

目录

一，	背景介绍	1
二，	系统运行概念	2
三，	软件架构	3
四，	业务主流程	5
五，	AWB 算法流程	7
六，	AE 算法流程	8
七，	AGC 算法流程	9
八，	曝光控制模式说明	9

一， 背景介绍

由于摄像机和人眼视觉在原理上的不同， 在不同场景下记录下来的画面和人眼看上去差别会比较大， 主要表现在场景色温的影响和光线强度的影响。 人眼对不同物体的色温和光线强度会自动调节， 以取得更合适的效果， 简单的摄像机则没有此功能， 在某些场景下看起来会偏色； 同时人眼对亮度的分辨程度动态范围更宽， 所以可以同时看到很亮的物体和很暗的物体； 而简单的摄像机对信号进行量化后， 动态范围大大降低， 不能同时看到很亮和很暗的物体。

针对上述情况，可以通过调节图像的色温等参数，还有采用调节曝光时间和增益强度等方法， 提高图像质量， 减少和人类视觉上的差异。 自动调节比手动设置参数可以适应很多的场景， 所以一般情况下都使用自动调节白平衡参数和曝光参数的方法，也就是常说的 2A(Auto White Balance & Auto Exposure) 处理。

2A 处理是视频处理流程中比较上游的环节，能够在视频压缩和做其他处理前对图像及其成像条件进行准确的校正非常关键， 否则由于图像信息已经损失太多， 在其他环节很难再正确的还原，研究鲁棒性高，性能好的 2A 算法对图像处理非常有意义。

目前， 2A 是图像处理相关产品的必备模块，比如数码相机，数码摄像机，监控用途的网络摄像机等等，都含有 2A 模块，处理效果也基本令人满意。

从软硬件的实现方式来讲，目前 2A 算法主要有一下几种实现方式： 1，专用芯片方式，图像的控制都在芯片上实现，软件只需要设置少数几个参数； 2，软件方式，一般是在 DSP 上运行算法软件，分析原始视频数据，然后进行相应控制，参数可以比较灵活的设置； 3，软硬结合方式，专用芯片负责分析原始视频数据并生成中间数据，由软件对该数据进行进一步分析和处理，这种方式可以取得速度和灵活性的折中； 4，FPGA 方式，一般在成本影响较小的情况下使用，将算法烧到 FPGA 上运行。

目前大多算法基于专用芯片来实现的， 其缺点是方案成本会升高， 而且不能更加灵活的设置参数； 另外的缺点就是， 各种产品中应用的算法还没有充分优化， 处理没有达到最佳的效果。 比如，某些知名品牌的数码相机， 也会经常拍出曝光过度或者曝光不足的照片； 又比

如，大部分数字摄像机在对着大区域色块时，该区域会褪色。究其原因，是因为目前的算法模型存在某些缺陷，需要进一步改进。

二，系统运行概念

本模块的设计目标是，基于 TI 达芬奇平台的 H3A(Hardware 3A，自动白平衡，自动曝光再加上自动聚焦)模块，研究 2A 算法的一系列特性，取得一种普适性比较好的模型，找出各种模型参数，并在达芬奇平台上具体实现该算法程序。

如前所述，软硬件结合的实现方法既可以取得高性能的硬件加速，又可以通过软件实现新的算法，而且 TI 的达芬奇平台本身就包括 H3A 模块，不需要额外的芯片实现，所以成本上也非常具有优势。本项目就是采用软硬件结合的方法，首先通过驱动从 H3A 硬件模块获取中间数据，然后使用软件算法对中间数据进一步分析，获取控制参数来设置前端的颜色参数和曝光参数。实际上，本项目本身研究的算法模型是各种实现方式都可以使用的，并不局限于软硬件结合的实现方式。

达芬奇平台的 H3A 模块中对于自动曝光和自动白平衡使用的是同一个引擎，称之为 AE/AWB Engine。该引擎首先将整个视频图像在水平和垂直方向上划分为多个子采样，每个子采样被称为一个窗口，然后以窗口为单位为对视频数据中的子采样的 RGB 颜色值进行累加，累加后的值做一个移位运算后输出中间数据 (1.如何做窗口划分 2.如何做累加)。可见 H3A 本身的处理是比较简单的，当然此模块也有一些选项可以配置，比如输出窗口中颜色值最大或者最小的值等等，我们只使用它的基本累加结果。累加值体现了该窗口各颜色值的平均值，这些值可以作为 2A 算法中色温评估和曝光电平评估的依据。对色温进行准确评估后，就可以进行响应控制了：如果色温偏红，就降低红色增益，提高蓝色和绿色增益，对其他颜色同理可推，这就是自动白平衡的基本原理；如果电平偏低 (这个应该是亮度电平?)，就增大曝光时间和增益，反之降低曝光时间和增益，这就是自动曝光控制的基本原理。

