Journal of System Simulation

Volume 33 | Issue 12 Article 25

1-13-2022

Low Power Visual Odometry Technology Based on Monocular Depth Estimation

Ma Rong
1. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China; ;

Qiurui Chen
1. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China; ;

Zhang Han
1. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China; ;

Mei Zheng
1. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China; ;

See next page for additional authors

Low Power Visual Odometry Technology Based on Monocular Depth Estimation

Abstract

Abstract: With the development of artificial intelligence, precision machinery and computing technology, micro-unmanned system will play an important role in the future battlefield. To solve the lack of monocular visual odometry scale, micro robot power consumption and load limits, *the monocular depth estimation technology is introduced and a low view dataset is collected. A convolutional neural network to predict depth information from a single image is built, and the structure of neural network model is optimized. The depth estimation with monocular visual odometry are combined and deployed on JetsonNano.* Experiments show that the combined monocular visual odometry can recover scale information in a specific environment, and the power consumption on Jetson Nano can be kept a low level, which can provide some research basis for the concealable and lightweight deployment of micro-unmanned system in the future battlefield.

Keywords

monocular visual odometry, monocular depth estimation, convolution neural network, structure optimization

Authors

Ma Rong, Qiurui Chen, Zhang Han, Mei Zheng, Wang Rui, and Wei Wei

Recommended Citation

Ma Rong, Chen Qiurui, Zhang Han, Mei Zheng, Wang Rui, Wei Wei. Low Power Visual Odometry Technology Based on Monocular Depth Estimation[J]. Journal of System Simulation, 2021, 33(12): 3001-3011.

第 33 卷第 12 期 系统仿真学报© Vol. 33 No. 12
2021 年 12 月 Journal of System Simulation Dec. 2021

基于单目深度估计的低功耗视觉里程计

马榕[1]，陈秋瑞[1]，张晗[1]，梅铮[1]，王锐[2]，魏伟[3]
(1. 北京仿真中心 航天系统仿真重点实验室，北京 100854；2. 北京航空航天大学，北京 100191；
3. 空军装备部驻北京地区第四军事代表室，北京 100041)

摘要：随着人工智能、精密机械和计算技术的发展，微小型无人系统在未来战场上将会扮演重要的角色。为解决单目视觉里程计尺度缺失以及微型机器人自身体积和载荷限制带来的功耗问题，*引入单目深度估计技术，构建了一个低视角数据集，通过搭建卷积神经网络从单张图像中预测深度信息，对神经网络模型进行结构优化，将深度估计与单目视觉里程计融合并部署到 JetsonNano 平台。*实验表明，融合后的单目视觉里程计能够在特定环境下恢复尺度信息，在 JetsonNano 上的功耗能够保持在较低水平，可为微型无人系统在未来战场上的隐蔽化、轻量化部署提供一定的研究基础。

关键词：单目视觉里程计；单目深度估计；卷积神经网络；神经网络结构优化
中图分类号：TP391.9 文献标志码：A 文章编号：1004-731X (2021) 12-3001-11
DOI: 10.16182/j.issn1004731x.joss.21-FZ0863

Low Power Visual Odometry Technology Based on Monocular Depth Estimation

Ma Rong[1], *Chen Qiurui*[1], *Zhang Han*[1], *Mei Zheng*[1], *Wang Rui*[2], *Wei Wei*[3]
(1. Science and Technology on Special System Simulation Laboratory, Beijing 100854, China; 2. BeiHang University, Beijing 100191, China;
3. The Fourth Military Representative Office of the Air Force Armament Department in Beijing, Beijing 100041, China)

Abstract: With the development of artificial intelligence, precision machinery and computing technology, micro-unmanned system will play an important role in the future battlefield. To solve the lack of monocular visual odometry scale, micro robot power consumption and load limits, *the monocular depth estimation technology is introduced and a low view dataset is collected. A convolutional neural network to predict depth information from a single image is built, and the structure of neural network model is optimized. The depth estimation with monocular visual odometry are combined and deployed on JetsonNano.* Experiments show that the combined monocular visual odometry can recover scale information in a specific environment, and the power consumption on Jetson Nano can be kept a low level, which can provide some research basis for the concealable and lightweight deployment of micro-unmanned system in the future battlefield.

