

基于深度学习的语义特征点提取匹配算法

杜小双, 施 展

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2023年4月16日; 录用日期: 2023年6月14日; 发布日期: 2023年6月30日

摘 要

针对传统ORB算法在特征点提取匹配方面存在的误提取和误匹配问题, 以及在语义SLAM应用方面的不足, 本文提出了一种基于深度学习的语义特征点提取匹配算法。首先, 利用语义分割网络获取匹配图像的语义信息, 再结合四叉树分配及自适应提取阈值算法, 以实现提取均匀分布的语义特征点。其次, 为了剔除特征点匹配环节中存在的误匹配, 本文设计了一种语义角度直方图匹配优化算法, 通过结合匹配语义特征点的角度和语义信息用于剔除误匹配, 从而获得准确的匹配特征点对。最后, 通过对比实验评估本文所提算法, 实验结果表明论文提出的语义特征点算法能够有效消除误匹配并实现提取特征点的均匀分布, 证明了本文算法在多种场景下的优越性能, 满足语义SLAM系统的需求。

关键词

特征点算法, 深度学习, 四叉树分配, 特征点匹配优化

Deep Learning-Based Semantic Feature Point Extraction and Matching Algorithm

Xiaoshuang Du, Zhan Shi

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 16th, 2023; accepted: Jun. 14th, 2023; published: Jun. 30th, 2023

Abstract

To address the issues of mis-extraction and mis-matching in traditional ORB algorithms for feature point extraction and matching, as well as the inadequacies in semantic SLAM applications, this paper proposes a semantic feature point extraction and matching algorithm based on deep learning. Firstly, the semantic information of matching images is obtained using a semantic segmentation network, and then combined with a quad-tree allocation and adaptive extraction thre-

should algorithm to achieve uniformly distributed semantic feature points. Secondly, to eliminate mis-matches in the feature point matching process, a semantic angle histogram matching optimization algorithm is designed, which combines the angle and semantic information of matching semantic feature points to eliminate mis-matches, thus obtaining accurate matching feature point pairs. Finally, through comparative experiments, the results demonstrate that the proposed semantic feature point algorithm effectively eliminates mis-matches and achieves a uniform distribution of extracted feature points, proving the superior performance of the algorithm in various scenarios and meeting the requirements of semantic SLAM systems.

Keywords

Feature Point Algorithm, Deep Learning, Quadtree Allocation, Feature Point Matching Optimization

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着计算机视觉和图像处理技术的飞速发展, 图像特征匹配在许多领域, 如视觉同步定位与地图构建(Simultaneous Localization and Mapping, SLAM) [1]、立体视觉、图像拼接、物体识别等, 起着至关重要的作用。尤其在视觉中, 优秀的图像特征匹配算法是正确求解相机位姿的关键[2]。图像匹配主要分为灰度匹配和特征点匹配[3], 其中基于特征点的匹配方法具有良好的稳定性, 对图像尺度、旋转、光照变化有良好的鲁棒性, 成为了当前视觉 SLAM 中主流的图像匹配方法。

对特征的提取与描述是特征点算法的关键所在, 对此的研究在很早就开展了, 在 2004 年由 Lowe 等[4]学者首先提出尺度不变特征匹配算法(Scale Invariant Feature Transform, SIFT), 利用高斯差分模板构建线性金字塔尺度空间并从中提取具有尺度不变性的极值点, 通过图像梯度直方图构建 128 维描述符, 但计算复杂度太高, 导致 SIFT 算法的匹配时间较慢。因此, Bay 等人在 2006 年[5]提出了加速稳健特征(Speeded up Robust Features, SURF)算法, 该算法使用盒状滤波器模板简化高斯尺度空间来改进 SIFT 算法在保证算法鲁棒性, 并将描述符从 128 维降低到 64 维, 提高了算法匹配效率。2011 年, Rublee 等[6]对加速测试特征点检测算法[7] (Features from Accelerated Segment Test, FAST)和二进制鲁棒独立特征描述子[8] (Binary Robust Independent Elementary Features, BRIEF)进行改进, 提出了具有旋转不变性的特征点提取算法 (Oriented FAST and Rotated BRIEF, ORB), 该算法采用二进制描述向量并结合汉明距离实现图像快速匹配, 实验表明, ORB 算法的匹配速度比 SURF、SIFT 算法相比提速 10 倍以上[9]。

除了这些经典的特征点算法外, 在其上进行改进的特征点算法研究也在快速发展, 在 2017 年, 由 Mur-Artal 等人[10]在其提出的 ORB-SLAM 算法中首次使用四叉树分配算法改善特征点分布密集的情况, 避免特征点因聚集而导致的误匹配问题, 使特征点算法满足视觉 SLAM 需求。李伦[11]在对 SURF 特征点算法的基础上, 提出了采用渐进采样一致算法(Progressive Sample Consensus, PROSAC)剔除误匹配点对的改进算法, 该算法增强了特征点提取的稳定性, 提高了匹配正确率。李国竣等[12]从阈值入手对 ORB 算法进行改进, 通过设置不同的阈值以满足不同条件下不同图像的特征提取要求。在 2022 年, 潘峰等人[13]将 GMS 算法融入进 ORB 特征点算法中, 通过四叉树分配算法均匀提取特征点并使用, 并融合 GMS 算法剔除误匹配, 其提取匹配效果优于原始 ORB 算法。

随着深度学习和视觉 SLAM 技术的进步, 结合神经网络的语义 SLAM 应运而生并迅速发展。不包含语义信息的传统特征点算法将无法理解环境信息并消除误匹配, 将不适用于语义 SLAM 中, 因此本文提出了一种基于深度学习的语义特征点提取匹配算法, 以满足语义 SLAM 系统需求。该算法结合语义分割网络和改进四叉树分配算法, 实现特征点提取的语义化和均匀化, 并采用语义角度直方图方法筛选匹配特征点对。实验评估显示, 与传统算法相比, 论文提出的算法具有较高的匹配准确性和稳健性, 更适应语义 SLAM 系统需求, 并对提高图像匹配准确性和稳健性具有重要意义。

