

车底危险物图像快速拼接算法

庄建军*, 金鑫

南京信息工程大学电子与信息工程学院, 江苏 南京 210044

摘要 针对车底危险物图像拼接中存在特征点匹配精度低、匹配速度慢、拼接处存在裂缝以及拼接时间久的问题, 提出一种车底危险物图像快速拼接算法。首先, 利用角点检测 (FAST) 算法进行图像特征点提取, 再用二进制鲁棒不变可扩展关键点 (BRISK) 算法对保留的特征点进行特征描述; 其次, 用快速最近邻搜索 (FLANN) 算法进行粗匹配; 接着, 使用渐进一致采样 (PROSAC) 算法进行特征点提纯; 最后, 利用拉普拉斯金字塔算法进行图像融合与拼接。实验结果表明: 在车底危险物图像数据中, 与 SIFT、SURF、ORB 算法相比, 所提算法的图像特征匹配精度分别提高 13.10 百分点、8.59 百分点、11.27 百分点, 匹配时间分别缩短 76.26%、85.36%、10.27%, 图像拼接时间分别缩短 63.73%、64.21%、20.07%, 拼接处不存在明显裂痕。因此, 基于 FAST、BRISK、PROSAC 和拉普拉斯金字塔组合的图像拼接算法是一种优质的图像快速拼接算法。

关键词 图像处理; FAST 算法; BRISK 算法; PROSAC 算法; 拉普拉斯金字塔算法; 图像融合

中图分类号 TP391

文献标志码 A

DOI: 10.3788/LOP231652

Fast Image-Stitching Algorithm of Dangerous Objects Under Vehicles

Zhang Jianjun*, Jin Xin

School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China

Abstract To address challenges involving low accuracy in feature point matching, low matching speed, cracks at the stitching points, and extended stitching time in vehicle undercarriage threat detection imaging, an optimized image-stitching algorithm is proposed. First, the corner detection (FAST) algorithm is used to extract image feature points, and then, the binary robust invariant scalable key point (BRISK) algorithm is used to describe the retained feature points. Second, the fast nearest neighbor search (FLANN) algorithm is used for coarse matching. Next, the progressive uniform sampling (PROSAC) algorithm is used for feature point purification. Finally, the Laplace pyramid algorithm is used for image fusion and stitching. The experimental results show that, when compared with SIFT, SURF, and ORB algorithms, the proposed algorithm improves the image feature matching accuracy by 13.10 percentage points, 8.59 percentage points, and 11.27 percentage points, respectively, in the image data of dangerous objects under the vehicle. The matching time is shortened by 76.26%, 85.36%, and 10.27%, respectively. The image-stitching time is shortened by 63.73%, 64.21%, and 20.07%, respectively, and there are no evident cracks at the stitching point. Therefore, the image-stitching algorithm based on the combination of FAST, BRISK, PROSAC, and Laplace pyramid is a high-quality fast image-stitching algorithm.

Key words image processing; FAST algorithm; BRISK algorithm; PROSAC algorithm; Laplace pyramid algorithm; image fusion

1 引言

如今, 犯罪分子越来越善于利用车辆来隐藏危险物, 这对公众安全造成极大威胁。在车辆的安全检查中, 车底的危险物最难检测, 因此车底检测已经成为安

保领域研究的重点和难点^[1]。目前, 基于深度学习的车底危险物检测成为主流方法^[2], 但在检测之前, 需要获取完整的车底危险物图像数据。然而, 由于车辆底盘较宽较长, 摄像头无法一次性拍摄到完整的车底危险物图像。因此, 车底危险物图像快速拼接算法变得

收稿日期: 2023-07-03; 修回日期: 2023-07-26; 录用日期: 2023-08-08; 网络首发日期: 2023-08-21

基金项目: 国家重点研发计划(2021YFE0105500)、国家自然科学基金(N62171228)、江苏高校“青蓝工程”资助

通信作者: *jjzhuang@nuist.edu.cn

尤为重要。

针对车底图像拼接这一现实问题,国内学者做过一系列研究。章加兵^[3]设计出地铁底部图像拼接与关键部件螺栓检测系统,针对多线阵相机拍摄的图像,采用 scale-invariant feature transform (SIFT)^[4] 算法对重叠区域进行特征匹配,但其使用的图像配准算法匹配时间长、正确率低,且加权平均融合算法减少了图像中的细节信息,融合的结果较差。席永明^[5]提出一种结合分形特征和灰度相关性的车底图像配准算法,首先利用分形特征对图像进行快速匹配,然后使用灰度相关性去除初步匹配中的错误匹配区域,但该算法特征匹配错误点较多,配准速度较慢,且使用灰度相关性进行精细匹配可能会导致特征点利用率低。邢昆^[6]提出的车辆检查系统中图像拼接部分采用 speeded-up robust features (SURF)^[7] 特征点检测、random sample consensus (RANSAC)^[8] 算法进行特征点提纯以及加权融合算法进行图像融合。SURF 算法通过 Hessian 矩阵来计算图像特征点的尺度和方向^[9],但鲁棒性略逊于 SIFT 算法,且 RANSAC 算法的特征点提纯速度较慢,内点数不足,会影响后续图像拼接的效果。

针对上述研究存在的问题,本文提出一种基于 features from accelerated segment test (FAST)^[10]、binary robust invariant scalable keypoints (BRISK)^[11]、progressive sample consensus (PROSAC)^[12] 以及拉普拉斯金字塔图像融合算法的图像拼接算法。为了验证所提算法的有效性,在车底危险物图像数据上与 SIFT 算法、SURF 算法、oriented FAST and rotated BRIEF (ORB)^[13] 算法、RANSAC 算法以及加权平均融合算法进行比较,主要从特征匹配速度、匹配准确率、特征点利用率、拼接速度和拼接效果等 5 个方面进行评价^[14]。实验结果表明,所提算法特征匹配速度、匹配准确率、特征点利用率以及拼接速度较 SIFT 算法、SURF 算法、ORB 算法都有较大提升,拼接后的图像无拼接缝、无重影,色彩和亮度过渡真实自然,是一种实时性高、拼接效果良好的图像快速拼接算法。

