

System: : pubImageData()函数:

trackerData().readImage 对每一帧图像进行跟踪, 先对图像亮度进行调整; cur 表示当前帧, forw 表示下一帧, prev 表示上一帧。当前传入的图像是 forw。在 cur 不为空 (即已经初始化) 的情况下, 用光流法追踪 forw 中的关键点。reduceVector 根据追踪的 status 对成员变量 (prev_pts (像素坐标), cur_pts, forw_pts, ids, cur_un_pts (归一化坐标), track_cnt) 进行更新。ids 是个 vector, 存储点的编号 (后续记录 ids 如何记录点的编号)。只有 status 是好的, reduceVector 才会把点留下。un 下标代表经过去畸变的点, 存储的是归一化坐标, track_cnt 表示当前点被追踪到的次数。未经过初始化时不进行光流法追踪。

对追踪到的点的 track_cnt 自增。

对成员变量进行 rejectwithF() 验证, 用本质矩阵去除 outlier, 只保留不是 outlier 的成员变量, 同样通过 reduceVector 函数。由于检测新的特征点在 reduceVector() 后面进行, 因此对 prev_pts 和 cur_pts 进行保留时能保证保留的是匹配点。

对追踪到的点进行蒙版 setMask，不让追踪到的特征点被后续提取特征点重复提取。还会对追踪到的特征点根据追踪次数排序，让追踪次数多的点排在 forw_pts 前面，ids 和 track_cnt 也要改变次序。

如果当前帧追踪的特征点比较少，则进行提取特征点。提取到特征点通过 addPoint() 添加到 forw_pts 中，新点对应的 ids push_back (-1) 。

更新成员变量，cur 赋值给 prev，forw 赋值给 cur，通过 undistortedPoints() 得到去畸变后的点的归一化坐标，并在 cur_un_pts 中存储。并计算特征点在图像中的运动速度（看程序是归一化 x 和 y 的速度？），保存在 pts_velocity 中，用于下一帧光流法跟踪。

trackerData().readImage 结束后用 trackerData().updateID 对新点的编号进行更新；如果新点的 ids 为 -1，则赋值为 n_id，同时对 n_id 进行自增，这样保证了新的点的 id 号越来越大。

为什么轨迹舍弃前几帧的位姿？