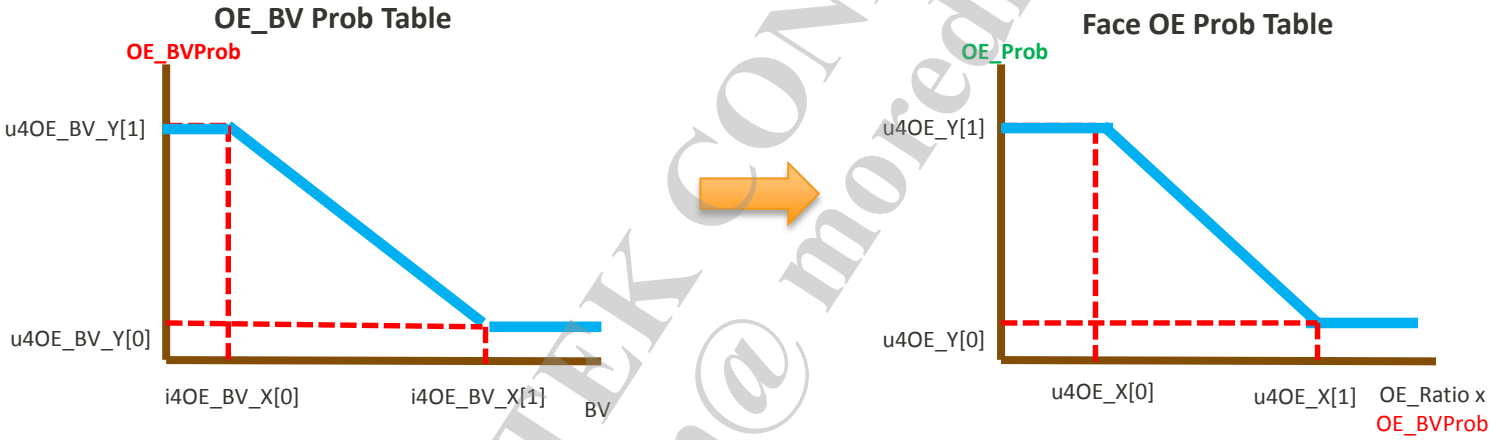


Tuning custom

```
static strFaceLocSize g_FaceLocSizeCheck =
{
    TRUE,          //bFaceLocSizeCheck;
    {20,    5},//u4Size_X[2];
    {1024,  0},//u4Size_Y[2];
    {200,   600},//u4Loc_X[2];
    {1024,  1020}, //u4Loc_Y[2];
```

Face Probability

- FaceP_OE
  - Controlled by BV and OE Ratio
- Formula
  - FaceP_OE = FaceP_LocSize * OE_Prob



Tuning custom

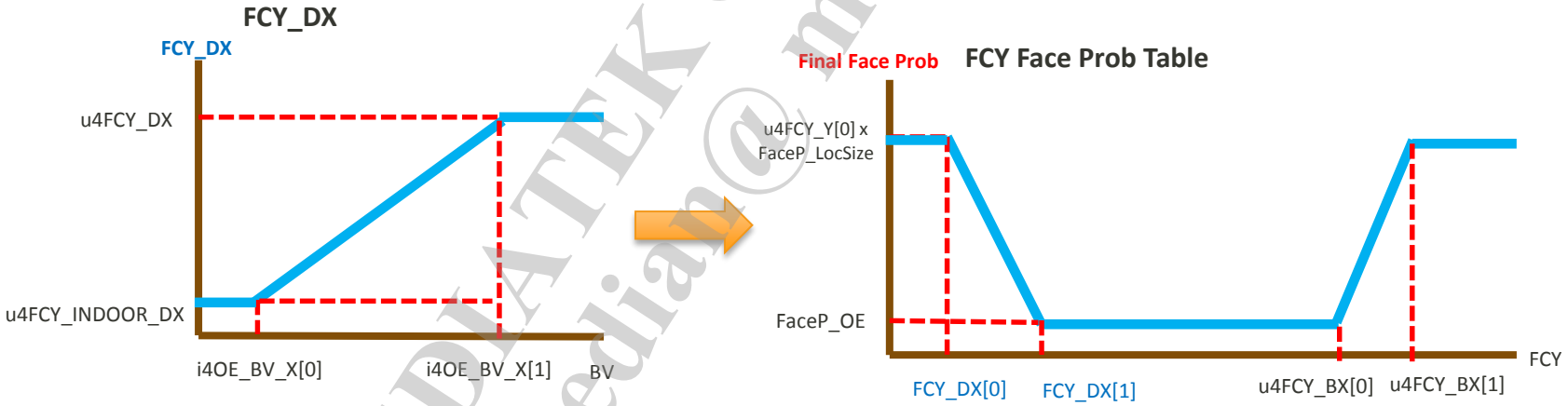
```
TRUE,
{10240,  20480},
{1024,  768},
{3500,  6000},
{1024,  1024},
                    //bFaceOECheck;
//u4OE_X[2];
    //u4OE_Y[2];
    //i4OE_BV_X[2]
    //u4OE_BV_Y[2];
```

Face Probability

- FaceP_FCY
  - FCY > u4FCY_BX[0] → Bright part → raise FaceP suppressed by OE
  - FCY < u4FCY_DX[1] → Dark part → raise FaceP suppressed by OE
  - Else → Final FaceP = FaceP_OE
- Dark part
  - FCY_DX is interpolated by u4FCY_DX and u4FCY_INDOOR_DX using BV

