

Rockchip_Development_Guide_FEC_CN

文件标识: RK-YH-YF-635

发布版本: V0.6.0

日期: 2025-06-07

文件密级: 绝密 秘密 内部资料 公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2025 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文档描述 FEC 模块的功能、如何使用与开发。

产品版本

芯片名称	内核版本
RK3588	Linux 5.10
RV1126B	Linux 6.1

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

各芯片系统支持状态

芯片名称	BuildRoot	Debian	Yocto	Android
RK3588	Y	Y	Y	Y
RV1126B	Y	Y	Y	N

修订记录

版本号	作者	修改日期	修改说明
V0.1.0	徐鸿飞	2024-12-04	初始版本
V0.2.0	黄尚斌、徐鸿飞	2024-12-10	增加生成矫正表方法说明
V0.3.0	徐鸿飞	2025-04-22	<p>第 2 章 功能描述 2.2 小节增加 RV1126B FEC 规格说明</p> <p>第 3 章 API 参考 增加 rk_ispfec_api_set_cfg、rk_ispfec_api_get_cfg、rk_ispfec_api_lock_process、rk_ispfec_api_unlock_process、rk_ispfec_api_set_mesh_config、rk_ispfec_api_get_mesh_config 函数说明。</p> <p>第 4 章 数据类型 增加 in_fourcc、out_fourcc、rk_ispfec_mesh_config、rk_ispfec_buffer_info 类型说明。</p> <p>第 5 章 FAQ 增加 RV1126B 相关说明</p>
V0.4.0	徐鸿飞	2025-04-24	<p>第 3 章 API 参考 增加 ISP_FEC_CONFIG_INIT 说明。</p> <p>第 4 章 数据类型 rk_ispfec_cfg_t 新增字段说明。 gen_mesh_online_info_t 新增字段说明。 其他数据类型描述涉及修改。</p> <p>第 5 章 FAQ 5.3 小节，增加 RV1126B 性能优化建议。</p>
V0.4.1	徐鸿飞	2025-04-24	<p>第 2 章 功能描述 2.2 小节，表 2-3、表 2-4 涉及修改。</p> <p>第 3 章 API 参考 rk_ispfec_api_set_mesh_config 【注意】描述涉及修改</p> <p>第 5 章 FAQ 5.3.2 小节，涉及修改。</p>
V0.4.2	徐鸿飞	2025-04-28	<p>第 3 章 API 参考 rk_ispfec_api_calFecMeshsize 涉及修改</p>

版本号	作者	修改日期	修改说明
V0.5.0	徐鸿飞	2025-06-06	<p>第 1 章 概述 增加 FEC 使用典型框图。 修改性能规格及约束。</p> <p>第 2 章 软件开发 增加软件架构说明，修改软件流程说明。</p> <p>第 3 章 增加 rk_ispfec_api_load_online_config_from_ini 说明。</p> <p>第 4 章 rk_ispfec_cfg_t 【注意事项】描述涉及修改。 in_fourcc、out_fourcc、rk_ispfec_update_mesh_mode 涉及修改。</p>
V0.6.0	徐鸿飞	2025-06-07	<p>第 5 章 新增 Proc 调试信息说明。</p> <p>第 6 章 章节编号涉及修改：从 5 修改为 6。</p> <p>第 7 章 章节编号涉及修改：从 6 修改为 7。</p>

目录

Rockchip_Development_Guide_FEC_CN

1. 概述
 - 1.1 FEC 框图
 - 1.1.1 图1-1 RK3588 FEC 典型框图
 - 1.1.2 图1-2 RV1126B FEC 典型框图
 - 1.2 性能规格及约束概述
 - 1.2.1 图像数据存储方案概述
 - 1.2.2 RK3588 性能规格及约束
 - 1.2.2.1 规格约束
 - 1.2.2.2 性能
 - 1.2.3 RV1126B 性能规格及约束
 - 1.2.3.1 规格约束
 - 1.2.3.2 性能
2. 软件开发
 - 2.1 软件架构
 - 2.2 软件流程
 - 2.2.1 数据流程框图
 - 2.2.2 主要软件流程
3. API 参考
 - 3.1 ISP_FEC_CONFIG_INIT
 - 3.2 rk_ispfec_api_init
 - 3.3 rk_ispfec_api_prepare
 - 3.4 rk_ispfec_api_process
 - 3.5 rk_ispfec_api_deinit
 - 3.6 rk_ispfec_api_calFecMeshsize
 - 3.7 rk_ispfec_api_set_cfg
 - 3.8 rk_ispfec_api_get_cfg
 - 3.9 rk_ispfec_api_lock_process
 - 3.10 rk_ispfec_api_unlock_process
 - 3.11 rk_ispfec_api_set_mesh_config
 - 3.12 rk_ispfec_api_get_mesh_config
 - 3.13 rk_ispfec_api_load_online_config_from_ini
4. 数据类型
 - 4.1 rk_ispfec_cfg_t
 - 4.2 in_fourcc
 - 4.3 out_fourcc
 - 4.4 mesh_info_t
 - 4.5 rk_ispfec_update_mesh_mode
 - 4.6 gen_mesh_online_info_t
 - 4.7 rk_ispfec_correct_direction
 - 4.8 rk_ispfec_correct_style
 - 4.9 rk_ispfec_mesh_config
 - 4.10 rk_ispfec_buffer_info
5. Proc调试信息说明
 - 5.1 概述
 - 5.2 FEC
6. FAQ
 - 6.1 如何配置内核注册 FEC 设备驱动
 - 6.1.1 RK3588
 - 6.1.2 RV1126B
 - 6.2 是否有集成 ispfec 的参考用例
 - 6.3 性能优化建议
 - 6.3.1 RK3588

- 6.3.2 RV1126B
- 6.4 如何动态调整 FEC 工作频率
 - 6.4.1 RK3588
 - 6.4.2 RV1126B
- 6.5 如何使用双 FEC 硬件，以提高处理速度
- 6.6 如何动态开启 FEC 驱动层的日志
 - 6.6.1 RK3588
 - 6.6.2 RV1126B
- 6.7 如何使用 openCV/matlab 生成 FEC 硬件支持的矫正表（网格）
- 6.8 如何开启 libispfec 库的调试日志
 - 6.8.1 RV1126B
- 7. 附录
 - 7.1 生成表（网格）方法说明
 - 7.1.1 生成表的步骤
 - 7.1.2 RK 标定工具和 OpenCV/matlab 标定工具的区别
 - 7.1.3 OpenCV/Matlab 生成定点化表参考用例
 - 7.1.3.1 RK3588