2 车底危险物图像快速拼接算法

为了快速且高效地完成车底危险物图像拼接,提出一种车底危险物图像快速拼接算法。图 1 为车底危险物图像快速拼接算法流程图。首先,使用 FAST 特征提取算法提取待拼接图像中的关键点,使用 BRISK 算法更好地捕捉图像特征,从而提高特征匹配的速度和准确性;其次,使用 FLANN 算法进行粗匹配,并采用 PROSAC 算法来剔除错误匹配点,减少迭代运算次数,提高匹配速度和稳定性;最后,使用拉普拉斯金字塔算法进行图像融合,以避免在拼接过程中出现鬼影和截断现象,可以使得图像过渡更加平滑、拼接后的图像看起来更加自然和连续。

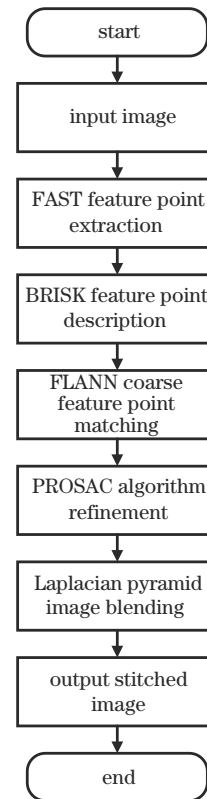


图 1 车底危险物图像快速拼接算法流程图

Fig. 1 Flow chart of fast stitching algorithm for dangerous images under car

2.1 FAST 特征提取

FAST 特征提取算法是一种基于图像像素强度变化的角点检测算法,能够快速准确地检测出特征点^[15],包括非特征点的筛选、特征点的识别以及非极大值抑制等步骤^[16]。具体而言,FAST 算法获取特征点的步骤如下:

1) 在待检测图像中任选一点 P , 在以像素 P 为中心、半径为 3 的圆上取 16 个像素点 (P_1, P_2, \dots, P_{16}), 如图 2 所示。

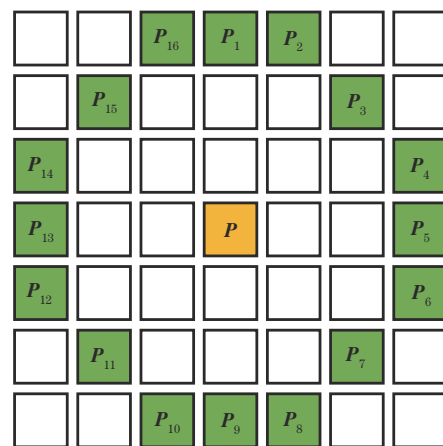


图 2 FAST 特征点检测算法的示意图

Fig. 2 Schematic diagram of FAST feature point detection algorithm

2) 首先定义一个大小为 30 的阈值 T , 然后, 计算中心像素 P 与其周围的 P_1, P_9, P_5, P_{13} 像素之间的差值。如果这些差值中至少有 3 个的绝对值超过阈值 T , 那么这个中心像素 P 就被认为是候选角点。该操作能快速而可靠地筛选出图像中的角点特征。

3) 如果一个像素 P 被确定为候选点, 接下来计算它周围的 16 个像素点 ($P_1 \sim P_{16}$) 与中心点 P 的像素差。如果这 16 个像素差中至少有连续的 9 个超过预先定义的阈值, 那么可以确定该中心点 P 是一个角点。

4) 非极大值抑制。考察特征点 P 周围的一个邻域, 并计算该邻域内所有特征点与圆周上 16 个像素点灰度差的绝对值的总和。比较这个得分值与邻域内其他特征点的得分值, 如果特征点 P 的得分值是邻域内最大的, 那么我们将保留特征点 P 。

