哈 尔 滨 工 业 大 学

硕士学位论文中期报告

题 目：基于双目视觉的四旋翼定位系统设计与实现

院　　（系）航天学院

学　　科　控制科学与工程

导　　师　马杰

研　究　生　何芳

学　　号　**15S104141**

中期报告日期　**2017 年 3 月 18 日**

研究生院制

目录

1 课题主要研究内容及进度情况

1.1 课题主要研究内容

随着四旋翼无人机研究的领域越来越广泛，对于实现避障、路径规划以及抓取等复杂任务的基础和前提就是对其准确地、实时地自主定位。摄像头作为一种功耗低、信息量丰富、可靠性高、质量轻的传感器，随着计算机视觉技术的发展被广泛使用，还可以确定场景中的三维深度信息，可以估算出精确的轨迹。即时定位与地图构建(SLAM)作为一种高性能的基于摄像头图像信息估计自身姿态和实时地重建场景的系统，随着实现无人机以及机器人自主性问题的提出，视觉SLAM 系统成为目前应用最为广泛且普适性更强的定位方法。

本课题采用摄像头作为传感器，信息量丰富，但处理的数据量过多导致计算量大，应用于四旋翼平台，受到四旋翼机载电脑处理能力的限制，需要解决在线实时姿态估计这一问题是主要难点。其次，四旋翼存在快速运动和大机动的情况，跟踪频繁丢失或明显出现漂移，提升定位系统的鲁棒性是另一关键问题。通过本课题研究解决以上两个问题，给出基于机载双目视觉的四旋翼定位系统，实现在四旋翼平台实时获取位置信息，在四旋翼出现大机动和快速运动情况下高效而鲁棒的实现定位功能。

本课题提出基于半直接法的实时双目视觉定位系统，可以实时地估计相机位姿，这对于获得良好的四旋翼飞行器位姿信息是十分重要的。本文主要从基于半直接法的实时双目视觉特征跟踪方法的设计与实现，跟踪系统的局部地图扩张及优化的框架的设计与实现，以及最终实验验证及性能分析等四个方面展开研究。具体研究如下：

（1）实时双目视觉 SLAM 系统分析设计

首先对现有先进的视觉SLAM定位系统进行评测，结合四轴飞行器的特点，从实时性、在缺乏特征的场景下的鲁棒性、大幅度旋转的跟踪鲁棒性等方面对比分析，为后续的研究打下基础。通过视觉 SLAM 系统分析，对影响系统实时性的主要原因及关键技术进行归纳总结，结合课题需求，提出了基于半直接法的双目视觉 SLAM 系统。

（2）基于半直接法的双目视觉特征跟踪方法设计

提出了一种基于半直接法的双目视觉系统估计 3 维点深度的方法，既可以用于跟踪同一时刻左右两帧图像相互匹配的像素位置对应的同一个空间点的深度信息，还可以用于跟踪相邻时刻帧与帧之间的相对位姿，以及相匹配对应的三维点的深度。同时，我们的方法不仅能使视觉 SLAM 系统前端 VO 达到实时的计算效率，又能够实时运行于四旋翼无人机平台，以及移动设备等。

（3）跟踪系统的局部地图扩张及优化的框架的设计与实现

实现基于半直接法的双目视觉特征跟踪的局部地图扩张及优化的框架。在视觉定位系统中仅使用前端估计相机位姿，是远不能达到我们所需的精度，系统添加实时地三角化和优化新测得的三维点进行局部建图，又能在无人机飞行到先前时刻飞行过的位置进行回环检测，优化位姿，减少漂移，该方法能在四旋翼无人机快速运动和严重模糊的情况下显著地提高鲁棒性。同时我们系统可以使用在四旋翼无人机平台中，可以得到实际的 IMU 的测量，则可以将 IMU 测量的数据集成到优化框架中来进一步提高鲁棒性和精度。

（4）四旋翼双目视觉定位系统实验验证

深入学习机器人操作系统(ROS)的运行机制，搭建大疆经纬 M100 实验平台，对定位系统进行嵌入式移植，并通过实验验证无人机正常飞行、出现大机动、快速运动等情况下，本系统提出的算法与现有先进 SLAM 系统在实时性、定位精度等性能指标对比，以及根据实验效果对误差进行分析。因此，最终需要完成整个系统的测试和实验验证。

1.2 课题进度情况

（1）实时进行特征跟踪、优化相机位姿方法，确定了基于特征点法和基于直接法的工作流程，分别学习、评测过基于这两种方法具有代表性、开源的 SLAM 系统，明确了这两种方法的特点和使用条件，以及应用于四旋翼无人机平台的可行性，并对这些 SLAM 系统的性能以及采用的新颖方法进行了研究分析和归纳总结。

（2）基于半直接法的双目视觉跟踪方法设计已基本完成，确定使用半直接法估计观测到空间点的深度信息的优化算法，推导完成，对算法的步骤进行了详细阐述。明确了该优化方法的特点和使用条件，并在机器人操作系统以及 Linux 操作系统中实现完成，使用现有四旋翼无人机飞行数据集进行实时性测试，并与现有开源的基于特征点法的特征追踪定位系统进行对比。

（3）对现已完成的系统的整体简化也正在进行中，分析当摄像头水平朝前运动的时候，估计的深度误差较大的原因，改进算法解决问题。

2 目前已完成的研究工作及成果

2.1 经典 SLAM 系统框架

视觉 SLAM 的目标，是通过单目、双目或者 RGB-D 相机获得场景图像，根据对图像处理进行定位和地图构建，完成这个过程，需要一个完善的算法框架，这个算法框架主要分成四个模块：视觉里程计(Visual Odometry，VO)、后端优化(Optimization)、回环检测(Loop Closing)、地图构建(Mapping)。整体视觉 SLAM 系统模块组成图如 2-1 所示。

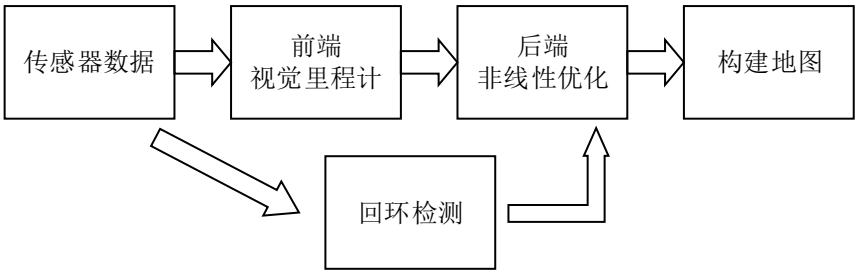

图 2-1 整体视觉 SLAM 系统模块组成图

完整的视觉 SLAM 流程分为以下几个步骤：

1. 读取视觉传感器的图像数据。在视觉 SLAM 中主要为相机的图像信息的读取和预处理。如果用在机器人中，还可能存在码盘、惯性传感器等信息的读取和同步。
2. 视觉里程计(Visual Odometry，VO)。视觉里程计任务是估算相邻图像间的相机运动，以及局部地图的样子。VO 又称为前端(Front End)。
3. 后端优化(Optimization)。后端接受不同时刻视觉里程计测量的相机位姿，以及回环检测的信息，对他们进行优化，得到全局一致的轨迹和地图。由于在 VO 之后，又称为后端(Back End)。
4. 回环检测(Loop Closing)。回环检测判断机器人是否曾经到达过先前的位置，如果检测到回环，他会把信息提供给后端进行处理。
5. 地图构建(Mapping)。它根据估计的轨迹，建立与任务要求对应的地图。

