ORB-SLAM Code Review 注释写作注意事项

本注释规范参阅Google C++ Style Guide[1]。以下对于ORB\_SLAM2工程中的代码所牵涉到的变量和函数提出相应注释规范。

1. 注释风格

对于/\*\* … \*/（块注释）和//...（行注释）这两种风格的注释均可使用。一般来说建议在类，函数和复杂算法的顶端之前使用块注释，在变量和具体实现中使用行注释。

一般来说IDE对于两种类型的高亮程度是不同的。例如：

块注释：

***/\*\****

***\**** @brief***:*** ***Frame类用于处理单，双，RGBD每一帧图像，完成了特征提取和深度计算等功能***

***\**** ***重要变量：mvKeys,*** ***mvKeysRight,mDescriptors,*** ***mDescriptorsRight,mvDepth,mGrid***

***\**** ***该类使用构造函数直接完成包括特征提取和深度计算的所有操作。使用时直接调用相应构造函数即可***

***\**** ***存在问题：Undistort函数在对于KeyPoint的去畸变是按照Caltech张正友的方法实现的，去畸变矩阵的意义不清楚***

***\**** ***如何去畸变不清楚。***

***\*/***

***Class Frame{…}***

行注释：

***// Copy constructor.***

*//结构函数--拷贝函数*

Frame(*const* Frame &frame);

1. 文件注释

ORB-SLAM中对于文件作者，工程等信息注释全面。如有需要请及时补充

1. 类注释

ORB-SLAM中类的命名意义明确，但作用不清楚，每个类在SLAM中的地位也不清楚。因此需要标明类的作用和主要和外界交互的函数和变量。例如上文中对于Frame的注释。可以使用使用brief关键字提供类的相关描述。

1. 变量注释

ORB-SLAM的变量命名相当规范。其中名称前面一小部分给出变量的存储方式：

m-→member variables, p-→pointers, v-→vector, n-→number(int), f-→float, b-→bool.

后面多采用单词标明变量含义。对于含义不清的，请注明含义。变量的具体功能，请根据变量的重要性提供。一般来说不是pointers的变量和pointers对应变量在类中被new了的，是比较重要的变量。例如：

std::vector<std::size\_t> mGrid[FRAME\_GRID\_COLS][FRAME\_GRID\_ROWS]; *//对地图区分grid，每一个cell里面对应的特征点数量*

1. 函数注释

函数头注释

ORB-SLAM多数函数能直接看懂意义。对于小型的settter和getter可仅注明对哪个变量进行了操作，特别明显的，也可以不写注释。对于大型的算法函数，以及和SLAM主要流程相关的函数，请务必注明函数的输入输出变量，函数的意义。对于函数中牵涉算法的，请简要注明算法的方法和参考文献。例如：

***/\*\****

***\**** @brief ***搜索左右图的匹配特征点对，计算并存储相应的右坐标值和深度值***

***\**** ***方法：对于左图中每一个Feature，获得右图中对应的搜索区域，在搜索区域中找到最相似的特征***

***\**** ***搜索区域是根据当前feature的octave，获得当前feature所在行的临近几行的范围，存储在vRowIndices中***

***\**** @param

***\*/***

void Frame::ComputeStereoMatches()

{…}

函数体注释：

对于重要算法步骤，请给出相应说明。尤其是在循环体开始之前。例如：

*//* *找到潜在点中描述子匹配最佳的点*

*for*(size\_t iC=0; iC<vCandidates.size(); iC++){...}

1. 算法注释