采用双视觉里程计的 SLAM 位姿图优化方法

曹剑飞¹⁾, 余金城²⁾, 潘尚杰¹⁾, 高枫²⁾, 于超²⁾, 胥智林²⁾, 黄正峰¹⁾, 汪玉^{2)*}

¹⁾ (合肥工业大学微电子学院 合肥 230009)

²⁾ (清华大学电子工程系 北京 100084)

(yu-wang@tsinghua.edu.cn)

摘 要: 后端轨迹优化是视觉同步定位与建图系统的重要组成部分, 可以显著地提高定位精度。然而, 现有的基于捆集约束法的优化方法在大场景中计算量大, 并且无法应用于端到端视觉里程计。针对这个问题, 提出了一种在前端采用 2 个视觉里程计的后端通用位姿图优化方法, 可以应用于端到端视觉里程计。该方法采用一个高速低精度的端到端视觉里程计以高频率运行, 同时一个低速高精度的视觉里程计以低频率运行, 局部优化通过 2 个里程计提供的约束条件使用高斯-牛顿法迭代优化; 在全局优化中基于关键帧进行场景匹配与局部优化同时进行。实验证明, 应用该优化方法的同步定位与建图系统可以在 KITTI 数据集上实时运行, 相较于 2 个视觉里程计都取得了精度上的较大提升, 并且对比现今开源的几种应用后端轨迹优化的著名同步定位与建图方法, 在轨迹误差、绝对轨迹误差、旋转误差和相对位姿误差上均取得了较低的误差, 兼顾了传统方法精度的优势和端到端方法速度上的优点。除此以外, 该优化方法还可以适用于其他更多的视觉里程计。

关键词: 同步定位与建图系统; 位姿图优化; 视觉里程计; 端到端; 实时后端优化

中图法分类号: TP391.41

DOI: 10.3724/SP.J.1089.2021.18663

A SLAM Pose Graph Optimization Method Using Dual Visual Odometry

Cao Jianfei¹⁾, Yu Jincheng²⁾, Pan Shangjie¹⁾, Gao Feng²⁾, Yu Chao²⁾, Xu Zhilin²⁾, Huang Zhengfeng¹⁾, and Wang Yu^{2)*}

¹⁾ (School of Microelectronics, Hefei University of Technology, Hefei 230009)

²⁾ (Department of Electronic Engineering, Tsinghua University, Beijing 100084)

Abstract: Backend trajectory optimization is an important part of the visual simultaneous localization and mapping system, which can significantly improve localization accuracy. However, the existing optimization methods based on the bundle adjustment have a large amount of calculation in large scenes and cannot be applied to end-to-end visual odometries. To solve this problem, a universal backend pose graph optimization algorithm with two visual odometries at the front end is proposed, which can be applied to end-to-end visual odometries. This method uses a high-speed but low-precision end-to-end visual odometry to run at high frequency, while a low-speed but high-precision visual odometry runs at a low frequency. Local optimization uses Gauss-Newton method iterative optimization through the constraints provided by two odometries. Global optimization is performed simultaneously which based on key frames scene matching. Experiments show that the simultaneous localization and mapping system which apply this optimization method can run in real-time on the KITTI dataset.

收稿日期: 2020-09-14; 修回日期: 2021-04-13. 基金项目: 国家自然科学基金(61874156). 曹剑飞(1997—), 男, 硕士研究生, 主要研究方向为多机协同任务; 余金城(1993—), 男, 博士研究生, 主要研究方向为多机协同任务; 潘尚杰(1997—), 男, 硕士研究生, 主要研究方向为嵌入式系统设计; 高枫(1997—), 男, 硕士研究生, 主要研究方向为多机协同任务; 于超(1994—), 女, 博士研究生, 主要研究方向为多机协同任务; 胥智林(1995—), 男, 硕士研究生, 主要研究方向为多机协同任务; 黄正峰(1978—), 男, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为芯片抗辐射加固; 汪玉(1982—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通讯作者, 主要研究方向为体系结构/应用程序协同设计、节能硬件计算系统设计、低功耗/可靠性感知系统设计。

Compared with the two visual odometries, the accuracy has been greatly improved. And compared with several well-known open source simultaneous localization and mapping methods that apply backend trajectory optimization, low errors have been achieved in trajectory error, absolute translational error, rotation error and relative pose error, taking into account the advantages of the accuracy of traditional methods and the advantages of high speed end-to-end methods. In addition, the optimization framework can also be applied to other more visual odometries.

