文章编号: 1001-0920(2018)06-1008-09 DOI: 10.13195/j.kzyjc.2017.0304

考虑多位姿估计约束的双目视觉里程计

张国良, 林志林†, 姚二亮, 徐 慧

(火箭军工程大学 控制科学与工程系，西安 710025)

摘 要: 为了提升复杂环境中双目视觉里程计的精度,提出一种考虑多位姿估计约束的双目视觉里程计方法. 首先,分别建立匹配深度已知点与深度未知点的数学模型,将深度未知点引入2D-2D位姿估计模型,从而充分利用图像信息;然后,基于关键帧地图点改进3D-2D位姿估计模型,并结合当前帧地图点更新关键帧地图点,从而增加匹配点对数,提高位姿估计精度;最后,根据改进的2D-2D及3D-2D位姿估计模型,建立多位姿估计约束位姿估计模型,结合局部光束平差法对位姿估计进行局部优化,达到定位精度高且累积误差小的效果. 数据集实验和实际场景在线实验表明,所提出方法满足实时定位要求,且有效地提高了自主定位精度.

关键词: 双目视觉里程计；位姿估计；局部光束平差法；数据集

中图分类号: TP242.6 **文献标志码:** A

Stereo visiual odometry with multi-pose estimation constraints

ZHANG Guo-liang, LIN Zhi-lin†, YAO Er-liang, XU Hui

(Department of Control Science and Engineering，Rocket Force Engineering University，Xi'an 710025，China)

Abstract: In order to improve the accuracy of the stereo visual odometry in complex environment, a method of stereo visiual odometry with multi-pose estimation constraints is proposed. Firstly, the mathematical models of matching points with the known depth and the unknown depth are established respectively, and then an improved 2D-2D pose estimation model considering the depth of unknown points is proposed, so that the image information can be used fully. Then, the 3D-2D pose estimation model is improved based on the keyframe mappoints, and the keyframe mappoints are updated according to the mappoints corresponding to the current frame. Therefore, the number of matched features can be more and the accuracy can be increased. Finally, according to the improved 2D-2D and 3D-2D pose estimation model, the pose estimation model with multi-pose estimation constraints is eatablished. Then the local bundle adjustment method is applied to optimize the estimated pose, so that the accuracy will be high and the cumulative error will be small. The experiments based on the datasets and the online experiment based on the actual scene show that, this method not only can meet the requirements of real-time location, but also can improve the accuracy of the mobile robot autonomous localization effectively.

Keywords: stereo visual odometry；pose estimation；local bundle adjustment；dataset

0 引 言

精确自主定位对于自主导航系统是至关重要的,近年来,基于视觉方法设计的视觉里程计逐渐成为自主定位的重要选择[1]. 视觉里程计是一种单纯利用单个或多个视觉传感器的输入对运载体(如汽车、人、机器人)的位姿进行准确估计的方法,它在机器人学、增强现实和自动驾驶等领域中有着重要的应用价值[2]. 视觉里程计具有使用成本低、功耗小、体积小、价格低廉、获取的信息丰富和定位精度更高等优点.

视觉里程计可分为单目视觉里程计和立体视觉里程计. 对于单目视觉里程计, Nister[3]通过建立两帧图像之间的二维特征对应关系,利用五点算法估计相对位姿,并利用RANSAC的方法迭代提纯,为视觉里程计研究提供了可参考的处理框架,对于后续的研究发展起到了推进作用[4]. 但单目视觉里程计最大的缺点是需要给定先验的空间尺度信息(即图像帧序列中的特征点的对应不能唯一确定,有一个变化的尺度因子),在模型求解过程中,会存在尺度因子的模糊性[5-6],并在计算过程中,需要更多的视觉特征点和图

收稿日期: 2017-03-22；**修回日期:** 2017-06-13.

责任编委: 毛志忠.

作者简介: 张国良 (1970−), 男, 教授, 博士, 从事先进控制理论与机器人技术等研究；林志林 (1993−), 男, 硕士生, 从事视觉里程计的研究.

†通讯作者. E-mail: lzl13468619594@163.com

像帧参与.

立体视觉里程计以双目视觉里程计为代表被广泛应用. 双目视觉里程计通过三角化的方法获得视野的景深信息,直接获得了图像特征点对应的世界三维坐标,解决了单目视觉中的尺度二义性[7]. 在立体视觉中,系统能通过视差图等方法对环境信息进行区分和感知,相对于单目视觉,其环境和地形的适应性更好,现亦广泛运用于粗糙地形和复杂环境中,如NASA的火星探测器. Scaramuzza等[8]提出的双目视觉里程计算法不同于以往采用3D-3D估计方式,而是开创性地采用3D-2D的运动估计方式,并采用了随机采样一致性算法(RANSAC)进行外点剔除,既提高了精度,又提升了实时性. 罗杨宇等[9]根据立体视觉算法得到匹配点对的三维对应关系,然后计算相机位姿后利用光束平差分段优化算法对其进行优化. 叶平等[10]利用欧氏距离进行立体匹配,采用特征点筛选、RANSAC算法和卡尔曼滤波等方法,提高了运动估计的准确性和鲁棒性. ORB-SLAM方法[11]利用非线性优化方法解决双目视觉里程计问题,其前端分为局部地图构建和跟踪线程,后端使用局部光束平差法[12-13]进行位姿估计的优化,估计精度较高. 但这些方法都没有考虑深度未知点的匹配信息,有很多图像信息因此被丢弃. Comport等[14]提出了2D-2D的位姿估计方法,该方法不再需要对所有双目匹配点对进行三角化,避免了三角化过程引入误差,但此类方法丢弃了过去时刻的3D环境信息,容易受累积误差的影响.

当物体距离远大于基线时,立体视觉也近似于单目视觉,即只能为在合适范围内的物体特征提供较为精确的景深信息. ORB-SLAM2中便使用了这一部分较精确的景深信息,但当图像中存在较大一部分超过合适范围的点时,如果不考虑这一部分点,则图像的很多信息会丢失,用于位姿估计的匹配点对将会很少,位姿估计的精度也会较低. 另外,ORB-SLAM2在进行初始位姿估计时仅基于前一图像帧进行位姿估计,在不使用闭环检测的情况下容易造成误差的累积,不利于大场景中的自主定位.

针对上述问题,本文在2D-2D位姿估计方法中,将景深精确度低或无景深信息的二维特征点纳入位姿估计模型中,充分利用图像信息,提高位姿估计的精度. 以ORB-SLAM2的研究为基础,在3D-2D位姿估计方法中,提出了考虑关键帧地图点的位姿估计方法,降低累积误差的影响. 建立考虑多位姿估计约束的位姿估计模型,同时利用来自于2D-2D和3D-2D位姿估计模型的约束进行位姿估计,以达到提升精度并降低累积误差的效果.

