

文章编号: 2095-2163(2020)02-0174-06

中图分类号: TP391.41

文献标志码: A

基于改进 BRISK 算法的图像特征提取方法研究

陈 婵, 管 启, 朱鸣镝

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘要: 以移动机器人视觉导航为应用背景,大量研究表明, BRISK 算法随着图像尺度不断增大, 错误匹配急剧增加。如果使用 BRISK 算法作为视觉里程计的特征匹配算法, 将无法为视觉 SLAM 后端提供准确的位置信息。针对这一问题, 本文提出了改进的 BRISK 算法, 在改进的算法中, 尺度空间金字塔通过构建 u_i 层来细化相邻图像间的尺度间隔, 并使用灰度质心法为关键点分配主方向, 取代了原算法利用长距离采样点对计算局部梯度的方法。实验结果表明, 该算法在运行时间相差不大的情况下, 尺度不变性上表现出较好的鲁棒性, 特征点匹配准确率有很大提高。

关键词: 视觉 SLAM; BRISK 算法; 尺度空间; 金字塔; 关键点主方向

Research on image feature extraction method based on improved BRISK algorithm

CHEN Chan, GUAN Qi, ZHU Mingdi

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

【Abstract】 With the mobile robot visual navigation as the application background, numerous studies show that, while the image scale continues to increase, the false matches of BRISK algorithm increase sharply. If the BRISK algorithm is used as the feature matching algorithm for visual odometers, it is impossible to provide accurate position information for the visual SLAM backend. In view of this problem, this paper proposes an improved BRISK algorithm. In the improved algorithm, the scale space pyramid refines the scale interval between adjacent images by constructing the u_i layer, and assigns the main direction to the key points by using the gray scale centroid method. It replaces the original algorithm to calculate the local gradient using long-distance sampling points to accelerate the algorithm. The experimental results show that the proposed algorithm exhibits better robustness in scale invariance with a slight increase in time consumption, and the accuracy of feature point matching is greatly improved.

【Key words】 visual SLAM; BRISK algorithm; scale space; pyramid; key point main direction

0 引言

视觉 SLAM (Visual Simultaneous Localization and Mapping)^[1-2] 是基于视觉的同时定位和建图系统, 该系统分为前端和后端, 前端也称为视觉里程计 (VO)。研究时是根据相邻帧图像的信息, 估计出粗略的相机运动, 给后端提供较好的初始值。基于特征点法的前端, 长久以来被认为是视觉里程计的主流方法。而且运行稳定, 对光照、动态物体不敏感, 是目前比较成熟的解决方案。

获取场景图像中的特征点信息通常包括特征点检测与特征点描述两个步骤。首先特征点检测算子检测出场景图像中的那些包含丰富结构信息的特征点。然后使用描述子对检测到的特征点进行描述。对特征点的描述主要是对特征点周围像素信息的描述, 最终将产生一个和特征点一一对应的描述符。

视觉 SLAM 中常用的特征点提取算法有

SIFT^[3]、SURF^[4]、ORB^[5]等, 由于 SIFT、SURF 算法的复杂度, 无法满足视觉 SLAM 系统对实时性的要求。实时特征提取算法 ORB 算法在 SLAM 中被广泛使用。由文献[6-8]得知, ORB 特有的二值描述符大大加速了特征提取与匹配算法的运行效率。类似的二值描述子特征提取算法还有 BRISK^[9]、FREAK^[10]等。

根据文献[8, 11-12]了解到, BRISK 算法在尺度不变性上表现较差, 如果使用 BRISK 算法作为视觉里程计中的特征点检测算法, 在尺度变化较大的情况下, 常常不能为后端提供准确的位置信息。本文提出的 BRISK 算法通过细化尺度空间的尺度间隔为 BRISK 建立新的尺度空间, 使得尺度尽可能接近理想的连续尺度, 并使用灰度质心法来代替原来使用长距离匹配点对为关键点分配主方向, 经过多次实验后发现改进的 BRISK 在匹配准确率上有很