Key words: simultaneous localization and mapping system; pose graph optimization; visual odometry; end-to-end; real-time backend optimization

同步定位与建图(simultaneous localization and mapping, SLAM)系统是近年来移动机器人领域的研究热点,它只依靠自身传感器信息而不借助外界绝对定位系统就能实时定位位置,可以应用于无人驾驶和空间探索等领域。

SLAM 可以分成传统的基于几何变换的和随着近年来深度学习兴起而涌现出的基于卷积神经网络(convolutional neural networks, CNN)的 2 大类系统。传统方法需要人为地设计如何对图像进行特征提取与匹配。这一环节存在 2 大问题:一是人为设计的特征提取方法可能存在理论缺陷导致忽略关键特征;二是这种方法在面临光照变化、场景变化和物体运动时性能会显著下降^[1]。同时,传统方法还面临着计算量大、计算流程专用固定、硬件加速困难以及只依靠单目相机难以获得深度信息等问题。基于 CNN 的 SLAM 系统分为用 CNN 取代原有模块的和基于 CNN 的端到端的 SLAM 系统。基于 CNN 的特征点提取与匹配方法具有更高的精度和环境鲁棒性,但计算速度较慢。而近年来涌现的诸多基于 CNN 的端到端 SLAM 方法,如 SfM-Learner^[2], DeMon^[3], UnDeepVO^[4]则可以快速地估计图像序列的帧间运动。与传统方法相比,基于 CNN 的端到端 SLAM 方法省去特征提取与匹配,直接输出帧间位姿变化,在线计算速度很快,并且具有较强的算法迁移能力。然而,现有的各种基于 CNN 的端到端 SLAM 方法在精度上仍然难以与传统方法相提并论;并且由于舍弃了图像处理环节,无法进行后端优化修正位姿。针对这个问题,本文提出一种对端到端 SLAM 方法进行后端优化,以提高位姿估计精度的方法。

视觉 SLAM 系统通常分为前端和后端。前端又称为视觉里程计(visula odometry, VO),根据相邻 2 帧图像估计它们之间发生的运动信息矩阵。因此,VO 逐帧地累积位姿就会存在累积误差的问题,

需要采取图优化的方法进行非线性优化。

然而,图优化算法的复杂度与优化变量数量呈三次方关系^[5],全局优化所有帧将消耗大量计算资源。为了能够使后端和前端 VO 一起实时进行优化,文献[6]提出了并行将后端优化分成局部优化和全局优化 2 部分,局部优化只维护当前帧附近的少量帧,可以有效地减少计算量,全局优化则以一个较低的频率并行进行。

尽管上述优化框架可以在一定条件下实时运行,但是其仍然有 2 点局限性。(1) 只有基于特征点的 SLAM 方法才可以利用捆集约束法进行局部优化,捆集约束法需要对图像提取像素特征点坐标再进行映射估计误差。因此,捆集约束法无法应用于端到端 VO,导致文献[6]的工作无法对端到端 VO 进行局部优化。(2) 捆集约束法的复杂度和特征点数量以及相机位姿数量呈三次方关系,因此,在更大的场景中,大量特征点的存在会严重降低计算效率,导致计算量越来越大,使得该算法无法实时进行。针对这些问题,本文提出通用的实时后端优化方法,其前端采用 2 个 VO,其中 Monodepth2^[7]可以实时运行,但精度低;另一个 VO 是深度光流视觉里程计(depth and flow for visual odometry, DFVO^[8]),拥有高精度但每帧计算时间长,本文对它进行隔帧输入。本文用 DFVO 的位姿关系结果作为局部优化的约束条件,因此,该优化方法可以用于端到端 VO,解决了因端到端 SLAM 没有局部优化约束条件而无法进行局部轨迹优化的问题;并且由于采用了位姿图优化的方法,省去了对空间中路标点的优化,即使在大场景环境下也能做到实时运行。

此外,本文还发现在对 DFVO 间隔帧输入图像时,在直线与弯道的位姿估计精度相差很大。本文分析了其产生的原因,并提出通过检测旋转情况动态调整 DFVO 约束权重的方法,保障优化的精度。

1 相关工作

1.1 VO

SLAM 系统中的 VO 目前有 3 种主流方法: (1) 传统基于图像几何变换的 VO, 如 FMD-SLAM^[9], SOFT-SLAM^[10] 和 ORB SLAM^[11] 系列. 其中, FMD-SLAM 融合了基于关键特征的多视图几何和直接法估计位姿, 利用最小化重投影误差来优化位姿. (2) SOFT-SLAM 在输入图像后计算梯度, 并且基于非极大值抑制原理提取角点, 随后用 5 点随机一致性采样方法计算初始位姿, 也是利用重投影误差优化位姿. (3) ORB SLAM 系列采用加速的分段特征检测(features from accelerated segment test, FAST)算法检测特征点, 之后计算特征点的描述子, 进行特征点匹配. SLAM 的传统方法通常存在计算量大、鲁棒性不高等问题.

随着深度学习方法的日益完善与兴起, 一些 SLAM 系统用深度学习的方法替代了传统方法中的一些模块, 如 CNN-SLAM^[12] 和 CodeSLAM^[13]. CNN-SLAM 将 LSD-SLAM^[14] 中的深度估计和图像匹配模块都替换成基于 CNN 的方法, 使系统能更加适应不同的应用场景. CodeSLAM 使用了一个 U-Net 神经网络提取图像特征及估计图像深度的不确定性.