1. 概述

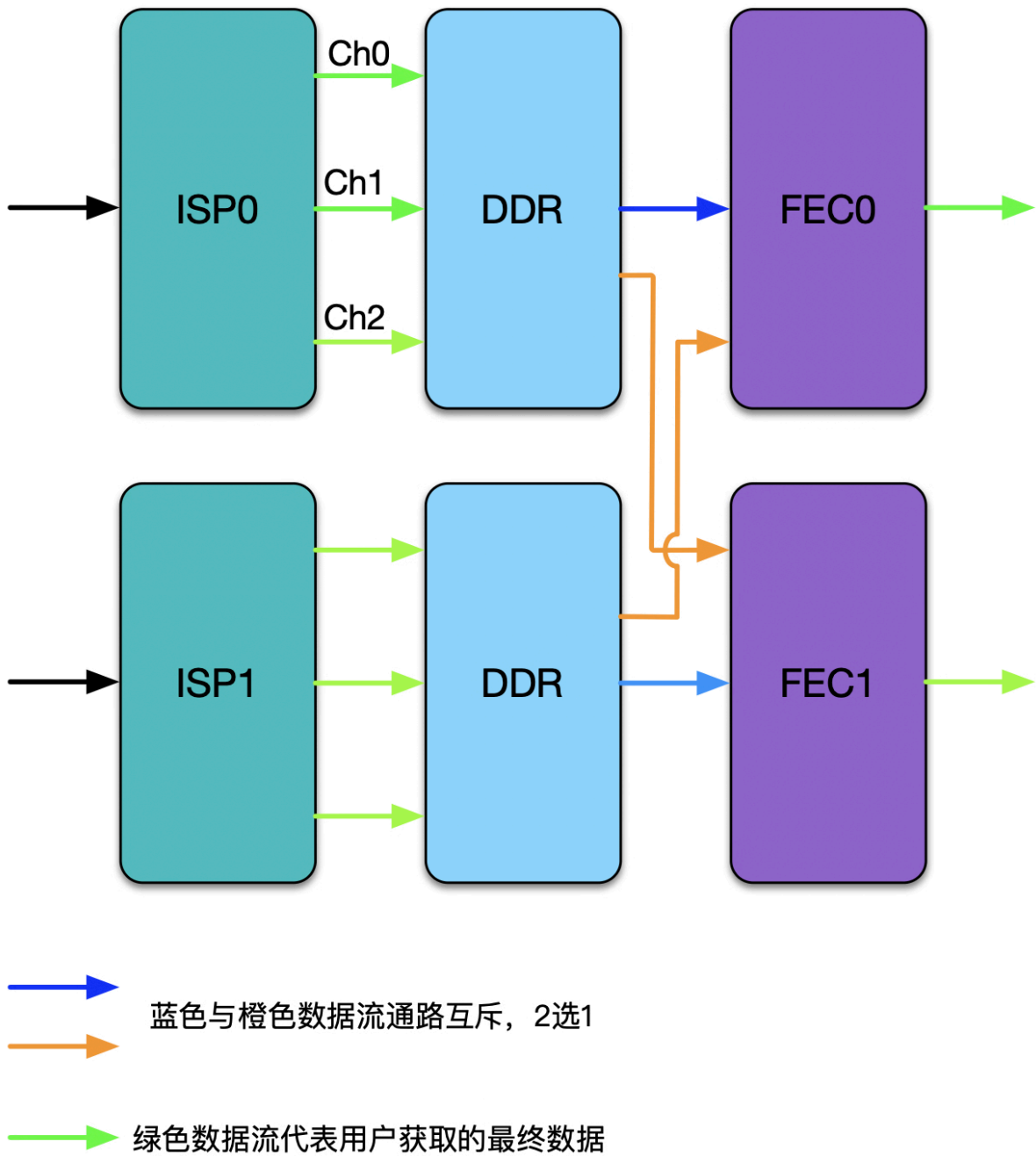
鱼眼畸变矫正（Fisheye Distortion Correction，以下简称FEC）实现校正由鱼眼镜头造成的图像畸变，使图像看起来更接近人眼自然视角的过程。鱼眼镜头因其超宽视角，通常会引入显著的桶形畸变，导致图像边缘被拉伸或压缩。

1.1 FEC 框图

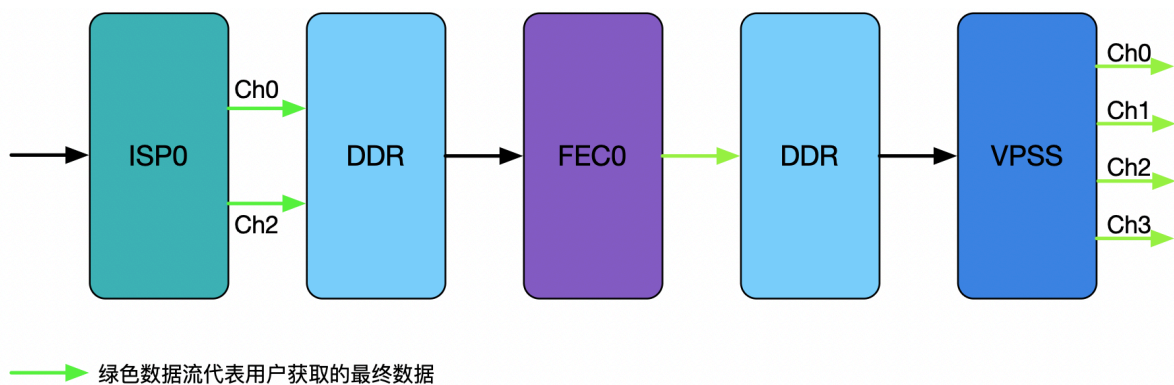
表1-1 FEC_IP_NUM 和芯片对应关系

芯片	FEC_IP_NUM
RK3588	2
RV1126B	1

1.1.1 图1-1 RK3588 FEC 典型框图



1.1.2 图1-2 RV1126B FEC 典型框图



1.2 性能规格及约束概述

1.2.1 图像数据存储方案概述

表1-1 图像数据存储概述

内存访问模式	描述
raster	逐像素、逐行连续地存储图像数据的最基本格式。
Tile4x4	将图像划分成多个 4x4 的 tile（块），每块内的像素按行或特定顺序连续排列，然后按 tile 顺序写入内存。
FBC64x4	将图像以 64x4 大小的 tile 为单位，对图像进行无损压缩，节省带宽和内存。

1.2.2 RK3588 性能规格及约束

1.2.2.1 规格约束

表1-2 输入输出格式

规格点	内存访问模式	数据格式
输入图像数据格式	raster	YUV422SP, YUV422I, YUV420SP
输出图像数据格式	raster	YUV422SP, YUV422I, YUV420SP
输出图像数据格式	FBC64x4	YUV422SP, YUV420SP

表1-3 输入输出分辨率

规格点	描述
输入图像分辨率	- 最小支持：128x128。 - 最大支持：8188x8188。 - 宽度和高度均为 4 像素对齐，stride 16 对齐。
输出图像分辨率	- 最小支持：128x128。 - 最大支持：8188x8188。 - 宽度和高度均为 4 像素对齐，stride 16 对齐。

1.2.2.2 性能

输入内存访问模式	输出内存访问模式	数据格式	性能
raster	raster	YUV422SP, YUV422I, YUV420SP	4k@30fps
raster	FBC64x4	YUV422SP, YUV422I, YUV420SP	4k@30fps

1.2.3 RV1126B 性能规格及约束

1.2.3.1 规格约束

表1-4 输入输出格式

规格点	内存访问模式	数据格式
输入图像数据格式	raster	YUV420SP
输入图像数据格式	tile4x4	YUV420SP
输出图像数据格式	raster	YUV420SP
输出图像数据格式	tile4x4	YUV420SP
输出图像数据格式	FBC64x4	YUV420SP
输出图像数据格式	QUAD (TO AVSP)	YUV420SP