作者简介: 陈 婵(1995-), 女, 硕士研究生, 主要研究方向: 基于双目相机的视觉 SLAM 图像匹配算法研究; 管 启(1989-), 男, 博士, 主要研究方向: 视觉 SLAM、图像处理; 朱鸣镝(1994-), 男, 硕士研究生, 主要研究方向: 基于单目相机的视觉 SLAM 图像匹配算法研究。

收稿日期: 2019-10-11

大提高,在尺度不变性上表现优异,同时具有很好的抗噪声干扰性能。

1 BRISK 检测与描述

1.1 尺度空间的构建

BRISK 算法中,尺度空间金字塔由 n 个普通层图像 *octaves* (用 c_i 表示) 和 n 个内层图像 *int tra - octaves* (用 d_i 表示) 组成,通常 $n = 4, i = \{1, 2, \dots, n - 1\}$ 。其中,原始图像 c_{i-1} 和 d_{i-1} 作为基准层图像,普通层图像 $c_i (i \neq 0)$ 均由 c_{i-1} 层不断 2 倍降采样生成,内层图像 $d_i (i \neq 0)$ 均由 d_{i-1} 层不断降采样生成,每个内层图像 d_i 都位于 c_i 和 c_{i+1} 之间,可以构建一个 $2n$ 层金字塔,如图 1 所示。其中内层图像 d_0 是由原图 c_0 的 1.5 倍降采样得到。如果使用 t 来代表尺度,则有: $t(c_i) = 2^i, t(d_i) = 2^i \cdot 1.5$ 。

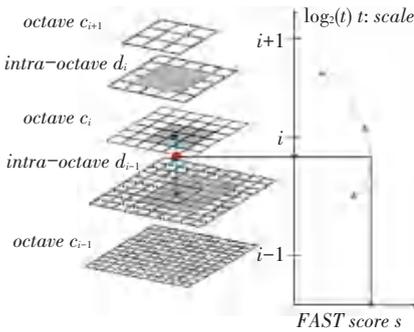


图 1 尺度空间特征点检测

Fig. 1 Scale spatial feature point detection

1.2 特征点检测与定位

(1) BRISK 算法在尺度空间中每个 *octaves* 和 *int tra - octaves* 上使用 AGAST9-16^[13] 算子进行关键点检测(额外增加的 d_{-1} 层使用的是 FAST5-8^[14] 算子),在此基础上对每个关键点计算 AGAST 得分。对检测出来的角点在上下两层使用非极大值抑制来筛选响应较强的点,这里使用 AGAST 得分作为每个角点的响应值。

(2) 对已经粗筛选出的特征点进行插值,这种插值是利用最小二乘法对 x, y 方向的二维二次插值,以得到准确的特征点坐标位置,并在尺度方向上进行抛物线插值,以得到极值点所在尺度,用这个尺度根据上下层极值点与关键点位置的偏移量再次对特征点位置进行二次修正,把这个位置作为特征点的位置。

1.3 构建描述子

BRISK 描述符由简单亮度比较测试构成的二进制字符串组成。这种方法在文献[15]中证明是非常高效的, BRISK 描述符的关键概念是对关键点的邻域进行采样的思想,如图 2 所示。在图 2 中,小蓝

色圆圈表示采样位置;红色虚线圆圈则以半径 σ 绘制,该半径 σ 正比于平滑采样点的高斯核的标准偏差。图案定义了在与关键点同心圆上等距离间隔的 N 个位置。把 N 个采样点两两组合成一对,将所有组合方式的集合称作采样点对集,用 A 表示,则有:

$$A = \{(P_i, P_j) \in R^2 \times R^2 \mid i \leq N, j < i\}, \quad (1)$$

其中, (P_i, P_j) 表示采样点对,将经过高斯滤波后的采样点 P_i, P_j 分别标记为 $I(P_i, \sigma_i)$ 与 $I(P_j, \sigma_j)$, 用 $g(P_i, P_j)$ 表示采样点对的局部梯度集合, 则有:

$$g(P_i, P_j) = (P_j - P_i) \cdot \frac{I(P_j, \sigma_j) - I(P_i, \sigma_i)}{\|P_j - P_i\|^2}, \quad (2)$$