Keywords: monocular visual odometry; monocular depth estimation; convolution neural network; structure optimization

引言

近年来，精确制导和网络对抗等技术在全球的扩散程度越来越高，分布式无人作战系统成为信息化战争中夺取信息优势、实施精确打击、完成特殊作战任务的重要手段之一，例如美国“黑骑士”无人战车，俄罗斯陆军“库尔干人-25”无人炮塔步兵战车，以及美国“大黄蜂”微型无人侦察机等等。无人系统在到达指定位置的过程中大多依赖里程计为其计算移动位置与距离。对体积与载荷有限的微型无人系统而言，单目视觉里程计是一个更好的选择。

收稿日期：2021-06-09 修回日期：2021-08-25
第一作者：马榕(1995-)，女，硕士，工程师，研究方向为计算机视觉与仿真。E-mail：marong_xidian@163.com
通讯作者：张晗(1978-)，女，硕士，高工，研究方向为高效能计算与仿真。E-mail：xia_mei2000@163.com

然而，单目视觉里程计的一大缺点在于它的尺度不确定性，由单目系统解算的位姿实际只有 6 个自由度，其形成的轨迹也就存在尺度问题。同时，微型机器人的体积与载荷限制也对视觉里程计的计算速度以及功耗提出了更高的要求和挑战。基于深度学习的单目深度估计技术能够通过卷积神经网络对单张图像进行深度预测，从而获取带有真实尺度的深度信息，另外，神经网络结构优化和模型剪枝技术在缩小模型尺寸、降低计算延迟方面有明显的效果。通过将优化后的单目深度估计神经网络模型与单目视觉里程计融合，有望改善单目视觉里程计定位与轨迹跟踪中的尺度问题，同时降低整个系统的功耗，在资源受限平台上达到实时计算或近实时计算。

1 研究现状

弥补单目视觉里程计尺度缺陷的方法主要包括 2 类：第 1 类是使用单目摄像头与其他传感器结合，例如惯性单元、激光雷达等，属于间接提供尺度，需要与单目信息融合，设计复杂，功耗高；第 2 类是使用本身带有深度的传感器，例如红外摄像头、深度摄像头，但没有从根本上解决单目视觉里程计的尺度缺陷问题。

随着深度学习的发展，基于深度学习的单目深度估计技术逐渐兴起，它通过构建神经网络，从单张图像中预测深度信息。它根据是否使用真实深度值训练分为 2 种方法：以真实深度图为监督信号的方法和以双目图像重建一致性为监督信号的方法。前者以 Laina 等[1]提出的将深度残差网络应用于深度估计的方法为代表，具有模型尺寸小、推理速度快的特点。后者以 Godard 等[2]提出的通过几何策略建立双目重建的方法为代表，直接输出结果是视差图，推理过程中需要结合双目相机中的基线参数才能获得真实的深度值，具有模型复杂、泛化性能较好的特点。

单目视觉里程计主要分为 2 个阶段：位姿估计和深度滤波。位姿估计利用帧间像素或特征点位置变换迭代优化，深度滤波利用三角测量估计像素或特征点的空间位置。常用的单目视觉里程计有直接法 DSO(Direct Sparse Odometry, DSO)[3]、基于特征点跟踪的 ORB-SLAM(Oriented FAST and Rotated BRIEF-Simultaneous Location and Mapping)[4]以及半直接法 SVO (Semi-direct Visual Odometry, SVO)[5]等。其中 SVO 使用了高效的特征匹配和鲁棒的深度滤波器，注重跟踪效果，追求高计算速度和低 CPU 占用率，适合在计算资源有限的嵌入式设备上使用。然而，这些单目视觉里程计存在一个严重缺点，它们无法获取环境的真实深度尺度，从而导致系统解算的位姿、轨迹与机器人真实的位姿和轨迹相差一个尺度因子。国内外很多学者对尺度的恢复做了工作[6]，通过增加附加传感器(Inertial Measurement Unit, IMU), GPS 等恢复尺度，但存在融合算法复杂、成本高的问题，且是一种间接提供尺度的方法，没有从根本上解决单目视觉里程计的尺度缺失问题。另外有 CNN-SVO(Convolutional Neural Networks-Semi-direct Visual Odometry, CNN-SVO)[7]使用双目训练场景将神经网络与视觉里程计结合，由于其计算真实尺寸需要数据集的双目基线参数参与，存在相机适应问题，且无法实时运行在嵌入式设备上。