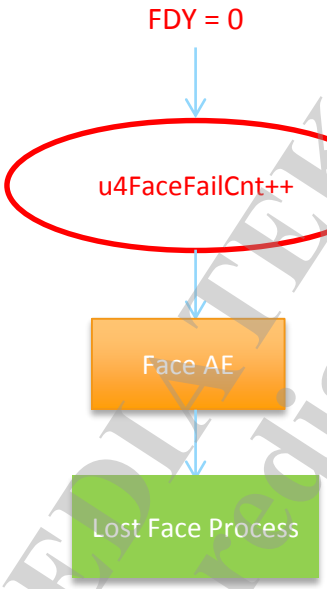
Tuning custom

```
{10,   30},        //u4FCY_DX[2];
{10,   30},        //u4FCY_INDOOR_DX[2];
{110,  95},         //u4FCY_BX[2];
{1024, 0},          //u4FCY_Y[2];
```



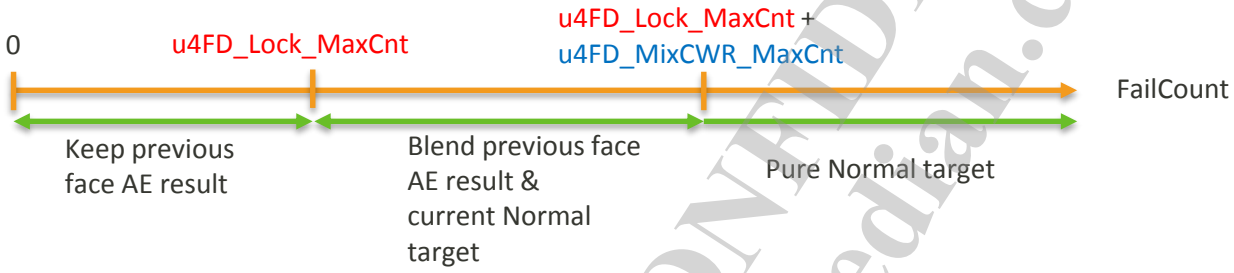
Metering Face Target (Face Not Found)

- Face not found flow
  - Add u4FaceFailCnt
  - Do Face AE
  - Do Lost Face Process
    - Lost face occasionally → Keep previous face AE result to avoid brightness change by AE
    - Lost face continuously → From previous Face AE result to pure Normal AE as time goes on



Lost Face Process (Face Not Found)

- Determine final AE target according to FailCount



- Tuning preference

Tuning custom

```
static strFaceSmooth g_AEFaceSmooth = {
    10,                 // u4MinYLowBound;          // metering and face boundary min Y value
   360,                 // u4MaxYHighBound;         // metering and face boundary max Y value
     0,                 // i4FaceBVLowBound;        // face boundary min BV value
  4000,                 // i4FaceBVHighBound;       // face boundary max BV value
   235,                 // tempFDY HighBound
     1,                 // tempFDY LowBound
    20,                 // face lost max count
   100,                 // face window scale ratio range 1-400% default 90%
    35,                 //face MS lowbound
     5,                 // u4FD_Lock_MaxCnt
     5,                 // u4FD_MixCWR_MaxCnt
     0,                 // u4FD_TemporalSmoothY
```

u4FD_Lock_MaxCnt	Smooth
large	Avoid Face detect fail, keep face info
small	Quickly converge to correct target

u4FD_MixCWR_MaxCnt	Smooth
large	Slower and smoother transition from lock AE to correct target
small	Faster transition from lock AE to correct target

Converge Stable THD

- Please refer to “Stable” chapter for stable algorithm
- Tuning Parameter

Tuning custom

```
static strAEStableThd g_AEStableThd =
{
    0,                      // u4InStableThd;  // 0.08EV
    1,                      // u4OutStableThd
    TRUE,                // enable ae different mode stable threshold setting
    1,                      // u4VideoInStableThd;  // 0.08EV
    1,                      // u4VideoOutStableThd
    1,                      // u4FaceInStableThd;  // 0.08EV
    3,                      // u4FaceOutStableThd
    15,                     // u4FaceOutB2TStableThd
    15,                     // u4FaceOutD2TStableThd
    0,                      // u4TouchInStableThd;  // 0.08EV
    3,                      // u4TouchOutStableThd
    FALSE,                  // Face Ae lock option
    FALSE                   // enable zero stable thd
};
```

- Tuning preference

Face Target consistency Out > = In	
In/Out THD large	Avoid Damping
In/Out THD Small	Better consistency

Re-converge Timing

- Because of a strong stabilization mechanism, we need a re-converge policy to meet scene changes
  - When the following condition meet, AE will be triggered to re-converge.

conditions	Related Parameters	Note
Face ID change	u4FD_ReConvergeWhenFaceChange	
Face size change	u4FD_ReConvergeWhenSizeChangeRat	Width or height changes over u4FD_ReConvergeWhenSizeChangeRat / 100
Face position change	u4FD_ReConvergeWhenPosChangeDist	X-position or y-position changes over u4FD_ReConvergeWhenPosChangeDist
AF refocus	u4FD_ReConvergeWhenAFDone	
Face number change	bEnableMultiFaceWeight	
Face lost		

Tuning custom

```
typedef struct {
    MBOOL bEnableMultiFaceWeight;
    FACE_PRIOR_ENUM Prior;
    MUINT32 u4FaceWeight[MAX_AE_METER_AREAS];

}strFaceWeight;
```

```
1,      // u4FD_LimitStableThdLowBnd
0,      // u4FD_LimitStableThdLowBndNum
1,      // u4FD_ReConvergeWhenFaceChange
15,     // u4FD_FaceMotionLockRat
3,      // u4FD_ImpulseLockCnt
25,     // u4FD_ImpulseUpperThd
18,     // u4FD_ImpulseLowerThd
2,      // u4FD_ROPTrustCnt
30,     // u4FD_ReCovergeWhenSizeChangeRat
120,    // u4FD_ReCovergeWhenPosChangeDist
1,      // u4FD_ReCovergeWhenAFDone;
1       // u4FD_OTFaceLock;
};
```

