DOI:10.19651/j.cnki.emt.2416875

基于改进 YOLOX 的动态视觉 SLAM 方法*

程 强[1,2] 张友兵[1,2] 周 奎[1,2]
(1.湖北汽车工业学院汽车工程师学院 十堰 442002; 2.湖北汽车工业学院 Sharing-X 重点联合实验室 十堰 442002)

摘 要:针对目前大多数传统的视觉 SLAM 系统通常默认环境是静态的,但实际环境中是存在移动目标或者障碍物的,所以往往包含大量的误匹配点和动态点会导致系统定位精度降低的问题。提出一种基于 ORB-SLAM3 主体框架和深度学习技术的语义 vSLAM 系统,结合目标检测与光流法,旨在改进视觉 SLAM 系统在动态环境中的定位精度。首先,利用改进的 YOLOX-S 目标检测算法对潜在的动态目标进行识别;然后,利用几何法与光流法相结合精确检测异常值,并根据物体和人类运动状态不断调整动态包围盒的阈值。最终,保留动态框中包含静态框中的点,同时消除动态框中的其他点。在 TUM 和 KITTI 数据集上进行精确性的评估,实验结果表明,在数据集高动态序列下,与 ORB-SLAM3,Crowd-SLAM 比较,绝对轨迹均方根误差分别平均减少 69.26%、16%,与 DynaSLAM 比较,在高动态场景中定位精度平均提升 15%,这验证了在动态场景中提升了系统定位精度,此外,真实场景测试结果显示,该系统在各种动态环境中均表现出良好的性能。

关键词:视觉 SLAM;ORB-SLAM3;光流法;动态环境;目标检测
中图分类号:TP391.41;TN98 **文献标识码**:A **国家标准学科分类代码**:510.1050

Dynamic visual SLAM method based on improved YOLOX

Cheng Qiang[1,2] Zhang Youbing[1,2] Zhou Kui[1,2]
(1. College of Automotive Engineering, Hubei Automotive Industry Institute, Shiyan 442002, China;
2. Sharing-X Key Joint Laboratory , College of Automotive Engineering, Shiyan 442002, China)

Abstract: Most traditional visual simultaneous localization and mapping (SLAM) systems typically assume a static environment; however, real-world environments often contain moving objects and obstacles, leading to a significant number of mismatched and dynamic points which can degrade localization accuracy. This paper proposes a semantic vSLAM system based on the ORB-SLAM3 framework and deep learning techniques, integrating object detection and optical flow methods to improve localization accuracy in dynamic environments. Firstly, an enhanced YOLOX-S object detection algorithm is utilized to identify potential dynamic targets. Subsequently, a combination of geometric and optical flow methods is employed to precisely detect outliers, with continuous adjustments to dynamic bounding box thresholds based on the motion states of objects and humans. Ultimately, points within static bounding boxes retained in dynamic frames are preserved, while others within dynamic frames are eliminated. The system′s accuracy is evaluated using the TUM and KITTI datasets. Experimental results demonstrate that under highly dynamic sequences, the proposed system achieves an average reduction of 69.26% and 16% in root mean square error of absolute trajectories compared to ORB-SLAM3 and Crowd-SLAM, respectively, and a 15% average improvement in localization accuracy in dynamic scenes when compared to DynaSLAM, thereby validating the enhanced system performance in dynamic environments. Moreover, the results of real-world scene tests demonstrate that the system performs well in various complex environments.

Keywords: visual SLAM;ORB-SLAM3;optical flow;dynamic environment;object detection

0 引 言

视觉同时定位与地图构建(visual simultaneous localization and mapping,VSLAM),是一种用于无人驾驶车辆、机器人等在未知环境中能够实现当前环境精确定位的基人技术。它利用携带的传感器对环境进行感知,并能

收稿日期:2024-09-12
*基金项目:湖北省科技重大专项(2020AAA001)、湖北省重点研发计划项目(2021BED004)、湖北省武汉市科技重大专项(2022013702025184)资助

过对周围环境进行重建。在视觉 SLAM 系统中，常用的传感器包括相机、激光传感器和惯性传感器。相机在视觉 SLAM 系统中被广泛应用，因为它可以采集到丰富的色彩纹理信息，并且价格实惠、占用空间小和能功耗低[1]。视觉 SLAM 系统利用相机作为主要传感器[2]。大多数方法适用于所有的相机种类，然而，如果该方法受到深度信息的限制，只能使用相对昂贵的 RGB-D 相机。并且，大多数 SLAM 系统都受到静态环境假设的限制，并受到现实世界中动态对象的干扰，导致许多不良或不稳定的数据关联。

基于特征去做图像配准是目前的主流匹配方法。最早由 Moravec 等[3]提出了特征点的概念和基于角点的检测算法，但该算法在图像经过旋转和尺度变化后无法实现图像配准。随后，Sun 等[4]提出并构建了图像尺度空间，主要使用差分运算和自相关矩阵进行角点检测，该方法使图像在不同尺度下具有更好的匹配效果(scale invariantfeaturetransform，SIFT)由 Lowe[5]提出，由于其 128 维的特征描述向量，它具有很强的鲁棒性和尺度不变性，但也因此在数据处理中消耗了大量时间。

对于动态对象干扰的问题，学者们通常会采用只跟踪稳定的静态特征点的思想。当然有些使用基于几何的方法，比如 RANSAC[6](随机样本一致)，用来消除静态或动态场景中的不匹配。虽然这个方法可以去除离群值，但当动态对象占据大部分视图时，它是没有效果的，有些会使用基于直接法，LSD-SLAM[7]是一种利用图像灰度值的 SLAM 系统，它基于直接法，即基于同一像素点的灰度值在一定的时间间隔在不同图像中保持固定基本不变的假设，类似于光流法。由于环境以及相机曝光机制的影响，往往难以满足这一假设，因此，直接法所基于的 SLAM 系统在处理动态环境时的稳定性受到影响。同时传统的特征点方法容易受到环境中运动物体的干扰，导致特征点数量不足和特征点错误匹配的问题。

近年来，越来越多的研究学者开始将深度学习技术与传统的几何方法相结合，以有效处理动态环境中的特殊对象。他们一般利用的主要思想是利用目标检测或者语义分割，能够得到被框图或者蒙版所包围的动态对象，并能通过几何的方法去除动态的特征点。文献[8]提出了 Crowd-SLAM 表明，对于语义分割的方法来说基于对象检测的方法更具有能够处理没有特殊定义的运动对象。刘丰宇等[9]提出了基于 ORB-SLAM2 框架上添加了潜在动态目标检测线程和基于 ORB-LK 光流金字塔的特征点速度计算线程，Chen 等[10]提出了一种方法，将目标检测和传统单目 SLAM 相结合，以应对动态物体对 SLAM 系统的干扰。该方法利用目标检测网络来确定场景中的动态物体，并通过先验知识和目标检测框的信息来剔除这些动态物体所在区域的特征点，但是将没有足够的数据关联用于姿态估计，SLAM 最后将失败。崔岸等[11]提出一种基于语义信息的动态特征点去除方法，利用实例分割结果获取语义特征点，使用自适应阈值的极线约束方法判定潜在动态特征点的状态，谢波等[12]提出一种单目视觉惯导里程计优化方法，通过 IMU 预积分信息与特征点深度增强方法对系统进行重定位，提高了系统的跟踪稳定性。SegNet[13]或 Mask-RCNN[14]等语义分割算法很难在分割精度、系统负载和检测类的数量方面取得良好的平衡。换句话说，当系统足够精确并且能够检测到许多类的对象时，系统就不能实时运行。

