

SigmaStar

CALIB lib API User Guide

Version 1.1

© 2023 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

Revision No.	Description	Date
1.0	● Initial Release	04/20/2023
1.1	● Update MI_CALIB_Conf_t structure	10/18/2023

目录

REVISION HISTORY	i
目录ii	
1. 概述.....	1
1.1. 算法说明	1
2. API 参考	2
2.1. 功能模块 API	2
2.2. MI_CALIB_Init	2
2.3. MI_CALIB_Deinit	3
2.4. MI_CALIB_Calibration	4
2.5. MI_CALIB_FineTune	6
3. CALIB 数据类型.....	8
3.1. CALIB 模块相关数据类型定义	8
3.2. MI_CALIB_Conf_t	8
3.3. MI_CALIB_BufBlock_t	12
3.4. MI_CALIB_BufType_e.....	13
3.5. MI_CALIB_RetCode_e	14
3.6. MI_CALIB_Handle_t.....	15
3.7. MI_CALIB_PhyAddr_t.....	15
4. CALIB 配置文件.....	17
4.1. CALIB 模块相关配置文件	17
4.2. Calib_in.json.....	17
4.3. Calib_out.json	19
4.4. reprojection_err.txt	20
4.5. Finetune_in.json	21
4.6. Finetune_out.json	22
4.7. sysConfig.ini	23

1. 概述

1.1. 算法说明

CALIB API 包含标定，Auto Fine-Tune 相关模块。

2. API 参考

2.1.功能模块 API

API 名	功能
MI_CALIB_Init	初始化 CALIB lib 需要的内存。
MI_CALIB_Deinit	去初始化 CALIB lib 需要的内存。
MI_CALIB_Calibration	开启模型标定和产线标定。
MI_CALIB_FineTune	开启 Auto Finetune 功能。

2.2.MI_CALIB_Init

➤ 功能

初始化 CALIB lib 需要的内存。

➤ 语法

MI_CALIB_RetCode_e MI_CALIB_Init([MI_CALIB_Handle_t](#) *pAlgHandle, MI_CALIB_InitPara_t *pstInitPara);

➤ 形参

参数名称	描述	输入/输出
pAlgHandle	CALIB lib handle	输出
pstInitPara	CALIB lib 的初始化结构体指针，暂未使用。	输入

➤ 返回值

返回值	结果
E_MI_CALIB_SUCCESS	成功
E_MI_CALIB_INVALID_PARA	失败，无效参数
E_MI_CALIB_ERROR	失败，其他错误

➤ 依赖

- 头文件: mi_calib.h
- 库文件: Calib.lib

※ 注意

无。

➤ 举例

```
MI_CALIB_Handle_t algHandle = NULL;  
MI_CALIB_Init(&algHandle, NULL);
```

2.3.MI_CALIB_Deinit

➤ 功能

去初始化 CALIB lib 需要的内存。

➤ 语法

MI_CALIB_RetCode_e MI_CALIB_Deinit([MI_CALIB_Handle_t](#) algHandle);

➤ 形参

参数名称	描述	输入/输出
algHandle	CALIB lib handle	输入

➤ 返回值

返回值	结果
E_MI_CALIB_SUCCESS	成功
E_MI_CALIB_INVALID_PARA	失败，无效参数
E_MI_CALIB_ERROR	失败，其他错误

➤ 依赖

- 头文件: mi_calib.h
- 库文件: Calib.lib

※ 注意

无。

➤ 举例

无。

2.4.MI_CALIB_Calibration

➤ 功能

开启模型标定和产线标定。

➤ 语法

```
MI_CALIB_RetCode_e MI_CALIB_Calibration(MI_CALIB_Handle_t algHandle, MI_CALIB_Conf_t *pMI_Conf, MI_CALIB_BufBlock_t *pstInJsonBuf, MI_CALIB_BufBlock_t *pstOutJsonBuf, MI_CALIB_BufBlock_t *pcalibErrBuf);
```

➤ 形参

参数名称	描述	输入/输出
algHandle	CALIB lib handle	输入
pMI_Conf	CALIB lib 标定结构体指针	输出
pstInJsonBuf	输入的 json 配置，请参考 Calib_in.json 文件	输入
pstOutJsonBuf	输出的 json 配置，请参考 Calib_out.json 文件	输出
pcalibErrBuf	输出的标定误差，请参考 reprojection_err.txt 文件	输出

