

# VSLAM 系统中的快速特征点匹配算法

谢东, 张泽均, 谢坚筱, 周雅凯, 蒙研

(浙江师范大学 物理与电子信息工程学院, 浙江 金华 321000)

**摘要:** 连续图像帧之间特征点匹配是视觉同步定位与建图 (VSLAM) 技术的关键技术; 针对 VSLAM 系统中连续图像帧之间特征点匹配耗时且精度低的问题, 一种基于局部像素运动模型的快速特征点匹配算法 (LPMM) 被提出; 该算法基于运动平滑约束假设, 利用连续图像帧中局部区域内的像素点的运动具有一致性这一特性, 将图像划分成局部网格区域后利用每个网格中的部分特征点估算该网格区域的运动向量, 并以此为基础, 计算下一帧匹配特征点的搜索范围的中心点; 最终在计算出下一帧的中心点的局部邻域内搜索与当前帧的特征点相匹配的特征点; 实验结果表明, 所提出的匹配算法与广泛使用的 ORB-SLAM2 中的投影匹配算法相比, 平均匹配速度提高 50% 以上, 匹配精度提升了约 4%。

**关键词:** VSLAM; 特征点匹配; 局部像素运动模型; 局部区域网格

## Fast Feature Point Matching Algorithm in VSLAM System

XIE Dong, ZHANG Zejun, XIE Jianxiao, ZHOU Yakai, MENG Yan

(School of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321000, China)

**Abstract:** Feature point matching between consecutive image frames is a key technology in visual simultaneous localization and mapping (VSLAM). To address the issues of time-consuming and low accuracy in feature point matching between consecutive image frames in VSLAM systems, a fast feature point matching algorithm based on local pixel motion model (LPMM) is proposed. Based on the assumption of motion smoothness constraint, this algorithm adopts the consistency of pixel motion within local regions in consecutive image frames to divide the image into local grid regions, and estimate the motion vector of each grid by using some feature points within the grid. On this basis, the center point of the search range for matching feature points in the next frame is calculated. Finally, the feature points that match the current frame's feature points are searched for within the local neighborhood of the calculated center point of the next frame. Experimental results show that the proposed matching algorithm, compared with the widely used projection matching algorithm in ORB-SLAM2, has an average matching speed increase of over 50% and an accuracy improvement of approximately 4%.

**Keywords:** VSLAM; feature point matching; local pixel motion model; local region grid

## 0 引言

视觉同步定位与建图 (VSLAM, visual simultaneous localization and mapping) 技术通过摄像头采集环境数据, 进而实现实时定位与建图, 且成本低<sup>[1]</sup>, 被广泛应用于移动机器人、自动驾驶汽车、无人机、物理视频游戏等领域<sup>[2]</sup>。近年来, 出现了许多优秀的 VSLAM 系统<sup>[3-8]</sup>, 其中被广泛使用的是 ORB-SLAM2<sup>[6]</sup> 和 ORB-SLAM3<sup>[7]</sup>。视觉 SLAM 的前端通过相机信息来估计自身运动姿态<sup>[9]</sup>, 其关键技术之一是连续图像帧之间的特征点匹配算法, 它对整个 VSLAM 系统的效率和精度都起着决定性作用。

特征点匹配问题本质上是两个集合的二次分配问题, 该问题的求解通常是一个 NP 难问题<sup>[10]</sup>。在 VSLAM 系统中, 常用的特征匹配算法有: 暴力匹配 (BF, brute force) 和快速最近邻搜索 (FLANN)<sup>[11]</sup> 算法和投影匹配 (Projec-

tion)<sup>[6-7]</sup>。BF 方法通过遍历所有特征点来搜索最相似的特征点作为匹配特征点, 该方法原理简单, 但算法整体时间复杂度为  $O(N^2)$ , 匹配过程比较耗时, 而且会产生较多的错误匹配导致整个 VSLAM 系统性能下降。FLANN 匹配算法使用随机 kd-树<sup>[12]</sup> 来加速匹配过程<sup>[13]</sup>。该方法在大规模数据中具有明显的加速匹配效果, 但其训练搜索数据过程十分耗时, 不利于连续图像帧之间的特征匹配。BF 和 FLANN 算法中产生的错误匹配结果, 需要利用其他剔除方法<sup>[14-16]</sup> 将其剔除。VSLAM 系统中投影匹配算法利用系统中估算的位姿信息和地图点计算特征匹配时的搜索范围, 该方法通过缩小特征点匹配的搜索空间来提高匹配速度, 但其匹配性能受限于位姿信息与地图点的精度, 同时, 每次匹配都需要重新计算投影矩阵, 这影响整个 VSLAM 系统的效率。近年来, 一些基于深度神经网络技术的特征点匹