算法流程如图 3 所示。从原理上分析, FAST 算

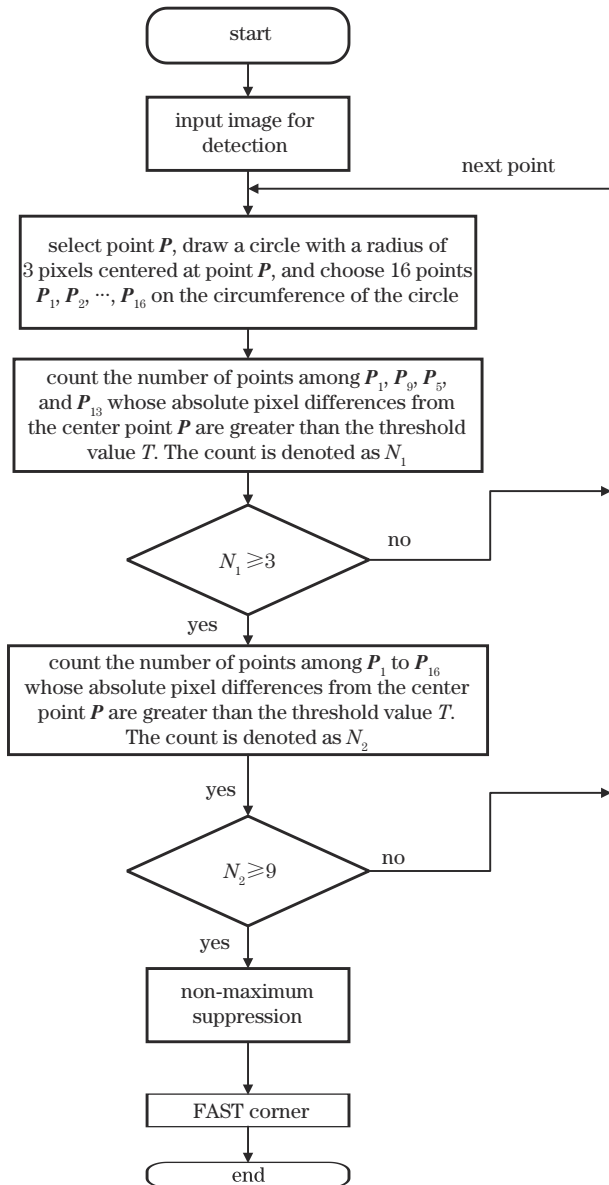


图 3 FAST 特征点提取流程图

Fig. 3 Flow chart of FAST feature point extraction

法原理简单, 计算量小, 且阈值 T 设置为 30 时, 更有利于角点提取, 因此选取 FAST 算法来快速准确地获取图像特征点。

2.2 BRISK 描述符引入

针对 BRIEF 描述子速度较慢, 且特征描述能力相对较弱, 容易受到噪声和旋转变换影响的问题, 所提算法采用 BRISK 描述符对提取到的特征点进行特征描述。BRISK 描述符是一种基于二进制的局部特征描述符, 以特征点为中心, 在其周围围绕 4 个同心圆, 这 4 个同心圆的半径是固定的, 在这些圆上均匀地采样像素点, 分别得到 10、14、15、20 个采样点^[17], 特征点本身也被看作一个采样点, 在这个过程中, 共有 60 个采样点, 将这 60 个采样点两两进行组合, 可以得到 1770 个组合。对应的数学表示形式为

$$B = \left\{ (P_i, P_j) \in \mathbf{R}^2 \times \mathbf{R}^2 \mid i \leq N, j < i, N = 60 \right\}, \quad (1)$$

式中: \mathbf{R}^2 代表二元组; P_i, P_j 为 60 个采样点中的一个。在 BRISK 中, 根据采样点间的距离长短从 B 中取出集合 M 和 N 。 M 和 N 的定义如下:

$$\begin{cases} M = \left\{ (P_i, P_j) \in B \mid d_{i,j} < \delta_M \right\} \\ N = \left\{ (P_i, P_j) \in B \mid d_{i,j} < \delta_N \right\} \end{cases}, \quad (2)$$

$$d_{i,j} = \|P_i - P_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

式中: P_i 的坐标为 (x_i, y_i) ; P_j 的坐标为 (x_j, y_j) ; $\delta_M = 9.75t$, $\delta_N = 13.67t$, t 表示尺度。

集合 N 用于构建特征点的角度属性。设 $(P_i, P_j) \in N$, 则这两个采样像素的局部梯度如式 (3) 所示:

$$\begin{cases} \Delta x = \frac{(x_i - x_j)[C(P_i, \sigma_i) - C(P_j, \sigma_j)]}{d_{i,j}^2} \\ \Delta y = \frac{(y_i - y_j)[C(P_i, \sigma_i) - C(P_j, \sigma_j)]}{d_{i,j}^2} \end{cases}, \quad (3)$$

式中: $C(P_i, \sigma_i)$ 为采样像素 P_i 高斯平滑后的灰度值; $C(P_j, \sigma_j)$ 为采样像素 P_j 高斯平滑后的灰度值; σ_i, σ_j 为高斯平滑时使用的标准差。特征点的角度 α 如式 (4) 所示:

$$\begin{cases} g_x = \frac{1}{l} \sum_N \Delta x \\ g_y = \frac{1}{l} \sum_N \Delta y \\ \alpha = \arctan\left(\frac{g_y}{g_x}\right) \end{cases}. \quad (4)$$

集合 M 用于构建特征点的描述符, BRISK 描述符是基于二值位字符串形成的, 字符串中每一位 a 如式 (5) 所示:

$$a = \begin{cases} 1, & C(P_j, \sigma_j) > C(P_i, \sigma_i) \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

式中: P_j^α 和 P_i^α 是经过 α 旋转以后的采样模板集合 M 中的一对采样点。

与 BRIEF 描述子相比, BRISK 描述符采用更复杂的二进制编码来计算描述符, 这提高了描述符的区分度, 减少了噪声和模糊的影响, 从而增强了描述符的鲁棒性, 这使得 BRISK 描述符能够在更多不同的图像环境下工作, 并且具有更高的匹配精度。

2.3 特征点粗匹配

采用 FLANN 算法进行特征点粗匹配, FLANN 算法可以在高维数据空间中搜索近似最近邻。具体步骤如下:

1) 提取两张图像的特征点;

2) 利用 FLANN 将其中一张图像中的特征点进行索引, 构建索引结构;

3) 遍历另一张图像中的特征点, 并利用 FLANN 在索引结构中进行最近邻搜索, 找到其对应的最近邻特征点;

4) 根据最近邻距离和次近邻距离之比进行特征点匹配, 筛选出最优的匹配结果。

FLANN 的特征匹配速度快, 适用于大规模数据集, 但精度有限, 匹配结果可能存在误匹配或漏匹配的情况, 因此后续需要对特征点的匹配结果进行提纯, 从而提升匹配正确率。

2.4 PROSAC 算法

RANSAC 算法从数据集中随机抽取样本点, 然后通过重复采样、估计和验证的过程来找到最优的模型, 但它没有考虑样本之间的差异性, 导致算法效率低、运算时间长, 特别是在特征点精匹配方面表现不佳。因此, 采用 PROSAC 算法来提高特征点匹配的精度, 为后续的图像融合做准备。

