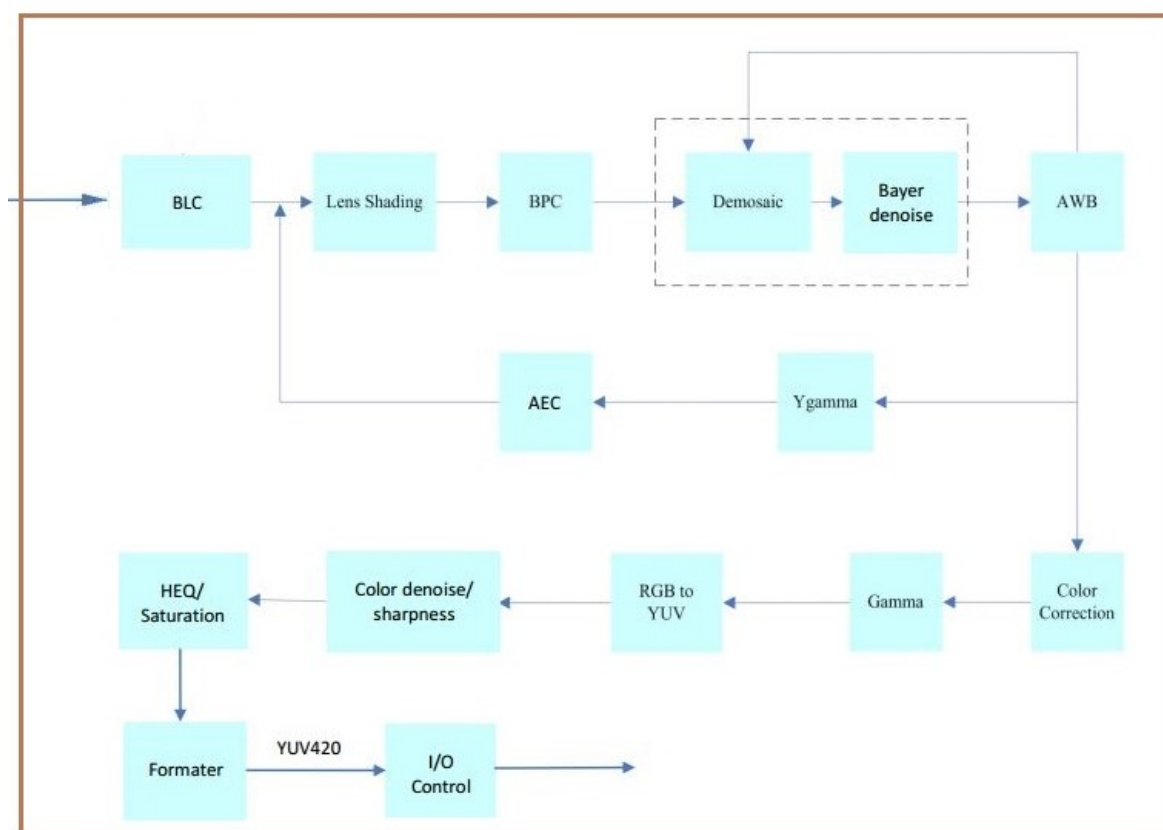


数字信号处理芯片



景物通过 Lens 生成的光学图像投射到 sensor 表面上，经过光电转换为模拟电信号，消噪声后经过 A/D 转换后变为数字图像信号，再送到数字信号处理芯片 (ISP) 中加工处理。所以，从 sensor 端过来的图像是 Bayer 图像，经过黑电平校正 (black level compensation)、镜头阴影矫正 (lens shading correction)、坏像素矫正 (bad pixel correction)、颜色插值 (demosaic)、Bayer 噪声去除、自动白平衡 (awb)、色彩矫正 (color correction)、gamma 矫正、色彩空间转换 (RGB 转换为 YUV)、在 YUV 色彩空间上彩噪去除与边缘加强、色彩与对比度加强，中间还要进行自动曝光控制等，然后输出 YUV (或者 RGB) 格式的数据，再通过 I/O 接口传输到 CPU 中处理。