根据采样点对间的距离,定义短距离点对子集 S 、长距离点对子集 L , 即:

$$S = \{(P_i, P_j) \in A \mid \|P_j - P_i\| < \delta_L\} \subseteq A, \quad (3)$$

$$L = \{(P_i, P_j) \in A \mid \|P_j - P_i\| > \delta_S\} \subseteq A, \quad (4)$$

其中, δ_S 为短距离阈值,通常取 $9.75t$; δ_L 为长距离阈值,通常取 $13.67t$; t 是特征点所在的尺度。

根据长距离子集计算特征点的方向,这里将用到如下数学公式:

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{l} \sum_{(P_i, P_j) \in L} g(P_i, P_j), \quad (5)$$

其中, g 为特征点主方向; l 是长距离点对子集 L 中所有采集点对间距之和; $g(P_i, P_j)$ 为 (P_i, P_j) 的梯度; g_x 与 g_y 分别为 L 中各采集点对在 x 方向和 y 方向梯度。在生成描述子之前,需要将模板沿中心顺时针旋转 θ 度,即:

$$\theta = \tan^{-1}(g_y, g_x), \quad (6)$$

按照公式(6)在短距离点对子集 S 中对比采样点对灰度值,最终生成二进制描述子。其对应公式可写为如下形式:

$$b = \begin{cases} 1, & I(P_j^\theta, \sigma_j) > I(P_i^\theta, \sigma_i); \\ 0, & \text{其他.} \end{cases} \quad (7)$$

2 改进的 BRISK 算法

2.1 细化尺度空间金字塔的构建

细化尺度空间金字塔由 n 个普通层图像 *octaves* (用 c_i 表示)、 n 个内层图像 *intra - octaves* (用 d_i 表示) 和 n 个上层图片 *up - octaves* (用 u_i 表示) 组成,通常 $n = 4, i = \{1, 2, \dots, n - 1\}$ 。其中,原始图像 c_0, d_0 和 u_0 作为基准层图像,普通层图像 $c_i (i \neq 0)$ 均由 c_{i-1} 层不断 2 倍降采样生成,内层图像 $d_i (i \neq 0)$ 均由 d_{i-1} 层不断降采样生成,上层图像 $u_i (i \neq 0)$

均由 u_{i-1} 层不断降采样生成,因此构建一个 $3n$ 层金字塔。其中,内层图像 d_0 是由原图 c_0 1.25 倍降采样得到,上层图像 u_0 是由原图 c_0 1.5 倍降采样得到。如果使用 t 来代表尺度,则有: $t(c_i) = 2^i, t(d_i) = 2^i \cdot 1.25, t(u_i) = 2^i \cdot 1.5$ 。