2 总体方案

针对单目视觉里程计的尺度缺失问题，本文采用基于深度学习的单目深度估计技术。通过卷积神经网络提取图像特征，与对应的深度图拟合，从单张图像预测深度信息。针对神经网络结构复杂，计算量大的问题，本文采用神经网络结构优化技术对网络进行轻量化设计，并对模型进行加速。最后，将优化后的单目深度估计模型与单目视觉里程计融合，为里程计提供深度尺度，从而改善单目视觉里程计的尺度缺失问题并加快模型计算速度，降低功耗。整体流程如图 1 所示。

DOI: 10.16182/j.issn1004731x.joss.21-FZ0863

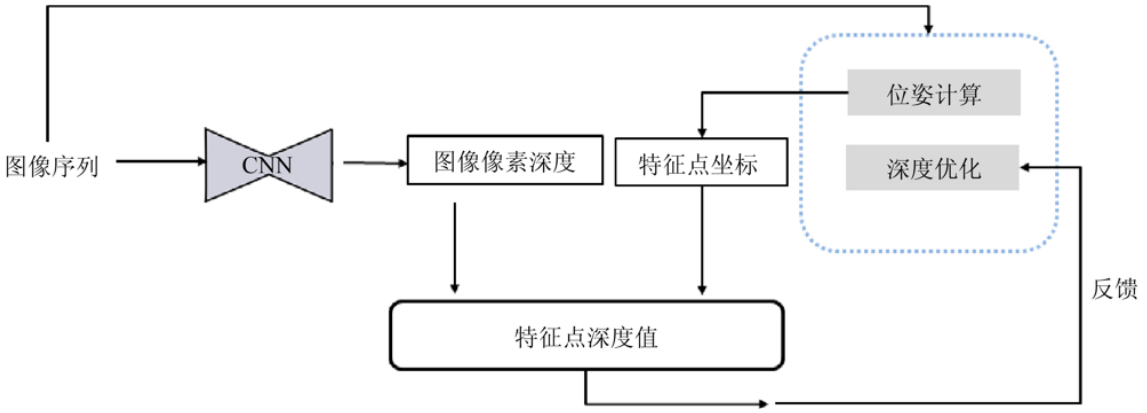

图 1　单目视觉结合深度估计方案流程图
Fig. 1　Flow chart of monocular vision combined depth estimation scheme

首先，通过单目摄像头获取图像序列，输入到单目深度估计神经网络预测图像像素深度，同时将该图像序列输入到单目视觉里程计，进行位姿计算。然后，将位姿计算产生的特征点坐标与深度估计得到的图像像素深度结合，得到特征点的深度值。最后，将特征点的深度值返回给单目视觉里程计的深度优化模块，结合关键帧与非关键帧上的信息进行深度优化，得到更准确的特征点的空间位置。

在这个过程中，有 3 方面的问题需要解决：①单目深度估计神经网络数据集问题，传统的单目深度估计数据集是平视角，不适用于微型机器人的低视角。②单目深度估计神经网络在资源受限设备上的精度与实时性问题，这需要良好的神经网络结构设计和优化作为保证。③单目深度估计与单目视觉里程计的融合过程，如何设计融合逻辑将决定单目视觉系统能否在真实环境中保持稳定工作。

3　数据集与数据处理

考虑到应用场景为微型机器人，它的视角相对于室内通用数据集 NYUDepthV2[8]的视角偏低，所以我们使用 RealsenseD435 深度相机，分辨率为 640×480@30fps，采集了 7 000 多张室内低视角数据集，场景包括办公室、走廊、大厅等，从而使模型更能适应低视角环境。通用数据集与本文采集的数据集如图 2, 3 所示。

图 2　室内通用数据集 NYUDepthV2 (平行视角)
Fig. 2　Indoor universal dataset NYUDepthV2(parallel perspective)

图 3　本文采集的数据集(低视角)
Fig. 3　Our dataset(low perspective)

无论是通用数据集还是本文收集的数据集，都存在深度空洞问题。由于光照变化、物体材质对红外光的吸收等环境因素，造成的深度图有深度值为 0 的“空洞”现象，如图 4 所示。

