# 基于多摄像头的 VSLAM 导航方案
# 项目进展报告

刘沛东
计算机视觉实验室
西湖大学
2023 年 1 月 17 日

该份报告将详细汇报我们在 ‘‘基于多摄像头的 VSLAM 导航方案’’ 项目中，截止至 2023 年 1 月 17 日的项目进度，该报告将详细描述我们在针对多摄像头视觉惯性里程计算法中所采用的方案和初步结果。

## 1. 视觉惯性里程计

在本章节中，我们将详细介绍我们开发的视觉惯性里程计（Visual-Inertial Odometry, VIO）算法和系统。其中：第 1.1节简要介绍符号惯例与约定；第 1.2节详细介绍我们的 VIO 系统框架与工作原理；第 1.3节简要汇报当前算法的性能表现。

### 1.1. 符号惯例与约定

**符号惯例.** 我们使用小写字母（如：$\lambda$) 表示标量, 粗体小写字母（如：$\mathbf{v}$) 表示标量数组, 粗体大写字母（如：$\mathbf{T}$) 表示矩阵，坐标系 $x$ 表示为 $\mathcal{F}_x$。符号 $\mathbf{T}_{B_k}^{W}$ 表示在时间戳 $k$ 将向量从帧 $\mathcal{F}_B$ 转换到帧 $\mathcal{F}_W$ 的转换矩阵。符号 $\mathbf{p}^{W_k}$ 表示在时间戳 $k$ 处向量 $\mathbf{p}$ 处在帧 $\mathcal{F}_W$ 中的值。符号 $\mathbf{b}_k$ 表示在时间戳 $k$ 处向量 $\mathbf{b}$ 的值。

**坐标系.** 在整个算法推导过程中主要使用了三个坐标系。它们分别是一个局部世界坐标系 $\mathcal{F}_W$ ，一个车身坐标系 $\mathcal{F}_B$ 和针对每一个相机 $C_i$ 的相机坐标系 $\mathcal{F}_{C_i}$。

**相机模型.** 我们使用统一相机模型 [8] 对项目中使用的广角相机进行建模。

遵循 [3] 中使用的成像模型：

$$\mathbf{I}_i(\mathbf{u}) = G(t_i V(\mathbf{u}) \xi_i(\mathbf{u})) \ . \tag{1}$$

上述公式中，$\mathbf{I}_i(\mathbf{u})$ 表示在帧 $i$ 中像素位置 $\mathbf{u}$ 的像素强度，$G(\cdot)$ 表示非线性响应函数，$t_i$ 表示曝光时间，$V(\mathbf{u})$ 表示镜头衰减函数（渐晕），$\xi_i(\mathbf{u})$ 表示潜在场景辐照度。$G(\cdot)$ 和 $V(\cdot)$ 是通过离线标定获得的。因此，我们可以得校正后的像素强度为

$$\hat{\mathbf{I}}_i(\mathbf{u}) = t_i \xi_i(\mathbf{u}) = \frac{G^{-1}(\mathbf{I}_i(\mathbf{u}))}{V(\mathbf{u})} \ . \tag{2}$$

该模型假设场景辐照度 $\xi_i(\mathbf{u})$ 是常数。然而，在实践中，$\xi_i(\mathbf{u})$ 可能会在图像之间略有变化。遵循 [11]，我们将其建模为

$$\xi_i(\mathbf{u}) = \hat{a}_i \hat{\xi}_i(\mathbf{u}) + \hat{b}_i \ , \tag{3}$$

其中 $\hat{\xi}_i(\mathbf{u})$ 是真实的 (恒定的) 场景辐照度。因此，校正后的像素强度变为

$$\hat{\mathbf{I}}_i(\mathbf{u}) = t_i \xi_i(\mathbf{u}) = t_i \hat{a}_i \hat{\xi}_i(\mathbf{u}) + t_i \hat{b}_i = a_i \hat{\xi}_i(\mathbf{u}) + b_i \ , \tag{4}$$