在 PROSAC 算法中, 生成集大小的确定和采样停止准则的选择是关键步骤。

1) 生成集大小: 在集合 U 中, 需要使用生长函数 $g(k)$ 定义排行在前列的 n 项的对应关系, 对应关系的采样次数设定为 k :

$$g(k) = \min\{n: T \geq k\}, \quad (6)$$

式中: k 为判断依据, 把采样次数大于 k 的最小值用 n 表示; T 表示在采样中, 样本点来自集合 U 的次数。

2) 采样停止准则: I 作为非正确模型局内点的概率不能高于 5%, 此为非随机性。若是在 U 中, 有比 I 局内点更多的情况, 而且此种情形在经历 k 次采样后依然未发现的概率高于 5%, 此为极大性。若同时满足非随机性和极大性, 则采样停止。

相比于 RANSAC 算法, PROSAC 算法能够有效地减少盲目性, 减少迭代次数以提高提纯速度^[18]。PROSAC 算法并不是从所有数据中提取采样点, 而是使用特殊的启发式评估函数来选择采样点, 对相似性较高的样本点进行排序, 避免陷入局部最优解, 提高了

算法的鲁棒性, 然后再从具有最高质量的数据子集中提取样本点, 最后通过样本数据去估计模型。PROSAC 算法能够动态地更新采样概率, 在找到最优解之前逐渐改进估计值, 获得更高精度的估计结果, 并且根据已经找到的最优解的质量来调整, 从而更好地搜索解空间, 提高算法的效率。

2.5 拉普拉斯金字塔算法

拉普拉斯金字塔图像融合算法是一种常用的图像融合算法, 其主要思想是通过多尺度分解, 然后将不同尺度的图像信息进行融合得到最终结果, 主要步骤如下:

1) 通过不断对图像进行下采样(即缩小图像尺寸)和高斯滤波构建高斯金字塔, 如式(7)所示, 对原始图像进行高斯金字塔分解, 得到多个不同尺度的图像。

$$\begin{cases} G_l(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) G_{l-1}(2i+m, 2j+n) \\ w(m, n) = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \end{cases}, \quad (7)$$

式中: l 为金字塔的层数; (m, n) 、 (i, j) 为图像大小的参数; $G_l(i, j)$ 为第 l 层高斯金字塔图像; $G_{l-1}(2i+m, 2j+n)$ 为第 $l-1$ 层高斯金字塔图像; $w(m, n)$ 为高斯卷积核。

2) 通过将高斯金字塔中每一层图像与其上一层图像在上采样后的高斯金字塔图像相减得到拉普拉斯金字塔, 如式(8)所示, 对每个尺度的图像进行拉普拉斯金字塔分解, 得到图像金字塔结构以及多个包含细节信息的图像。

$$L_l = G_l - \text{pyrup}(G_{l+1}), \quad (8)$$

式中: L_l 为第 l 层拉普拉斯金字塔图像; pyrup 为对图像进行上采样。

3) 从低到高依次遍历两个图像的金字塔结构, 将相同尺度的图像细节信息融合。

加权平均融合算法虽然可以较好地处理一些简单的图像融合任务, 但是在处理一些复杂的图像融合任务时, 它的权重需要手动调整, 容易出现融合效果不佳的情况。拉普拉斯金字塔算法是基于图像的多尺度分解, 它的每一层都包含原图像的一部分信息, 可以直接对每一层图像进行加权平均得到最终结果, 无须手动调整权重。在图像融合时, 它可以两个图像在不同尺度上的细节信息融合, 从而获得更加细致和自然的融合结果。因此, 拉普拉斯金字塔算法在对车底图像融合效果这样要求较高的场景下更具有优势。

3 实验结果与分析

为验证所提算法的效果, 分别进行图像配准和图

像融合拼接实验,共设计 5 组实验:实验 1 用于验证所提算法相对于 SIFT 算法、SURF 算法以及 ORB 算法在特征点匹配时间以及匹配准确率上的优势;实验 2 用于验证 PROSAC 算法相对于 RANSAC 算法在特征点精匹配上的时间与准确率的的优势;实验 3 用于验证拉普拉斯融合算法相对于加权平均融合算法在图像融合效果上的优势;实验 4 为消融实验,用于验证所提算法的整体效果;实验 5 用于验证所提算法相对于其他常用图像拼接算法在拼接效果上的优势。

3.1 实验数据

鉴于车底危险物图像较少,因此使用自建的车底

危险物图像数据。针对车底空间狭小且不易拍摄到车底图像的问题:首先,将汽车停在起降台上,使其升高一小段距离,再在车底放入不同的模拟危险物;其次,使用智能机器人小车搭载超清摄像头进入车底拍摄。机器人小车上装有 Wi-Fi 模块,能与手机上的 App 进行交互,在小车进入车底前,调整摄像头水平向上,小车每向前一次,通过手机上的 App 能看到车底的情况,这时按下拍摄按键,就能成功拍摄一张车底的碎片化图像。图 4 为 3 组待拼接图像示例,其中,每组图像车底和模拟危险物均不同,图像大小为 640×480 。

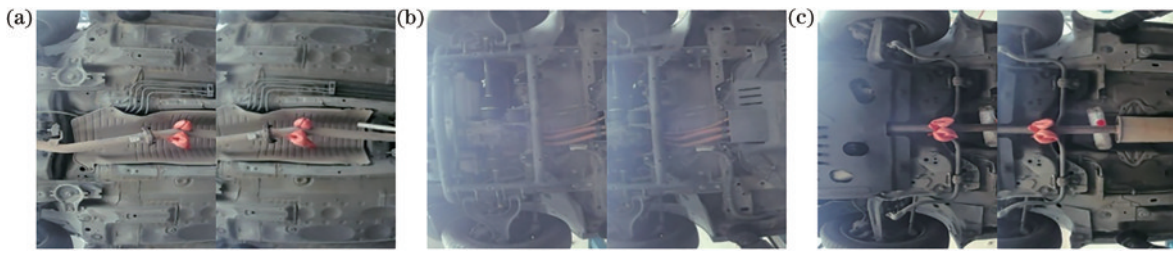


图 4 3 组待拼接图像。(a)图像组 1;(b)图像组 2;(c)图像组 3

Fig. 4 Three sets of images to be spliced. (a) Image group 1; (b) image group 2; (c) image group 3

