基于相机朝向变化的增量式位姿图分段算法

范涵奇，王劭靖

引用本文

范涵奇，王劭靖. 基于相机朝向变化的增量式位姿图分段算法[J]. 计算机科学, 2023, 50(7): 152-159.

FAN Hanqi, WANG Shaojing. Incremental Pose Graph Segmentation Algorithm Based on Camera Orientation Change [J]. Computer Science, 2023, 50(7): 152-159.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220400166

基于相机朝向变化的增量式位姿图分段算法

范涵奇　王劭靖
北方工业大学信息学院　北京 100144
(fhq@ncut.edu.cn)

摘　要　位姿图优化是估计相机轨迹过程中减少累积误差的重要方法,但随着相机的不断运动,位姿图的规模会不断增大导致优化速度下降,使得轨迹估计难以应用于 AR/VR(Augmented Reality/Virtual Reality)等实时性要求较高的领域。针对此问题,文中提出了一种基于相机朝向变化的增量式位姿图分段算法。所提算法能够将位姿图在相机发生朝向变化较大的时刻进行分段,从而只对这些朝向变化较大的相机进行位姿图优化,以有效减小位姿图优化的规模,提高优化速度。针对其余未进行位姿图优化的每个相机,分别将其所在轨迹段的起始相机和终止相机作为参考相机,将根据不同参考相机估计出的不同位姿进行加权平均,从而求解出相机的最终位姿,既避免了非线性优化的大量计算,又降低了噪声的影响,达到了较高的精度。在 EuRoC,TUM 和 KITTI 数据集上进行了实验,结果表明,所提算法在减小位姿图优化规模的基础上保证了相机轨迹的精度。

关键词:位姿图;相机朝向;轨迹分段;加权平均;增量式
中图法分类号　TP242

Incremental Pose Graph Segmentation Algorithm Based on Camera Orientation Change

FAN Hanqi and WANG Shaojing
School of Information,North China University of Technology,Beijing 100144,China

Abstract　In camera trajectory estimation,pose graph is one of the most widely used tools to reduce the cumulative error. However,the scale of the pose graph grows as the cameras move,which would render trajectory estimation impossible in the real-time required applications like AR/VR(augmented reality/virtual reality). To reduce the size of the pose graph optimization,this paper proposes an algorithm that segments the trajectory of cameras incrementally by the change of orientation. The orientation changes significantly where the trajectory is segmented by the proposed algorithm. Efficiency is improved by optimizing the cameras with obvious orientation changes instead of the entire trajectory. In addition,the starting camera and the ending camera of the trajectory segment are utilized as references to estimate different poses for each camera within the trajectory segment,followed by the weighted average method is used on its different poses to solve the final pose. Which not only avoids a large amount of computation of nonlinear optimization,but also reduces the influence of noise to achieve high accuracy. Experiments on EuRoC,TUM,and KITTI datasets show that the proposed scheme reduces the size of the pose graph optimization and ensures the accuracy of trajectory.

Keywords　Pose graph,Camera orientation,Trajectory segmentation,Weighted average,Incremental

1　引言

位姿图优化[1]是 SLAM(Simultaneous Localization and Mapping)[2]后端重要的优化算法,对提高估计的轨迹精度起着关键作用。

SLAM 系统主要分为前端和后端[3]两大部分。前端主要负责机器人的定位并建立位姿图,后端主要负责地图的优化。主流的视觉 SLAM 系统[4-7]在后端使用的是位姿图优化和 Bundle Adjustment(BA)等非线性优化方法。但 BA 同时对相机的位姿和三维空间点坐标进行优化,时间复杂度为优化变量数量的三次方[8],较难实现快速的收敛[9],对 SLAM 系统的实时性产生了影响。与 BA 相比,位姿图优化只优化了相机位姿,就极大减少了需要优化的变量,缩短了优化时间,并且能够在闭环检测后[10]有效减少相机轨迹的漂移,从而减少累积误差。因此,近年来的 SLAM 系统[11-12]都只进行位姿图优化而不进行全局 BA,从而在加快优化速度的同时保证较高的优化精度。

随着相机的不断运动,位姿图的规模会不断增长导致优化效率下降,限制了 SLAM 系统在 AR/VR(Augmented Reality/Virtual Reality)等计算资源有限的平台上的应用。因此,如何减少位姿图优化的规模从而加快优化速度[12-19]是当前的研究热点。

到稿日期:2022-04-16　返修日期:2022-10-11
基金项目:国家重点研发计划(2020YFC0811004)
This work was supported by the National Key Research and Development Program of China(2020YFC0811004).
通信作者:王劭靖(494046824@qq.com)

2 相关工作

Johannsson 等[14]认为，当位姿图中时间上相邻的两个相机节点都检测到回环时，说明相机正重访(Revisit)之前探索过的区域，将这些时间上相邻且都检测到回环的节点视为冗余节点，将多个冗余节点和闭环节点之间的约束(Constraint)进行合并后直接将冗余节点去除，以此减小位姿图优化的规模，但这种方法只适用于相机重复经过同一个区域的情况。

Zeng 等[12]从神经生物学的角度对认知地图进行分析，通过计算当前相机节点与相邻相机节点之间的位移距离和朝向变化的乘积，来判定当前相机节点是否为可以删除的冗余节点，从而减小位姿图优化的规模，但在大型室外场景中应用时效果不明显。

Schmuck 等[15]提出了基于信息论检测冗余相机节点的方法，但该方法计算相机节点之间互信息的时间较长，需要额外的线程才能满足 SLAM 实时的要求。

与上述通过去除冗余节点来减小位姿图优化规模的方法不同，Liu 等[13]将位姿图分解为多个单链图(Single-Chain graph)和平行链图(Parallel-Chain graph)后分别进行优化，以加快优化速度。但 Liu 等提出的方法只适用于同时存在单链图和平行链图的情况，若位姿图中只有单链图或平行链图，位姿图则不会被分解为多个子图，优化时间也不会缩短。

Wang 等[19]在假设里程计的漂移与行驶距离成正比的前提下，根据位姿图中位姿节点之间的距离建立局部子图，从而减小位姿图优化的规模。类似地，Ding 等[18]和 Zhao 等[17]通过计算局部子图之间的重叠率来挑选出需要边缘化(Marginalization)的局部子图，并且 Ding 等[18]采用基于 Chow-Liu 树的方法将边缘化后产生的稠密图近似为稀疏树。但只有机器人在固定空间内反复行驶时，该方法减小位姿图优化规模的效果才较为显著。