2.1 视觉 SLAM 前端定位算法理论基础

移动机器人视觉 SLAM 是指自主移动机器人在未知环境中，行进时通过自身位姿估计和视觉传感器观测信息增进式地构建环境地图，同时利用该地图实现自主定位和导航。正如 SLAM 的完整算法框架中所述，视觉 SLAM 主要分为视觉前端和优化后端，前端也称为视觉里程计(VO)。VO 的主要问题是根据图像来估计相机的运动，根据视觉传感器获取到的相邻时刻的图像，粗略估计出相机的运动以及地图中路标点 3D 位置信息，给后端提供较好的初始值进行迭代优化。

这个前端处理流程，按照是否需要特征提取，分为特征点法的前端以及不提供特征的直接法前端。通过研究前端 VO 定位方法，针对本课题为提高无人机立体视觉定位系统的实时性的目标，针对前端获取位姿这两种方法进行对比分析。

2.2.1 基于特征点法定位

基于特征点法的前端 VO，是目前比较成熟的解决方案，在现今流行的 SLAM 前端 VO 中单目、双目、RGB-D 相机中均有应用，基于特征点法的 VO 通常从图像中选取比较有代表性的点，即特征点，在计算机视觉的领域已经提出许多图像特征，如 SIFT、SUFT、FAST、ORB 等。使用特征点法估计相机运动时，处理的主要流程是提取特征点、计算特征描述子以及特征匹配，把特征点看作固定在三维空间的不动点。根据他们在相机中的投影位置，通过最小化重投影误差来优化相机运动，估计两帧之间的相机运动和场景结构，从而实现一个基本的 VO。在这个过程中我们需要精确地知道空间点在两个相机中投影后的像素位置，这也是为什么要对特征点进行匹配或跟踪的理由。

近些年来，基于视觉的定位系统的在移动机器人的应用越来越广泛，移动在执行一些复杂的任务时需要能对周围环境实时的自主建图，立体视觉相对单目视觉可以获得环境的绝对深度，对环境的建图更加准确，因此基于立体视觉的里程计算法更加适用于本系统。结合课题的需求，主要分析基于双目视觉定位的基本系统框图，如图 2-2 所示。基于特征点的双目立体视觉 VO 的主要执行步骤如下：

（1）图像对的获取：采用立体相机获取来自同一场景的立体图像对的观测图像序列，并转换成灰度图像对 $I_{t|1}$, $I_{t+1|1}$, $I_{t|r}$, $I_{t+1|r}$ 运算效率会更高。

（2）图像校正及视差图的获取：通过几何校正得到无畸变图像，使得立体图像外极线配准，分别计算从左右相机获取 t 时刻和 t+1 时刻的图像的视差图 D_t, D_{t+1}。

（3）特征检测：在校正后的图像上使用 FAST 特征检测算法对图像进行特征检测。

（4）立体特征匹配：分别对 t 时刻和 $t+1$ 时刻的特征点对进行立体特征匹配，使用视差图 D_t, D_{t+1} 来计算上一步检测到特征点的三维坐标，得到两个点云 W_t, W_{t+1}。

（5）特征追踪：采用跟踪算法检测出的特征到图像 I_t 中，跟踪丢失的特征在点云 W_t, W_{t+1} 中移除。如果跟踪特征有所丢失，特征数小于某个阈值，则重新进行特征检测。

（6）三角化 3D 点云：对上一步得到的点云 W_t, W_{t+1} 进行三角化处理，得到特征点的三维坐标。

（7）内点检测：在上一步中更新得到的 t 时刻的点云 W_t 中的任意两个特征点

之间的距离，在$t+1$时刻的点云W_{t+1}中相应的两个特征点之间的距离必须相等。选取出这样一个特征点的最大团作为内点集合。其中最大团至少有 8 个特征点。

（8）运动估计：结合前后两个时刻图像匹配特征点的三维坐标做运动估计，采用 Levenberg-Marquardt 算法最小化重投影误差，估计出t时刻和$t+1$时刻之间的相对旋转矩阵 R 和平移向量 T，其中重投影误差需要满足小于一定阈值。

（9）位置优化：由局部捆集调整构成，进行位置优化。

（10）重复上述过程，计算相机姿态是通过递增的方式，对每次计算的相对量进行累加，估算出绝对位置以及姿态。

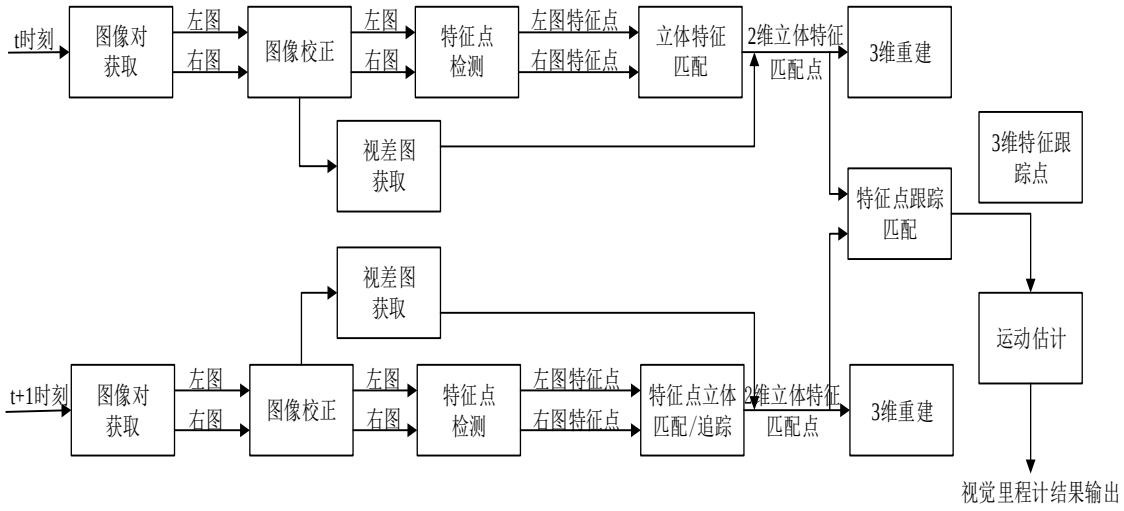

图 2-2 双目视觉定位系统原理框图

2.2.2 基于直接法定位

随着现今一些基于直接法的 SLAM 系统的流行，如 SVO、LSD-SLAM、DSO-SLAM，直接法成为一种新的潮流，直接法根据像素的亮度信息，估计相机的运动，可以完全不用计算关键点和描述子。于是，既避免了特征的计算时间，也避免了特征缺失不可工作的情况。只要场景中存在明暗变化（可以是渐变，不形成局部的图像梯度），直接法就能工作。根据使用像素的数量，直接法分为稀疏、稠密和半稠密三种。直接法是为了克服特征点法的上述缺点而存在的。