基于此,本文提出一种考虑多位姿估计约束的改进双目视觉里程计. 在2D-2D位姿估计方法中,分别建立与深度未知点和深度已知点匹配的误差模型,既获取了更多的图像间约束,又能够利用双目立体视觉获取的精确景深信息约束由单目视觉下的相机位姿估计引入的尺度因子模糊性,减小尺度误差,提升相机位姿估计的精确度. 基于关键帧地图点改进3D-2D位姿估计方法,通过引入关键帧,建立与当前帧的匹配关系进行位姿估计,降低累积误差的影响. 同时,结合当前帧地图点更新关键帧的地图点,提升地图点精确度并增加匹配点对的数量. 对比实验表明,所提出方法在定位精度方面表现出了良好的效果.

1　双目相机模型

假设地图点$\boldsymbol{P}$在两幅图像上的投影坐标(u, v)中的v是相同的,即双目图像已经矫正. 以左相机坐标系为相机坐标系,地图点$\boldsymbol{P}$在相机坐标系下的坐标为$\boldsymbol{P}_c = (x_c, y_c, z_c)$. 图1为双目视觉模型. 其中: b为左右相机之间的固定基线, f为相机焦距, u_l和u_r分别为地图点$\boldsymbol{P}$在左右相机上投影坐标的横坐标, $d = u_l - u_r$为视差.

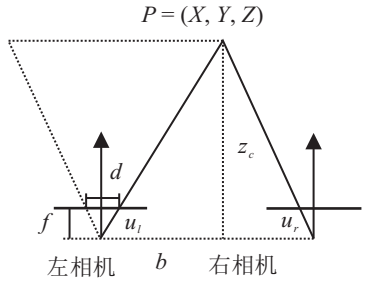


图1　双目视觉模型

地图点深度z_c可通过图1中三角形相似关系计算得到,即

$$z_c = \frac{f \cdot \text{baseline}}{d}. \tag{1}$$

由投影坐标和深度可以获得地图点在相机坐标系下包含真实尺度信息的坐标,以π函数表示为

$$\pi(u_l, v_l, z_c) = \begin{bmatrix} (u_l - cx) \cdot z_c / f \\ (v_l - pcy) \cdot z_c / f \\ z_c \end{bmatrix}. \tag{2}$$

相应地,π^{-1}函数即为投影函数,表示利用相机内参将相机坐标系下的3D点坐标投影到图像平面,得到2D图像坐标. 由于采用双目相机,需要完整考虑投影点的信息. 若地图点$\boldsymbol{P}$在左右相机上均有投影并且投影坐标已知,则π_{lr}^{-1}为

$$\pi_{lr}^{-1}(\boldsymbol{P}_c)=\begin{bmatrix} f\cdot x_c/z_c+cx \\ f\cdot y_c/z_c+cy \\ f\cdot(x_c-\text{baseline})/z_c+cx \end{bmatrix}, \quad (3)$$

包含地图点在右相机上的投影的横坐标信息. 若只有在左相机的投影坐标已知,或该地图点在右相机上没有投影,则π_l^{-1}为

$$\pi_l^{-1}(\boldsymbol{P}_c)=\begin{bmatrix} f\cdot x_c/z_c+cx \\ f\cdot y_c/z_c+cy \end{bmatrix}, \quad (4)$$

其中(f,cx,cy)为相机内参.

2 改进双目视觉里程计

双目视觉里程计是对图像帧提取的特征点进行匹配,然后估计相机位姿变换关系的过程.

根据估计时所利用信息涉及到的是二维特征还是三维地图点,可以将位姿估计分为3类[15]:

1) 2D-2D: 前一帧与当前帧所使用的信息均只涉及到二维特征[16].

2) 3D-3D: 前一帧与当前帧所使用的信息均只涉及到三维地图点,为此,需要每一帧三角化得到特征点对应的三维地图点[17].

3) 3D-2D: 前一帧所使用的信息是三维地图点,而当前帧使用的是与前一帧地图点形成匹配的二维特征[18].

根据文献[19],以上3种估计方式中,2D-2D位姿估计方式和3D-2D位姿估计方式更优于3D-3D位姿估计方式,这是因为三角化对像素误差有放大作用,得到的3D点在深度方向上的不确定性较大,使用3D-3D位姿估计方式会将这种不确定性放大,并且放大后的误差无法在优化过程中被消除. 另外,在使用远距离的匹配点对进行运动估计时,由于三角化点的深度不确定性较强,2D-2D位姿估计方式相对于3D-2D位姿估计方式会更加精确,且2D-2D位姿估计方式能够更加充分地使用图像信息. 综上考虑,本文在ORB_SLAM2的基础上,结合来自于2D-2D位姿估计模型的约束,使用图像中深度未知的特征点信息,提高位姿估计精度,同时基于关键帧地图点改进3D-2D位姿估计方式,降低累积误差的影响,进一步提升定位精确度.

2.1 考虑深度未知点的2D-2D位姿估计模型

每一帧图像中的特征点可分为3类,如图2所示,圆点是与上一帧的地图点形成匹配而获得深度的点,正方形点是通过双目相机模型获得深度的点,三角形点是深度未知的点.

假设图像帧k对应的相机坐标系为C_k,对应的特征点集合为Ω_k,图像中第i个点的坐标为$\boldsymbol{X}_k^i$,图像帧m和图像帧n之间的相机位姿变换矩阵为$\boldsymbol{T}_m^n$,旋转矩阵表示为$\boldsymbol{R}_m^n$,平移矩阵表示为$\boldsymbol{t}_m^n$,将深度已知点的坐标记为$\boldsymbol{X}_k^i=[x_k^i,y_k^i,z_k^i]^{\mathrm{T}}$,深度未知点的坐标记为$\tilde{\boldsymbol{X}}_k^i=[\tilde{x}_k^i,\tilde{y}_k^i,1]^{\mathrm{T}}$, $\boldsymbol{X}_k^i=z_k^i\tilde{\boldsymbol{X}}_k^i$.