原理虽然简单，但是整个环节中会有很多重要的细节，这些细节把握不好，得到的结果就会偏差很多或者不稳定。下面是对本项目算法的基本思想的简单描述。

对于自动白平衡算法，主要难题是准确的找到评价区域，也就是原来是白色，被环境光照射后偏色的区域，而到底是偏色还是本来就是这种颜色是无法区分的，通用的做法是找到接近白色的区域，计算其色温，校正的结果就是将改区域的平均颜色校正为白色，评价是否接近白色的标准就非常重要了。本项目参考了基于 YUV 空间的 Nakano, Lee 等人提出的根据红色和蓝色色差进行评估的方法，提出了基于 HSI 空间的基于饱和度的进行评估的方法，前提是先将 RGB 空间转换为 HIS 空间。这种方法规避了红色和蓝色分量之间的关系难以处理的问题，饱和度是一个单独的参数，反映了颜色的纯度，一般来说饱和度很低的区域，应该就是原来是白色的区域。这些区域因为被染色了，所以会偏色，但还不至于达到很饱和的程度。

对于自动曝光算法，主要难题就是处理曝光控制的稳定性和速度的矛盾。当场景变化时，比如由明转暗或者由暗转明，就要对曝光参数做调节，也维持总体的亮度水平。如果调节的过快，由于反馈的作用，可能会引起亮度的振荡，如果调节的过慢，则会由于不能很快适应

新的场景而丢失场景信息。本项目要找到一种基于精确的线性化曝光模型的自适应控制方法，在避免振荡的同时，能很快达到期望的亮度水平。同时，这个模型还考虑到交流市电频率下的日光灯，荧光灯，交通红绿灯的暗场亮场问题，通过对曝光时间与市电频率进行同步，并对同步引起曝光时间出现的阶跃进行补偿，最终达到线性状态。

三，软件架构

2A 库生成的是一个静态链接库，集成在视频前端处理程序 Myvtest 中调用，同时提供一个测试测试程序，在 Myvtest 启动之后单独运行。软件在架构上分 4 层，说明如下：

	描述	相关文件
应用层	测试程序或者实际的程序	main.c bin/aewtest
业务层	根据算法层获取的数据和用户设定的参数控制设备层参数。	src/dahua_aew.c inc/dahua.aew.h lib/libaew.a
算法层	AWB 和 AE 算法实现	src/aew.h src/ae.c src/awb.c src/agc.c(自动 gamma 校正，还未实现，芯片或者驱动可能有问题)
设备层	操作前端设备，传感器设备，2A 设备的驱动程序。	src/aew.h src/ad9923_app.c src/dm365_aew_app.c src/davinci_preview_app.c

其中，业务层对应用层提供一致的接口 (inc/dahua_aew.h)，保证整个库的可移植性；设备层对业务层也提供一致的接口 (src/aew.h),保证可以支持不同的 CCD,CMOS 和不同的 DSP 平台，当需要更换传感器或者 DSP 平台时，需要重新实现对应的设备，并在业务层创建设备时选用新的设备；算法层基本是统一的，对不同设备，AWB 和 AE 的控制算法区别不大。