1．基于直接法定位的工作原理

如图 2-3 所示，考虑某个空间点P和两个时刻的相机。空间点P的世界坐标为$[X,Y,Z]$，他在两个相机上成像，记非齐次像素坐标为p_1,p_2。

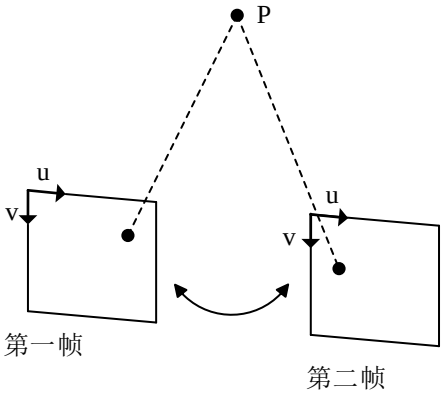

图 2-3 直接法示意图

我们的目标是求第一个相机到第二个相机的相对位姿变换。我们以第一个相机为参照系，设第二个相机旋转和平移为 R,t（对应李代数 ζ）。同时，两个相机的内参相同，记为 K，完整的投影方程如下：

$$\begin{aligned} p_1 &= \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_1 = \frac{1}{Z_1} KP, \\ p_2 &= \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_2 = \frac{1}{Z_2} K(RP+t) = \frac{1}{Z_2} K(\exp(\zeta\hat{})P)_{1:3} \end{aligned} \tag{2-1}$$

其中 Z_1 是 P 的深度，Z_2 是 P 的第二个相机坐标系下的深度，也就是 $RP+t$ 的第三个坐标值。由于 $\exp(\zeta\hat{})$ 只能和齐次坐标相乘，所以我们乘完之后要取出前三个元素。

相比特征点法，在直接法中，由于没有特征匹配，我们无从知道哪一个 p_2 与 p_1 对应着同一个点。直接法的思路是根据当前相机的位姿估计值，来寻找 p_2 的位置。但若相机位姿不够好，p_2 的外观和 p_1 会有明显差别。于是，为了减小这个差别，我们优化相机的位姿，来寻找与 p_1 更相似的 p_2。同样可以通过解一个优化的问题，但此时最小化的不是冲投影误差，而是光度误差，也就是 P 的两个像的亮度误差：

$$e = I_1(p_1) - I_2(p_2) \tag{2-2}$$

这里 e 是一个标量。同样，优化的目标为该误差的二范数，暂时取不加权的形式，为：

$$\min_{\zeta} J(\zeta) = \| e \|^2 \tag{2-3}$$

同样是基于灰度不变的假设。在直接法中，我们假设一个空间点在各个视角下，成像的灰度是不变的。假设有 N 个空间点 P，那么，整个相机位姿估计问题变为：

$$\min_{\zeta} J(\zeta)=\sum_{i=1}^{N} e_i^T e_i, \qquad e_i = I_1(p_{1,i}) - I_2(p_{2,i}) \tag{2-4}$$

这时的优化变量是相机位姿ζ。为了求解这个优化问题，关系误差e是如何随着相机位姿ζ变化的，需要分析他们的导数关系。因此，使用李代数上的扰动模型。给$\exp(\zeta)$左乘一个小扰动$\exp(\delta\zeta)$，得：

$$\begin{aligned}\exp(\zeta\oplus\delta\zeta) &= I_1\left(\frac{1}{Z_1}KP\right) - I_2\left(\frac{1}{Z_2}K\exp(\delta\zeta\wedge)\exp(\zeta\wedge)P\right)\\ &\approx I_1\left(\frac{1}{Z_1}KP\right) - I_2\left(\frac{1}{Z_2}K(1+\delta\zeta\wedge)\exp(\zeta\wedge)P\right)\\ &= I_1\left(\frac{1}{Z_1}KP\right) - I_2\left(\frac{1}{Z_2}K\exp(\zeta\wedge)P + \frac{1}{Z_2}K\,\mathrm{e}\,\delta\zeta\wedge\,\mathrm{xp}(\zeta\wedge)P\right)\end{aligned} \tag{2-5}$$

记为：

$$\begin{aligned} q &= \delta\zeta\wedge\exp(\zeta\wedge)P\\ u &= \frac{1}{Z_2}Kq\end{aligned} \tag{2-6}$$

这里q为P在扰动之后，位于第二个相机坐标系下的坐标，而u为他的像素坐标，利用一阶泰勒展开，有：

$$\begin{aligned}\exp(\zeta\oplus\delta\zeta) &= I_1\left(\frac{1}{Z_1}KP\right) - I_2\left(\frac{1}{Z_2}K\exp(\zeta\wedge)P + u\right)\\ &\approx I_1\left(\frac{1}{Z_1}KP\right) - I_2\left(\frac{1}{Z_2}K\exp(\zeta\wedge)P\right) - \frac{\partial I_2}{\partial u}\frac{\partial u}{\partial q}\frac{\partial q}{\partial\delta\zeta}\delta\zeta\\ &= e(\zeta) - \frac{\partial I_2}{\partial u}\frac{\partial u}{\partial q}\frac{\partial q}{\partial\delta\zeta}\delta\zeta\end{aligned} \tag{2-7}$$

我们可以看淡一阶导数由于链式法则分成了三项，这三项都是容易计算的：

（1）$\frac{\partial I_2}{\partial u}$为u处的像素梯度；

（2）$\frac{\partial u}{\partial q}$为投影方程关于相机坐标系下的三维点的导数。记$q=[X,Y,Z]^T$，导数为：

$$\frac{\partial u}{\partial q} = \begin{bmatrix}\frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z}\\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z}\end{bmatrix} = \begin{bmatrix}\frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2}\\ 0 & \frac{f_y}{Z} & -\frac{f_x Y}{Z^2}\end{bmatrix} \tag{2-8}$$