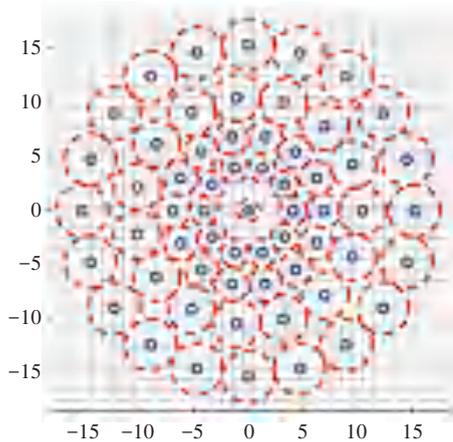


图2 采样点 $N=60$ 的 BRISK 采样模式

Fig. 2 BRISK sampling mode at sampling point $N=60$

本文引入细化的尺度空间,一方面原因是,在视觉信息处理模型中通常引入一个尺度的参数,就是要通过连续变化尺度参数获取不同尺度下的视觉信息,然后综合这些信息才能够深入挖掘图像的本质特征。在图像的特征匹配中,研究中同样也希望通过连续变化尺度参数获取不同尺度下的图像,获取连续尺度下检测到的特征点,特征匹配才能真正达到尺度不变性,本文对原算法尺度空间的 c_i 和 d_i 层之上增加了 u_i 层,更细化了尺度间隔,增加了更多尺度下的视觉信息;另一方面这样的改变会更加细化特征的选择,原因是在特征点粗定位时,会将中间层的特征点与上下两层的特征点使用非极大值抑制,增加了 u_i 层同时也增加了更多的中间层,拓宽了特征搜索范围,增大匹配的概率,有利于粗定位的改善;也有助于通过插值得到特征点精确位置,在精度上也有一定程度上的提升。从2个方面出发提高整个系统鲁棒性,为后续准确的匹配提供了良好基础。

2.2 灰度质心法确定关键点主方向

灰度质心法是假设特征点所在图像块的灰度质心 C 偏移了图像块的几何中心 O ,以向量 \vec{OC} 的方向作为特征点的方向。这里对各研发步骤可做阐释分述如下。

(1) 在一个小的图像块中,定义图像块的矩为:

$$m_{pq} = \sum_{x,y \in F} x^p y^q I(x,y), \quad (8)$$

(2) 通过矩可以找到图像块的质心,其对应公

式可表示为:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right), \quad (9)$$

(3) 连接图像块几何中心 O 与质心 C 得方向向量 \vec{OC} , 于是特征点的方向可以定义为:

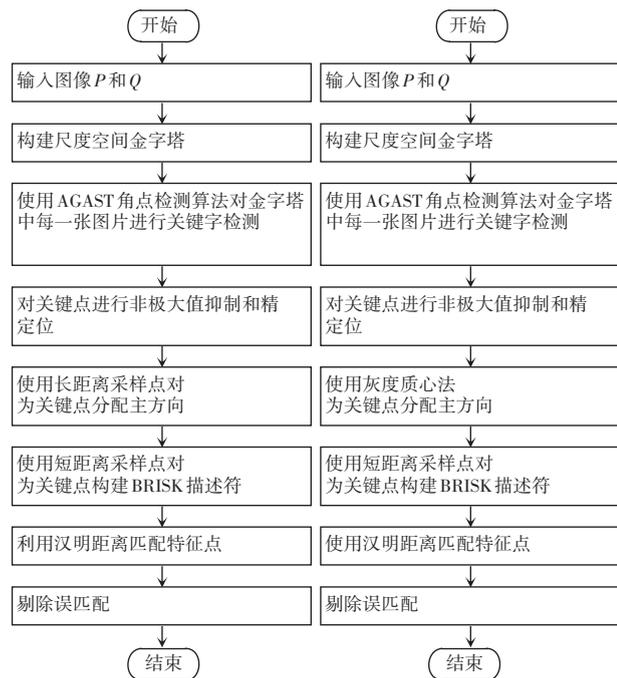
$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right). \quad (10)$$

这样就为每个关键点确定了主方向,使得 AGAST 角点具有了旋转不变性。

通过以上步骤,AGAST 角点就具有了尺度与旋转不变性,显著提高了其在不同图像中检测性能。

2.3 2种算法的流程图

综合前述研究可知,推得2种算法的设计流程如图3所示。



(a) 原算法流程图

(b) 改进算法流程图

(a) Flow chart of original algorithm (b) Flow chart of improved algorithm

图3 2种算法的流程图

Fig. 3 Flow charts of two algorithms

3 实验

本文的硬件环境为 Windows10, 64 位操作系统, 工作站型号: intel (R) Xeon (R) Silver4116, 实验平台为 VS2015+OpenCV3.4。本文采用牛津大学网站 (<http://www.robots.ox.ac.uk/~vgg/affine>) 的标准 Oxford 数据集作为测试图像^[16]。研究中使用不同的几何和光度变换以及不同的场景类型来评估真实图像上的描述符。本文拟将评估4种图像变换: 旋转+尺度变换 (boat/bark 数据集); 视点变换 (graff/