业务层函数接口定义如下：

```
// 创建 AEW 线程，分辨率变化时先要销毁线程，再重新开启
// AEW 窗口设置成和视频窗口一致即可
// prev_share_fd：共享 preview 设备的文件句柄号，如果为 -1，由 aew 模块自己创建文件句柄
int dahua_aew_start(AEW_WINDOW *window, int prev_share_fd);
```

```

// 设置颜色参数
int dahua_aew_set_color(COLOR_PARAM *param);

// 设置白平衡参数
int dahua_aew_set_wb_mode(WB_MODE mode);

// 设置日夜模式
int dahua_aew_set_dnc_mode(DNC_MODE mode);

// 设置视频制式
// vstd:0-pal 制 , 1-n 制 , 2-secam 制
int dahua_aew_set_vstd(VSTD vstd);

// 获取支持的曝光等级数
// 返回值 <0 , 获取失败 , >=0 表示等级数 , 具体等级保存在数组 speeds中。
// speeds 数组长度必须足够大 , 取 16 差不多了。
int dahua_aew_get_exposure_speeds(VSTD vstd, float *speeds);

// 设置曝光等级
// 0 自动曝光
// 1~n-1 手动曝光等级
// n 带时间上下限的自动曝光
// n+1 自定义时间手动曝光
// (n 表示支持的曝光等级数 , 即 dahua_aew_get_exposure_speeds接口返回值 )
// value1: 自动曝光时间下限或者手动曝光自定义时间 , 毫秒为单位 , 取值 0.1ms~80ms
// value2: 自动曝光时间上限 , 毫秒为单位取值 0.1ms~80ms , 且必须不小于 exposureValue1
// 取值
int dahua_aew_set_exposure_level(int level, float value1, float value2);

// 设置增益值
// gain: 固定增益值 , 或者自动增益的上限 , 取值 0-100。
// auto_gain: 自动增益 , 0-不开启 , 1-开启 , 同时 gain 成员表示自动增益的上限。
int dahua_aew_set_gain(int gain, int auto_gain);