➤ 返回值

返回值	结果
E_MI_CALIB_SUCCESS	成功
E_MI_CALIB_INVALID_PARA	失败，无效参数
E_MI_CALIB_ERROR	失败，其他错误

➤ 依赖

- 头文件: [mi_calib.h](#)
- 库文件: [Calib.lib](#)

※ 注意

- 无。

➤ 举例

```
MI_CALIB_RetCode_e MI_CALIB_ReadJsFile(const char *jsFile, MI_CALIB_BufBlock_t* pstJsonBuf)
{
    struct stat fileInfo;
    FILE * fp = NULL;
    unsigned int readBytes = 0;

    int result = stat(jsFile, &fileInfo);
    if (result < 0)
    {
        printf("stat file error!\n");
    }

    pstJsonBuf->pBufAddr = (void *)MI_CALIB_Malloc(fileInfo.st_size);
    fp = fopen(jsFile, "r");

    readBytes = fread(pstJsonBuf->pBufAddr, 1, fileInfo.st_size, fp);
    printf("readBytes:%d\n", readBytes);
    printf("st_size:%d\n", fileInfo.st_size);
    if (readBytes != fileInfo.st_size)
    {
        printf("read json file failed!\n");
    }
    pstJsonBuf->u32BufSz = readBytes;

    return E_MI_CALIB_SUCCESS;
}

MI_CALIB_RetCode_e MI_CALIB_WriteJsFile(const char *jsFile, MI_CALIB_BufBlock_t *pJsonBuf)
{
    char *pBuf = (char *)pJsonBuf->pBufAddr;
    FILE *fp = fopen(jsFile, "w");
    fwrite(pBuf, 1, strlen(pBuf), fp);
    fclose(fp);

    return E_MI_CALIB_SUCCESS;
}

MI_CALIB_RetCode_e MI_CALIB_FreeBlockBuf(MI_CALIB_BufBlock_t* pstJsonBuf)
{
    if (pstJsonBuf->pBufAddr != NULL)
    {
        mem_free(pstJsonBuf->pBufAddr);
        pstJsonBuf->u32BufSz = 0;
    }

    return E_MI_CALIB_SUCCESS;
}

void Calib_DEMO(const char *JSpath, const char *calibOutPath, const char *projectionErrPath)
{
    MI_CALIB_Handle_t algHandle = NULL;
    MI_CALIB_BufBlock_t stInJsonBuf, stOutJsonBuf, calibErrBuf;
    MI_CALIB_Conf_t mi_calib_conf;

    MI_CALIB_Init(&algHandle, NULL);
    MI_CALIB_ReadJsFile(JSpath, &stInJsonBuf);
    MI_CALIB_Calibration(algHandle, &mi_calib_conf, &stInJsonBuf, &stOutJsonBuf, &calibErrBuf);
    MI_CALIB_Deinit(algHandle);
}
```

```
MI_CALIB_WriteJsFile(calibOutPath, &stOutJsonBuf);
MI_CALIB_WriteJsFile(projectionErrPath, &calibErrBuf);

MI_CALIB_FreeJsFile(&stInJsonBuf);
MI_CALIB_FreeJsFile(&stOutJsonBuf);
MI_CALIB_FreeJsFile(&calibErrBuf);
}

int main(int argc, char **argv)
{
    Calib_DEMO("Calib_in.json", "Calib_out.json", "reprojection_err.txt");
}
```

2.5.MI_CALIB_FineTune

➤ 功能

开启 Auto Finetune 功能。

➤ 语法

```
MI_CALIB_RetCode_e MI_CALIB_FineTune(MI\_CALIB\_Handle\_t algHandle,
                                       const char *confPath,
                                       MI\_CALIB\_BufBlock\_t *pstInJsonBuf,
                                       MI\_CALIB\_BufBlock\_t *pstCalibJsonBuf,
                                       int *stateflag);
```

