第 58 卷 第 20 期/2021 年 10 月/激光与光电子学进展 研究论文

激光与光电子学进展 先进成像

基于改进双流网络结构的视觉里程计

张海东，徐一鸣*，王栗，卞春磊，周方杰

南通大学电气工程学院，江苏 南通 226019

摘要 由于传统的视觉里程计(VO)存在实现过程繁琐、计算复杂等问题，提出了一种基于改进双流网络结构的VO。所提VO使用双流卷积神经网络结构，能够将RGB图像、深度图像同时馈入模型进行训练，并采用Inception网络结构对卷积层进行改进，减少参数数量。同时，在卷积层中加入注意力机制，提升网络对图像特征的辨识度和系统的鲁棒性。为了评估所提模型，在KITTI数据集上进行了模型的训练与测试，并与VISO2-M、VISO2-S和SfMLearner进行对比。结果表明，相较于同样使用单目相机的VISO2-M和SfMLearner，所提模型在旋转误差和平移误差方面取得了较大的改善，可与使用双目相机的VISO2-S相媲美。

关键词 图像处理；成像系统；视觉里程计；注意力机制；深度学习；双流网络

中图分类号 TP391 **文献标志码** A **doi**：10.3788/LOP202158.2010006

Visual Odometry Based on Improved Dual-Stream Network Structure

Zhang Haidong, Xu Yiming*, Wang Li, Bian Chunlei, Zhou Fangjie

School of Electrical Engineering, Nantong University, Nantong, Jiangsu 226019, *China*

Abstract Because conventional visual odometry (VO) has cumbersome implementation process and complex calculation problems, a VO based on an improved dual-stream network structure is proposed. The proposed VO uses a dual-stream convolutional neural network structure that can simultaneously feed RGB and depth images into the model for training, use the Inception network structure to improve the convolutional layer, and reduce the number of parameters in the convolutional layer. Simultaneously, an attention mechanism is introduced to the convolutional layer to enhance the network's recognition of image features and the system's robustness. After being trained and tested on the KITTI dataset, the proposed improved model is compared with the VISO2-M, VISO2-S, and SfMLearner. The results show that the proposed model's rotation and translation errors are significantly reduced compared with VISO2-M and SfMLearner when using monocular cameras and comparable to VISO2-S when using binocular cameras.

Key words image processing; imaging system; visual odometry; attention mechanism; deep learning; dual-stream network

OCIS codes 100.2960; 110.4155; 200.4260; 330.4150

1 引 言

同时定位与建图(SLAM)[1]作为解决机器人在未知的环境中不断获取位置信息并逐渐构建环境地图问题的方法，在计算机视觉、移动机器人、自动驾驶等领域受到广泛关注[2]。根据传感器的不同，基于SLAM的研究主要分为激光SLAM[3]和视觉SLAM(VSLAM)[4]，激光SLAM的研究起步较早，相关技术趋于成熟[5]。得益于计算机性能的不断提升、计算机视觉领域的不断发展，VSLAM的研究也日趋火热。

收稿日期：2020-11-11；**修回日期**：2020-12-09；**录用日期**：2021-01-02

基金项目：江苏省产学研合作项目(BY2020177)、江苏省高等学校自然科学研究重大项目(18KJA470003)、国家自然科学基金(62103205)

通信作者：*yimingx@ntu.edu.cn

视觉里程计(VO)[6]作为 VSLAM 的前端，主要研究如何通过相机获取到连续图像序列，估计出相机位姿。传统的 VO 主要基于特征匹配[7]的方法，首先通过尺度不变特征变换(SIFT)[8]、加速稳健特征(SURF)[9]、Oriented FAST and Rotated BRIEF(ORB)[10]等算法提取出图像的特征点，然后对特征点进行描述、匹配及外点的剔除[11]，最后求解匹配到的特征点对，得到相机的位姿。许多经典的 VSLAM 系统都应用这种方法实现视觉里程计，例如 Parallel Tracking and Mapping(PTAM)[12]、ORB-SLAM[13-15]等。传统的 VO 通常在位姿估计时有着较高的准确性，但复杂的计算过程、繁琐的实现步骤、不适用于弱纹理区域等缺点都限制其进一步发展。

近年来，随着深度学习的快速发展，许多学者[16-19]将深度学习应用到 VO 的研究中。相较于传统的 VO，基于深度学习的 VO 采用端到端的模式，既避免了大量的几何运行过程，又不对弱纹理区域过于敏感。2015 年，Kendall 等[20]提出 PoseNet，将卷积神经网络(CNN)用于 VO 的研究中，PoseNet 可实时输出 6 自由度的相机位姿。VO 是根据一系列连续的图像序列，不断地估计出相机位姿的过程，呈现出较强的时序性。2017 年，Wang 等[21]注意到 VO 时序性的特点，提出了 DeepVO。DeepVO 基于 CNN 特征提取，采用具有长短时记忆(LSTM)结构的循环神经网络(RNN)组成深度递归卷积神经网络(RCNNs)，在提取图像特征的同时，考虑了 VO 依赖时间序列的问题。2018 年，Li 等[22]在 DeepVO 的基础上，提出无监督深度学习 VO 方案(UnDeepVO)，得益于无监督的网络架构，UnDeepVO 可以使用大量未标注的数据集进行训练，从而减少了标注数据集的工作量。2019 年，Sheng 等[23]考虑到 SLAM 系统中关键帧的选取与 VO 之间相互影响的问题，提出了基于几何和视觉指标的协同优化方案。2020 年，Liu 等[24]注意到深度信息对 VO 位姿预测的重要性，使用双流 CNNs 对深度信息馈入网络进行训练与预测。2021 年，张再腾等[25]将注意力模块嵌入到 CNNs 以增强网络架构的表达力。

本文基于之前的研究，主体网络架构采用双流 CNNs，能够同时提取图像的 RGB 和深度特征，在双流网络中嵌入注意力机制，增强模型提取图像特征的能力。并采用 Inception 结构替换内核较大的卷积层，降低双流 CNNs 带来较大计算量的影响。

