

架构要点(前端 + 后端 = 图的两大分层)

前端(RAW/Bayer 域)

- **RAW** 输入源: 真实 dump + 噪声/PRNU/DSNU/滚快门/抖动模拟; 可输出多曝光/多摄同步帧。
- **RAW** 预处理节点: DPC/BLC/LSC(AI 或经典)、RAW 去噪 + 去马赛克(UNet/轻量注意)。
- **Burst/Bracket** 缓冲: 滑窗聚合为 Flow(带帧列表与对齐/深度等元数据)。
- **HDR** 栈: DL 对齐(含运动显著性 mask) → 内容自适应融合 → RAW/HDR TM(可走预览快路径)。

后端(RGB/YUV 域)

- 轻量 **RGB** 模块: AWB(统计+学习)、EE(可解释锐化)、Gamma/CCM/CSC。
- 多摄协同: 校准/统一坐标与色彩 → 无感变焦(内容感知拼接与 seam 抑制) → 多摄堆栈。
- 视频栈: EIS(运动建模 → 轨迹优化) → RS-aware 几何重采样 → VSR/RTSR(时序一致性)。
- **IQA** 与策略: 离线 BRISQUE/NIQE/LPIPS/NIMA + 在线轻量 IQA; 语义分类(天空/肤色/逆光/低照) → 策略引擎映射(TM 曲线、HDR 强度、EIS 强度、变焦模式)。

关键设计, 确保“能撑起计算摄影”的 10 条硬指标

1. 图结构一等公民: 每个模块是节点; HDR、夜景、变焦、视频都只是不同子图(示例已写入 YAML 的 `pipelines`)。
2. 统一数据模型: `Frame`(图像+曝光/ISO/姿态/内参/IMU) 与 `Flow`(帧组+对齐/深度)。这保证多帧/多摄/视频在一个接口下工作。

3. 可切换实现:同一节点支持 `classic|ai` 参数(示例:`demosaic.method`), 方便 A/B 与回退。
4. 早期 **RAW** 域 **AI**:RAW→RGB 之前就把噪声/马赛克/LSC 处理掉, 为后端融合与 TM 打好地基。
5. 对齐即能力上限:单独抽象 `align_dl` 节点, 输出光流/单应/深度等, 供 HDR、堆栈、EIS、VSR 共享。
6. 多摄先校准:`multicam_calib` 节点统一几何/色彩, `seamless_zoom` 做跨摄 blending 与无缝切换。
7. 时序一致性:视频侧把 EIS 放在前、VSR 在后, 且在 `Flow.alignment` 中留时序轨迹, 避免“先超分再稳像”的伪影放大。
8. 策略闭环:`scene_semantics` + `policy_engine` 作为控制平面, 控制数据平面的 TM/HDR/EIS/变焦强度, 保证观感稳定。
9. **IQA** 门禁:`iqa_offline` 是“汇点”, 所有子图都能接入, 做 nightly 回归和 AB 决策。
10. 仿真到落地:节点参数里保留 `tiled/int8/model_name/latency_ms_budget` 等, 后续 SoC 移植只需替换 kernel/Runtime, 不动图。

典型子图组合(直接用 YAML 里的 pipelines)

- **Photo:单摄 HDR**
`raw → RAW-AI → burst → align → hdr_fusion → tone_mapping → awb → ee → iqa`
- **Photo:多摄无感变焦**
`multicam_mux → calib → seamless_zoom → iqa`
- **Video: HDR + EIS + VSR(实时)**
`raw → RAW-AI → burst → align → tone_mapping → semantics → policy`
`raw → burst → eis_motion → warp_resample → vsr → iqa(并行分支后合流/对比)`

如何开始跑起来(建议 3 步)

1. 按 **YAML** 装配最小可运行链:从 `photo_single_cam` 开始, 把 `raw_ai_dn_dm/demosaic/awb` 的占位实现写进 `isp_graph_api_stub.py`。
2. 接 **RAW** 样片与仿真器输入:先跑经典实现拿到基线, 再切换到 AI 实现看 IQA 提升。
3. 接入 **HDR** 对齐/融合与 **IQA**:把 `align_dl`、`hdr_fusion`、`tone_mapping` 实现连起来, 导出报告。