针对现有视觉 SLAM 系统在动态环境下普遍存在精度较低的问题，本文提出了一种实时语义 vSLAM 系统。本文 SLAM 采用 ORB-SLAM3 作为基础架构，通过融合轻量级 YOLOX[15]网络，减小模型的大小。并利用 TensorRT 优化后的 YOLOX 提取环境中 70 种不同对象类的语义信息，用来检测不同环境下的对象。使用目标检测网络得到潜在的动态网络后，使用几何法与 Lucas-Kanade 光流法相结合的方法处理，通过判断离群点并结合一种能够自动调节阈值的算法，此算法用来检测除人类以外的其他类，当盒子内的离群点超过阈值时，盒子将被判定为移动，该阈值将根据不同的对象自动调整。这是一种几乎不受噪声点影响的算法，它可以对潜在的物体进行运动的检查。最后，为了使被检测的类有利于系统的准确性，保留动态框内外静态框的特征点，所以从系统获得的动态框内通过剔除动态框内的其他点这个方法，来提升 SLAM 系统在动态环境中的定位精度。

1　算法框架

1.1　ORB-SLAM3 系统

ORB-SLAM3 是一种基于特征的紧密集成视觉惯性 SLAM 系统，它通过最大后验概率估计来实现定位和地图构建这意味着在整个 SLAM 系统中，包括 IMU[16]初始化阶段，在计算位置和姿态估计时都使用了最大后验概率估计方法，以提高位置和姿态的估计精度。ORB-SLAM3 是一个多地图系统，它是通过引入新的位置识别方法和改进的召回机制来实现。这些改进使得 ORB-SLAM3 能够在缺乏视觉信息的环境下长时间运行。它具备丢失保存地图时启动新地图的功能，并能够将新地图与之前建立的地图区域无缝合并。这种多地图系统有助于提升系统的健壮性和可靠性。

1.2　改进的 ORB-SLAM3 系统

为了提升 SLAM 系统在动态环境下的定位精度，本文在 ORB-SLAM3 基础上引入了一些额外的模块，包括目标检测语义模块、光流几何模块、运动框判定模块和动态点剔除模块如图 1 所示，保留动态框中包含的静态框中的点，同时消除动态框中的其他点。

2　目标检测语义模块

在视觉 SLAM 中，语义信息是通过分析图像中特征点所属的不同物体类别的概率来获取的，通常通过计算机视

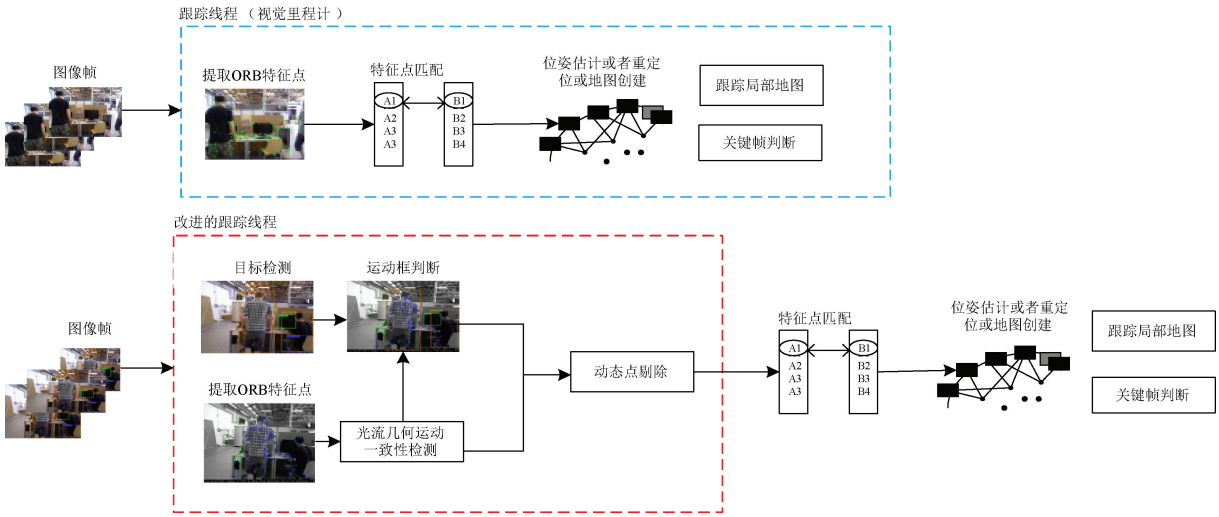


图 1　传统的跟踪线程与改进的跟踪线程

Fig. 1　Traditional tracking thread and improved tracking thread

觉技术进行检测和识别,在本文中,研究者采用了基于深度学习的方法来预测图像中不同对象的边界框,并从中识别出潜在的物体。通过引入这些语义信息,提高 SLAM 系统对于环境中物体的理解和感知能力,从而增强系统的稳定性。这些创新性的方法为进一步改进视觉 SLAM 技术提供了新的思路和方向,如图 2 所示。

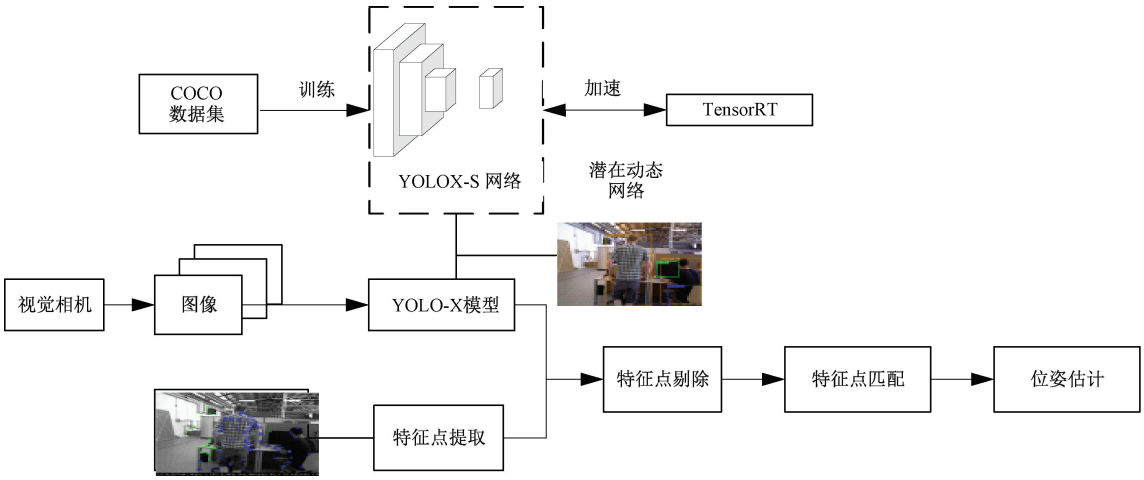


图 2　目标检测模块

Fig. 2　Object detection module

改进 YOLOX 目标检测网络:

本文采用 YOLOX-S 目标检测接口可以实现对图像或视频中的目标物体进行快速准确的检测和跟踪,它在性能上优于 YOLO 系列的最新版本。这意味着 YOLOX 在保持高计算速度的同时,具有极佳的精度。还使用了 COCO 数据集,COCO 数据集中样本量为 330 000 张图像,70 个物体类别,对其进行了训练来识别更多的类别和更精确的识别物体,此外,为了减少处理时间,本文使用 GPU 加速方法 TensorRT[17] 对 YOLOX-S 模型的网络进行优化。TensorRT 的加速原理包括网络层和张量的融合、低精度计算和精度校准、多流执行以及动态张量内存等技术,通过这些方式生成高效的推理引擎。最后形成一个优化的 engine,本文把处理好的优化引擎来进行一个预测的工作。

YOLOX-S[18] 它包括主干特征提取网络、加强特征提取网络和分类回归层。YOLOX-s 采用了 CSPDarknet 作为主干网络,并借鉴了 YOLOv5 中的 Focus 网络结构。CSPDarknet 在图像处理过程中,通过每隔一个像素采样一个值,并堆叠特征层的方式将宽高信息集中到通道信息中,并展输入通道数 4 倍。此外,CSPDarknet 还使用了 SPP[19] 结构进行最大池化操作,从而提高了网络对感受视野的把握能力。YOLOX-S 选择了 SiLU 激活函数,YOLOX-S 的目标检测损失函数由 Reg 模块、Obj 模块和 Cls 模块组成,分别用于判断特征点的回归参数、是否包含物体以及所属的物体类别。YOLOX-S 应用了 SimOTA[20]

技术来动态匹配正样本，以适应不同尺寸的目标，进一步提升了目标检测的性能和准确率，本文在主干网络中加入空间金字塔池化（SPP），在颈部网络引入 AFF 模块，增强颈部网络特征融合能力如图 3 所示。

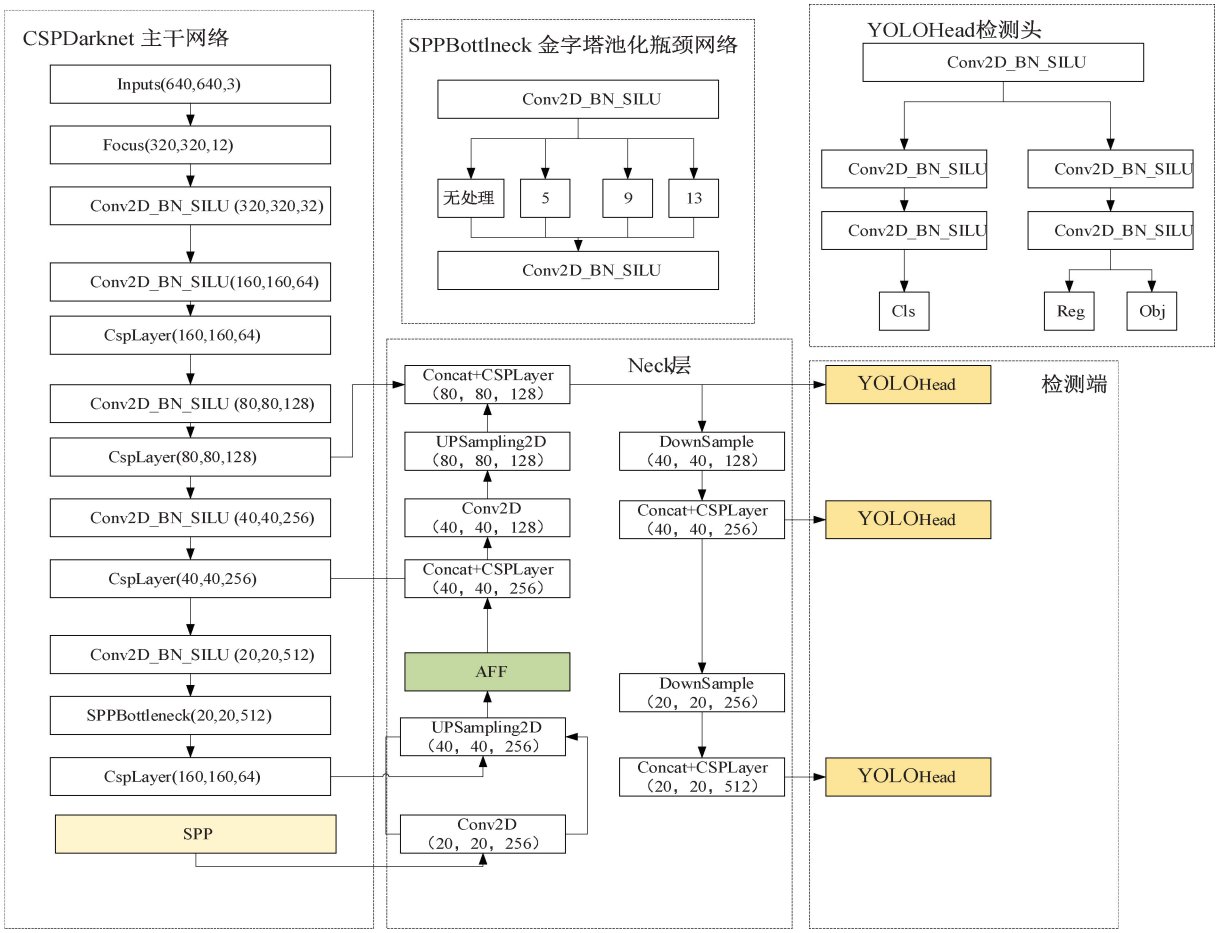


图 3　改进 YOLOX 目标检测网络

Fig. 3　Improved YOLOX target detection network

3　ORB 特征点与光流几何模块

3.1　ORB 特征点提取

点的特征提取的方法很多，常用的特征点有 SIFT、SURF、ORB 等，有些特征点，具有一定的旋转不变性和在光照条件下具有一些相似的表达，但实用的 SLAM 系统主要考虑实时性问题。在同一幅图像中同时提取 1 000 个特征点，可以计算 SIFT 算法的处理时间约为 5 228.7 ms，SURF 算法约为 217.3 ms，而 ORB 算法仅约为 15.3 ms。可以明显看出在速度方面，ORB 算法有显著的改进，这一显著的速度差异使得 ORB 算法在实时性要求较高的 SLAM 系统中成为优选方案。

考虑到 SLAM 系统对实时运行的需求，选择使用 ORB 特征点是出于多方面的考虑。首先，从速度角度来看，ORB 算法能够快速提取和匹配图像中的特征，满足实时处理的要求。其次，ORB 特征点在不同尺度的图像中表现出良好的适应性，能够在多样化的场景中稳定地检测到相应的特征点能够有效提升系统的实时性和鲁棒性。

3.2　光流法

在计算机视觉中，可以通过光流来描述图像中像素的运动状态。光流分为稀疏光流和稠密光流。稀疏光流是指描述图像中部分像素运动状态的方法，而稠密光流则能够描述所有像素的运动状态。其中，Hom-Schunck 光流是一种代表性的稠密光流方法，而 Lucas-Kanade（LK）[21] 光流则是主要的稀疏光流方法。在 SLAM 系统中，为了降低计算成本，通常会采用稀疏光流方法，只计算 SLAM 系统中提取的 ORB 特征点的光流场。因此，在 SLAM 系统中使用了 LK 光流。

LK 光流如图 4 所示。基于假设：图像中的像素亮度在连续帧之间不会发生明显变化，即帧间的间隔相对较短。此外，LK 光流还假设相邻像素具有相似的运动。通过将相机采集到的图像视为关于时间的函数，可以将图像中坐标 (x,y) 处的像素点在时间 t 时刻的灰度值表示为 $I(x,y),t)$ 总结而言，LK 光流是一种稀疏光流方法，用于