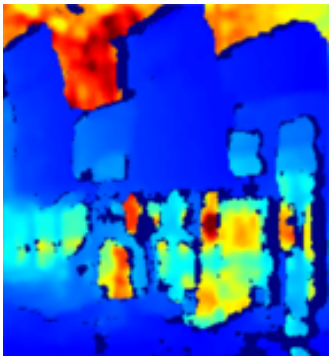

图 4　深度空洞
Fig. 4　Hole in depth

为改善深度空洞对神经网络模型的影响，本文在损失函数中对深度空洞进行遮罩。深度为 0 的空洞位置遮罩为 false，表示不参与损失函数的计算，如式(1)所示。

$$Mask(\cdot)=\begin{cases}\text{true}, & \text{if } depth>0\\ \text{false}, & \text{if } depth\leqslant 0\end{cases} \tag{1}$$

为了增强数据多样性和鲁棒性，对数据集应用了数据增强方法，主要包括光照变化、几何变换、色彩变换等。

4　基于深度学习的单目深度估计

4.1　网络模型结构

结合微型机器人对模型尺寸和推理速度的需求，本文采用以真实深度图作为监督信号的方法。该方法分为 2 个主要部分：编码器和解码器。编码器通过堆叠卷积操作对不同分辨率的特征图进行特征提取。通常的堆叠方法是卷积(conv)与池化(pool)，卷积操作进行特征提取，池化操作则是对特征图进行下采样，不断减小分辨率。与标准卷积和池化不同，本文采用了深度可分离卷积[9]，并以步长为 2 的卷积操作代替池化操作。这样做的好处是能够以学习参数的方式模拟一个池化过程，减少信息丢失。深度可分离卷积由逐通道卷积与逐点卷积组成。逐通道卷积首先对输入的特征图进行 3×3 的标准卷积，然后对结果进行 BatchNorm 归一化操作，并输入激活函数 ReLU；逐点卷积将逐通道卷积的计算结果进行 1×1 标准卷积，归一化，并输入激活函数，如图 5 所示。本文采用 MobileNet[10]网络模型的 pool 之前的 5 个层作为特征提取的骨干网络。

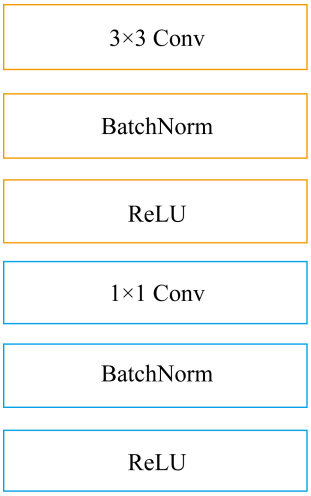

图 5　深度可分离卷积模块组成
Fig. 5　Module composition of depthwise separable convolution

对于解码器部分，本文采用深度可分离卷积结合插值的方式不断恢复图像分辨率。由于编码器通常包含多次下采样操作逐步降低特征图的空间分辨率，来从输入中提取更高层次的特征，这个过程中很多图像细节会丢失，使解码器难以恢复像素级密集的数据。本文采用跳跃连接(skip connection)[11]将编码器中高分辨率的特征图整合到解码器中的相同分辨率的特征图上，有助于解码器利用浅层细节重建更详细密集的输出。具体地，将编码器的第 2, 4, 6 层输出与解码器的第 2, 3, 4 层输出之间对应建立 skip connection，更好的学习浅层特征和深层特征。整体的网络模型结构如图 6 所示。

4.2　多尺度损失函数

通常基于监督的单目深度估计模型将输入图

像预测的深度图与真实深度图对应深度的方差和(SSE)作为损失函数，但是对于像素密集型回归任务来说，这种损失函数的结构过于单一，容易使训练目标陷入局部极小值。本文根据[12]将多尺度损失函数引入像素深度预测中，即网络的总损失是解码器在每一个尺度上的单个损失的组合。

如图 7 所示，首先对解码器产生的低分辨率深度图进行上采样，直至与输入图像的分辨率相同，然后将该深度图与真实深度图做 SSE 损失计算，形成一个尺度上的损失函数 SSE_{map_i}。解码器中共有 5 次上采样过程，由于第一次上采样之前的特征图的分辨率过低，所以不对该层进行损失计算，故总的损失函数是 4 个尺度上的损失函数的加和，如式(2)所示：

$$L = SSE_{\text{map}_0} + SSE_{\text{map}_1} + SSE_{\text{map}_2} + SSE_{\text{map}_3} \quad (2)$$