1.Bayer

图像在将实际的景物转换为图像数据时，通常是将传感器分别接收红、绿、蓝三个分量的信息，然后将红、绿、蓝三个分量的信息合成彩色图像。该方案需要三块滤镜，这样价格昂贵，且不好制造，因为三块滤镜都必须保证每一个像素点都对齐。

通过在黑白 **cmos** 图像传感器的基础上，增加彩色滤波结构和彩色信息处理模块就可以获得图像的彩色信息，再对该彩色信息进行处理，就可以获得色彩逼真的彩色图像。通常把彩色图像传感器表面覆盖的滤波称为彩色滤波阵列（**Color Filter Arrays, CFA**）。

目前最常用的滤镜阵列是棋盘格式的，已经有很多种类的，其中绝大多数的摄像产品采用的是原色贝尔模板彩色滤波阵列（**Bayer Pattern CFA**），**R**、**G**、**B** 分别表示透红色、透绿色和透蓝色的滤镜阵列单元。由于人的视觉对绿色最为敏感，所以在 **Bayer CFA** 中 **G** 分量是 **R** 和 **B** 的二倍，在每个像素点上只能获取一种色彩分量的信息，然后根据该色彩分量的信息通过插值算法得到全色彩图像。

2.黑电平校正-----BLC(Black level Correction)

1、黑电平形成的原因有多种，主要的形成原因如下面两点：

(1) **CMOS** 传感器采集的信息经过一系列转换生成原始 **RAW** 格式数据。以 **8bit** 数据为例，单个 **pixel** 的有效值是 **0~255**，但是实际 **AD** 芯片（模数转换芯片）的精度可能无法将电压值很小的一部分转换出来，因此，**sensor** 厂家一般会在 **AD** 的输入之前加上一个固定的偏移量，使输出的 **pixel value** 在 **5(非固定)~255** 之间，目的是为了让暗部的细节完全保留，当然同时也会损失一些亮部细节，由于对于图像来说，我们的关注度更倾向于暗部区域，**ISP** 后面会有很多增益模块（**LSC**、**AWB**、**Gamma** 等），因此亮区的一点点损失是可以接受的。

(2) **sensor** 的电路本身会存在暗电流，导致在没有光线照射的时候，像素单位也有一定的输出电压，暗电流这个东西跟曝光时间和 **gain** 都有关系，不同的位置也是不一样的。因此在 **gain** 增大的时候，电路的增益增大，暗电流也会增强，因此很多 **ISP** 会选择在不同 **gain** 下减去不同的 **bl** 的值。

如多 **sensor** 输出 **raw** 数据中附加的黑电平值，需要在 **ISP** 最前端去干净。如果不去干净，干扰信息会影响后端 **ISP** 各模块的处理，尤其会导致 **AWB** 容易不准，出现画面整体偏绿或者整体偏红现象。

2、黑电平

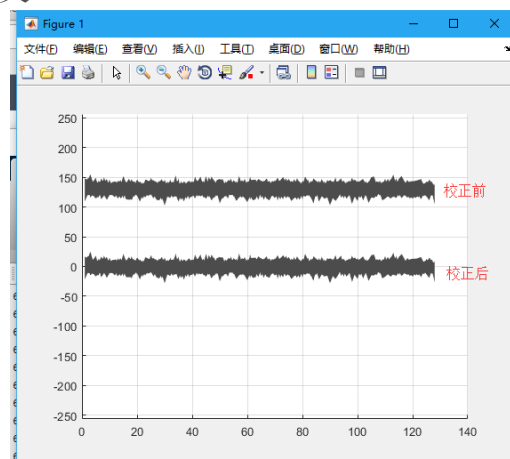
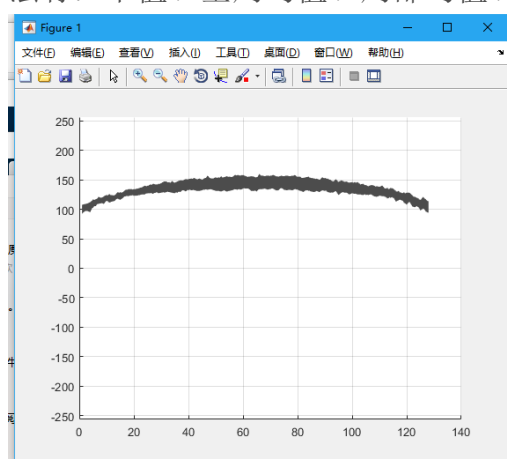
Black Level 是用来定义图像数据为 0 时对应的信号电平。由于暗电流的影响，传感器出来的实际原始数据并不是我们需要的黑平衡（数据不为 0）。所以，为减少暗电流对图像信号的影响，可以采用的有效的方法是从已获得的图像信号中减去参考暗电流信号。一般情况下，在传感器中，实际像素要比有效像素多，像素区头几行作为不感光区（实际上，这部分区域也做了 RGB 的 color filter），用于自动黑电平校正，其平均值作为校正值，然后在下面区域的像素都减去此校正值，那么就可以将黑电平矫正过来了。做了 black level 矫正与没做 black level 矫正的对比，没做 black level 矫正的图片会比较亮，影响图像的对比度。

