

1 数学知识介绍

2 PTAM 算法的介绍

关键词，双管线系统，为后面的算法奠定了基础

3 PTAM 算法的数学推导

我们首先简述 PTAM 实现单目相机 SLAM 的原理。单目相机模型不同于双目相机，实时追踪时相机视界中的点不能和其它相机视界中的点进行匹配，只能和自己的关键帧匹配，从而加大了 3D 重建中定位的难度。PTAM 算法提出利用单目相机实时追踪特征点的可行性，实现三维重建。

3.1 PTAM 算法工作的基本流程

PTAM 算法主要思想是将 Tracking 和 Mapping 两个过程放在不同的管线 (进程) 中进行: Tracking 进程专门实现相机位置的估计, Mapping 进程则用于进行关键帧之间的误差消除。

如果记 W 为真实世界的坐标系, PTAM 算法将维护一个关键帧集合: $Img = \{I_1, I_2, \dots, I_m\}$, 这 m 个关键帧分别对应 m 个相机坐标系 K_i 。我们用 $E_{K_i W}$ 表示从世界坐标系到相机坐标系的仿射变换 (Affine Transformation)。

3.2 PTAM 管线之一: Tracking

追踪进程解决的一个根本问题是:

当读入了新的关键帧之后, 原来我在重建过程中提取的 3 维空间中的特征点现在在照片中的坐标是什么? 我现在的相机 z 姿态怎么估计?

给定

3.2.1 关键特征点的投影和配对

我们假定程序可以从映射进程得到一个关键帧集合 Img 以及 3 维重建的特征点的坐标集合 (相对于世界坐标系) $P = P_W = \{\mathbf{p}_{1W}, \dots, \mathbf{p}_{sW}\}$, 为了统一形式, 将第 j 个点坐标标记为 $\mathbf{p}_{jW} = (p_{jx}, p_{jy}, p_{jz}, 1)$

$$\mathbf{p}_{jK_{t+1}} = E_{K_{t+1}} \mathbf{p}_{jW} \quad (1)$$

3.2.2 相机模型参数的求解和位置的更新

为了保证算法的稳定性, 追踪进程会进行两次: 第一次会从三维模型中抽取 50 个特征点投影匹配, 第二次则抽取 1000 个特征点进行匹配。

3.3 PTAM 管线之二: Mapping

3.3.1 关键帧的选择和插入

3.3.2 利用 Bundle Adjustment 极小化误差

4 PTAM 算法的实际应用

4.1 在一般电脑上运行

4.2 在智能手机上运行

5 PTAM 算法的评价