影像学习

算法学习

影像的学习我之前工作其实已经接触过了，但是我其实一直没有懂，什么是影像算法，又怎么用，昨天晚上我了解了一下，觉得其实影像系统也不复杂。

首先说我自己的学习思路：

摄像头的组成：

结合终端和嵌入式的特征，首先影像必须需要原始数据，原始数据通过摄像头传入，摄像头分为了以下**四大核心模块**：

模块类别	核心组成部件	核心作用
1. 光学系统	镜头组（由多片镜片组成）、光圈、对焦马达	负责 “收集光线” 并将场景清晰地投射到 CMOS 传感器上，类似人眼的 “晶状体”
2. 成像与信号转换模块	CMOS 图像传感器	将镜头投射的 “光信号” 转化为 “电信号”（核心环节）
3. 信号处理模块	图像信号处理器（ISP）、降噪 / 白平衡电路	对 CMOS 输出的原始电信号进行优化：修正色彩偏差（白平衡）、降低噪点、提升对比度，最终生成可用的数字图像
4. 控制与接口模块	主控芯片、对焦 / 防抖控制电路、数据接口（USB/CSI 等）	控制摄像头的工作状态（如启动、对焦、曝光），并将处理后的图像信号传输到电脑、手机等终端设备

四大核心模块中，软件处理部分需要注意的就是**CMOS传感器**传回来的RAW（**RAW Image Format**）原始图像格式，如果直接看是灰度图。**OPPO 的 RAW 格式通常以 DNG（Digital Negative）格式存储。**
然后还有一部分就是**ISP（图像信号处理器）处理**，ISP可以在对图像进行处理后将处理后的RGB/YUV/JPEG/PNG格式传回给软件部分。

ISP 是一个专门用于处理图像信号的硬件模块。它可以对图像传感器输出的原始图像数据进行一系列的处理和优化操作，如去噪、色彩校正、白平衡调整、锐化、自动曝光等，这些处理过程是通过 ISP 内部的数字信号处理单元（DSP）等硬件电路来实现。
同时，ISP 也具有加速处理的功能。它作为专用的硬件模块，能够高速处理图像信号，将原本需要主处理器花费大量时间和资源来处理的图像信号处理任务承担下来，

减轻主处理器的负担，提高整个系统的运行效率，因此可以说它具有加速处理的作用。

ISP 的核心任务是完成从 RAW 到 RGB/YUV 的处理（这是它的 “本职工作”），而在后续的图像 pipeline 中，它更多是通过预处理为其他模块（如编码器）提供优化后的数据，起到协同加速整个系统的作用。
pipeline（理解为流水线），包含了所有流程，也把所有流程分割成了各个部分，下面是摄像头成像的**pipeline（RAW-RGB-YUV-JEPG/PNG或MPEG4/MOV）：**

Pipeline 阶段（子步骤）	负责模块	核心任务	输入数据	输出数据
1. 光信号采集	光学系统 + CMOS	镜头聚光→CMOS 将光转电信号	光线	RAW 原始数据
2. RAW 数据预处理	ISP	坏点修复、黑电平校正	RAW 数据	优化后的 RAW 数据
3. RAW 转 RGB	ISP	拜耳插值（去马赛克）	优化后的 RAW 数据	RGB 数据
4. RGB 优化	ISP	降噪、白平衡、HDR 合成	RGB 数据	优化后的 RGB 数据
5. RGB 转 YUV	ISP	色彩空间转换（分离亮度 / 色度）	优化后的 RGB 数据	YUV 数据
6. 压缩编码	硬件编码器	JPEG 压缩（或视频编码）	YUV 数据	JPEG 图片（或视频流）
7. 输出 / 存储	控制 / 接口模块	传输到屏幕 / 保存到存储	JPEG 数据	可显示 / 可存储的图像

但是还有一个流程就是RAW直接转YUV，就不需要RGB这个流程了（节约资源但品质差）。

RAW转RGB还是YUV？

这里就要讲一下了，首先光学系统+CMOS传感器首先采集到的信息是RAW格式，然后一般会ISP直接处理

1. 通过拜尔插值转化为RGB再通过Y/UV算法变成YUV格式。
2. 直接融合拜尔插值+Y/UV(色彩空间转换)变成YUV格式。

传统路径（RAW→RGB→YUV）	融合路径（RAW 直接→YUV）
1. RAW 预处理（坏点修复、黑电平校正） 2. RAW→RGB（拜耳插值 / 去马赛克） 3. RGB 优化（降噪、白平衡、锐化） 4. RGB→YUV（色彩空间转换）	1. RAW 预处理（坏点修复、黑电平校正） 2. **RAW→YUV（融合拜耳插值 + 色彩空间转换）** 3. YUV 优化（降噪、白平衡、锐化，在 YUV 空间完成）