2. 特征点算法与神经网络理论基础

2.1. Oriented FAST 角点

ORB 算法是由 Oriented FAST 角点[14]和 Rotated BRIEF 描述子[8]相结合组成的算法, 这种算法旨在改进特征点的性能, 同时保持高度精确和鲁棒性。

Oriented FAST 角点在 FAST 角点的基础上增加了特征点对尺度和旋转的描述, FAST 角点通过比较该点与周围其他像素灰度值的差别来确定的, 具有较高的检测效率。提取如图 1 所示, 具体过程如下:

- 1) 选取像素 p , 设其点亮度值为 I_p 。
- 2) 设置亮度阈值 T , 其值取 I_p 的 20%。
- 3) 以像素 p 为中心, 选取半径为 3 像素的圆上 16 个像素点。
- 4) 遍历对比, 若所选像素点中有 N 个连续像素点的亮度值大于 $I_p + T$ 或小于 $I_p - T$, 该像素点 p 为角点。
- 5) 循环以上步骤, 遍历图像中各像素。

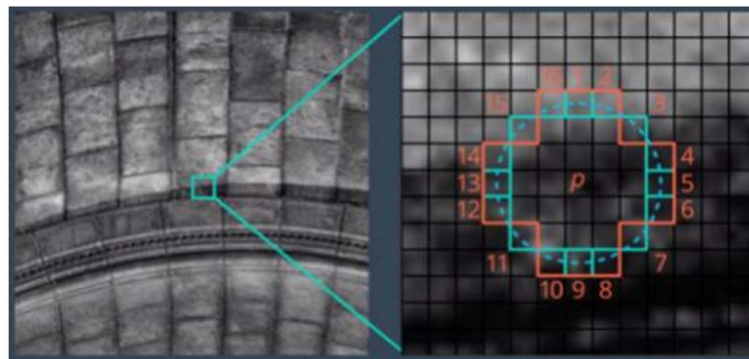


Figure 1. FAST corner detection

图 1. FAST 角点检测

针对普通 FAST 角点存在的旋转匹配问题, 改进的 Oriented FAST 角点通过计算灰度质心实现旋转不变性。首先在图像帧中某个图像块 B , 定义图像块的矩如式 1 所示。

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), p, q = 0, 1 \quad (1)$$

式 1 中 $I(x,y)$ 表示在像素坐标 (x,y) 处图像的灰度值, p, q 取 0 或者 1, m_{pq} 表示图像的矩。将图像的灰度值作称为质心, 根据式 1 计算得到质心 C 的位置:

$$c = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2)$$

连接矩阵块 B 的几何中心 O 和灰度质心 C 可得到方向向量 \overline{OC} , 因此 FAST 角点的方向角表示为:

$$\theta = \arctan(m_{01}/m_{10}) \quad (3)$$

经过改进的 Oriented FAST 在尺度和旋转变换下具有不变性, 这增强了特征点在各种场景中的表示能力, 从而提高了鲁棒性。

2.2. Rotated BRIEF 描述子

Rotated BRIEF 描述子由 256 位二进制字符组成, 具体计算步骤如下: 在以特征点原点为中心的像素区域 P 中, 随机选择一对像素点, 对它们的灰度值进行比较, 得到描述子 τ 的当前比特位的二进制值:

$$\tau(p; a, b) = \begin{cases} 1 & I(a) < I(b) \\ 0 & I(a) \geq I(b) \end{cases} \quad (4)$$

式 4 中 $I(a)$ 是点 a 处灰度值, $I(b)$ 是点 b 处灰度值。进行 256 对像素点的对比, 最终形成的 256 维二进制字符即为当前特征点的 BRIEF 描述子, 如式 5 所示:

$$f(P) = \sum_{1 \leq i \leq 256} 2^{i-1} \tau(p; a, b) \quad (5)$$

当前得到的 BRIEF 描述子没有方向性, 而通过质心方向 θ 可以实现 Rotated BRIEF 描述子。他们可以由 2×256 矩阵 S 表示:

$$S = \begin{pmatrix} a_1, a_2, \dots, a_{256} \\ b_1, b_2, \dots, b_{256} \end{pmatrix} \quad (6)$$

通过式 6 得到的质心方向 θ 可以计算对应的旋转矩阵 R_θ 如式 7:

$$R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (7)$$

由矩阵与方向, 可得旋转描述子 S_θ :

$$S_\theta = R_\theta * S \quad (8)$$

改进的 Rotated BRIEF 描述子利用 FAST 角点在提取阶段获取的角点方向信息, 使 ORB 特征点的描述子在图像旋转时具有良好的旋转不变性。

经过优化的 Rotated BRIEF 描述子结合了 FAST 角点在提取阶段获得的角点方向信息, 使得 ORB 描述子在图像发生旋转时保持出色的旋转不变性。利用 Rotated BRIEF 描述子为位姿估计提供精确的特征点匹配对, 这在视觉里程计中具有至关重要的作用。

2.3. 语义分割网络

在本文语义特征点算法中, 采用 YOLACT++ 网络[15]提取环境语义信息。YOLACT++ 网络是一种简单的全卷积实时语义分割网络, 使用 Resnet101-FPN 作为主干网络, 使用在 MSCOCO 数据集上训练得到的网络参数进行物体检测。假设 YOLACT++ 网络的输入是一个 $x \times y \times 3$ 的 RGB 图像矩阵, 则网络的输出就是一个 $x \times y \times n$ 的掩码矩阵, 其中 n 是图像中物体的数量。对于网络最后的每个输出通道 $m \in n$, 都会得到一个对象的掩码矩阵, 该网络实际分割效果图 2 所示。

根据图 2 可知, YOLACT++ 网络分割输出了物体掩码, 置信度与目标定位框, 其输出结果展示了网络精细的图像分割效果, 将其物体掩码提取出来可视化后如图 3 所示。