➤ 形参

参数名称	描述	输入/输出
algHandle	CALIB lib handle	输入
confPath	Finetune 配置文件, 请参考 sysConfig.ini 文件	输入
pstInJsonBuf	输入的 json 配置, 请参考 Finetune_in.json 文件	输入
pstOutJsonBuf	输入的 Calib_out.json 配置, 请参考 Calib_out.json 文件	输入/输出
stateflag	Finetune state flag	输出

➤ 返回值

返回值	结果
E_MI_CALIB_SUCCESS	成功
E_MI_CALIB_INVALID_PARA	失败, 无效参数
E_MI_CALIB_ERROR	失败, 其他错误

➤ 依赖

- 头文件: `mi_calib.h`
- 库文件: `Calib.lib`

※ 注意

- 必须先 call [MI CALIB Calibration](#), 产生模型标定的结果后, 再 call [MI CALIB FineTune](#)。

➤ 举例

```
void Finetune_DEMO(const char *JSPath, const char *calibOutPath, const char *sysConfig, const char
*finetuneOutPath)
{
    MI_CALIB_Handle_t algHandle = NULL;
    MI_CALIB_BufBlock_t stInJsonBuf, stOutJsonBuf;
    int ftstateflag[10];

    MI_CALIB_Init(&algHandle, NULL);
    MI_CALIB_ReadJsFile(JSPath, &stInJsonBuf);
    MI_CALIB_ReadJsFile(calibOutPath, &stOutJsonBuf);

    MI_CALIB_FineTune(algHandle, sysConfig, &stInJsonBuf, &stOutJsonBuf, ftstateflag);

    MI_CALIB_WriteJsFile(finetuneOutPath, &stOutJsonBuf);
    MI_CALIB_Deinit(algHandle);
}

int main(int argc, char **argv)
{
    Calib_DEMO("Finetune_in.json", "Calib_out.json", "projectionErr.txt");
    Finetune_DEMO("Finetune_in.json", "Calib_out.json", "sysConfig.ini", "Finetune_out.json");
}
```

3. CALIB 数据类型

3.1.CALIB 模块相关数据类型定义

数据类型	定义
MI_CALIB_Conf_t	CALIB lib 标定配置结构体
MI_CALIB_BufBlock_t	CALIB lib 使用的内存块结构体
MI_CALIB_BufType_e	CALIB lib 的内存块的类型
MI_CALIB_RetCode_e	CALIB lib 的返回值类型
MI_CALIB_Handle_t	CALIB lib 的句柄类型
MI_CALIB_PhyAddr_t	CALIB lib 的内存块地址物理地址类型

3.2.MI_CALIB_Conf_t

➤ 说明

定义 CALIB lib 标定配置结构体。

➤ 定义

```
typedef struct MI_CALIB_Conf_t
{
    int camNum;
    int camPairNum;
    int camLensType;
    int imgWidth;
    int imgHeight;
    int inputType;
    int intrBoardWidth;
    int intrBoardHeight;
    int intrBoardSize;
    int extrBoardWidth;
    int extrBoardHeight;
```

```
int extrBoardSize;

int blendType;
int mapProjType;
int mapCropType;
int checkCalibMode;
int fastCalibMode;
int productCalibMode;
int productTuneMode;
double reserveAlpha;
int seamFunnel;
int waveCalib;
int fixDist;
int dumpMode;

// DPU
int edge_width;
int edge_height;
int tile_width;
int tile_StepX;
int min_d;
int num_d;
int en_dpu_crop;
char k_coef[200];

// map-gen
int orgWidth;
int orgHeight;
int scaleWidth;
int scaleHeight;
int in_seq_mode;
```

```
int out_seq_mode;
int src_mode;
int hw_mode;
int seprate_bin_mode;
char camRotation[200];
int gridSize;
int preCropType;

char ptguiPath[200];
char imgPath[200];
char paraPath[200];
char outputPath[200];
char moduleName[200];
char imgFormat[200];
char imgPrefix[200];
char imgDepthList[200];
char camIDList[200];
char camPairIDList[2][200];
char camCenterID[200];
char rotateAngleList[200];
char blendWidthPercentList[200];

char camOffsetX[200];
char camOffsetY[200];
char imgEdgeTop[200];
char imgEdgeBottom[200];
char imgEdgeLeft[200];
char imgEdgeRight[200];
char calibOutPath[200];
char configPath[200];
```

```
int VerticalNum;

    int updateBinMode;
    int autoCalcMode;
    int FOVX;
    int FOVY;
    int prjCenterX;
    int prjCenterY;
    int yaw;
    int pitch;
    int roll;
    char mask_en;
    char connect_type;
    char stitch_mode;
    int debug_mapgen;

    int genMapType;
    int gridsize;
    int rotationtype;
    int crop_En;
    int correctAlpha;
    int correctBeta;
    int cropwidth;
    int cropheight;
    int offx;
    int offy;
} MI_CALIB_Conf;
```