其中 $a_i$ 和 $b_i$ 是未知模型参数。简单起见，我们将此模型表示为

$$\mathbf{I}_i(\mathbf{u}) = a_i \xi_i(\mathbf{u}) + b_i \tag{5}$$

在后面的章节中。$a_i$ 可以初始化为 $t_i$，而 $b_i$ 可以简单地初始化为 0。

**车辆状态.** 我们通过其位置、速度、方向、加速度计和陀螺仪的偏差以及每个摄像机的照明参数来描述时间戳 $k$ 处的车辆状态 $\mathcal{S}_k$。位置和速度分别用 $\mathbf{p}^{W_k}$ 和 $\mathbf{v}^{W_k}$ 表示。方向由旋转矩阵 $\mathbf{R}_{B_k}^{W} \in \mathrm{SO}(3)$ 表示。加速度计和陀螺仪的离散时间偏差分别用 $\mathbf{b}_{ad_k}$ 和 $\mathbf{b}_{gd_k}$ 表示。相机 $C_j$ 的光照参数由 $a_k^{C_j}$ 和 $b_k^{C_j}$ 表示。我们得到在时间戳 $k$ 处从车身坐标系 $\mathcal{F}_B$ 到世界坐标系 $\mathcal{F}_W$ 的齐次变换矩阵 $\mathbf{T}_{B_k}^{W} \in \mathrm{SE}(3)$ 为

$$\mathbf{T}_{B_k}^{W} = \begin{bmatrix} \mathbf{R}_{B_k}^{W} & \mathbf{p}^{W_k} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} . \tag{6}$$

我们将时间戳 $k$ 处的线性加速度计测量值表示为 $\hat{\mathbf{a}}^{B_k}$。同样，时间戳 $k$ 处的陀螺仪测量值表示为 $\hat{\boldsymbol{\omega}}^{B_k}$。分别使用 $\mathbf{a}^{B_k}$ 和 $\boldsymbol{\omega}^{B_k}$ 表示真实的线性加速度计和陀螺仪测量值，该测量值是无噪声和无偏差的。重力加速度用 $\mathbf{g}$ 表示。相机 $C_j$ 在时间戳 $k$ 捕获的图像表示为 $\mathbf{I}_k^{C_j}$。之后进一步将在图像 $\mathbf{I}_k^{C_j}$ 中像素坐标 $\mathbf{u}$ 处的像素强度表示为 $\mathbf{I}_k^{C_j}(\mathbf{u})$，这个像素强度是灰度图像的标量。为简单起见，最新关键帧捕获的图像表示为 $\mathbf{I}_{KF}^{C_j}$。类似地，该图像中像素坐标 $\mathbf{u}$ 处的像素强度表示为 $\mathbf{I}_{KF}^{C_j}(\mathbf{u})$。

### 1.2. 多相机系统的直接稀疏惯性里程计

在本节中，我们将介绍用于多摄像头系统的 VO/VIO 算法。我们遵循 [3]，它描述了一个单目相机的 VO 框架，该框架使用的直接对齐算法基于最小化稀疏像素集的光度成本函数。我们将公式进行扩展，使其支持多相机系统，该多相机系统在算法中建模为通用相机，并且集成了来自惯性测量单元（IMU）的测量值。该算法设计灵活，支持各种硬件配置。特别的，该算法可以处理单目 VO、单目 VIO、立体 VO、立体 VIO 和多相机 VO (VIO) 等设置。正如 图 1 所示，我们的算法由两个线程组成，一个跟踪器（tracker）和一个局部定位器（local mapper）。跟踪器将当前图像与最新关键帧进行对齐，从而实现实时的车辆位姿估计。局部定位器主要通过优化车辆位姿和估计的 3D 点云来减少漂移。如果使用惯性测量单元的测量值，局部定位器还会联合优化车辆速度和 IMU 偏差。其采用了计算密集型批量优化方法来进行姿态和结构优化。因此，与跟踪器相比，它以低得多的帧速率运行，并且只处理关键帧。对于新的关键帧，它会使用多视图立体算法来进行 3D 点云的初始化。算法描述如下。