表1-5 输入输出分辨率

规格点	描述
输入图像分辨率	- 最小支持：64x64。 - 最大支持：8160x8160。 - 宽度和高度均为 4 像素对齐，stride 16 对齐。
输出图像分辨率	- 最小支持：64x64。 - 最大支持：8160x8160。 - 宽度和高度均为 4 像素对齐，stride 16 对齐。

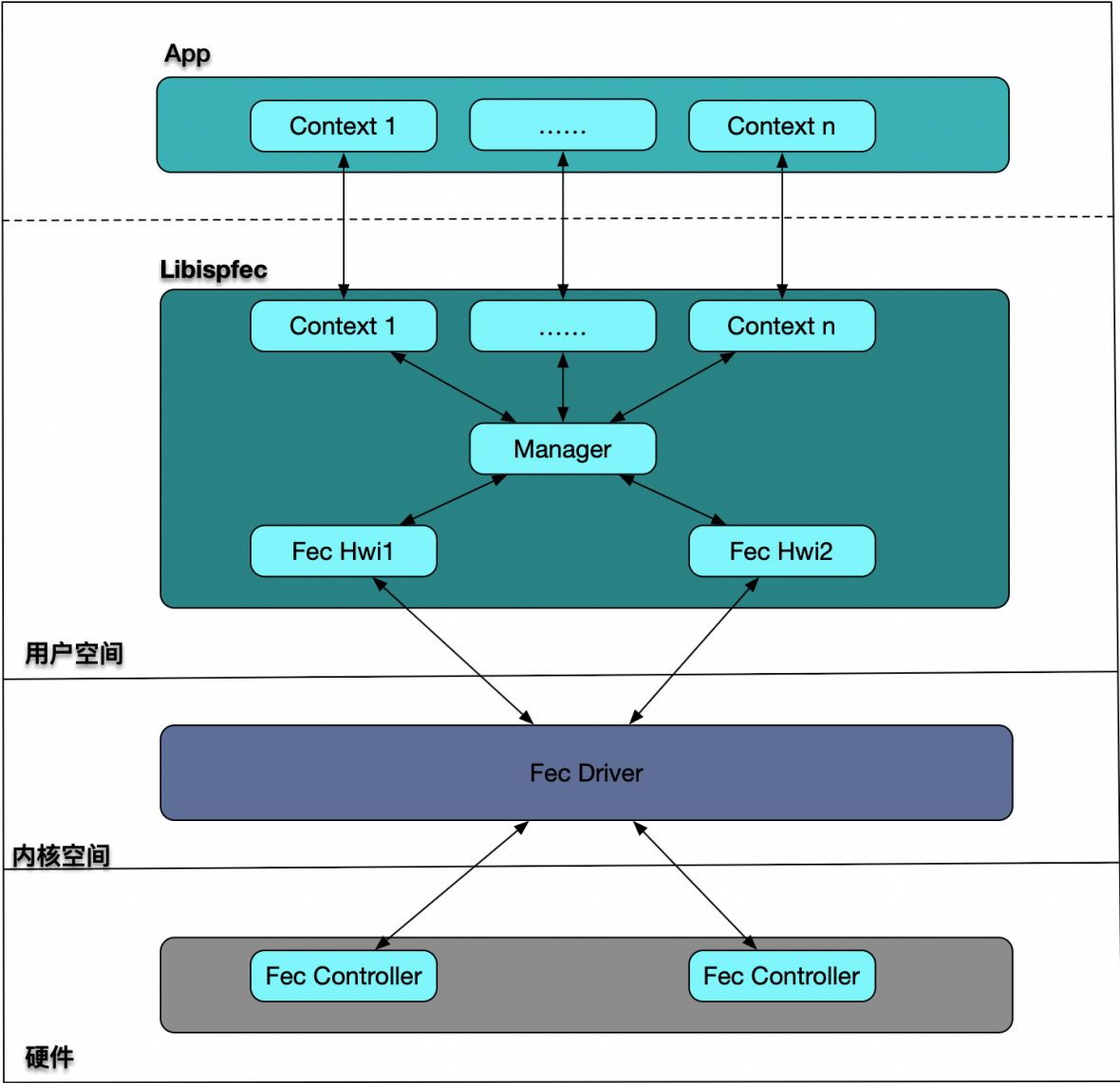
1.2.3.2 性能

输入内存访问模式	输出内存访问模式	数据格式	性能
raster	raster	YUV420SP	4k@30fps (性能下降多, 不推荐使用)
tile4x4	tile4x4	YUV420SP	4k@30fps
tile4x4	FBC64x4	YUV420SP	4k@30fps
tile4x4	QUAD (TO AVSP)	YUV420SP	4k@30fps

