

# C/C++ 内部编程规范

## 1 注释

所有文件都必须有头注释，所有函数都必须有函数注释，注释规则如下

### 1.1 文件头注释

文件头注释必须包含以下信息，并在@brief字段后描述文件内容和用途，引用或取自开源项目的需要注明来源

```
/*
**
* @copyright Copyright (c) 2021 埃夫特智能装备股份有限公司
*
* @author Yi Tinghao (yitinghao@efort.com.cn)
* @version 0.1
* @date 2021-07-01
*
* @file detection2d.h
* @brief 用于2D视觉检测的函数库。提供以下功能：
*     特征点检测算子：
*         SIFT,
*         SURF,
*         ORB
*     直线检测：
*         Hough Transform,
*         Canny Line,
* 用例参考 https://efortdev.git/feat2d/example
**
*/
```

### 1.2 函数注释

函数注释必须包含以下信息，并在@brief字段后描述函数主要流程，业务算法函数需要注明出处如参考文献或参考方法/项目网址

```

/*****
***
* @brief 使用AABBCC算法的角点检测器，函数将输入图像image根据thresh使用AGAST算法检测角
点，
* 并在keypoints中返回角点信息。算法通过中心像素和周边的灰度差异判别角点，并将预处理后的潜
* 在角点进行Kmeans聚类得到最终的角点
* 代码引自
https://gitee.com/mirrors/opencv/blob/master/modules/features2d/include/opencv2/
features2d.hpp
* 算法详细解释参考 http://archive.www6.in.tum.de/www6/Main/ResearchAgast.html
* @param image 待检测的灰度图图像
* @param keypoints 检测出的角点数组
* @param threshold 检测时用于区分周边与中心像素的阈值
* @return int 检测到的角点的数量

*****/
/

```

## 1.3 插件推荐

VSCode中使用Doxygen Documentation Generator插件并在setting.json中直接添加以下内容，并修改作者和邮件信息。

使用时在需要添加的文件头或函数上方输入"/\*\*"后按回车即可。其它IDE需要调整至与上述文件格式一致，以便后续Doxygen能够自动生产文档。

```

{ // Doxygen documentation generator set
  "doxygen.file.copyrightTag": [
    "@copyright Copyright (c) {year} 埃夫特智能装备股份有限公司"
  ],
  "doxygen.file.customTag": [
    "@par 修改日志:",
    "<table>",
    "<tr><th>Date      <th>Version <th>Author  <th>Description",
    "<tr><td>{date} <td>1.0    <td>wangh    <td>内容",
    "</table>",
  ],
  "doxygen.file.fileOrder": [
    "copyright",
    "empty",
    "author",
    "version",
    "date",
    "empty",
    "file",
    "brief"
  ],
  "doxygen.file.fileTemplate": "@file {name}",
  "doxygen.file.versionTag": "@version 0.1",
  "doxygen.generic.authorEmail": "yitinghao@efort.com.cn",
  "doxygen.generic.authorName": "Yi Tinghao",
  "doxygen.generic.authorTag": "@author {author} ({email})",

```



## 3.2 文件命名

使用小写+下划线("\_")命名, 文件夹使用小写+短线("-")命名, 非代码和文件夹的文件, 在符合主流习惯的前提下可以使用其他命名规则

## 3.3 文件引用和路径

项目文件夹和构建文件夹均放置在"/home/efort/"路径下, 除非有其他合理原因, 否则全部使用绝对路径。数据库、配置等静态文件在源文件夹下编辑和管理, 在构建时通过CMake复制入构建文件夹, 项目使用时, 加载可执行文件所在目录同级的各文件夹下的对应文件。

## 4. 宏

### 4.1 头文件宏

```
#ifndef DETECTION3D_H //使用“文件名_H”防止重复引用
#define DETECTION3D_H

#endif//DETECTION3D_H
```

### 4.2 其他宏

```
#define ER_SAXPS_WAITTIME 1000 //项目特定的宏都要加ER_项目名_的前缀, 通用组件和库只加ER_前缀
```

## 5. 禁止事项

1. 禁止使用全局变量, 用静态指针或单例代替
2. 禁止使用相对路径, 对于可能动态变化的路径应该使用占位符+路径的模式例如

```
${PROJECT_DIR}/config/robot_config.yaml
```