

一种基于区域密度划分 改进的ORB特征提取 算法

王 鹏

盐城工学院电气工程学院, 江苏 盐城

收稿日期: 2023年6月9日; 录用日期: 2023年7月23日; 发布日期: 2023年7月31日

摘 要

原ORB算法在面对特征点分布不均匀的图像做特征提取时, 常会出现特征点提取效率低, 特征点冗余的问题, 导致特征点在匹配时出现误匹配, 甚至位姿丢失的情况。针对这一问题, 提出了一种基于区域密度划分改进的ORB特征提取算法。首先对图像构造图像金字塔, 确保图像的尺度不变性。其次用FAST特征点提取算法对图像进行特征点提取, 提取完毕后, 使用核密度估计方法, 以每个特征点为中心, 计算该点周围的密度值, 并采用聚类算法将密度分布相似的特征点看成一簇, 根据每一簇的特征点密度和图像平均密度的比值, 将图像区域划分为三类, 即特征点密集区域, 特征点稀疏区域和特征点均匀区域。然后利用自适应二叉树法对三个区域图像进行分割, 并根据每个区域的特征点密度计算阈值, 达到特征点筛选均匀的目的。然后利用自适应非极大值抑制法对特征点进行最佳筛选, 并使用BRIEF算法计算出特征点的描述子。最后进行特征点匹配。实验结果表明, 本文算法相较于传统的ORB算法有效地减少了冗余特征点的数量, 降低了特征点的重叠率, 在特征匹配的精度和效率上有了明显的提升。

关键词

ORB算法, 特征点提取, 特征点密度, 二叉树, 自适应非极大值抑制

An Improved ORB Feature Extraction Algorithm Based on Region Density Division

Peng Wang

School of Electrical Engineering, Yancheng Institute of Technology, Yancheng Jiangsu

Received: Jun. 9th, 2023; accepted: Jul. 23rd, 2023; published: Jul. 31st, 2023

文章引用: 王鹏. 一种基于区域密度划分改进的 ORB 特征提取算法[J]. 建模与仿真, 2023, 12(4): 4233-4245.
DOI: 10.12677/mos.2023.124386

Abstract

When the original ORB algorithm is used for feature extraction in the face of images with uneven feature point distribution, the problems of low feature point extraction efficiency and redundancy of feature points often occur, resulting in mismatching of feature points and even loss of pose. To solve this problem, an improved ORB feature extraction algorithm based on region density division was proposed. Firstly, the image pyramid is constructed to ensure the scale invariance of the image. Secondly, the FAST feature point extraction algorithm is used to extract the feature points of the image. After the extraction, the kernel density estimation method is used to calculate the density values around the point with each feature point as the center, and the clustering algorithm is used to treat the feature points with similar density distribution as a cluster. According to the ratio of the feature point density of each cluster to the average image density, the image area is divided into three categories. That is, feature point dense region, feature point sparse region and feature point uniform region. Then the adaptive quadtree method is used to segment the image of three regions, and the threshold is calculated according to the density of feature points in each region, so as to achieve the purpose of uniform feature point screening. Then, adaptive non-maximum suppression method was used to optimize the feature points, and BRIEF algorithm was used to calculate the descriptor of the feature points. Finally, feature point matching is carried out. Experimental results show that compared with the traditional ORB algorithm, the proposed algorithm can effectively reduce the number of redundant feature points, reduce the overlap rate of feature points, and significantly improve the accuracy and efficiency of feature matching.

Keywords

ORB Algorithm, Feature Point Extraction, Feature Point Density, Adaptive Quadtree, Adaptive Non-Maximum Suppression

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

近年来随着人工智能和相关技术的发展，智能化，科技化已经逐步走进了现代社会。移动机器人正在为人类的生活提供便利。因此移动机器人的控制和路径规划在研究领域就显得尤为重要，而实现移动机器人的路径规划，需要定位与建图技术作为支撑，因此基于机器视觉的定位和同步建图技术(V-SLAM)已经成为机器人领域研究的热点话题。在视觉 SLAM [1]领域，其中最核心的就是特征点的提取与匹配，特征点的提取通常使用基于梯度或角点的方法，例如 Harris 角点检测器[2]、SIFT [3]、SURF [4]、FAST [5]、ORB 等，其中 Harris 角点检测器对于拐角处、角落处的特征点具有很好的响应，且响应值大小可以用于衡量角点的强度。但是其不适用于纹理比较丰富的区域，对于尺度变化、旋转变化比较敏感；SIFT 可以在不同尺度下检测出特征点，具有良好的尺度和旋转不变性，对于光照变化、噪声等干扰也有一定的鲁棒性，但是由于其特征点描述子的维度较高，需要消耗大量的存储空间，计算量较大，不适用于实时应用；SURF 算法利用积分图、近似 Hessian 矩阵和 Haar 小波转换运算对 SIFT 算法进行了改进，但仍存在着特征描述子计算量大的问题；FAST 算法计算速度快，能够快速地检测出图像中的角点，但是由于其仅仅是通过比较目标像素周围的邻近像素来判断角点，忽略了像素的分布情况，因此在特征点分布杂乱