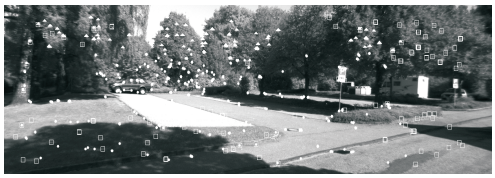


图2 图像特征点

使图像帧间相似的特征点形成匹配点对,则匹配成功的点存在如下关系:

$$z_k^i\tilde{\boldsymbol{X}}_k^i=\boldsymbol{R}_{k-1}^k\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^k, \quad (5)$$

或

$$z_k^i\tilde{\boldsymbol{X}}_k^i=z_{k-1}^i\boldsymbol{R}_{k-1}^k\tilde{\boldsymbol{X}}_{k-1}^i+\boldsymbol{t}_{k-1}^k. \quad (6)$$

每一时刻只取左相机的图像进行匹配,对于当前帧中的特征点,根据与之形成匹配的前一帧特征点是否具有已知深度[20],分情况进行处理:

1) 匹配前一帧图像中深度已知点. 将式(5)展开可得

$$z_k^i\begin{bmatrix}\tilde{x}_k^i\\ \tilde{y}_k^i\\ 1\end{bmatrix}=\begin{bmatrix}\boldsymbol{R}_{k-1}^k(1)\\ \boldsymbol{R}_{k-1}^k(2)\\ \boldsymbol{R}_{k-1}^k(3)\end{bmatrix}\boldsymbol{X}_{k-1}^i+\begin{bmatrix}\boldsymbol{t}_{k-1}^k(1)\\ \boldsymbol{t}_{k-1}^k(2)\\ \boldsymbol{t}_{k-1}^k(3)\end{bmatrix}. \quad (7)$$

其中: $\boldsymbol{R}_{k-1}^k(i)$为旋转矩阵的第$i$行, $\boldsymbol{t}_{k-1}^k(i)$为平移矩阵中的第$i$行. 进而有

$$\begin{cases} z_k^i\tilde{x}_k^i=\boldsymbol{R}_{k-1}^k(1)\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^k(1), \\ z_k^i\tilde{y}_k^i=\boldsymbol{R}_{k-1}^k(2)\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^k(2), \\ z_k^i=\boldsymbol{R}_{k-1}^k(3)\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^k(3). \end{cases} \quad (8)$$

将z_k^i消去,可以得到如下两个方程:

$$\begin{cases} (\boldsymbol{R}_{k-1}^k(1)-\tilde{x}_k^i\boldsymbol{R}_{k-1}^k(3))\boldsymbol{X}_{k-1}^i+ \\ \boldsymbol{t}_{k-1}^k(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^k(3)=0, \\ (\boldsymbol{R}_{k-1}^k(2)-\tilde{y}_k^i\boldsymbol{R}_{k-1}^k(3))\boldsymbol{X}_{k-1}^i+ \\ \boldsymbol{t}_{k-1}^k(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^k(3)=0. \end{cases} \quad (9)$$

2) 匹配前一帧图像中深度未知点. 将式(6)展开可得

$$z_k^i\begin{bmatrix}\tilde{x}_k^i\\ \tilde{y}_k^i\\ 1\end{bmatrix}=\begin{bmatrix}\boldsymbol{R}_{k-1}^k(1)\\ \boldsymbol{R}_{k-1}^k(2)\\ \boldsymbol{R}_{k-1}^k(3)\end{bmatrix}z_{k-1}^i\tilde{\boldsymbol{X}}_{k-1}^i+\begin{bmatrix}\boldsymbol{t}_{k-1}^k(1)\\ \boldsymbol{t}_{k-1}^k(2)\\ \boldsymbol{t}_{k-1}^k(3)\end{bmatrix}. \quad (10)$$

进而有

$$\begin{cases}(\boldsymbol{R}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{R}_{k-1}^{k}(3))z_{k-1}^i\tilde{\boldsymbol{X}}_{k-1}^i+\\ \boldsymbol{t}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(3)=0,\\ (\boldsymbol{R}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{R}_{k-1}^{k}(3))z_{k-1}^i\tilde{\boldsymbol{X}}_{k-1}^i+\\ \boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(3)=0.\end{cases}\tag{11}$$

将z_{k-1}^i消去,可以得到

$$(\boldsymbol{t}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(3))(\boldsymbol{R}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{R}_{k-1}^{k}(3))\tilde{\boldsymbol{X}}_{k-1}^i=(\boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(3))(\boldsymbol{R}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{R}_{k-1}^{k}(3))\tilde{\boldsymbol{X}}_{k-1}^i.\tag{12}$$

即

$$[(\boldsymbol{t}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(3))\boldsymbol{R}_{k-1}^{k}(2)+(\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(1))\boldsymbol{R}_{k-1}^{k}(3)-(\boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(3))\cdot\boldsymbol{R}_{k-1}^{k}(1)]\tilde{\boldsymbol{X}}_{k-1}^i=0.\tag{13}$$

综合两种情况,由所有匹配点对所获得的方程可以估计当前帧位姿$\boldsymbol{R}$和$\boldsymbol{t}$. 然而,由于匹配误差等的存在,方程组无法得到精确解,通过最小化如下特征点匹配误差进行位姿估计:

$$\min_{\boldsymbol{R}_{k-1}^{k},\boldsymbol{t}_{k-1}^{k}}\Big(\sum_{i=1}^{N_1}(\|{}_0e_k^i\|_2+\|{}_1e_k^i\|_2)+\sum_{i=1}^{N_2}\|{}_2e_k^i\|_2\Big).\tag{14}$$

其中:N_1表示与深度已知点形成的匹配点对数,N_2表示与深度未知点形成的匹配点对数;${}_0e_k^i$、${}_1e_k^i$指与深度未知点匹配时对应的方程误差,有

$${}_0e_k^i=(\boldsymbol{R}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{R}_{k-1}^{k}(3))\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(3),\tag{15}$$

$${}_1e_k^i=(\boldsymbol{R}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{R}_{k-1}^{k}(3))\boldsymbol{X}_{k-1}^i+\boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(3);\tag{16}$$

${}_2e_k^i$指与无深度点匹配时对应的方程误差,有

$${}_2e_k^i=[(\boldsymbol{t}_{k-1}^{k}(1)-\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(3))\boldsymbol{R}_{k-1}^{k}(2)+(\tilde{x}_k^i\boldsymbol{t}_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(1))\boldsymbol{R}_{k-1}^{k}(3)-(t_{k-1}^{k}(2)-\tilde{y}_k^i\boldsymbol{t}_{k-1}^{k}(3))\boldsymbol{R}_{k-1}^{k}(1)]\tilde{\boldsymbol{X}}_{k-1}^i.\tag{17}$$

2.2　基于改进3D-2D位姿估计模型的位姿跟踪

2.2.1　考虑关键帧地图点的位姿估计

在ORB_SLAM2中,进行初始位姿估计时将上一帧的地图点投影至当前帧构造匹配点对,该方法虽然能够形成较多的匹配点对,但上一帧位姿估计误差会不断累积使匹配过程产生较大的误差. 若将关键帧的地图点投影至当前帧来构造匹配点对[21-22],则虽然相对而言形成的匹配点对会较少,但能够有效减少累积误差的影响. 因此,本文综合考虑上一帧与关键帧的地图点,一方面增加更多的匹配点对,为位姿估计提供更多的约束条件,另一方面减弱上一帧累积误差的影响. 如图3所示,空心点为关键帧地图点,实心点为上一帧地图点.