（3）$\frac{\partial q}{\partial\delta\zeta}$为变换后的三维点对变换的导数，这在李代数章节已经介绍过了：

$$\frac{\partial q}{\partial\delta\zeta} = [I, -q\wedge] \tag{2-9}$$

在实践中，由于后两项只与三维点 q 有关，与图像无关，把它合并到一起：

$$\frac{\partial u}{\partial \delta\zeta}=\begin{bmatrix}\frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x+\frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z}\\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y-\frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z}\end{bmatrix} \quad (2\text{-}10)$$

这个 2×6 的矩阵在上一讲中也出现过。于是，推导了误差相对于李代数的雅克比矩阵：

$$J=-\frac{\partial I_2}{\partial u}\frac{\partial u}{\partial \delta\zeta} \quad (2\text{-}11)$$

对于 N 个点的问题，可以用这种方法计算优化问题的雅克比矩阵，然后使用 G-N 或 L-M 计算增量，迭代求解。

2. 直接法分类

(1)稀疏直接法（半直接法 semi-direct）：通常采用数百个至上千个关键点，假设他周围像素是不变的。这种稀疏直接法不必计算描述子，并且只使用数百个像素，因此速度最快，适用于实时性较高而且计算资源有限的场合和平台，但只能计算稀疏的重构。（semi-direct）可以做到非常快速的效果。

(2)半稠密直接法(semi-dense)，如果像素梯度为 0，不会对计算运动增量有任何贡献。因此，可以考虑只使用带有梯度的像素点，舍弃像素梯度不明显的地方，可以重构一个半稠密结构。

(3)稠密直接法：稠密重构需要计算所有像素（一般为几十万至几百万个），因此多数不能再现有的 CPU 上实时计算，需要 GPU 的加速。但是，正如前面所讨论的，梯度不明显的点，在运动估计中不会有太大的贡献，再重构是也会难以估计位置。

2.2.3 特征点法与直接法对比分析

结合上述对特征法以及直接法原理、特点分析，针对本课题解决的主要问题，从实时性、场景适应性、鲁棒性等几个方面对比分析这两种方法的优缺点。

(1) 实时性

基于特征点法中提取特征点、计算描述子以及特征匹配需要付出大量的计算量，非常耗时。实际处理过程中，SIFT 目前在 CPU 上是无法实时计算的，而 ORB 也需要近 20 毫秒的计算。进来相对先进的 SLAM 系统之一 ORB-SLAM 就是基于 ORB 特征点进行定位，整个 ORB-SLAM 以 30 毫秒/帧的速度运行，那么一大半时间都花在计算特征点上。相对的，在直接法中，我们并不需要知道点与点之间的对应的关系，根据像素的光度信息，通过最小化光度误差估计相机运动，可以避免计算关键点和描述子，既避免了特征的计算时间。同时直接法在提高相机频率的情况下，可以降低每帧的计算量，减少时间消耗。因此，直接法相比特征法实时性相对较好。

(2) 提供信息的丰富程度

基于特征点法是从图像中选取比较有代表性的点，忽略了除特征点以外的所有信息。一张图像有几十万个像素，而提取的特征点只有几百个，只使用特征点丢弃了大部分可能有用的图像信息。相对比，直接法使用图像中的所有像素，信息丰富。

(3) 特征缺失情况下的鲁棒性

相机有时会运动到特征缺失的地方，往往这些地方都没有什么明显的纹理信息。例如，有时我们会面对一堵白墙，或者一个空荡荡的走廊。这些场景下特征点数量会明显减少，在这种情况下，基于特征点的方法，我们可能找不到足够的匹配点来计算相机运动。相对的，在直接法中，只要求有像素梯度即可，无需特征点。因此，直接法可以在特征缺失的场合下使用。在比较极端的例子是只有渐变的一张图像，可能无法提取角点类的特征，但可以用直接法估计其运动。因此，直接法相比特征点法在特征缺失情况下的鲁棒性更好。

(4) 光照变换环境下的鲁棒性

直接法是基于灰度不变这个假设条件的，实际当中很可能不成立。由于物体的材质不同，像素会出现高光和阴影部分。有时，相机会自动调整曝光参数，使得图像整体变亮或者变暗，光照变化时亦会出现这种情况。这些时候灰度不变假设都是不成立的。相对比，特征点法对光照具有一定的容忍性，保证特征不变的情况下，精度较高。而直接法由于计算灰度间的差异，整体灰度变化会破坏灰度不变的假设，使算法失败。针对这一点，目前的直接法开始使用更细致的光度模型标定相机，以便在曝光时间变化时也能让直接法工作。因此，直接法相比特征点法在光照变换环境下的鲁棒性更好。

(5) 快速运动的鲁棒性

基于特征点法是从图像中选取特征点，这些特征点在相机视角发生一定变化后仍会保持不变，所以我们会在各个图像中找到相同的特征点。因此，在相机运动比较快的时候，图像帧与帧之间的运动比较大，还是可以找到一定数量的匹配特征点，仍可估计出帧间运动。相对比，直接法是基于灰度不变这个假设条件才可使用的，该假设条件成立的基础是需满足帧间运动是微小的。同时，直接法完全依靠梯度搜索，降低目标函数来计算相机位姿。其目标函数中需要取像素点的灰度值，而图像是强烈非凸的函数，这使得优化算法容易进入极小，只能在帧与帧之间的相对运动较小的条件下直接法才能成功。因此，特征点法相比直接法在相机快速运动的鲁棒性更好。

(6) 地图构建

特征点法仅使用图像中的个别像素，再重构地图时，只能重构稀疏地图，相对比，直接法分为稀疏、稠密和半稠密三种。还具有恢复稠密、半稠密结构的能力。稠密和半稠密地图更适用于移动机器人执行路径规划、避障等任务。

2.3 基于半直接法的立体视觉 SLAM 系统设计与实现

近些年来，基于特征点法的定位跟踪被认为是视觉里程计的主流方法，它运行稳定、对光照、动态物体不敏感，是目前比较成熟的解决方案，许多先进的 SLAM 系统都采用基于特征点法实现前端定位、跟踪功能，比如 RD-SLAM 、PTAM、ORB-SLAM2。RD-SLAM 检测场景的外观和结构变化，提出了一种修改过的 RAMSAC 方法来实现动态场景中的定位。PTAM 采用一种新颖的基于关键帧的并行跟踪和地图创建框架。为了获得实时的性能，将跟踪从地图创建中独立出来；为了保证较高的精度，可以在关键帧之间做集束调整(BA)。随着 SVO，LSD-SLAM、DSO-SLAM 这些基于直接法的 SLAM 系统的流程，直接法本身也得到越来越多的关注。LSD-SLAM 采用可以在 CPU 上实时计算的半稠密图代替稀疏的特征点。因此，可以直接根据半稠密深度图来是实现定位，从而缺乏特征的场景下提高鲁棒性。SVO 也采用直接跟踪，但是采用稀疏的特征点，节省了特征提取的巨大开销。因此这个策略可以实现很高的频率。同时 SVO 和 LSD-SLAM 都提出采用滤波技术来获得鲁棒的深度估计。