wall 数据集);图像模糊(bikes/tree 数据集);光照变换(leuven 数据集)。

3.1 实验效果

为了验证本文算法在提高匹配质量和匹配效果上的有效性,从数据集中随机选取一组旋转+尺度图片(boat 数据集),和使用手机拍摄实验室杂乱场景下的图片进行试验,将本文算法与 SURF、ORB、BRISK 进行比较。实验效果如图 4 所示,在实验室拍摄的图片中,对比图 4(a)和图 4(c)可以发现 SURF 与 ORB 算法几乎没有正确匹配,相比之下本文算法匹配效果最好。对比 boat 数据集,本文算法在成功匹配点个数与 BRISK 算法相差不大,实验验证了使用灰度质心法为关键点确定主方向的有效性。进而,研究求得算法运行时间的结果比较见表 1。

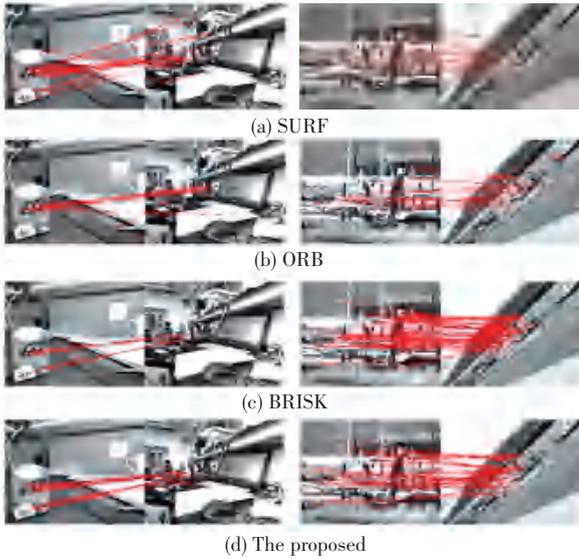


图 4 2 组图片在 4 种不同算法下的匹配效果

Fig. 4 Matching effect of two groups of pictures under four different algorithms

表 1 算法运行时间比较

Tab. 1 Comparisons of algorithm runtime

确定主方向的方式	关键点数目/个	每个关键点分配主方向的时间/us
使用灰度质心法	500	3.45
	1 000	2.75
	1 500	2.73
对称匹配对的局部梯度	500	5.60
	1 000	5.12
	1 500	4.83

3.2 鲁棒性测试

3.2.1 评估标准

本文使用类似于文献[17]中提出的标准。该标准提出了 3 种不同的指标:假定匹配率(Putative Match Ratio),精确度(Precision)和匹配得分

(Match Score)。对此研究内容可详述如下。

(1)假定匹配率。能够体现描述符的独特性,并描述检测到的特征的哪一部分最初将被识别为匹配(尽管可能不正确)。具体数学公式如下:

$$Putative\ Match\ Ratio = \frac{\#Putative\ Matches}{\#Features}, \quad (11)$$

其中, Putative Match 表示在所有的匹配对中,最近邻与次近邻比值小于 0.7 的匹配对数,且保证匹配是一对一的匹配; Features 的大小为 2 幅待匹配的图像中检测到关键点个数较少的那个。

(2)精度。定义了一组假定匹配中的正确匹配数(inlier ratio)。具体数学公式如下:

$$Precision = \frac{\#Correct\ Matches}{\#Putative\ Matches}, \quad (12)$$

在该等式中,正确匹配的数量是基于已知相机位置几何验证的假定匹配。文中使用牛津数据集中提供的 GroundTruth 数据。通过 2 幅图像间的单应性矩阵,将配对的关键点映射到一幅图像中,如果彼此的距离在 2.5 像素内,则该匹配对是正确的。