其实这样就少了一步RGB优化，所以色彩精度会变小，但是功耗会小很多，处理速度也会变快。

RGB格式一般就是我们屏幕上显示的时候会转换为的模式，我问了个问题：

“手机显示的时候也应该需要RGB吧，感觉如果经过中间转换，然后保留下来也挺好”

答案：

ISP 输出 YUV，比输出 RGB 节省 50%+ 的 “数据搬运成本”

显示前的 “YUV→RGB 转换”，成本几乎可忽略

所以总结一下现在我们知道了流程

RAW数据通过传感器和光学模块采集后→ISP直接进行处理→生成RGB或者YUV数据（记住RGB和YUV是数据，工作中能看到相应图片是因为被特殊处理了，特殊工具打开，RAW也一样，其实JPEG也一样）

这里的话就涉及到了ISP，ISP是个图像信号处理器，ISP模块分为了SoC/CMOS集成式和外挂独立ISP两种。

SoC集成式的主要是手机用: SoC 集成的 ISP 通常具有较强的处理能力和丰富的功能
CMOS 的 ISP主要是物联网摄像头和智能手表等设备用：成本低
外挂的独立ISP主要是更高端更需要精度专注于影像模块的使用：可灵活适配不同传感器，支持复杂场景算法。要使用的话非常复杂。

这里你肯定会好奇，既然摄像头的核心模块是ISP,那么外挂和SoC又有个ISP，怎么处理呢？

手机为例：

低端机中ISP做预处理（功耗低），然后SoC的ISP做后面包括拜尔插值YUV转换等操作。

高端机中直接用的没有ISP的摄像头，SoC直接代替做全部流程。

这里直接**对比不同应用场景的高端摄像头模块和单片机摄像头模块**

对比维度	三星 GN1 摄像头（高端手机主摄）	OV7670 模块（STM32F103 常用）
核心架构	光学系统（7P 镜头 + 红外滤光片）+ 1/1.3 英寸大底 CMOS（无内置 ISP）+ 驱动电路（含 VCM 对焦 / 光学防抖）	简易光学镜头（3P 塑料镜片）+ 1/6.5 英寸小底 CMOS（含基础 ISP 功能）+ 极简驱动电路（无自动对焦）
ISP 方案	无内置 ISP，依赖手机 SoC 的 ISP（如骁龙 Spectra、天玑 Imagiq）完成全流程处理（RAW→RGB→YUV）	CMOS 内置基础 ISP，可直接输出 YUV422/RGB565（支持简单降噪、白平衡），无需外部 ISP
分辨率与像素	5000 万像素（默认输出 1250 万像素，通过像素合并提升画质），支持 4K 60fps 视频	30 万像素（640×480），最高支持 VGA（640×480）@30fps 视频
接口类型	MIPI-CSI2 高速接口（传输速率达 4Gbps），仅输出 RAW 数据	8 位并行接口（或通过 FIFO 转接 SPI），输出 YUV/RGB 等预处理后的数据
感光能力	1.2μm 像素尺寸（合并后 2.4μm），支持双原生 ISO（高感光低噪点），低光性能优异	3.0μm 像素尺寸（小底，实际进光量低），无原生 ISO 优化，低光噪点明显
特殊功能	支持光学防抖（OIS）、相位对焦（PDAF）、HDR 多帧合成、10bit 色深	无防抖、无自动对焦，仅支持基础曝光控制，8bit 色深
功耗与体积	功耗较高（约 150-200mW），体积较大（需容纳光学防抖结构）	功耗极低（约 10-20mW），体积小巧（20×20mm 模组）
适配主控	仅支持高端手机 SoC（如骁龙 8 系、天玑 9 系），需复杂驱动和算法协同	适配 8 位 / 32 位 MCU（如 STM32F103），驱动简单（I2C 配置 + 并行数据读取）
典型应用场景	高端手机主摄（追求高画质、夜景、变焦）	嵌入式简易监控、玩具车摄像头、低功耗物联网设备（仅需基础图像采集）
数据处理链路	光学镜头→CMOS（RAW 输出）→SoC ISP（全流程处理）→YUV→编码（JPEG / 视频）	简易镜头→CMOS（内置 ISP 预处理）→直接输出 YUV/RGB→STM32（存储 / 传输，无二次处理）

区别还是很大的。

🤔 为什么要分为预览算法和拍照算法

🎗️ 什么是APS