Figure 2. Segmentation results of the YOLACT++ network
图 2. YOLACT++网络分割效果图



Figure 3. Visualization of object masks
图 3. 物体掩码可视化图

根据图 3 中的物体掩码, 本文在提取 ORB 特征点时, 根据其位置从语义信息图中提取相应的语义信息, 为每个特征点赋予相应的语义信息, 从而实现了 ORB 特征点的语义化。

3. 改进语义 ORB 算法

3.1. 改进语义 ORB 特征点提取流程

本文提出的改进 ORB 语义特征点使用语义信息并结合改进四叉树分配算法, 具体提流程如图 4。

从图 4 可知, 在输入图像后, 处理过程分为两个分支。其中一个分支将图像输入神经网络, 以提取图像的语义信息矩阵; 另一个分支则对图像进行预处理, 构建图像金字塔, 为均匀提取角点做好准备。在图像预处理过程中, 首先将三通道的 RGB 图像转换为单通道的灰度图像, 接着利用二维高斯分布函数作为卷积核对图像进行降噪处理, 如式 9:

$$Q(x, y) = \frac{1}{2\pi\beta} \exp\left(-\left(\frac{x^2 + y^2}{2\beta^2}\right)\right) \quad (9)$$

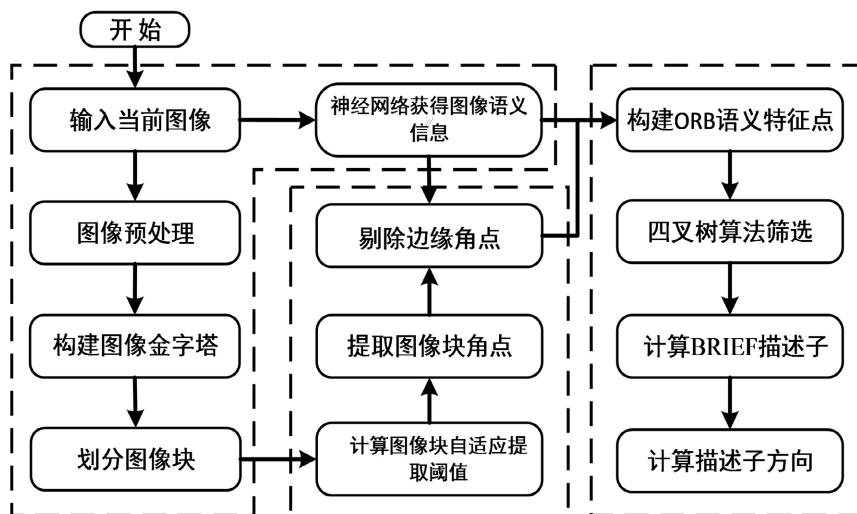


Figure 4. Flowchart for improved ORB semantic feature point extraction

图 4. 改进 ORB 语义特征点提取流程图

该卷积核的公式如式 9 所示, 其中参数 β 表示模糊半径, 其值与图像的滤波平滑度成正比。此外, 该流程中还建立了 8 层图像金字塔, 将缩放尺度设置为 1.2, 并使用双线性插值的方式来保证图像效果。通过使用图像金字塔, 相机图像在不同尺度下依然能够正确匹配。

3.2. 改进四叉树分配算法

改进 ORB 特征点算法分为提取与分配两个部分, 在提取阶段, 为了避免特征点提取过于集中的问题, 改进 ORB 特征点算法使用了基于图像块的角点提取方式, 将整个图像分割成多个图像块。为了平衡效率和性能, 算法选择 30×30 像素作为提取 FAST 角点的图像块尺寸, 并且对于图像末尾不足 30 像素的区域, 将其整体视为一个图像块进行提取。为了确保在低纹理区域能够有效地提取 FAST 角点, 本文提出了一种基于图像块的自适应阈值算法。该算法根据特定条件自动调整提取阈值 T , 具体而言, 该阈值的设定方式为:

$$T = \frac{1}{mn_i} \sum_{x=1}^{n_i} (I(x) - m)^2 + t_i \quad (10)$$

式 10 中 n_i 代表图像块中像素点的总数, m 表示图像块的像素平均值, $I(x)$ 代表位置 x 处像素的灰度值, 而 t_i 为初始阈值, 用于避免自适应阈值 T 过小而导致提取无效的特征点。根据经验, t_i 的值设置为 10。通过使用自适应阈值算法, 可以根据不同纹理程度的图像块采用相应的阈值提取角点, 从而使本文所改进的 ORB 算法能够适应不同纹理特征的环境, 进而在每个图像块中都能提取出合适数量的 FAST 角点。

使用本文的自适应阈值算法后, 每个图像区域都提取到了足够数量的 FAST 角点。因此, 在确保角点数量的情况下, 需要对提取的角点进行筛选, 以保证它们分布均匀且具有良好的质量。为此, 本文采用了四叉树分配算法对提取到的 FAST 角点进行筛选。该算法的大致流程如图 5 所示。

图 5 中红色为提取的特征点, 蓝色带叉为停止分裂区域, 其具体流程如下:

- 1) 将图像区域里的特征点输入算法中, 并原始图像初始化根节点。
- 2) 将当前节点的区域通过四叉树算法分别划分为四个新的节点, 将当前节点中特征点根据所处位置分配给对应的子节点。