#### 1.2.1 硬件配置

为了使我们的算法能够适配不同的硬件配置，我们假设一共有 $N$ 个相机。对于每个相机，它可以配置为用于运动跟踪的参考相机，或者配置为用于静态双目匹配的辅助相机。一个参考相机可以没有辅助相机，在这种情况下，运动双目算法将会用于双目匹配过程。为了便于算法演示，我们将相机 $C_i$ 作为相机 ${}^{r}C_i$ 的参考相机。不失一般性地，我们假设我们有 $N_r$ 个参考相机和 $N_a$ 个辅助相机，其中 $N_r + N_a = N$。

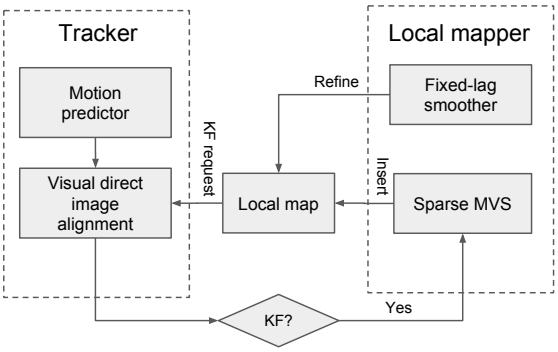


Figure 1: VIO 流程示意图

#### 1.2.2 跟踪器

正如 图 1 所示，跟踪器由两个部分组成：一个运动预测器和一个直接图像对齐模块。直接图像对齐模块用于估计当前车辆相对于最新关键帧的位姿。我们使用运动预测器提供的车辆位姿来初始化对齐模块以避免局部最小值。算法将根据惯性测量数据的可用性，考虑使用恒速运动模型或者使用基于惯性的运动模型。两个模型的详细信息描述如下。

**恒速运动预测模型：** 如果没有可用的惯性测量数据或者测量数据噪声较大（如：由于车辆发动机的振动），我们将使用恒速运动模型进行当前车辆位姿的预测。我们将当前车辆位姿定义为 $\mathbf{T}_{B_k}^{W}$，这是从时间戳 $k$ 处的车辆坐标系 $\mathcal{F}_B$ 到世界坐标系 $cF_W$ 的转换矩阵。时间戳 $k$ 对应于捕获的图像时间戳。类似地，我们将最近两帧的车辆位姿分别定义为 $\mathbf{T}_{B_{k-1}}^{W}$ 和 $\mathbf{T}_{B_{k-2}}^{W}$。如果我们假设从时间戳 $k-2$ 到 $k$ 的运动速度是恒定的，我们可以预测当前车辆位姿为

$$\mathbf{T}_{B_k}^{W} = \mathbf{T}_{B_{k-1}}^{W}\mathbf{T}_{B_k}^{B_{k-1}} = \mathbf{T}_{B_{k-1}}^{W}\mathbf{T}_{B_{k-1}}^{B_{k-2}} , \tag{7}$$

其中

$$\mathbf{T}_{B_{k-1}}^{B_{k-2}} = (\mathbf{T}_{B_{k-2}}^{W})^{-1}\mathbf{T}_{B_{k-1}}^{W}. \tag{8}$$

**基于惯性的运动预测模型：** 恒速运动模型的缺点是它不适用于车辆运动状态突然变化和非恒速运动的情况。

与视觉测量（∼30 Hz）相比，惯性测量通常以更高的频率（∼1000 Hz）进行采样。因此，惯性测量值可以更好地捕捉车辆运动状态。如果惯性测量值可用并且它们的噪声水平足够低以提供良好的运动预测，我们使用基于惯性的运动模型进行运动预测。