车联网领域[20]中的交通数据有很多的轨迹数据，这些轨迹数据与 SLAM 领域中的位姿图十分相似。轨迹数据量通常很大以至于在进行数据分析时会严重影响运算效率，并且车辆的朝向在直线轨迹中几乎不发生改变。Keogh 等[21]提出了开放窗口(Opening window)算法，将车辆运动的轨迹用多个线段子轨迹近似，极大减少了轨迹数据。Latif 等[16]使用了类似的方法将位姿图根据弯曲程度划分为多个直线段，认为直线段中相机节点之间的约束都是线性约束，不需要加入到非线性优化中，从而只对直线段的首尾相机节点进行位姿图优化，也就是运动过程中朝向发生较大变化的相机节点，极大减小了位姿图优化的规模。但在实际 SLAM 的应用中，直线段中也会存在发生朝向变化较大的相机节点，若只优化直线段的首尾相机节点，则会有一部分非线性约束不能加入到位姿图优化中，导致优化精度下降。此外，位姿图在发生弯曲时，相机不一定会发生朝向改变，若只根据位姿图的弯曲程度进行分段，则会有一部分线性约束加入到位姿图优化中，导致增加不必要的计算，降低优化速度。

本文针对位姿图规模过大导致优化速度下降的问题，提出了基于相机朝向变化的增量式位姿图分段算法。该算法能够在不限制机器人运动方式的前提下，在机器人运动过程中只挑选出朝向变化较大的相机进行位姿图优化，以减小优化的规模。针对其余未进行位姿图优化的每个相机，将其所在轨迹段的起始相机和终止相机分别作为参考相机，由此估计出该相机的不同位姿，将不同位姿的加权平均结果作为该相机的最终位姿，从而在减小位姿图优化规模的同时保证相机轨迹的精度。

3 基于相机朝向变化的增量式位姿图分段算法

当相机在运动过程中朝向几乎不发生改变时，只要这段运动的首个相机的朝向优化好后，其余相机的朝向就能够直接得到，而不必通过非线性优化的方式得到，并且在这些相机朝向已知的情况下，对这些相机位置的优化也不必以非线性优化的方式进行。因此，这些朝向几乎不发生变化的相机不需要被加入到位姿图优化中以非线性优化的方式进行优化。因此本文只对运动过程中朝向变化较大的相机进行位姿图优化，从而减小位姿图优化的规模。

两个原点重合的空间直角坐标系只需要两次转动即可完全重合。如图 1 所示，坐标系 3 只需两次转动即可与坐标系 1 完全重合：第一次令坐标系 3 的 Z_3 轴偏转 b 度与坐标系 1 的 Z_1 轴重合，转动后的坐标系为坐标系 2，第二次令坐标系 2 绕着自身的 Z_2 轴旋转 a 度即可与坐标系 1 完全重合。坐标系 3 可以视为相机的朝向，坐标系 1 可以视为世界坐标系的朝向，因此 a 和 b 两个旋转角度的变化可以代表相机在运动过程中的朝向变化。无论相机在直线轨迹中还是弯曲轨迹中，只要朝向发生改变，对应的 a 和 b 两个旋转角度就会发生变化。本文由此提出在机器人运动过程中，根据旋转角度 a 随时间变化的图像以及旋转角度 b 随时间变化的图像对机器人的轨迹进行分段，从而有效地在机器人任意运动过程中只挑选出朝向变化较大的相机进行位姿图优化。

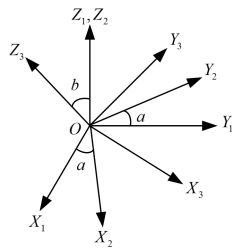
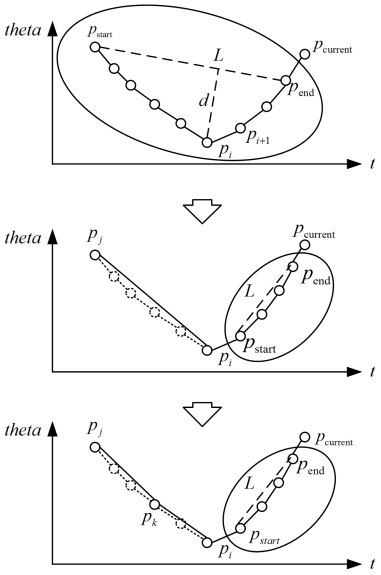


图 1 两个原点重合的空间直角坐标系经过两次旋转完全重合的示意图

Fig. 1 Schematic diagram of two three-dimensional coordinate system with the same origin completely overlapping after two rotations

图 2 为对 a 和 b 其中一个角度随时间变化的函数图像进行分段的示意图。图 2 中每一个圆节点都代表着函数图像上的节点，p_{current} 代表当前帧时刻对应在函数图像上的节点。从图 2 的顶部图像可以看出，通过 p_{start} 和 p_{end} 可计算直线 L，由此可以计算 p_{start} 和 p_{end} 之间所有节点相对于直线 L 的偏移量，即实线椭圆框内的节点到直线 L 的垂直距离。当其中偏移距离最大的 p_i 的偏移距离大于距离阈值时，说明函数图像在此处的弯曲程度较大，也说明相机在此时刻朝向改变的趋势较大，因此将函数图像在 p_i 处进行分段。如图 2 的中间图像所示，p_j 和 p_i 为函数段的首尾节点，并令 p_{start} 更新为 p_{i+1}，令 p_{end} 更新为 p_{current}，以此增量式地在机器人运动过程中对 p_{start}，p_{end} 和 L 进行更新，通过不断计算更新后的 p_{start} 和 p_{end} 之间

所有节点相对于直线 L 的偏移量，最终能够将函数图像分为多个函数段。



注：顶部图像和中间图像展示了对函数图像根据弯曲程度进行分段的过程，底部图像展示了对函数段根据函数值的差距进行分段的过程。

图 2 角度随时间变化的函数图像分段示意图

Fig. 2 Schematic diagram of segmenting the function of angle with respect to time