3、黑电平校正算法

一般 BLC 模块会放在 ISP 比较靠前的位置，因为我们希望图像在进入其他模块之前能够还原最为真实的图像。有些 sensor 会在 sensor 内部集成 BLC 的模块，那么此时 ISP 里的 BLC 模块只做微调即可。在 isp 部分减掉的其实不是 black level，而是 sensor 统一做出来的 pedestal。sensor 端不会将 bl 减完，因为 sensor 输出不能为负数，若将 bl 减完，就等于小于 0 的部分直接就丢掉了，这样做会改变 noise 的分布。

目前主流黑电平校正方案有两种：

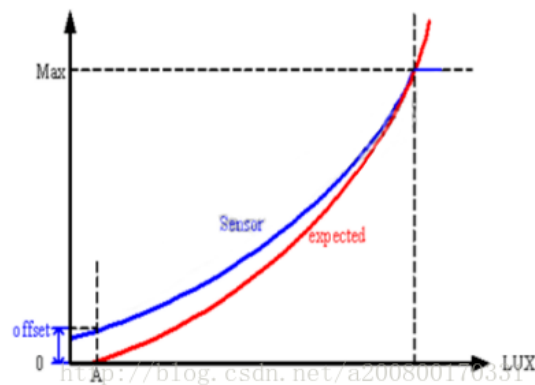
(1) 由于硬件设计人员在设计 BLC 模块时需要考虑效果和成本，因此目前市场上使用的 ISP 一般采用的方法是在 sensor 输出的图像上减去一个固定数值，该固定值，对 RGB 各通道可以是一样，也可以是不一样，目前大多数厂家如安霸、海思等基本都是采用该方案；BLC 各个通道均需要校正，目前比较常用的方法有：中值、全局均值、局部均值、自定义。



校正前需要根据图像的具体情况进行分析，若图像平面趋于平整，则推荐使用全帧均值；若图像出现一些峰值，有明显突出山峰等，推荐使用中值的方法；若出现某个角的值比较高，可能由于电源或者其他的原因引起的，则推荐使用局

部计算的方法；此外，还有自定义、最大值等方法，需要根据不同图像的情况去选择方法，终其目的都是一样的。

(2) 利用黑电平随温度和 gain 的漂移曲线，利用一次函数的方式进行校正，但是对于不同 sensor，漂移曲线不一样，因此该方案没有作为通用方案，如下图：



3.镜头阴影校正-----LSC(Lens Shading Correction)

luma shading：会造成图像边角偏暗，就是所谓的暗角。



由于 Lens 的光学特性，Sensor 影像区的边缘区域接收的光强比中心小，所造成的中心和四角亮度不一致的现象。镜头本身就是一个凸透镜，由于凸透镜原理，中心的感光必然比周边多，造成图像四周亮度相对中心亮度逐渐降低，以及，由于图像光照在透过镜头照射到 pixel 上时，边角处的焦点夹角大于中心焦点夹角，造成边角失光。表现在图像上的效果就是亮度从图像中心到四周逐渐衰减，且离图像中心越远亮度越暗。这个现象就是光学系统中的渐晕。

