Low-drift and real-time lidar odometry and mapping

作者：Ji Zhang

5.4、激光雷达里程计算法

Algorithm 1: Lidar Odometry

input : $\bar{\mathcal{P}}_{k-1}$, $\mathcal{P}_k$, $\boldsymbol{T}_k^L(t)$ from the last recursion at initial guess
output : $\bar{\mathcal{P}}_k$, newly computed $\boldsymbol{T}_k^L(t)$
begin
    if at the beginning of a sweep **then**
        $\boldsymbol{T}_k^L(t) \leftarrow \mathbf{0}$;
    end
    Detect edge points and planar points in $\mathcal{P}_k$, put the points in $\mathcal{E}_k$ and $\mathcal{H}_k$, respectively;
    for a number of iterations **do**
        for each edge point in $\mathcal{E}_k$ **do**
            Find an edge line as the correspondence, then compute point to line distance based on (7) and stack the equation to (9);
        end
        for each planar point in $\mathcal{H}_k$ **do**
            Find a planar patch as the correspondence, then compute point to plane distance based on (8) and stack the equation to (9);
        end
        Compute a bisquare weight for each row of (9);
        Update $\boldsymbol{T}_k^L(t)$ for a nonlinear iteration based on (10);
        if the nonlinear optimization converges **then**
            Break;
        end
    end
    if at the end of a sweep **then**
        Project each point in $\mathcal{P}_k$ to t_{k+1} and form $\bar{\mathcal{P}}_k$;
        Return $\boldsymbol{T}_k^L(t)$ and $\bar{\mathcal{P}}_k$;
    end
    else
        Return $\boldsymbol{T}_k^L(t)$;
    end
end

激光雷达里程计算法如算法 1 所示。该算法将上次扫描的点云$\overline{\mathcal{P}}_{k-1}$、当前扫描的增长点云$\mathcal{P}_k$和上次递归的位姿变换$T_k^L(t)$作为初始猜测作为输入。如果开始新的扫描，则 $T_k^L(t)$ 设置为零以重新初

始化（第 4-6 行)。 然后，该算法从$\mathcal{P}_k$中提取特征点以在第 7 行构造$\mathcal{E}_k$和$\mathcal{H}_k$。对于每个特征点，我们在$\overline{\mathcal{P}}_{k-1}$中找到其对应关系（第 9-19 行)。 运动估计适用于稳健的拟合框架 (Andersen 2008)。 在第 15 行，该算法为每个特征点分配一个二次方权重，如下式所示。与它们的对应关系具有较大距离的特征点被分配较小的权重，而距离大于阈值的特征点被认为是异常值，并被分配零权重。

$$w = \begin{cases} (1-\alpha^2)^2 & -1 < \alpha < 1, \\ 0 & \text{otherwise}, \end{cases} \tag{11}$$

where

$$\alpha = \frac{r}{6.9459\sigma\sqrt{1-h}}.$$

上式中，r 为最小二乘问题中对应的残差，σ 为残差与中位数的绝对偏差，h 为杠杆值或矩阵$J(J^TJ)^{-1}J^T$对角线上的对应元素，其中 J 与 (10) 中使用的雅可比矩阵相同。然后，在第 16 行，位姿变换被更新一次迭代。 如果发现收敛或满足最大迭代次数，则非线性优化终止。如果算法到达扫描结束，则使用扫描期间估计的运动将 $\mathcal{P}_k$ 投影到时间戳 t_{k+1}，形成$\overline{\mathcal{P}}_k$。这为下一次扫描匹配到$\overline{\mathcal{P}}_k$做好了准备。否则，算法仅返回变换$T_k^L(t)$用于下一轮递归。