该模型基于质点的运动学，使用惯性测量预测当前车辆位姿 $\mathbf{T}_{B_k}^{W}$。特别的，我们使用最新关键帧的运动状态 $\mathcal{S}$ 和两者之间的惯性测量来进行运动预测。用于流形 IMU 测量预积分 [4] 的离散 IMU 运动学模型也用于该运动传播过程。

**稀疏直接跟踪器：** 考虑到运动预测器已经提供了初始位姿估计，我们使用一个稀疏直接跟踪器来获得更加精确的估计。与之后将解释的局部定位器不同，该跟踪器只使用参考相机的图像进行运动跟踪。特别的，我们通过最小化以下能量函数来估计最新关键帧和当前帧之间的相对车辆姿态 $\hat{\mathbf{T}}_{B_{KF}}^{B_k}$

$$\hat{\mathbf{T}}_{B_{KF}}^{B_k} = \underset{\mathbf{T}_{B_{KF}}^{B_k}}{\operatorname{argmin}} \sum_{i=1}^{N_r} \sum_{\mathbf{u}\in\boldsymbol{\Omega}(\mathbf{I}_{KF}^{{}^rC_i})} (\mathbf{I}_{KF}^{{}^rC_i}(\mathbf{u}) - \mathbf{I}_{k}^{{}^rC_i}(\hat{\mathbf{u}}))^2 \ . \tag{9}$$

此处，$N_r$ 是参考相机的个数，$\boldsymbol{\Omega}(\mathbf{I}_{KF}^{{}^rC_i})$ 是从参考相机 ${}^rC_i$ 的关键帧图像中采样的特征像素点集。对于关键帧图像中的每个特征点 $\mathbf{u}$，$\hat{\mathbf{u}}$ 是其在当前帧中对应的像素位置，表示为

$$\hat{\mathbf{u}} = \pi(\mathbf{T}_B^C \cdot \mathbf{T}_{B_{KF}}^{B_k} \cdot (\mathbf{T}_B^C)^{-1} \cdot \pi^{-1}(\mathbf{u}, d)) \ . \tag{10}$$

$\mathbf{T}_B^C$ 表示从车身坐标系到相机坐标系的转换矩阵。$\pi$ 和 $\pi^{-1}$ 分别表示相机投影函数和反投影函数。$d$ 表示关键帧中特征点 $\mathbf{u}$ 的深度。上述等式中唯一的未知变量是 $\mathbf{T}_{B_{KF}}^{B_k}$，因为 $\mathbf{T}_B^C$ 可以通过离线传感器校准获得，而 $d$ 则可以通过多视图双目初始化然后再由局部定位器进一步优化得到。上述公式遵循标准的正向组合方法 [1]，该方法要求在优化的每次迭代中计算残差的 Hessian 矩阵和 Jacobian 矩阵。为了提高效率，我们因此采用逆合成方法 [1] 来最小化能量函数。同时，为了使对异常值较敏感的最小二乘估计量更稳定，我们进一步对所有残差应用鲁棒的损失函数（Huber 损失）。

**异常值去除：** 除了使用鲁棒的损失函数外，我们还使用了特定的异常值去除机制来增强跟踪器和局部定位器的稳定性。我们根据参考帧和当前帧之间的模板匹配分数检测异常值，其中采用的分数为零均值归一化互相关 (ZNCC) 得分 [10]。在分数小于预定义的阈值时，我们认为该特征匹配异常并将其从优化中移除。由于跟踪器之前已经使用了特征点周围的图像块，并且在直接图像对齐期间已经计算了扭曲的图像块。因此，异常值去除步骤的计算开销很小。