3.2 实验环境

实验在 Visual Studio 和 OpenCv 下基于 C++ 编程语言完成。硬件环境如下:操作系统 Windows 11; CPU AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz;内存大小 16 GB。

3.3 评价指标

为量化所提算法与传统算法的差异^[19],统计特征点检测时间(Fpdt)、粗匹配数目(Coamc)、正确匹配数目(Cormc)、特征点匹配时间(Fpmt)、匹配精度

(Ma)^[20]、精匹配数目(Amc)、特征点利用率(Fpur)、特征点提纯时间(Fprt)、拼接时间(St)等参数并对其进行比较。其中,匹配精度和特征点利用率分别为正确匹配数目与粗匹配数目之比和精匹配数目与粗匹配数目之比。

3.4 特征点匹配结果对比

图 5 为 SIFT 算法、SURF 算法、ORB 算法以及所提算法的图像特征点匹配效果图,该效果图是经过特征点提纯后的效果图。

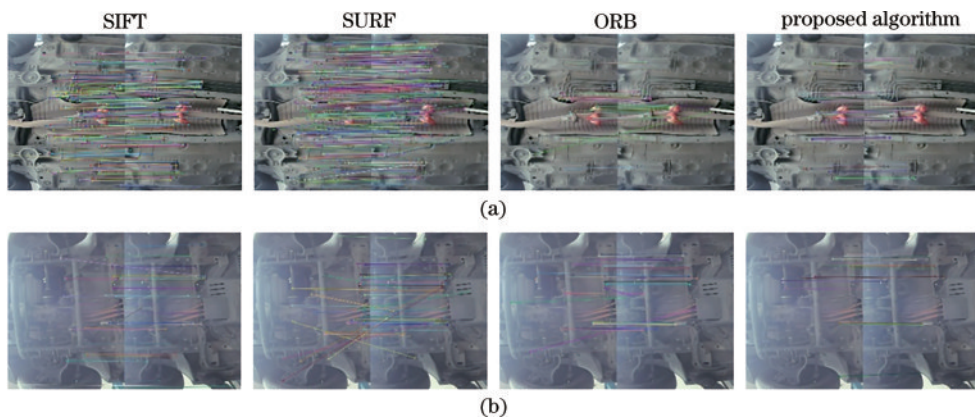


图 5 4 种方法的图像特征点匹配效果图。(a)图像组 1;(b)图像组 2

Fig. 5 Image feature point matching renderings of four methods. (a) Image group 1; (b) image group 2

由图 5 可知,与 SIFT 算法、SURF 算法、ORB 算法相比,所提算法有效控制了特征匹配的范围,使得匹配精度有所上升,并显著控制了特征点的数目,使得算法效率提升。表 1 为图像组 1 和 2 中 4 种方法的图像特征点匹配指标。

从表 1 的数据可以看出,所提算法在两组图像实验中的特征点检测时间、特征点匹配时间和匹配精度相比 SIFT 算法、SURF 算法和 ORB 算法都表现出明显的优势。在图像组 1 中,所提算法相对于这 3 种算法,在特征点匹配时间上分别减少 81.32%、82.71%

表 1 图像组 1 和 2 中 4 种算法的图像特征点匹配指标

Table 1 Image feature point matching indexes of four algorithms in image group 1 and 2

Index	Image group 1				Image group 2			
	SIFT	SURF	ORB	Proposed algorithm	SIFT	SURF	ORB	Proposed algorithm
Fpdt /s	0.491	0.672	0.361	0.342	0.376	0.501	0.354	0.318
Coamc	1425	2007	501	378	299	515	410	367
Cormc	1097	1709	435	349	203	354	253	289
Fpmt /s	1.194	1.290	0.237	0.223	0.264	0.634	0.089	0.076
Ma /%	76.98	85.15	86.82	92.33	67.89	68.73	61.70	78.74

和 5.90%，在匹配精度上分别提高 15.35 百分点、7.18 百分点和 5.51 百分点。而在图像组 2 中，所提算法相对于这 3 种算法，在特征点匹配时间上分别减少 71.21%、88.01% 和 14.61%，在匹配精度上分别提高 10.85 百分点、10.01 百分点和 17.04 百分点。由此可见，所提算法在不同的图像情况下都表现出良好的鲁棒性和快速匹配速度，为图像匹配和拼接技术在实际

环境变化中的应用提供了可靠的保证。

3.5 特征点提纯结果对比

为了对比 RANSAC 算法和改进的 PROSAC 算法关于特征点提纯的效果，依旧选取上文提到的图像组 1 和 2，并分别使用 ORB+RANSAC 算法和 ORB+PROSAC 算法、FAST+BRISK+RANSAC 算法和所提算法进行实验，如表 2 所示。

表 2 图像组 1 和 2 中特征点提纯实验对比

Table 2 Experimental comparison of purification of feature points in image groups 1 and 2

Index	Image group 1				Image group 2			
	ORB+RANSAC	ORB+PROSAC	FAST+BRISK+RANSAC	Proposed algorithm	ORB+RANSAC	ORB+PROSAC	FAST+BRISK+RANSAC	Proposed algorithm
Cormc	501	501	378	378	410	410	367	367
Amc	104	119	82	105	43	56	48	68
Fpur /%	20.76	23.75	21.69	27.77	10.48	13.65	13.07	18.52
Fprt /s	0.235	0.213	0.219	0.138	0.185	0.157	0.158	0.102