LCE / DCE

Outline

- Contrast 2.0 = LCE 5.0 + DCE 1.0
  - LCE 5.0
  - DCE 1.0

Contrast 2.0 Policy

GMA

Global Tone

LCE

GMA

Local Tone

LCE

GMA

DCE

Contrast

LCE 5.0

Difference between *LCE 4.6* and *LCE 5.0*

- LCE
  - LCE Tone Curve more precisely
    [Hardware] Control point : 5 → 8
  - Reduce tuning effort :
    Manually generate Tone Curve
    → Automatically generate Tone Curve
  - Accurate Face Brightness Control
    LCE-AE link

Target increase

Target 1800
Strength 800

Target 2048
Strength 800

Target 2248
Strength 800

Strength increase

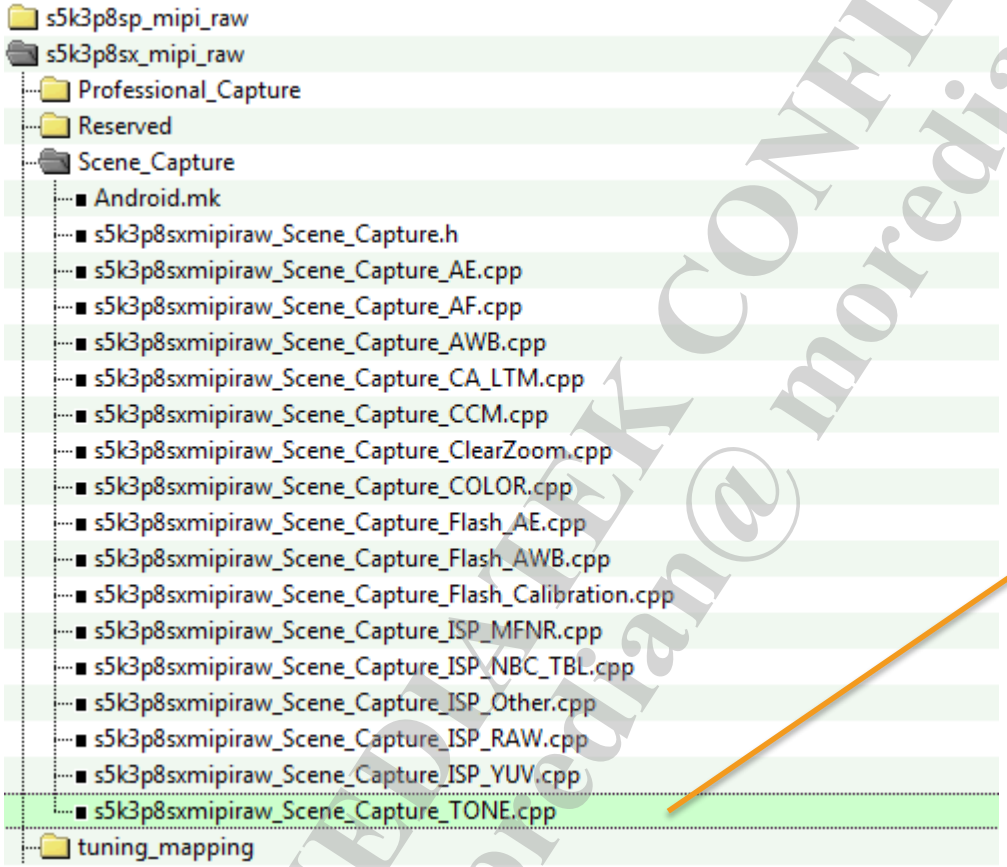
Target 2048
Strength 639

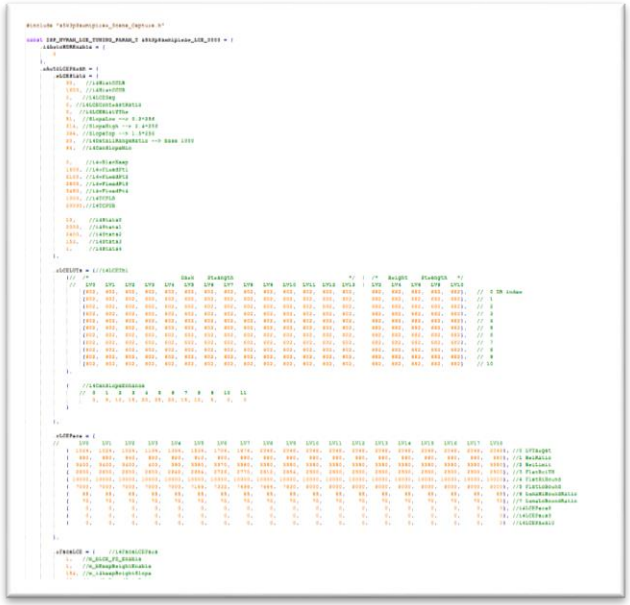
Target 2048
Strength 800

Target 2048
Strength 1000

Tone Curve Generation

- [Sensor]_[Scenario]_TONE.cpp





Target tuning

```
.rLCEPara = {
//      LV0    LV1    LV2    LV3    LV4    LV5    LV6    LV7    LV8    LV9    LV10   LV11   LV12   LV13   LV14   LV15   LV16   LV17   LV18
    {  1024,  1024,  1024,  1194,  1364,  1534,  1704,  1874,  2048,  2048,  2048,  2048,  2048,  2048,  2048,  2048,  2048,  2048,  2048}, //0 LVTarget
     {   950,   950,   940,   930,   920,   910,   900,   890,   880,   880,   880,   880,   880,   880,   880,   880,   880,   880,   880}, //1 BriRatio
     {  3400,  3400,  3400,   400,   390,  3380,  3370,  3360,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350}, //2 BriLimit
     {  2600,  2600,  2600,  2600,  2642,  2684,  2726,  2770,  2812,  2854,  2900,  2900,  2900,  2900,  2900,  2900,  2900,  2900,  2900}, //3 FlatBriTH
     { 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000}, //4 FlatHiBound
     {  7000,  7000,  7000,  7000,  7000,  7166,  7332,  7498,  7664,  7830,  8000,  8000,  8000,  8000,  8000,  8000,  8000,  8000,  8000}, //5 FlatLoBound