收稿日期: 2024-04-22; 修回日期: 2024-05-07。

基金项目: 浙江省自然科学基金项目 (LZ24F030009)。

作者简介: 谢东 (1998-), 男, 硕士研究生。

通讯作者: 张泽均 (1984-), 男, 博士, 讲师。

引用格式: 谢东, 张泽均, 谢坚筱, 等. VSLAM 系统中的快速特征点匹配算法[J]. 计算机测量与控制, 2024, 32(10): 222-227.

配的算法<sup>[8,10,17]</sup>被提出, 作为代表性的是文献 [10] 提出的一种基于学习的特征匹配网络 SuperGlue<sup>[10]</sup>, 和文献 [17] 提出的一种深度网络 LightGlue<sup>[17]</sup>。其中 SuperGlue 利用图像对的端到端训练来学习 3D 世界的几何变换和规律性的先验知识并通过共同寻找对应点并拒绝不可匹配的点对来匹配两组局部特征。并且在网络中同时引入了一种基于注意力的灵活上下文聚合机制, 使 SuperGlue 能够联合推理底层 3D 场景和特征分配。而 LightGlue 在 SuperGlue 的基础上通过对架构进行简单而有效的修改, 提出了训练高性能深度特征匹配器的方法, 在准确性、效率和训练易用性方面优于现有的 SuperGlue。LightGlue 具有自适应的特性, 可以根据图像对的难度进行灵活调整, 并通过预测对应关系并允许模型自省, 可以在易于匹配的图像对上实现更快的推理速度, 而在具有挑战性的图像对上仍然保持准确性。虽然 SuperGlue 与 LightGlue 的性能优于其他学习方法, 并在具有挑战性的现实室内和室外环境中的姿态估计任务中取得了先进的结果。但其推理过程需要依赖于现代 gpu, 无法做到仅在 CPU 上运行, 昂贵的计算成本使其在具有运行实时性要求同时计算资源受限的 SLAM 系统中无法满足要求。基于深度神经网络的匹配算法目前还难以成功运用在 VSLAM 系统中。

在 VSLAM 系统中, 由于连续图像帧之间存在较强相关性这一特性, 使得连续图像帧之间, 局部区域内像素点的运动具有一致性; 基于此, 本文提出一种基于图像局部区域像素运动模型的快速特征点匹配算法 (LPMM, local pixel motion matching)。该算法的主要思想是利用局部区域像素的运动信息, 缩小特征点匹配的搜索空间, 提高特征点匹配的速度。在公开数据集上的实验结果表明, 所提出的匹配算法在保证 VSLAM 定位精度的同时, 显著提升了连续图像帧之间特征匹配的性能。

## 1 VSLAM 系统与问题描述

### 1.1 VSLAM 系统

VSLAM 系统的一般架构如图 1 所示, 主要包括 4 大组成部分: 前端视觉里程计、后端优化、回环检测和建图模块。前端负责处理图像帧并估算相机姿态; 所得姿态数据服务于后续的回环检测、优化及建图环节。回环检测利用图像数据和姿态信息查找闭环线索。后端承担地图与轨迹优化任务, 解决关联性和一致性问题, 提高定位和建图精度。通过优化, 确保地图和轨迹与真实环境相符, 增强 SLAM 系统的整体效能。建图模块则致力于从图像信息中还原场景三维结构, 并管理地图数据。

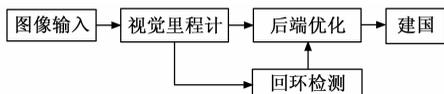


图 1 VSLAM 系统架构图

输入的每一帧图片提取特征点, 并根据特征匹配进行数据关联, 最后估计出相机的位姿信息, 因此, 前端根据特征匹配估计出的位姿信息好坏将直接影响整个 VSLAM 系统的定位精度与鲁棒性。

### 1.2 VSLAM 系统中连续帧之间的特征点匹配问题

图像中的特征点匹配是计算机视觉和图像处理中的重要任务, 用于在不同图像或图像中的不同区域之间找到相对应的特征点或特征描述符。特征匹配作为图像配准、目标跟踪、三维重建等许多计算机视觉任务的基础, 在实际应用中具有重要意义。一般特征匹配的详细流程如图 2 所示。