由表 2 可知，在两组图像实验中，与 RANSAC 算法相比，PROSAC 算法提取的精匹配对数量更多，即提取的内点数量更多。因此，在相同的粗匹配对数量下，PROSAC 算法能够更快、更准确地提取图像特征点，减少噪声点的数量，从而提高特征点检测的效率和特征点的利用率。所提算法特征点利用率比 ORB+RANSAC 算法平均提高 4.45 百分点，更高的特征点利用率，意味着算法整体的鲁棒性能更有保障。在特征点提纯时间的对比中，无论是在 ORB 算法还是在所提算法中，PROSAC 算法的配准时间均小于 RANSAC 算法，可见其能进一步加速特征匹配的过程。

3.6 图像融合效果对比实验

本实验中，分别采用加权平均融合算法和拉普拉斯金字塔图像融合算法来对比两组车底危险物图像的融合效果，所得到的实验效果对比图如图 6 所示。

从图像融合效果对比图可以看出，加权平均融合法得出的结果总是差强人意，图像融合之后，存在明显的缝隙，并且颜色不自然，但使用拉普拉斯金字塔图像融合算法得出的图像，其重叠部分的接缝不明显，整体的全景图像更平滑可观，最终达成的图像效果更好。

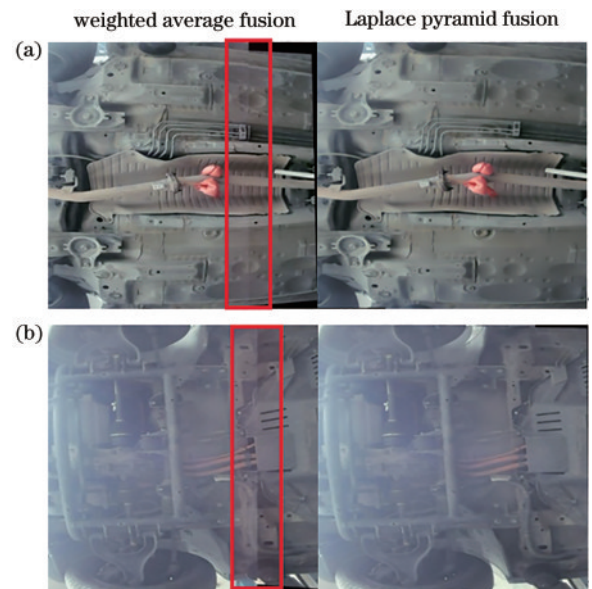


图 6 图像融合效果对比图。(a) 图像组 1；(b) 图像组 2
Fig. 6 Image fusion effect comparison. (a) Image group 1; (b) image group 2

3.7 消融实验

选用图像组 3 进行消融实验，表 3 为消融实验数据，分别为 FAST+BRISK 算法、FAST+BRISK+RANSAC 算法、FAST+BRISK+PROSAC 算法、

表 3 消融实验数据
Table 3 Ablation experimental data

Algorithm	Fpdt /s	Fpmt /s	Ma /%	Fpur /%	St /s
①ORB	0.231	0.133	73.33	18.91	0.362
②FAST+BRISK	0.189	0.085	86.54		
③FAST+BRISK+RANSAC	0.190	0.087	86.11	20.15	
④FAST+BRISK+PROSAC	0.189	0.086	87.82	25.58	
⑤FAST+BRISK+PROSAC+ Weighted average fusion	0.189	0.086	87.82	25.58	0.328
⑥Proposed algorithm	0.189	0.086	87.82	25.58	0.297

FAST+BRISK+PROSAC+加权平均融合算法以及所提算法。

消融实验②与①对比表明,采用BRISK描述符替换ORB算法中的BRIEF描述子后,特征点检测时间减少18.18%,特征点匹配时间减少35.33%。消融实验④与③对比表明,采用PROSAC算法进行提纯,不但可以提高特征点匹配精度,而且能够提升特征点利用率。消融实验⑥和⑤对比表明,与加权融合平均算法相比,拉普拉斯金字塔融合算法使得拼接时间更短,速度更快。总体而言,所提算法在特征点检测时间、特征点匹配时间、匹配精度、特征点利用率以及拼接时间上都更有优势,是一种快速且高效的图像拼接算法。

3.8 两幅及多幅图像拼接对比实验

在两幅图像拼接实验中,选用图像组1、2以及图像组3这3组图像进行图像拼接对比实验,分别采用

SIFT算法、SURF算法、ORB算法和所提算法。通过拼接时间、重影情况、明显拼缝的存在情况以及均方根误差(E_{RMSE})来全面评估图像拼接的质量。而对于图像配准精度的评价,则可以通过比较不同算法得到的均方根误差值来进行准确度的衡量。

$$E_{RMSE} = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \sqrt{[f'(i,j) - f(i,j)]^2}, \quad (9)$$

式中: M 、 N 是待拼接图像重叠区域的大小; $f'(i,j)$ 代表参考图像重叠区域像素; $f(i,j)$ 代表待拼接图像重叠区域像素。均方根误差越小,拼接精度越高,拼接效果越好。两幅图像拼接效果图如图7所示。