// 销毁 AEW 线程
int dahua_aew_stop();

```

设备层函数接口定义如下：

```

// H3A 设备
typedef struct H3A_DEVICE
{

```

```

        int (*open)( AEW_WINDOW *window );
        int (*close)( void );
        int (*read)( AEW_PAXEL_STAT *stat, AEW_PAXEL_INFO * info, int paxel_count );
    }H3A_DEVICE;

void create_dm365_h3a(H3A_DEVICE* device);

// 预览设备
typedef struct PREVIEW_DEVICE
{
    int (*open)( int share_fd );
    int (*close)( void );
    int (*set_wb)( WB_GAIN_PARAM *param );
    int (*set_gamma)( GAMMA_TABLE *table );
    int (*set_color)( COLOR_PARAM *param, DNC_MODE mode);
}PREVIEW_DEVICE;

void create_davinci_previewer(PREVIEW_DEVICE* device);

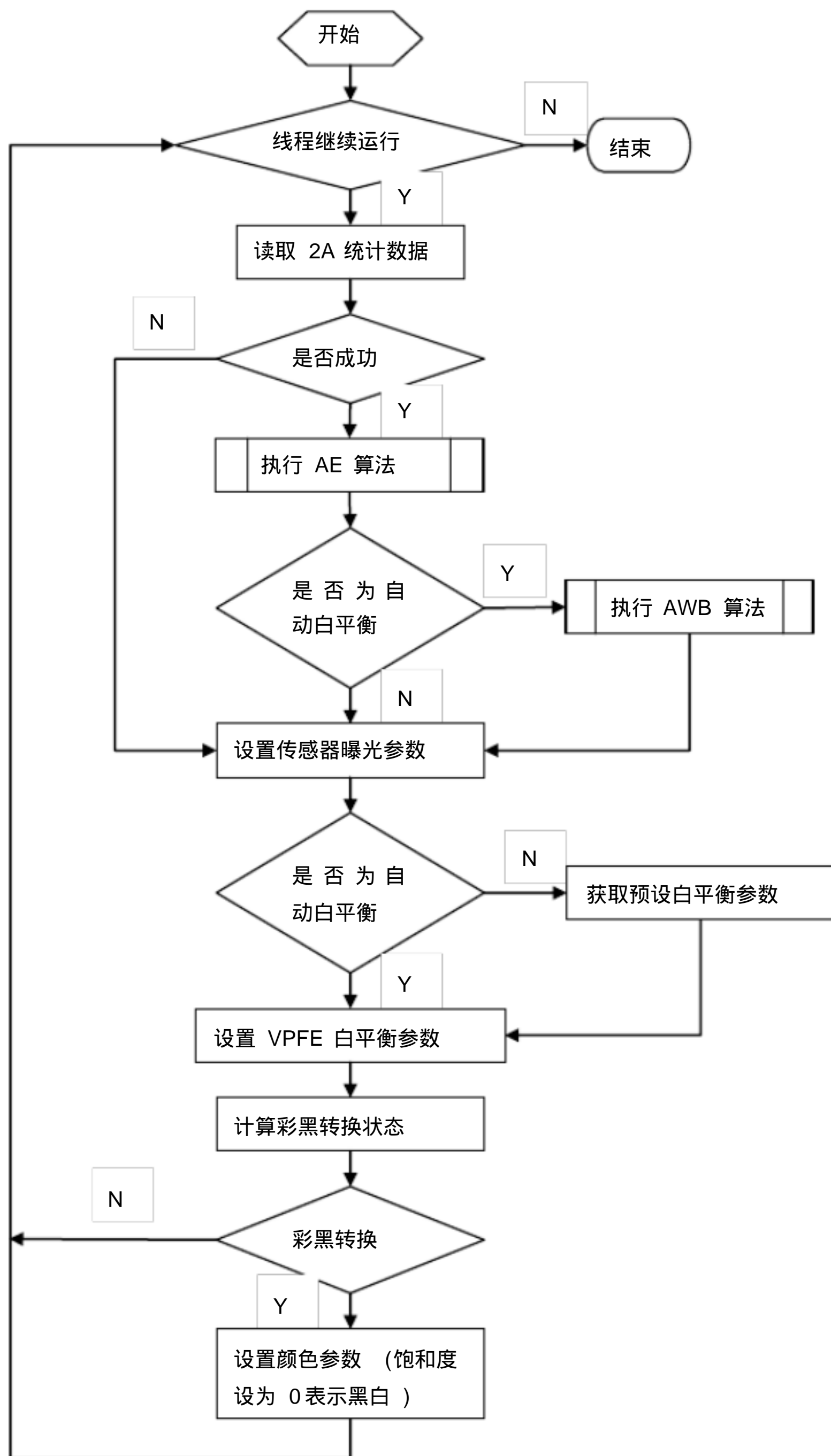
// 前端设备
typedef struct SENSOR_DEVICE
{
    int (*open)( void );
    int (*close)( void );
    int (*set_exposure)( EXPLORE_PARAM * param);
    int (*get_exposure_speeds)(VSTD vstd, float *speeds);
    int (*set_vstd)(VSTD vstd);
    int (*set_exposure_level)(int level, float value1, float value2);
    int (*set_gain)(int gain, int auto_gain);
}SENSOR_DEVICE;

void create_ad9923_ccd(SENSOR_DEVICE* device);