2　所提方法

2.1　系统架构

在所提系统中，连续的 RGB 图和深度图同时馈入神经网络，用于网络的训练与测试。系统总体结构如图 1 所示。RGB 流与深度流结合形成双流 CNNs 结构，用于提取图像特征，然后对双流网络提取的特征进行拼接，最后通过全连接层输出相机的位姿信息，即三维平移向量和以欧拉角形式表示的三维旋转向量，如图 1 中实线框部分所示。训练时，使用网络输出的位姿信息与真值来计算损失，计算的损失在优化器作用下用于网络参数的调整，如图 1 中虚线框部分所示。测试时，网络输出的位姿信息与真值或者其他网络输出的结果作比较，进而对所提模型进行评测，如图 1 中点框部分所示。

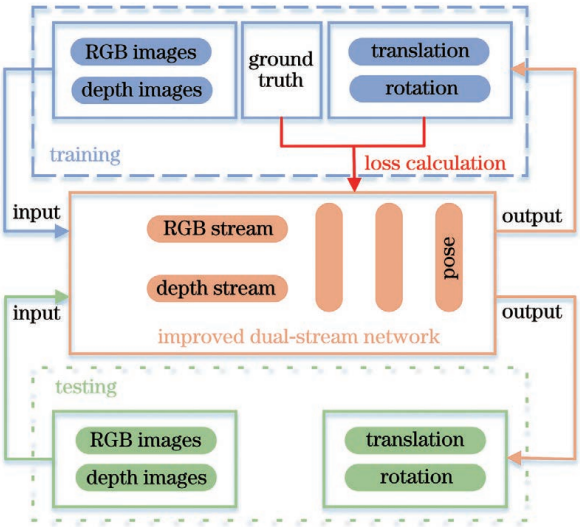

图 1　系统总体结构

Fig. 1　System overall structure

2.2　网络框架

图 2 为所提方法的网络框架，其中 I_k、D_k 分别为第 k 帧 RGB 图和深度图。RGB 图和深度图在输入网络之前需要进行一定的预处理，RGB 图沿 R、G、B 通道分离成 3 通道图像，并与下一帧图像沿通道堆叠成 416×128×6 图像。深度图只有一个通道，为了与 RGB 图在形式上保持一致，复制其通道，并链接下一帧。网络框架中各层网络参数如表 1 所示，网络的主体为一系列的卷积层，前两层 Inception 结构[26]分别用来替代内核为 7×7、5×5 的卷积层，该部分主要用于提取图像中的基本特征，后续 3×3 的卷积层用于提取图像细节。并且在 CNNs 中加入注意力机制，增强系统对特征的辨识能力。