从图7可看出:图像组1中,SIFT算法和ORB算法的图像中有明显的拼接缝,SURF算法的图像中出现了重影;图像组2中,SIFT算法和SURF算法的图像中出

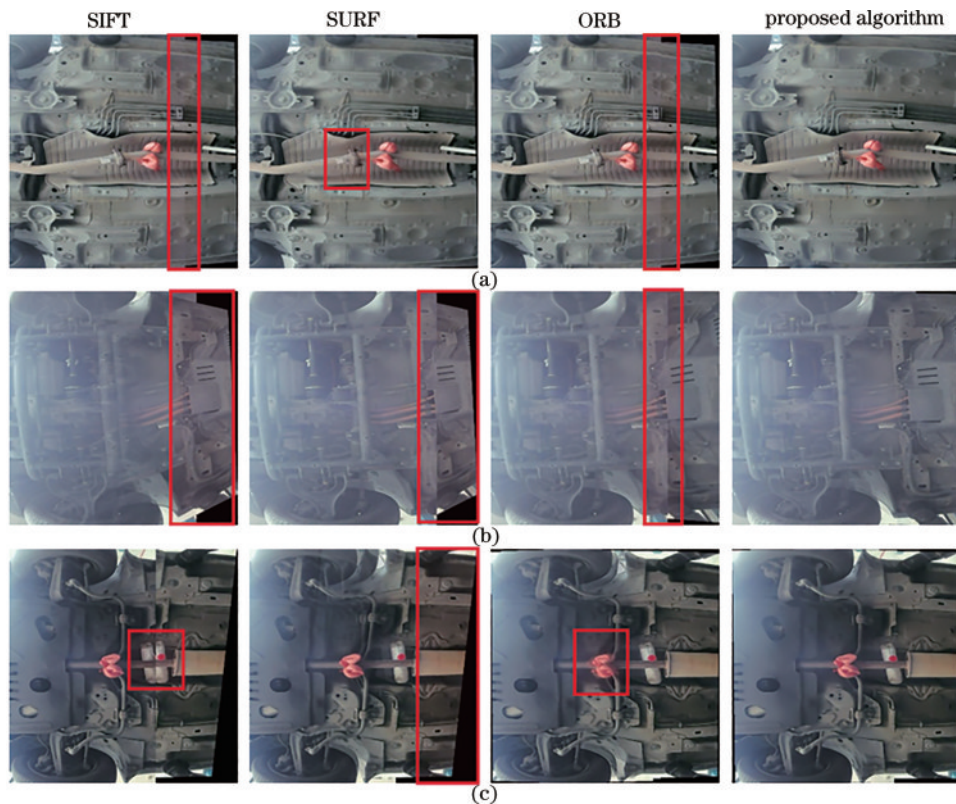


图 7 两幅图像拼接对比实验。(a)图像组1;(b)图像组2;(c)图像组3

Fig. 7 Two images mosaic comparison experiment. (a) Image group 1; (b) image group 2; (c) image group 3

现拼接断裂的情况,影响整个图像的拼接效果,ORB 算法的图像中出现明显的拼接缝;图像组 3 中,SIFT 算法和 ORB 算法的图像中车底的危险物处出现明显的拼接重影,SIFT 算法和 SURF 算法出现了明显的拼接断裂情况且边界有黑框。但所提算法在这 3 组图像的拼接中,没有出现明显的拼接缝、重影以及拼接断裂情况,车底的危险物清晰可见,图像拼接效果较好。

表 4 为两幅图像拼接效果的客观评价:图像组 1

中,在图像拼接总时间上,所提算法与 SIFT 算法、SURF 算法以及 ORB 算法相比,分别缩短了 65.07%、65.15% 和 21.08%;图像组 2 中,在图像拼接总时间上,分别缩短 49.93%、49.00% 和 21.19%;图像组 3 中,在图像拼接总时间上,分别缩短 76.20%、78.49% 和 17.96%。所提算法在 3 组图像实验中的 E_{RMSE} 均最低,由此可见其拼接的图像精度最高,效果最好。

表 4 两幅图像拼接实验数据

Table 4 Two image stitching experimental data

Image grouping	Algorithm	Is there any ghosting	Visible stitching seams	E_{RMSE}	Stitching time /s
Image group 1	SIFT	No	Yes	2.06	1.715
	SURF	Yes	Yes	2.08	1.719
	ORB	No	Yes	1.91	0.759
	Proposed algorithm	No	No	1.19	0.599
Image group 2	SIFT	No	Yes	2.42	0.765
	SURF	No	Yes	1.95	0.751
	ORB	No	Yes	2.19	0.486
	Proposed algorithm	No	No	1.47	0.383
Image group 3	SIFT	Yes	Yes	1.87	1.248
	SURF	No	Yes	2.08	1.381
	ORB	Yes	No	1.75	0.362
	Proposed algorithm	No	No	1.41	0.297

图 8 为使用所提算法得到的多张车底危险物图像拼接图,可以看出,最终拼接的图像没有明显的拼接缝,没有出现重影现象,且重叠区域的过渡十分自然。此外,图像的亮度变化也非常均匀。综合来看,拼接效果较为出色,达到了较好的结果。

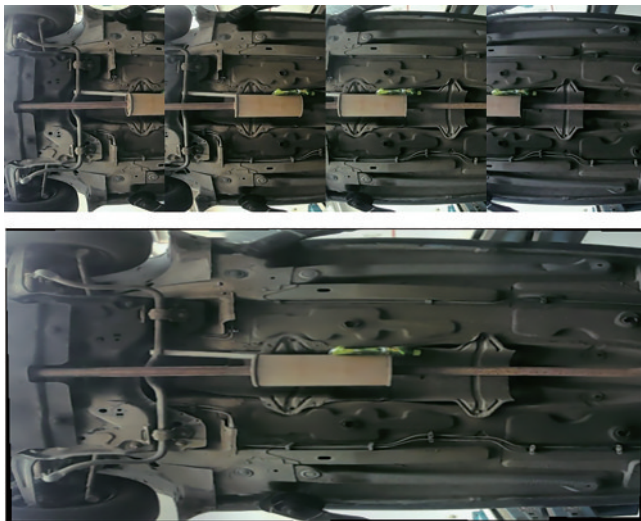


图 8 多幅图像拼接效果图

Fig. 8 Multiple image stitching renderings

4 结 论

提出一种车底危险物图像快速拼接算法,通过车

底危险物图像进行实验,所提算法的特征点检测平均时间为 0.33 s,特征点匹配平均时间为 0.149 s,匹配平均精度为 85.54%、拼接平均时间为 0.426 s。所提算法与传统的 SIFT、SURF、ORB 算法相比,图像特征匹配精度与特征点利用率均有所提高,匹配时间和图像拼接时间均有所减少。在图像拼接效果中,所提算法没有产生明显的拼接痕迹,图像拼接更加自然,其能够实现快速、高效、准确的图像拼接。