最后一类方法是基于深度学习的端到端 VO, 图像经过神经网络直接输出位姿变化, 代表方法有 Depth-VO-Feat^[15] 和 Monodepth2. Depth-VO-Feat 是一个自监督端到端 VO, 使用 Disp Net 从单幅图像中预测图像深度, VO 网络从 2 幅输入图像中预测 2 帧位姿, 利用预测的位姿从左目图像恢复右目图像. 本文对 Monodepth2^[7] 进行了优化, 其基于 CNN 的端到端 VO 对每帧图像都进行估计, 并在单目图像上预测深度和位姿估计网络使用深度估计网络预测运动图像间位姿. Monodepth2 是自监督方法, 无须提供样本真值标定, 它通过在运动图像序列上训练一个建立在自监督损失函数上的架构来实现; 它使用图像序列中连续时间帧和位姿的重投影关系进行训练. Monodepth2 所需的计算量相当小, 运行速度超过 100 帧/s, 但其位姿估计精度较低, 也无法用捆集约束法对它进行位姿优化.

本文提供优化位姿约束的 VO 是 DFVO^[8], 同时利用了基于几何的和深度学习方法, 其基于 CNN 实现深度预测, 并提取和匹配图像特征点; 它成功地结合深度学习与对极几何和 perspective- n -point

(PnP)方法. 具体来说, 通过训练 2 个 CNN 估计单视图深度和双视图光流作为中间输出. DFVO 的核心思想是找出前后光流一致性图中偏差最小的前 N 个点对作为假定的正确特征点匹配, 几何方法依然使用 PnP 优化位姿, 这种方法的鲁棒性非常好, 具备比较强的环境适应和迁移能力, 系统不会受到比例尺漂移问题的困扰. DFVO 的精度相当高, 超过了现有绝大多数的 VO, 但是所需的计算量很大, 无法实时地应用于现实场景. 因此, 在本文中对 DFVO 以较低的频率输入图像.

1.2 后端位姿优化策略

2011 年, Kümmerle 等^[16]提出了高扩展性和高效的图优化框架 G^2o , 为 SLAM 的后端优化提供了良好的开源方法. 任何可以以图形式表达的优化问题都可以用 G^2o 解决.

为了解决捆集约束法在优化变量中引入路标点导致计算量大的问题, Lee 等^[17]和 Latif 等^[18]采取了位姿图优化的方法, 舍弃对路标点的优化. 位姿图优化的优化变量只有每帧位姿, 从而显著地降低了计算量.

为进一步降低后端优化的计算量以达到实时优化的目标, 文献[6]是第 1 个提出双窗口并行优化方法的论文, 其在单个优化框架内将点-位姿约束和位姿-位姿约束耦合在一起. 其中内部窗口用于尽可能准确地对局部区域进行建模, 外部窗口中的位姿图用于稳定外围区域.

也有选用其他优化约束条件的, 如 OKVIS^[19] 利用惯性传感器作为额外约束, 将旧状态部分边缘化, 同时基于关键帧进行优化对象筛选, 维护了有限大小的优化窗口, 从而确保实时运行.

LDSO^[20]是一个将捆集约束法和位姿图优化交替进行的方法, 其局部优化采用捆集约束法, 全局优化为了减少计算量和内存消耗使用了位姿图优化.

1.3 回环检测

回环检测的本质其实是场景识别, 本文采用的场景识别方法是广义局部聚合描述符层网络(net vector of locally aggregated descriptors, NetVLAD)^[21], 它用于解决大规模视觉地点识别, 目标是快速、准确地找出照片的拍摄地点. NetVLAD 设计了一种新的 CNN 架构, 可以直接针对地点识别任务进行端到端训练. NetVLAD 具有很强的算法迁移能力, 可以很容易地嵌入任何 CNN 架构, 并且在训练的过程中比较容易收敛. 为了获得更紧凑的图像表示, 使用了主向量分析法. 在文献[21]的实验以及

本文的实验中都证明 NetVLAD 能够快速、准确地进行场景识别。

2 通用图优化原理

现在主流的后端优化方法都是基于图优化实现的, 图优化是将 SLAM 问题以图的形式表示出来, 在图优化中顶点表示机器人位姿, 边表示 2 个顶点的位姿约束关系. 通过构建整体误差函数再进行迭代达到优化估计位姿结果的目的. 与基于一阶马尔可夫性质的滤波器法相比, 图优化由于引入了全面的变量和约束, 借助最小二乘可以实现全局优化. 通用图优化是本文构建图模型之后的求解算法.