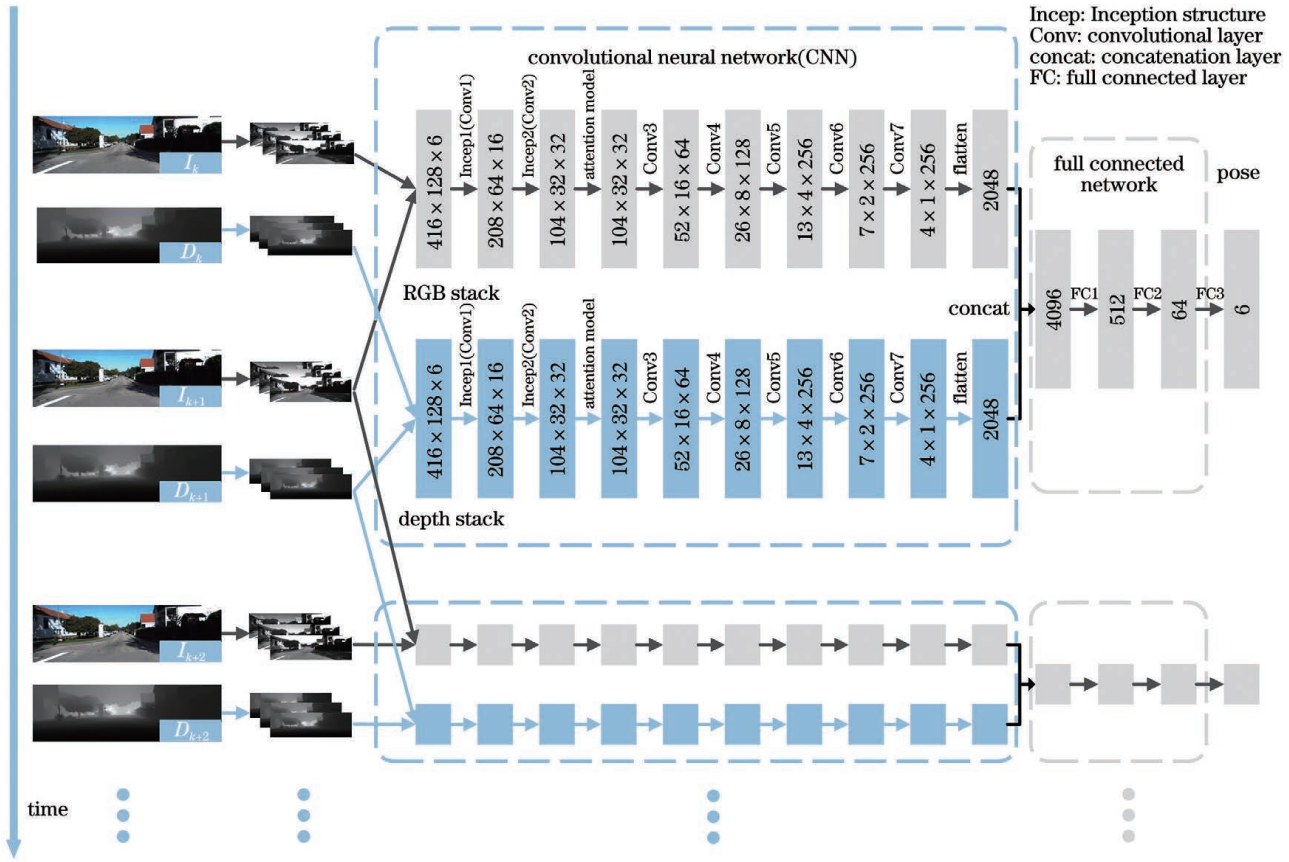

图 2 网络框架

Fig. 2 Network framework

表 1 网络层参数

Table 1 Network layer parameters

Layer	Filter size	Stride	Padding	Output data dimension		
				Width	Height	Depth (number of channels)
Input				416	128	6
Incep1(Conv1)	Multiple (7×7)	2	3	208	64	16
Incep2(Conv2)	Multiple (5×5)	2	2	104	32	32
Attention model	Multiple			104	32	32
Conv3	3×3	2	1	52	16	64
Conv4	3×3	2	1	26	8	128
Conv5	3×3	2	1	13	4	256
Conv6	3×3	2	1	7	2	256
Conv7	3×3	2	1	4	1	512
Flatten						2048
Concat						4096
FC1						512
FC2						64
FC3						6

2.3 网络层

网络采用双流 CNNs 架构，深度流与 RGB 流有着同样的卷积层，这意味着待训练的网络参数量也增加将近一倍，为了降低对网络复杂度和运行速度的影响，选择用 Inception 结构替换 CNNs 中内核较大的卷积层。Inception 结构如图 3 所示。1×1 的卷积核能够调整图像维度，从而达到减少网络参数量的目的；Max pooling 的使用能够剔除一定的冗余信息；3×3、5×5、7×7 的卷积核能够使网络自由地获取图像特征，增加了网络的宽度和增强了尺度适应性。

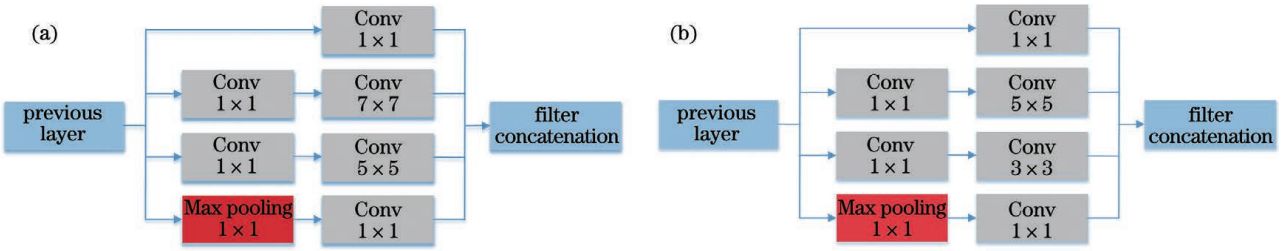

图 3　Inception 结构。(a)Incep1 结构；(b)Incep2 结构

Fig. 3　Inception structure. (a) Incep1 structure; (b) Incep2 structure

Incep1、Incep2 分别替代 CNNs 中 7×7、5×5 的卷积层，随后加入注意力模块，在卷积层中嵌入注意力模块可以提升 CNNs 对图像特征的辨识度和系统的鲁棒性。注意力模块采用的是 Convolutional Block Attention Module（CBAM）结构[27]，如图 4 所示，包括通道注意力模块和空间注意力模块，分别从通道维度和空间维度关注重要特征。