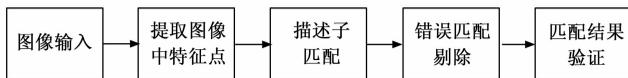


图 2 特征匹配一般过程

首先是从输入的每幅图像中提取特征点或特征描述符, 然后对提取出的特征点或描述符, 通过使用距离度量 (如欧氏距离、汉明距离等) 来计算特征之间的相似度进行匹配。匹配得到的特征点可能包含一些错误匹配或者噪声点, 因此需要进行特征筛选。常用的筛选方法包括基于阈值的筛选、基于几何一致性的筛选等。对于筛选后的匹配结果, 通常会采用几何关系的验证 (如基本矩阵或单应性矩阵的计算和验证) 或基于运动模型的验证, 进行验证以确保匹配的准确性。

在 VSLAM 系统中连续图像帧之间的特征匹配任务区别于普通场景中的两张图像之间的匹配任务, 主要解决的是数据关联问题, 不仅具有很高的实时性要求, 而且存在一些先验信息可以被利用。如图 3 所示, 图中  $X, Y, Z$  所构成的坐标系为 VSLAM 系统中定义的世界坐标系, 其中左边为  $t-1$  时刻相机所拍摄的图像  $I_{t-1}$ , 右边为当前  $t$  时刻相机拍摄的图像  $I_t$ 。假设在  $I_{t-1}$  图像中提取  $X$  个特征点  $F_x (x = 1, 2, \dots, X)$ , 在  $I_t$  时刻的图像中提取到  $Y$  个特征点  $F'_y (y = 1, 2, 3, \dots, Y)$ 。对于当前帧, 在 VSLAM 中已经存在  $K$  个地图点  $M_k (k = 1, 2, \dots, K)$ , 需要解决的问题是将当前图像帧  $I_t$  中提取的特征点  $F'_y$  与上一帧图像  $I_{t-1}$  中特征点  $F_x$  所对应地图点进行关联, 进而实现图像帧  $I_t$  与  $I_{t-1}$  之间的特征匹配。

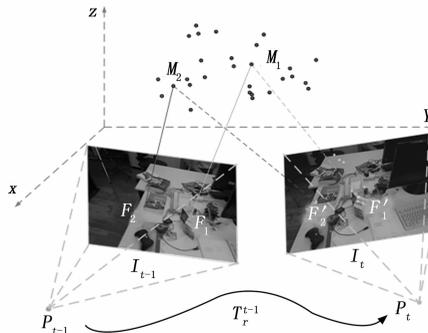


图 3 VSLAM 中数据关联

在间接法的 VSLAM 系统中, 前端视觉里程计需要对

在连续图像帧的匹配中， $I_{t-1}$  帧的位姿已知且当前帧  $I_t$  的位姿可以被预测出来。假设  $I_{t-1}$  帧的位姿为  $P_{t-1}$ ，前两帧  $I_{t-2}$  到  $I_{t-1}$  之间的相对位姿变换为  $T_r^{-1}$ ；那么当前帧  $I_t$  在世界坐标系中的位姿  $P_t$  可以根据如下公式估算出来：

$$P_t = T_r^{(t-1)} \cdot P_{t-1} \quad (1)$$

同时，由于在 VSLAM 中相邻图像帧之间的运动变换很小，因此相邻图像帧之间存在大量的相似局部结构<sup>[18]</sup>可以被利用于加速特征点匹配过程。

## 2 基于局部像素运动特征匹配算法

### 2.1 运动平滑约束假设

运动平滑度约束假设<sup>[15]</sup>描述的是在一幅图像的局部结构中相邻像素会同时移动，因为它们通常落在一个刚性物体或结构中。因此，在静态场景中，连续图像帧中刚体物体或者结构区域内的像素点均满足这一假设。由于 VSLAM 系统运行环境大多为静态场景，而场景中的物体绝大部分为刚体物体，因此，这一假设可以广泛应用到 VSLAM 系统中，以提高整个系统的性能。

如图 4 所示，图中左边为上一帧的图像，右边为当前帧的图像，其中矩形区域内表示相邻两帧之间提取到的特征点匹配关系。从图中可以看出，上一帧中矩形区域内的像素相机发生运动后出现在当前帧中仍然保留了局部结构相似性，因此该区域内所有的特征点位置也具有稳定的局部结构相似性。