函数段上每个节点的函数值代表对应时刻的角度值，函数段的函数值分为递增、递减以及恒定不变 3 种情况。针对前两种情况，当函数段首尾节点的函数值的差距大于角度阈值时，说明该函数段中存在朝向改变较大的节点，因此需要继续分段。如图 2 的底部图像所示，针对以 p_j 和 p_i 为首尾节点的函数段，依次计算函数段首节点 p_j 和后面节点的函数值的差距，在函数值的差距大于角度阈值的节点 p_k 处进行分段，并继续依次计算 p_k 和后面节点的函数值差距，不断在函数值的差距大于角度阈值的时刻进行分段，一直计算到函数段的尾结点 p_i 为止。由此能够有效挑选出函数段内朝向改变较大的节点，p_i，p_j 和 p_k 所在时刻对应的相机就是位姿图中朝向变化较大的相机。当出现函数段的函数值恒定不变的情况时，说明相机的朝向在运动过程中并没有发生变化，因此无需进行分段操作。

具体的基于相机朝向变化的增量式位姿图分段算法伪代码如算法 1 所示。

算法 1 基于相机朝向变化的增量式位姿图分段算法

输入：p_{start}，p_{end}，$p_{current}$，相机集合 FPG=$\{x_1,\cdots,x_m\}$

输出：朝向变化较大的相机集合 RPG=$\{x_1,\cdots,x_n\}$

```
1. if FPG.size()=2 then
2.     p_start←p_0
3.     p_end←p_1
4. else
5.     L=line(p_start,p_end)
6.     i=argmax_n distance(p_n,L) ∧ n∈(start,end)
7.     if distance(p_i,L)>th_distance then
8.         RPG.insert(x_start)
9.         RPG.insert(x_i)
10.        if |theta_i−theta_start|>th_theta then
11.            temp=theta_start
12.            for k=start+1:i do
13.                if |theta_k−temp|>th_theta then
14.                    RPG.insert(x_k)
15.                    temp=theta_k
16.                endif
17.            endfor
18.        endif
19.        p_start←p_i+1
20.        p_end←p_current
21.    else
22.        p_end←p_current
23.    endif
24. endif
```

其中 p_i 代表第 i 帧时刻对应的函数图像中的节点，x_i 代表第 i 帧时刻对应的位姿图中的相机节点，$distance(p_i,L)$ 代表节点 p_i 到直线 L 的垂直距离，$th\_distance$ 代表距离阈值，th_theta 代表角度阈值，$theta_i$ 代表第 i 帧时刻对应的函数值，也就是第 i 帧时刻对应的角度值。在分别对 a 和 b 随时间变化的函数图像进行分段操作后，对应的位姿图就会被分为多个轨迹段，每一个轨迹段的首尾节点就为需要进行位姿图优化的相机。由于前端视觉里程计只估计出了时间上相邻的相机之间的约束，而本文挑选出的需要进行位姿图优化的相机大部分在时间上不相邻，因此本文通过合并相邻相机之间的约束来确定需要优化的相机之间的约束，相机之间的约束包括相机之间的相对位姿变换以及协方差矩阵，则有：

$$\boldsymbol{\xi}_{kj}=\boldsymbol{\xi}_{k,k+1}\oplus\boldsymbol{\xi}_{k+1,k+2}\oplus\cdots\oplus\boldsymbol{\xi}_{j-1,j} \tag{1}$$

$$\boldsymbol{\Sigma}_{kj}=\boldsymbol{J}_{k,k+1}\boldsymbol{\Sigma}_{k,k+1}\boldsymbol{J}_{k,k+1}^{T}+\cdots+\boldsymbol{J}_{j-1,j}\boldsymbol{\Sigma}_{j-1,j}\boldsymbol{J}_{j-1,j}^{T} \tag{2}$$

其中，$\boldsymbol{\xi}$ 和 $\boldsymbol{\Sigma}$ 代表相机之间的相对位姿变换和协方差矩阵，$\oplus$ 代表李代数上的代数运算，$\boldsymbol{J}$ 代表相机之间的相对位姿变换关于相机位姿的雅可比矩阵。例如 $\boldsymbol{\xi}_{kj}$ 和 $\boldsymbol{\Sigma}_{kj}$ 分别代表相机 k 与相机 j 之间的相对位姿变换和协方差矩阵，$\boldsymbol{J}_{k,k+1}$ 代表 $\boldsymbol{\xi}_{k,k+1}$ 关于相机 k 与相机 $k+1$ 的雅可比矩阵。Smith 等[22]最早使用该方法来确定不同时刻机器人之间的关系。

由于轨迹段的首尾相机节点在参与全局优化后得到的位姿都较为准确，因此本文分别将轨迹段的首尾相机作为轨迹段内未进行全局优化的相机的参考相机。由此轨迹段内的每个相机都能够求出两个位姿，将每个相机的两个位姿的加权平均结果作为相机的最终位姿，以此降低噪声的影响，提高相机位姿的精度，并且避免了非线性优化的繁重的计算，能以较快的速度求解出较为准确的相机位姿。假设未进行全局优化的相机 m 在以相机 j 和相机 k 为首尾节点的轨迹段内，则有：

$$\boldsymbol{T}_{wm_1}=\begin{bmatrix}\boldsymbol{R}_{wm_1} & \boldsymbol{t}_{wm_1}\\ 0 & 1\end{bmatrix}=\boldsymbol{T}_{wj}^{\mathrm{opt}}*\boldsymbol{T}_{jm} \tag{3}$$

$$\boldsymbol{T}_{wm_2}=\begin{bmatrix}\boldsymbol{R}_{wm_2} & \boldsymbol{t}_{wm_2}\\ 0 & 1\end{bmatrix}=\boldsymbol{T}_{wk}^{\mathrm{opt}}*\boldsymbol{T}_{km} \tag{4}$$