的复杂场景和光照变化明显的场景下检测效果较差,存在明显的误检和漏检问题;2011年,Rublee等提出了ORB算法,ORB算法在FAST的基础上,增加了BRIEF特征点描述子[6],该特征描述子表示为一个二进制串,相比于SIFT和SURF等算法的高维描述子,BRIEF描述子具有更低的维度,提高了算法的特征点提取速度。但是也正是由于其低纬度的描述子,其特征描述没有SIFT和SURF等算法的描述子描述精准,因此其精度和鲁棒性不如SURF,并且其没有尺度不变性,容易产生特征点冗余,这些冗余的特征点在特征匹配时会导致误匹配。

针对现有算法存在的问题[7],许多学者对其进行了改进:文献[8]对原ORB算法进行改进,提出了一种自适应灰度变化的特征提取算法,该算法在对比目标像素与相邻像素的灰度差异时,能够根据图像平均灰度自适应设定阈值。尽管该算法在光照不足时能够提取到更多的特征点,但也导致特征点冗余,尤其是光照充足时,特征点数量较多,处理起来相对耗时。文献[9]结合ORB和SIFT的优点,提出一种新的SIRB算法,该算法将SIFT中保证图像尺度不变的方法加以运用,并用ORB中的BRIEF描述子描述特征点,实验表明,该算法比SIEF更快速,比ORB更精确,但是该算法相比于ORB在实时性上还是差出不少。文献[10]提出一种基于区域划分的改进ORB算法,该方法提取的特征点更加均匀合理,有效避免了特征点扎堆的问题,但是由于其采用的传统的二叉树法,其提取深度和阈值是固定的,会降低整个特征提取过程的效率。针对以上不足,本文提出了一种全新的基于特征密度划分区域,并改进传统二叉树法的改进ORB算法,不仅使得特征点分布更加均匀,并且在保证系统性能的前提下,降低了特征点的数量,优化系统效率。

2. 传统的ORB算法

2.1. 构建图像金字塔

相机在提取特征点的过程中,由于相机的前后运动会导致图像或大或小,不具备尺度不变性,因而通过对同一幅图像进行一个固定倍率的缩放,就能得到这幅图像不同分辨率的一系列图片。这种方法称为构造图像金字塔,这样可以有效解决图像由于尺度变化而造成特征点无法匹配的情形,从而实现尺度不变性。

金字塔的底层是原始图像,金字塔层数越高,图像的分辨率越高,相应的图像的宽度和长度越小,图像面积也越小,所能提取到的特征点数量也越少,基于这个原理,可以按照图像的面积将特征点平均分摊到每层金字塔的图像上,假设原始图像的长为 M ,宽为 N , A 为总特征点数量,缩放因子为 s ($0 < s < 1$),那么金字塔的总面积为

$$S = M \times N \times \sum_{i=0}^{n-1} (s^2)^i = M \times N \times \frac{1 - (s^2)^n}{1 - s^2} \quad (1)$$

单位面积的特征点数量为

$$A_{\text{avg}} = \frac{A}{S} = \frac{A(1 - s^2)}{M \times N \times [1 - (s^2)^n]} \quad (2)$$

那么第 α 层的特征点数量为

$$A_{\alpha} = \frac{A(1 - s^2)}{1 - (s^2)^n} (s^2)^{\alpha} \quad (3)$$

本文对输入的图像构造8层图像金字塔,构造的图像金字塔如图1所示。

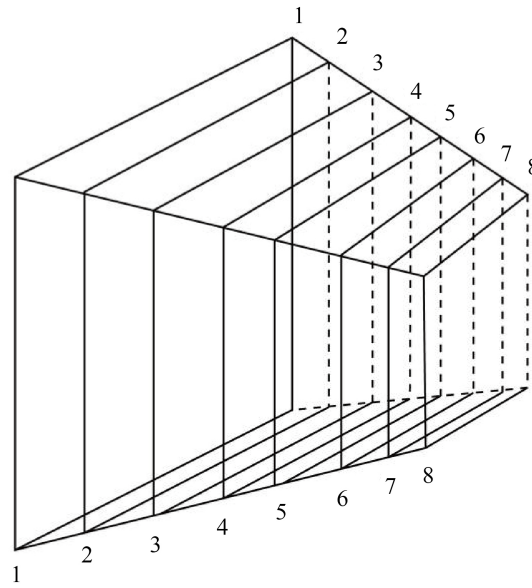


Figure 1. Image pyramid
图 1. 图像金字塔

2.2. FAST 特征点提取

FAST特征点提取算法[11]因为其能快速并准确的提取角点而被广泛运用。如图2所示，FAST特征点提取是以目标像素点P为圆心，通常以3个像素距离作为半径，做一个检测圆，根据圆上16个像素点与目标像素的灰度差来判断点P是否为角点，当圆上连续12个像素点与P点的灰度误差的绝对值大于阈值时，则为FAST角点，其判定步骤如下。