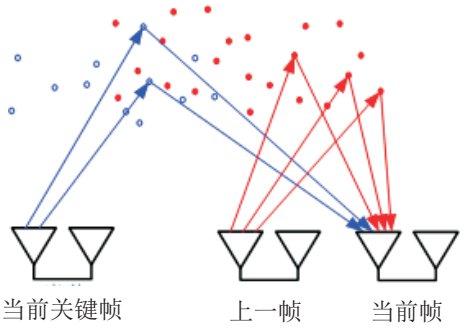


图3　考虑关键帧地图点的位姿估计

假设关键帧及上一帧中的待匹配地图点为$P=\{\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_N\}$,相机获取到一帧图像$\mathcal{F}$,从中提取出一组特征点$Q=\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_N\}$.

首先,对待匹配地图点进行筛选,筛选原则如下:1) 将这些地图点投影到当前帧上,剔除投影位置在当前帧范围外的地图点;2) 计算这些地图点与当前帧所对应的相机中心的距离d,如果$d\notin[d_{\min},d_{\max}]$,则剔除;3) 计算当前帧视线矢量$\boldsymbol{v}$与这些地图点平均观测方向矢量$\boldsymbol{n}$之间的夹角,如果$\dfrac{\boldsymbol{v}\cdot\boldsymbol{n}}{|\boldsymbol{v}|\cdot|\boldsymbol{n}|}<\cos(60°)$,则剔除. 其中:当前帧的视线矢量$\boldsymbol{v}$指相机的投影平面法向量,地图点的平均观测方向矢量$\boldsymbol{n}$指可观测到该地图点的关键帧的光学中心与该地图点连线的方向矢量的平均. 将筛选后地图点与当前帧特征点进行匹配,然后将其投影到当前帧中,理想情况下应满足

$$\boldsymbol{q}_i=\pi^{-1}(\boldsymbol{R}_0^k\boldsymbol{P}_i+\boldsymbol{t}_0^k),\ \forall i\in\boldsymbol{N}.\tag{18}$$

实际上,由于匹配误差等的存在,式(18)无法求解,通过最小化如下地图点投影误差求解当前帧位姿$\boldsymbol{R}$和$\boldsymbol{t}$:

$$\min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{N_3}\|{}_3e_k^i\|_2.\tag{19}$$

其中

$${}_3e_k^i=\boldsymbol{q}_i-\pi^{-1}(\boldsymbol{R}_0^k\boldsymbol{P}_i+\boldsymbol{t}_0^k),\tag{20}$$

N_3为待匹配地图点与当前帧特征点构成的匹配点对数.

2.2.2　关键帧的生成

严格控制关键帧的生成有利于减少累积误差的影响,但这也会导致关键帧与当前帧之间的匹配点对过少,因此本文设立关键帧生成标准进行权衡. 要生成一个关键帧,必须同时满足以下几个条件:

1) 从上一关键帧生成起至少经过20帧图像;

2) 跟踪到至少50个地图点;

3) 与上一关键帧的距离超过设定的阈值.

为了衡量两帧图像的距离,定义距离函数为

$$\mathrm{dist}(\boldsymbol{\xi}_{ij}) = \boldsymbol{\xi}_{ij}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{\xi}_{ij}. \tag{21}$$

其中:$\boldsymbol{\xi}_{ij}$为第i帧与第j帧关键帧之间相对位姿变换对应的李代数表示[23],$\boldsymbol{W}$为对角权值矩阵. 设定一个阈值$d_{\mathrm{threshold}}$,只有在$\mathrm{dist}(\boldsymbol{\xi}_{ij}) > d_{\mathrm{threshold}}$时才能满足条件3).

因为对多个关键帧进行基于局部光束平差法的位姿优化较为耗时,而普通相机采样频率一般在30 fps以下,因此条件1)设置了从上一关键帧起至少经过20帧图像,即时间上至少经过0.67 s,保证了在生成新关键帧前有充足的时间进行基于光束平差法的位姿优化;条件2)保证了提供的地图点足够多,所进行的位姿估计足够精确;条件3)保证了在相机运动较小时不会生成过多冗余的关键帧.

2.2.3 关键帧地图点的更新

一般而言,随着相机的移动,当前帧与关键帧之间的距离不断变大,两帧之间可形成匹配点对的点会越来越少. 为了提升关键帧地图点质量并增加其数量,本文在对当前帧位姿进行估计后,对关键帧地图点进行更新.

基于双目相机模型生成关键帧的地图点,并将关键帧中的深度未知点和观测帧较少的地图点所对应的特征点视为待更新对象,其中观测帧定义为:若地图点被某图像帧观测到,则称该图像帧为地图点的观测帧. 为待更新对象在当前帧中寻找匹配点,若该匹配点具有对应的地图点,且其观测帧较多,则将该地图点作为待更新对象所对应的地图点.

通过上述方法对关键帧地图点进行更新,关键帧地图点数量逐渐增加,且部分地图点的观测帧会越来越多,这有助于该关键帧与下一帧进行匹配时形成更多且更精确的匹配点对.

3 考虑多位姿估计约束的双目视觉里程计

3.1 考虑多位姿估计约束的位姿跟踪

本文建立考虑多位姿估计约束的位姿估计模型,同时利用来自于2D-2D和3D-2D位姿估计模型的约束进行位姿估计,既充分使用图像信息,增加进行运动估计时的约束,提升位姿估计精度,又可以减弱累积误差的影响,进一步提高定位精确度,总体流程图如图4所示. 基于此,将2D-2D和3D-2D运动估计模型对应的误差函数进行整合,构建如下误差函数:

$$F(\boldsymbol{\xi}) = \sum_{i=1}^{N_1}(\|_0 e_k^i\|_2 + \|_1 e_k^i\|_2) + \sum_{i=1}^{N_2}\|_2 e_k^i\|_2 + \sum_{i=1}^{N_3}\|_3 e_k^i\|_2. \tag{22}$$