### 1.2.3 关键帧选择和特征提取

**关键帧选取：** 我们定义了两个标准用于关键帧的选取。运动跟踪背后的隐含假设之一是参考关键帧和当前帧之间的场景差异足够小。反之，如果当前帧和参考关键帧之间的差异变得太大，我们便创建一个新的关键帧。直观上，帧之间的差异应该基于图像内容的变化而不是基于绝对位姿差异来衡量（因为前者强烈依赖于场景的深度，而后者则忽略它）。因此，我们使用均方光流作为检测场景变化的标准之一。准确来说，我们计算

$$f = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|_2 \ , \tag{11}$$

其中 $\|\cdot\|_2$ 表示欧式范数运算，$\mathbf{u}_i$ 是参考关键帧中的一个特征点，$\hat{\mathbf{u}}_i$ 是它在当前帧中对应的特征点。如果 $f$ 高于阈值，我们将创建一个新的关键帧。

如果局部定位器使用了惯性测量值或者算法使用惯性运动预测模型，我们还根据自最新关键帧以来经过的时间进行关键帧选择。原因是经过的时间较长会导致相邻关键帧之间的惯性约束不可靠，这种约束将提供不充足的信息或不可靠的运动预测。

**特征提取：** 一旦创建了新的关键帧，我们便对关键帧中参考相机捕获的图像进行稀疏特征采样。我们根据它们的梯度大小从每张图像中均匀地采样 $N$ 个稀疏特征。特别的，采样后我们将图像进行网格划分，并从每个网格中选择梯度值最高的点。如果网格中特征点的最高梯度值小于预定义的阈值，则不从该网格中选取特征点。

**特征表示和深度初始化：** 正如 [3] 实现的那样，我们通过对每个稀疏特征预定义一个样式块 (如：$5 \times 5$ 样式) 来进行所有像素的采样。所有的这些像素都用于与运动跟踪和局部定位。特别的，从跟踪器和局部定位器创建的视觉残差块是针对每个像素而不是每个块的。进一步来说，我们假定来自同一块的像素位于同一 3D 平面上，而该平面可以通过其逆平面的深度和平面法线

进行参数化，其中逆平面深度和平面法线由双目匹配算法初始化。局部定位器会在联合优化期间对采样块的平面深度进行优化，而非优化每个像素的光线深度。算法通过这种方式减少了要优化的变量数量并提高了优化器的计算效率。

**稀疏 MVS:** 双目匹配用于初始化从新关键帧采样的每个特征的深度。我们没有使用沿着基线进行视差搜索的方式进行深度计算，而是使用了平面双目扫描 [5]。这种方式使我们能够直接对鱼眼图像进行操作，从而避免对图像进行反失真和校正而造成视野损失。平面双目扫描还支持多视图双目匹配，例如，使用多基线双目扫描 [6] 以获得更准确的深度估计。

基于 [5] 的实现，我们使用 GPU 来加速平面双目扫描。我们在两个方向上扫描平面：正面平行于关键帧的观察方向和平行于地平面方向。对于每个方向，我们生成 64 个假设平面，它们之间的视差步长大小恒定。这些平面覆盖了相机前面的 $[0.5,\ 30]$ m 的范围。因此一共有 128 个假设平面被评估了。我们计算来自 $\mathbf{I}^{C_i}$ 的 $7 \times 7$ 图像块和来自 $\mathbf{I}^{C_j}$ 的变形图像块之间的 ZNCC 分数。最后选择模板匹配得分最大的假设平面作为采样特征点所在的平面。

### 1.2.4 局部定位器

为了最小化位姿漂移，局部定位器联合优化所有车辆状态参数和 3D 场景几何。显然，随着车辆继续移动，状态总数会随着时间的推移而增加。为了限制优化程序的运行时间，我们使用状态边缘化方式删除旧状态。我们只保留固定数量的先前状态，这使我们算法在运行过程中维持固定的计算复杂度 [2]。在 [9] 提到，边缘化状态将导致所有与其连接的剩余状态在边缘化后相互连接，这使得对应的 Hessian 矩阵不再稀疏。不稀疏的 Hessian 矩阵反过来又增加了在结构优化期间计算 Schur 补码的时间成本。为了提高效率，我们遵循 [3,7] 并且只填充不涉及几何项的 Hessian 项。通过这种方式实现了 “部分边缘化”。我们还通过只对选定的关键帧进行优化来进一步提高效率。