(3)匹配得分。是假定的匹配率和精度的乘积。具体数学公式如下:

$$Matching\ Score = \frac{\#Correct\ Matches}{\#Features}. \quad (13)$$

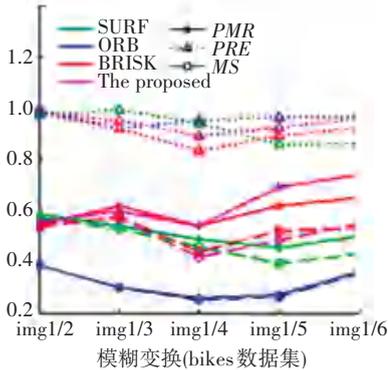
该指标描述了正确匹配的初始特征的数量,并且与前两个度量一样,匹配分数可能受到描述符独特性和匹配标准的影响。

3.2.2 实验结果分析

图 5 展示了本次研究对 4 种算法的描述子性能评估的结果。

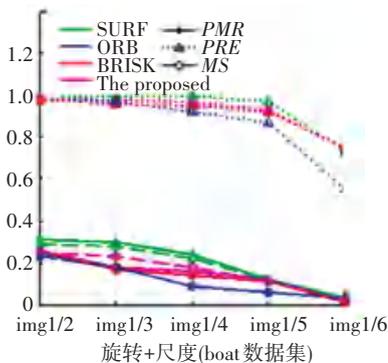
在图 5 中,横坐标为以每个场景数据集中的第一张图像与其余的 5 幅图像组成的匹配对,节点形状含义:‘- * -’代表 Putative Match Ratio 标注中简称为 PMR;‘...Δ...’ Precision 标注中简称为 PRE;‘- - -’代表 Match Score 标注中简称为 MS。经过观察图 5(a),研究发现在图像发生较大程度的模糊变化时,本文算法综合表现最好,说明构建细化的尺度空间可以使算法在尺度上具有更强的鲁棒性。经过观察图 5(b),分析发现在旋转和尺度变化的数据集中,4 种算法的假定匹配率和匹配得分普遍偏低。其中,SURF 算法综合表现最好,BRISK 算法与本文算法效果相当,说明 SURF 算法、BRISK 算法和本文算法的描述子具有较好的独特性。经过观察图 5(c),分析发现 4 种算法描述子对光照的鲁棒性相

差不多,对图像光照变化均有较强的鲁棒性。经过观察图 5(d),进一步发现在较大的视角变化时,ORB 和 SURF 算法在 $img1/3$ 到 $img1/5$ 之间精度出现急剧下降,说明这两种算法对较大视角变化没有较好的鲁棒性,综合来看,BRISK 算法表现最优,本文算法与 BRISK 算法表现相差不大,能够达到一定程度上的有效匹配。



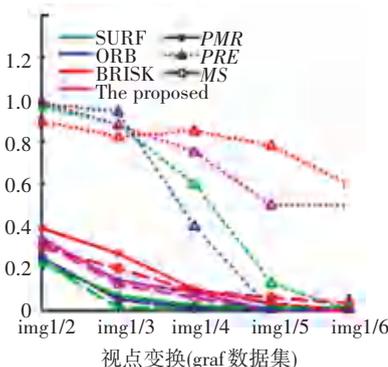
(a) 模糊变换数据集上不同匹配指标的结果测试

(a) The test results of different matching indices on ambiguity change datasets



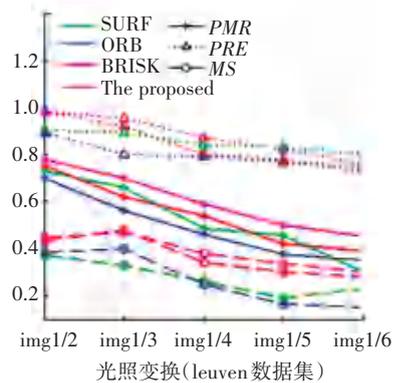
(b) 旋转+尺度数据集上不同匹配指标的结果测试

(b) The test results of different matching indices on rotation + scale change datasets