计算图像中提取的 ORB 特征点的光流场。它基于像素亮度不变、帧间间隔短以及相邻像素具有相似运动的假设。图像中 t 时刻的位置代表为 $I(x,y)$ 的像素点,在 $t+1$ 时刻它在图像中 $I(x+\mathrm{d}x,y+\mathrm{d}y)$ 的位置将会变成,当然在这段时间内,这两个位置的灰度值应该是相同的。

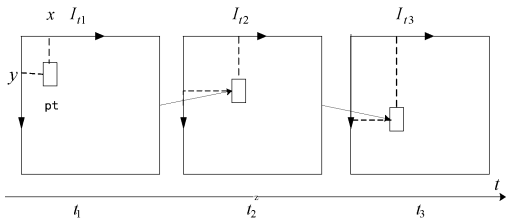


图 4　LK 光流

Fig. 4　LK optical flow translation

由于灰度不变,则有:

$$I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t)=I(x,y,t) \tag{1}$$

对左边进行泰勒展开,保留一阶项,得到:

$$I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t)\approx I(x,y,t)+\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial t}\mathrm{d}t \tag{2}$$

因为光流法有一个灰度不变假设的前提,于是上个时刻等于下个时刻的灰度值,于是:

$$\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial t}\mathrm{d}t=0 \tag{3}$$

两边同时除以 $\mathrm{d}t$,可以得到:

$$\frac{\partial I}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial I}{\partial y}\frac{\partial y}{\partial t}=-\frac{\partial I}{\partial t} \tag{4}$$

式中:$\frac{\partial x}{\partial t}$ 为特征点在 X 轴上的运动速度,而 $\frac{\partial y}{\partial t}$ 为在 Y 轴上的速度,分别记为 u、v。$\frac{\partial I}{\partial x}$ 为图像在该点处 X 轴方向的梯度,$\frac{\partial I}{\partial y}$ 则是在 Y 轴方向的梯度,分别记为 I_x,I_y。把特征点灰度对时间的变化量记为 I_t,将其写成矩阵形式,如式(5)所示。

$$\begin{bmatrix}I_x & I_y\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}=-I_t \tag{5}$$

仅凭借式(5)无法推导出特征点的速度 u、v,必须引入额外的约束。设置它有 n 个数量的像素点因为这个方程组是个超定方程,也就是说方程组内有冗余,可以通过最小二乘法求出最优解,最优解尽可能的同时满足 n 个方程。最后可以得出每个特征点的光流信息(u、v)将会被记录到特征点信息的结构体中,如式(6)所示,本文定义了一个特征点的数据结构,确保光流法所需的输入格式与 ORB 特征点的格式一致,并在光流计算后更新特征点的坐标,从而实现格式的统一。

$$\begin{cases}I_{1x}u+I_{1y}v+I_{1t}=0\\I_{2x}u+I_{2y}v+I_{2t}=0\\\cdots\\I_{nx}u+I_{ny}v+I_{nt}=0\end{cases} \tag{6}$$

3.3　光流金字塔

如果目标剧烈运动,假设就不成立,这会使得最终求出的光流值存在较大误差。为了缓解这种情况,可以在计算中缩小图像尺寸,从而缩小像素点移动,这样就可以使得 LK 光流法应用于目标剧烈运动的场景如图 5 所示,然后通过光流金字塔,意味着光流估计将在不同的尺度下进行,并且对输入图像进行了金字塔处理。在如果匹配点与像素边缘之间的距离非常小,或者匹配对中心的像素块差异很大,则会将其丢弃。

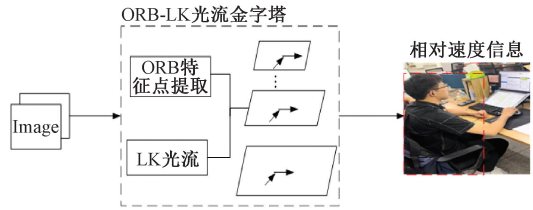


图 5　光流金字塔

Fig. 5　Optical flow pyramid translation

4　运动框判断模块

首先,利用(random sample consensus,RANSAC)算法,从数据集中随机选择一小部分样本作为内点集(inliers),并利用这些内点估计出一个模型。然后,利用该模型计算其他数据点与模型之间的误差,并将误差小于某个阈值的数据点加入到内点集中。不断迭代这个过程,直到找到一个满足条件的内点数量最多的模型。通过这个算法来寻找基础矩阵,以计算当前帧的极线。基础矩阵的作用是将上一帧中的点映射到它们在当前帧中对应的搜索域,即对极线的映射。令 p_1 和 p_2 分别表示上一帧和当前帧中成功匹配的点,并且 P_1 和 P_2 表示它们的齐次坐标形式:

$$\begin{aligned}&P_1=\lfloor u_1,v_1,1\rfloor,P_2=\lfloor u_2,v_2,1\rfloor,\\&p_1=[u_1,v_1],p_2=[u_2,v_2]\end{aligned} \tag{7}$$

其中,u、v 为图像的像素坐标,L_1 为极线,其计算方法如下:

$$L_1=\begin{vmatrix}X\\Y\\Z\end{vmatrix}=\boldsymbol{F}P_1=\boldsymbol{F}\begin{vmatrix}u_1\\v_1\\1\end{vmatrix} \tag{8}$$

$\boldsymbol{F}$ 表示基本矩阵。计算的方法是匹配点及其对应的极线如下:

$$D=\frac{|P_2^TFP_1|}{\sqrt{\|X^2\|+\|Y^2\|}} \tag{9}$$

其中,D 为距离。如果 D 的值超过了设定的阈值,特征点就被判定为离群值。

然后分别确定人与物的阈值。这是因为人和物体以不同的方式运动。准确地说，人们经常移动当地的关节。也就是说，只有一两个点是动态的。因此，当在人员框中签出一个离群值时，该框将被标记为动态。其他物体并不总是像人类一样移动一小部分。因此，需要考虑更多的特征点来判断。另一方面，有的特征点较少，有的特征点较多。在这种情况下，如果设置了固定的阈值，它将不适用于所有对象。

所以本项研究提出了一种创新的动态边界框判断算法，通过综合考虑宽高比例和光流跟踪结果，能够准确区分出动态边界框和静态边界框。算法首先将各类框内的离群值与相应框内光流跟踪的特征点总数进行对比，若得到的计算结果超过预先设定的阈值，则将该边界框标记为动态。

这一算法不仅具有对多种目标类型具有通用性的优势，而且有效解决了目标被遮挡或摄像机移动一定程度时误判目标为动态的问题。为降低假阴性率对 SLAM 系统的影响，研究设定了一个相对较低的阈值。对于移动的人物框图盒子，设定了一个离群值作为阈值；而对于动态对象的判断，则将阈值设定为盒内离群值的 40%。

此外，本文设计的系统允许根据具体情况手动调整阈值，以便根据不同场景灵活应对。该算法的具体流程如算法 1 所示。

算法 1　运动框判断算法

Algorithm 1　Motion box detection algorithm

输入 人物的边界框集合 $person_n$，其他物体的边界框集合 $object_n$；

输出 动态边界框集合 DB_n，静态边界框集合 SB_n；

```
function procedure judging(set)
  o←set //离群值的总数
  if o>0 then
    set→DB_n;
  else
  set→SB_n;
  end if
  end function
for each bounding box person_n of person do
    if W(i)/H(i)<3 then//将边界框分为上部 W(i)和下部 H(i)，分割比例为 0.3 和 1.3
    Judging(W(i));
    Judging(H(i));
    else
    Judging(person_n);
    end if
  end for
  for each bounding box of object_n do
  (s,o);//光流跟踪在动态中追踪到的关键点和离群点的总数
  If o/s>40% then
    object_n → DB_n;
  else
    object_n → SB_n;
  end if
 end for
```