用图优化表示 SLAM 问题如图 1 所示, 边对应位姿之间的约束, 一个节点为一帧的位姿, 所有节点的结合组成了整个系统的运动轨迹. 将所有的误差项相加得到总体误差函数为

$$F(X) = e_{12}^T \Omega_{12} e_{12} + e_{23}^T \Omega_{23} e_{23} + e_{31}^T \Omega_{31} e_{31} + e_{24}^T \Omega_{24} e_{24} \quad (1)$$

其中, e 表示一个误差分量; Ω 表示各误差分量权重的信息矩阵. 对于式(1)中每个误差分量可以具体表示为

$$F(X) = \sum_{\langle i, j \in c \rangle} e(X_i, X_j, Z_{ij})^T \Omega_{ij} e(X_i, X_j, Z_{ij}) \quad (2)$$

其中, X 表示路标点; Z_{ij} 表示联系状态变量 i 和状态变量 j 的观测, 数学上表现为一个约束方程.

图 1 位姿图优化

有了目标函数式(2), 就可以进行最小二乘优化. 主流的方法是采取高斯-牛顿法. 高斯-牛顿法的思想是对误差函数采用一阶泰勒展开, 然后构建一个新的误差函数, 求这个误差函数的极小值, 以极小值作为新的迭代点. 重复迭代, 直到满足终止条件.

首先对目标函数进行一阶泰勒展开, 即

$$f(x + \Delta x) \approx f(x) + J(x)\Delta x,$$

求解使得 $f(x + \Delta x)$ 最小的 Δx ; x 表示空间中路标点的位置. 由于目标函数必须是标量, 因此必须用它的平方形式表达目标函数. 当前的目标是为了寻找下降矢量 Δx , 使 $\|f(x + \Delta x)\|^2$ 达到最小. 因此, 展开目标函数为

$$\begin{aligned} \|f(x) + J(x)\Delta x\|^2 &= \\ (f(x) + J(x)\Delta x)^T (f(x) + J(x)\Delta x) &= \\ (\|f(x)\|_2^2 + 2f(x)^T J(x)\Delta x + \Delta x^T J(x)^T J(x)\Delta x). \end{aligned}$$

求上式关于 Δx 的导数, 并令

$$2J(x)^T f(x) + 2J(x)^T J(x)\Delta x = 0,$$

则 $J(x)J(x)^T \Delta x = -J(x)^T f(x)$.

把左边的系数定义为 H , 右边定义为 b , 那么上式变为 $H\Delta x^* = b$; 再用舒尔消元的方法求解线性方程, 更新变量的值, 不断迭代求得使目标函数达到最小值时即为最优解.

3 本文方法

本节将详细介绍本文提出的两阶段位姿优化方法, 包括提出的优化框架、局部优化和全局优化的方法, 以及联合优化的算法.

本文搭建了如图 2 所示的 SLAM 系统, 为所提出的后端优化系统进行测试, 它由 VO、回环检测和后端优化 3 个组件组成. 相机图像将传送到 VO 和回环检测; 回环检测模块将对一帧图像生成总体描述子并存储到内存中, 并将该帧的总体描述子与之前存储的数据计算欧几里得度量, 以便发现机器人是否回到过之前走过的位置. 回环检测模块的结果将作为后端优化当中全局优化的约束条件, 因此, VO 和回环检测的结果都将输入后端优化模块进行位姿优化.

3.1 优化框架

如图 3 所示, 滑动优化窗口包含着当前帧和之前的几帧, 并且随着当前帧的移动一起向前移动. 当前帧继续向前移动了一个滑动窗口大小的帧数时, 就产生了新的局部优化窗口. 系统在运行的过程中每隔一段时间会产生一个关键帧, 将所有关键帧加入全局优化窗口. 当检测到回环时, 就会进行一次全局优化. 在本文的优化框架中, 局部优化窗口仅需包含几帧位姿进行优化. 基于关键帧和回环检测的全局优化也极大地减少了全局优化中变量的数量和优化频率. 因此, 本系统框架的计

图2 本文系统概览

图3 本文方法优化框架

算量需求较小,在后续实验中证明可以满足实时性的要求。

3.2 局部优化

如图4所示,线程上的每条垂直线代表一帧的到来。每帧图像都会输入 Monodepth2,生成初始位姿数据。同时,每间隔固定帧数将输入一幅图像到 DFVO 中,生成相邻 2 帧之间的相对位姿矩阵。DFVO 输出的帧与帧之间的位置关系就可以作为位姿图优化的约束条件。局部优化进行的频率取决于滑动窗口的大小。

图4 局部优化方法示意

由于 DFVO 速度较慢,至少需要间隔 3 帧输入一幅图像才能满足实时的要求。然而,这可能会导致 2 幅图像的差异过大,共视关系的减少将直接导致相机位姿估计不准。相机在作直线运动时图像的变化较小,即使间隔 3 帧输入对在直线上运动不会产生大的精度影响。而在相机发生旋转时,图像