➤ Calib 相关成员

成员名称	描述
camNum	Camera 个数

成员名称	描述
camPairNum	Camera 对数
camLensType	Camera 镜头类型，0：广角镜头 1：超广角镜头
imgWidth	图像宽度
imgHeight	图像高度
inputType	标定类型，0：模型，1：产线标定，2：PTGui，3：Hugin
intrBoardWidth	内参标定时水平棋盘格角点个数
intrBoardHeight	内参标定时竖直棋盘格角点个数
intrBoardSize	内参标定时棋盘格子尺寸，单位 mm
extrBoardWidth	外参标定时水平棋盘格角点个数
extrBoardHeight	外参标定时竖直棋盘格角点个数
extrBoardSize	外参标定时棋盘格子尺寸，单位 mm
ptguiPath	ptGui/Hugin 参数路径
imgPath	标定图像路径
outputPath	输出图像路径
imgPrefix	图像前缀名，过滤使用
camIDList	Camera id 列表
camPairIDList	Camera Pair id 列表
camCenterID	中心 camera id
rotateAngleList	Camera 旋转角度列表
calibOutPath	标定输出路径

※ 注意事项

无

➤ 相关数据类型及接口

MI_CALIB_Calibration

3.3.MI_CALIB_BufBlock_t

➤ 说明

定义 CALIB lib 使用的内存块结构体。

➤ 定义

```
typedef struct
{
    void *pBufAddr;
```



```
MI_CALIB_PhyAddr_t phyAddr;
uint32_t u32BufSz;
MI_CALIB_BufType_e eBufType;
} MI_CALIB_BufBlock_t;
```

➤ 成员

成员名称	描述
pBufAddr	内存块的虚拟地址
phyAddr	内存块地址物理地址，在离线库中该参数无效
u32BufSz	内存块的大小
eBufType	内存块类型，在离线库中该参数无效

※ 注意事项

无。

➤ 相关数据类型及接口

MI_CALIB_Calibration、MI_CALIB_FineTune

3.4.MI_CALIB_BufType_e

➤ 说明

定义 CALIB lib 的内存块的类型。

➤ 定义

```
typedef enum
{
    E_MI_CALIB_BUFTYPE_NONCONTINUED = 0,
    E_MI_CALIB_BUFTYPE_CONTINUED = 1
} MI_CALIB_BufType_e;
```

➤ 成员

成员名称	描述
E_MI_CALIB_BUFTYPE_NONCONTINUED	非连续内存，离线算法库下只支持非连续内存

E_MI_CALIB_BUFTYPE_CONTINUED	连续内存
------------------------------	------

※ 注意事项
无

➤ 相关数据类型及接口
[MI_CALIB_Calibration](#)、[MI_CALIB_FineTune](#)

3.5.MI_CALIB_RetCode_e

➤ 说明
定义 CALIB lib 的返回值类型。

➤ 定义

```
typedef enum
{
    E_MI_CALIB_SUCCESS = 0,
    E_MI_CALIB_INVALID_PARA = 1,
    E_MI_CALIB_ERROR = 2
} MI_CALIB_RetCode_e;
```

➤ 成员

成员名称	描述
E_MI_CALIB_SUCCESS	成功
E_MI_CALIB_INVALID_PARA	失败，无效参数
E_MI_CALIB_ERROR	失败，其他错误

※ 注意事项
无

- 相关数据类型及接口

[MI_CALIB_Init](#)、[MI_CALIB_Deinit](#)、[MI_CALIB_Calibration](#)、[MI_CALIB_FineTune](#)