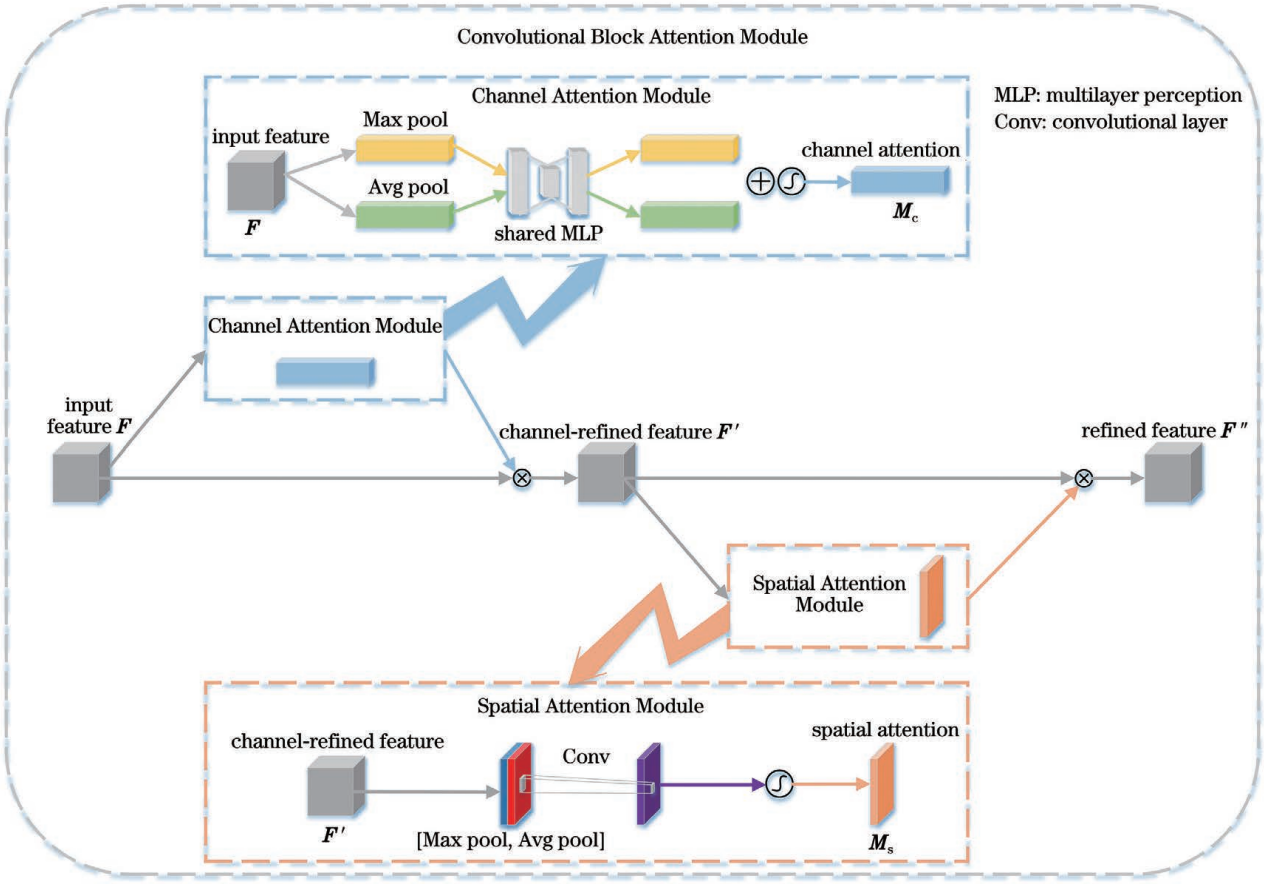

图 4　CBAM 结构图

Fig. 4　CBAM structure diagram

图 4 中，$\boldsymbol{F}$ 表示输入图像特征，经过通道注意力模块输出通道注意力 $\boldsymbol{M}_c$，通道注意力与 $\boldsymbol{F}$ 相乘得到通道优化图像特征 $\boldsymbol{F}'$，整个过程为

$$\boldsymbol{F}'=\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{F})\otimes\boldsymbol{F}=\sigma\{\mathrm{MLP}[\mathrm{Avg\ pool}(\boldsymbol{F})+\mathrm{Max\ pool}(\boldsymbol{F})]\}\otimes\boldsymbol{F},\tag{1}$$

式中：⊗表示逐元素相乘；σ 表示 sigmoid 操作；MLP 表示经过多层感知器输出；Avg pool 和 Max pool 分别表示平均池化操作和最大池化操作。

然后，将通道优化后的图像特征 $\boldsymbol{F}'$ 作为输入，经过空间注意力模块输出空间注意力 $\boldsymbol{M}_{\mathrm{s}}$，进而得到整个注意力模块优化后的图像特征 $\boldsymbol{F}''$，由 $\boldsymbol{F}'$ 得到 $\boldsymbol{F}''$ 的过程为

$$\boldsymbol{F}''=\boldsymbol{M}_{\mathrm{s}}(\boldsymbol{F}')\otimes\boldsymbol{F}'=\sigma\{f^{7\times7}[\mathrm{Avg\ pool}(\boldsymbol{F}'),\mathrm{Max\ pool}(\boldsymbol{F}')]\}\otimes\boldsymbol{F}',\tag{2}$$

式中：$f^{7\times7}$ 表示对图像进行 7×7 卷积操作；$[x,y]$ 表示对 x，y 进行拼接。

2.4　损失函数

神经网络输出一个 6 维向量用于表示位姿，包括旋转与平移两个部分，损失函数采用均方误差，对旋转和平移分别计算。设 $\boldsymbol{\varphi}_{k,k+1}$ 和 $\boldsymbol{t}_{k,k+1}$ 为第 k 帧到第 $k+1$ 帧的旋转欧拉角和平移向量，它们的真值可由 KITTI Visual Odometry 数据集提供的真实轨迹求出，由神经网络输出的旋转欧拉角和平移向量为 $\hat{\boldsymbol{\varphi}}_{k,k+1}$ 和 $\hat{\boldsymbol{t}}_{k,k+1}$，则损失函数为

$$L_{\mathrm{loss}}=\mathrm{argmin}\frac{1}{N}\sum_{k=1}^{N}\left\|\hat{\boldsymbol{t}}_{k,k+1}-\boldsymbol{t}_{k,k+1}\right\|_2^2+\mu\left\|\hat{\boldsymbol{\varphi}}_{k,k+1}-\boldsymbol{\varphi}_{k,k+1}\right\|_2^2,\tag{3}$$

式中：N 表示样本数量；μ 表示平移与旋转关系的尺度因子，用于调节二者在损失函数中所占的权重，根据多次实验调整，选取 $\mu=50$，实验取得效果较好；$\|*\|_2$ 表示 $*$ 的二范数。

3　实　　验

3.1　数据集

KITTI Visual Odometry 是由 Geiger 等[28]驾驶载有相机、激光雷达等传感器的汽车，在城市、乡村道路上采集整理的数据集。两个 RGB 相机和两个灰度相机用来采集双目图像信息，高精度激光雷达捕捉周围距离信息，全球定位系统(GPS)、惯性测量单元(IMU)等用来获取汽车的位姿信息。数据集包含 22 个序列，每个序列提供激光雷达数据、左右相机的 RGB 图和灰度图，其中序列(00～10)提供汽车行驶真实位姿。考虑到数据集中缺少深度图，对 RGB 图和对应时刻的激光雷达数据进行处理，得到相应的深度图。最后将数据集中的所有 RGB 图和深度图的尺寸都调整为 416×128，如图 5 所示，图 5(a)为 RGB 图像，图 5(b)为对应的深度图。