间的变化较大,仍然间隔 3 帧输入会导致旋转的估计精度大大降低。因此,对 DFVO 进行间隔帧输入对直线运动上的估计精度降低很有限,但在转弯时精度则显著下降。这一点在下一节的实验部分也会得到证明。

为了解决转弯时 DFVO 估计不准,有可能会提供错误约束条件的问题,在本文算法中会对旋转矩阵实时监测,一旦监测到在一段时间内的连续旋转,则可以认为目前处于弯道上,将根据旋转程度自适应地降低此时 DFVO 提供的位姿约束的权重,从而防止错误的权重影响优化。

在每次优化的过程中,将滑动窗口内的 K 帧 Monodepth2 位姿矩阵和窗口内 DFVO 提供的位姿约束写入 G^2o 文件,调用 G^2o 进行优化。优化之后的结果将实时更新位姿,并且在新的位姿的结果基础上,进行 Monodepth2 的位姿预测作为下一次优化的输入。

3.3 全局优化

在本文的优化框架中,每帧位姿都会被局部优化,但只有出现关键帧时才可能进行全局优化。本文关键帧的选取方法基于距离,机器人每运动了一定的位移后就会选取一个关键帧。与直接间隔固定帧选取关键帧会受到移动速度的影响相比,这种方法可以保证在整体的运行轨迹上均匀地选取关键帧。所有的关键帧都将加入全局优化。

此外,将对关键帧利用 NetVLAD 的方法进行回环检测,一旦检测到机器人来到曾经到过的位置,则进行一次全局优化。优化之后许多关键帧的位姿往往会发生较大的变化,而实际情况应当不仅仅是关键帧的位姿需要优化,关键帧周围的所有帧的位姿也应该相应调整。

为此,本文提出了一种位姿更新方法,对全局优化后检测到的位姿变化较大的关键帧周围的普通帧,基于关键帧位姿和局部优化提供的相对位

姿关系更新位姿. 该方法以较低的计算量、在不增加图优化的优化变量的前提下, 维护全局优化和局部优化的统一性.

3.4 本文算法

本文提出的后端优化算法步骤如下.

输入. 相机图像、Monodepth2 逐帧提供的初始相对位姿矩阵 P , DFVO 间隔 I 帧提供的位姿约束、滑动窗口的范围为 K 帧.

输出. 优化后每帧的绝对位姿矩阵 M .

Step1. 按照帧编号记录 Monodepth2 提供的当前帧初始位姿.

Step2. 当帧编号是 K 的整数倍时进行局部优化.

Step2.1. 写入 G^2o 文件:

Step2.1.1. 根据初始位姿矩阵转化四元数写入图优化顶点位姿;

Step2.1.2. 根据初始相邻 2 帧之间相对位姿矩阵转化四元数写入图优化边约束;

Step2.1.3. 根据 DFVO 提供的间隔帧之间相对位姿矩阵和此时检测旋转矩阵, 得出的约束权重写入图优化边约束;

Step2.2. 调用 G^2o 库进行图优化.

Step3. 读取优化后的位姿矩阵, 在此基础上根据 Monodepth2 提供的相对位姿矩阵计算之后的绝对位姿, 将其作为下次优化的初始位姿矩阵.

Step4. 检测当前帧是否为关键帧, 若是, 转 Step5; 否则结束当前循环.

Step5. 输入当前帧图像到 NetVLAD 网络, 生成并记录图像的总描述子, 与之前记录下的所有描述子对比, 寻找是否有差异在阈值范围内的帧编号. 若存在, 则转 Step6, 否则结束当前循环.

Step6. 根据当前所有关键帧的位姿矩阵和回环检测的信息采用与 Step2 同样的优化方法进行全局优化.

Step7. 在进行完一次全局优化后, 检测哪些关键帧的位姿矩阵发生了较大的变化.

Step8. 对位姿发生较大变化的关键帧根据局部优化后的相对位姿矩阵更新其周围普通帧的位姿矩阵.

本文改进了 G^2o 优化库的优化策略, G^2o 库在对位姿进行优化时会把所有的位姿列为同一置信度, 通常, 全局优化时会根据回环检测的结果需要匹配早期帧和后期帧, 大幅度改变了早期帧的位姿. 但实际上由于累计误差, 早期帧的精度高于后期帧. 为此, 本文改进了 G^2o 优化库, 大幅度提高早期帧的置信度, 以优化后期帧位姿为主, 在很大程度上保证了 G^2o 优化后的精度.

4 实验

本节对本文方法从速度和精度上进行评估,

包括对 DFVO 和 Monodepth2 的精度提升以及与其他 SLAM 的比较. 本文的实验在 Intel i7-8750h CPU, NVIDIA 1080 TI GPU 上进行. 选用 KITTI 2000 数据集^[22]序列, 该数据集是在室外驾车途中以 10 帧/s 的频率采集的 4540 幅图像. 使用下列评价指标: 绝对轨迹误差(absolute translational error, ATE)和轨迹误差, 用来测量预测相机的位姿 $[x, y, z]$ 和真值之间的误差; 相对位姿误差(relative pose error, RPE), 用来测量帧与帧之间的相对位姿误差; 旋转误差, 是相机每运动 100m 偏移的角度, 用来测量相机旋转角度的总体误差.