- 1) 计算检测圆上P1、P5、P9、P13与目标像素P的灰度差，若他们的绝对值大于阈值的个数不小于3个，则计入特征候选点，否则不是特征点。
- 2) 计算检测圆上其他像素点与目标像素P之间的灰度差绝对值，若其中有12个连续像素的灰度差绝对值大于阈值，则像素点P为特征点，否则不是特征点。
- 3) 遍历图像中所有像素点，重复上述步骤。

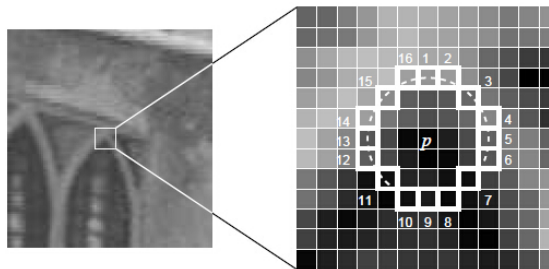


Figure 2. FAST feature point extraction
图2. FAST特征点提取

2.3. 四叉树法处理特征点

同一幅图像不同区域之间纹理有差异，有些区域图像纹理丰富，能提取到较多特征点，导致特征点

扎堆，造成特征点冗余，因此利用四叉树法[6]来处理特征点，使特征点分布相对均匀。

首先确定初始节点数目，根据图像的长宽比取整来决定，接着根据初始节点数量来分割图像，如图3所示，图中初始节点数为4，将图像分割成4个区域计为4个节点，特征点也被分配到相应的节点中，接着在包含多个特征点的节点中再次划分区域，计为子节点，并为了避免计算复杂，在每次划分子节点后都需要删除父节点，其树状结构如图4所示，若节点中不包含特征点，则删除该节点。直到节点中特征点数量不超过1个，则认为特征点均匀化成功。

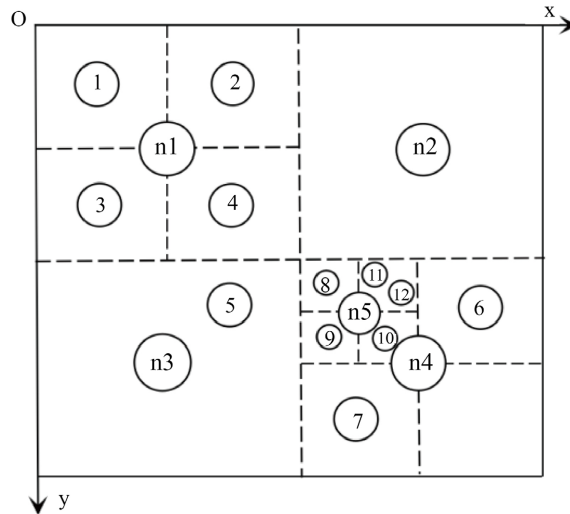


Figure 3. Image region segmentation
图3. 图像区域分割

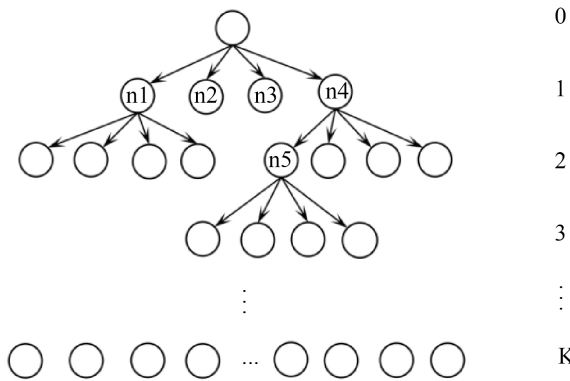


Figure 4. Quadtree tree structure
图4. 四叉树树状结构

2.4. BRIEF 特征点描述子

为了更好的描述特征点，ORB 算法引入 BRIEF 描述子[12]，该描述子通过比较特征点邻域内两个随机像素(比如 A 和 B)的大小关系：如果 A 比 B 大，则取 1，如果 A 比 B 小，则取 0。在特征点的邻域内取了 128 个这样的点对，最终得到 128 维度由 0, 1 组成的向量。那么描述子的第 i 个值 $T(P(A,B))$ 表示为：

$$T(P(A,B)) = \begin{cases} 1, & A > B \\ 0, & A \leq B \end{cases} \quad (4)$$

如图 5 所示为用 ORB 算法进行特征提取和匹配的效果图，从图中可以看出 ORB 算法提取的特征点过于密集，并且分布不均匀，在进行匹配时会出现特征点冗余和大量误匹配的现象。