其中，$\boldsymbol{T}_{wj}^{\mathrm{opt}}$ 代表相机 j 经过位姿图优化后的位姿，即相机 j 到世界坐标系 w 的相对位姿变换，$\boldsymbol{T}_{wk}^{\mathrm{opt}}$ 代表相机 k 经过位姿图优化后的位姿；$\boldsymbol{T}_{jm}$ 代表位姿图优化前相机 m 和相机 j 之间的相对位姿，$\boldsymbol{T}_{km}$ 代表位姿图优化前相机 k 和相机 m 之间的相对位姿；$\boldsymbol{T}_{wm_1}$ 和 $\boldsymbol{T}_{wm_2}$ 分别代表相机 m 以首节点相机 j 和尾节点相机 k 为参考相机计算出的位姿，$\boldsymbol{R}_{wm_1}$ 和 $\boldsymbol{t}_{wm_1}$ 分别为 $\boldsymbol{T}_{wm_1}$

的旋转矩阵和平移向量，$\boldsymbol{R}_{wm_2}$ 和 $\boldsymbol{t}_{wm_2}$ 分别为 $\boldsymbol{T}_{wm_2}$ 的旋转矩阵和平移向量。由此可以计算相机 m 的两个位姿之间的差异：

$$\Delta \boldsymbol{T}_{m_1m_2}=\begin{bmatrix}\boldsymbol{R}_{m_1m_2} & \Delta \boldsymbol{t}_{m_1m_2}\\ 0 & 1\end{bmatrix}=\boldsymbol{T}_{wm_1}^{-1}*\boldsymbol{T}_{wm_2} \tag{5}$$

其中，$\Delta \boldsymbol{R}_{m_1m_2}$ 和 $\Delta \boldsymbol{t}_{m_1m_2}$ 分别为 $\Delta \boldsymbol{T}_{m_1m_2}$ 的旋转矩阵和平移向量，相机 m 的最终位姿 $\boldsymbol{T}_{wm}^{\mathrm{opt}}$ 为：

$$\boldsymbol{T}_{wm}^{\mathrm{opt}}=\begin{bmatrix}\boldsymbol{R}_{wm}^{\mathrm{opt}} & \boldsymbol{t}_{wm}^{\mathrm{opt}}\\ 0 & 1\end{bmatrix} \tag{6}$$

$$\boldsymbol{R}_{wm}^{\mathrm{opt}}=\boldsymbol{R}_{wm_1}*Slerp(\Delta \boldsymbol{R}_{m_1m_2},\alpha) \tag{7}$$

$$\boldsymbol{t}_{wm}^{\mathrm{opt}}=\boldsymbol{t}_{wm_1}+Lerp(\boldsymbol{R}_{wm}^{\mathrm{opt}}*\Delta \boldsymbol{t}_{m_1m_2},\alpha) \tag{8}$$

其中，$\boldsymbol{R}_{wm}^{\mathrm{opt}}$ 和 $\boldsymbol{t}_{wm}^{\mathrm{opt}}$ 分别为 $\boldsymbol{T}_{wm}^{\mathrm{opt}}$ 的旋转矩阵和平移向量，$Slerp$ 代表四元数的球面线性插值，$Lerp$ 代表线性插值，α 代表插值因子。Jang 等[23]使用该方法计算两个关键帧之间的普通帧的位姿。

4 实验结果与分析

本文的实验环境为装有 Ubuntu 18.04、配置为 Inter Core i7-5500 CPU、主频 2.4GHz、8GB RAM 的计算机。为了检测本文算法的有效性，本文在 KITTI[24]，EuRoC[25] 以及 TUM[26] 数据集上进行测试评估。KITTI 数据集为汽车上搭载的相机采集的大型室外场景的视频序列，EuRoC 数据集为飞行器上搭载的相机采集的室内小场景的视频序列，TUM 数据集为手持相机采集的室内场景的视频序列，并且 3 个数据集都提供了真实轨迹数据。

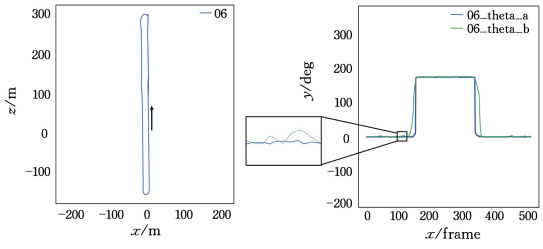
由于 ORB-SLAM2[6] 中具备位姿图优化的功能，并且只有在发生闭环时才会进行位姿图优化，因此本文分别将根据位姿图的弯曲程度把位姿图分为多个直线段的方法[16]、根据里程计距离建立局部子图的方法[19] 以及本文提出的基于相机朝向变化的增量式位姿图分段算法整合到 ORBSLAM2 中，在 KITTI，EuRoC 和 TUM 数据集中存在闭环的序列上进行实验。为了方便描述，本文将根据位姿图的弯曲程度把位姿图分为多个直线段的方法称为线段拟合法，该方法与本文算法的不同点为：该方法对位姿图进行分段时不考虑相机朝向的变化。本文将线段拟合法挑选出的相机进行位姿图优化的实验结果、局部子图法挑选出的相机进行位姿图优化的实验结果、本文算法挑选出的相机进行位姿图优化的实验结果与 ORBSLAM2 中对全体相机进行位姿图优化的实验结果进行对比。

首先，为了验证本文算法中 a 和 b 两个角度的变化可以代表相机在运动过程中的朝向变化，图 3 给出了 KITTI06 的真实轨迹与本文算法中 a 和 b 两个旋转角度在 KITTI06 序列上随时间变化的函数图像。由于 KITTI 数据集是汽车在公路上行驶的过程中采集的，因此在直行时角度 a 的值和角度 b 的值应几乎未发生变化，只有在汽车转向时才会发生明显的变化。从图 3 中可以看出：

(1)只有在第 120 个关键帧到第 150 个关键帧之间以及第 320 个关键帧到第 350 个关键帧之间的弯道行驶时，角度 a 和角度 b 发生了明显的变化，其余时刻的直线行驶时角度 a 和角度 b 几乎未发生变化。

(2)汽车在第 120 个关键帧之前以及第 350 个关键帧之后的正向行驶中，角度 a 和角度 b 接近 0 度；与之相反，汽车在第 150 关键帧到第 320 关键帧之间逆向行驶时，角度 a 和角度 b 接近 180 度。

另外，从本文算法在 KITTI06 数据集上运行时的演示视频[27] 可以看出，汽车在向前行驶时不是严格的直线运动，因此如图 3 中方框内的图像所示，向前行驶时角度 b 的值不是恒定不变的；并且行驶的公路不是完全平坦的，因此角度 a 的值不是恒定不变的。由此可以证实，角度 a 和角度 b 的变化可以代表相机在运动过程中的朝向变化。