特别的，我们定义了一个包含 $k$ 个车辆状态的滑动窗口。此滑动窗口之外的所有状态都被视为旧状态，并通过部分边缘化删除。如 [4] 中所述，我们通过流形预积分对两个相邻关键帧之间的惯性测量值进行预积分。我们使用 $\mathcal{S}_*$ 来表示滑动窗口内所有车辆状态的集合，且它们以向量形式表示。局部定位器估计滑动窗口内所有的车辆状态 $\mathcal{S}_*$ 以及所有特征采样的逆平面深度 $\mathbf{D}_*$，计算公式为

$$\hat{\mathcal{S}}_*, \hat{\mathbf{D}}_* = \underset{\mathcal{S}_*, \mathbf{D}_*}{\operatorname{argmin}} \ \mathrm{E}_0(\mathcal{S}_*) + \mathrm{E}_{imu}(\mathcal{S}_*) + \mathrm{E}_{vision}(\mathcal{S}_*, \mathbf{D}_*). \tag{12}$$

此处，$\mathrm{E}_0(\mathcal{S}_*)$ 是从部分边缘化获得的先验能量项或初始先验项，$\mathrm{E}_{imu}(\mathcal{S}_*)$ 是相邻关键帧之间的惯性能量项，$\mathrm{E}_{vision}(\mathcal{S}_*, \mathbf{D}_*)$ 是滑动窗口内所有关键帧之间的视觉能量项，使用高斯-牛顿算法进行优化。下文中将更加详细的描述各个符号。

公式中有两种**先验项**。一个是来自最初的先验。为了解决视觉里程计问题中的自由度不可观测问题，我们需要固定初始状态，以便相对于它估计所有后续状态。因此，它们通常被表述为以下形式

$$\mathrm{E}_0(\mathcal{S}_0) = \frac{1}{2} \left\| \hat{\mathcal{S}}_0 - \mathcal{S}_0 \right\|_{\mathbf{\Sigma}_0} , \tag{13}$$

其中 $\hat{\mathcal{S}}_0$ 表示初始状态的值，$\mathcal{S}_0$ 是要估计的初始状态变量，$\Sigma_0$ 是初始状态的协方差矩阵。为了使 $\mathcal{S}_0$ 等于 $\hat{\mathcal{S}}_0$，$\mathbf{\Sigma}_0$ 通常选择具有无穷小对角线项的矩阵。当初始状态在滑动窗口内时，该先验项仅出现在能量函数中，一旦它离开窗口，它将与所有其他能量项一样被边缘化 [2,9]。

另一个先验项是部分边缘化的直接结果。部分边缘化将先验引入所有与边缘化状态相关的剩余状态 [2,9]。被删除的状态的信息存储在先验 Hessian 和 Jacobian 矩阵中，防止了删除状态导致的信息大量丢失。因此，我们可以得到剩余状态的先验能量项如下

$$\mathrm{E}_0(\mathcal{S}_*) = \mathbf{J}_m^T(\hat{\mathcal{S}}_{*0} - \mathcal{S}_*) + \frac{1}{2} \left\| \hat{\mathcal{S}}_{*0} - \mathcal{S}_* \right\|_{{\mathbf{H}_m}^{-1}} , \tag{14}$$

其中 $\hat{\mathcal{S}}_{*0}$ 是一组先前迭代估计的 $\mathcal{S}_*$ 的先验车辆状态，$\mathbf{J}_m$ 和 $\mathbf{H}_m$ 分别是 Jacobian 和 Hessian 矩阵，从部分边缘化中累积得到。