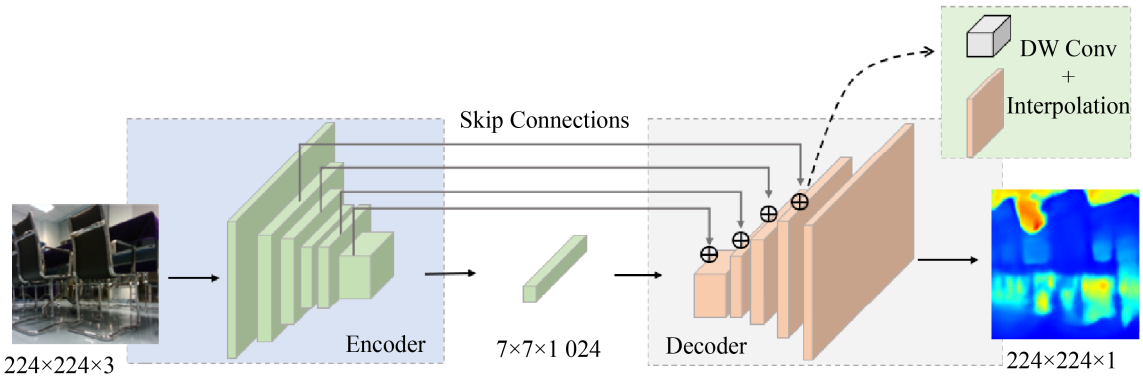

图 6 单目深度估计网络模型整体结构
Fig. 6 Structure of monocular depth estimation network

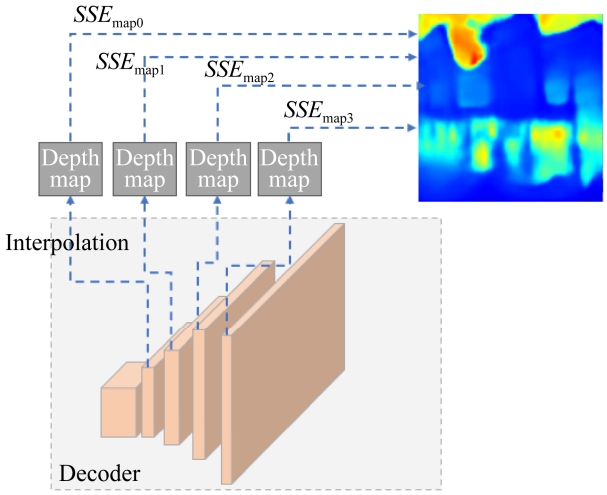

图 7 多尺度损失函数
Fig. 7 Multi-scale loss

4.3 神经网络模型优化

神经网络模型训练完毕，通常存在很多冗余参数。为了进一步降低网络推理延迟，本文借鉴滤波器剪枝技术 Netadapt[13]的思想。对于一个预先训练过的网络，为其设定时延资源限制，通过对每层滤波器维度删减压缩，从而达到设定值的约束。不同于文献[13]，本文修改每层滤波器的剪枝标准，使用 L_1 规范对滤波器排序，从最小值开始剪，直到满足该层资源预算。当所有层均满足资源预算，并且总资源也满足预算要求时，完成剪枝。

算法具体流程如下。

算法 1：滤波器剪枝

输入：预训练过的网络模型 Net_0、资源预算 budget、资源减少预期ΔR_i

输出：满足资源约束的网络模型 $Net^{\wedge}$

i=0;

$Res_i = \text{MeasureOnDevice}(Net_i)$

while $Res_i > \text{budget}$ do

$Con_i = Res_i - \Delta R_i$;

for k from 1 to K do

$N_{\text{filt}_k}, Res_{\text{simp}_k} = \text{ChooseNumFilters}(Net_i, k, Con_i)$

$Net_{\text{simp}_k} = \text{ChooseWhichFilters}(Net_i, k, N_{\text{filt}_k})$

$Net_{\text{simp}_k} = \text{ShortTrain}(Net_{\text{simp}_k})$

$Net_{i+1}, Res_{i+1} = \text{ChooseHighestAcc}(Net_{\text{simp}_k}, Res_{\text{simp}_k})$

$i = i + 1$;