图 4 局部结构相似性

### 2.2 局部区域运动向量估计

为了估算图像中，局部区域的运动向量，以此为基础，确定特征点匹配的搜索范围。首先，将图像划分成  $M$  个网格， $G_i$  ( $i=1, 2, \dots, M$ )，网格  $G_i$  是平面坐标点的集合  $G_i = \{(x, y)\}$ 。如图 5 所示，根据运动平滑约束假设，上一帧  $I_{t-1}$  中每个网格中的局部结构会出现在下一帧  $I_t$  中的某个网格中。对于某个网格区域，选取其中几个特征点为基础，来估算该网格区域的运动向量，这些特征点称为该网格区域的锚点。

假设上一帧中某个网格  $G_i$  内特征点  $F$  与地图点  $M$  已经关联好。通过如下公式计算  $F$  投影出现在当前帧中的位置  $(u, v)$ ：

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot T_{cw} \cdot P_w \quad (2)$$

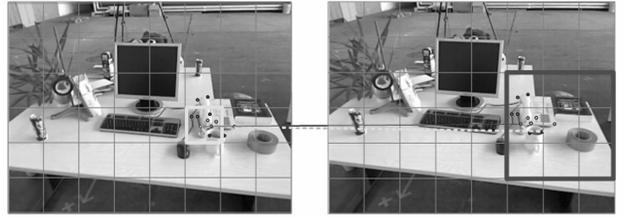


图 5 网格划分

其中： $P_w$  表示地图点  $M$  的世界坐标， $T_{cw}$  表示当前帧相机从世界坐标系到相机坐标系的变换矩阵，即相机的位姿， $K$  表示相机内参由相机厂家提供， $Z$  表示特征点在真实世界中所对应的点在相机坐标系中的距离。得到特征点  $F$  在当前帧中坐标  $(u'_i, v'_i)$  后，对比以  $(u'_i, v'_i)$  为中心，半径为  $R$  的邻域内所有特征点，选择与特征点  $F$  描述子的汉明距离最小的特征点作为匹配特征点。成功匹配的特征点，将被作为整个网格的锚点计算网格内的像素运动情况。

假设上一帧图像  $I_{t-1}$  中某个网格  $G_i$  ( $i=1, 2, \dots, M$ ) 中特征点为  $\{F_1, F_2, \dots, F_n\}$ ，在当前帧图像  $I_t$  中与它匹配的特征点为  $\{F'_1, F'_2, \dots, F'_n\}$ 。图 6 显示了上一帧图像  $I_{t-1}$  与当前帧图像  $I_t$  之间某个网格内的锚点匹配情况。假设  $I_{t-1}$  中该网格内的锚点  $F_i$  在  $I_t$  中匹配到的特征点  $F'_i$ ， $F_i$  与  $F'_i$  之间箭头表示匹配特征点对之间的像素运动向量，其匹配特征点对  $F_i$  和  $F'_i$  之间的像素运动向量的计算方法如下：

$$a_i = [u'_i - u, v'_i - v] \quad (3)$$

其中： $(u, v)$  和  $(u'_i, v'_i)$  分别表示  $I_{t-1}$  和  $I_t$  中相匹配的特征点  $F_i$  和  $F'_i$  的像素坐标。

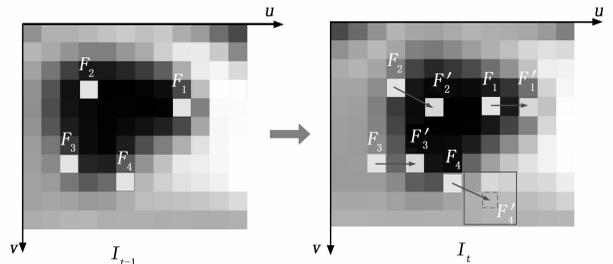


图 6 运动向量计算

网格  $G_i$  的运动向量  $a$  用该网格内锚点运动向量的加权平均估算，其计算方法如下：

$$a = \frac{\sum_{i=0}^n \omega_i a_i}{\sum_{i=0}^n \omega_i} \quad (n=1, 2, \dots, N) \quad (4)$$