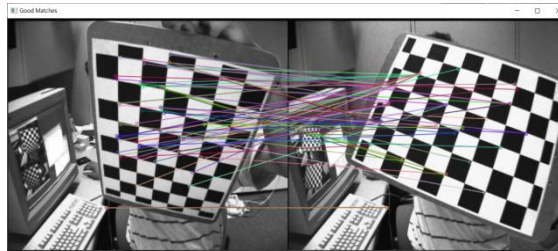


Figure 5. ORB algorithm rendering
图5. ORB算法效果图

3. 改进的 ORB 算法

当面对特征点分布不均匀的图像，原 ORB 算法在对特征点进行二叉树处理时，有以下两个缺点：

1) 二叉树算法的计算复杂度随着特征点数量的增加而增加。当特征点数量很大时，二叉树算法的计算时间会变得很长，影响 ORB 算法的运行效率。

2) 二叉树算法的分割方式是固定的，无法适应不同的数据集特征点分布。例如，在某些数据集中，特征点分布可能不是很均匀，而二叉树算法无法自适应地分割特征点，这可能会导致一些区域的特征点过于密集，而另一些区域的特征点过于稀疏。基于以上问题，本文提出了一种基于密度划分的改进 ORB 算法，在特征提取后，利用核密度估计[13]的方法计算图像特征点的平均密度，根据每个特征点周围的密度分布函数将图像划分为特征密集区域，特征稀疏区域以及无特征区域，并采用局部自适应深度二叉树法和自适应非极大抑制法对特征密集区域和特征稀疏区域的特征点进行筛选，以达到特征点分布均匀和提高特征点提取速度的目的。该算法的流程图如图 6 所示。

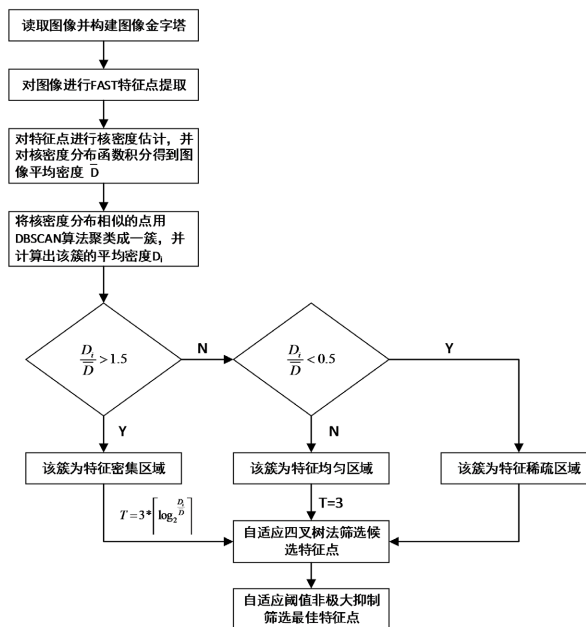


Figure 6. Algorithm flow chart
图 6. 算法流程图

3.1. 核密度估计划分区域

在对图像进行特征提取后，以每个特征点为中心，使用高斯核函数作为核函数，计算每个特征点周围的密度，然后对所有特征点的密度进行加权平均，得到整个区域的密度分布。

假设现有一组特征点 $\{p_1, p_2, \dots, p_n\}$ ，其中 $p_i = (x_i, y_i)$ ，表示第 i 个特征点的坐标，以点 p_i 为中心，高斯核函数 $K_{h(x)}$ 的形式为：

$$K_{h(x)} = \frac{1}{2\pi h^2} e^{-\left(\frac{x^2+y^2}{2h^2}\right)} \quad (5)$$

其中， h 是带宽参数，控制核函数的宽度和平滑程度。 x 和 y 分别表示点 p_i 到其他点 p_j 的水平和垂直距离。带宽参数可以提前在数据集上通过交叉验证等方法来确定。

在以点为中心的情况下，点 p_j 对点 p_i 的密度贡献为：

$$K_h(p_i - p_j) = K_h(x_i - x_j, y_i - y_j) \quad (6)$$

根据核密度估计的公式，以点 p_i 中心的密度估计值 $\hat{f}(p_i)$ 为：

$$\hat{f}(p_i) = \frac{1}{n} \sum_{i=1}^n K_h(p_i - p_j) \quad (7)$$

通过对所有特征点的密度进行加权平均，可以得到整个区域的密度分布，即核密度函数[14]。核密度函数 $f(x, y)$ 的计算公式为：

$$f(x, y) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i, y - y_i) \quad (8)$$