图 5　处理后的数据集。(a)RGB 图；(b)深度图

Fig. 5　Processed data set. (a) RGB image; (b) depth image

同时，由于序列(00～10)提供的真值形式为每帧图像对应的相机位姿，而模型训练需要图像帧间的旋转、平移，需要对真值进行预处理，方法如图 6 所示。

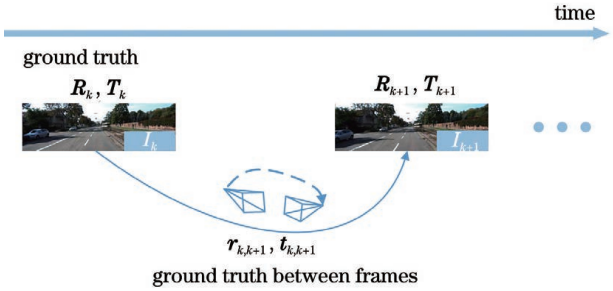

图 6　真值处理

Fig. 6　Ground truth processing

其中 $\boldsymbol{R}_k$、$\boldsymbol{T}_k$、$\boldsymbol{R}_{k+1}$、$\boldsymbol{T}_{k+1}$ 分别为第 k、$k+1$ 帧图像对应相机位姿的旋转信息与平移信息，$\boldsymbol{r}_{k,k+1}$、$\boldsymbol{t}_{k,k+1}$ 为第 k 帧到第 $k+1$ 帧旋转矩阵与平移向量，有

$$\begin{bmatrix}\boldsymbol{r}_{k,k+1} & \boldsymbol{t}_{k,k+1}\\ 0 & 1\end{bmatrix}=\begin{bmatrix}\boldsymbol{R}_k & \boldsymbol{T}_k\\ 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{R}_{k+1} & \boldsymbol{T}_{k+1}\\ 0 & 1\end{bmatrix}。\tag{4}$$

然后，再将旋转矩阵 $\boldsymbol{r}_{k,k+1}$ 表示为欧拉角的形式 $\boldsymbol{\varphi}_{k,k+1}$，设旋转矩阵 $\boldsymbol{r}=\begin{bmatrix}r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{bmatrix}$，欧拉角

表示为 $\boldsymbol{\varphi}=(\theta_x,\theta_y,\theta_z)$，转换公式为

$$\begin{cases}\theta_x=\arctan 2(r_{32},r_{33})\\ \theta_y=\arctan 2(-r_{31},\sqrt{r_{32}^2+r_{33}^2})\\ \theta_z=\arctan 2(r_{21},r_{11})\end{cases},\tag{5}$$

式中：$\arctan 2(r_{32},r_{33})$表示正切值 r_{32}/r_{33} 对应的弧度值。这样就得到 3 维向量表示的旋转真值和 3 维向量表示的平移真值。

3.2　实验环境与参数调整

实验训练与测试是在配备 NVIDIA GeForce RTX 2080Ti GPU 和 Intel 酷睿 i9 9900KF CPU 的台式机上进行的。通过谷歌开源的软件库 Tensorflow2.0 来实现网络模型的搭建，训练时，没有使用批量归一化处理，选择自适应矩估计(Adam)优化器进行参数优化，batch 和学习率分别设为 16 和 0.0001，迭代 100 个周期。同时，为了防止过拟合，使用了 Dropout，并将每次训练随机丢失参数的比率设为 20%。

4　分析与讨论

4.1　数据集实验结果

由于 KITTI Visual Odometry 数据集是在驾驶的汽车上采集的，汽车行驶时在上下方向(相机坐标系中 y 轴)变化较小，为了更好地呈现对比效果，在绘制轨迹图时忽略 y 轴。图 7 展示了序列 00～10 中的部分轨迹图，其中实线是数据集提供的真值，星号标记虚线、加号标记虚线、虚线、点线分别是 SfMLearner[29]、VISO2-M[30]、VISO2-S[31]、所提方法预估的轨迹。