2.3.1 ORB-SLAM2 系统

对现有的 SLAM 系统结合论文以及开源出的代码进行实际测试，综合分析，ORB-SLAM2 系统中基本延续了 PTAM 的算法框架,但对框架中的大部分组件都做了改进，在工程角度也更加适合应用、平台移植。ORB-SLAM2 采用 ORB 特征来做跟踪、地图创建、重定位和回环检测。同时通过优化姿态图来使环路闭合。测试对比着几种先进的 SLAM 系统得出结论，ORB-SLAM2 可以实时运动在 PC 上，适用于各种场合，室内的或者室外的，大场景或者小场景。同时该系统采用了所有 SLAM 相同的功能：追踪，地图构建，重定位和闭环检测、甚至全自动位置初始化。选用了比较适合的策略，地图重构的方法采用云点和关键帧技术，具有很好的鲁棒性，生成了精简的、可追踪的地图，当场景的内容改变时，地图构建可持续工作。

对 ORBSLAM2 系统进行测试发现，该系统仍有所有基于特征点方法的 SLAM 系统的共性问题：移动端应用的实时性以及图像模糊跟踪丢失等问题。尽管 ORBSLAM 系统在 PC 机以 30ms/帧的速度进行实时计算，但在嵌入式平台上表现不佳，针对应用平台是四旋翼无人机，仍不能实现实时处理。由于无人机在发生大机动的情况下会导致获取的图像模糊，在这种情况下，定位系统的鲁棒性较差、经常出现跟踪丢失的情况。

ORBSLAM2 系统，该系统是一个基于特征识别的 SLAM 系统，正如传统的基于特征点的方法有一个缺点，该方法需要提取特征点、对其计算描述子、以及特征匹配等步骤，这些处理是很耗时的，在移动端、处理能力较差的嵌入式平台不满足实时性。同时针对鲁棒性也就是图像模糊跟踪丢失无法继续定位的问题，

是由于获取的图像模糊很难提取到特征点，找不到足够的匹配点计算相机运动导致的。课题现在的主要目的就是解决针对无人机应用平台使用 ORBSLAM2 系统存在的问题进行改进。

2.3.2 基于半直接法的 ORB-SLAM2 系统前端算法设计

针对应用在无人机平台使用 ORBSLAM2 系统存在的问题，根据上述对直接法(direct-method)和特征法(feature-method)的对比分析，可以发现直接法可以完全不用计算关键点和描述子，既避免了特征的计算时间，也避免了特征缺失的情况。由于计算关键点和描述子以及特征匹配占用整个系统的一大半时间，所以直接法可以很大程度提升实时性。同时只要满足场景中存在明暗变化（可以是渐变，不形成局部的图像梯度），直接法就能工作，可以一定程度上改善特征点法由于图像模糊导致的特征缺失无法准确跟踪的问题。同时前面介绍过直接法分为稀疏、半稠、稠密三种，对于应用在实时性较高而且计算资源有限的无人机平台上，稀疏直接法(semi-direct)是最好的选择，该方法不必计算描述子，并且只使用数百个像素，对图像中的特征点图像块进行直接匹配来获取相机位姿，而不像直接匹配法那样对整个图像使用直接匹配。稀疏直接法(semi-direct)方法和直接法(direct-method)另一个不同的是它利用特征块的配准来对直接法(direct-method)估计的位姿进行优化。因此速度快，鲁棒性好，更适用于该课题的需求。

对于现有的整幅图像的直接法(direct-method)常见于 RGB-D 相机，稀疏直接法(semi-direct)方法常应用于单目视觉中。结合已看过的文献和综述，直接法还没有应用在双目系统上，对于现有的双目系统获取空间点 3D 位置信息常采用的还是特征法(feature-method)，根据提取出的特征点进行左右图像匹配，获取视差图以及深度信息，为提升整个系统的实时性，尽量避免使用特征点进行位姿估计，由于本系统采用双目摄像头，所以在不提取特征点的前提下怎么进行双目匹配估计深度，是待解决的主要问题之一。

本文提出了一种基于半直接法的双目视觉系统估计 3 维点深度的方法，本方法的关键是一种基于半直接法的双目匹配方法，本算法主要分为以下几步：

1. 初始化过程。使用第一时刻左图像为基准，构建金字塔，提取 Fast 特征点，如果第一帧图像提取的特征点数满足一定阈值，然后根据第一时刻右帧图像用左帧图像的特征点做光流跟踪，解算该时刻左右两帧图像之间的单应矩阵，同时求解出当前跟踪到的特征点的内点数，如果内点数满足一定阈值，则标记设置该时刻的左帧图像为关键帧，添加到地图中，用于地图扩展。同时，同时对该时刻左右两帧图像进行最小化图像重投影残差，估计初始时刻左右两帧之间的内殿对应的空间点的深度。否则，使用后面时刻的图像重新进行初始化。

2. 基于半直接法的双目匹配算法具体过程。该部分算法的优化变量是图像中特征点对应的 3 维空间中的深度变量。

（1） 通过相机标定，平行双目视觉中左右相机之间的相对位姿$T_{l,r}$已经确定，通过之前多帧之间的特征检测以及深度估计，我们已经知道第$k-1$帧中特征点位置以及他们的深度，还可估计出当前时刻k中特征块的初始位置。

（2） 知道当前时刻左帧图像I_{k_l}中的某个特征在图像平面的位置(u,v)，需其深度d，能够将该特征投影到三维空间p_k，该三维空间的坐标系是定义在当前时刻左摄像机I_{k_l}坐标系的。所以，我们要将它投影到当前时刻右摄像机I_{k_r}图像中，需要进行位姿转换$T_{l,r}$，得到该点在当前时刻右摄像机坐标系中的三维坐标p_k。最后通过摄像机内参数，投影到当前时刻右帧图像I_{k_r}的图像平面(u',v')，完成重投影，如图2-4所示。

（3） 迭代优化更新 3D 点的深度 。按理来说对于空间中同一个点，被极短时间内的相邻两帧拍到，它的亮度值应该没啥变化。但由于深度是假设的一个值，所以重投影的点不准确，导致投影前后的亮度值是不相等的。不断优化位姿使得残差最小，就能得到优化后的深度信息d。