color shading：中心和四周颜色不一致，体现出来一般为中心或者四周偏色。



由于各种颜色的波长不同，经过了透镜的折射，折射的角度也不一样，因此会造成 color shading 的现象。在图像边缘的地方，其 R、G、B 的值也会出现

偏差，导致 CA(chroma aberration)的出现，因此在矫正渐晕的同时也要考虑各个颜色通道的差异性。

校正

lens shading 的校正是分别对于 bayer 的四个通道进行校正，每个通道的校正过程是相对独立的过程。

常用的镜头校正的具体实现方法是，首先确定图像中间亮度比较均匀的区域，该区域的像素不需要做矫正；以这个区域为中心，计算出各点由于衰减带来的图像变暗的速度，这样就可以计算出相应 R、G、B 通道的补偿因子(即增益)。

4.坏点校正-----BPC(Bad Pixel Correction)

a.坏点

图像传感器中的传感器上每一光线采集的点形成的阵列工艺存在缺陷，或光信号进行转化的过程中出现错误，会造成图像上有些像素的信息有误，导致图像中的像素值不准确，这些有缺陷的像素即为图像坏点(Bad pixel)。

静态坏点：

亮点：一般来说像素点的亮度值是正比于入射光的，而亮点的亮度值明显大于入射光乘以相应比例，并且随着曝光时间的增加，该点的亮度会显著增加；

暗点：无论在什么入射光下，该点的值接近于 0；

动态坏点：

在一定像素范围内，该点表现正常，而超过这一范围，该点表现的比周围像素要亮。与 sensor 温度、增益有关，sensor 温度升高或者 gain 值增大时，动态坏点会变的更加明显。

导致的问题：

图像存在坏点比较多或动态坏点很多的情况下，会造成图像的边缘出现伪色彩的情况，坏点也会造成图像部分 pixel 闪烁的现象。

b.坏点修复方法

一般情况下，RGB 信号应与景物亮度呈线性响应关系，但由于 Sensor 部分 pixel 不良导致输出的信号不正常，出现白点或黑点。坏点修复方法通常有两种：一种是自动检测坏点并自动修复，另一种是建立坏点像素链表进行固定位置的坏像素点修复，这种方式是 OTP 的方式。

静态坏点校正

静态坏点的校正是基于已有的静态坏点表，比较当前点的坐标是否与静态坏点表中的某个坐标一致，若一致则判定为坏点，然后再计算校正结果对其进行校正。

一般情况下，每个 **sensor** 的坏点都不一样，需要 **sensor** 厂商给出每个 **sensor** 的静态坏点表，但是出于成本的考虑，很多 **sensor** 厂商并没有给出，而用户校正的话只能一个一个对其进行校正，因此对于一些低成本的 **sensor**，静态坏点校正的实用性不是很强。