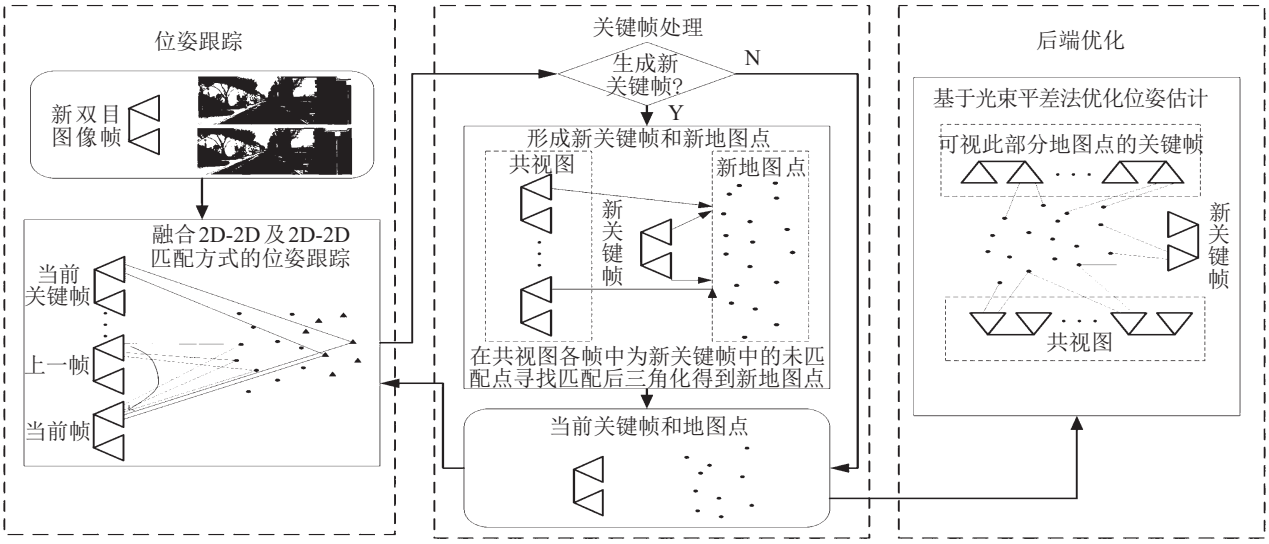


图4 考虑多位姿估计约束的双目视觉里程计流程

为了解决上述非线性优化问题,将相机位姿$\boldsymbol{R}$和$\boldsymbol{t}$表示为对应的李代数,有

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{\upsilon} \\ 0 & 0 \end{bmatrix} \in \mathrm{se}(3). \tag{23}$$

其中:$\boldsymbol{\xi} = [\boldsymbol{\omega}, \boldsymbol{\upsilon}]^{\mathrm{T}} \in \boldsymbol{R}^6, \boldsymbol{\omega} \in \boldsymbol{R}^3, \boldsymbol{\upsilon} \in \boldsymbol{R}^3$. 定义符号$\bigwedge$为某向量对应的反对称矩阵,即对于$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$,有

$$\hat{\boldsymbol{\omega}}=\begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}, \tag{24}$$

$$\boldsymbol{R}=\exp_{\mathrm{se}(3)}(\hat{\boldsymbol{\omega}}),\ \boldsymbol{t}=\boldsymbol{J}_l\boldsymbol{\upsilon}, \tag{25}$$

且

$$\boldsymbol{J}_l=\sum_{n=0}^{+\infty}\frac{1}{(n+1)!}\hat{\boldsymbol{\omega}}^n. \tag{26}$$

位姿跟踪结果为

$$\boldsymbol{\xi}^*=\underset{\boldsymbol{\xi}}{\operatorname{argmin}}\,F(\boldsymbol{\xi}). \tag{27}$$

本文使用g2o工具[24]通过Guass-Newton方法对上述非线性优化问题进行求解,g2o中的节点即为相机位姿,每一个来自于2D-2D和3D-2D位姿估计模型的约束条件作为g2o的边. 在每一个迭代过程中,求得的位姿增量如下:

$$\delta\boldsymbol{\xi}^{(n)}=-(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}}e, \tag{28}$$

其中

$$\boldsymbol{J}=\frac{\partial e(\boldsymbol{\varepsilon}\circ\boldsymbol{\xi}^{(n)})}{\partial\boldsymbol{\varepsilon}}\Big|_{\boldsymbol{\varepsilon}}=0. \tag{29}$$

各个误差项对位姿的偏导表示如下:在2D-2D位姿估计方法中,对于深度已知点,有

$$\begin{cases}\dfrac{\partial_0 e_k^i}{\partial\boldsymbol{\omega}}=-[1,0,-\tilde{x}_k^i](\hat{\tilde{\boldsymbol{X}}}'+\hat{\boldsymbol{t}}_{k-1}^k),\\ \dfrac{\partial_0 e_k^i}{\partial\boldsymbol{\upsilon}}=[1,0,-\tilde{x}_k^i];\end{cases} \tag{30}$$

$$\begin{cases}\dfrac{\partial_1 e_k^i}{\partial\boldsymbol{\omega}}=-[0,1,-\tilde{y}_k^i](\hat{\tilde{\boldsymbol{X}}}'+\hat{\boldsymbol{t}}_{k-1}^k),\\ \dfrac{\partial_1 e_k^i}{\partial\boldsymbol{\upsilon}}=[0,1,-\tilde{y}_k^i].\end{cases} \tag{31}$$

在2D-2D位姿估计方法中,对于深度未知点,有

$$\begin{cases}\dfrac{\partial_2 e_k^i}{\partial\boldsymbol{\omega}}=\boldsymbol{t}_{k-1}^{k\mathrm{T}}\hat{\tilde{\boldsymbol{X}}}_k^i\hat{\tilde{\boldsymbol{X}}}'-\hat{\tilde{\boldsymbol{X}}}'^{\mathrm{T}}\hat{\tilde{\boldsymbol{X}}}_k^i\cdot\hat{\boldsymbol{t}}_{k-1}^k,\\ \dfrac{\partial_2 e_k^i}{\partial\boldsymbol{\upsilon}}=\tilde{\boldsymbol{X}}'^{\mathrm{T}}\cdot\hat{\tilde{\boldsymbol{X}}}_k^i.\end{cases} \tag{32}$$

在3D-2D位姿估计方法中,有

$$\begin{cases}\dfrac{\partial_3 e_k^i}{\partial\boldsymbol{\omega}}=\boldsymbol{J}_K\boldsymbol{R}_0^{k-1}(\hat{\boldsymbol{P}}'+\hat{\boldsymbol{t}}_{k-1}^k),\\ \dfrac{\partial_3 e_k^i}{\partial\boldsymbol{\upsilon}}=-\boldsymbol{J}_K\boldsymbol{R}_0^{k-1}.\end{cases} \tag{33}$$