5　动态点剔除模块

当被静态框，动态框标记后，由于框的界定，可以删除静态框外的点和动态框内的点。本文定义的是 bool=false 的关键点会添加到最终的有效的关键点和描述子的集合中，保留了静态盒中属于人盒或与人盒相交的关键点。这意味着静态盒越多，静态点越稳定，也就是说，系统的精度越高。动态点剔除算法如算法 2 所示。

算法 2　动态点剔除算法

Algorithm 2　Dynamic point clipping algorithm

输入 动态边界框集合 DB_n，静态边界框集合 SB_n；当前帧的关键点集合 K_n；

输出 动态边界框集合 DB_n，静态边界框集合 SB_n，特征点 (u_n,v_n) 被判断为静态的当前帧的关键点集合 S_n；

```
  bool_n ← false;
for each key point (u_n,v_n) in K_n of person do
  for each bounding box DB_n of person do
    if (u_n,v_n) in DB_n then//在动态框集合中
      bool_n ← ture;
  end if
end for
for each bounding box BO_n of other objects do
  if (u_n,v_n) in BO_n then//在物体框集合内
    bool_n ← false;
    end if
end for
  if bool_n = false then
    S_n ← (u_n,v_n); //添加到最终的有效的关键点和描述子的集合中
  end if
end
```

6　实验及分析

6.1　实验数据集

利用双目相机数据集，RGB-D 数据集和真实环境的单目相机来评估本文的系统。

首先使用 TUM 数据集进行目标检测网络的实验，由

于 TUM RGB-D 数据集包含了 RGB-D 相机在动态环境下捕获的许多图像序列,以及精确的地面真实轨迹和相机参数。动态 SLAM 主要是为了改进高动态场景下的 SLAM 系统,该数据集包含高动态场景和低动态场景两种情况。在高动态场景中,人物会持续行走,而在低动态场景中,人物坐在椅子上并没有明显的运动。数据集的名称中的 static、xyz、halfsphere、rpy 表示了 4 种不同的相机状态:static 表示相机处于基本静止状态。没有明显的移动,xyz 表示相机沿着 X、Y、Z 轴进行移动 halfsphere 表示相机在直径为 1 m 的半球表面上进行移动,rpy 表示相机在翻滚、俯仰和偏航轴上进行旋转。所以被视觉 SLAM 研究人员作为其中一个标准数据集之一。并常用于评价 SLAM 系统的表现。

KITTI 数据集为自动驾驶相关的各种任务提供精确的地面真实轨迹、相机参数和对象标注。在这个数据集中,有 22 幅立体图像序列被专门设计用来评估 vSLAM 算法的性能。这些子集由一辆行驶中的车辆记录,总行驶距离为 39.2 km,每个序列描述了不同的驾驶情况。本文使用的序列在 00-06 且整合了最新的视觉里程计(visual odometry,VO)和视觉惯性里程计(visual-inertial odometry,VIO)系统,并将它们与本文现在的工作进行比较。它不仅提供了地面真实信息。通过分析这些序列,可以综合评估 vSLAM 系统在各种真实世界驾驶场景中的性能。

最后,运行时分析证明了系统具有良好的精确性。同时,将单目相机模式系统部署在现实环境中,以展示其合理的效果。

6.2　实验环境及设置

使用一台配备了 Intel(R) Xeon(R) CPU E5-2686 v4 CPU,NVIDIA GeForce RTX 3080 GPU 和 32 GB 内存的服务器上运行,在 Ubuntu20.04 上进行实验,使用 YOLOX-s 模型进行实验,在准确性和实时性之间达到了良好的平衡。然后,使用摄像头(HBV-1780-2)在真实环境中对本文的系统进行测试。采用 CUDA12.1,SLAM 与目标检测部分采用 python,C++编写。

6.3　融合 YOLOX-s 视觉 SLAM 前端特征分析

图 6 显示了前端特征提取,第 1 行表示被提及的语义模块检测到的边界框。第 2 行结果来自检测到的边界框来检测动态点。第 3 行的结果与第 2 行相同,但多了动态的显示框对其中运动进行判断。动态点被标记为蓝色,静态点被标记为绿色。

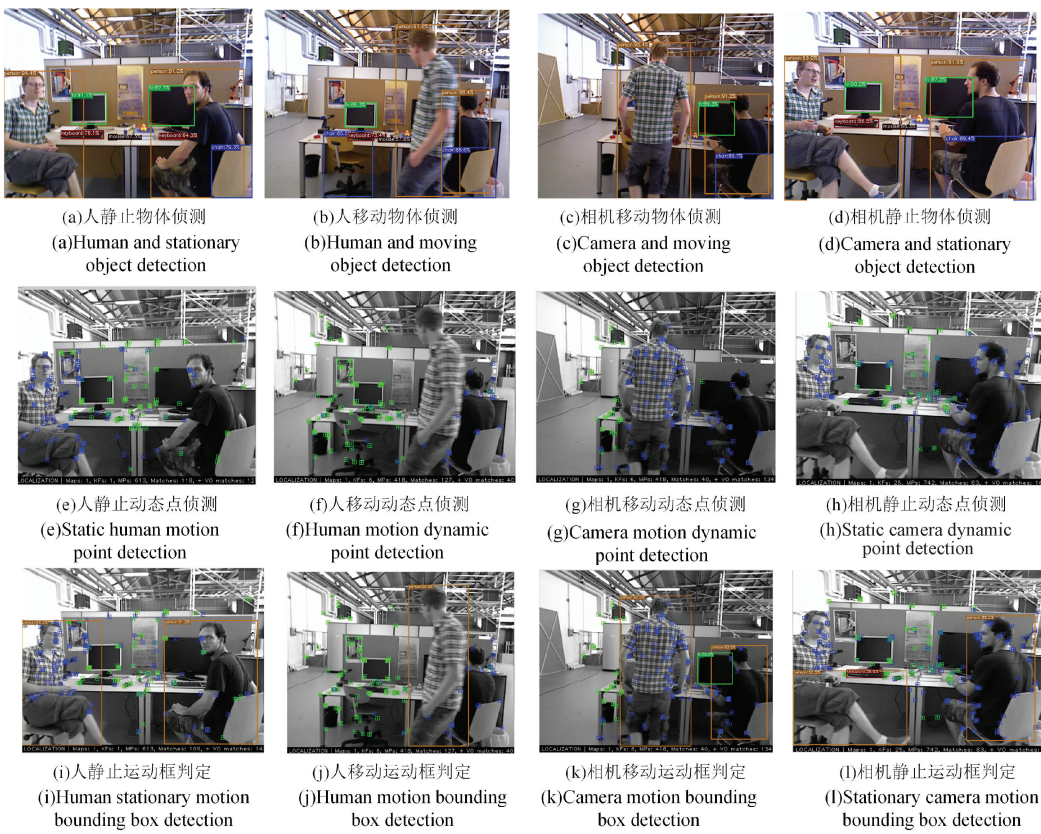


图 6　视觉 SLAM 前端特征提取

Fig. 6　Visual SLAM front-end feature extraction

6.4　改进的 YOLOX-S 消融实验分析

为验证本文所提模块的有效性,现添加空间金字塔池化 SPP,在引入 AFF 模块,进行实验,实验结果如表 1,其中 Baseline 是 YOLOX-S 算法。

表 1　消融实验

Table 1　Ablation study

序号	SPP	AFF	mAP/%	模型参数量/M	模型的大小/MB
Baseline			70.5	9.1	35.0
1	√		72.3	9.5	36.0
2		√	73.1	9.3	35.5
3	√	√	74.0	9.7	37.0