由于在硬件设计的时候需要占用大量的 **memory**，考虑到芯片面积以及一些其他原因，因此静态坏点有大小的限制，不可以无限制的校正。

sensor 的静态坏点表一旦写入存储，**dpc** 模块会自动替换坏点表中所示坏点。

tuning

每个平台标定方法各不相同，但是一般情况下都是根据以下几个步骤进行标定：在黑暗坏环境下标定亮点坏点，有光均匀图像的环境下标定暗点，合并坏点表。

动态坏点校正

动态坏点的校正可以实时的检测和校正 **sensor** 的亮点与暗点，并且校正的坏点个数不受限制。动态坏点校正相对静态坏点校正具有更大的不确定性。

动态 **dpc** 可以分为两个步骤，分别为坏点检测和坏点校正。

5.去马赛克-----Demosaic

目的就是有 **bayer** 数据恢复出完整的 **RGB** 数据。当光线通过 **Bayer** 型 **CFA** (**Color Filter Arrays**)阵列之后，单色光线打在传感器上，每个像素都为单色光，从而理想的 **Bayer** 图是一个较为昏暗的马赛克图。**demosaiced** 并不是和字面的意思一样是为了去除电影中的一些打马赛克的图像，而是数字图像处理中用来从不完整 **color samples** 插值生成完整的 **color samples** 的方法(因为 **bayer pattern** 看起来像一个个马赛克，因此称为去马赛克)。在 **sensor** 端通常需要使用 **CFA** 滤镜来得到 **Bayer pattern**，而在后面的处理中需要把 **bayer pattern** 变成完整的 **RGB444**(真彩色)图像。在 **ISP** 中需要有这么一个模块来做。在传统的 **ISP** 中有很多算法可以来做这个插值，包括最近邻域法，**bilinear** 插值，**cubic** 插值等。

错误插值算法：算法不好，会导致 color artifact(人工错色)：

色调的突然变化导致 **R/G/B** 不协调。在细节多的部分和边缘的地方容易出现这种问题。

a. false color 假色



b. zipped effect 拉链效应



错色也是常常出现在物体的边缘或者明亮变化剧烈的区域。

因此一般的算法中都需要增加边缘检测部分，对边缘进行特殊的处理。

6.贝尔噪声-----Bayer Denoise

噪声来源

RAW 图像的噪声来源分为三类：shot noise，是由于光探测的物理特性；存在像素中的 thermal noise；来自读取过程的 read noise。ISP 通常使用例如 BM3D 和 NLM 的去噪算法，来提高图像的 SNR（一个评价去噪性能的指标），同时注意不能模糊掉一些例如边缘和纹理的图像的重要特征。去噪算法是很昂贵的，因为他们会利用上空间信息，这一点在低照度的场景下尤其困难。

使用 cmos sensor 获取图像，光照程度和传感器问题是生成图像中大量噪声的主要因素。同时，当信号经过 ADC 时，又会引入其他一些噪声。这些噪声会使图像整体变得模糊，而且丢失很多细节，所以需要去噪处理。空间去噪传统的方法有均值滤波、高斯滤波等。但是，一般的高斯滤波在进行采样时主要考虑了像素间的空间距离关系，并没有考虑像素值之间的相似程度，因此这样得到的模糊结果通常是整张图片一团模糊。所以，一般采用非线性去噪算法，例如双边滤波器，在采样时不仅考虑像素在空间距离上的关系，同时加入了像素间的相似程度考虑，因而可以保持原始图像的大体分块，进而保持边缘。

7.自动白平衡-----AWB(Automatic White Balance)

先验：白光 RGB 分量为(R=G=B=255)；灰色的光(R=G=B)。

灰度世界算法 GW(自动白平衡广泛应用)

假设：一副有足够色彩变化的图像，RGB 的均值趋于相等。

$$K = \frac{R_{ave} + G_{ave} + B_{ave}}{3}$$
$$R_A = \frac{K}{R_{ave}} R_B$$

全反射理论算法 PR(自动白平衡最常用之一)