其中

$$\tilde{\boldsymbol{X}}'=\boldsymbol{R}_{k-1}^k\cdot\tilde{\boldsymbol{X}}_{k-1}^i, \tag{34}$$

$$\boldsymbol{X}'=\boldsymbol{R}_{k-1}^k\cdot\boldsymbol{X}_{k-1}^i, \tag{35}$$

$$\boldsymbol{P}'=\boldsymbol{R}_{k-1}^k\cdot\boldsymbol{P}_i. \tag{36}$$

通过不断更新位姿来最小化误差值,可以得到相机的位姿估计

$$\boldsymbol{\xi}^{(n+1)}=\delta\boldsymbol{\xi}^{(n)}\circ\boldsymbol{\xi}^{(n)}. \tag{37}$$

其中定义符号$\circ$为

$$\boldsymbol{\xi}_{ki}:=\boldsymbol{\xi}_{kj}\circ\boldsymbol{\xi}_{ij}:=$$
$$\log_{\mathrm{SE}(3)}(\exp_{\mathrm{se}(3)}(\boldsymbol{\xi}_{kj})\cdot\exp_{\mathrm{se}(3)}(\boldsymbol{\xi}_{ji})). \tag{38}$$

3.2 基于局部光束平差法优化位姿估计

由于许多相机位姿是相互关联的,在位姿跟踪环节所估计出的位姿不可避免地会存在累积误差. 为了减小累积误差,在每次生成关键帧后,利用局部光束平差法进行位姿优化,如图4后端优化部分,该方法通过利用多个时刻的相机位姿和地图点之间的约束进行关键帧的位姿优化. 本文局部光束平差法优化的对象为:

1) 当前帧的相机位姿$\boldsymbol{\xi}_{c0}$;

2) 当前帧$\boldsymbol{\kappa}_c$的共视图[25]covisible($\boldsymbol{\kappa}_c$)中各帧对应的相机位姿,定义这些位姿构成的集合为$\mathcal{I}$;

3) 当前帧$\boldsymbol{\kappa}_c$及其共视图covisible($\boldsymbol{\kappa}_c$)中各帧观测到的地图点,定义这些地图点位置构成的集合为$\mathcal{M}$.

另外,观测到地图点集合$\mathcal{M}$的其他图像帧的位姿参与了优化但不被优化,定义这些位姿构成的集合为$\mathcal{U}$. 通过最小化如下误差函数进行局部光束平差法优化:

$$\underset{\hat{\boldsymbol{\xi}}_{i0}\in\hat{\boldsymbol{\xi}}_{c0}\cup\mathcal{I}}{\arg\min}\sum_{\hat{\boldsymbol{\xi}}_{i0}\in\mathcal{I}\cup\mathcal{U}}\sum_{\boldsymbol{p}_k\in\mathcal{M}}\|\boldsymbol{q}'_{ik}-\pi^{-1}(\exp(\hat{\boldsymbol{\xi}}_{i0})\cdot\boldsymbol{p}_k)\|_2, \tag{39}$$

其中$\boldsymbol{q}'_{ik}$为$\hat{\boldsymbol{\xi}}_{i0}$对应的图像帧中与地图点$\boldsymbol{p}_k$匹配的特征点像素坐标.

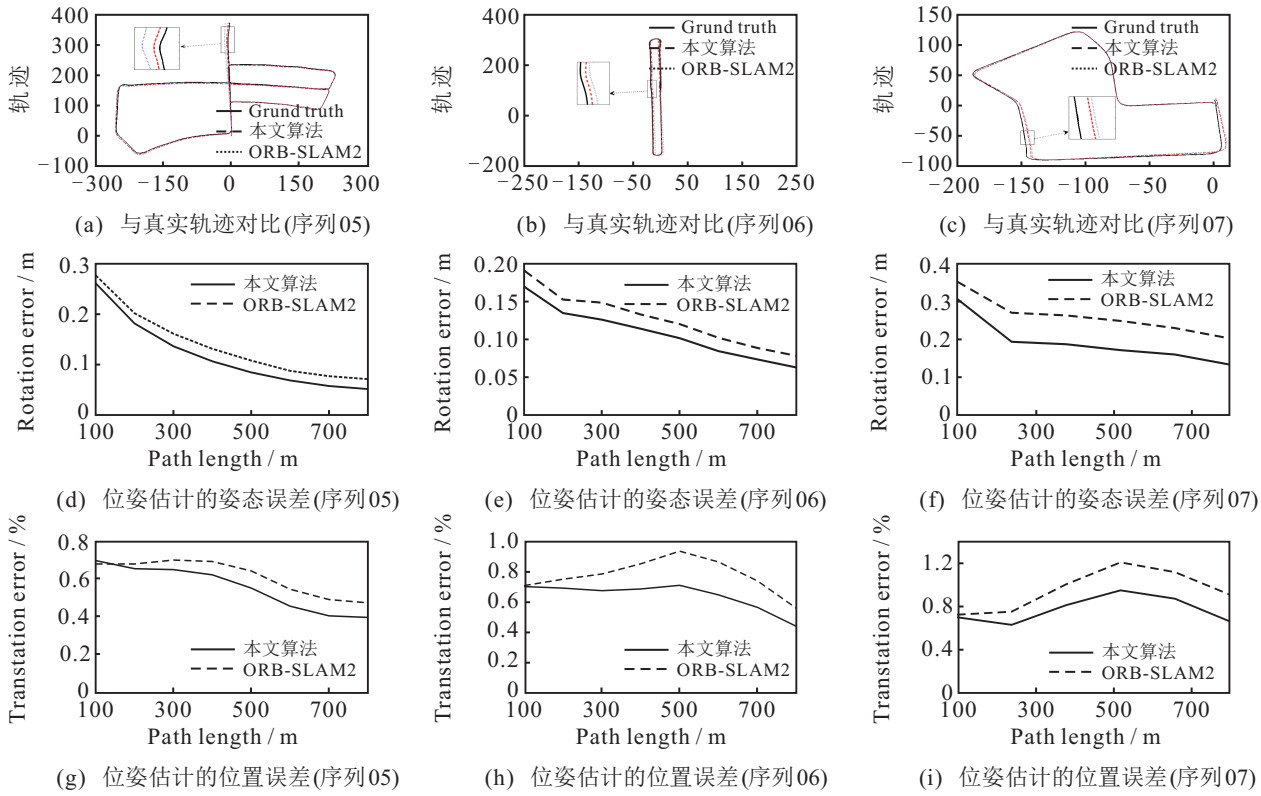
4 实验分析

实验所用电脑配置为:CPU I7处理器,主频2.5 GHz,内存4 G,不使用GPU加速,系统为Ubuntu 14.04. 首先使用Kitti数据集进行实验,与ORB_SLAM2方法进行对比,并手持Bumblebee传感器进行在线实际场景实验.