2. 软件开发

2.1 软件架构

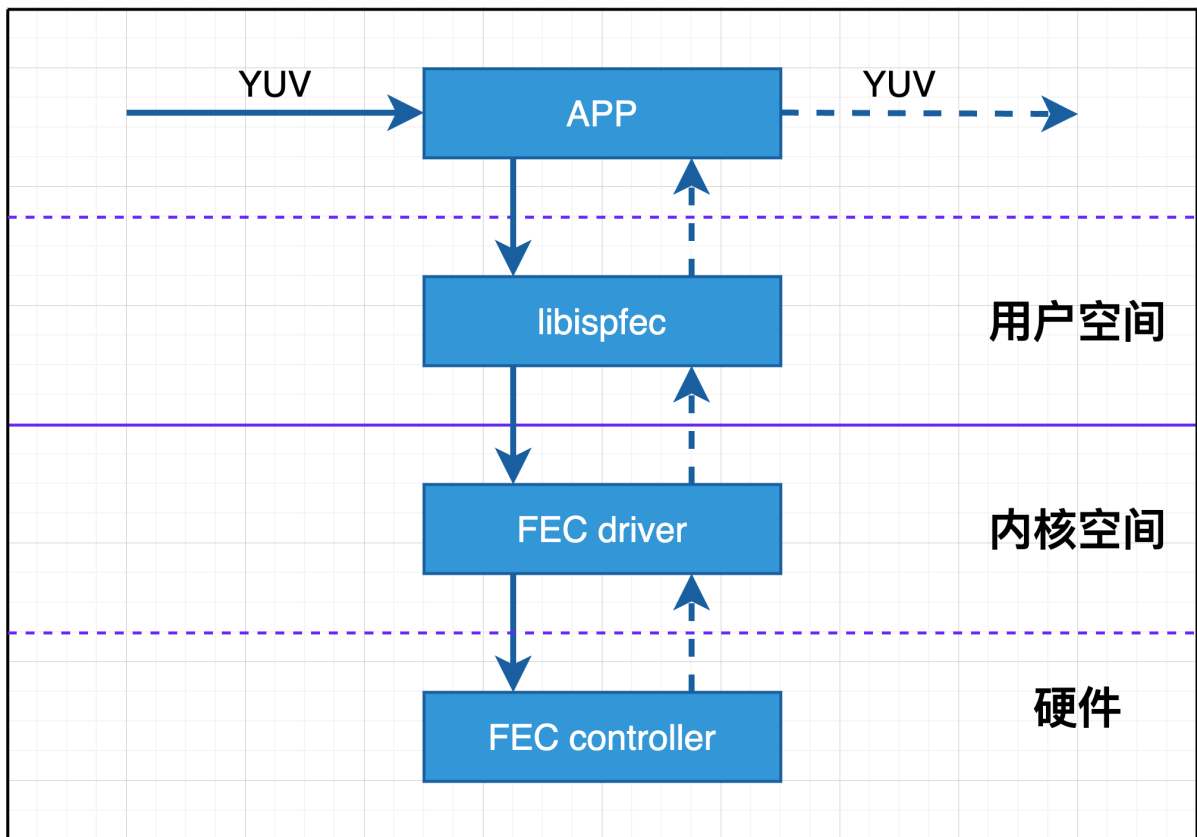
图2-1 libispfec 软件架构图



2.2 软件流程

2.2.1 数据流程框图

图2-2 libispfec 数据流程框图



2.2.2 主要软件流程

- 应用调用 [rk_ispfec_api_init](#) 初始化 libispfec 实例。
- 应用调用 [rk_ispfec_api_process](#) 将图像送入 libispfec，接口阻塞至一帧图像处理结束。
- libispfec 内部通过 ioctl 接口将输入图像送入 FEC 驱动，驱动配置启动 FEC 硬件处理。
- FEC 硬件处理完成，libispfec 内部 ioctl 接口返回。
- 应用 [rk_ispfec_api_process](#) 接口返回，获取处理完的输出图像。
- 应用调用 [rk_ispfec_api_deinit](#) 反初始化 libispfec 实例。

3. API 参考

- [ISP FEC CONFIG INIT](#): 初始化实例配置的宏。
- [rk_ispfec_api_init](#): 初始化 ispfec 实例。
- [rk_ispfec_api_prepare](#): 预处理 ispfec 实例。
- [rk_ispfec_api_process](#): 对一帧图像进行畸变矫正处理。
- [rk_ispfec_api_deinit](#): 反初始化 ispfec 实例。
- [rk_ispfec_api_calFecMeshsize](#): 计算 mesh buffer 所需的大小。
- [rk_ispfec_api_set_cfg](#): 配置当前实例的参数。
- [rk_ispfec_api_get_cfg](#): 获取当前实例的配置参数。
- [rk_ispfec_api_lock_process](#): 锁住 ispfec 内部的处理流程，以同步的方式更新畸变矫正表内容。
- [rk_ispfec_api_unlock_process](#): 解锁 ispfec 内部的处理流程。

- [rk_ispfec_api_set_mesh_config](#): 配置畸变矫正表更新模式及各模式下相关参数。
- [rk_ispfec_api_get_mesh_config](#): 获取畸变矫正表更新模式及各模式下相关参数。
- [rk_ispfec_api_load_online_config_from_ini](#): 根据 ini 文件，解析出镜头畸变相关系数。

3.1 ISP_FEC_CONFIG_INIT

【描述】

初始化实例配置的宏。

【语法】

```
#define ISP_FEC_CONFIG_INIT(cfg)
```

【参数】

参数名称	描述	输入/输出
cfg	ispfec 实例的初始化配置参数指针。	输入

【返回值】

无

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- 该宏会将 cfg 清0，然后自动对结构体内部 version 和 size 赋值。

【举例】

- 应用调用 [ISP_FEC_CONFIG_INIT](#) 对 cfg 进行初始化。
- 对 cfg 其他参数进行赋值。
- 应用调用 [rk_ispfec_api_init](#) 做实例初始化。

【相关主题】

[rk_ispfec_cfg_t](#)

3.2 rk_ispfec_api_init

【描述】

初始化 ispfec 实例。

【语法】

```
rk_ispfec_ctx_t* rk_ispfec_api_init(rk_ispfec_cfg_t* cfg);
```

【参数】

参数名称	描述	输入/输出
cfg	ispfec 实例的初始化配置参数指针。	输入

【返回值】

返回值	描述
非 NULL	成功。
NULL	失败。

【需求】

- 头文件：rk_ispfec_api.h
- 库文件：libispfec.so 或者 libispfec.a

【注意】

- 可一次初始化多个实例，但必须判断 [rk_ispfec_api_init](#) 函数返回成功后才能使用实例。
- 返回值不能为空指针或非法指针。

【相关主题】

[rk_ispfec_cfg_t](#)

3.3 rk_ispfec_api_prepare

【描述】

预处理 ispfec 实例。

【语法】

```
int rk_ispfec_api_prepare(rk_ispfec_ctx_t* ctx, rk_ispfec_cfg_t* cfg);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
cfg	ispfec 配置参数指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- 该接口主要用来 [rk_ispfec_api_init](#) 之后, 修改和重配置 ispfec 实例参数。
- 如果 cfg 和 [rk_ispfec_api_init](#) 配置的参数一致, 可不调用该接口。

【相关主题】

[rk_ispfec_cfg_t](#)

3.4 rk_ispfec_api_process

【描述】

对一帧图像进行畸变矫正处理。

【语法】

```
int rk_ispfec_api_process(rk_ispfec_ctx_t* ctx, int src_fd, int dst_fd);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
src_fd	输入图像 buffer 句柄。	输入
dst_fd	输出图像 buffer 句柄。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- src_fd 和 dst_fd 必须是对应输入和输出图像的 buffer 句柄。

3.5 rk_ispfec_api_deinit

【描述】

反初始化 ispfec 实例。

【语法】

```
void rk_ispfec_api_deinit(rk_ispfec_ctx_t* ctx);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入

【返回值】

无

3.6 rk_ispfec_api_calFecMeshsize

【描述】

计算畸变矫正表的大小，单位为点数。

【语法】

```
int rk_ispfec_api_calFecMeshsize(int width, int height);
```

【参数】

参数名称	描述	输入/输出
width	图像宽度	输入
height	图像高度	输出

【返回值】

对应分辨率使用的矫正表大小。

【注意】

- 该接口计算的是表的点数大小： $lut_x * lut_y$ 。如果要计算分配表 buffer 大小要转成字节数。

芯片平台	矫正表	表 buffer 大小 (Bytes)
RK3588		
	mesh_xint	Meshsize * 2
	mesh_xfra	Meshsize
	mesh_yint	Meshsize * 2
	mesh_yfra	Meshsize
RV1126B		
	mesh_info	Meshsize * 6

- 示例代码请参考 sample_fec.c 文件中的 init_ispfec_bufs 函数实现。

3.7 rk_ispfec_api_set_cfg

【描述】

配置当前实例的参数。

【语法】

```
int rk_ispfec_api_set_cfg(rk_ispfec_ctx_t* ctx, rk_ispfec_cfg_t* cfg);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
cfg	ispfec 配置参数指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【相关主题】

[rk_ispfec_cfg_t](#)

3.8 rk_ispfec_api_get_cfg

【描述】

获取当前实例的配置参数。

【语法】

```
int rk_ispfec_api_get_cfg(rk_ispfec_ctx_t* ctx, rk_ispfec_cfg_t* cfg);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
cfg	ispfec 配置参数指针。	输出

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【相关主题】

[rk_ispfec_cfg_t](#)

3.9 rk_ispfec_api_lock_process

【描述】

锁住 ispfec 内部的处理流程，以同步的方式更新畸变矫正表内容。

【语法】

```
int rk_ispfec_api_lock_process(rk_ispfec_ctx_t* ctx);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- 该函数必须与 [rk_ispfec_api_unlock_process](#) 成对使用, 否则将死锁。
- 调用 [rk_ispfec_api_unlock_process](#) 前, 不能再调用其他 api 函数, 否则将死锁。