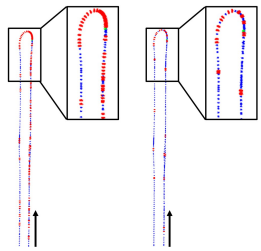


注：箭头的位置及方向为相机运动起点及方向，粗实线为角度 a 随时间变化的函数图像，细实线为角度 b 随时间变化的函数图像，方框内为放大图像。

图 3 KITTI06 的真实轨迹与 a 和 b 两个旋转角度在 KITTI06 序列上随时间变化的函数图像

Fig. 3 Real trajectory of KITTI06 and the function graph of the rotation angle a and b with respect to time on the KITTI06 sequence

此外，为了验证本文算法能够有效挑选出朝向变化较大的相机节点，图 4 为本文算法与线段拟合法在 KITTI06 序列上对位姿图分段的部分示意图。由于 KITTI 数据集在轨迹弯曲的地方一定会发生较大的朝向变化，从图 4 中可以看出，本文算法能够将轨迹弯曲处的相机节点挑选出来，证实了本文算法能够有效挑选出朝向变化较大的节点。而线段拟合法由于只使用直线段近似相机轨迹，没有考虑相机的朝向变化，因此对朝向改变不敏感且只能将轨迹弯曲处的一部分相机节点挑选出来。另外，由于汽车在向前行驶时不是严格的直线运动，并且公路是不平整的，因此也会出现朝向变化，本文算法就能够在汽车向前行驶中比线段拟合法有效挑选出更多的朝向变化较大的相机节点，最终的优化精度也会更高。



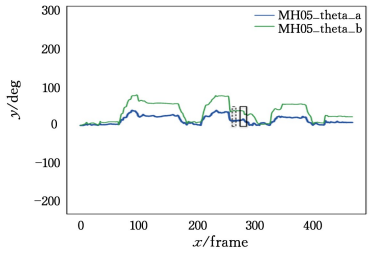
注：左图为本文算法的位姿图分段示意图，右图为线段拟合法的位姿图分段示意图；方框为轨迹的放大图像，大节点为轨迹段的首尾节点，小节点为轨迹段内的节点，箭头的位置及方向为相机的运动起点及方向。

图 4 本文算法与线段拟合法在 KITTI06 序列上对位姿图进行分段的示意图

Fig. 4 Schematic diagram of segmenting the pose graph by our algorithm and the line fitting method on KITTI06 sequence

本文同样在 EuRoC 数据集上进行测试，并提供了本文

算法在 MH05 序列上运行时对位姿图进行分段的演示视频[28]。图 5 为本文算法的 a 和 b 两个角度在 MH05 序列上随时间变化的函数图像。



注：粗实线为角度 a 随时间变化的函数图像，细实线为角度 b 随时间变化的函数图像，虚线框位于第 260 个关键帧，实线框位于第 270 个关键帧。

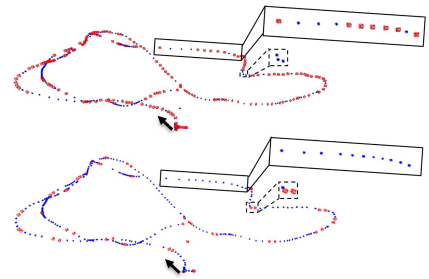
图 5 本文算法的 a 和 b 两个旋转角度在 MH05 序列上随时间变化的函数图像

Fig. 5 Function graph of the rotation angle a and b with respect to time on MH05 sequence

图 6 为本文算法与线段拟合法在 MH05 序列上对位姿图进行分段的示意图。可以看出：

(1)虚线框内角度 a 和角度 b 的函数图像几乎不发生变化，说明轨迹在发生弯曲时不一定会出现朝向的改变，因此不必对这些相机节点进行位姿图优化。本文方法能够避免将这些相机加入到位姿图优化中，而线段拟合法会在轨迹弯曲的地方进行分段操作，导致出现多优化一部分相机节点的情况，降低了优化速度。

(2)实线框内角度 a 和角度 b 的函数图像发生了较大的改变，说明相机在直线运动过程中也会出现朝向变化较大的情况。本文方法能够有效挑选出直线运动过程中朝向变化较大的相机节点进行位姿图优化，而线段拟合法只能挑选出直线运动的首尾节点，导致一部分直线运动中朝向变化较大的节点不能加入到位姿图优化中，降低了优化精度。



注：上图为本文算法的位姿图分段示意图，下图为线段拟合法的位姿图分段示意图；其中方框为轨迹的放大图像，大节点为轨迹段的首尾节点，小节点为轨迹段内的节点，箭头的位置及方向为相机运动起点及方向，虚线框内为第 260 个关键帧附近的弯曲轨迹，实线框内为第 270 个关键帧附近的直线轨迹。

图 6 本文算法与线段拟合法在 MH05 序列上对位姿图分段的示意图

Fig. 6 Schematic diagram of segmenting the pose graph by our algorithm and the line fitting method on MH05 sequence

本文也从精度和速度两个方面评估了本文算法的性能，使用均方根误差(RMSE)、均值(MEAN)、标准差(STD)来衡量真实相机位姿与优化后的相机位姿之间的误差大小，以此评判位姿优化的准确性。其中算法 1 中的距离阈值 *th_distance* 取 0.15，角度阈值 *th_theta* 取 3，插值因子取 0.5。表 1 列出了 ORBSLAM2、线段拟合法、局部子图法和本文算法优化出的相机轨迹与真实值之间的 RMSE，MEAN，STD 的对比。

表 1 本文算法、ORBSLAM2、线段拟合法、局部子图法优化出的相机轨迹与真实值之间的 RMSE，MEAN，STD 的对比

Table 1 Comparison of RMSE, MEAN and STD between trajectories optimized by OURS, ORBSLAM2, Line Fitting method, SubMap method and the real trajectory

(单位：m)