我们使用 **惯性能量项**对两个连续关键帧之间的惯性测量进行建模。术语的制定基于 [4] 中描述的方法：

$$\mathrm{E}_{imu}(\mathcal{S}_*) = \sum_{i=1}^{k-1} \| \mathbf{r}(\mathcal{S}_i, \mathcal{S}_{i+1}) \|_{\mathbf{\Sigma}_{imu_{i,i+1}}} . \tag{15}$$

其中，$\mathbf{r}(*)$ 是包含车辆位姿，车辆速度和 IMU 偏移值的残差的两个连续关键帧之间的残差向量，$\mathbf{\Sigma}_{imu_{i,i+1}}$

是两帧间 IMU 测量值的协方差矩阵。可以从 [4] 获取详细的推导过程。

**视觉能量项**包含所有光度误差残差的总和，被建模为

$$\mathrm{E}_{vision}(\mathcal{S}_*, \mathbf{D}_*) = \sum_{m=1}^{k}\sum_{i=1}^{N_r}\sum_{\mathbf{u}\in\mathbf{\Omega}(\mathbf{I}_m^{{}^rC_i})}\sum_{n\in\mathbf{\Theta}(\mathbf{u})} \sum_{j\in\mathbf{\Phi}(\mathcal{S}_m,\mathbf{u})} \|r(\mathcal{S}_m, {}^rC_i, \mathcal{S}_n, C_j, \rho)\|_\Sigma . \tag{16}$$

其中，$k$ 仍然是关键帧的个数，$N_r$ 是参考相机的个数。${}^rC_i$ 和 $C_j$ 分别指第 $i$ 和第 $j$ 个相机。$\mathbf{\Omega}(\mathbf{I}_m^{{}^rC_i})$ 指从第 $m$ 个关键帧中的第 $i$ 个参考相机图像中采样的特征点集。$\mathbf{\Theta}(\mathbf{u})$ 是观察到特征点 $\mathbf{u}$ 的车辆状态的索引集。$\mathbf{\Phi}(\mathcal{S}_m, \mathbf{u})$ 是观察到特征点 $\mathbf{u}$ 的第 $m$ 个关键帧中相机的索引集。$r(\mathcal{S}_m, {}^rC_i, \mathcal{S}_n, C_j, \rho)$ 是辐照度残差，用于测量同一 3D 场景点的两个投影之间的（归一化）强度差异（参见下面的残差定义）。此外，我们在实现中同时使用相机间和相机内的辐照度残差。最后，$\Sigma$ 是从相机光度校准中获取的光度误差残差的标量方差。

设 $\mathbf{I}_m^{{}^rC_i}$ 为第 $i$ 个相机在第 $m$ 个关键帧中拍摄的图像。考虑从该图像中采样的特征点 $\mathbf{u}$。使用平面双目扫描，我们可以得到其对应的平面逆深度 $\rho$ 以及该平面的法线 $\mathbf{n}$。因此，对应于 $\mathbf{u}$ 的 3D 点 $\mathbf{P_u}$ 由下式给出

$$\mathbf{P_u} = \frac{1}{\rho\cos\theta}\pi_{{}^rC_i}^{-1}(\mathbf{u}) \ , \tag{17}$$

其中 $\pi_{{}^rC_i}^{-1}$ 是相机的逆投影函数，$\cos\theta = -\mathbf{n}^T \cdot \pi_{{}^rC_i}^{-1}(\mathbf{u})$ 是 $\mathbf{u}$ 对应的视线与平面法线的夹角。车辆在时间戳 $m$ 和 $n$ 处的位姿 $\mathbf{T}_{B_m}^W$ 和 $\mathbf{T}_{B_n}^W$ 由跟踪器初始估计，而车辆坐标系 $\mathcal{F}_B$ 到相机坐标系 $\mathcal{F}_{C_l}$ 之间的相对变换 $\mathbf{T}_B^{C_l}$ 可以离线估计。我们使用这些转换，通过将 $\mathbf{P_u}$ 投影到图片中来计算在第 $n$ 个关键帧中的第 $j$ 个相机的像素 $\hat{\mathbf{u}}$ 对应的特征 $\mathbf{u}$。给定 $\mathbf{u}$ 和 $\hat{\mathbf{u}}$，我们使用它们在辐照度上的差异 (参见：1.1节) 来定义残差如下

$$r(\mathcal{S}_m, {}^rC_i, \mathcal{S}_n, C_j, \rho) = \xi_m^{{}^rC_i}(\mathbf{p}) - \xi_n^{C_j}(\hat{\mathbf{p}}) \ . \tag{18}$$