【举例】

主要使用场景如下:

- 应用调用 [rk_ispfec_api_lock_process](#) 接口, 获得锁。
- 应用更新 `cfg->mesh_info.vir_addr` 的数据。
- 更新完毕, 调用 [rk_ispfec_api_unlock_process](#), 释放锁。

【相关主题】

[rk_ispfec_api_unlock_process](#)

3.10 rk_ispfec_api_unlock_process

【描述】

解锁 ispfec 内部的处理流程。

【语法】

```
int rk_ispfec_api_unlock_process(rk_ispfec_ctx_t* ctx);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- 该函数必须与 [rk_ispfec_api_lock_process](#) 成对使用。

【相关主题】

[rk_ispfec_api_lock_process](#)

3.11 rk_ispfec_api_set_mesh_config

【描述】

配置畸变矫正表更新模式及各模式下相关参数。

【语法】

```
int rk_ispfec_api_set_mesh_config(rk_ispfec_ctx_t* ctx, rk_ispfec_mesh_config_t* config);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
config	mesh 更新模式及各模式下相关参数。	输入

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【注意】

- mode 配置为 RK_ISPFEC_UPDATE_MESH_ONLINE, [rk_ispfec_api_init](#) 必须也是 RK_ISPFEC_UPDATE_MESH_ONLINE 模式, 否则接口返回错误。
- 其他模式有切换的话, 必须赋值模式下的完整参数。

【举例】

- 示例代码请参考 sample_fec.c 文件中的 test_api_set_mesh_* 函数实现。

【相关主题】

3.12 rk_ispfec_api_get_mesh_config

【描述】

获取畸变矫正表更新模式及各模式下相关参数。

【语法】

```
int rk_ispfec_api_get_mesh_config(rk_ispfec_ctx_t* ctx, rk_ispfec_mesh_config_t* config);
```

【参数】

参数名称	描述	输入/输出
ctx	ispfec 实例指针。	输入
config	mesh 更新模式及各模式下相关参数。	输出

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件: rk_ispfec_api.h
- 库文件: libispfec.so 或者 libispfec.a

【相关主题】

3.13 rk_ispfec_api_load_online_config_from_ini

【描述】

根据 ini 文件，解析出镜头畸变相关系数。

【语法】

```
int rk_ispfec_api_load_online_config_from_ini(const char* ini_path, gen_mesh_online_info_t* info_out)
```

【参数】

参数名称	描述	输入/输出
ini_path	存放畸变标定结果的 ini 文件路径	输入
info_out	根据 ini 文件解析出的镜头畸变相关系数	输出

【返回值】

返回值	描述
0	成功。
非0	失败。

【需求】

- 头文件：rk_ispfec_api.h
- 库文件：libispfec.so 或者 libispfec.a

【相关主题】

[gen_mesh_online_info_t](#)

4. 数据类型

ispfec 相关数据类型对应如下：

- [rk_ispfec_cfg_t](#)：ispfec 实例参数配置。
- [in_fourcc](#)：输入像素格式（V4L2 fourcc code）。
- [out_fourcc](#)：输出像素格式（V4L2 fourcc code）。
- [mesh_info_t](#)：定义存储网格数据的参数配置。
- [rk_ispfec_update_mesh_mode](#)：定义 ispfec 网格配置模式。
- [gen_mesh_online_info_t](#)：定义 ispfec 内部自动生成网格参数配置。
- [rk_ispfec_mesh_config](#)：定义更新畸变矫正表的模式及各模式下的相关参数。
- [rk_ispfec_buffer_info](#)：通过 buffer 方式更新畸变矫正表的结构体参数。
- [rk_ispfec_correct_direction](#)：定义内部生成网格的方式，畸变矫正方向配置。
- [rk_ispfec_correct_style](#)：定义内部生成网格的方式，畸变矫正风格配置。

4.1 rk_ispfec_cfg_t

【说明】

定义 ispfec 实例配置参数。

【定义】

```
typedef struct rk_ispfec_cfg_s {
```

```

// v2.0.0 新增
int version;
int size;

int in_width;
int in_height;
int out_width;
int out_height;

int in_fourcc;
int out_fourcc;

#ifdef RKFEC_HW_V20
int border_mode;
int cross_buf_mode;
rk_ispfec_bg_val bg_val;

mesh_info_t mesh_info;
#else
mesh_info_t mesh_xint;
mesh_info_t mesh_xfra;
mesh_info_t mesh_yint;
mesh_info_t mesh_yfra;
#endif

enum rk_ispfec_update_mesh_mode mesh_upd_mode;
union {
char mesh_file_path[128];
gen_mesh_online_info_t mesh_online;
} u;

// v3.0.0 新增
int in_offset_x;
int out_offset_x;
int in_stride;
int out_stride;

// v3.1.0 新增
char* calib_ini_from;
} rk_ispfec_cfg_t;

```

【成员】

成员名称	描述
version	当前的 API 版本号，固定配置为 <code>RK_ISP_FEC_API_VERSION</code> ，建议使用宏 ISP_FEC_CONFIG_INIT 进行快捷初始化。 rk_ispfec_api_init 会使用该字段进行版本校验。 版本：v2.0.0 新增
size	当前的 API 版本号，固定配置为 <code>sizeof(rk_ispfec_cfg_t)</code> ，建议使用宏 ISP_FEC_CONFIG_INIT 进行快捷初始化。 rk_ispfec_api_init 会使用该字段进行版本校验。 版本：v2.0.0 新增
in_width	输入图像宽度。
in_height	输入图像高度。
out_width	输出图像宽度。
out_height	输出图像高度。
in_fourcc	输入图像数据格式。详见 in_fourcc 。
out_fourcc	输出图像数据格式。详见 out_fourcc 。
border_mode	当矫正表内的像素坐标范围超出输入图的边界，即超出 $[(0,0) \sim (SrcW - 1, srcH - 10)]$ 时，会触发 FEC 的填充功能。 border_mode=0 ：填充 <code>bg_val</code> 。 border_mode=1 ：填充输入图边界像素。
cross_buf_mode	由于 FEC 对图像的局部缩小有倍数限制（大概为3.75倍），当mesh表对应的图像内容缩小超过这个限制时，会触发 FEC 的填充功能。 cross_buf_mode=0 ：填充 <code>bg_val</code> 。 cross_buf_mode=1 ：填充 FEC 当前取数块的的边界值。
bg_val	y、u、v值。常用的为黑色，即y=0, u=128, v=128。
mesh_info	存储畸变矫正表 buffer 信息。仅 RV1126B 适用。
mesh_xint	网格 x 方向整数部分参数属性。仅 RK3588 适用。
mesh_xfra	网格 x 方向浮点部分参数属性。仅 RK3588 适用。
mesh_yint	网格 y 方向整数部分参数属性。仅 RK3588 适用。
mesh_yfra	网格 y 方向浮点部分参数属性。仅 RK3588 适用。
mesh_upd_mode	网格更新模式，详见 rk_ispfec_update_mesh_mode 。
mesh_file_path	外部生成网格文件，存放网格文件的路径。
mesh_online	库内部生成网格所需的标定参数及配置。详见 gen_mesh_online_info_t 。
in_offset_x	输入图像水平方向位移。 单位：像素。 范围： [0 ~ (in_width - 1)] 版本：v3.0.0 新增