权值  $\omega_i$  的计算式如下：

$$\omega_i = \theta - d_i \quad (5)$$

其中： $\theta$  表示阈值， $d_i$  表示与运动向量  $s$  关联的两个特征点的特征描述子之间的汉明距离。

基于式 (2) 和 (3) 计算的网格  $G_i$  的运动向量  $a$ ，利用刚性运动平滑约束的假设，可以估计连续视图像中网格

$G_i$  内其他特征点的像素运动信息。已知  $I_{t-1}$  中某特征点的像素坐标  $(u, v)$ , 估算该特征点在  $I_t$  中的位置  $(u', v')$  的方法如下:

$$[u', v'] = [u, v] + \mathbf{a} \quad (6)$$

本文提出的算法主要是通过利用局部结构内的像素运动向量式 (6), 来快速搜索待匹配特征点, 与 ORB-SLAM2 的基于投影计算的方法不一样, 不需要所有特征点均通过复杂的投影方式来计算出现在当前帧中的位置, 从而大幅提升的算法的运行效率。

### 2.3 基于局部运动信息的特征点快速匹配算法

VSLAM 系统中连续图像帧的特征点匹配算法的目的是: 将上一图像帧  $I_{t-1}$  中的特征点  $F(u, v)$  与当前图像帧  $I_t$  中的特征点  $F_i(u'_i, v'_i)$  进行特征匹配。本文提出的快速匹配算法的主要思想如下:

1) 将上一图像帧  $I_{t-1}$  中的特征点进行网格划分, 对于某个网格  $G_i$ , 利用公式 (2) ~ (4) 计算该网格的像素运动向量  $\mathbf{a}$ ;

2) 对于网格  $G_i$  中的特征点  $F_i$ , 利用公式 (5) 计算其在当前图像  $I_t$  中的像素坐标  $(u', v')$ ;

3) 在视频帧  $I_t$  中, 对以  $(u', v')$  为中心,  $r$  为半径的邻域内, 搜索与特征点  $F_i$  关联的候选特征点。

4) 在与特征点  $F_i$  相关联的候选特征点集合中搜索与其特征描述子距离最小的特征点作为匹配特征点。

基于局部运动信息的特征点快速匹配算法描述如下:

算法 1:

输入: 上一帧特征点集:  $F_x$ , 当前帧特征点集:  $F'_x$ , 与  $F_x$  中特征点关联的地图点集:  $M_k$ , 邻域半径  $r, \theta$

输出: 特征点匹配关系:  $R$

初始化: 图像划分成  $80 \times 80$  像素大小的  $M$  网格  $G_i (i=1, 2, \dots, M)$ ;  $R = \varphi$ ; 候选特征点集:  $C = \varphi$

Foreach  $G_i$  do:

利用投影匹配方法获取  $G_i$  中  $N$  个特征匹配对作为锚点

利用公式估算该网格中特征点的像素运动向量  $\vec{a}$ ;

Foreach  $F_i$  in  $G_i$  do:

根据公式通过  $\vec{a}$  计算出待匹配特征点出现在当前帧中的位置  $(u', v')$

Endfor

$C \leftarrow$  获取  $F'_x$  中以  $(u', v')$  为中心, 围半径为  $r$  圆域内所有候选特征点

$R \leftarrow$  获取  $C$  中与  $F_i$  描述子距离最近特征点的匹配关系。

Endfor

本文提出的基于局部运动信息的快速匹配算法中, 为了权衡计算网格运动向量的精度与效率, 公式中计算网格的运动向量时, 本文所有实验中,  $N$  取 3。公式 (4) 中, 阈值  $\theta$  的取值为 16。

## 3 实验结果与分析

### 3.1 实验数据集与环境

实验数据集是 EuRoC<sup>[19]</sup> 和 TUM<sup>[20]</sup> 中 RGB-D 数据集,

其中 EuRoC 数据集是由瑞士苏黎世联邦理工学院 (ETH Zurich) 的机器人系统实验室 (ASL) 创建和维护的一个开放数据集。该数据集旨在为视觉定位和 SLAM 算法的研究和评估提供丰富的实验数据。EuRoC 数据集共包含 11 个双目序列, 这些序列均是由无人机在两个不同的房间和一个大型工业环境中飞行时记录下来的, 并且, 无人机的真实轨迹由动作捕捉系统 (Vicon 和 Leica MS50) 提供。在评估 SLAM (Simultaneous Localization and Mapping) 算法效能时, 充分考虑不同场景复杂度是非常重要的。EuRoC 数据集正是根据场景的纹理丰富程度、光照条件的变化幅度, 以及无人机在数据采集过程中经历的剧烈姿态变化、高速移动等因素, 将数据集划分为 easy (简单)、medium (中等) 和 difficult (困难) 3 种难度等级, 以适应不同 SLAM 算法在各种复杂环境下的性能测试和对比评估。通过这种方式, 研究人员可以更好地衡量算法在不同场景复杂度下的稳健性和准确性。而 TUM RGB-D 数据集由一系列运用 Microsoft Kinect 传感器在多元化的室内环境中录制的 39 个独立序列构成, 其内容涵盖了多种应用场景, 包括但不限于: 手持设备实现的 SLAM 技术、机器人实施的 SLAM 算法、动态目标追踪、三维物体重建以及数据验证和校准等多个子集。每个类别内部又细分了多个数据样本, 这些样本适用于各类任务的性能评估与测试。TUM 除了验证集, 其他数据集均带有标准的轨迹和一些比较工具。