3.6.MI_CALIB_Handle_t

- 说明

定义 CALIB lib 的句柄类型。

- 定义

```
typedef void *MI_CALIB_Handle_t;
```

- 成员

成员名称	描述
------	----

- ※ 注意事项

无

- 相关数据类型及接口

[MI_CALIB_Init](#)、[MI_CALIB_Deinit](#)、[MI_CALIB_Calibration](#)、[MI_CALIB_FineTune](#)

3.7.MI_CALIB_PhyAddr_t

- 说明

定义 CALIB lib 的内存块地址物理地址类型。

- 定义

```
typedef unsigned long long int MI_CALIB_PhyAddr_t;
```

- 成员

成员名称	描述
------	----

- ※ 注意事项

无

➤ 相关数据类型及接口

[MI_CALIB_Calibration](#)、[MI_CALIB_FineTune](#)

4. CALIB 配置文件

4.1.CALIB 模块相关配置文件

数据类型	定义
Calib_in.json	标定输入的 json 配置文件
Calib_out.json	标定输出的 json 配置文件
reprojection_err.txt	标定输出的投影误差文件
Finetune_in.json	Finetune 输入的 json 配置文件
Finetune_out.json	Finetune 输出的 json 配置文件
sysConfig.ini	Finetune 算法参数文件

4.2.Calib_in.json

➤ 说明

定义 CALIB lib 标定输入的 json 配置文件。

➤ 定义



Calib_in.json

```

> {
>   "CalibInfo": {
>     "module": "Stitch",
>     "inputType": 0,
>     "blendType": 1,
>     "ptguiPath": "",
>     "imgPath": "./pattern/STITCH/stitch4_case1/stitchCalib1080_1920/",
>     "paraPath": "",
>     "outputPath": "./out/STITCH/stitch4_case1/",
>     "imgWidth": 1080,
>     "imgHeight": 1920,
>     "imgFormat": "YUV420",
>     "imgPrefix": "ispdev0_chn10",
>     "imgDepthList": "4000,10000,10000,10000",
>     "rotateAngleList": "0,0,0,0",
>     "intrBoardWidth": 8,
>     "intrBoardHeight": 11,
>     "intrBoardSize": 95,
>     "extrBoardWidth": 4,
>     "extrBoardHeight": 11,
>     "extrBoardSize": 95,
>     "camNum": 4,
>     "camIDList": "0,1,2,3",
>     "camPairNum": 3,
>     "camPairIDlist_0": "0,1,2",
>     "camPairIDlist_1": "1,2,3",
>     "camCenterID": "2",
>     "camLensType": 1,
>     "camOffxList": "0,0,0,0",
>     "camOffyList": "0,0,0,0",
>     "imgEdgeTopList": "0,0,0,0",
>     "imgEdgeBottomList": "0,0,0,0",
>     "imgEdgeLeftList": "0,0,0,0",
>     "imgEdgeRightList": "0,0,0,0",
>     "mapProjType": 2,
  }
}

```

➤ Calib 相关成员

成员名称	描述
camNum	Camera 个数，双目为 2，三目为 3
camPairNum	Camera 对数，双目为 1，三目为 2
camLensType	Camera 镜头类型，0：广角镜头 1：超广角镜头
imgWidth	图像宽度，单位 pixel
imgHeight	图像高度，单位 pixel
imgFormat	图像格式，RGB/YUV420
inputType	标定类型，0：模型标定，1：产线标定，2：PTGui，3：Auto-FineTune
intrBoardWidth	内参标定时水平棋盘格角点个数
intrBoardHeight	内参标定时竖直棋盘格角点个数
intrBoardSize	内参标定时棋盘格子尺寸，单位 mm
extrBoardWidth	外参标定时水平棋盘格角点个数
extrBoardHeight	外参标定时竖直棋盘格角点个数
extrBoardSize	外参标定时棋盘格子尺寸，单位 mm
ptguiPath	ptGui/Hugin 参数路径
imgPath	标定图像路径
outputPath	输出结果路径
imgPrefix	图像前缀名，过滤使用
camIDList	Camera id 列表