其中， n 是特征点的总数。 x 和 y 分别表示区域中每个像素点的水平和垂直坐标。

通过对图像的核密度分布函数进行积分，可以得到整个图像的特征点平均密度 \bar{D} ：

$$\bar{D} = \frac{1}{S} \iint f(x, y) dx dy \quad (9)$$

得到每个特征点的密度分布后，根据其核密度分布函数，将核密度分布相似的特征点用 DBSCAN 算法聚类成一簇，并将该簇中特征点核密度分布函数的平均值作为该簇的密度值 D_i 。

若 $D_i/\bar{D} > 1.5$ ，则该簇为特征密集区域。

若 $0.5 < D_i/\bar{D} < 1.5$ ，则该簇为特征均匀区域。

若 $D_i/\bar{D} < 0.5$ ，则该簇为特征稀疏区域。

3.2. 局部自适应四叉树

在对图像特征点划分区域之后，对于特征点密度不同的区域，采用局部自适应四叉树法来筛选特征点，该方法基于四叉树法将图像划分成四个节点，并对划分出来的节点统计特征点数量，并将子节点中的特征点数量与自适应阈值 T 比较，若该子节点中的特征点数量大于阈值 T ，则继续分割，将该节点再分割成四个子区域，若该节点的特征点数量小于等于阈值 T ，则停止分割。经过多次试验验证，当图像特征点分布较为均匀或稀疏时，阈值为 3，实验效果最佳，而当特征点分布较为密集时，过小的阈值则会需要过多的分割深度，导致耗时较为严重，因此针对特征点密集的区域，需要根据该其余特征点密度确定自适应阈值。其自适应阈值 T 的计算公式为：

$$T = \left\lceil 3 \times \log_2 \frac{D_i}{\bar{D}} \right\rceil \quad (10)$$

其中 $\frac{D_i}{D}$ 为区域特征点密度与图像平均密度的比值， $\lceil \cdot \rceil$ 为向上取整符号。

3.3. 自适应非极大抑制

经过以上步骤，特征点已被部分筛选，但特征点密集区域由于特征点数量过多，还是会导致节点内的筛选的特征点距离较近，不能成为最佳特征点。因此本文采用自适应阈值非极大抑制的方法来筛选最佳特征点。将区域内筛选出的特征点按照特征点密度大小进行排序,得到密度从大到小的特征点序列：

$$\{(x_1, y_1, s_1), (x_2, y_2, s_2), \dots, (x_i, y_i, s_i)\} \quad (11)$$

其中 (x_i, y_i) 是特征点的坐标， s_i 是特征点的密度。

选取密度最大的特征点 (x_i, y_i, s_i) ，将其加入到最佳特征点的集合中，对于剩余的特征点，计算其与选取特征点之间的距离：

$$d_i = \min_{(x_i, y_i) \in \text{points}} \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2} \quad (12)$$

对于特征点距离 d_i 小于阈值的特征点 (x_i, y_i, s_i) ，将它从候选特征点中移除，否则，将它加入到最佳特征点的集合中，重复以上步骤，直到全部特征点迭代完为止。自适应距离阈值的计算公式为：

$$T = k \times \frac{1}{\sqrt{s_j}} \quad (13)$$

其中 s_j 是以特征点 p_j 为中心的密度估计值， k 表示自适应参数，通常情况下，当区域的特征点密度较高时，距离阈值也应该相应地减小，这时可以设置 k 为一个较小的值，以保证算法的灵敏度。当区域的特征点密度较低时，距离阈值应该相应地增大，这时可以设置 k 为一个较大的值，以避免出现过多的误检。

4. 实验结果与分析

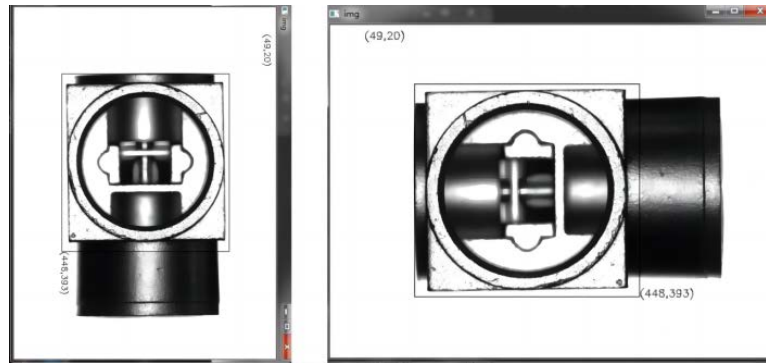
本实验在 windows10 操作系统 + Visual Studio 2022 平台 + Opencv3.5.1 环境下进行操作的，通过测试不同的图片数据集，比较不同的算法提取特征点的分布均匀度和特征点提取速度。其中 outside 数据集是室外环境下特征点复杂的两幅图像，book 数据集是尺度差异较大的两幅图像，workpiece 数据集是旋转差异较大的两幅图像，room 数据集是压缩度不同的两幅图像。该数据集图像如图 7 所示。