成员名称	描述
out_offset_x	输出图像水平方向位移。 单位：像素。 范围：【 0 ~ (out_width - 1)】 版本：v3.0.0 新增
in_stride	输入图像的虚宽。等于 0 时，库自动赋值为 in_width。 版本：v3.0.0 新增
out_stride	输出图像的虚宽。等于 0 时，库自动赋值为 out_width。 版本：v3.0.0 新增
calib_ini_from	保存的镜头标定的畸变系数 ini 文件路径。mesh_upd_mode 为 RK_ISPFEC_UPDATE_MESH_ONLINE_FROM_INI 时使用。 non-NULL：库解析 ini 文件，并覆盖 mesh_online 相关参数。 版本：v3.1.0 新增

【注意事项】

mesh_upd_mode 模式相关配置	应用配置方法
RK_ISPFEC_UPDATE_MESH_ONLINE	配置 mesh_online 参数。
RK_ISPFEC_UPDATE_MESH_FROM_FILE	配置 mesh_file_path 和 mesh_info 内的文件路径信息。
RK_ISPFEC_UPDATE_MESH_FROM_BUFFER	配置表数据至 mesh_info 内的 buffer。
RK_ISPFEC_UPDATE_MESH_ONLINE_FROM_INI	配置存储畸变相关系数的 ini 文件路径给 calib_ini_from，libispfec 内部解析参数使用。

【相关主题】

[6.1 生成表（网格）方法说明。](#)

4.2 in_fourcc

【说明】

输入像素格式（V4L2 fourcc code）。

【定义】

RV1126B 支持以下输入格式

- V4L2_PIX_FMT_NV12: raster, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_NV21: raster, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_TILE420: TILE4x4, YUV420 semi-planar (8-bit)

4.3 out_fourcc

【说明】

输出像素格式 (V4L2 fourcc code)。

【定义】

RV1126B 支持以下输出格式

- V4L2_PIX_FMT_NV12: raster, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_NV21: raster, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_TILE420: TILE4x4, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_FBC0: FBC64x4, YUV420 semi-planar (8-bit)
- V4L2_PIX_FMT_QUAD: QUAD, YUV420 semi-planar (8-bit)

4.4 mesh_info_t

【说明】

定义存储网格数据的参数配置。

【定义】

```
typedef struct mesh_info_s {  
    int dmaFd;  
    void* vir_addr;  
    int size;  
    char mesh_file[64];  
} mesh_info_t;
```

【成员】

成员名称	描述
dmaFd	存储网格数据的 buffer 句柄。
vir_addr	存储网格数据的 buffer 虚拟地址。
size	网格大小。
mesh_file	外部生成网格的方式下, 存储网格数据的文件名。

4.5 rk_ispfec_update_mesh_mode

【说明】

定义 ispfec 网格配置模式。

【定义】

```

enum rk_ispfec_update_mesh_mode {
    RK_ISPFEC_UPDATE_MESH_ONLINE = 0,           // generate lut online
    RK_ISPFEC_UPDATE_MESH_FROM_FILE,          // external file import mesh
    RK_ISPFEC_UPDATE_MESH_FROM_BUFFER,        // update mesh from buffer
    descriptors
        RK_ISPFEC_UPDATE_MESH_ONLINE_FROM_INI //Load lens parameters from INI
file
};

```

【成员】

成员名称	描述
RK_ISPFEC_UPDATE_MESH_ONLINE	标定生成镜头畸变系数，应用传入标定系数，库生成表。
RK_ISPFEC_UPDATE_MESH_FROM_FILE	标定生成矫正表文件，将文件路径传入，库应用矫正表。
RK_ISPFEC_UPDATE_MESH_FROM_BUFFER	应用生成表，通过 buffer 的方式传入。
RK_ISPFEC_UPDATE_MESH_ONLINE_FROM_INI	应用将存储畸变相关系数的 ini 文件路径传入，libispfec 内部解析参数使用。

4.6 gen_mesh_online_info_t

【说明】

定义 ispfec 内部自动生成网格参数配置。

【定义】

```

typedef struct gen_mesh_online_info_s {
    double light_center[2];
    double coeff[4];
    enum rk_ispfec_correct_direction    direction;
    enum rk_ispfec_correct_style        style;

    int    correct_level;

    // v3.1.0 新增
    int calib_width;
    int calib_height;
    int calib_level_maxLimit;
} gen_mesh_online_info_t;

```

【成员】

成员名称	描述
light_center	镜头光心位置，标定参数。
coeff	镜头畸变系数，标定参数。
direction	配置矫正方向的类型。
style	配置矫正风格。
correct_level	配置矫正强度。
calib_width	标定图的宽度。 版本：v3.1.0 新增
calib_height	标定图的高度。 版本：v3.1.0 新增
calib_level_maxLimit	标定出支持的最大强度。 版本：v3.1.0 新增

【注意事项】

- light_center 和 coeff 需要使用 RK IQ tool 工具标定出镜头畸变相关系数并填入。

4.7 rk_ispfec_correct_direction

【说明】

定义内部生成网格的方式，畸变矫正方向配置。

【定义】

```
enum rk_ispfec_correct_direction {
    RK_ISPFEC_CORRECT_DIRECTION_X = 0x1,
    RK_ISPFEC_CORRECT_DIRECTION_Y,
    RK_ISPFEC_CORRECT_DIRECTION_XY
};
```

【成员】

成员名称	描述
RK_ISPFEC_CORRECT_DIRECTION_X	只矫正水平方向的畸变。
RK_ISPFEC_CORRECT_DIRECTION_Y	只矫正垂直方式的畸变。
RK_ISPFEC_CORRECT_DIRECTION_XY	同时矫正水平和垂直方向的畸变，默认方式。

4.8 rk_ispfec_correct_style

【说明】

定义内部生成网格的方式，畸变矫正风格配置。

【定义】

```
enum rk_ispfec_correct_style {  
    RK_ISPFEC_KEEP_ASPECT_RATIO_REDUCE_FOV = 0x1,  
    RK_ISPFEC_COMPRES_IMAGE_KEEP_FOV,  
};
```

【成员】

成员名称	描述
RK_ISPFEC_KEEP_ASPECT_RATIO_REDUCE_FOV	损失图像视场角，图像保持比例。
RK_ISPFEC_COMPRES_IMAGE_KEEP_FOV	保留较多视场角，图像中心会有压缩。