4.1 DFVO 隔帧输入测试

实验中, 每 4 帧向 DFVO 输入一帧图像, 使用 DFVO 输出的位姿作为局部优化约束, 首先测试 DFVO 在这种条件下的实验结果. 结果发现它在 KITTI 数据集 00 序列的不同路段上表现差异很大. 如图 5 所示, 图 5a 是 DFVO 在 0~1100 帧的轨迹, 图 5b 是在 2500~3000 帧上一个弯道的轨迹. 蓝色实线代表轨迹位姿, 红色虚线代表真值轨迹. 对 DFVO 进行间隔帧输入在 0~1100 帧表现出了很好的精度, 具体的精度指标如表 1 所示. 但是在 2500~3000 帧的一个弯道上产生了巨大的误差, 该误差会由于误差累计保留下来并影响后续的位姿估计.

图 5 DFVO 隔帧输入在 00 序列不同路段上的表现

表 1 DFVO 在 00 序列不同路段的精度

DFVO	轨迹误差/%	旋转误差/(°)	ATE/m	RPE/m
1~1100 帧	1.990	0.945	5.649	0.166
2500~3000 帧	58.249	28.616	130.191	0.452

这种情况发生的原因是在弯道时相机旋转较快, 前后图像之间的差异明显大于在直线上的运动, 导致位姿变化的估计不准. 对于传统 VO 方法, 急转弯使帧之间的数据关联变得困难^[23].

DFVO 使用光流法追踪图像中像素的运动, 图像之间的光流差异过大会导致 2 帧图像之间匹配点不正确. 如图 6 所示, 可以看出当相机发生旋

转时, 前后 2 帧图像之间可以轻易发生较大的变化, 导致光流图和原始图像之间产生显著差异. 进一步会破坏前向和后向光流的一致性, 最终导致匹配点不正确.

图 6 前向和后向光流图以及匹配点

针对这个问题, 本文提出对图像像素运动的实时监控方法, 一旦发现相机在进行较快速度的旋转, 就会在位姿图优化中降低 DFVO 约束的权重, 以保证轨迹的精度.

4.2 本文局部优化方法的效果

应用本文局部优化方法模拟真实的机器人使用场景时, 对整个系统逐帧输入, 局部优化会伴随机器人运动进行若干次. 如图 7 所示, 实线表示估计轨迹, 虚线表示真值轨迹, 直观地展示了对 DFVO 和 Monodepth2 的轨迹优化, 可以看出轨迹的精度有了显著的提升, 具体的精度指标提升如表 2 所示. 除此以外, 还与其他应用了局部优化的前端 VO 的精度做了对比.

图 7 局部优化效果图

实验结果表明, 在对 DFVO 和 Monodepth2 应用了本文提出的局部优化方法后, 与其他几种著名的方法对比, 本文方法在 ATE、RPE、轨迹误差、旋转误差上均显著低于对比方法的平均误差, ATE 误差低于对比的所有方法. ORB-SLAM 展现出低的旋转误差, 但由于尺度缩放的问题展现出了较高的轨迹误差. 由此可见, 本文的端到端 SLAM 的位姿优化方法对位姿精度提升明显, 在精度上超越了几种著名传统 SLAM 方法. DFVO 基于 CNN 实现图像特征点提取与匹配相比传统方法具有更高的精度和环境鲁棒性, 对其间隔帧输入弥补了其速度缺陷. Monodepth2 基于 CNN 实现端到端位姿输出, 计算速度很快, 并且具有较强的算法迁移能力, 对其使用本文的后端优化方法弥补了其精度缺陷. 因此, 本文方法兼顾了传统 SLAM 精度的优势和端到端 SLAM 的速度优势.

表 2 与其他未进行全局优化的 SLAM 系统的精度对比

方法	轨迹误差/%	旋转误差/(°)	ATE/m	RPE/m
DFVO ^[8]	13.46	4.43	170.00	0.430
Monodepth2 ^[7]	8.28	2.18	148.10	0.092
ORB-SLAM2 ^[24] (无 LC)	11.43	<u>0.58</u>	40.65	0.169
VISO2 ^[25]	10.53	2.73	79.24	0.221
DSO ^[20]			113.20	
ORB-SLAM3 ^[26] (无 LC 和 IMU)	7.62	0.52	<u>38.22</u>	0.105
Depth-VO-Feat ^[15]	6.23	2.44	64.45	0.084
SfM-Learner ^[27]	21.32	6.19	104.87	0.282
SC-SfMLearner ^[28]	11.01	3.39	93.04	0.139
对比方法平均	11.24	2.81	94.61	0.190
本文(10 帧滑动窗口)	<u>6.66</u>	2.58	36.77	<u>0.090</u>
本文(15 帧滑动窗口)	7.23	3.32	43.11	0.093

注: 表中加粗数据表示该列中最小值, 下划线数据表示该列中次小值.