假设：图像最亮的点(定义为 R+G+B 的最大值)是白点，以此进行映射。

$$R_A = \frac{R_W}{R_{\max}} R_B;$$

$$G_A = \frac{G_W}{G_{\max}} G_B;$$

$$B_A = \frac{B_W}{B_{\max}} B_B;$$

R_{\max} ……红色通道最亮点的值，其他类似

R_W ……平衡后红色通道最亮点的值（取255或稍小一些），其他类似

GW 和 PR 正交组合算法 QCGP

GW 和 PR 算法，都是基于假设来处理的，有各自优点和不足，因此将两者结合，缺点改进，优点叠加，能得到更好的算法。采用分配系数的形式，决定两种方法的贡献

$$u'R_{\text{ave}}^2 + v'R_{\text{ave}} = K_{\text{ave}};$$

$$u'R_{\text{max}}^2 + v'R_{\text{max}} = K_{\text{max}};$$

$$K_{\text{ave}} = \frac{R_{\text{ave}} + G_{\text{ave}} + B_{\text{ave}}}{3};$$

$$K_{\text{max}} = \frac{R_{\text{max}} + G_{\text{max}} + B_{\text{max}}}{3};$$

$$R_A = u'R_B^2 + v'R_B$$

然后通过公式的联立求出 u 和 v ，最后求得 RA

色温估计算法



一、色温估计

色温：图像的平均色差

色差：蓝绿色差和红绿色差（Cb 和 Cr），白色色差为 0。

YCbCr 是色彩空间的一种，通常会用于影片中的影像连续处理，或是数字摄影系统中。其中 Y 表示图像的亮度(luma)。

色温估计：在一定约束下，检索到白色像素，进而估计平均色差。

二、增益计算

在一中找出了待选白色区域点，在区域内进行增益，通过增益算法，将 Cb 和 Cr 调整到 0(或接近于 0)。对每个像素进行计算


三、色温校正

在二中找到了增益的计算方法，然后在其他非白色区域引用该增益计算方法。对色温进行校正。将图像每个预测白色像素点的 R 和 B 通道分别乘上 u, v 。重新计算色温校正后图像的色差，判断色差 Cb, Cr 是否已经达到足够小，未达到则返回增益计算。

灰度世界	效果一般，会出现严重色偏 处理色彩单一图像效果不佳
全反射	亮度最大区域不是白色效果不佳 注意RGB溢出大于255
色温估计	迭代原理，重复处理不会出现色偏 改进方法效果较好，可进一步研究

8.色彩校正-----Color Correction

将camera rgb 色彩空间转换为sRGB 色彩空间。
camera_rgb -> XYZ -> sRGB

 CCM 一般在gamma 前面

因此CCM 不是保证图像在CCM 之后变为srgb 色彩空间，而是在gamma 后为srgb色彩空间。

CCM 的计算方法：

ccm 计算时的一些附加功能：
(a) ccm 各patch 的权重；
(b) 可以考虑噪声等选项；
(c) 可以提高或者降低饱和度

(1) 24色卡对用的srgb 求逆 gamma, 然后作为target

`color_target=(color_check_srgb).^(gamma);`

(2) 求得输入图像24色卡每个patch对用的R G B mean；

for color error < th

(3) 设置ccm matrix初始值。

(4) 计算color error

if (color error<th) || (color error达到最小值) || (迭代次数>n)
break;

else

(5) 计算新的ccm matrix 值

end

end

<https://blog.csdn.net/zsl091125>

调试CCM 时的注意事项：

(1) 计算ccm 时曝光需要正常。

(2) ccm 会增强彩噪，在高ISO 时需要降低ccm的saturation，甚至关闭ccm模块。

(3) gamma 变动时，ccm 也需要跟随自己变动。

<https://blog.csdn.net/zsl091125>

由于人类眼睛可见光的频谱响应度和半导体传感器频谱响应度之间存在差别，还有透镜等的影响，得到的 RGB 值颜色会存在偏差，因此必须对颜色进行校正，通常的做法是通过一个 3x3 的颜色变化矩阵来进行颜色矫正。

颜色校正主要为了校正在滤光板处各颜色块之间的颜色渗透带来的颜色误差。一般颜色校正的过程是首先利用该图像传感器拍摄到的图像与标准图像相比较，以此来计算得到一个校正矩阵。该矩阵就是该图像传感器的颜色校正矩阵。在该图像传感器应用的过程中，及可以利用该矩阵对该图像传感器所拍摄的所有图像来进行校正，以获得最接近于物体真实颜色的图像。

一般情况下，对颜色进行校正的过程，都会伴随有对颜色饱和度的调整。颜色的饱和度是指色彩的纯度，某色彩的纯度越高，则其表现的就越鲜明；纯度越低，表现的则比较黯淡。RGB 三原色的饱和度越高，则可显示的色彩范围就越广泛。

9.伽马校正-----Gamma Correction

人眼对外界光源的感光值与输入光强不是呈线性关系的，而是呈指数型关系的。在低照度下，人眼更容易分辨出亮度的变化，随着照度的增加，人眼不易分辨出亮度的变化。而摄像机感光与输入光强呈线性关系，为方便人眼辨识图像，需要将摄像机采集的图像进行 gamma 校正。