成员名称	描述
camPairIDList	Camera Pair id 列表
camCenterID	中心 camera id
rotateAngleList	Camera 旋转角度列表
module	模块名，拼接功能：“Stitch”，黑光融合：NIR
dumpMode	是否 dump 角点检测图；默认值是 0，代表不输出角点检测图和去畸变图；1 代表输出；

※ 注意事项
无

➤ 相关数据类型及接口

[MI_CALIB_Calibration](#)

4.3.Calib_out.json

➤ 说明

定义 CALIB lib 标定输出的 json 配置文件。

➤ 定义



Calib_out.json

<pre> "intrinsic": { "K_0": { "[0,0]": 1519.42021043319, "[0,1]": 0, "[0,2]": 745.904240190406, "[1,0]": 0, "[1,1]": 1520.23114868782, "[1,2]": 1329.82266758562, "[2,0]": 0, "[2,1]": 0, "[2,2]": 1 }, "D_0": { "[0,0]": -0.344430170836949, "[0,1]": 0.0936422422552936, "[0,2]": 0, "[0,3]": 0, "[0,4]": 0 }, "P_0": { "[0,0]": 1050.46752929688, "[0,1]": 0, "[0,2]": 746.862716863543, "[1,0]": 0, "[1,1]": 1038.990234375, "[1,2]": 1329.51248938381, "[2,0]": 0, "[2,1]": 0, "[2,2]": 1 } }, </pre>	<pre> "extrinsic": { "R_0": { "[0,0]": 0.0329366941325894, "[0,1]": 0.0249664509248669, "[0,2]": 0.999145560220248, "[1,0]": -0.0291966697152809, "[1,1]": 0.99928534233201, "[1,2]": -0.0240074796247715, "[2,0]": -0.99903089474608, "[2,1]": -0.0283809959059444, "[2,2]": 0.0336420928931854 }, "T_0": { "[0,0]": 51.8321246827122, "[1,0]": -2.8676184536315, "[2,0]": -93.1431287644178 } }, </pre>
--	---

➤ 成员

成员名称	描述
intrinsic K _i D _i P _i	Camera i 内参 K,D,P

成员名称	描述
extrinsic R _i T _i	Camera pair i 外参 R,T

※ 注意事项

无。

➤ 相关数据类型及接口

[MI_CALIB_Calibration](#)、[MI_CALIB_FineTune](#)

4.4.reprojection_err.txt

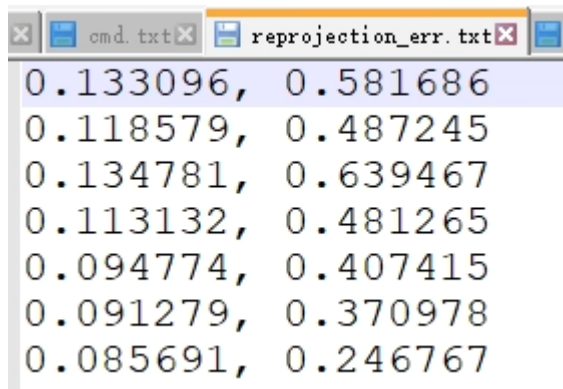
➤ 说明

定义 CALIB lib 标定输出的投影误差文件。

➤ 定义



reprojection_err.txt



➤ 成员

成员名称	描述
每个 Camera 投影误差	Camera i 平均投影误差 最大投影误差
每个 Camera pair 投影误差	Camera pair i 平均投影误差 最大投影误差

※ 注意事项

无。

- 相关数据类型及接口

[MI CALIB Calibration](#)、[MI CALIB FineTune](#)

4.5.Finetune_in.json

- 说明

定义 CALIB lib Finetune 输入的 json 配置文件。

- 定义



finetune_in.json

```
{
  "finetune": {
    "module": "Stitch",
    "ftimgPath": "./pattern/STITCH/stitch4_case3/finetune/pattern6/",
    "ftoutputPath": "./pattern/STITCH/stitch4_case3/finetune/pattern6/out/",
    "ftimgFormat": "YUV420",
    "ftimgPrefix": "ispdev0_chnl0",
    "imgWidth": 1920,
    "imgHeight": 1080,
    "imgFormat": "YUV420",
    "imgPrefix": "ispdev0_chnl0",
    "imgDepthList": "5000,5000,5000,5000",
    "rotateAngleList": "270,270,270,270",
    "camNum": 4,
    "camIDList": "0,1,2,3",
    "camPairNum": 3,
    "camPairIDList_0": "0,1,2",
    "camPairIDList_1": "1,2,3",
    "camCenterID": "2"
  }
}
```