6.5　评价指标

在实验中，本文利用绝对轨迹误差(ATE)和相对位姿误差(RPE)对实验结果进行了测量。

绝对轨迹误差(ATE)是指实时估计位姿和真实位姿之间的差异。该指标可以直观地反映算法的定位精度和实时估计轨迹的全局一致性。对于视觉 SLAM 系统而言，保持实时估计轨迹的全局一致性是十分关键的。对于双目 SLAM 和 RGB-D SLAM 等实时位姿估计方法，估计得到的位姿通常不在与真实轨迹相同的坐标系中。为了评估估计位姿的准确性，可以使用最小二乘法计算从估计位姿到真实位姿的转换矩阵 S。该转换矩阵属于特殊欧氏群，通过计算估计位姿与真实位姿之间的差异。假设第 i 帧的相机估计位姿为 P_i，真实位姿为 Q_i，参考文献[22]的做法第 i 帧的 ATE 数值 F_i 定义为：

$$F_i = Q_i^{-1} S P_i \tag{10}$$

相对位姿误差(RPE)是指两个时间间隔为固定差值 t 的帧之间的实时估计位姿和真实位姿之间的差异。它通过将时间戳对齐后的实时估计位姿与真实位姿之间的变化量进行计算，并对所得到的变化量进行差值处理来表示。通常用 RPE 来表示，假设第 i 帧的相机估计位姿为 P_i，真实位姿为 Q_i，与第 i 帧相隔时间 t 的相机估计位姿为 P_{i+1}，真实位姿为 Q_{i+1}，因此第 i 帧的 RPE 数值 E_i 可以计算得出：

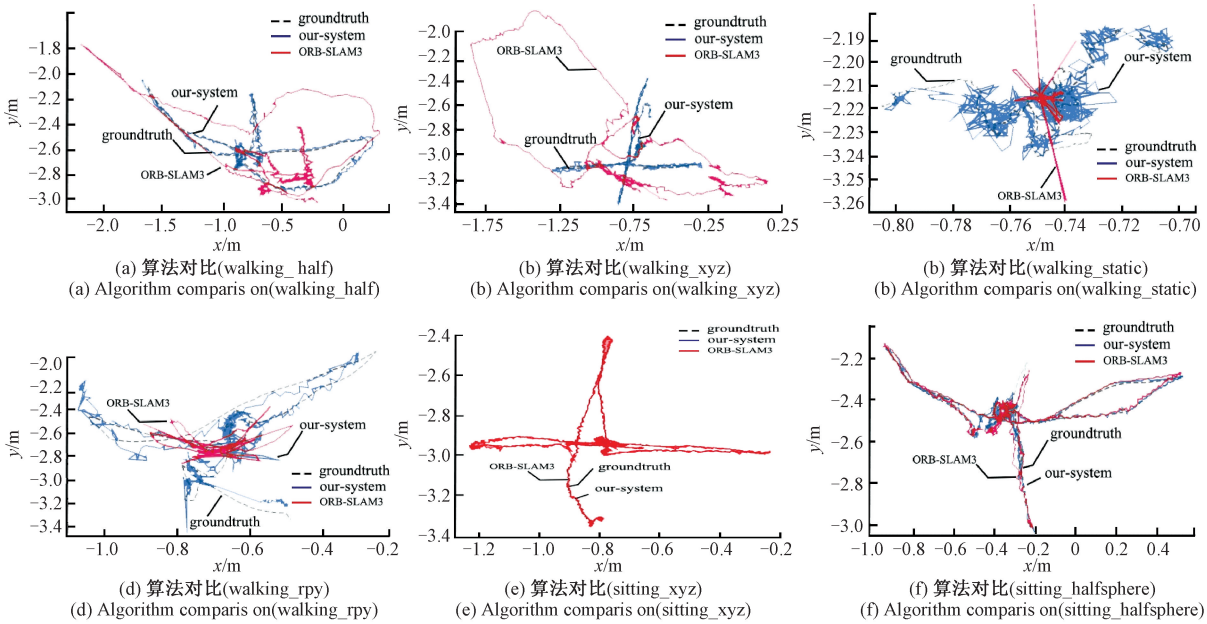
$$E_i = (Q_i^{-1} Q_{i+t})^{-1} (P_i^{-1} P_{i+t}) \tag{11}$$

要注意的是，在实际使用中，RPE 和 ATE 的评估需要结合实际情况，选择适合的指标来评价算法性能。

6.6　位姿估计误差分析

为了得到定位的精度需要判断系统的定位误差，因此本文进行位姿估计误差的分析实验，使用了 evo 工具来比较 ORB-SLAM3 系统的估计相机位姿和提供的真实位姿之间的误差，数据集提供的一个真实的位姿是保存在 groundtruth.txt 中，同时使用需要一个估计的相机位姿保存在 CameraTrajectory.txt 中。

图 7 显示本系统与 ORB-SLAM3 系统与真实值在 TUM 数据集上的估计轨迹的对比虚线表示相机的实际运动轨迹，蓝线表示估计的系统的运动轨迹，红线表示 ORB-SLAM3 系统的运动轨迹。可以看出，在低动态场景下，3 种方法的轨迹估计精度均较高。在高动态场景中，本文提出的系统的轨迹估计与实际轨迹的偏差较小，显示出更好的性能。相比之下，ORB-SLAM3 系统在高动态情况下的轨迹估计效果较差，偏差明显增大。



(a) 算法对比(walking_half)
(a) Algorithm comparis on(walking_half)

(b) 算法对比(walking_xyz)
(b) Algorithm comparis on(walking_xyz)

(b) 算法对比(walking_static)
(b) Algorithm comparis on(walking_static)

(d) 算法对比(walking_rpy)
(d) Algorithm comparis on(walking_rpy)

(e) 算法对比(sitting_xyz)
(e) Algorithm comparis on(sitting_xyz)

(f) 算法对比(sitting_halfsphere)
(f) Algorithm comparis on(sitting_halfsphere)

图 7　估计轨迹与真实轨迹图

Fig. 7　Estimated trajectory and ground truth trajectory plot

表 2、3 是将本文所设计的视觉 SLAM 系统与其他动态场景下 SLAM 系统进行对比，包括提出 Crowd-SLAM 系统是针对动态环境进行的优化，使用本文提出的 SLAM 与 Crowd-SLAM 以及 ORB-SLAM3 进行对比实