     {    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85}, //6 LumaHiBoundRatio
     {    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70,    70}, //7 LumaLoBoundRatio
     {     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0}, //i4LCEPara8
     {     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0}, //i4LCEPara9
     {     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0}  //i4LCEPara10
```

variable name: LVTarget
data range: 0 - 4095
The value to determine the brightness of LV Target

根据不同的LV套用不同的LV target
Target越大,整个画面越亮.

Strength tuning

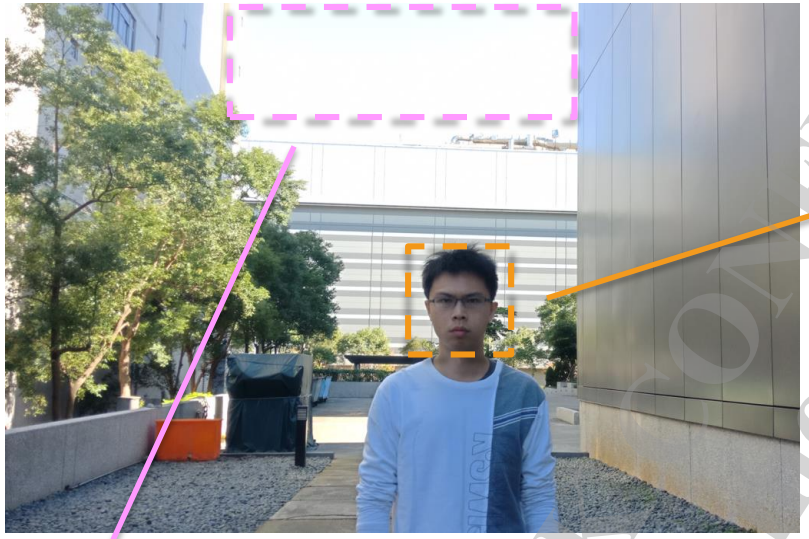
- Dark Strength table : LV and DR idx
- Bright Strength table : LV and DR idx

```
.rLCELUTs = {//i4LCETbl
    {//  /*                                  Dark    Strength                                    */  |  /*   Bright    Strength   */
     //    LV0   LV1   LV2   LV3   LV4   LV5   LV6   LV7   LV8   LV9   LV10  LV11  LV12  LV13  |  LV2   LV4   LV6   LV8   LV10
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  0 DR index
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  1
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  2
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  3
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  4
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  5
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  6
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  7
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  8
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682},   //  9
         {602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,  602,     682,  682,  682,  682,  682}    // 10
    },
```

variable name: i4LCETb1
The Dark/Bright Strength table

LCE 5.0 Improvement in FACE scene

LCE 4.6



The brightness of face keeps.