序列	RMSE				MEAN				STD			
	OURS	ORB SLAM	Line Fiting	Sub Map	OURS	ORB SLAM	Line Fiting	Sub Map	OURS	ORB SLAM	Line Fiting	Sub Map
MH_05	0.0536	0.0512	0.1268	0.0676	0.0501	0.0471	0.1064	0.0571	0.0359	0.0262	0.0502	0.0402
V1_02	0.0649	0.0623	0.0816	0.0735	0.0626	0.0596	0.0773	0.0695	0.0181	0.0171	0.0261	0.0240
V1_03	0.0779	0.0721	0.1413	0.1065	0.0698	0.0634	0.1246	0.0944	0.0346	0.0341	0.0665	0.0492
V2_01	0.0600	0.0578	0.0731	0.0693	0.0563	0.0535	0.0696	0.0594	0.0227	0.0200	0.0277	0.0254
Fr2_desk	0.0390	0.0388	0.0635	0.0412	0.0328	0.0318	0.0358	0.0341	0.0231	0.0222	0.0246	0.0236
Fr3_office	0.0170	0.0163	0.0290	0.0195	0.0167	0.0154	0.0175	0.0170	0.0062	0.0054	0.0085	0.0076
KITTI00	7.0747	6.9880	7.6619	7.3397	6.4644	6.3008	6.6814	6.6501	2.9045	2.8738	3.1740	3.0410
KITTI02	13.0960	13.0080	14.6350	13.8740	11.4860	11.4020	12.7400	12.2510	6.4356	6.1124	7.2023	7.0432
KITTI05	1.5433	1.5378	1.6941	1.6571	1.3367	1.3325	1.4900	1.4363	0.6352	0.6299	0.8593	0.7012
KITTI06	1.9059	1.8694	2.5676	2.3094	1.6486	1.6333	2.1069	1.9572	0.9735	0.8911	1.4676	1.2258
KITTI07	0.8285	0.8198	1.1064	0.9294	0.7637	0.7452	0.9785	0.8368	0.3412	0.3215	0.5164	0.4044

表 2 列出了本文算法、ORBSLAM2、线段拟合法、局部子图法的优化规模和优化时间的对比。表 1、表 2 中，KFS 代表关键帧的数量，Time 代表优化总耗时。可以看出：

(1)本文算法优化出的相机轨迹精度接近 ORBSLAM2 对全体相机进行位姿图优化后的轨迹精度，并且优于线段拟合法和局部子图法优化出的轨迹精度。

(2)相比 ORBSLAM2，本文算法的优化时间更短、优化规模更小，并且由于本文算法在对位姿图分段时考虑了相机的朝向改变，因此能够比线段拟合法有效地挑选出更多朝向变化较大的相机。

(3)本文算法在 V103 序列上的优化规模比局部子图法的优化规模小，并且本文算法的优化精度高于局部子图法的优化精度，说明通过对本文算法挑选出的朝向变化较大的相机进行优化能够有效保证优化精度。

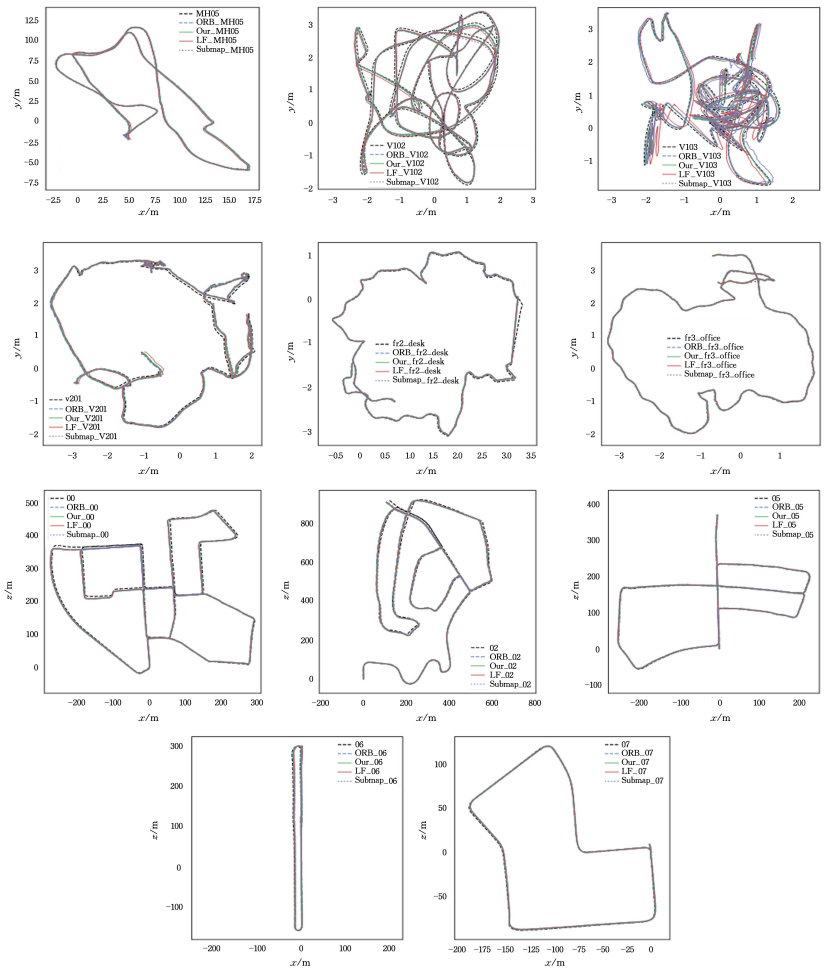
表 1 和表 2 证实了本文算法能够在减小位姿图优化规模的同时保证相机轨迹的优化精度。

表 2 本文算法、ORBSLAM2、线段拟合法、局部子图法的优化规模和优化时间对比

Table 2 Comparison of the optimization scale and optimization time of OURS, ORBSLAM2, Line Fitting method, SubMap method