4.9 rk_ispfec_mesh_config

【说明】

定义更新畸变矫正表的模式及各模式下的相关参数。

【定义】

```
typedef struct rk_ispfec_mesh_config {  
    enum rk_ispfec_update_mesh_mode mode;  
  
    union {  
        // File mode configuration (RK_ISPFEC_UPDATE_MESH_FROM_FILE)  
        struct {  
#ifdef RKFEC_HW_V20  
            char mesh_path[256]; // Single mesh file path (V20)  
#else  
            char mesh_xint[256]; // X-axis integer mesh file  
            char mesh_xfra[256]; // X-axis fractional mesh file  
            char mesh_yint[256]; // Y-axis integer mesh file  
            char mesh_yfra[256]; // Y-axis fractional mesh file  
#endif  
        } file_cfg;  
  
        // Buffer mode configuration (RK_ISPFEC_UPDATE_MESH_FROM_BUFFER)  
        struct {  
#ifdef RKFEC_HW_V20  
            rk_ispfec_buffer_info_t mesh; // Single buffer information  
#else  
            struct {  
                rk_ispfec_buffer_info_t xint; // X-axis integer part  
                rk_ispfec_buffer_info_t xfra; // X-axis fractional part
```

```

        rk_ispfec_buffer_info_t yint; // Y-axis integer part
        rk_ispfec_buffer_info_t yfra; // Y-axis fractional part
    } mesh; // Multiple buffers configuration
#endif
} buffer_cfg;

// Online generation mode (RK_ISPFEC_UPDATE_MESH_ONLINE)
gen_mesh_online_info_t online_cfg; // Online generation parameters
} params;
} rk_ispfec_mesh_config_t;

```

【成员】

成员名称	描述
mode	畸变矫正表更新模式。详见 rk_ispfec_update_mesh_mode 。
file_cfg	应用生成矫正表文件，指定矫正表文件的路径。
buffer_cfg	应用生成矫正表，通过 buffer 的方式传入的方式，buffer 信息的配置。详见 rk_ispfec_buffer_info 。
online_cfg	ispfec 内部根据传入的畸变系数生成网格，畸变矫正系数相关的参数。详见 gen_mesh_online_info_t 。

【注意事项】

- 配置模式必须与 [rk_ispfec_api_init](#) 时的初始化模式一致。

4.10 rk_ispfec_buffer_info

【说明】

通过 buffer 方式更新畸变矫正表的结构体参数。

【定义】

```

typedef struct rk_ispfec_buffer_info {
    int dma_fd; //< DMA buffer handle (effective when need_copy=0)
    void* vir_addr; //< Source virtual address for copy (required when
need_copy=1)
    size_t size; //< Buffer capacity, must be >= actual data size
    bool need_copy; //< Memory copy mode flag. Configuration rules:
//< - DMA handle mode: set 0 to:
//< a. Directly use dma_fd
//< b. vir_addr will be ignored
//< - Memory copy mode: set 1 to:
//< a. Copy size bytes from vir_addr to the in-use DMA
buffer
//< b. dma_fd will be ignored
} rk_ispfec_buffer_info_t;

```

【成员】

成员名称	描述
dma_fd	dma buffer 句柄 (need_copy=0 时, 才会被替换更新)。
vir_addr	拷贝该虚拟地址的数据到内部矫正表缓冲区 (need_copy=1 时, 才会被拷贝更新)。
size	矫正表 buffer 的大小。
need_copy	0: 库会将矫正表 buffer 替换为 dma_fd 对应的 buffer。 应用要管理好该 dma buffer, 不能释放传入的 buffer。 1: 库会将 vir_addr 内容拷贝到当前在使用的内部矫正表 buffer 中。 更新完后 vir_addr 可以被释放, 输入的 dma_fd 将被忽略。

5. Proc调试信息说明

5.1 概述

调试信息采用了 Linux 下的 proc 文件系统, 可实时反映当前系统的运行状态, 所记录的信息可供问题定位及分析时使用。

【文件目录】

```
/proc/rkfec_offline
```

【信息查看方法】

- 在控制台上可以使用 `cat` 命令查看信息, 例如 `cat /proc/rkfec_offline`; 也可以使用其他常用的文件操作命令, 例如 `cp /proc/rkfec_offline .`, 将 `rkfec_offline` 文件拷贝到当前目录。
- 在应用程序中可以将上述文件当作普通只读文件进行读操作, 例如 `fopen`、`fread` 等。

【说明】

RK3588 不支持获取 `/proc/rkfec_offline` 调试信息。

5.2 FEC

【调试信息】

```
# cat /proc/rkfec_offline
rkfec_offline Version:v00.01.00
aclk_fec    500000000
hclk_fec    148500000
clk_fec     500000000
Interrupt   Cnt:16234 ErrCnt:0
Input       Format:NV12 Size:3840x2160 Offset(0) Sizeimage(12441600)
Fec offline (frame:16235 rate:33ms state:working time:10ms frameloss:0
frm_oversdtim_cnt:0)
```

```
Output      Format:NV12 Size:3840x2160 Offset(0) Sizeimage(12441600) (frame:16234
rate:33ms frameloss:0
CTRL       RD_fmt:semi RD_mode:rast WR_fmt:semi WR_mode:rast WR_fbce_unc:off
(0x104)
CORE_CTRL  Bic:precise Lut_density:32x16, Border_fill:bg, Cross_fill:bg
(0x80000001)
RD_VIR     Y:3840 C:3840
WR_VIR     Y:3840 C:3840
BG_VALUE   Y:0 U:0 V:0
LUT        Size: 16577
STATUS0    0x4
STATUS1    0x2a5555
Cacheline  128B
```

【调试信息分析】

记录当前 FEC 模块的使用情况。

【参数说明】

参数	子参数	描述
rkfec_offline Version		模块驱动版本号。
aclk_fec		模块的高速 AXI 总线时钟频率，用于数据传输（DMA）。
hclk_fec		模块的 AHB 时钟频率，用于寄存器访问。
clk_fec		模块核心部分的主时钟信号频率。
Interrupt		模块触发的中断计数。
Input		
	Format	输入图像像素格式。
	Size	输入图像分辨率，单位：像素。
	Offset	输入图像的像素偏移，单位：像素。
	Sizeimage	输入图像大小，单位：字节。
Fec offline		
	frame	当前处理图像的帧号。
	rate	当前输入图像的帧率。
	state	当前硬件工作状态。
	time	统计硬件处理一帧的时间，帧尾更新。
	frameloss	统计由于硬件处理异常，导致丢帧的帧数。
	frm_oversdtim_cnt	统计硬件处理时间超出 /sys/module/video_rkfec/parameters/standardfps 配置的 帧率处理时间的次数。
Output		
	Format	输出图像像素格式。
	Size	输出图像分辨率，单位：像素。
	Offset	输出图像的像素偏移，单位：像素。
	Sizeimage	输出图像大小，单位：字节。
	frame	输出图像的帧号。
	rate	输出图像的帧率。
	frameloss	输出图像的丢帧帧数。
CTRL		模块的控制寄存器，对应硬件寄存器（FEC_FEC_CTRL）各个 字段。
	RD_fmt	模块读取输入图像的数据格式。
	RD_mode	模块读取输入图像的内存访问模式。
	WR_fmt	模块输出图像的数据格式。

参数	子参数	描述
	WR_mode	模块输出图像的内存访问模式。
	WR_fbce_unc	调试使用。
CORE_CTRL		模块的核心控制寄存器，对应硬件寄存器（FEC_FEC_CORE_CTRL）各个字段。
	Bic	像素双立方插值的模式选择，不同模式对应不同的双立方参数。默认：precise。
	Lut_density	畸变矫正表的表密度。
	Border_fill	当矫正表内的像素坐标范围超出输入图的边界，即超出 [(0,0) ~ (SrcW - 1, srcH - 1)] 时，会触发 FEC 的填充功能。 bg : 填充 bg_val。 border : 填充输入图边界像素。
	Cross_fill	由于 FEC 对图像的局部缩小有倍数限制（大概为3.75倍），当 mesh表对应的图像内容缩小超过这个限制时，会触发 FEC 的填充功能。 bg : 填充 bg_val。 border : 填充 FEC 当前取数块的的边界值。
RD_VIR		输入图像虚宽，对应硬件寄存器（FEC_FEC_RD_VIR_STRIDE）。
WR_VIR		输出图像虚宽，对应硬件寄存器（FEC_FEC_WR_VIR_STRIDE）。
BG_VALUE		超限制填充的像素值。
LUT		矫正表大小，单位：字节。
STATUS0		模块的 DMA 状态寄存器。
STATUS1		模块的 DMA 状态寄存器。
Cacheline		缓存行的大小，单位：字节。