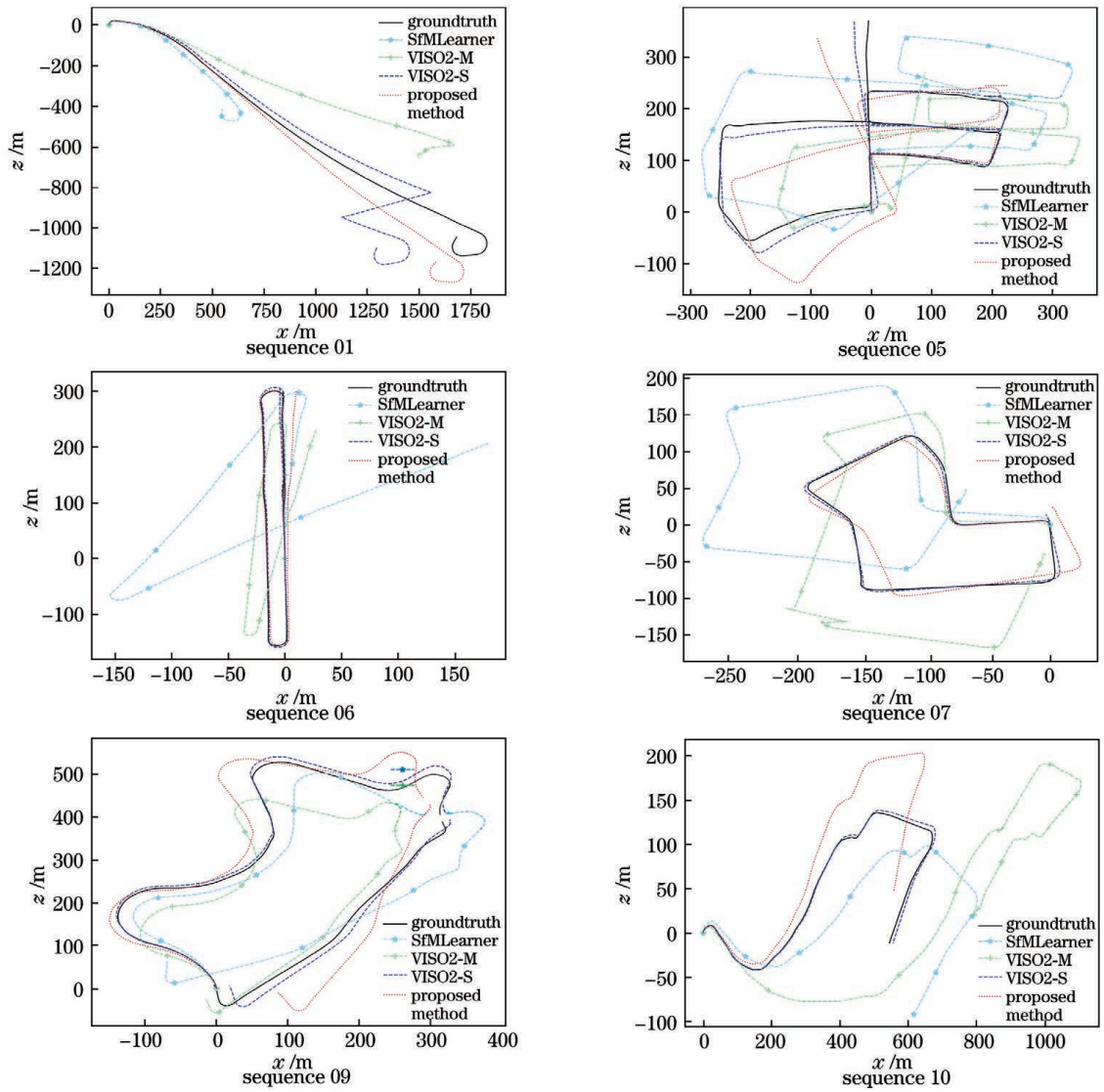

图 7　部分序列 00～10 轨迹图

Fig. 7　Diagram of partly sequence 00–10 trajectory

表 2 为 VISO2-M、SfMLearner、VISO2-S 和所提方法在序列 00～10 上的旋转误差和平移误差，它们都是以均方根误差的形式展现的，并求出了在 10 个序列上的均值。其中 VISO2-S 使用的是双目相机，其余都是单目相机。由于旋转均方根误差 r_{rmse} 相对较小，已经做×100 处理，即旋转均方根误差的单位为 $10^2(°)\cdot m^{-1}$。同时注意到表 2 中 VISO-S 在序列 01 上的数据（加粗标记）存在异常，在序列 01 第 747 帧图像对应坐标为(1558.792，−825.5814)，而第 748 帧图像对应坐标为(1125.423，−947.8420)，对应图 7 序列 01 中的轨迹突变部分。在计算 10 个序列均方根误差均值时，得到平移均方根误差均值 $\bar{t}_{rmse}=4.00$，考虑到异常数据给 $\bar{t}_{rmse}$ 带来较大的影响，使其不能正确表征算法的性能，故在计算时剔除了异常数据。

表 2　VISO-M、SfMLearner、VISO-S 和所提方法在 KITTI Visual Odometry 数据集上测试结果

Table 2　Test results of VISO-M, SfMLearner, VISO-S, and proposed method on the KITTI Visual Odometry data set

Sequence	VISO2-M		SfMLearner		VISO2-S		Proposed method	
	t_{rmse} /%	r_{rmse} / [$10^2(°)\cdot m^{-1}$]	t_{rmse} /%	r_{rmse} / [$10^2(°)\cdot m^{-1}$]	t_{rmse} /%	r_{rmse} / [$10^2(°)\cdot m^{-1}$]	t_{rmse} /%	r_{rmse} / [$10^2(°)\cdot m^{-1}$]
00	12.05	3.67	23.54	6.19	1.86	0.53	5.02	2.11
01	25.55	7.39	59.31	2.79	**23.90**	**0.31**	2.29	0.72
03	16.64	1.34	11.97	4.18	2.10	0.42	3.19	1.24
04	22.86	1.69	21.13	3.28	2.10	0.27	1.68	1.23
05	16.29	4.58	17.14	4.66	1.50	0.50	2.86	1.36
06	11.19	1.99	14.87	5.58	1.54	0.33	1.02	0.50
07	38.07	9.20	20.98	6.31	1.90	0.93	2.97	1.95
08	33.56	2.48	17.41	3.75	1.96	0.60	5.82	2.36
09	13.62	1.72	11.11	4.07	1.85	0.46	3.38	1.26
10	66.11	8.08	23.30	4.06	1.28	0.53	4.92	1.94
Mean	25.59	4.21	22.08	4.49	1.79	0.51	3.32	1.47

4.2　数据集实验结果分析与讨论

通过图 7 所示的轨迹图对比所提模型和 VISO2-S、SfMLearner，可以明显看出：在这几个序列中，所提模型的轨迹（点线）更接近真值；而与 VISO2-S 轨迹（虚线）相比，所提模型的轨迹在直线部分和 VISO2-S 不相上下，在转弯部分稍次之，并且由于没有加入回环检测，误差会不断累积，图 7 中序列 05、07、09 很好地印证了这一点。

对表 2 中的数据进行定量分析，所提模型的轨迹在序列 00～10 上的平移均方根误差的均值较 VISO2-M 下降了 87%，较 SfMLearner 下降了 85%，旋转均方根误差的均值较 VISO2-M 下降了 65%，较 SfMLearner 下降了 67%。与使用双目相机的 VISO2-S 相比，平移均方根误差的均值是其 1.85 倍，旋转均方根误差的均值是其 2.88 倍。

所提模型能够通过输入的 RGB 图和深度图直接输出位姿信息，并且在直线部分能够保持很好的效果。在转弯部分出现的误差有待进一步降低，可能是在定义损失函数时旋转部分与平移部分的配置有问题，不能简单地通过调整两部分的权重来解决问题。

4.3　真实环境实验验证

为了进一步测试所提方法的性能，在真实环境下进行了验证性实验，深度图像信息的采集主要依赖 Intel RealSense D435i 深度相机。该深度相机配有红外激光发射器、红外双目相机，可通过双目视觉与红外结合实现深度测量，测量距离可达 10 m。图 8 为真实环境下的图像采集平台。