(a) outside 数据集



(b) book 数据集



(c) workpiece 数据集



(d) room 数据集

Figure 7. Experimental data set image
图 7. 实验数据集图像

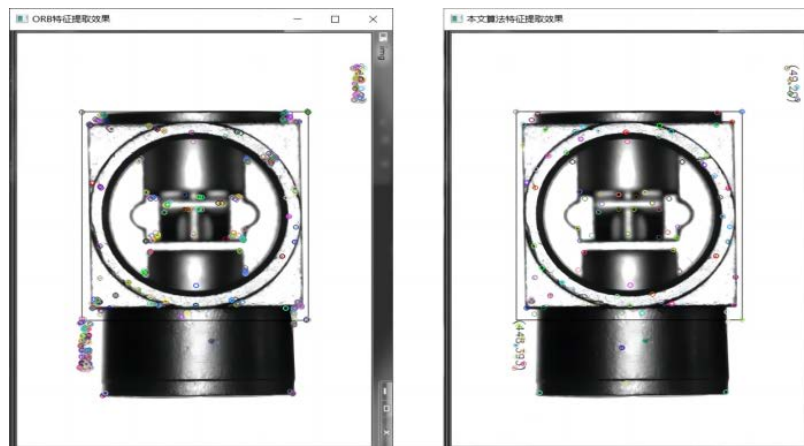
实验时分别采用 orb 算法和本文的算法对每个图片组的第一张图像进行实验对比。
不同的算法对图像特征点的提取如图 8 所示，左边为 ORB 特征提取结果，右边为本文算法的特征提取效果。



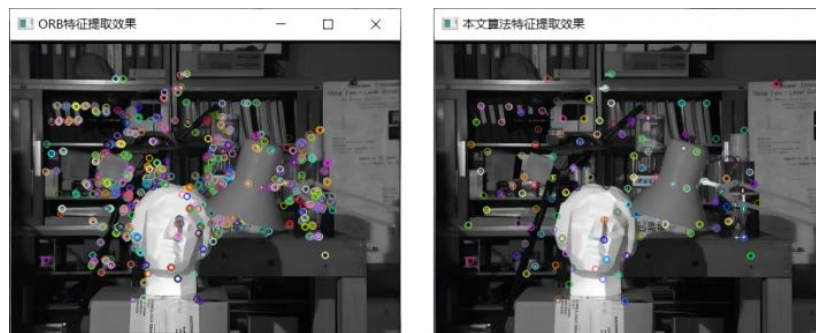
(a) outside 数据集特征提取结果



(b) book 数据集特征提取结果



(c) workpiece 数据集特征提取结果

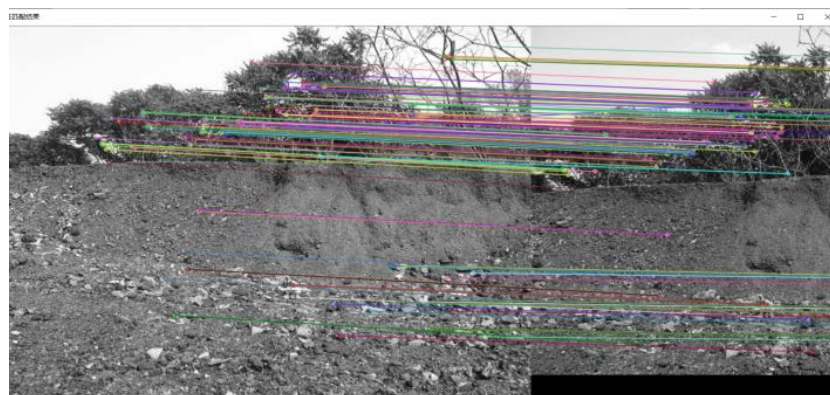


(d) room 数据集特征提取结果

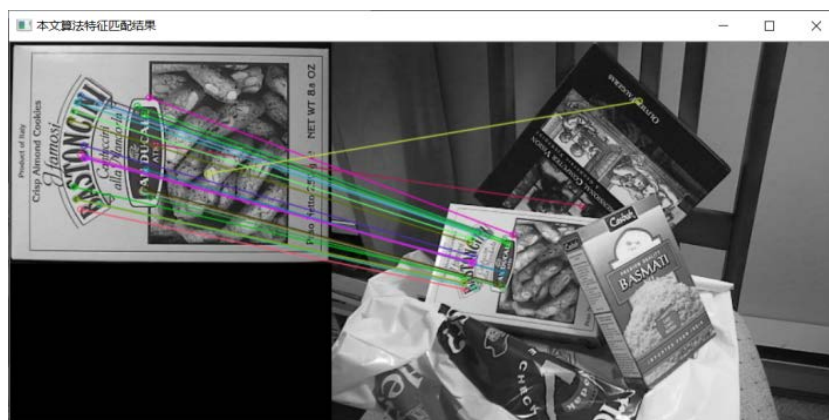
Figure 8. Feature extraction effect of different algorithms
图 8. 不同算法特征提取效果