6. FAQ

6.1 如何配置内核注册 FEC 设备驱动

【解决】

6.1.1 RK3588

```
- 使能 config 配置项
CONFIG_VIDEO_ROCKCHIP_ISPP_FEC=y

- 使能 FEC DTS 节点配置
// FEC IP 0
&rkispp0 {
    status = "okay";
};

&fec0_mmu {
    status = "okay";
};

// FEC IP 1: RK3588 有两个 FEC IP。
&rkispp1 {
    status = "okay";
};

&fec1_mmu {
    status = "okay";
};
```

6.1.2 RV1126B

```
- 使能 config 配置项
CONFIG_VIDEO_ROCKCHIP_FEC=y

- 使能 FEC DTS 节点配置
&rkfec {
    status = "okay";
};

&rkfec_mmu {
    status = "okay";
};
```

【确认方法】

```
cat /sys/class/video4linux/video*/name | grep -i fec
```

6.2 是否有集成 ispfec 的参考用例

参考 IspFec/sample/sample_fec.c。

6.3 性能优化建议

【现象】

应用程序在超出FEC性能或者系统带宽紧张的情况下，如何优化处理性能。

【分析】

FEC 硬件处理速度和系统带宽关系较大，系统带宽紧张的情景下，FEC 硬件处理时间会增加，性能下降。

【解决】

6.3.1 RK3588

- 优化系统带宽。
- 参考[5.4](#)，提高 FEC 的工作频率。
- 参考[5.5](#)，使用双 FEC 硬件。
- 提高 FEC 模块的总线优先级（可能会影响其他模块）。

6.3.2 RV1126B

- 输入推荐使用 TILE 格式，输出推荐使用 FBCE/TILE 格式。NV12 格式性能下降多，不推荐使用。
- 尝试使用以下命令修改 cacheline

```
- 配置新值：  
echo 1 > /sys/module/video_rkfec/parameters/cache_linesize  
  
可配置值【1-2】  
1: 64B  
2: 128B(默认)  
  
- 查询当前配置值：  
cat /sys/module/video_rkfec/parameters/cache_linesize
```

- 优化系统带宽。
- 参考[5.4](#)，提高 FEC 的工作频率。
- 提高 FEC 模块的总线优先级（可能会影响其他模块）。

6.4 如何动态调整 FEC 工作频率

【解决】

6.4.1 RK3588

```
- mount debugfs
mount -t debugfs debugfs /sys/kernel/debug

- 查看当前工作频率
cat /sys/kernel/debug/clk/clk_summary | grep fish

- 根据需要修改工作频率
echo Y > /sys/module/video_rkispp/parameters/clk_dbg
echo 702000000 > /sys/kernel/debug/clk/clk_fisheye0_core/clk_rate
echo 702000000 > /sys/kernel/debug/clk/clk_fisheye1_core/clk_rate
```

6.4.2 RV1126B

```
- 查看当前工作频率
cat /sys/kernel/debug/clk/clk_summary | grep fec

- 根据需要修改工作频率
echo Y > /sys/module/video_rkfec/parameters/clk_dbg
echo 500000000 > /sys/kernel/debug/clk/aclk_fec/clk_rate
echo 500000000 > /sys/kernel/debug/clk/clk_core_fec/clk_rate
```

6.5 如何使用双 FEC 硬件，以提高处理速度

【解决】

初始化两个 ispfec 实例，并且创建双线程独立运行。

6.6 如何动态开启 FEC 驱动层的日志

【解决】

6.6.1 RK3588

```
echo 3 > /sys/module/video_rkispp/parameters/debug
echo 8 > /proc/sys/kernel/printk
```

6.6.2 RV1126B

```
echo 4 > /sys/module/video_rkfec/parameters/debug
echo 8 > /proc/sys/kernel/printk
```

6.7 如何使用 openCV/matlab 生成 FEC 硬件支持的矫正表（网格）

【解决】

参考[6.1 生成表（网格）方法说明](#)。

6.8 如何开启 libispfec 库的调试日志

【解决】

6.8.1 RV1126B

```
- 初始化 libispfec 之前执行以下命令：  
echo 4 > /sys/module/video_rkfec/parameters/user_debug
```

7. 附录

7.1 生成表（网格）方法说明

7.1.1 生成表的步骤

- 使用 RK 标定工具或者 openCV/matlab 标定工具标定出参数。
- 根据标定参数，使用 RK 标定算法接口或者 OpenCV/matlab 接口生成浮点表。
- 将浮点表转换成硬件加速（如 FEC 模块）处理所需的定点化表。

7.1.2 RK 标定工具和 OpenCV/matlab 标定工具的区别

名称	RK 标定工具	OpenCV/Matlab 标定工具
适用镜头类型	普通镜头、广角镜头、fov180度以内的鱼眼镜头	普通镜头、广角镜头、鱼眼镜头等
算法模型	多项式拟合模型	小孔成像模型、鱼眼模型、CMei模型等
支持的畸变类型	径向畸变(主要是桶形畸变)	径向畸变(桶形畸变、枕形畸变)、切向畸变等
适用场景	主要用于分级校正需求，可以调整不同校正强度，可以只校正水平方向、只校正垂直方向、两个方向都校正，可以配置是否保存水平最大视场角等功能。	主要用于通用校正需求，一般都是将畸变完全校正为直线，可以根据OpenCV/Matlab标定算法的特性进行输出图FOV的裁剪等，也可以根据特殊镜头类型实现定制化畸变校正需求。
ispfec 使用表方式	支持两种方式： - 通过标定工具直接生成离线的定点化表文件，ispfec 从文件中读取使用。 - 应用将标定参数通过 API 传入，ispfec 内部在线生成表使用。 两种方式矫正效果一致。	通过标定参数生成离线的定点化表并存入文件，ispfec 从文件中读取使用。

【注意】

- 两套标定工具的参数物理含义有差异，不能混用，调用的浮点表生成接口也不同。
- 标定时所用的棋盘格标定图分辨率必须与FEC的输入图分辨率严格一致，不能存在旋转、左右翻转、上下翻转、缩放等变换关系。
- 不支持变焦镜头，不同焦段的镜头需要分别进行标定，使用不同的定点化表。
- RV1109/RV1126等平台的FEC仅支持输入图和输出图分辨率一致。
- RK3588等平台的FEC可以支持输入输出分辨率不一致，支持通过FEC实现图像缩放功能。放大倍数无限制，但缩小倍数不能超过3.75倍。

7.1.3 OpenCV/Matlab 生成定点化表参考用例

7.1.3.1 RK3588

参考 `IspFec/sample/sample_generate_mesh.cpp`。