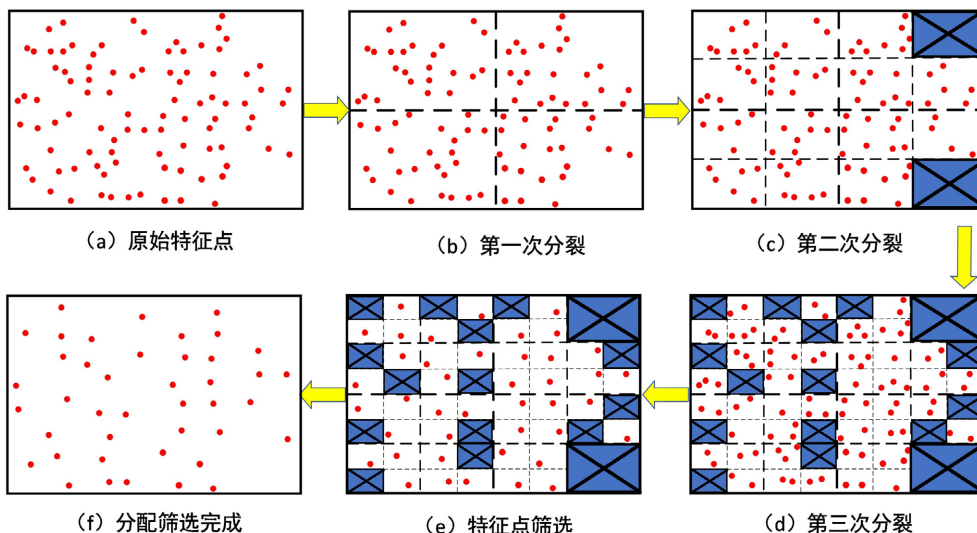


Figure 5. Flowchart of quad tree allocation algorithm
图 5. 四叉树分配算法流程示意图

3) 记录每个子节点区域内的特征点数量, 若数量为零, 则将该节点不再进行分裂, 若数量不为零则继续进行四叉树分裂。

4) 计算当前已经划分的节点数量, 若数量超过所需数量则停止分裂, 同时保存节点内质量最好的特征点, 若节点数量未超过所需数量, 则返回运行步骤(2), 对当前存在的有效节点继续进行四叉树分裂。

从图 5 可看出四叉树分配算法首先将图像划分为根节点, 然后通过递归分裂的方式生成子节点。在此过程中, 根据特征点的存在情况以及需求数量, 对分裂过程进行控制。最终, 在各节点区域内保留响应值最大的角点, 以实现特征点的有效提取。

3.3. 边缘角点剔除

在利用四叉树分配和自适应阈值提取得到 ORB 语义特征点之后, 需要结合神经网络得到语义掩码进行边缘点检测以消除边缘点。边缘点是指位于分割掩码边缘区域的角点, 如图 6 所示。

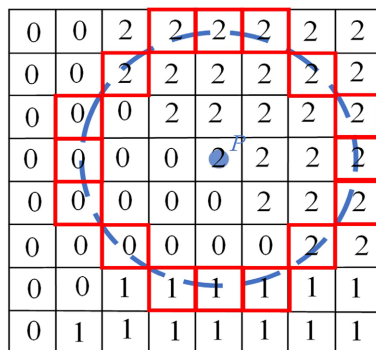


Figure 6. Illustration of edge and corner points
图 6. 边缘角点示意图

图 6 中角点 P 为边缘角点, 在本文中为了判断是否为边缘点, 需要使用图像的语义信息矩阵。FAST 角点提取涉及红圈中的灰度值判断, 在本文方法具体实现中, 以角点原点坐标为中心, 考察 7 × 7 矩形块

内的像素。若矩形块内存在与角点原点不同的掩码编号, 则判定该点为边缘点并剔除。若完全相同, 则保留非边缘点。通过剔除边缘点, 仅保留稳健的语义特征点有利于语义特征点的准确匹配。

3.4. 语义角度直方图匹配优化算法

针对特征点匹配中存在的错误匹配特征点对问题, 本文提出了一种语义角度直方图优化算法, 以剔除误匹配点对。该方法分为两个部分: 匹配点对角度差直方图筛选和语义类别筛选。利用 ORB 特征点所采用的 Oriented FAST 角点算法, 可以获取特征点的角度信息。其匹配特征点的角度信息如图 7 所示:

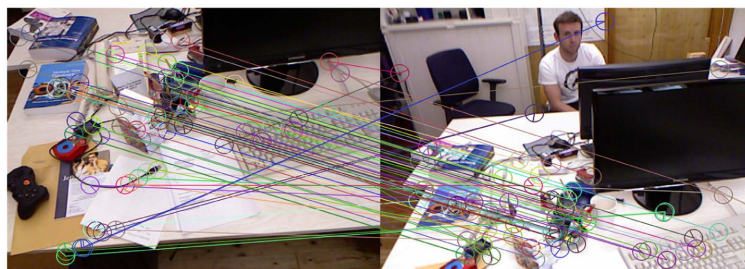


Figure 7. Illustration of matching feature point angles
图 7. 匹配特征点点角度示意图

在图 7 中, 两幅图像间的匹配点特征点由连线表示, 带有线圈的圆圈表示匹配特征点的方向角度。设第一帧匹配特征点的角度为 α , 第二帧匹配特征点的角度为 β , 则两帧之间匹配点的角度差可表示为:

$$\begin{cases} \theta = b - a & b - a > 0 \\ \theta = (b - a) + 360 & b - a < 0 \end{cases} \quad (11)$$

根据式 11, 当两帧之间的角度差大于 0 时, 角度差 θ 等于两帧角度之差; 而当角度差小于 0 时, 角度差 θ 等于两帧角度之差加上 360°。因此, 角度差 θ 的取值范围为 0°到 360°。将此范围划分为 60 个直方图区间, 将计算得到的角度差值分配至相应的角度直方图区间内, 统计每个区间内匹配特征点的数量, 并绘制类似于的角度直方图。如图 8 所示, 选取数量最多的区间作为正确匹配特征点, 剔除其余错误匹配。此外, 为了降低计算误差, 保留正确区间相邻区间内的匹配点, 这样的处理方法可以提高匹配特征点的准确性。

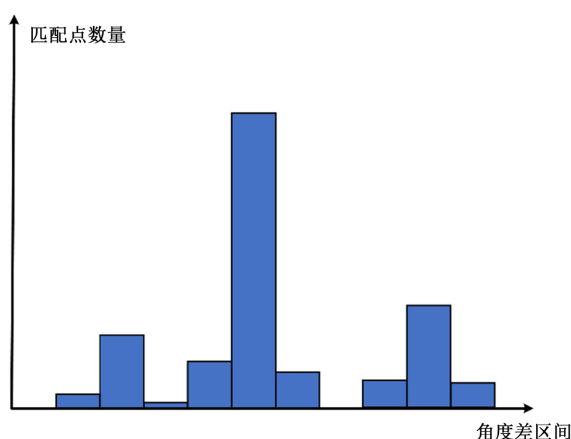


Figure 8. Illustration of angle histogram filtering
图 8. 角度直方图筛选示意图

语义类别筛选通过对比特征点对的语义类别信息, 对匹配特征点对进行优化。具体而言, 当两个匹配点的语义类别相同时, 保留该匹配对; 若不同, 则予以剔除。本文采用的筛选方法效果如图 9 所示。