4.3 联合优化的效果

最后使用本文的优化策略进行局部优化和全局优化. 如图 8 所示, 图 8a 是本文方法联合优化后的轨迹图, 图 8b 是 Monodepth2 全局优化后的轨迹图. 相比之下轨迹精度有了显著提升.

表 3 将本文全局优化后与其他 6 种进行回环检测全局优化的 SLAM 系统的精度和帧率进行了对比. 其中, ORBSLAM2, ORBSLAM3 以及 LDSO 本身带有回环检测, 对其他 3 种方法应用了基于 NetVLAD 的全局优化方法. 可以看出, 本文方法的运行速度优于对比方法.

图 8 联合优化之后的轨迹图

表 3 与其他进行回环检测全局优化的 SLAM 精度对比

方法	轨迹 误差/%	旋转 误差/(°)	ATE/m	RPE/m	帧率/ (帧·s ⁻¹)
DFVO ^[8]	13.23	3.85	125.47	0.42	3.30
Monodepth2 ^[7]	9.84	8.18	55.49	0.15	103.09
ORB-SLAM2 ^[24] (自带回环检测)	<u>2.35</u>	<u>0.35</u>	6.03	0.21	8.63
LDSO ^[20] (自带回环检测)	3.85	1.00	35.69	0.08	9.10
ORB-SLAM3 ^[26] (自带回环检测)	0.72	0.25	<u>8.64</u>	0.02	8.52
Depth-VO-Feat ^[15]	6.14	3.24	47.13	0.12	<u>28.26</u>
本文	5.23	2.12	12.31	<u>0.07</u>	9.74
对比方法平均	6.02	2.81	46.40	<u>0.17</u>	26.90

注. 表中加粗数据表示该列中最小值, 下划线数据表示该列中次小值.

精度对比结果显示, 本文方法的 4 种误差指标均显著低于其他 6 种方法误差的平均值. DFVO 和 Monodepth2 在某些指标上的误差甚至高于未进行全局优化时. 这是因为本身的 VO 位姿误差与真值差异过大, 全局优化并不能弥补先前的巨大误差. 尽管 ORB-SLAM2, ORB-SLAM3 以及 LDSO 在某些精度指标上优于本文方法, 但它们的系统运行帧率均低于本文方法, 并且在本文的实验平台上未能达到实时运行的要求.

4.4 系统中各组件的运行时间

表 4 列出了本文系统中各组件的运行时间, DFVO 是最消耗计算资源的线程, 占用了 73.78% 的运算时间. 由于本文中采取多个降低后端优化计算量的举措, 因此局部优化和全局优化总耗时只占整个系统的 7.22%. 采用基于关键帧的回环检测方法虽然单次的 NetVLAD 耗时较长, 但是进行的频率较低, 并没有占用很多计算时间. KITTI 数据集 00 序列的长度是 470.58 s. 而本文的仿真实验跑完 00 序列一共耗时 466.15 s, 满足了实时定位的要求.

表 4 系统中各组件的运行时间与执行频率

组件	单次执行 时间/ms	执行频率/ (帧/次)	线程运行 总时间/s	系统运行 总时间/s
DFVO ^[8]	303.0	4	343.91	466.15
Monodepth2 ^[7]	9.7	1	44.04	
局部优化	20.4	10~15	9.26	
NetVLAD	422.0	43	44.56	
全局优化	231.0	43	24.39	

5 结 语

本文提出了一个可以广泛应用于端到端 SLAM 的后端优化框架, 其中局部优化约束条件解决了诸如 CNN SLAM 等端到端方法无法进行局部优化的问题. 同时, 本文探索了基于几何的 VO 在弯道上会发生位姿估计偏差的原因. 通过降低弯道时约束条件权重的方法, 保证了轨迹精度. 实验结果表明, 应用本文的后端优化方法, 不仅在很大程度提高了 2 种 VO 的精度, 与其他著名的应用了局部优化的传统 SLAM 方法相比, 本文方法的精度也有所领先. 最后, 很重要的一点是本文方法可以在 KITTI 数据集上做到实时运行. 尽管如此, 本文方法还没有在实际环境中进行测试, 下一步希望将本文方法部署在嵌入式系统中进行进一步实验.