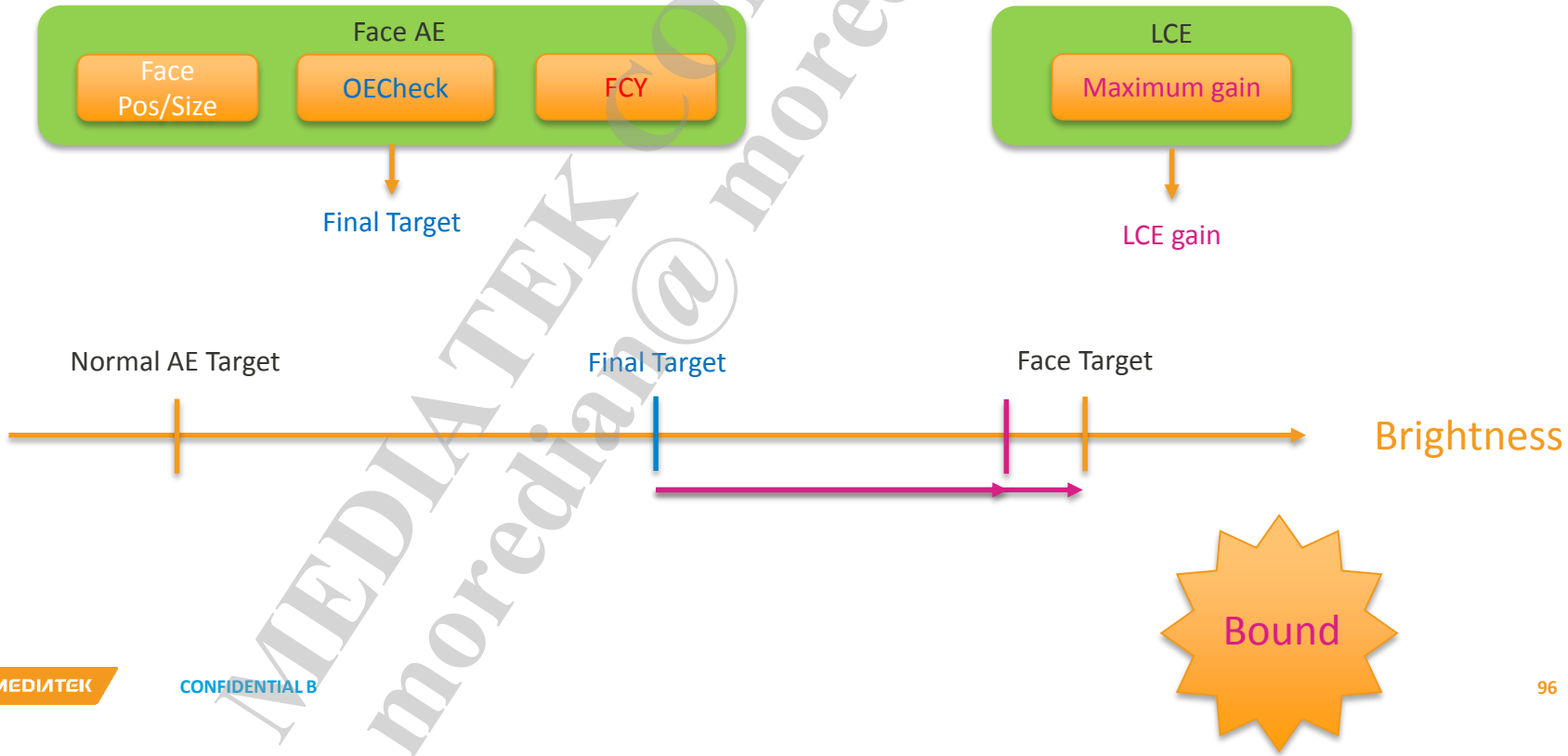
LCE 5.0

Overexposure at background is improved, like sky and the building behind.

Accurate Face Brightness Control

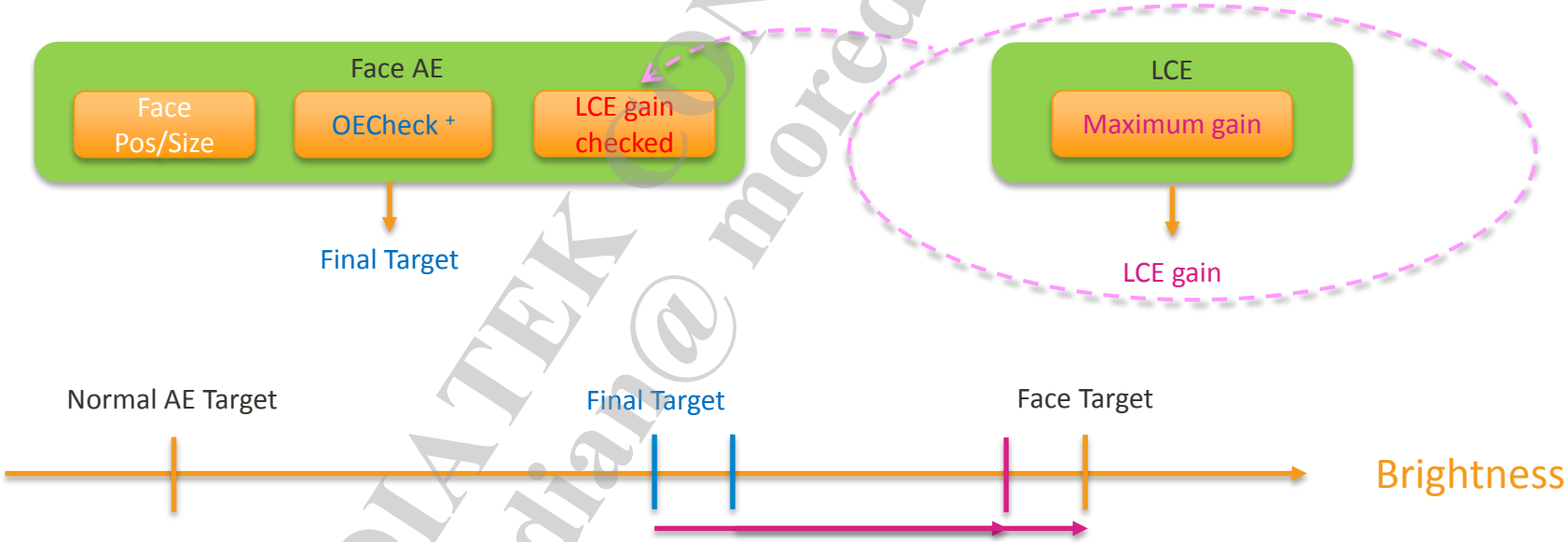
- ISP 4.6

  - AE and LSE consider the face brightness separately.