```

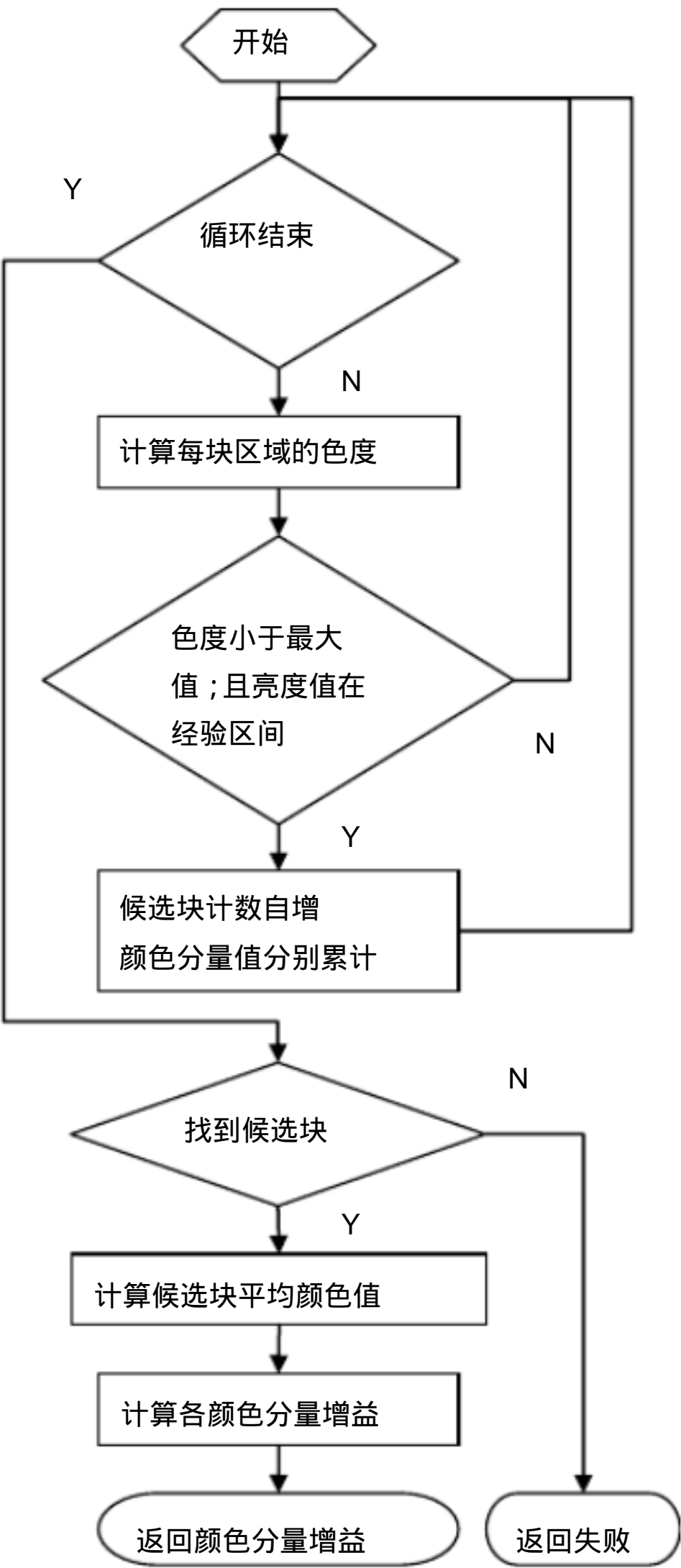
四， 业务主流程

业务流程在 dahua_aew.c文件中的 dahua_aew_thread 线程体现，外部调用业务开始接口后，线程里的循环体一直循环执行，直到外部调用停止接口。



五， AWB 算法流程

根据 H3A 的统计值，找到饱和度和亮度符合经验值的块，计算其颜色平均值，在计算出校正增益，以将其平均值校正为灰颜色（色度为 0）。饱和度经验值是根据色卡测试的结果，接近灰色的区域被认为是被环境光染色的， 计入候选块； 差别较大的认为是物体本身的颜色，不计入候选块。 亮度经验值是理论计算结合实际测试得到， 亮度很低的块由于基数小，计算出的饱和度误差会比较大； 亮度很高的块颜色已经溢出， 饱和度也会不准确， 都不计入候选块。