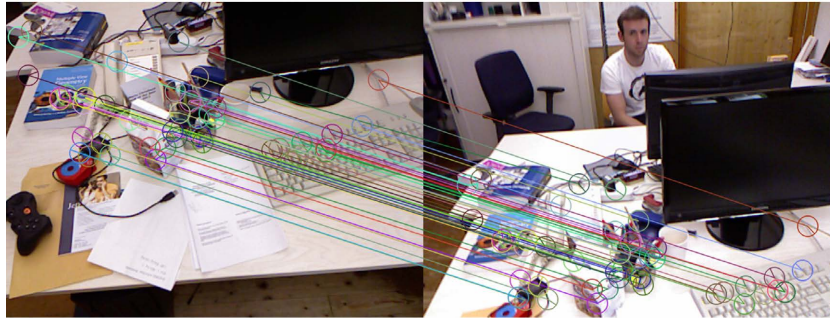


Figure 9. Illustration of feature point matching filtering effect
图 9. 匹配特征点筛选效果

对比图 9 与图 7 可以看出论文提出的语义角度直方图优化算法有效地筛选并剔除了误匹配, 保留了正确的匹配点对。提高了论文提出的改进语义特征点的匹配精度。

4. 实验与结果分析

为了验证论文提出的改进语义特征点算法在特征点算法中的性能表现, 本章将论文提出的改进语义特征点算法与普通 ORB 算法, 四叉树分配算法特征点在特征点提取与特征点匹配两个方面进行实验评估, 并在匹配环节使用多种环境场景多种匹配阈值进行评估分析。

4.1. 特征点提取实验

为了验证本本中提出的特征点提取方法的有效性, 本节将采用改进的语义特征点提取策略, 并将其与标准 ORB 特征点提取算法以及采用四叉树分配策略的 ORB 特征点提取方法进行对比分析。在同一张图片中分别提取了 500 个特征点以比较各方法的特征点分布。普通 ORB 算法采用 OpenCV 图像处理库中的实现版本, 其提取效果如图 10 所示。



Figure 10. Extraction effect of regular ORB algorithm
图 10. 普通 ORB 算法提取效果

在图 10 中, 左侧图呈现了提取的特征点在 RGB 图像上的分布, 而右侧图则展示了将这些特征点投影至由神经网络分割生成的掩码上, 以更直观地反映特征点的分布情况。由图可知, 标准 ORB 算法主要关注具有最大响应值的 FAST 角点, 从而在图像中纹理较为丰富的区域聚集了大量特征点。然而, 这种

特征点分布过于密集, 导致物体掩码上特征点数量不足, 进而不利于对图像内物体的语义信息进行有效感知。

在四叉树分配算法中, 为了提取 500 个特征点通常需要首先提取大量的特征点, 然后通过筛选来减少其数量。本文提出了一种自适应阈值算法, 与传统四叉树分配算法相比, 后者采用固定的 25 提取阈值。两种算法提取到的特征点总数如表 1 所示。

Table 1. Total number of extracted feature points

表 1. 提取特征点总数量

算法	特征点数量
自适应阈值算法	1468
固定 25 阈值	1526

根据表 1, 可以发现固定阈值为 25 的四叉树算法与论文算法的特征点提取数量相近。图 11 展示了两种算法得到的特征点。

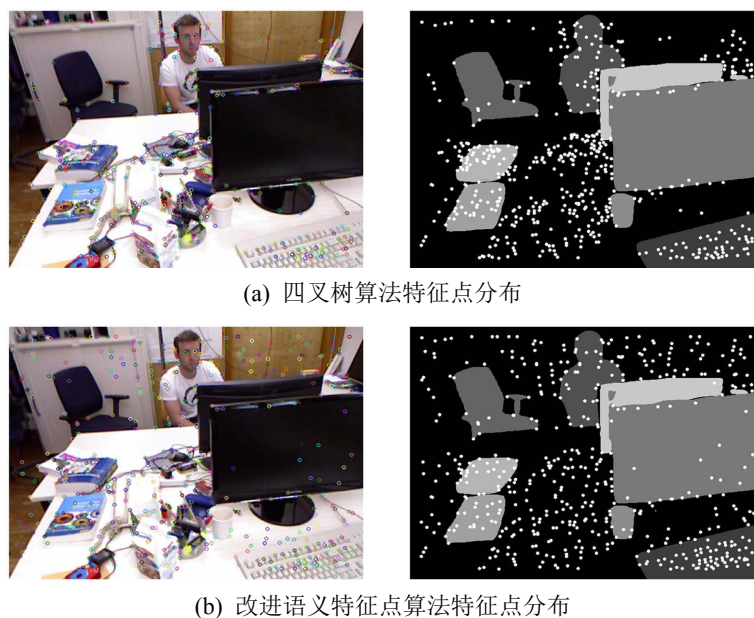


Figure 11. Distribution effect of feature point extraction

图 11. 特征点提取分布效果

在图 11 中, 图 11(a)展示了四叉树分配算法特征点分布, 而图 11(b)展示了本文语义特征点分布。从右侧的掩码特征点分布图中可以看出, 在提取接近数量的待分配特征点的前提下, 本文的改进语义特征点所采用的自适应阈值算法相较于传统四叉树算法的固定阈值提取, 能够实现更加均匀的特征点分布, 满足了语义 SLAM 系统对提取特征点的要求。此外, 改进语义特征点利用图像的语义信息剔除了位于掩码边缘的特征点, 这有利于减少在进行语义特征点匹配时可能出现的误匹配情况。