从图 9 可以看出，针对不同的数据集，ORB 算法提取的特征点分布过于密集，特征点分布均匀程度低，特征点冗余严重，重叠率高。本文算法提取的特征点在特征点分布均匀度上优于 ORB 算法，并筛选出更具代表性的特征点，极大减少特征点的冗余程度，为特征点匹配的准确性和效率提供了基础。

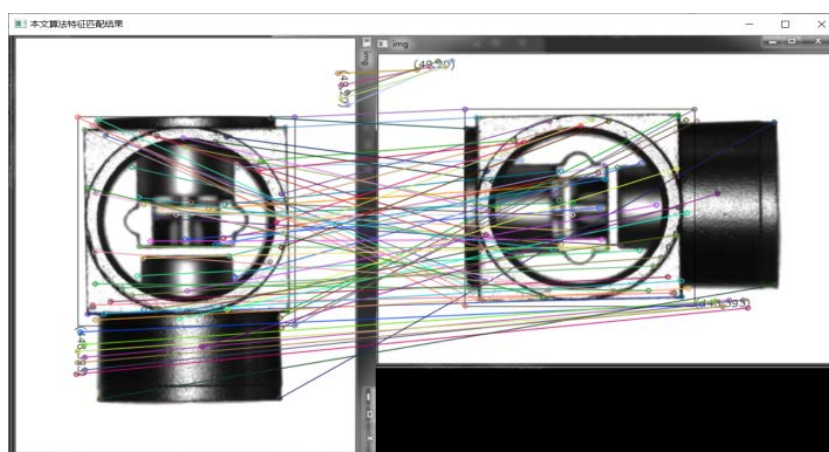
为了证实本文算法中特征点的可靠性，本实验分别对每组数据集的两幅图像做特征匹配，图 9 为本文算法的特征匹配结果。



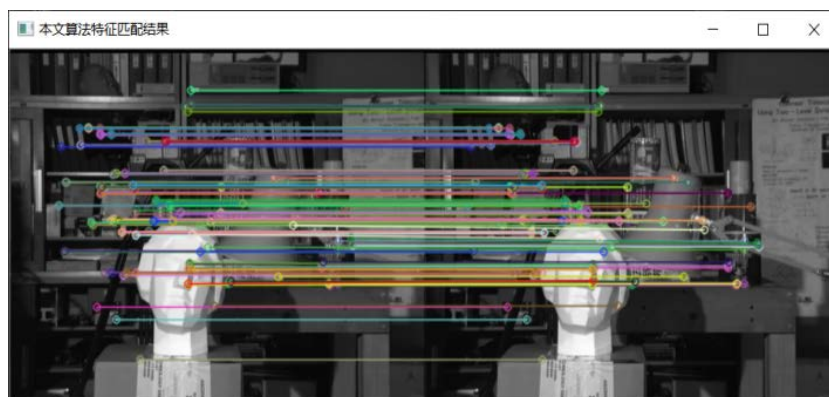
(a) outside 数据集特征匹配结果



(b) book 数据集特征匹配结果



(c) workpiece 数据集特征匹配结果



(d) room 数据集特征匹配结果

Figure 9. The feature matching results of the algorithm in this paper**图 9.** 本文算法特征匹配结果

从图 9 可以看出, 针对不同的数据集, 本文算法在匹配时只出现极少数误匹配的情况, 这表明本文算法在应对特征不同的图像时, 仍能保证较高的匹配精度。表 1 反应了不同的算法在应对不同的数据集时, 其匹配精度 m 和特征提取与匹配的时间 t 。其中匹配精度是一副图像中正确匹配的点对数与总提取特征点的比, 若两坐标之间的欧几里得距离不超过两个像素, 则认为是正确匹配。

Table 1. Match accuracy and time
表 1. 匹配精度和时间

	算法	不同算法在不同数据集下的匹配精度			
		outside	book	workpiece	room
精度%	ORB 算法	92.35	89.66	94.24	98.78
	本文算法	94.86	93.37	97.21	98.41
时间 ms	ORB 算法	50.65	43.78	39.71	43.26
	本文算法	42.48	33.62	25.97	28.53

通过表 1 可以看出, 本算法在不同数据集下均可以稳定运行, 与传统的 ORB 算法相比, 本文算法在大多数情况下能保持较高的特征匹配精度, 此外, 在处理不同的数据集时, 该算法的匹配精度也保持的较为稳定, 没有出现明显的波动。从运行时间上来看, 本文算法由于在特征处理时, 减少了对特征稀疏区域的特征点处理, 因此耗时较少。在应对不同的数据集时, 算法的运行时间较原 ORB 算法分别提高了 16.13%, 23.21%, 34.60%, 34.05%, 具有较好的实时性。