本文将所提出的特征匹配算法集成在 ORB-SLAM2 系统中进行测试。ORB-SLAM2 是首个集成了单目、双目、RGB-D 摄像头支持的开源 VSLAM 解决方案, 且具备闭环检测、重定位和地图复用等功能。整个 ORB-SLAM2 系统主要由跟踪、局部建图、回环检测三大线程构成。VSLAM 的前端主要是由跟踪线程完成, 该线程的主要任务是接收相机输入的图像数据进行世界坐标系的初始化以及求解每一帧相机的位姿。对于采用 RGB-D 相机和双目立体相机作为传感器时, 可以直接利用测量得到的深度值进行初始化。在初始化完成后, 跟踪线程开始对每一帧图像进行特征提取, 和跟踪特征点。在跟踪阶段, 系统采用了 3 种不同跟踪策略: 恒速模型跟踪、参考关键帧跟踪和重定位跟踪。其中恒速模型跟踪, 是基于短时间内恒定的相机速度进行粗略预测, 以提高跟踪效果, 在大部分情况下, 系统采用恒速模型跟踪, 因此在实验中将 LPMM 匹配算法集成到恒速跟踪模型中进行测试。成功进行帧间跟踪后, 系统进入局部地图跟踪阶段, 利用共视关系通过最小化重投影误差来优化当前帧的位姿, 提高位姿估计结果的准确性。如果帧间跟踪或局部跟踪失败, 系统会根据情况进入重定位阶段。成功进行两次跟踪后, 系统判断是否可以建立关键帧, 然后将关键帧插入局部建图线程和地图中。

所有实验均在同一台笔记本电脑上进行, 其系统配置为: Intel G4560 CPU (双核四线程, 3.5 GHz), 16 G 的 ddr4 内存, Ubuntu20.04 系统。

### 3.2 特征点匹配结果与分析

本文对比了几种常用的特征匹配算法在 ORB-SLAM2 中的匹配效果。如图 7 所示,从图中可以看出 BF 和 FLANN 匹配均存在大量的错误匹配,需要后续结合其他算法进一步提出其中错误匹配才能建立良好的匹配关系;而投影匹配和本文所提出的 LPMM 匹配均能获得良好的匹配效果,且其中 LPMM 匹配的特征点数目要略多于投影匹配。

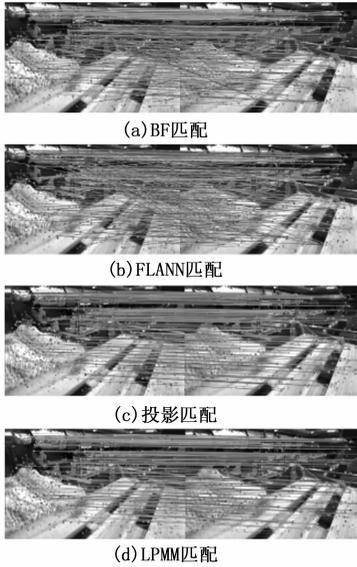


图 7 不同算法匹配效果对比

为了有效地衡量匹配算法的性能,本文进一步从匹配的精确率<sup>[21]</sup>和时间两个方面对算法进行评估。表 1 中给出了不同特征点数目情况下,4 种匹配算法的精度对比。实验数据是数据集 EuRoc 中的 Mh\_01\_easy 测试序列,其中特征点数从 500~3 000。表 1 中显示,随着特征点数目的增加,4 种方法的匹配精准率都略有下降,相比之下,本文提出的 LPMM 方法始终保持最高的匹配精准率,平均匹配精准率为 90.11%。