- Calib 相关成员

成员名称	描述
camNum	Camera 个数
camPairNum	Camera 对数
imgWidth	图像宽度
imgHeight	图像高度
ftimgPath	Finetune 图像路径
ftimgFormat	图像格式，RGB/YUV420
ftoutputPath	Finetune 输出结果路径
ftimgPrefix	图像前缀名，过滤使用
camIDList	Camera id 列表
camPairIDList	Camera Pair id 列表
camCenterID	中心 camera id
rotateAngleList	Camera 旋转角度列表
module	模块名，拼接功能：“Stitch”

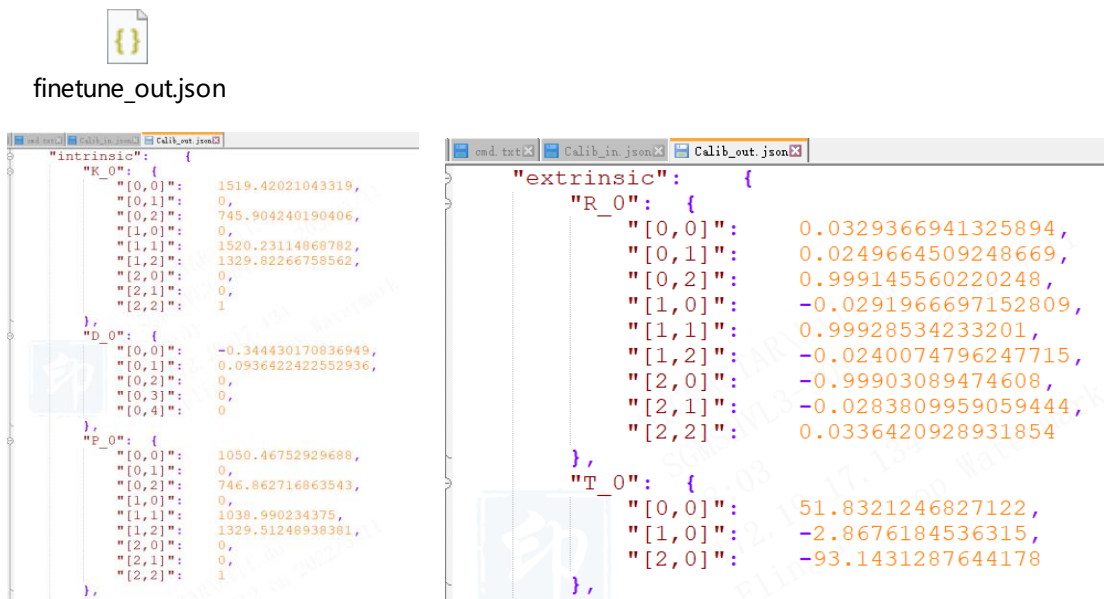
※ 注意事项
无

➤ 相关数据类型及接口
[MI CALIB FineTune](#)

4.6.Finetune_out.json

➤ 说明
定义 CALIB lib Finetune 输出的 json 配置文件。

➤ 定义



➤ 成员

成员名称	描述
intrinsic K _i D _i P _i	Camera i 内参 K,D,P
extrinsic R _i T _i	Camera pair i 外参 R,T

※ 注意事项
无。

➤ 相关数据类型及接口
[MI CALIB FineTune](#)

4.7.sysConfig.ini

- 说明
- 定义 CALIB lib Fintune 算法参数文件。
- 定义



- 成员

成员名称	描述
intrinsic K_i D_i P_i	Camera i 内参 K,D,P
extrinsic R_i T_i	Camera pair i 外参 R,T

- ※ 注意事项
- 无。
- 相关数据类型及接口
- [MI_CALIB_FineTune](#)