4.1 Kitti数据集实验

Kitti数据集[26]为车载双目相机采集的城市、高速路、乡村场景,该双目相机基线为54 cm,帧率为10 Hz,图像分辨率为1 392 × 512. 其中序列00、02、05、06、07和09包含了闭环.

为了评价视觉里程计的精度,排除闭环对相机位姿估计精度的影响,关闭本文方法和ORB_SLAM2算法的闭环检测[27],图5为3组实验结果对比. 所采用的评价标准为100 m, 200 m, ···, 800 m轨迹分别对应的相对位姿估计误差.



(a) 与真实轨迹对比(序列05) (b) 与真实轨迹对比(序列06) (c) 与真实轨迹对比(序列07)

(d) 位姿估计的姿态误差(序列05) (e) 位姿估计的姿态误差(序列06) (f) 位姿估计的姿态误差(序列07)

(g) 位姿估计的位置误差(序列05) (h) 位姿估计的位置误差(序列06) (i) 位姿估计的位置误差(序列07)

图 5 Kitti数据集对比实验结果

表 1 Kitti数据集对比实验表

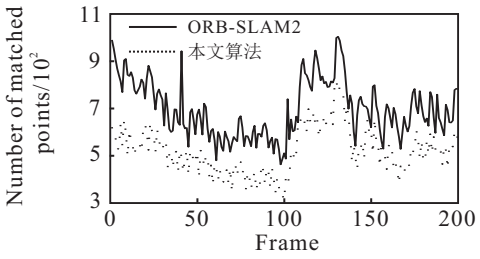
Kitti序列	场景属性		My method		ORB_SLAM2	
	长度 / m	环境	t_error / %	R_error / (deg / 100 m)	t_error / %	R_error / (deg / 100 m)
00	3 714	城市	0.74	0.26	0.84	0.30
01	4 268	高速路	1.34	0.17	1.54	0.18
02	5 075	城乡	0.75	0.26	0.82	0.30
03	563	乡村	0.68	0.16	0.72	0.17
04	397	乡村	0.39	0.16	0.44	0.12
05	2 223	城市	0.59	0.21	0.62	0.24
06	1 239	城市	0.66	0.19	0.78	0.22
07	695	城市	0.78	0.36	0.96	0.46
08	3 225	城乡	1.01	0.31	1.02	0.29
09	1 717	城乡	0.79	0.24	0.84	0.23
10	919	城乡	0.61	0.21	0.62	0.23
平均	—	—	0.76	0.23	0.84	0.25

表1对本文方法和对比方法在不同序列下的位姿估计误差进行了对比,加下划线的为误差较低者. 由实验结果可见,使用本文所提出的考虑多位姿估计约束的位姿估计方法后,大多数序列的位姿估计精度都得到了提升. 其中,相对于ORB_SLAM2,姿态估计的精度提高了约0.02 deg / 100 m,位置估计的精度提高了约0.08 %. 这是因为考虑多位姿估计约束的方法使更多的匹配点对(包含3D-2D匹配点对和2D-2D匹配点对)参与了位姿估计,为位姿估计增加了约束,有助于提升位姿估计精度[28].

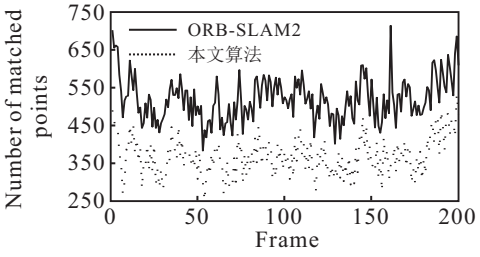
由于ORB_SLAM2进行当前帧位姿初始估计时仅使用了上一帧的地图点,受上一帧位姿估计误差的影响较大,且对上一帧的深度未知点不进行匹配,失去了较多的图像信息. 本文方法一方面使用了关键帧对应的地图点,增加了3D-2D估计时的地图点数量,并且降低了上一帧累积误差的影响,同时根据

与当前帧的匹配关系,更新关键帧的地图点. 另一方面引入了利用上一帧中深度已知点及深度未知点的2D-2D位姿估计模型,充分利用了上一帧的图像信息,进一步提升了位姿估计精度.

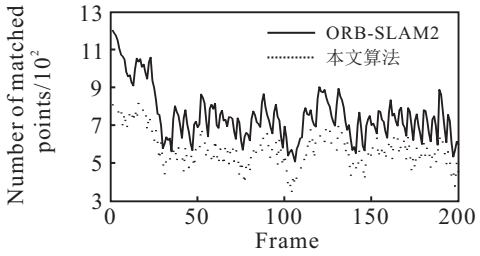
图6为序列05、06、07中估计各帧位姿时所使用匹配点对数量,可见本文方法使用匹配点对数量明显增加. 通过对比实验可以看出,不加闭环检测的情况下,在大范围场景中进行位姿估计时,本文方法的估计精度要好于ORB_SLAM2.



(a) 序列05



(b) 序列06



(c) 序列07

图6　匹配点对数量对比

4.2　实际场景在线实验

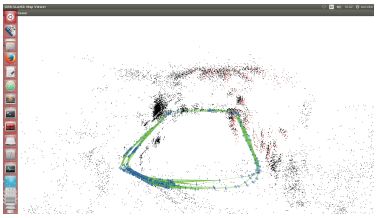
本文以实验室的模拟室内场景为实验场景,如图7(a)所示. 手持Bumblebee传感器(图7(b))进行在线实验,传感器采用全局快门曝光方式,帧率为30 fps,图像分辨率为640 × 480,基线为12 cm. 手持相机沿图中路线绕行实验场景3圈,进行在线定位. 在线实验结果如图8所示. 图8(a)观察视角与图7(a)一致,图8(b)为估计得到的运动轨迹的俯视图.

由于该场景中存在大面积白墙、玻璃等,对视觉里程计是一个较大的挑战. 从估计得到的运动轨迹看,三圈轨迹之间没有明显的错位现象,均很好地吻合实际运动轨迹,表明定位效果较好. 该实验表明,本文方法能够达到实时且精确进行相机位姿估计的效果.