4.2. 特征点匹配实验

本章采用 TUM 数据集中的室内图像数据集进行实验, 确保方法有效性和实验可靠性。实验从数据

集中选取三种室内场景的图像序列, 分别为 fr1_rpy, fr2_desk 和 fr3_sitting_xyz, 三种室内场景如图 12 所示。

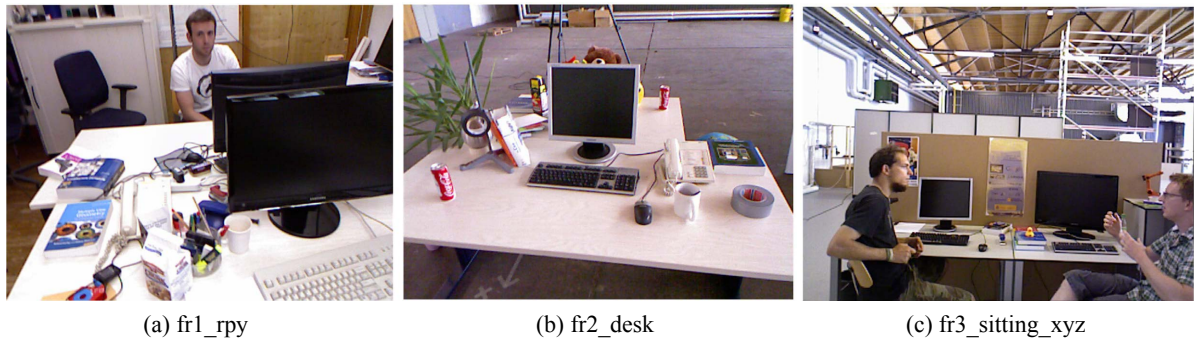


Figure 12. Illustration of experimental scene image
图 12. 实验场景图像示意图

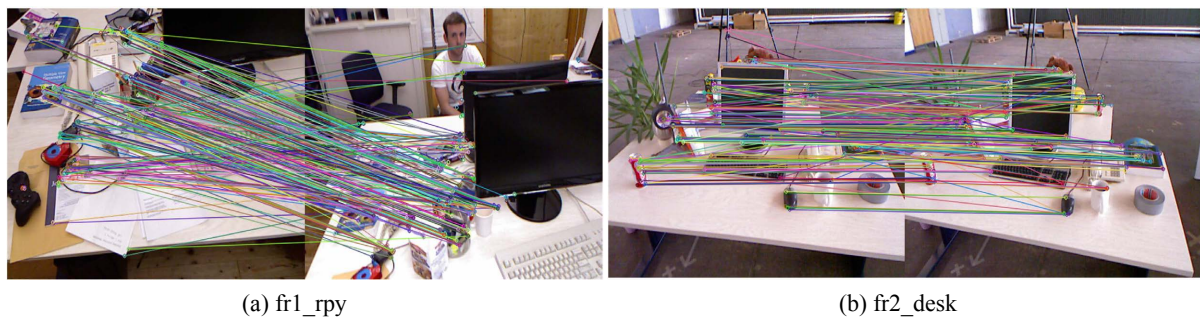
图 12 为选定场景序列图像, 在选定的匹配图像中, 包含多种室内场景和相机运动, 如相机上下旋转(fr1_rpy)、左右旋转(fr2_desk)和平移(fr3_sitting_xyz)。选定这些图像进行对比实验可以确保本文方法的可信性。

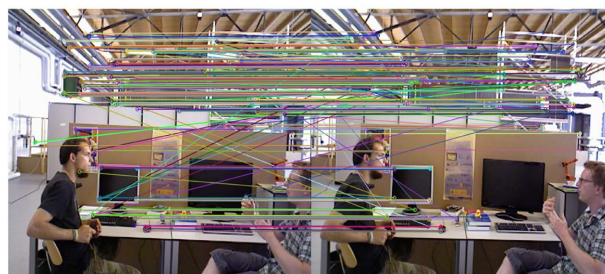
本章实验采用三组不同环境匹配图像提取同样 500 个特征点, 采用 50, 65, 80 三种提取阈值, 匹配方法使用汉明距离暴力匹配, 比较普通 ORB 算法, 四叉树分配算法与本文语义特征点算法的特征点匹配精准度, 三种算法的匹配特征点数量如表 2 所示。

Table 2. Number of matching feature points in three different scenarios
表 2. 三种场景下匹配特征点数量

匹配方法	fr1_rpy/个数			fr2_desk/个数			fr3_sitting_xyz/个数		
	50	65	80	50	65	80	50	65	80
普通 ORB 方法	172	363	480	211	338	462	217	327	461
四叉树方法	72	204	370	133	256	429	117	204	393
本文方法	51	87	110	114	154	179	107	127	163

表 2 中为不同匹配方法使用三种阈值下的匹配点数量, 可以看出未进行均匀化的普通 ORB 方法在三对匹配图像的三种阈值下的匹配点数量上都有巨大的优势。图 12 为普通 ORB 方法在 65 匹配阈值下匹配点图, 在图中两张图像为匹配图像对, 相连线段为匹配图像的匹配特征点对。

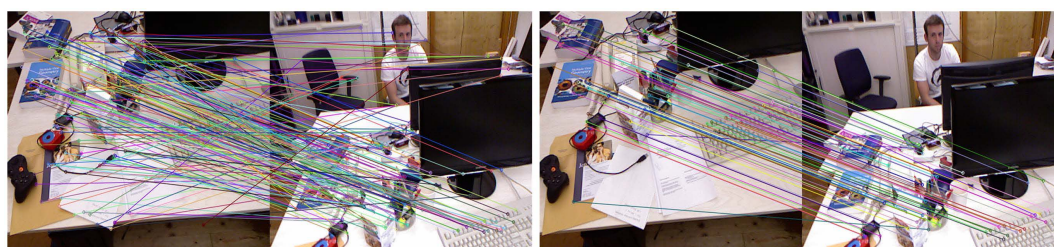




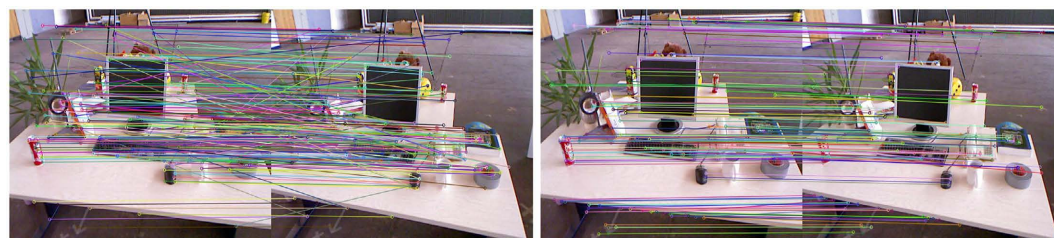
(c) fr3_sitting_xyz

Figure 13. Illustration of matching points using regular ORB method**图 13.** 普通 ORB 方法匹配点示意图