在 SLAM 系统中,除了定位精度,实时性也是很重要的一方面。表 2 中给出了 4 种匹配方法在不同特征点数目下平均每帧匹配所花费的时间。从表 2 中看出 LPMM 在不同特征点数目下均能获得较高的匹配速度,平均每帧所需的匹

表 1 不同特征点数目下正确匹配特征点数目和匹配精准率

特征点数目	Projection	BF	FLANN	LPMM(ours)
	correct, (precise/%)	correct, (precise/%)	correct, (precise/%)	correct, (precise/%)
500	77,(93.90)	73,(62.39)	56,(47.86)	78,(96.30)
1 000	195,(89.86)	174,(58.59)	128,(43.10)	197,(92.92)
1 500	272,(86.08)	256,(62.75)	178,(43.63)	278,(90.85)
2 000	323,(83.90)	306,(63.75)	213,(44.38)	326,(88.35)
2 500	421,(81.59)	399,(63.43)	268,(42.61)	427,(86.09)
3 000	505,(81.58)	482,(66.48)	320,(44.14)	509,(86.13)
Average	299(86.15)	282(62.90)	194(44.29)	303(90.11)

表 2 不同特征点数目下平均每一帧匹配时间 ms

特征点数目(地图点)	Projection	BF	FLANN	LPMM
500 (117)	0.342	0.768	2.965	0.293
1 000 (297)	0.798	3.107	5.717	0.410
1 500 (408)	1.243	7.005	8.275	0.549
2 000 (480)	1.567	11.381	10.544	0.669
2 500 (629)	2.007	17.162	13.228	0.810
3 000 (725)	2.469	23.856	15.770	0.970
Average	1.404	10.547	9.417	0.617

配时间均小于 1 ms,与 ORB-SLAM2 系统中的匹配算法 Projection 相比,平均效率提高了 60%以上。主要原因是利用像素运动向量估计特征点出现在当前帧中的位置要比利用投影方式更加准确,且无需复杂的投影计算,实现了在较小的局部范围内搜索匹配特征点对。

### 3.3 ORB-SLAM2 运行结果与分析

表 3 中给出了在数据集 EuRoc 中,采用 4 种匹配算法的 ORB-SLAM2 系统重构的地图的数值指标。为了保证匹配 ORB-SLAM2 能够稳定地运行,4 种匹配算法在实验中均采用了基于特征点旋转一致性的错误匹配剔除,即统计所有特征匹配对的两个特征点方向夹角,保留统计数目前 3 个的夹角范围所对应的特征点。表中 Time 表示每一帧的匹配时间, RMSE 表示 ORB-SLAM2 系统重构的地图轨迹与轨迹真值之间旋转部分加上平移部分的绝对轨迹均方根误差。每个序列的特征点数目均为 2 000。根据表格 3,在测试的

表 3 平均每一帧匹配时间 (ms) 和 RMSE 对比

序列	Projection Time/RMSE	BF Time/RMSE	FLANN Time/RMSE	LPMM Time/RMSE
MH_01_easy	1.655/1.984 8	11.736/1.986 7	10.752/1.985 5	0.752/1.984 6
MH_02_easy	1.587/1.983 5	11.381/1.985 2	10.579/1.984 2	0.757/1.983 1
MH_03_medium	1.723/1.983 1	12.402/1.982 6	11.007/1.982 8	0.816/1.982 8
MH_04_difficult	1.618/1.979 8	11.031/1.987 2	10.231/1.998 5	0.793/1.985 3
MH_05_difficult	1.688/1.985 3	12.200/1.984 1	10.511/1.986 4	0.822/1.985 8
V1_01_easy	1.579/1.913 8	9.342/1.914 4	9.924/1.914 8	0.791/1.914 0
V1_02_medium	1.480/1.978 3	9.425/1.977 0	9.819/1.978 1	0.747/1.978 2
V1_03_difficult	1.362/1.963 4	7.427/1.971 4	8.645/1.972 9	0.714/1.973 6
All. avg	1.587/1.971 5	10.618/1.973 6	10.183/1.975 4	0.774/1.973 4

8 个序列中采用 LPMM 的匹配方法所需的时间最少, 平均所需时间仅为 0.774 ms, 远远低于 BF 所需的 10.618 ms 和 FLANN 的 10.183 ms。采用 LPMM 的匹配方法平均处理每一帧的效率相比 Projection 提升了 50%。对于定位精度, 我们测试的 4 种方法在不同的序列中虽然各有优劣, 但定位精度都相差不大。影响定位精度的因素有几个: 最大特征点数目, 运行过程中正确跟踪到地图点的数目, 和是否产生闭环。在运行过程中我们没有关闭 ORB-SLAM2 中的闭环检测模块和 BA 模块, 最终得到的位姿均是经过 BA 优化后的位姿。