将上述基于半直接法的双目匹配过程公式化如下：通过不断优化空间点深度d最小化残差损失函数。

$$d=\arg\min_{d}\frac{1}{2}\sum_{i\in\Re}\|\delta I(T_{k_l,k_r},u_i)\|^2 \tag{2-12}$$

其中：

$$\delta I(T_{k_l,k_r},u_i)=I_{k_r}\left(\pi(T_{k_l,k_r}\cdot\pi^{-1}(u,d_u))\right)-I_{k_l}(u) \tag{2-13}$$

结合半直接法对空间点的深度d做优化，迭代求解。从估计的摄像机姿态可见的 3D 点被投影到图像中，得到对应 2D 特征位置的估计原理，如图 2-4 所示。

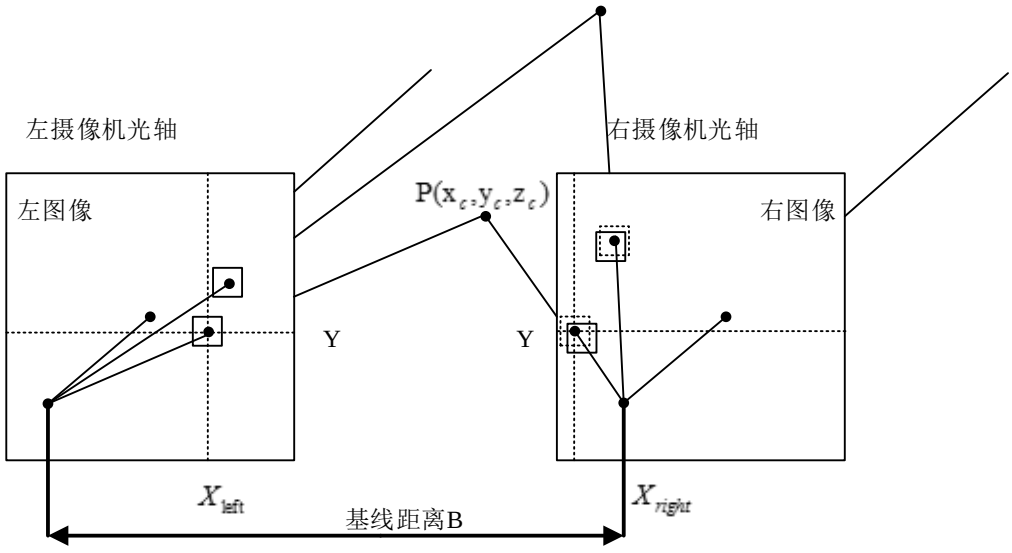

图 2-4 3D 点被投影到图像中对应 2D 点特征位置的估计原理图

该基于半直接法的双目定位系统前端的流程图，如 2-5 所示。

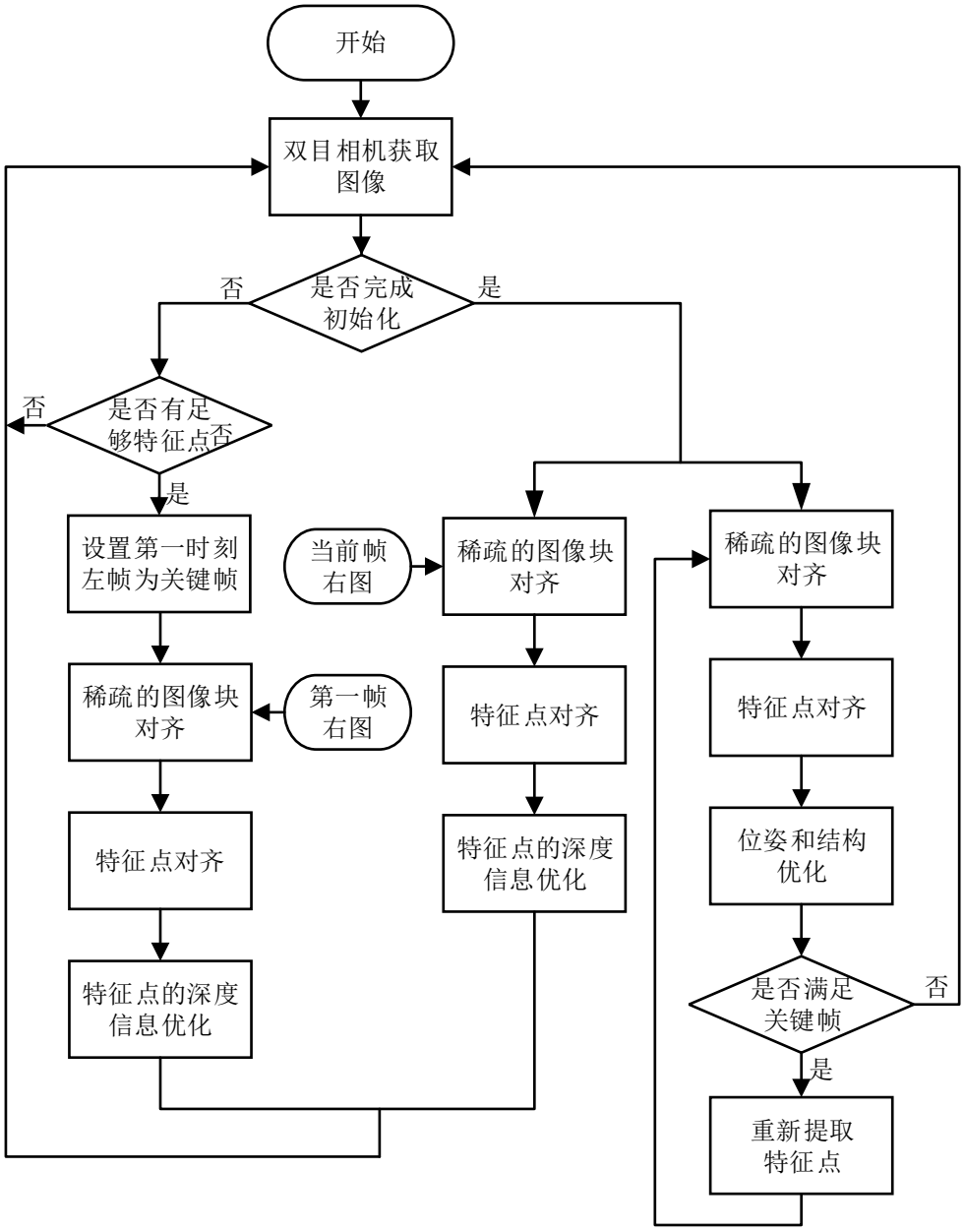

图 2-5 基于半直接法的双目定位系统前端的流程图

2.3.3 实验结果分析

该实验针对基于特征点法和基于直接法两种方法各自的优缺点，选取这两种方法典型的 SLAM 系统在处理器为 Inter(R) Core(TM) i5-2450M CPU @ 2.50GHz 的笔记本电脑上进行测试。该实验采用连接到 MAV 的下视摄像机记录的视频进行处理。该视频中 MAV 的飞行路径长 84m，平均在地面以上 1.2m 处飞行。同时该系统采用 ROS 内部的 RViz 实时观测视频中无人机在飞行过程中的路径以及相机姿态，并建立稀疏地图，为后续对定位信息的应用提供条件。实验运行界面如图 2-6 所示。