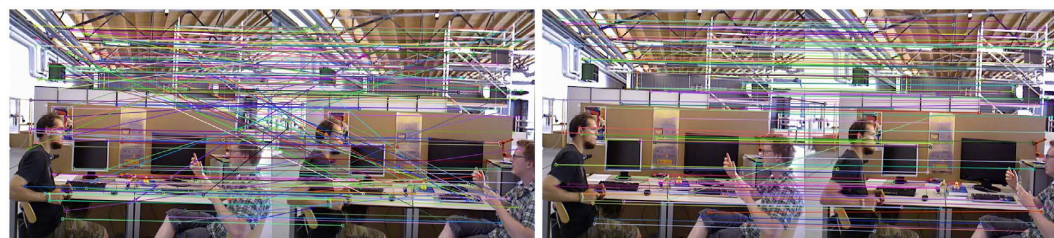
从图 13 中可以观察到, 传统的 ORB 方法在提取特征点时, 主要集中于纹理丰富的区域。这一特性使得它能够匹配到更多的特征点, 然而, 这也导致匹配点过于密集且存在误匹配的情况, 并不满足语义 SLAM 对特征点算法的需求。因此需要采用四叉树分配算法来实现特征点的均匀分布。而本文所提出的语义特征点方法与四叉树分配方法在阈值为 65 时的匹配效果如图 14 所示。



(a) fr1_rpy



(b) fr2_desk



(c) fr3_sitting_xyz

Figure 14. Comparison of matching feature points between two algorithms**图 14.** 两种算法匹配特征点对比示意图

在图 14 中, 左侧展示了基于四叉树分配方法的匹配结果, 而右侧则展示了论文提出的基于语义特征点方法的匹配结果。通过与图 13 的对比可以发现, 尽管使用四叉树分配方法得到的匹配点数量减少, 但

其分布更加均匀, 更符合视觉 SLAM 的需求。将本文提出方法与四叉树分配方法进行比较, 从图 14 中可以观察到, 在 65 匹配阈值下, 本文提出的优化方法能够剔除大部分误匹配, 保留正确的匹配特征点对。然而, 在相同的匹配阈值下, 四叉树分配算法匹配特征点仍存在大量误匹配。因此, 通过图 12 与图 14 的对比, 可以看出论文提出的改进语义特征点方法在 65 阈值下能够有效地去除大部分误匹配。而在相同条件下, 两种对比方法均出现了大量误匹配, 已无法求解正确的变换位姿, 这进一步证明了对匹配点对进行筛选优化的重要性。

论文提出的改进语义特征点方法通过语义角度直方图优化匹配结果, 结合特征点的语义类别和角度差实现精确匹配。本文算法在 TUM 数据集的三种室内环境, 50 与 80 匹配阈值下的特征点匹配效果如图 15 所示。