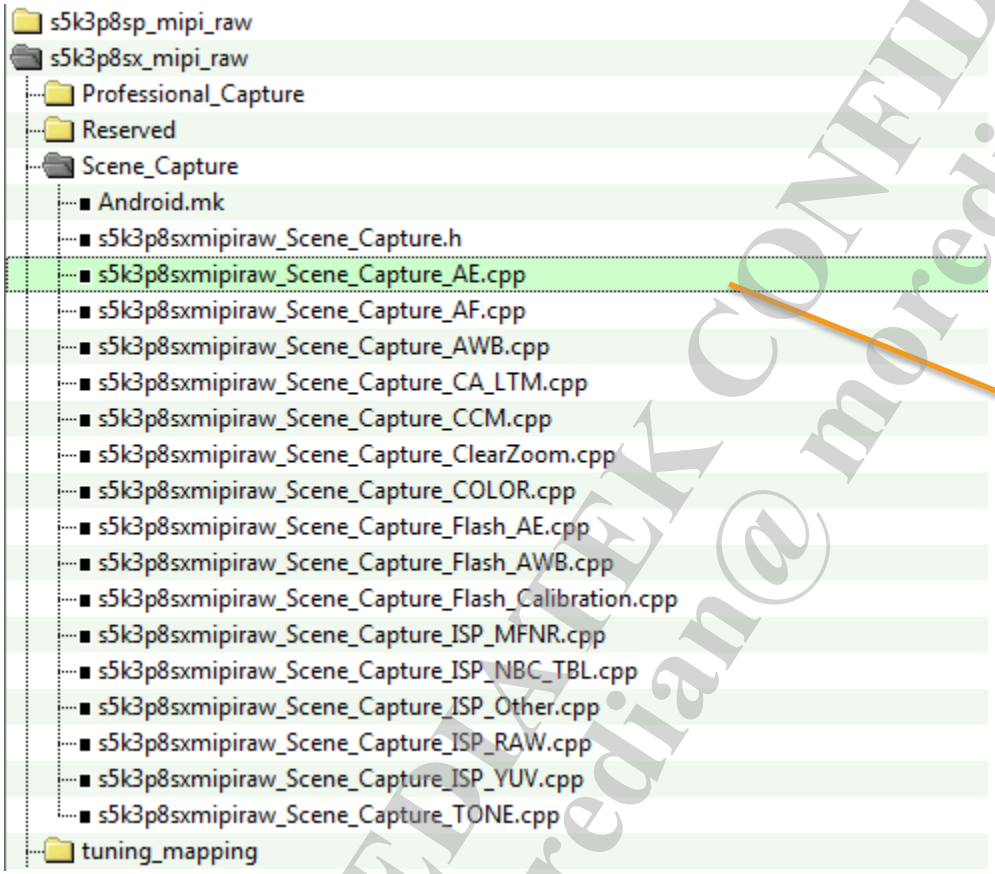
Accurate Face Brightness Control

- ISP 5.0
  - AE and LSE consider the face brightness simultaneously.
    → LCE-AE link



Accurate Face Brightness Control

- [Sensor]_[Scenario]_AE.cpp



LCE-AE link method

```
.rHistConfig = {
    0,
    30,
    200,
    300,
    0,
    1024,
    {1024, 1024, 13, 210, 200},
    {200, 300, 1024, 0, 141},
    {250, 400, 450, 450, 500}
},
```

variable name: LCE-AE link enable
data range: 0 / 1
The flag to disable/enable the LCE-AE link method

OE Check+

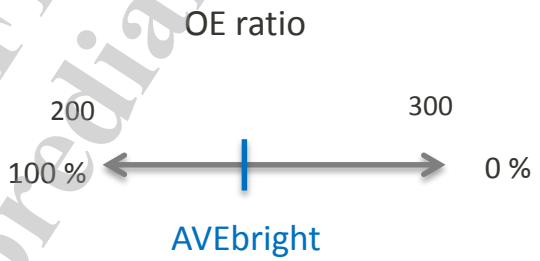
```
.rHistConfig = {
    0,
    30,
    200,
    300,
    0,
    1024,
    {1024, 1024, 13, 210, 200},
    {200, 300, 1024, 0, 141},
    {250, 400, 450, 450, 500}
},
```

variable name: bright part ratio
data range: 0 - 1000
The ratio to define the bright part to calculate AVEbright

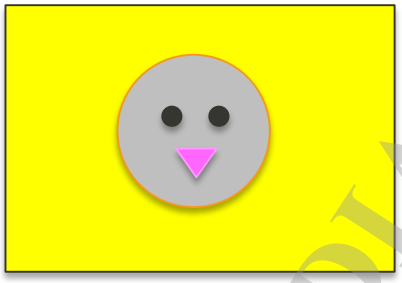
OE Check+

```
.rHistConfig = {
    0,
    30,
    200,
    300,
    0,
    1024,
    {1024, 1024, 13, 210, 200},
    {200, 300, 1024, 0, 141},
    {250, 400, 450, 450, 500}
},
```

variable name: OE table
The ratio table for determine OECheck ratio for mixing the face and normal target



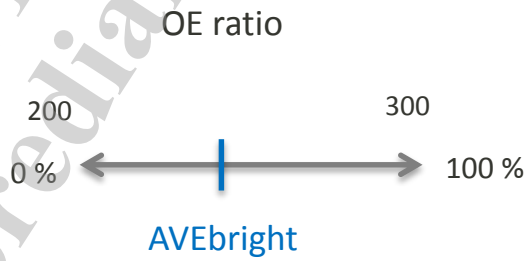
Normal AE Target should be mixed more



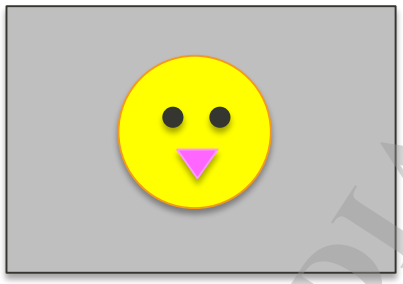
OE Check+

```
.rHistConfig = {
    0,
    30,
    200,
    300,
    0,
    1024,
    {1024, 1024, 13, 210, 200},
    {200, 300, 1024, 0, 141},
    {250, 400, 450, 450, 500}
},
```

variable name: inverse OE table
The ratio table for determine OECheck ratio for mixing the face and normal target