Gamma 校正是对输入图像灰度值进行的非线性操作，使输出图像灰度值与输入图像灰度值呈指数关系。

10.色彩空间转换

YUV 是一种基本色彩空间，人眼对亮度改变的敏感性远比对色彩变化大很多，因此，对于人眼而言，亮度分量 Y 要比色度分量 U、V 重要得多。所以，可以适当地抛弃部分 U、V 分量，达到压缩数据的目的。

YCbCr 其实是 YUV 经过缩放和偏移的改动版，Y 表示亮度，Cr、Cb 表示色彩的色差，分别是红色和蓝色的分量。在 YUV 家族中，YCbCr 是在计算机系统中应用最多的成员，其应用领域很广泛，JPEG、MPEG 均采用此格式。一般人们所讲的 YUV 大多是指 YCbCr。YCbCr 有许多取样格式，如 4 : 4 : 4, 4 : 2 : 2, 4 : 1 : 1 和 4 : 2 : 0。

11.颜色域去噪-----Color Denoise

为了抑制图像的彩色噪声，一般采用低通滤波器进行处理。例如使用 $M \times N$ 的高斯低通滤波器在色度通道上进行处理。

12. 自动曝光----- AE (Automatic Exposure)

不同场景下，光照的强度有着很大的差别。人眼有着自适应的能力因此可以很快的调整，使自己可以感应到合适的亮度。而图像传感器却不具有这种自适应能力，因此必须使用自动曝光功能来确保拍摄的照片获得准确的曝光从而具有合适的亮度。

AE 模块实现的功能是：根据自动测光系统获得当前图像的曝光量，再自动配置镜头光圈、**sensor** 快门及增益来获得最佳的图像质量。自动曝光的算法主要分**光圈优先、快门优先、增益优先**。光圈优先时算法会优先调整光圈到合适的位置，再分配曝光时间和增益，只适合 **p-iris** 镜头，这样能均衡噪声和景深。快门优先时算法会优先分配曝光时间，再分配 **sensor** 增益和 **ISP** 增益，这样拍摄快速运动物体。增益优先则是优先分配 **sensor** 增益和 **ISP** 增益，再分配曝光时间，图像噪声会比较小。

自动曝光的实现一般包括三个步骤：**光强测量、场景分析和曝光补偿**。光强测量的过程是利用图像的曝光信息来获得当前光照信息的过程。按照统计方式的不同，分为全局统计，中央权重统计或者加权平均统计方式等。全局统计方式是指将图像全部像素都统计进来，中央权重统计是指只统计图像中间部分，这主要是因为通常情况下图像的主体部分都位于图像的中间部分；加权平均的统计方式是指将图像分为不同的部分，每一部分赋予不同的权重，比如中间部分赋予最大权重，相应的边缘部分则赋予较小的权重，这样统计得到的结果会更加准确。

场景分析是指为了获得当前光照的特殊情况而进行的处理，比如有没有背光照射或者正面强光等场景下。对这些信息的分析，可以提升图像传感器的易用性，并且能大幅度提高图像的质量，这是自动曝光中最为关键的技术。目前常用的场景分析的技术主要有模糊逻辑和人工神经网络算法。这些算法比起固定分区测光算法具有更高的可靠性，主要是因为模糊规则制定或者神经网络的训练过程中已经考虑了各种不同光照条件。在完成了光强测量和场景分析之后，就要控制相应的参数使得曝光调节生效。主要是通过设定曝光时间和曝光增益来实现的。通过光强测量时得到的当前图像的照度和增益值与目标亮度值的比较来获得应该设置的曝光时间和增益调整量。在实际情况下，相机通常还会采用镜头的光圈/快门系统来增加感光的范围。

13. 锐化-----Sharp

CMOS 输入的图像将引入各种噪声，有随机噪声、量化噪声、固定模式噪声等。ISP 降噪处理过程中，势必将在降噪的同时，把一些图像细节给消除了，导致图像不够清晰。为了消除降噪过程中对图像细节的损失，需要对图像进行锐化处理，还原图像的相关细节。