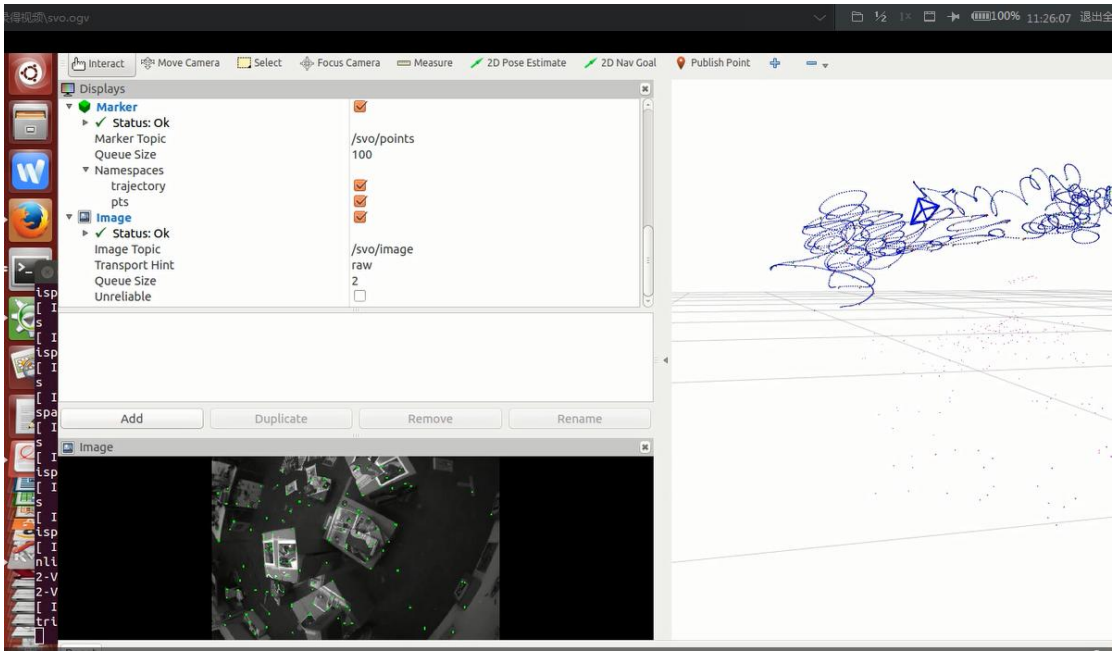

图 2-6 实验运行界面

本实验选取基于特征点法的 ORB-SLAM 系统、基于半直接法的 SVO 系统在笔记本电脑上的实验运行，同时针对基于半直接法的 SVO 系统采用两种不同的参数设置，一个针对速度优化，一个针对精度优化（表 2-1）。当实际应用在嵌入式平台上时，仅使用快速参数设置。

表 2-1 实验测试系统参数设置

	双目 SVO		ORB-SLAM
	速度优化 Fast	精度优化 Accurate	
每帧图像提取特征点	120	200	1200
最多的关键帧数	10	50	/
开启局部 BA 优化	否	是	是

该实验对比基于两种参数设置方式的 SVO 与单目的 ORB-SLAM 系统。从实时性和准确性两个方面做对比。

（1） 运行时间评估

图 2-7 示出了分别利用两种参数设置的 SVO 以及 ORB-SLAM 系统来计算在笔记本电脑上的相机运动所需的时间对比。表 2-2 示出利用两种参数设置的 SVO 以及 ORB-SLAM 系统处理每帧图像所用时间以及每秒的处理速度。

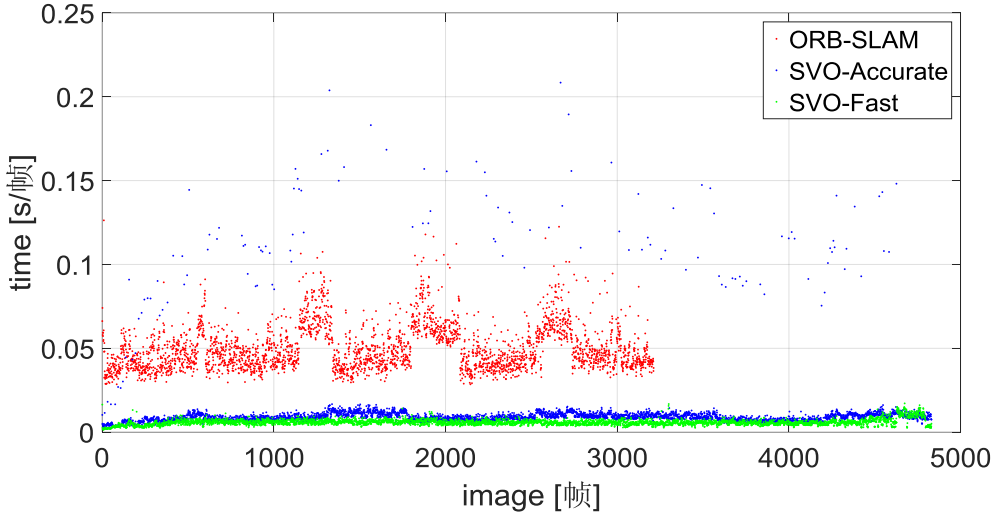

图 2-7 两种参数设置的 SVO 及 ORB-SLAM 系统运行时间对比

表 2-2 两种参数设置的 SVO 及 ORB-SLAM 系统处理每帧的时间均值

	SVO-Fast	SVO-Accurate	ORB-SLAM
处理每帧的时间均值（帧/s）	0.00585	0.011500	0. 04994
每秒处理的图像帧数（fps）	140	80	20

采用笔记本电脑测试系统，图中红色的点代表 ORB-SLAM 系统处理每帧图像用的时间，相应时间为 20fps。蓝色的点代表 SVO 系统在采用精度优化参数设置情况下处理每帧图像所用的时间，能够以大约每秒 100 帧（fps）的速度处理帧。绿色的点代表 SVO 系统在采用速度优化参数设置情况下处理每帧图像所用的时间，能够以大约每秒 170 帧（fps）的速度处理帧。主要区别是基于半直接法的 SVO 不需要在运动估计期间提取特征，其构成 ORB-SLAM 的大部分时间（笔记本电脑上的 30ms）。从图中明显看出基于半直接法的 SVO 系统在处理时间方面明显优于基于特征点法的 ORB-SLAM 系统。同时速度优化参数设置的 SVO 系统在一定程度上提升了实时性。在我们应用于嵌入式操作平台时，选择采用速度优化的参数设置方式。