Normal AE Target should be mixed less



LCE Maximum Gain

```
.rHistConfig = {
    0,
    30,
    200,
    300,
    0,
    1024,
    {1024, 1024, 13, 210, 200},
    {200, 300, 1024, 0, 141},
    {250, 400, 450, 450, 500}
},
```

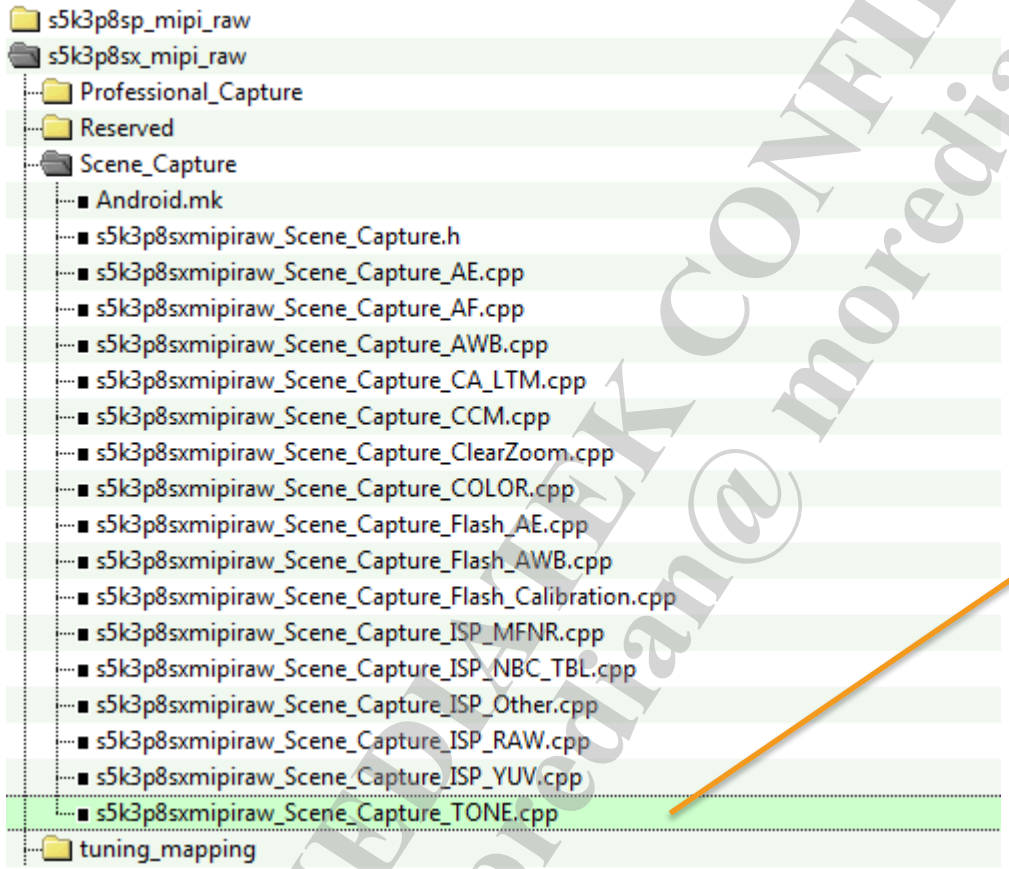
variable name: Maximum LCE gain table
The maximum LCE gain for face enhancement

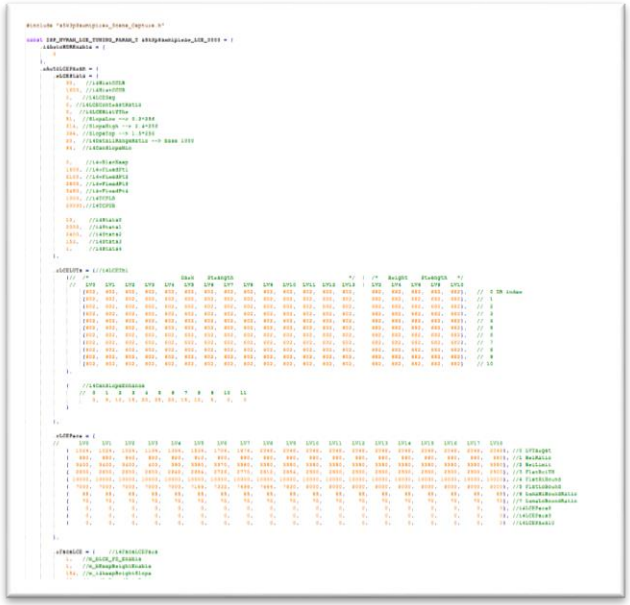
LV	0	5	10	15	18
Maximum LCE gain	250	400	450	450	500