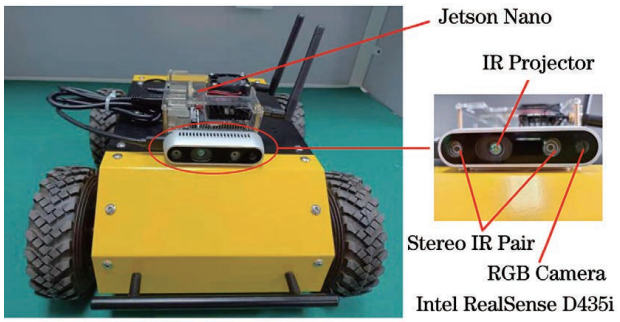

图 8　图像采集平台

Fig. 8　Image acquisition platform

在数据采集过程中，四驱小车底盘驱动整个实验平台按固定轨迹运动，深度相机与 NVIDIA Jetson Nano 嵌入式开发板负责在运动过程中实时采集存储 RGB 图、深度图及左、右相机图像。

由于深度相机的测距范围有限，并且相机的位姿的真值需要高精密仪器才能准确采集，本实验组在室内开展了实验，并使用 VISO2-S 估计的运行轨迹作为参考，实验场景与实验结果如图 9 所示。

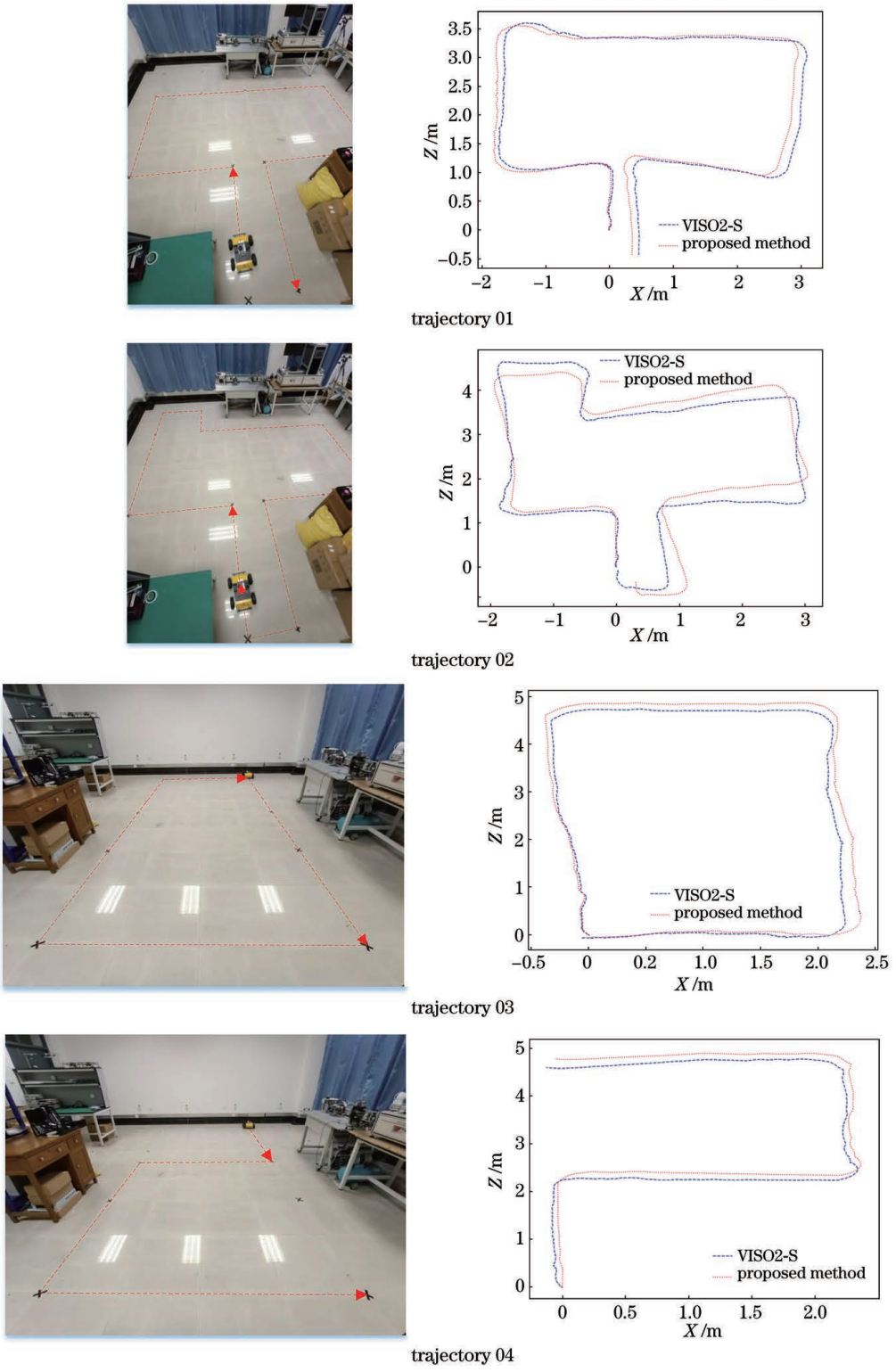

图 9　实验场景与轨迹图

Fig. 9　Experimental scene and trajectory diagram

从图 9 可以看出，所提方法的轨迹与 VISO2-S 运行轨迹基本一致，偏差多出现在转弯部分，且随着时间不断累积，这与分析与讨论部分得出的结果相吻合。

5 结　论

提出了一种基于改进双流网络结构的视觉里程计。双流网络结构的设计能够将深度图像用于模型的训练，使模型能够具有处理深度信息的能力。在 RGB 流和深度流中分别加入注意力机制，加强模型提取特征的能力，同时采用 Inception 结构替换卷积层中较大的卷积核，不仅减少了网络参数量，提高运行速率，而且增加了网络宽度。在 KITTI 数据集上进行测试时，对所提方法与 VISO2-M、SfMLearner、VISO2-S 进行了对比，同时在真实环境中进行了实验验证。结果表明，相较于使用单目的 VISO2-M、SfMLearner，所提模型在平移和旋转方面的性能有了明显的提升，平移性能可与使用双目相机的 VISO2-S 相媲美，旋转性能稍次之。

在接下来的工作中，将充分考虑图像帧间时序性的问题，以所提模型为基础，在双流网络与全连接层之间加入长短时记忆，使模型具有学习图像帧间时序性的能力。同时，通过重新设计损失函数、对表征旋转的特征进行强化等一系列方法，进一步降低旋转误差。

参 考 文 献

[1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE Robotics & Automation Magazine, 2006, 13(2): 99-110.