5. 结语

针对原 ORB 算法在面对特征点分布不均匀的图像做特征提取时, 常会出现特征点提取效率低, 特征点冗余的问题, 提出了一种基于区域密度划分改进的 ORB 特征提取算法。首先对图像构造图像金字塔, 其次用 FAST 特征点提取算法对图像进行特征点提取, 并使用核密度估计方法, 以每个特征点为中心, 计算该点周围的密度值, 并采用聚类算法将密度分布相似的特征点看成一簇, 根据每一簇的特征点密度和图像平均密度的比值, 将图像区域划分为三类。然后利用自适应二叉树法根据特征点密度计算阈值, 对三个区域图像进行分割, 并利用自适应非极大值抑制法对特征点进行最佳筛选, 以达到特征点筛选均匀的目的。通过实验数据可以看出, 与传统的 ORB 算法相比, 本文算法提取的特征点均匀程度更高, 在特征匹配时有着更高的精度, 同时算法的整个运行时间更短, 计算效率得到明显的改善, 但是算法还存在一些问题。在接下来的工作中, 可以对特征点的聚类方式进一步进行优化, 可以考虑采用更高级的聚类算法来实现特征点划分, 以此来进一步提高特征点提取和匹配的准确性和效果。

参考文献

- [1] 王朋, 郝伟龙, 倪翠, 张广渊, 巩慧. 视觉 SLAM 方法综述[J/OL]. 北京航空航天大学学报: 1-9. <https://doi.org/10.13700/j.bh.1001-5965.2022.0376>, 2023-05-26.
- [2] Semma, A., Hannad, Y., Siddiqi, I., Djeddi, C. and El Youssfi El Kettani, M. (2021) Writer Identification Using Deep Learning with FAST Keypoints and Harris Corner Detector. *Expert Systems with Applications*, 184, Article No. 115473.
- [3] 郑泰皓, 方子彬. 视觉 SLAM 特征与提取算法综述[J]. 计算机时代, 2020(7): 16-21. <https://doi.org/10.16644/j.cnki.cn33-1094/tp.2020.07.005>
- [4] 张龙, 郑灿. 基于 SURF 和 RANSAC 的改进目标识别算法[J]. 中国设备工程, 2022(23): 140-143.
- [5] Sun, Q., Zhang, Y., Wang, J.G. and Gao, W. (2017) An Improved FAST Feature Extraction Based on RANSAC Method of Vision/SINS Integrated Navigation System in GNSS-Denied Environments. *Advances in Space Research*, **60**, 2660-2671.
- [6] 张妮妮, 赵金龙, 何佳, 陈绥阳, 王杰. 一种基于阶阵列的 BRIEF 特征描述子[J]. 计算机技术与发展, 2023, 33(5): 81-87.
- [7] Leng, X. and Yang, J.H. (2015) Improvement of ORB Algorithm. *Proceedings of the 2015 International Conference on Materials Engineering and Information Technology Applications*, Guilin, China, 30 August 2015, 903-906. <https://doi.org/10.2991/meita-15.2015.169>

-
- [8] 李国竣, 徐延海, 段杰文, 韩石磊. 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+48. <https://doi.org/10.13474/j.cnki.11-2246.2021.0269>
- [9] 许宏科, 秦严严, 陈会茹. 基于改进 ORB 的图像特征点匹配[J]. 科学技术与工程, 2014, 14(18): 105-109+128.
- [10] 孙浩, 王朋. 一种基于区域划分的改进 ORB 算法[J]. 北京航空航天大学学报, 2020, 46(9): 1763-1769. <https://doi.org/10.13700/j.bh.1001-5965.2020.0054>
- [11] 陈继清, 桂海宁, 王志奎, 龙腾, 黄春林. 基于特征点法的自适应 SLAM 改进方法[J]. 广西大学学报(自然科学版), 2022, 47(6): 1611-1625. <https://doi.org/10.13624/j.cnki.issn.1001-7445.2022.1611>
- [12] 夏俊楠, 魏伟, 尹力, 洪梦瑶, 薄立明. 基于四叉树算法的“三区空间”形态效率评价方法[J]. 地球信息科学学报, 2023, 25(3): 450-467.
- [13] 朱宏辉, 王嘉豪, 朱轶. 基于特征点密度峰值的视觉伺服目标选择方法[J]. 计算机工程与设计, 2021, 42(8): 2350-2357. <https://doi.org/10.16208/j.issn1000-7024.2021.08.034>
- [14] 吴际洲. 基于核密度估计和特征点检测的运动对象分割改进模型[J]. 中国高新技术企业, 2012(9): 10-12. <https://doi.org/10.13535/j.cnki.11-4406/n.2012.09.062>