#### 4 结束语

本文基于连续视频帧之间的局部像素运动信息, 提出了一种适用于 V-SLAM 系统的快速特征匹配算法, 有效解决了基于特征点匹配的 SLAM 系统中, 特征点匹配耗时且匹配精度不足的问题。算法在提升匹配精度的同时也大幅提升匹配速度。实验结果表明, 在 ORB-SLAM2 系统中与常用匹配算法相比, 本文提出算法在保证整个 SLAM 系统建图精度的同时减少了 50% 以上特征匹配所消耗的时间。本文所提出的算法为实现 SLAM 在低功耗设备上实时运行提供了良好的基础。本文的算法在锚点获取时借助了图像帧的位姿信息进行投影, 从而限制了其应用场景, 下一步将改进锚点获取方式使之能够适用于普通图像配准任务, 进行快速特征匹配。

#### 参考文献:

[1] 权美香, 朴松昊, 李 国. 视觉 SLAM 综述 [J]. 智能系统学报, 2016, 11 (6): 768-776.

[2] ABASPUR KAZEROUNI I, FITZGERALD L, DOOLY G, et al. A survey of state-of-the-art on visual SLAM [J]. Expert Systems with Applications, 2022, 205: 117734.

[3] NEWCOMBE R A, LOVEGROVE S J, DAVISON A J. DTAM: Dense tracking and mapping in real-time [C] //2011 International Conference on Computer Vision, 2011: 2320-2327.

[4] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces [C] //2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007: 225-234.

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

[6] MUR-ARTAL R, TARDÓ S 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.

[7] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM [J]. IEEE Transactions on Robotics,

2021, 37 (6): 1874-1890.

[8] TANG J, ERICSON L, FOLKESSON J, et al. GCNv2: efficient correspondence prediction for real-time SLAM [J]. IEEE Robotics and Automation Letters, 2019, 4 (4): 3505-3512.

[9] 行芳仪, 徐 成, 高宏伟. 高效高精度光照自适应的 ORB 特征匹配算法 [J]. 电子测量与仪器学报, 2023, 37 (7): 140-147.

[10] SARLIN P E, DETONE D, MALISIEWICZ T, et al. SuperGlue: learning feature matching with graph neural networks [C] //2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, 2020: 4937-4946.

[11] HAJEBI K, ABBASI-YADKORI Y, SHAHBAZI H, et al. Fast approximate nearest-neighbor search with k-nearest neighbor graph [C] //Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. Barcelona, Catalonia, Spain: AAAI Press, 2011: 1312-1317.

[12] TIAN W, WANG Z, ZHU Q. A research of feature matching algorithm based on monocular vision SLAM [C] //2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), 2019: 12-16.

[13] SILPA-ANAN C, HARTLEY R. Optimised KD-trees for fast image descriptor matching [C] //2008 IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, AK, USA: IEEE, 2008: 1-8.

[14] 张旭辉, 杨红强, 白琳娜, 等. 基于改进 RANSAC 特征提取的掘进装备视觉定位方法研究 [J]. 仪器仪表学报, 2022, 43 (12): 168-177.

[15] BIAN J W, LIN W Y, LIU Y, et al. GMS: grid-based motion statistics for fast, ultra-robust feature correspondence [J]. International Journal of Computer Vision, 2020, 128 (6): 1580-1593.

[16] MA J, ZHAO J, JIANG J, et al. Locality preserving matching [J]. International Journal of Computer Vision, 2019, 127 (5): 512-531.

[17] LINDENBERGER, PHILIPP. LightGlue: Local Feature Matching at Light Speed [C] //2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023: 17581-17592.

[18] SHAO F, LIU Z, AN J. A discriminative point matching algorithm based on local structure consensus constraint [J]. IEEE Geoscience and Remote Sensing Letters, 2021, 18 (8): 1366-1370.

[19] 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.

[20] 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, 2012: 573-580.

[21] 崔建国, 孙长库, 李玉鹏, 等. 基于 SURF 的快速图像匹配改进算法 [J]. 仪器仪表学报, 2022, 43 (8): 47-53.