序列	Loop_ID	OURS		ORBSLAM		Line Fitting		SubMap	
		KFS	Time/s	KFS	Time/s	KFS	Time/s	KFS	Time/s
MH_05	1	181	0.0610	356	0.544	54	0.0065	118	0.033
V1_02	1	46	0.0060	71	0.030	31	0.0050	35	0.005
V1_03	1	27	0.0030	83	0.016	31	0.0040	39	0.005
	2	95	0.0130	129	0.035	79	0.0070	99	0.014
V2_01	1	57	0.0070	82	0.033	32	0.0030	40	0.004
Fr2_desk	1	123	0.0160	163	0.105	81	0.0090	93	0.011
Fr3_office	1	155	0.0180	225	0.290	83	0.0110	112	0.013
KITTI00	1	195	0.0370	482	0.263	97	0.0120	159	0.029
	2	391	0.0590	752	0.476	149	0.0290	286	0.049
	3	441	0.0840	988	0.593	206	0.0480	316	0.063
	4	586	0.1060	1380	1.313	288	0.0380	472	0.086
KITTI02	1	614	0.0820	1597	2.306	413	0.0520	542	0.067
	2	637	0.1360	1663	2.539	420	0.0740	553	0.011
KITTI05	1	102	0.0160	348	0.177	81	0.0110	98	0.014
	2	225	0.0450	649	0.451	149	0.0270	216	0.039
	3	236	0.0590	693	0.467	155	0.0290	228	0.043
KITTI06	1	121	0.0238	415	0.310	47	0.0100	98	0.020
KITTI07	1	105	0.0120	250	0.100	59	0.0070	83	0.009

图 7 为 EuRoC，TUM 和 KITTI 数据集上本文算法优化出的轨迹、ORBSLAM2 优化出的轨迹、线段拟合法优化出的轨迹、局部子图法优化出的轨迹以及真实轨迹的对比图。

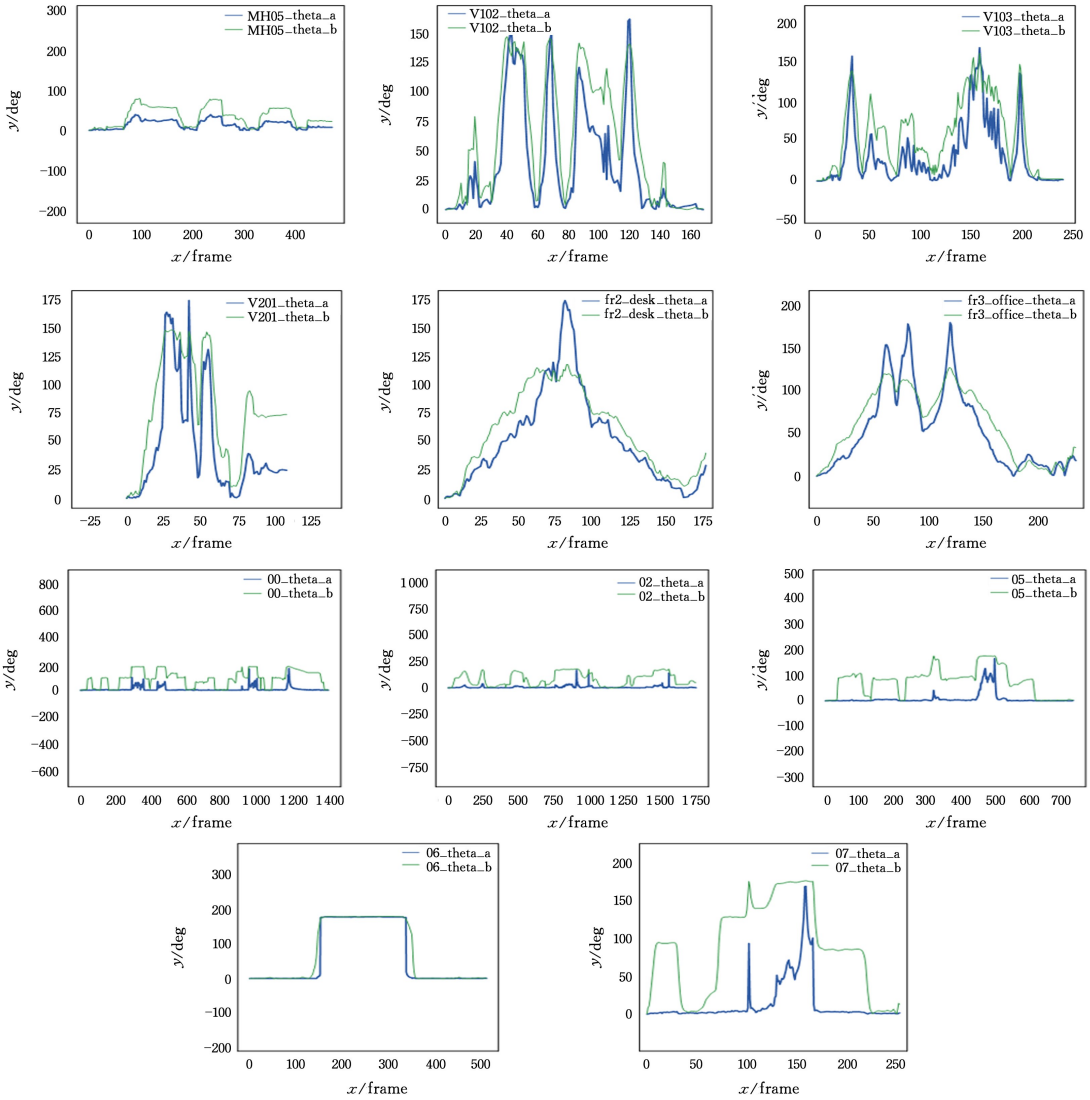


注：长细虚线为真实轨迹，长粗虚线为 ORBSLAM2 优化出的轨迹，短粗虚线为局部子图法优化出的轨迹，细实线为线段拟合法优化出的轨迹，粗实线为本文算法优化出的轨迹。

图 7 在 EuRoC，TUM 和 KITTI 数据集上本文算法、ORBSLAM2、线段拟合法、局部子图法优化出的轨迹和真实轨迹对比图

Fig. 7 Comparison of trajectories optimized by our algorithm, ORBSLAM2, line fitting method, submap method and the real trajectory on EuRoC, TUM and KITTI dataset

图 8 为本文算法的 a 和 b 两个旋转角度在 EuRoC，TUM 和 KITTI 数据集上随时间变化的函数图像。可以看出，本文算法优化出的轨迹与 ORBSLAM2 对全体相机进行位姿图优化后得到的轨迹几乎重合，并且比其他方法优化出的轨迹更接近于真实轨迹，说明了本文算法能够保证相机轨迹的优化精度。