验,并对实验结果 ATE 和 PRE 进行对比和分析。本文的结果中给出了均方根误差(RMSE),平均误差,误差中值,和标准偏差(S. D.)的值。其中,RMSE 和 S. D. 更能体现系统的鲁棒性和稳定性,所以本文重点关注这两个值[23]。更能体现系统的鲁棒性和稳定性,所以本文重点关注这两个值,而且最好的结果是用了粗体突出显示。发现在 ATE 误差下,与 ORB-SLAM3 相比,本文提出的系统,系统定位在均方根误差下的误差平均减少都在 60%以上,在 sitting 低动态下系统误差也都有减少,6 个场景的平均 ATE 误差下误差减少平均率在 69.26%,其中在高动态环境下,误差平均减少率在 82.26%,在低动态环境下,误差平均减少率在 24.66%。系统定位在标准偏差下与 ORB-SLAM3 相比精确度大部分都有提升,6 个场景的误差减少平均率在 5.94%左右,值得注意在低动态 Siting/xyz 下,误差有所上升。在 RPE 误差下 6 个场景下,误差减少没有明显的变化总体都在下降趋势。与 Crowd-SLAM 相比,本文提出的系统在 ATE 误差下,误差平均减少率在 16%。

表 2　TUM 数据集的绝对轨迹误差结果

Table 2　Absolute trajectory error results of the TUM dataset　　m

序列	ORB-SLAM3				Crowd-SLAM				本文			
	RMSE	Mean	Median	S. D.	RMSE	Mean	Median	S. D.	RMSE	Mean	Median	S. D.
Walking/rpy	**0.019 9**	**0.006 3**	**0.003 3**	0.028 9	0.047 3	0.021 9	0.013 3	0.041 9	0.026 5	0.014 8	0.010 8	**0.022 0**
Walking/xyz	**0.019 0**	**0.007 9**	0.017 4	0.014 3	0.014 6	0.011 6	0.009 2	0.007 9	**0.012 8**	0.010 2	**0.008 3**	**0.007 7**
Walking/halfsphere	0.023 2	**0.008 0**	**0.006 1**	0.021 8	0.017 1	0.014 0	0.001 1	0.009 9	**0.015 6**	0.012 4	0.010 0	**0.009 4**
Walking/static	0.005 0	**0.001 4**	**0.000 7**	0.004 7	0.006 6	0.005 6	0.004 8	0.003 5	**0.004 9**	0.004 9	0.004 1	**0.003 2**
Sitting/halfsphere	0.011 7	**0.006 9**	0.006 0	0.004 4	0.018 1	0.013 8	0.010 4	0.011 6	**0.011 5**	0.008 8	**0.004 4**	**0.004 3**
Siting/xyz	**0.008 3**	**0.007 1**	**0.006 3**	**0.004 2**	0.012 9	0.010 7	0.009 2	0.007 2	0.009 5	0.007 9	0.006 2	0.005 3

表 3　TUM 数据集的相对轨迹误差结果

Table 3　Relative trajectory error results of the TUM dataset　　m

序列	ORB-SLAM3				Crowd-SLAM				本文			
	RMSE	Mean	Median	S. D.	RMSE	Mean	Median	S. D.	RMSE	Mean	Median	S. D.
Walking/rpy	0.162 5	0.140 8	0.115 5	0.081 1	0.058 5	0.043 2	0.034 1	0.039 4	**0.043 2**	**0.033 1**	**0.026 0**	**0.027 7**
Walking/xyz	0.270 1	0.244 1	0.236 0	0.115 1	0.016 2	0.013 7	0.012 1	0.008 5	**0.014 8**	**0.012 6**	**0.011 2**	**0.007 6**
Walking/halfsphere	0.341 8	0.294 5	0.220 6	0.173 4	0.022 4	0.018 6	0.015 5	0.012 4	**0.020 1**	**0.017 4**	**0.015 5**	**0.010 1**
Walking/static	0.024 7	0.021 0	0.018 1	0.013 0	0.008 6	0.007 5	**0.006 6**	0.004 2	**0.007 9**	**0.007 1**	0.006 7	**0.003 5**
Sitting/halfsphere	0.031 1	0.027 4	0.024 4	0.014 6	0.019 8	0.016 7	0.013 9	0.010 7	**0.016 0**	**0.013 9**	**0.012 4**	**0.007 9**
Siting/xyz	0.009 2	0.008 2	**0.006 8**	**0.004 9**	0.015 0	0.012 9	0.011 3	0.007 7	**0.009 1**	**0.008 1**	0.007 2	0.005 4

表 4 通过比较 KITTI 数据集上的 ATE 上的均方根误差而言,通过包含 IMU 集成的方法显示了此工作的有效性。在表中,ORB-SLAM3 表示基本系统。DynaSLAM 表示先进的户外 VO 系统,而+IMU 则表示在自动驾驶中使用的 VIO 系统。与 ORB-SLAM3 系统相比,提出的方法已经证明了能够降低绝对轨迹误差。而相对轨迹误差减少 10%左右。在环境 02 中,方法的效果不佳,主要原因是车流量较大以及静止车辆与移动车辆之间的距离较短,这可能导致错误点的剔除,从而影响性能。与 01 序列相比,02 序列在 IMU 增强方法下表现出明显的差异,这主要是由于该环境中复杂的光照条件影响了判断。

表 5 表示一个时间的比较,计算每个图像在不同子模块中的平均处理时间。同时也比较了 ORB-SLAM3 和本文提出的方法来跟踪时间,该系统跟踪时间仅仅比原始的 ORB-SLAM3 高出了 26.31 ms,经由 TensorRT 加速的语义模块的处理时间为每帧 10.23 ms,比几何模块快 4.09 ms。

表 4　KITTI 数据集绝对轨迹误差结果

Table 4　Absolute trajectory error results of the KITTI dataset　　m

序列	ORB-SLAM3	DynaSLAM	+IMU	本文
00	1.04	1.40	0.94	**0.88**
01	6.96	6.84	**2.08**	6.74
02	3.06	6.70	**0.74**	3.57
03	**0.26**	0.60	1.72	0.47
04	0.21	0.20	1.80	**0.16**
05	0.46	0.80	—	**0.42**
06	0.42	0.80	**0.26**	0.38

6.7　真实场景测试

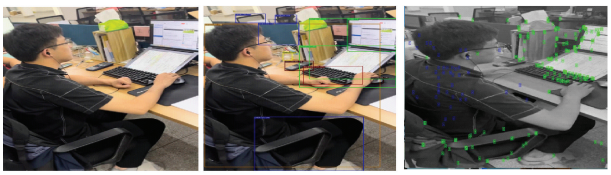
如图 8 使用单目摄像头采集的真实环境数据来测试

表 5　比较运行时间

Table 5　Comparison of execution time　　ms

方法	语义部分	几何部分	跟踪部分
ORB-SLAM3	—	—	11.12
本文	10.23	14.32	37.43

本文的系统。此时画面中的人正在站起，处于移动状态。展示了方法在处理真实动态场景中的有效性，图 8(a)显示图像中的对象，图 8(b)表示目标检测，图 8(c)显示了蓝色的动态点和绿色的静态点。

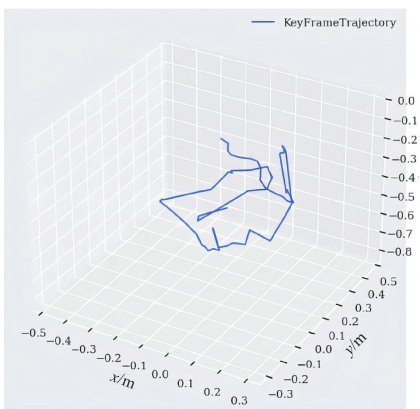


(a) 原始图像 (a) Original image　(b) 目标检测 (b) Object detection　(c) 动态点侦测 (c) Dynamic point detection