我们可以用较少的特征可靠地跟踪摄像机的原因是使用深度滤波器，这确保了被跟踪的特征是可靠的。运动估计准确的参数设置在笔记本电脑上平均需要 17ms。运行时间的增加主要是由于局部 BA 优化，其在每个关键帧运行并且花费 18ms。建图线程用新帧来更新所有深度过滤器所需的时间高度依赖于过滤器的数量。在选择关键帧之后，滤波器的数目较高，并且在滤波器收敛时迅速减小。平均来说，建图线程比运动估计线程更快，因此它不是限制因素。

（2） 准确性

该实验采用的视频中无人机飞行的真实轨迹通过动捕系统获得，飞行路经长84m，MAV 平均在地面以上 1.2m 处。图 2-8 示出了随时间变化的位置信息的真实路径、两种参数设置的基于半直接法 SVO 系统以及基于特征点法的 ORB-SLAM 系统得到的 MAV 的飞行轨迹。

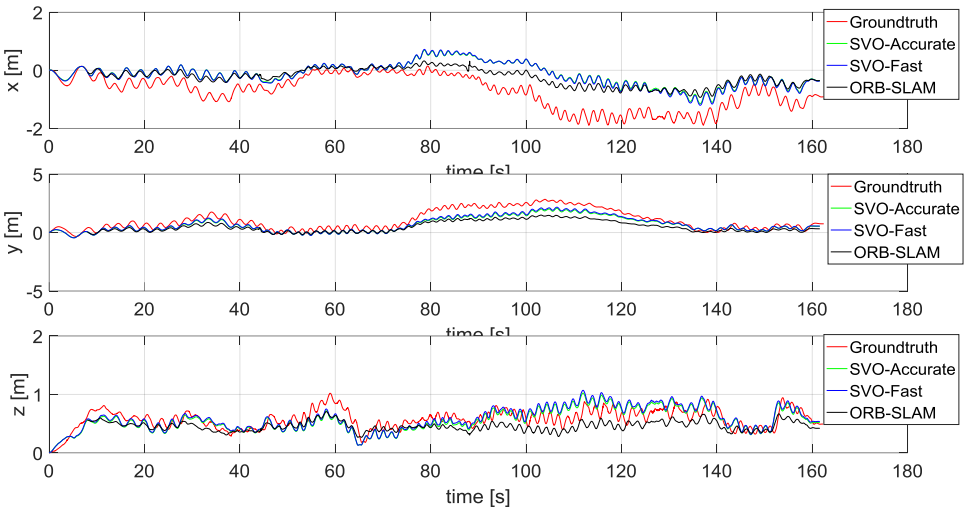

图 2-8 两种参数设置的 SVO 及 ORB-SLAM 系统定位精度对比

总体而言，SVO 的两个版本比 ORB-SLAM 系统更准确。从图 2-7 中可以看出，速度和精度的参数设置之间的精度差异并不显著。在每次迭代（速度参数设置）时分别优化姿态和观察到的 3D 点的位置信息，对于 MAV 运动估计是相对准确的。

同时，该指标测试时，采用 RGBD-SLAM 系统中提出的概念，相对姿态误差（Relative pose error，RPE），RPE 表示在固定时间间隔 Δt 上测量轨迹的局部精度。相对姿态误差对应于轨迹的漂移，其对于评价视觉定位系统特别有用。一般采用从所有时刻的相对姿态误差，计算所有时间内其平移分量的均方根误差（RMSE），本实验采用该方法，计算以 m/s 为单位的位姿平移分量漂移的均方根误差以及位姿平移分量漂移的均值，如表 2-3 所示。

表 2-3 两种参数设置的 SVO 及 ORB-SLAM 系统定位误差

	位置的均方根误差[m/s]	位置误差的中值[m/s]
SVO-Fast	0.0059	0.0047
SVO-Accurate	0.0051	0.0038
ORB-SLAM	0.0164	0.0142

3．后期拟完成的研究工作及进度安排

针对目前已完成的研究工作，接下来首先对现阶段实现的整个跟踪系统进行简化，完善在实现中带来的不足。当摄像头水平朝前运动的时候，基于半直接法

定位中的深度滤波做的不好，在此基础上添加深度滤波功能，验证深度滤波对提升 SLAM 系统精度的影响。其次在对整个跟踪系统搭建局部地图扩张以及优化的框架，优化相机位姿以及创建的地图，再添加该优化框架的基础上，实验验证对系统精度的提升程度，验证其可行性。最后整个 SLAM 系统应用到四旋翼无人机上，再一步验证该视觉 SLAM 系统实时性的实时性以及可行性。最后完成硕士论文。具体进度安排如下：

2017.03.18—2017.03.24	对现有的 SLAM 前端（跟踪系统）获得的深度信息进行滤波处理，提高深度信息的精度。
2017.03.25—2017.04.20	设计整个跟踪系统的局部地图扩张以及优化的框架；实现整个基于半直接法定位以及根据局部地图扩张进行地图优化的双目 SLAM 系统。
2017.04.20—2017.05.10	完成整个系统的实验平台搭建，对代码进行平台移植及优化，进行试验验证，达到实时性、精度指标要求。
2017.05.10—2017.05.31	撰写毕业论文。
2017.06	修改毕业论文，准备论文答辩。

4．存在的困难与问题

目前存在的问题主要有以下三个方面。

（1） 目前设计的基于半直接法的双目视觉定位前端系统在应用中，初始化时刻需尽量满足是平面模型，对于平视的双目摄像头，初始化时刻获取的空间点不满足要求，初始化成功率较低，需要更深入得理解算法并加以改进。

（2） 在测试过程中发现修改关键帧的判断阈值，使得关键帧增加速度加快。这么做有利于特征点数量的增加，也有利于重定位时最近关键帧和当前帧的匹配。但是同时再此基础上测试估计的深度值变换较大，为什么在特征点增加的情况下深度估计准确度却下降这个问题还需要解决。

（3） 代码实现的功能验证过程需要掌握机器人操作系统(ROS)，由于在嵌入式平台算法移植基础薄弱，很多知识需要从头学习，认真学习机器人操作系统的使用方法，解决系统不兼容等问题，需要耐心调试，多查阅书籍和资料。上网查询，向老师、师兄请教等，积极主动的解决问题。

5．如期完成全部论文工作的可能性

能够如期完成全部论文工作。