为了最小化公式（16）的视觉能量项，我们调整车辆和相机参数 (即位姿 $\mathbf{T}_{B_m}^W$ 以及对图像形成进行建模的每个相机的参数 $a_i$ 和 $b_i$) 以及每个相机的逆平面深度。我们不考虑每个特征 $\mathbf{u}$ 的单个像素，而是考虑每个特征的像素块。块中的每个像素都被扭曲到其他图像中，并根据公式（18）贡献一个残差。在这种设置下，基于平面参数化 $\mathbf{u}$ 的深度使我们只需为每个块设定一个参数。

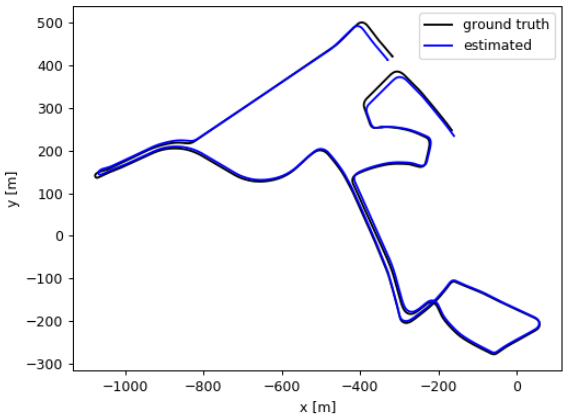


Figure 2: 视觉里程计估计出的轨迹与 RTK 提供的真值轨迹对比图.

我们当前的算法仅包括直接对齐残差。将来，我们还将添加来自 2D-3D 特征匹配的重投影误差，以便将视觉定位合并到 VIO 流程中。

### 1.3. 实验结果

由于我们的硬件系统还在搭建过程中，我们利用一个已有的数据集对我们的算法进行了初步测试，该数据集为一个多相机的数据集，是在城市环境下采集所得，我们利用该数据集配置了四对双目相机，分别分布于车的前后左右四个方向，整个数据的路径长度大概在 3893.3 米，我们开发的多目相机视觉里程计实现了 0.22% 的漂移，即每 100 米累计错误误差在 0.22 米左右，具体的结果如图 2 所示。

## References


[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A Unifying framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004.

[2] T.-C. Dong-Si and A. I. Mourikis. Motion tracking with fixed-lag smoothing: algorithm and consitency analysis. In *IEEE International Conference on Robotics and Automation*, 2011.

[3] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv*, 2016.

[4] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1), 2017.

[5] C. Hane, L. Heng, G. H. Lee, A. Sizov, and M. Pollefeys. Real-time direct dense matching on fisheye images using plane-sweep stereo. In *International Conference on 3D Vision (3DV)*, 2015.

[6] D. Honegger, T. Sattler, and M. Pollefeys. Embedded real-time multi-baseline stereo. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5245–5250. IEEE, 2017.

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. T. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *International Journal of Robotics Research*, 2015.

[8] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007.

[9] G. Sibley, L. Matthies, and G. Sukhatme. Sliding Window Filter with Application to Planetary Landing. *JFR*, 2010.

[10] L. D. Stefano, S. Mattoccia, and F. Tombari. ZNCC-based template matching using bounded partial correlation. *PRL*, 2005.

[11] X. Zheng, Z. Moratto, M. Li, and A. I. Mourikis. Photometric patch-based visual-inertial odometry. In *IEEE International Conference on Robotics and Automation*, 2017.