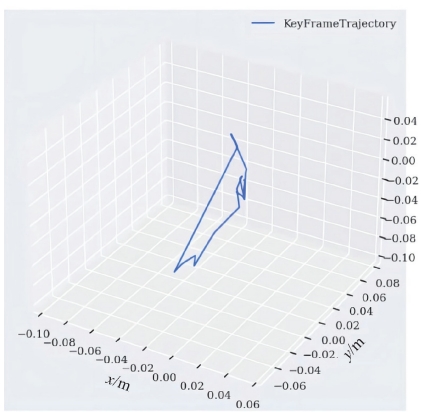
图 8　办公室实验场景图

Fig. 8　Diagram of the office scene

采用本文改进算法与 ORB-SLAM3 算法对上述两个分别进行算法测试，生成轨迹如图 9 所示，在缺乏真实世界场景中的真实轨迹的情况下，评估了小位移和抖动对室内环境的影响，可以看出本文算法影响更小。



(a) ORB-SLAM3
(a) ORB-SLAM3 algorithm



(b) 改进算法
(b) Improved al gorithm

图 9　两种算法轨迹比较图

Fig. 9　Trajectory comparison chart of two algorithms

7　结　　论

本文提出了一种视觉 SLAM 系统，为了能够减少环境中的动态物体对系统的影响，并处理动态环境中已知和有限未知的动态对象。本文使用方法保留更多的静态框，保留更多的静态框，就意味着能增加关键点的数量，同时减少动态对象的影响。在 TUM 数据集和 KITTI 数据集上对本文算法进行了测试。相比 ORB-SLAM3 系统，本文系统位姿估计精度提升了 69.26%。对比 Crowd-SLAM，DynaSLAM 等典型的动态 SLAM 系统，精度均有部分提升。并展示了一个可观的实时性的性能。并且本文的方法在 TUM 数据集的室内场景下证明了优越性，而 KITTI 数据集上实验则证明了它在双目环境相机下能够有效运行。此外在真实环境的单目相机下仍然能够保持良好的性能。当然在接下来工作中能够使用目标检测来检测更多的语义对象，为未来在动态环境下更好的能够构建语义地图用来完成自动驾驶汽车行驶，机器人的任务等打下坚实基础。下一步，将对整个网络模型进行修改，将网络移植到单独的嵌入式系统自动驾驶控制器上，为了加强模型训练和优化系统，可以改进目标检测方法，充分利用提取出的语义信息构建稀疏语义地图。这样可以更有效地进行系统优化。从而优化系统并实现更高层次的自动驾驶定位技术。

参考文献

[1] CHANGDI L, LEI Y, SHUMIN F. Large-scale, real-time 3D scene reconstruction using visual and IMU sensors [J]. IEEE Sensors Journal, 2020, 20(10):5597-5605.

[2] 张晨阳，黄腾，吴壮壮. 基于 K-means 聚类与深度学习的 RGB-D SLAM 算法[J]. 计算机工程，2022，48(1)：236-244，252.
ZHANG CH Y, HUANG T, WU ZH ZH. RGB-D SLAM algorithm based on K-means clustering and deep learning [J]. Computer Engineering, 2022, 48(1): 236-244, 252.

[3] MORAVEC H P. Towards automatic visual obstacle avoidance [M]. California: Morgan Kaufmann Publishers Inc, 1977.

[4] SUN Q F. An improved harris corner detection algorithm [J]. International Conferences on Communications, Signal Processing, and Systems, 2020, DOI: 10.1007/978-981-13-6504-1_14.22.

[5] LOWE D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[6] 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.

[7] ENGEL B, SCHÖPS T, CREMERS D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European Conference on Computer Vision(ECCV 2014), 2014: 834-849.

[8] VIRGOLINO C J S, MARCELO G, ANTONIO M M. Crowd-SLAM: Visual SLAM towards crowded environments using object detection[J]. Journal of Intelligent & Robotic Systems, 2021, 102(2): 235-250.

[9] 刘丰宇, 程向红, 曹毅. 基于深度学习与特征点速度约束的室内动态 SLAM 方法[J]. 中国惯性技术学报, 2023,31(5):438-443.
LIU F Y, CHENG X H, CAO Y. Indoor dynamic SLAM method based on deep learning and feature point velocity constraints[J]. Journal of Chinese Inertial Technology, 2023,31(5), 438-443.

[10] CHEN W, FANG M, LIU Y H, et al. Monocular semantic SLAM in dynamic street scene based on multiple object tracking[C]. IEEE International Conference on Cybernetics and Intelligent Systems and IEEE Conference on Robotics, Automation and Mechatronics, 2017: 599-604.

[11] 崔岸,张新颖,马耀辉. 复杂环境下基于自适应极线约束的 AGV 视觉 SLAM 算法[J]. 中国惯性技术学报, 2024,32(3):234-241.
CUI AN, ZHANG X Y, MA Y H. AGV visual SLAM algorithm based on adaptive epipolar constraints in complex environments[J]. Journal of Chinese Inertial Technology, 2024, 32(3): 234-241.

[12] 谢波,张国良,李歆,等. 一种单目 VIO 定位精度与跟踪稳定性优化方法[J]. 国外电子测量技术,2023, 42(4):23-30.
XIE B, ZHANG G L, LI X, et al. Optimization method for positioning accuracy and tracking stability of monocular VIO[J]. Foreign Electronic Measurement Technology,2023,42(4): 23-30.

[13] VIJAY B, ALEX K, ROBERTO C. SegNet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[14] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]. 2017 IEEE International Conference on Computer Vision(ICCV), 2017: 2980-2987.

[15] GE ZH, LIU S T, WANG F, et al. YOLOX: Exceeding YOLO series in 2021[J]. ArXiv preprint arXiv:2107.08430,2021.

[16] CARON F, DUFLO S, POMORSKI D, et al. GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects[J]. Information Fusion, 2017, 7(2): 221-230.

[17] 周立君,刘宇,白璐,等. 使用 TensorRT 进行深度学习推理[J]. 应用光学,2020,41(2):5.
ZHOU L J, LIU Y, BAI L, et al. Deep learning inference using TensorRT[J]. Applied Optics, 2020, 41(2): 5.

[18] REDMON J, FARHADI A. YOLOv3: An incremental improvement[J]. ArXiv preprint arXiv: 1804.02767, 2018.

[19] LIU Y H, GROSS L, LI ZH Q, et al. Automatic building extraction on high-resolution remote sensing imagery using deep convolutional encoder-decoder with spatial pyramid pooling[J]. IEEE Access, 2019, 7(1):128774-128786.

[20] GE ZH, LIU S T, LI Z M, et al. OTA: Optimal transport assignment for object detection. [C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),2021:303-312.

[21] ZHANG Z, SCARAMUZZA D. A tutorial on quantitative trajectory evaluation for visual(inertial) odometry[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 7244-7251.

[22] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012: 573-580.

[23] WU Y K, LUO L, YIN SH J,et al. An FPGA based energy efficient DS-SLAM accelerator for mobile robots in dynamic environment[J]. Applied Sciences, 2021, 11(4):1828.

作者简介

程强,硕士研究生,主要研究方向为深度学习、计算机视觉。

E-mail:915679860@qq.com

张友兵(通信作者),硕士,教授,主要研究方向为智能驾驶。

E-mail:zhangyb@huat.edu.cn

周奎,硕士,主要研究方向为嵌入式系统。

E-mail:44727726@qq.com