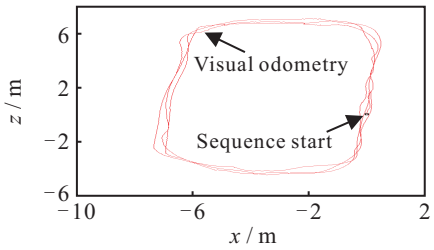
(a) 实验场景

(b) 实验设备

图7　在线实验场景与实验设备



(a) 运动轨迹和地图点



(b) 运动轨迹估计结果

图8　实际场景在线实验结果

5　结　论

本文从视觉里程计的精确度出发,提出了考虑多位姿约束的双目视觉里程计. 一方面,通过将深度未知点纳入2D-2D位姿估计模型中,充分利用了图像信息,提升了位姿估计精度;另一方面,考虑关键帧的地图点来改进3D-2D位姿估计模型,降低了累积误差的影响,进一步提高了定位精确度. 最后,同时考虑来自于2D-2D和3D-2D位姿估计模型的约束,改进双目视觉里程计提升定位精度. Kitti数据集和实际场景在线实验表明,所提出的改进双目视觉里程计有效地提高了视觉里程计的自主定位精度,并且满足实时定位的要求.

参考文献(References)

[1] Fuentes Pacheco J, Ruiz Ascencio J, Rend'on Mancha J M. Visual simultaneous localization and mapping: A survey[J]. Arti Cial Intelligence Review, 2015, 43(1): 55-81.

[2] Fraundorfer F, Scaramuzza D. Visual odometry part II:

Matching, robustness, optimization, and applications[J]. IEEE Robotics & Automation Magazine, 2012, 19(2): 78-90.

[3] Nister D. An efficient solution to the five-point relative pose problem[J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2004, 26(6): 756.

[4] Nister D, Naroditsky O, Bergen J. Visual odometry for ground vechicle applications[J]. J of Field Robotics, 2006, 23(1): 3–20.

[5] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]. IEEE Int Conf on Computer Vision. Piscataway: IEEE, 2011: 2320-2327.

[6] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]. IEEE Int Conf on Robotics and Automation. Piscataway: IEEE, 2014: 15-22.

[7] Maimone M, Cheng Y, Matthies L. Two years of visual odometry on the mars exploration rovers[J]. J of Field Robotics, 2007, 24(2): 169–186.

[8] Scaramuzza D, Fraundorfer F. Visual odometry[J]. Robotics & Automation Magazine IEEE, 2011, 18(4): 80-92.

[9] 罗杨宇, 刘宏林.基于光束平差法的双目视觉里程计研究[J].控制与决策, 2016, 31(11): 1936-1944.
(Luo Y Y, Liu H L. Research on binocular vision odometer based on bundle adjustment method[J]. Control and Decision, 2016, 31(11): 1936-1944.)

[10] 叶平, 李自亮, 孙汉旭. 基于立体视觉的球形机器人定位方法[J]. 控制与决策, 2013, 28(4): 631-636.
(Ye P, Li Z L, Sun H X. Stereovision-based localization for ball-shaped robot[J]. Control and Decision, 2013, 28(4): 631-636.)

[11] Mur-Artal R, Montiel J M M, Tardós J D. Orb-slam: A versatile and accurate monocular slam system.[J]. IEEE Trans on Robotics, 2015, 31(5): 1147-1163.

[12] Mouragnon E, Lhuillier M, Dhome M, et al. Real time localization and 3d reconstruction[C]. 2006 IEEE Computer Society Conf on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2006: 363-370.

[13] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]. IEEE and ACM Int Symposium on Mixed and Augmented Reality. Nara: IEEE, 2007: 1-10.

[14] Comport A, Malis E, Rives P. Accurate quadrifocal tracking for robust 3d visual odometry[C]. IEEE Int Conf on Robotics and Automation. Piscataway: IEEE, 2007: 40-45.

[15] Huang T S, Netravali A N. Motion and structure from feature correspondences: A review[J]. Proc of the IEEE, 1994, 82(2): 252-268.

[16] 杨鸿, 钱堃, 戴先中, 等. 基于 Kinect 传感器的移动机器人室内环境三维地图创建[J]. 东南大学学报: 自然科学版, 2013, 43(1): 183-187.
(Yang H, Qian K, Dai X Z, et al. Kinect-based 3D indoor environment map building for mobile robot[J]. J of Southeast University: Natural Science Edition, 2013, 43(1): 183-187.)

[17] Arun K S, Huang T S, Blostein S D. Least-squares fitting of two 3-d point sets[J]. IEEE Trans on Pattern Analysis, 1987, 9(5): 698–700.

[18] Hannah M. Computer matching of areas in stereo images[D]. Stanford: Stanford University, 1974.

[19] Nister D, Naroditsky O, Bergen J. Visual odometry[C]. Computer Vision and Pattern Recognition. Piscataway: IEEE, 2004: 652-659.

[20] Zhang J, Kaess M, Singh S. Real-time depth enhanced monocular odometry[C]. Int Conf on Intelligent Robots and Systems. Piscataway: IEEE, 2014: 4973-4980.

[21] Lim H, Lim J, Kim H J. Real-time 6-DOF monocular visual SLAM in a large-scale environment[C]. IEEE Int Conf on Robotics and Automation. Piscataway: IEEE, 2014: 1532-1539.

[22] 艾青林,余杰,胡克用, 等. 基于ORB关键帧匹配算法的机器人SLAM实现[J]. 机电工程, 2016, 33(5): 513-520.
(Ai Q L, Yu J, Hu K Y, et al. Realization of SLAM based on improved ORB keyframe detection and matching for robot[J]. J of Mechanical & Electrical Engineering, 2016, 33(5): 513-520.)

[23] Strasdat H. Local accuracy and global consistency for efcient visual SLAM[D]. London: Imperial College, 2012.

[24] Kümmerle R, Grisetti G, Strasdat H, et al. g2o: A general framework for graph optimization[C]. IEEE Int Conf on Robotics and Automation. Piscataway: IEEE, 2011: 3607-3613.

[25] Strasdat H, Davison A J, Montiel J M M, et al. Double window optimisation for constant time visual SLAM[C]. Int Conf on Computer Vision. Piscataway: IEEE, 2011: 2352-2359.

[26] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The KITTI dataset[J]. The Int J of Robotics Research, 2013, 32(11): 1231–1237.

[27] Mur-Artal R, Tard'os J D. Fast relocalisation and loop closing in keyframe-based SLAM[C]. IEEE Int Conf on Robotics and Automation. Piscataway: IEEE, 2014: 846-853.

[28] Strasdat H, Montiel J M M, Davison A K. Visual SLAM: Why filter[J]. Image and Vision Computing, 2012, 30(2): 65–77.

(责任编辑：郑晓蕾)