参考文献(References):

- [1] Fuentes-Pacheco J, Ascencio J R, Rendon-Mancha J M. Visual simultaneous localization and mapping: a survey[J]. Artificial Intelligence Review: An International Science and Engineering Journal, 2015, 43(1): 55-81
- [2] Zhang L Q, Li G, Li T H. Temporal-aware SfM-Learner: unsupervised learning monocular depth and motion from stereo video clips[C] //Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval. Los Alamitos: IEEE Computer Society Press, 2020: 253-258
- [3] Ummenhofer B, Zhou H Z, Uhrig J, et al. DeMoN: depth and motion network for learning monocular stereo[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 5038-5047
- [4] Li R H, Wang S, Long Z Q, et al. UnDeepVO: monocular visual odometry through unsupervised deep learning[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2018: 7286-7291
- [5] Jeong J, Mulligan J, Correll N. Trinocular visual odometry for

- divergent views with minimal overlap[C] //Proceedings of the IEEE Workshop on Robot Vision. Los Alamitos: IEEE Computer Society Press, 2013: 229-236
- [6] Strasdat H, Davison A J, Montiel J, *et al.* Double window optimisation for constant time visual SLAM[C] //Proceedings of International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2011: 2352-2359
- [7] Godard C, Aodha O M, Firman M, *et al.* Digging into self-supervised monocular depth estimation[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 3827-3837
- [8] Zhan H Y, Weerasekera C S, Bian J W, *et al.* Visual odometry revisited: what should be learnt?[OL]. [2020-09-14]. <https://arxiv.org/pdf/1909.09803.pdf>
- [9] Tang F L, Li H P, Wu Y H. FMD stereo SLAM: fusing MVG and direct formulation towards accurate and fast stereo SLAM[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2019: 133-139
- [10] Cvisic I, Csic J, Markovic I, *et al.* SOFT-SLAM: computationally efficient stereo visual simultaneous localization and mapping for autonomous unmaned aerial vehicles[J]. *Journal of Field Robotics*, 2018, 35(4): 578-595
- [11] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. *IEEE Transactions on Robotics*, 2015, 31(5): 1147-1163
- [12] Tateno K, Tombari F, Laina I, *et al.* CNN-SLAM: real-time dense monocular SLAM with learned depth prediction[C] // Proceedings of the IEEE Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6566-6574
- [13] Bloesch M, Czarnowski J, Clark R, *et al.* CodeSLAM - learning a compact, optimisable representation for dense visual SLAM[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 2560-2568
- [14] Engel J, Schps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM[C] //Proceedings of European Conference on Computer Vision. Heidelberg: Springer, 2014: 834-849
- [15] Zhan H Y, Garg R, Weerasekera C S, *et al.* Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 340-349
- [16] Kümmerle R, Grisetti G, Strasdat H, *et al.* G²o: a general framework for graph optimization[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2011: 3607-3613
- [17] Lee D, Myung H. Solution to the SLAM problem in low dynamic environments using a pose graph and an RGB-D sensor[J]. *Sensors*, 2014, 14(7): 12467-12496
- [18] Latif Y, Cadena C, Neira J. Robust loop closing over time for pose graph SLAM[J]. *International Journal of Robotics Research*, 2013, 32(14): 1611-1626
- [19] Leutenegger S, Lynen S, Bosse M C, *et al.* Keyframe-based visual-inertial odometry using nonlinear optimization[J]. *The International Journal of Robotics Research*, 2015, 34(3): 314-334
- [20] Gao X, Wang R, Demmel N, *et al.* LDSO: direct sparse odometry with loop closure[C] //Proceedings of the IEEE/RSI International Conference on Intelligent Robots and Systems. Los Alamitos: IEEE Computer Society Press, 2018: 2198-2204
- [21] Arandjelovic R, Gronat P, Torii A, *et al.* NetVLAD: CNN architecture for weakly supervised place recognition[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(6): 1437-1451
- [22] Geiger A, Lenz P, Stiller C, *et al.* Vision meets robotics: the KITTI dataset[J]. *International Journal of Robotics Research*, 2013, 32(11): 1231-1237
- [23] Clark R, Wang S, Wen H K, *et al.* VINet: visual-inertial odometry as a sequence-to-sequence learning problem[C] // Proceedings of the 31st AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2017: 3995-4001
- [24] Mur-Artal R, Tardos J D. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras[J]. *IEEE Transactions on Robotics*, 2017, 33(5): 1255-1262
- [25] Geiger A, Ziegler J, Stiller C. StereoScan: dense 3D reconstruction in real-time[C] //Proceedings of the IEEE Intelligent Vehicles Symposium. Los Alamitos: IEEE Computer Society Press, 2011: 963-968
- [26] Campos C, Elvira R, Rodríguez J J G, *et al.* ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM[OL]. [2020-09-14]. <https://arxiv.org/pdf/2007.11898.pdf>
- [27] Zhou T H, Brown M, Snavely N, *et al.* Unsupervised learning of depth and ego-motion from video[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6612-6619
- [28] Bian J W, Li Z C, Wang N Y, *et al.* Unsupervised scale-consistent depth and ego-motion learning from monocular video[OL]. [2020-09-14]. <https://arxiv.org/abs/1908.10553>