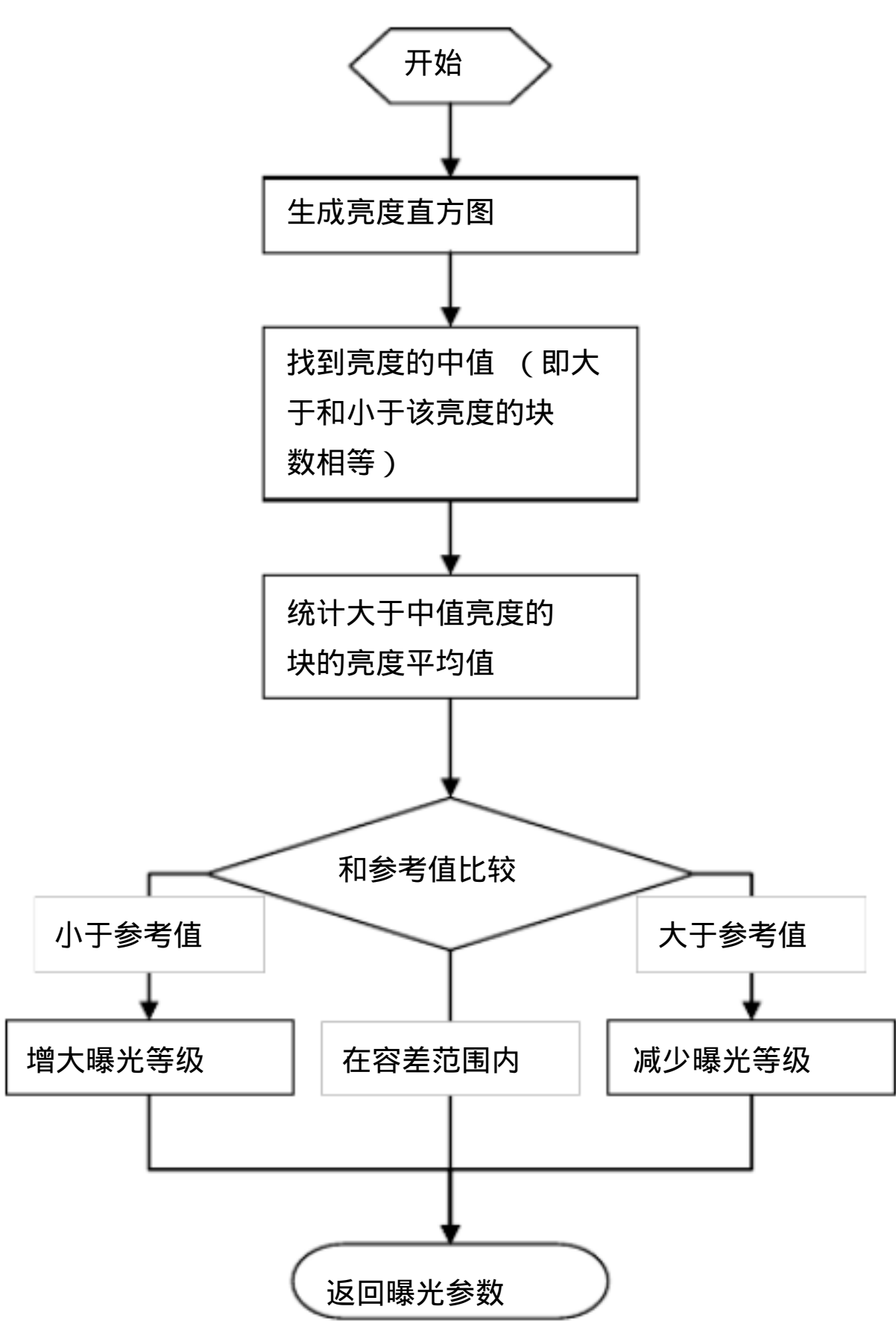


六， AE 算法流程

根据 H3A 的统计值，对全局亮度进行评价（目前只支持全局模式，不支持局部模式），和参考亮度进行对比，计算出新的曝光等级值，设置到前端，利用这个反馈环路，将全局亮度逐步控制在参考亮度左右。

具体方法：如果比参考亮度小，则增大曝光等级；如果比参考亮度小，则减小曝光等级；如果在容差范围内，就是说和参考亮度差值的绝对值比较小的时候，保持目前的曝光等级，避免抖动。

另外计算全局亮度时，采用了中值算法，而不是简单的平均值算法，是为了强调高量区域对整体亮度的影响，一般情况下，一定区域已经是高亮度时，不应该再提高整体的亮度，否则会造成高亮区域颜色溢出。



七， AGC 算法流程

自动 GAMMA 校正（不是指自动增益控制）基本原理是，根据 VPFE 硬件直方图的结构，自动生成 GAMMA 曲线，再设置到 VPFE，这样可以对不同的场景随时取得最好的色彩平衡，主观感觉就是图像动态范围变大，透彻度提升。但是由于 365 的 gamma 设置看起来有些问题，设置后图像偏灰，怎么调也调不好，所有没有实现。

八， 曝光控制模式说明

曝光主要是两个参数的控制，一个是曝光时间，一个是增益，这两个参数也可以由应用层单独指定或者同时指定，没有指定的参数根据 AE 的结果自动调整。在曝光时间和增益都是自动的情况下，曝光时间随曝光等级增大，增益也同时增大，两者是单独调整的，暂无优先级控制。控制策略如下表：

曝光模式	对应处理	特点
自动曝光	将 AE 计算出来的曝光等级换算为曝光时间，从不同的制式的曝光时间表中取出最接近的值，换算为 AD 寄存器值。	只支持大曝光时间 曝光时间与当前制式下的市电频率匹配，保证不会出现闪烁。
手动曝光	从列表中指定一个曝光时间，时间值直接换算为 AD 寄存器值。	只支持小曝光时间。
上下限曝光	将 AE 计算出来的曝光等级换算为曝光时间，再用上下限参数截取，最后换算成 AD 寄存器值。	支持的范围较灵活，但 VD 信号未同步时可能会出现闪烁。
自定义曝光	直接输入曝光时间，换算成 AD 寄存器值。	直接设置曝光时间，对测试比较方便。

增益模式	对应处理	特点
自动增益	将 AE 计算出来的曝光等级换算为增益值，再用自动增益的上限进行解决，最后换算为 AD 寄存器值。 如果曝光模式是自动曝光模式，由于自动调节时曝光时间不连续，存	增益上限可以设置，默认值为 AD 最大增益的一半，用于控制信噪比。 自动增益和自动曝光同时开启的情况下，参数值存在阶跃，又由于增益寄存器和曝光时间寄存器的设置

	在一个阶跃，计算增益值时也会考虑进去。		存在时间差，可能会引起画面短暂闪烁。
手动曝光	直接输入增益百分比，换算成寄存器值。	AD	直接设置增益值，对测试比较方便。