(c) 视点变换数据集上不同匹配指标的结果测试

(c) The test results of different matching indices on viewpoint transformation datasets



(d) 光照变换数据集上不同匹配指标的结果测试

(d) The test results of different matching indices on illumination transformation datasets

图 5 仿真测试结果

Fig. 5 Simulation test results

3.3 时间评估

算法匹配速度比较:在相同的匹配环境下,使用同样配置的计算机,对相同的一对图像进行比较,测试算法的执行时间。研究得到的同一对图像匹配 100 次的执行时间统计结果见表 2。结果显示本文算法执行时间略大于原算法执行时间。

表 2 算法运行时间比较

Tab. 2 Comparisons of running time of algorithms

	暴力匹配对数/个	特征点提取与匹配时间消耗/ms
BRISK 算法	500	8.0
	1 000	7.6
	1 500	5.5
改进的 BRISK 算法	500	8.3
	1 000	7.9
	1 500	5.8

4 结束语

本文提出了一种改进的 BRISK 算法,有效地解决了 BRISK 对尺度不变性表现欠佳的缺点,同时继承了 BRISK 的快速检测的特性。此外,使用灰度质心法为关键点分配主方向,有效地减少了描述子构建过程中的计算时间。实验结果表明,本文提出的方法大大提高了 BRISK 的特征点匹配性能。与使用 BRISK 方法的视觉里程计相比,使用本文方法能有效改善视觉里程计的位置估计和方向估计。此外,本文所提算法在特征点匹配精度和计算时间方面表现良好,可以实际应用于实时单目视觉里程计。

参考文献

- [1] LIU Haomin, ZHANG Guofeng, BAO Hujun. A survey of monocular simultaneous localization and mapping [J]. Journal of Computer-Aided Design and Computer Graphics, 2016, 28(6): 855.
- [2] DAVISON A J, REID I, MOLTON N, et al. MonoSLAM: Real-time single camera SLAM [J]. IEEE Transactions on Pattern

Analysis and Machine Intelligence, 2007,29(6): 1052.

[3] LOWE D G. Distinctive image features from scale - invariant keypoints[J].International Journal of Computer Vision,2004, 60 (2):91.

[4] BAY H, ESS A, TUYTELAARS T, et al. SURF: Speeded up robust features [J].Computer Vision and Image Understanding, 2008,110(3):346.

[5] RUBLEE E, RABAU D, KONOLIGE K, et al. ORB: An efficient alternative to SIFT or SURF [C]//International Conference on Computer Vision. Barcelona, Spain: IEEE Computer Society, 2011:2564.

[6] WANG X, HE X, NIE L, et al. Item silk road: Recommending items from information domains to social users[C]//SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM, 2017: 185.

[7] 成怡, 佟晓宇. 基于改进 ORB 算法的机器人视觉 SLAM 方法研究[J]. 电子技术应用, 2019,45(1):10.

[8] MOUATS T, AOUF N, NAM D, et al. Performance evaluation of feature detectors and descriptors beyond the visible[J]. Journal of Intelligent & Robotic Systems, 2018,92(3):33.

[9] LEUTENEGGER S, CHLI M, SIEGWART R Y. BRISK: Binary robust invariant scalable keypoints [C]// IEEE International Conference on Computer Vision, ICCV 2011. Barcelona, Spain:IEEE, 2011:2548.

[10] ALAHI A, ORTIZ R, VANDERGHEYNST P. FREAK: Fast retina keypoint[C]// 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, Rhode Island: IEEE, 2012:510.

[11] GUCLU O, CAN A B. A comparison of feature detectors and

descriptors in RGB - D SLAM methods [M]// KAMEL M, CAMPILHO A. Image Analysis and Recognition. ICIAR 2015. Lecture Notes in Computer Science. Cham: Springer, 2015, 9164:297.