参 考 文 献

- [1] 高春艳, 吕盛璠, 吕晓玲, 等. 一种安保机器人车底高危目标检测方法[J]. 机械设计与制造, 2022(8): 282-286.
Gao C Y, Lü S F, Lü X L, et al. High-risk target detection method for security robot vehicle bottom[J]. Machinery Design & Manufacture, 2022(8): 282-286.
- [2] 徐哲玮, 刘昭, 包建东, 等. 基于改进 YoloX-s 的密贴检查器故障检测方法[J]. 电子测量技术, 2022, 45(12): 91-98.
Xu Z W, Liu Z, Bao J D, et al. A fault detection method of the closure detectors based on the improved YoloX-s [J]. Electronic Measurement Technology, 2022, 45(12): 91-98.
- [3] 章加兵. 地铁底部图像拼接与关键部件螺栓检测系统研究[D]. 南京: 南京理工大学, 2022.
Zhang J B. Research on image mosaic of subway bottom and bolt detection system of key components[D]. Nanjing: Nanjing University of Science and Technology, 2022.

- [4] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 91-110.
- [5] 席永明. 车底安全视觉检测系统中的图像拼接技术研究[D]. 长沙: 湖南大学, 2012.
Xi Y M. Research on image mosaic technology in vehicle bottom safety visual inspection system[D]. Changsha: Hunan University, 2012.
- [6] 邢昆. 基于图像处理的车辆检查系统研究[D]. 廊坊: 北华航天工业学院, 2018.
Xing K. Research on vehicle inspection system based on image processing[D]. Langfang: North China Institute of Aerospace Engineering, 2018.
- [7] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF) [J]. *Computer Vision and Image Understanding*, 2008, 110(3): 346-359.
- [8] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. *Communications of the ACM*, 1981, 24(6): 381-395.
- [9] 李振宇, 田源, 陈方杰, 等. 基于改进 ORB 和 PROSAC 的无人机航拍图像拼接算法[J]. *激光与光电子学进展*, 2019, 56(23): 231003.
Li Z Y, Tian Y, Chen F J, et al. Aerial image stitching algorithm for unmanned aerial vehicles based on improved ORB and PROSAC[J]. *Laser & Optoelectronics Progress*, 2019, 56(23): 231003.
- [10] Rosten E, Drummond T. Machine learning for high-speed corner detection[M]//Leonardis A, Bischof H, Pinz A. *Computer vision-ECCV 2006. Lecture notes in computer science*. Heidelberg: Springer, 2006, 3951: 430-443.
- [11] Leutenegger S, Chli M, Siegwart R Y. BRISK: binary robust invariant scalable keypoints[C]//2011 International Conference on Computer Vision, November 6-13, 2011, Barcelona, Spain. New York: IEEE Press, 2012: 2548-2555.
- [12] Chum O, Matas J. Matching with PROSAC-progressive sample consensus[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), June 20-25, 2005, San Diego, CA, USA. New York: IEEE Press, 2005: 220-226.
- [13] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF[C]//2011 International Conference on Computer Vision, November 6-13, 2011, Barcelona, Spain. New York: IEEE Press, 2012: 2564-2571.
- [14] 许潇文, 黄玉清, 李磊民. 基于 AKAZE-ORB 的接触网图像拼接算法[J]. *制造业自动化*, 2020, 42(7): 97-101.
Xu X W, Huang Y Q, Li L M. OCS image mosaic algorithm based on AKAZE-ORB[J]. *Manufacturing Automation*, 2020, 42(7): 97-101.
- [15] 刘天赐, 宋延嵩, 李金旺, 等. 基于 ORB 特征的高分辨率图像拼接改进算法[J]. *激光与光电子学进展*, 2021, 58(8): 0810004.
Liu T C, Song Y S, Li J W, et al. Improved algorithm for high-resolution image stitching based on ORB features [J]. *Laser & Optoelectronics Progress*, 2021, 58(8): 0810004.
- [16] 王慎谦, 张荣国, 李晓波, 等. 基于 ORB 的无人机图像快速拼接算法[J]. *太原科技大学学报*, 2021, 42(5): 374-379.
Wang S Q, Zhang R G, Li X B, et al. Fast image mosaic of UAV based on ORB algorithm[J]. *Journal of Taiyuan University of Science and Technology*, 2021, 42(5): 374-379.
- [17] 杜港, 侯凌燕, 佟强, 等. 基于 BRISK 和改进 RANSAC 算法的图像拼接[J]. *液晶与显示*, 2022, 37(6): 758-767.
Du G, Hou L Y, Tong Q, et al. Image mosaicing based on BRISK and improved RANSAC algorithm[J]. *Chinese Journal of Liquid Crystals and Displays*, 2022, 37(6): 758-767.
- [18] 宋振凯. 基于 ARM 的交通路口图像拼接系统设计与实现[D]. 西安: 西京学院, 2022.
Song Z K. Design and implementation of traffic intersection image mosaic system based on ARM[D]. Xi'an: Xijing University, 2022.
- [19] 徐启文, 唐振民, 姚亚洲. 基于改进 SURF 算法的图像拼接研究[J]. *南京理工大学学报*, 2021, 45(2): 171-178.
Xu Q W, Tang Z M, Yao Y Z. Research on image mosaic based on improved SURF algorithm[J]. *Journal of Nanjing University of Science and Technology*, 2021, 45(2): 171-178.
- [20] 赵小强, 岳宗达. 一种面向图像拼接的快速匹配算法[J]. *南京理工大学学报*, 2016, 40(2): 165-171.
Zhao X Q, Yue Z D. Fast matching algorithm for image mosaic[J]. *Journal of Nanjing University of Science and Technology*, 2016, 40(2): 165-171.