注：粗实线为角度 a 随时间变化的函数图像，细实线为角度 b 随时间变化的函数图像。

图 8 本文算法的 a 和 b 旋转角度在 EuRoC，TUM 和 KITTI 数据集上随时间变化的函数图像

Fig. 8 Function graph of the rotation angles a and b with respect to time on EuRoC, TUM and KITTI datases

结束语 本文的主要创新点在于提出了基于相机朝向变化的增量式位姿图分段算法。该算法的主要思想是只优化朝向变化较大的相机，以此减小位姿图优化的规模，提高优化效率。本文将相机的朝向变化用两个角度的变化表示，本文算法在不限制相机运动方式的前提下，根据两个角度随时间变化的函数图像增量式地挑选出相机朝向变化较大的时刻并对位姿图进行分段，从而只对朝向变化较大的相机进行位姿图优化。此外，未进行位姿图优化的相机将参考其所在轨迹段的起始相机和终止相机估计出不同位姿，将不同位姿的加权平均结果作为该相机的最终位姿，从而在减少计算复杂度的基础上保证较高的精度。在 EuRoC，TUM 和 KITTI 数据集上进行实验，结果表明，本文算法能够减小位姿图优化的规模，并且保证相机轨迹的精度，在优化效率和准确性之间取得较好的平衡。未来将在此基础上继续研究提高优化精度的方法。

参 考 文 献

[1] MOREIRA G, MARQUES M, COSTEIRA J P. Fast pose graph optimization via Krylov-Schur and Cholesky factorization[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. IEEE, 2021: 1898-1906.

[2] TIAN Y, CHEN H W, WANG F S, et al. Overview of SLAM Algorithms for Mobile Robot [J]. Computer Science, 2021, 48(9): 223-234.

[3] FAN H Q, ZHANG S. Camera Localization and Map Reconstruction Algorithm Based on Temporal-Spatial Consistency [J]. Robot, 2019, 41(1): 40-49.

[4] WANG D, SHI C X, WANG Y Q. Loop Closure Detection Method Based on Unsupervised Deep Learning[J]. Computer Science, 2020, 47(10): 228-232.

[5] ENGEL J，KOLTUN V，CREMERS D. Direct sparse odometry [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2017，40(3)：611-625.

[6] FORSTER C，PIZZOLI M，SCARAMUZZA D. SVO：Fast semi-direct monocular visual odometry[C]// 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE，2014：15-22.

[7] MUR-ARTAL R，TARDÓS 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.

[8] SUMIKURA S，SHIBUYA M，SAKURADA K. OpenVSLAM：A versatile visual SLAM framework[C]// Proceedings of the 27th ACM International Conference on Multimedia. ACM，2019：2292-2295.

[9] LI X. Robust 6-DOF Camera Relocalization in Multi-view Structure from Motion[D]. Philadelphia：Temple University. 2021.

[10] LI X，YUAN L，LATECKI L J，et al. Pushing the Envelope of Rotation Averaging for Visual SLAM[J]. arXiv：2011. 01163，2020.

[11] SCHLEGEL D，COLOSI M，GRISETTI G. Proslam：Graph SLAM from a programmer's Perspective[C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE，2018：3833-3840.

[12] ZENG T，SI B. A brain-inspired compact cognitive mapping system[J]. Cognitive Neurodynamics，2021，15(1)：91-101.

[13] LIU T，CHEN Y，JIN Z，et al. Spare Pose Graph Decomposition and Optimization for SLAM[C]// MATEC Web of Conferences. EDP Sciences，2019，256：05003.

[14] JOHANNSSON H，KAESS M，FALLON M，et al. Temporally scalable visual SLAM using a reduced pose graph[C]// 2013 IEEE International Conference on Robotics and Automation. IEEE，2013：54-61.

[15] SCHMUCK P，CHLI M. On the redundancy detection in keyframe-based slam[C]// 2019 International Conference on 3D Vision(3DV). IEEE，2019：594-603.

[16] LATIF Y，NEIRA J. Go straight，turn right：Pose graph reduction through trajectory segmentation using line segments[C]// 2013 European Conference on Mobile Robots. IEEE，2013：144-149.

[17] ZHAO M，GUO X，SONG L，et al. A General Framework for Lifelong Localization and Mapping in Changing Environment [C]// 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS). IEEE，2021：3305-3312.

[18] DING W，HOU S，GAO H，et al. Lidar inertial odometry aided robust lidar localization system in changing city scenes[C]// 2020 IEEE International Conference on Robotics and Automation(ICRA). IEEE，2020：4322-4328.

[19] WANG Y，FUNK N，RAMEZANI M，et al. Elastic and efficient lidar reconstruction for large-scale exploration tasks[C]// 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE，2021：5035-5041.

[20] WANG C，LI J，HE Y，et al. Segmented trajectory clustering-based destination prediction in IoVs[J]. IEEE Access，2020，(8)：98999-99009.

[21] KEOGH E，CHU S，HART D，et al. An online algorithm for segmenting time series[C]// Proceedings 2001 IEEE International Conference on Data Mining. IEEE，2001：289-296.

[22] JANG Y，SHIN H，KIM H J. Pose correction algorithm for relative frames between keyframes in SLAM[C]// Proceedings of the Asian Conference on Computer Vision. 2020.

[23] GEIGER A，LENZ P，STILLER C，et al. Vision meets robotics：The kitti dataset[J]. The International Journal of Robotics Research，2013，32(11)：1231-1237.

[24] BURRI M，NIKOLIC J，GOHL P，et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research，2016，35(10)：1157-1163.

[25] WANG S J. KITTI06 Experiment [EB/OL]. (2022-03-08) [2022-04-16]. https：//youtu. be/-aZTWqphSvM.

[26] WANG S J. MH05 Experiment[EB/OL]. (2022-03-08)[2022-04-16]. https：//youtu. be/RRQFTnmzLWw.

[27] SMITH R，SELF M，CHEESEMAN P. Estimating uncertain spatial relationships in robotics[M]// Autonomous robot vehicles. New York：Springer，1990：167-193.

[28] STURM J，ENGELHARD N，ENDRES F，et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE，2012：573-580.

FAN Hanqi，born in 1983，Ph. D，associate professor. His main research interests include computer vision and visual simultaneous localization and mapping.

WANG Shaojing，born in 1995，master. His main research interest is visual simultaneous localization and mapping.

（责任编辑：喻藜）