[2] Nedjah N, de Luiza M M, de Oliveira P J A. Simultaneous localization and mapping using swarm intelligence based methods[J]. Expert Systems with Applications, 2020, 159: 113547.

[3] Hu M C, Ao H R, Jiang H Y. Experimental research on feature extraction of laser SLAM based on artificial landmarks[C]//2019 Chinese Control and Decision Conference (CCDC), June 3-5, 2019, Jiangxi, China. Beijing: Chinese Association of Automation, 2019: 5495-5500.

[4] Bavle H, de la Puente P, How J P, et al. VPS-SLAM: visual planar semantic SLAM for aerial robotic systems[J]. IEEE Access, 2020, 8: 60704-60718.

[5] Lu S D, Tu M Y, Luo X Y, et al. Laser SLAM pose optimization algorithm based on graph optimization theory and GNSS[J]. Laser & Optoelectronics Progress, 2020, 57(8): 081024.
陆世东, 涂美义, 罗小勇, 等. 基于图优化理论和 GNSS 激光 SLAM 位姿优化算法[J]. 激光与光电子学进展, 2020, 57(8): 081024.

[6] Nister D, Naroditsky O, Bergen J. Visual odometry [C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 27-July 2, 2004, Washington, DC, USA. New York: IEEE Press, 2004: 652-659.

[7] Xu G F, Zeng J C, Liu X X. Visual odometer based on optical flow method and feature matching[J]. Laser & Optoelectronics Progress, 2020, 57(20): 201501.
许广富, 曾继超, 刘锡祥. 融合光流法和特征匹配的视觉里程计[J]. 激光与光电子学进展, 2020, 57(20): 201501.

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

[9] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF)[J]. Computer Vision and Image Understanding, 2008, 110(3): 346-359.

[10] 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, 2011: 2564-2571.

[11] Zheng G Q, Zhou Z P. Improved augmented reality registration method based on VSLAM[J]. Laser & Optoelectronics Progress, 2019, 56(6): 061501.
郑国强, 周治平. 一种基于视觉即时定位与地图构建的改进增强现实注册方法[J]. 激光与光电子学进展, 2019, 56(6): 061501.

[12] Klein G, Murray D. Parallel tracking and mapping on a camera phone[C]//2009 8th IEEE International Symposium on Mixed and Augmented Reality, October 19-22, 2009, Orlando, FL, USA. New York: IEEE Press, 2009: 83-86.

[13] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.

[14] Mur-Artal R, Tardós J D. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.

[15] Campos C, Montiel J M M, Tardós J D. Inertial-only optimization for visual-inertial initialization[C]//2020 IEEE International Conference on Robotics and Automation (ICRA), May 31-August 31, 2020,

Paris, France. New York: IEEE Press, 2020: 51-57.

[16] Ummenhofer B, Zhou H Z, Uhrig J, et al. DeMoN: depth and motion network for learning monocular stereo[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 5622-5631.

[17] Mahjourian R, Wicke M, Angelova A. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 5667-5675.

[18] Liu Q, Li R H, Hu H S, et al. Using unsupervised deep learning technique for monocular visual odometry[J]. IEEE Access, 2019, 7: 18076-18088.

[19] Yang X H, Li X J, Guan Y, et al. Overfitting reduction of pose estimation for deep learning visual odometry[J]. China Communications, 2020, 17(6): 196-210.

[20] Kendall A, Grimes M, Cipolla R. PoseNet: a convolutional network for real-time 6-DOF camera relocalization[C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 2938-2946.

[21] Wang S, Clark R, Wen H K, et al. DeepVO: towards end-to-end visual odometry with deep recurrent convolutional neural networks[C]// 2017 IEEE International Conference on Robotics and Automation (ICRA), May 29-June 3, 2017, Singapore. New York: IEEE Press, 2017: 2043-2050.

[22] Li R H, Wang S, Long Z Q, et al. UnDeepVO: monocular visual odometry through unsupervised deep learning[C]//2018 IEEE International Conference on Robotics and Automation (ICRA), May 21-25, 2018, Brisbane, QLD, Australia. New York: IEEE Press, 2018: 7286-7291.

[23] Sheng L, Xu D, Ouyang W L, et al. Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep SLAM[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV), October 27-November 2, 2019, Seoul, Korea (South). New York: IEEE Press, 2019: 4301-4310.

[24] Liu Q, Zhang H D, Xu Y M, et al. Unsupervised deep learning-based RGB-D visual odometry[J]. Applied Sciences, 2020, 10(16): 5426.

[25] Zhang Z T, Zhang R F, Liu Y H. Visual odometry algorithm based on deep learning[J]. Laser & Optoelectronics Progress, 2021, 58(4): 041501.
张再腾, 张荣芬, 刘宇红. 一种基于深度学习的视觉里程计算法[J]. 激光与光电子学进展, 2021, 58(4): 041501.

[26] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions[C]//2015 IEEE Conferenceon Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE Press, 2015: 1-9.

[27] Woo S, Park J, Lee J Y, et al. CBAM: convolutional block attention module[M]//Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11211: 3-19.

[28] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: the KITTI dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.

[29] Zhou T H, Brown M, Snavely N, et al. Unsupervised learning of depth and ego-motion from video[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6612-6619.

[30] Kitt B, Geiger A, Lategahn H. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme[C]// 2010 IEEE Intelligent Vehicles Symposium, June 21-24, 2010, La Jolla, CA, USA. New York: IEEE Press, 2010: 486-492.

[31] Geiger A, Ziegler J, Stiller C. StereoScan: dense 3D reconstruction in real-time[C]//2011 IEEE Intelligent Vehicles Symposium (IV), June 5-9, 2011, Baden-Baden, Germany. New York: IEEE Press, 2011: 963-968.