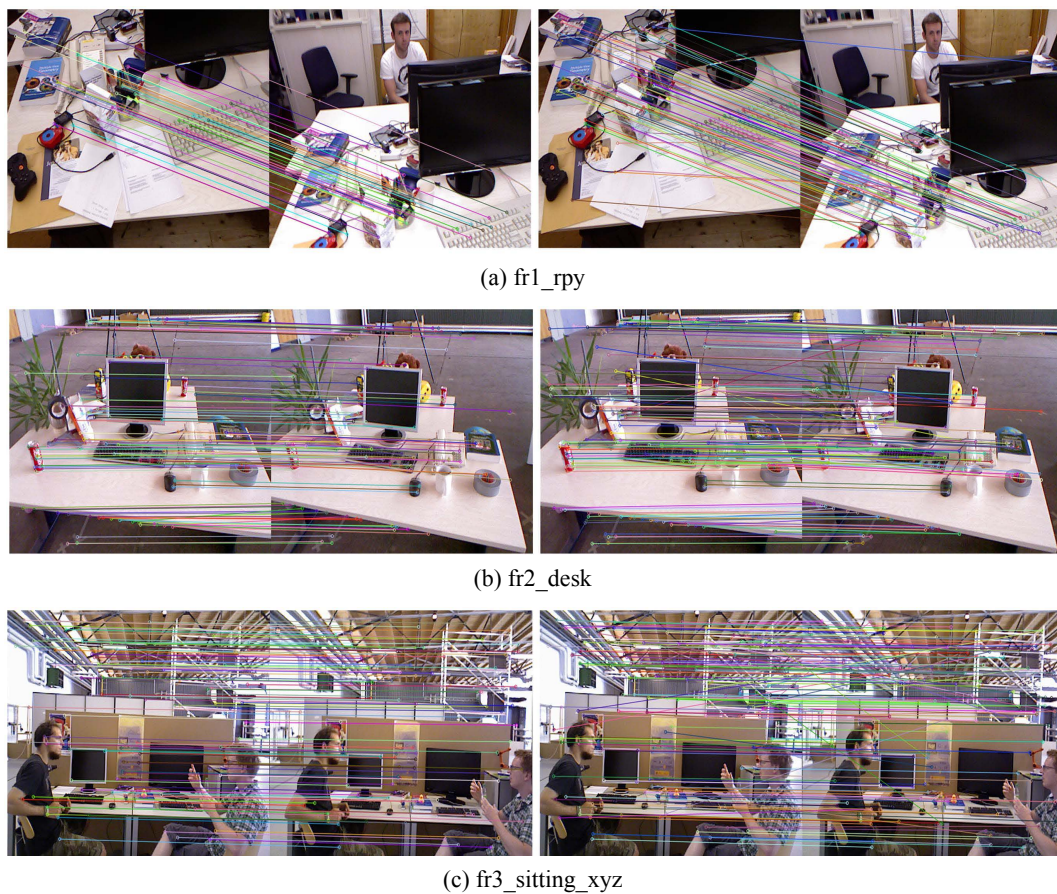


Figure 15. Illustration of image feature point matching under two thresholds

图 15. 两种阈值下图像特征点匹配示意图

在图 15 中展示了两个匹配图像对, 它们之间的连接线段表示匹配特征点。左图使用描述子匹配阈值为 50, 而右图为 80。在阈值为 50 的情况下, 结合本文提出的语义特征点匹配优化方法, 可以确保得到的匹配特征点都是正确的, 没有不符合相机运动方向的错误匹配。然而, 在阈值为 80 的宽松匹配条件下, 理论上会有大量误匹配, 但通过本文提出的语义直方图筛选算法, 仍能筛选掉了大部分错误匹配, 并保留了正确的匹配特征点对。因此, 实验结果表明本文使用的语义角度直方图优化算法能够稳定有效地筛选掉特征点匹配环节中出现的误匹配, 大大提高了本文语义特征点的匹配精度。

5. 结束语

本文提出了一种改进的语义特征点算法,旨在满足语义 SLAM 系统对特征点的需求。该算法使用 YOLACT++网络对图像进行分割,提取匹配图像内的语义信息,并在特征点提取环节结合语义信息与改进四叉树分配算法实现特征点的语义化与均匀化。同时,在特征点匹配环节,本文提出了基于本文语义特征点算法的语义角度直方图匹配优化算法,该算法通过语义筛选与角度直方图筛选两个环节来剔除误匹配,从而获得更精准的特征点匹配对。实验对比评估表明,与对比方法相比,本文提出的语义特征点算法在提取上实现了特征点的语义化与均匀化,并满足了语义 SLAM 系统对特征点的需求。此外,本文提出的匹配优化算法在多种场景多种匹配阈值下匹配精度均高于对比算法,实现了本文语义特征点的精准匹配。因此,本论文算法具有重要的理论和实际应用价值。

参考文献

- [1] 高翔,张涛. 视觉 SLAM 十四讲: 从理论到实践[M]. 北京: 电子工业出版社, 2017.
- [2] Wu, Y. (2019) Research on Feature Point Extraction and Matching Machine Learning Method Based on Light Field Imaging. *Neural Computing and Applications*, **31**, 8157-8169. <https://doi.org/10.1007/s00521-018-3962-7>
- [3] Ma, J., Jiang, X., Fan, A., et al. (2021) Image Matching from Handcrafted to Deep Features: A Survey. *International Journal of Computer Vision*, **129**, 23-79. <https://doi.org/10.1007/s11263-020-01359-2>
- [4] Lowe, D.G. (2004) Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**, 91-110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [5] Bay, H., Tuytelaars, T. and Gool, L.V. (2006) Surf: Speeded Up Robust Features. In: Leonardis, A., Bischof, H. and Pinz, A., Eds., *ECCV 2006: Computer Vision-ECCV 2006, Lecture Notes in Computer Science*, Vol. 3951, Springer, Berlin, 404-417. https://doi.org/10.1007/11744023_32
- [6] Rublee, E., Rabaud, V., Konolige, K., et al. (2011) ORB: An Efficient Alternative to SIFT or SURF. *Proceedings of 2011 International Conference on Computer Vision*, Barcelona, 6-13 November 2011, 2564-2571. <https://doi.org/10.1109/ICCV.2011.6126544>
- [7] Viswanathan, D.G. (2009) Features from Accelerated Segment Test (Fast). *Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services*, London, 6-8 May 2009, 6-8.
- [8] Calonder, M., Lepetit, V., Strecha, C., et al. (2010) Brief: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P. and Paragios, N., Eds., *ECCV 2010: Computer Vision-ECCV 2010, Lecture Notes in Computer Science*, Vol. 6314, Springer, Berlin, 778-792. https://doi.org/10.1007/978-3-642-15561-1_56
- [9] Tareen, S.A.K. and Saleem, Z. (2018) A Comparative Analysis of Sift, Surf, Kaze, Akaze, Orb, and Brisk. *Proceedings of 2018 International Conference on Computing, Mathematics and Engineering Technologies*, Sukkur, 3-4 March 2018, 1-10. <https://doi.org/10.1109/ICOMET.2018.8346440>
- [10] Mur-Artal, R. and Tardós, J.D. (2017) Orb-SLAM2: An Open-Source Slam System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, **33**, 1255-1262. <https://doi.org/10.1109/TRO.2017.2705103>
- [11] 李伦. 基于 PROSAC 算法的室内定位算法研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2018.
- [12] 李国竣, 徐延海, 段杰文, 等. 利用局部自适应阈值方法提取 ORB-SLAM 特征点[J]. 测绘通报, 2021(9): 32-36.
- [13] 潘峰, 沈建, 新秦顺, 等. 融合 GMS 的 ORB 特征点提取与匹配算法[J]. 计算机工程与设计, 2022, 43(8): 2244-2251.
- [14] Rosten, E. and Drummond, T. (2006) Machine Learning for High-Speed Corner Detection. In: Leonardis, A., Bischof, H. and Pinz, A., Eds., *ECCV 2006: Computer Vision-ECCV 2006, Lecture Notes in Computer Science*, Vol. 3951, Springer, Berlin, 430-443. https://doi.org/10.1007/11744023_34
- [15] Bolya, D., Zhou, C., Xiao, F., et al. (2022) Yolact++ Better Real-Time Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 1108-1121. <https://doi.org/10.1109/TPAMI.2020.3014297>