DCE 1.0

Tone Curve Generation

- [Sensor]_[Scenario]_TONE.cpp





Strength tuning

- Dark Strength table : LV and DR idx
- Bright Strength table : LV and DR idx

```
{ //i4DCETbl1
  // DarkStrength                                    BrightStrength
  // LV0  LV2  LV4  LV6  LV8  LV10 LV12 LV14 LV16 LV0  LV2  LV4  LV6  LV8  LV10 LV12 LV14 LV16  TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  0 DR index
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  1
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  2
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  3
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  4
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  5
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  6
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  7
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  8
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, //  9
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // 10
```

variable name: i4DCETbl1
data range: 0 - 100
The Dark/Bright Strength table

Strength tuning

- Dark Strength table in face case : LV
- Bright Strength table in face case : LV

```
{ //i4DCETbl2
  // LV0    LV2   LV4   LV6   LV8  LV10  LV12  LV14  LV16   LV0   LV2   LV4   LV6   LV8  LV10  LV12  LV14  LV16  TBD
    {   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0, 100}, // Face_DarkStrength : Face_BrightStrength
    {3300, 3300, 3300, 3300, 3300, 3300, 3300, 3300, 3300, 3500, 3500, 3500, 3500, 3500, 3500, 3500, 3500, 3500, 100}, // SkyDetectThr : SkyLimitThr
    {  20,   20,   20,   20,   20,   20,   20,   20,   20,   10,   10,   10,   10,   10,   10,   10,   10,   10, 100}, // SkyProtectOnThr : SkyProtectOffThr
    { 150,  150,  150,  150,  150,  150,  150,  150,  150,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // ContourLimitThr
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
}
```

variable name: i4DCETbl2[0] – Face DarkStrength/Face BrightStrength
data range: 0 - 100
The Dark/Bright Strength table

Disable LCE/DCE

- Disable strength:

```
.rLCELUTs = {//i4LCETbl
    {//  /*                                Dark    Strength                                    */ |  /*  Bright    Strength   */
     //    LV0   LV1   LV2   LV3   LV4   LV5   LV6   LV7   LV8   LV9   LV10  LV11  LV12  LV13  |  LV2   LV4   LV6   LV8   LV10
         {1024,1024, 1024, 1024, 1024, 1024, 1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  0 DR index
         {1024,1024, 1024, 1024, 1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  1
         {1024,1024, 1024, 1024, 1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  2
         {1024,1024, 1024, 1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  3
         {1024, 1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  4
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  5
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  6
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  7
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  8
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0},   //  9
         {1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,  1024,     0,  0,  0,  0,  0}    // 10
    },


    { //rDCELUTs
        { //i4DCETbl1
          // DarkStrength                                  BrightStrength
          // LV0  LV2  LV4  LV6  LV8  LV10 LV12 LV14 LV16 LV0  LV2  LV4  LV6  LV8  LV10 LV12 LV14 LV16  TBD
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  0 DR index : , BrightStrength
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  1
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  2
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  3
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  4
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  5
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  6
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  7
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  8
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0},  //  9
            { 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, // 10
        },
```

Disable LCE/DCE

- Disable target:

```
.rLCEPara = {
//      LV0    LV1    LV2    LV3    LV4    LV5    LV6    LV7    LV8    LV9    LV10   LV11   LV12   LV13   LV14   LV15   LV16   LV17   LV18
    {  0,   0,   0,   0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0}, //0 LVTarget
    {   950,   950,   940,   930,   920,   910,   900,   890,   880,   880,   880,   880,   880,   880,   880,   880,   880,   880,   880}, BriRatio
    {  3100,  3100,  3100,  3100,  3090,  3080,  3370,  3360,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350,  3350}, BriLimit
    {  2600,  2600,  2600,  2600,  2642,  2684,  2726,  2770,  2812,  2854,  2900,  2900,  2900,  2900,  2900,  2900,  2900,  2900,  2900}, FlatBriTH
    { 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000}, FlatHiBound
    {  7000,  7000,  7000,  7000,  7000,  7166,  7332,  7498,  7664,  7830,  8000,  8000,  8000,  8000,  8000,  8000,  8000,  8000,  8000}, //5 FlatLoBound
    {    85,    85,    85,    85,    90,    95,    95,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85,    85}, //6 LumaHiBoundRatio
    {    70,    70,    70,    70,    75,    80,    80,    70,    70,    70,    70,    70,    78,    80,    80,    80,    80,    80,    80}, //7 LumaLoBoundRatio
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //i4LCEPara8
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //i4LCEPara9
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}  //i4LCEPara10

},
```

```
{ //i4DCETbl2
  // LV0    LV2    LV4    LV6    LV8    LV10   LV12   LV14   LV16   LV0    LV2    LV4    LV6    LV8    LV10   LV12   LV14   LV16   TBD
    {    0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0, 100}, // Face_DarkStrength : Face_BrightStrength
    { 3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700,  3700, 100}, // SkyDetectThr : SkyLimitThr
    {   20,    20,    20,    20,    20,    20,    20,    20,    20,    10,    10,    10,    10,    10,    10,    10,    10,    10, 100}, // SkyProtectOnThr : SkyProtectOffThr
    { 150,  150,  150,  150,  150,  150,  150,  150,  150,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // ContourLimitThr
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
    { 30,  30,  30,  30,  30,  30,  30,  30,  30,  80,  80,  80,  80,  80,  80,  80,  80,  80,  100}, // TBD
}
```

- Disable face LCE:

```
.rFaceLCE = {      //i4FaceLCEPara
    0,    //m_bLCE_FD_Enable
    1,    //m_bKeepBrightEnable
    154,  //m_i4keepBrightSlope
    89,   //m_u4LoBoundGainRatio
    128,  //m_u4HiBoundGainRatio
    83,   //m_u4BrightTH
    308,  //MaxLCEGain_L
    308,  //MaxLCEGain_H
```

Appendix

Open AE log adb command:

CCU:
adb shell setprop debug.drv.ccu_drv 6
adb shell setprop debug.ccuif.ccu_drv 6

ae algo:
adb shell setprop debug.ae.enable 9

3a:
adb shell setprop debug.aaa.pvlog.enable 1

lce:
adb shell setprop debug.dynamic_lce.log 1
adb shell setprop debug.lce.core.enable 1
adb shell setprop debug.lce.face.log.enable 1
adb shell setprop debug.mapping_mgr.enable 2

everyday genius