[12] 刘晓璐. 典型二进制描述子研究及性能评估[D]. 西安:西安电子科技大学,2014.

[13] MAIR E, HAGER G D, BURSCHEKA D, et al. Adaptive and generic corner detection based on the accelerated segment test [M]// DANILIDIS K, MARAGOS P, PARAGIOS N. Computer Vision - ECCV 2010. ECCV 2010. Lecture Notes in Computer Science.Berlin/ Heidelberg: Springer,2010,6312: 183.

[14] ROSTEN E, DRUMMOND T. Machine learning for high - speed corner detection[C]//Proceedings of the 9th European Conference on Computer Vision. Graz, Austria:Springer-Verlag,2006,1:430.

[15] TOLA E, LEPETIT V, FUA P. Daisy: An efficient dense descriptor applied to wide baseline stereo[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2010,32 (5):815.

[16] CHONG C W, RAVEENDRAN P, MUKUNDAN R. A comparative analysis of algorithms for fast computation of Zernike moments[J]. Pattern Recognition, 2003, 36(3):731.

[17] HEINLY J, DUNN E, FRAHM J M. Comparative evaluation of binary features[C]//Proc. of European Conference on Computer Vision. Florence, Italy:Springer-Verlag,2012:759.

[18] MIKOLAJCZYK K, SCHMID C. An affine invariant interest point detector [M]//HEYDEN A, SPARR G, NIELSEN M, et al. Computer Vision - ECCV 2002. ECCV 2002. Lecture Notes in Computer Science. Berlin/ Heidelberg:Springer,2002,2350: 128.

[19] 石祥滨, 张劲松, 刘进立, 等. 一种适合于大尺寸航拍图像的特征点匹配方法[J]. 沈阳航空航天大学学报,2013,30(4):40.

(上接第 173 页)

人脸识别的驾驶人员疲劳检测系统对疲劳驾驶进行检测。在模拟平台中,模拟 100 组驾驶人员行驶情况,并且其中每一组的驾驶人员状态都不相同,分别利用 2 种方法对 100 组各进行一次检测,且在检测的过程中,保证每组进行两次行驶的状态保持一致。

3.2 实验结果分析

对照组与实验组分别对 100 组驾驶人员的行驶过程中的疲劳检测结果见表 1。

表 1 检测结果对比

Tab. 1 Comparison of test results

组别	正常行驶状态 人数/人	疲劳驾驶 状态人数/人	实际疲劳驾驶 状态人数/人	精度度/%
对照组	61	39	60	65
实验组	43	57	60	95

表 1 中,精度度的计算公式为:

$$\text{精度度} = \frac{\text{疲劳驾驶状态人数}}{\text{实际疲劳驾驶状态人数}} \times 100\% \quad (2)$$

由表 1 中可以看出,实验组检测结果的精度度

明显高于对照组,说明本文设计的检测系统可以更加精确地检测出驾驶人员的疲劳驾驶情况,具有非常高的实用价值。

4 结束语

本文基于人脸识别技术的优势,设计一套全新的驾驶人员疲劳检测系统,具有较高的社会价值。本文对驾驶人员的眼部状态进行精准的检测,在检测到疲劳驾驶时可以对驾驶人员进行及时的预警,同时还对人脸识别、人眼定位等方面实现了全新的设计。但在研究运算的过程中,由于能力有限,运算量较小,因此还存在一定的误差,在下一步工作中还将结合其它更加成熟的算法加以研究。

参考文献

[1] 刘金星,付阳,李可民. 基于 USB3.0 传输的人脸识别近红外相机系统[J]. 电声技术, 2018, 42(8):9.

[2] 唐阳山,徐忠帅,杨语尧. 基于面部表情的驾驶员疲劳特征提取[J]. 辽宁工业大学学报(自然科学版), 2018, 38(6):404.

[3] 马发民,吴红乐,张林,等. 基于 BP 神经网络的疲劳判定[J]. 计算机与数字工程, 2017, 45(6):1187.