$Net^{\wedge} = \text{LongTrain}(Net_i)$

return $Net^{\wedge}$

算法 1 的步骤为：

step 1：ChooseNumFilters 过程，按照给定资源约束决定某一层需要减掉的滤波器数量。这一步使用渐进的方法，逐渐减少目标层中的滤波器数量，并测量剪除这些滤波器之后的资源变化量。当剪除的数量是能够满足当前资源约束的最大滤波器数量时，就将该层的剪枝模型作为备选。另外，当从一个层中剪除一些滤波器后，对应的下一层中的输入通道数也应该剪除相应的数量。因此，每次对一层的滤波器进行操作后，还需要考虑其他层的资源消耗情况。

step 2：ChooseWhichFilters 过程，确定剪枝标准。根据第一步中剪除的数量与剪枝标准确定保留哪些滤波器。为了保持算法的简单和剪枝过程的快速，本论文选择使用 L_1 剪枝标准。首先根据滤波器的 L_1 范数对滤波器排序，从最小的开始剪除，直到剪除数量达到第一步中确定的数量为止。

step 3：ShortTrain 与 LongTrain 过程，恢复精度。在每一层剪枝过程中，精度通常随着减除滤波器数量的增加而下降，此时需要重训练来恢复精度。具体地，在每一层剪除完毕需要进行短期重训练(short-train)，网络模型全部剪除完毕后进行长期重训练(long-train)，这一步在适应资源减少较多的小网络时尤为重要，否则精度会下降很大，从而导致算法选择错误的网络模型方案。

5 单目深度估计与视觉里程计融合

5.1 SVO

本文采用介于直接法与特征点法之间的方法—半直接法 SVO 作为基础单目视觉里程计。SVO 通过对图像的特征块进行直接匹配来获取相机位姿，而不是像直接法对整个图像使用像素匹配，在速度上有一定提升。

SVO 使用 2 个并行线程，1 个用于位姿计算的跟踪线程(tracking)，1 个用于深度滤波与特征点映射的建图线程(mapping)。建图线程与跟踪线程相互依赖，建图线程以相机位姿已知为前提，而跟踪线程又依靠深度滤波的结果优化位姿估计。跟踪线程主要的作用是位姿计算。在跟踪线程中，通过最小化参考图像块和新帧中以重投影位置为中心的图像块之间的光度残差来获得新帧的相机位姿。

如图 8 所示，跟踪线程包括 3 部分：稀疏图像对齐、特征配准、位姿与结构优化。

稀疏图像对齐主要对稀疏的特征块使用直接法配准，获取粗略的相机位姿，如图 9 所示。

投影方程表示为

$$\boldsymbol{u}_{\mathrm{C}} = f(\boldsymbol{P}_{\mathrm{C}}) \tag{3}$$

式中：空间点 P 在相机坐标系下的坐标为 $\boldsymbol{P}_{\mathrm{C}}$；$\boldsymbol{u}_{\mathrm{C}}$ 为归一化平面上特征点的投影周围的特征块。设 $\boldsymbol{u}$ 为该帧上特征块的集合，那么第 i 帧与第 i+1 帧的光度残差为：

$$\delta I(\boldsymbol{T},\boldsymbol{u}) = I_{i+1}(f(\boldsymbol{T}\cdot f^{-1}(\boldsymbol{u},\boldsymbol{Z}_u))) - I_i(\boldsymbol{u}) \tag{4}$$

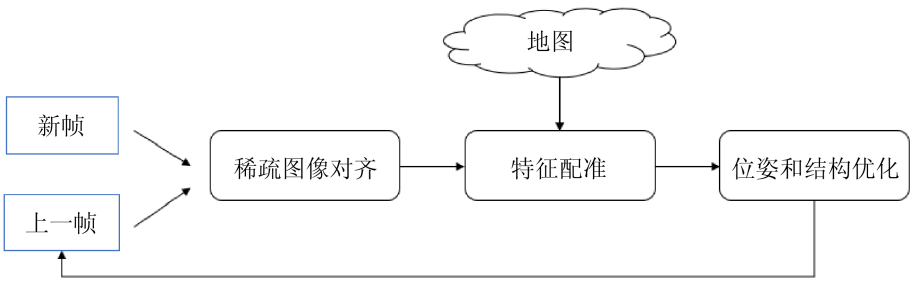

图 8 跟踪线程的 3 个阶段

Fig. 8 Three phases of tracking thread

DOI: 10.16182/j.issn1004731x.joss.21-FZ0863

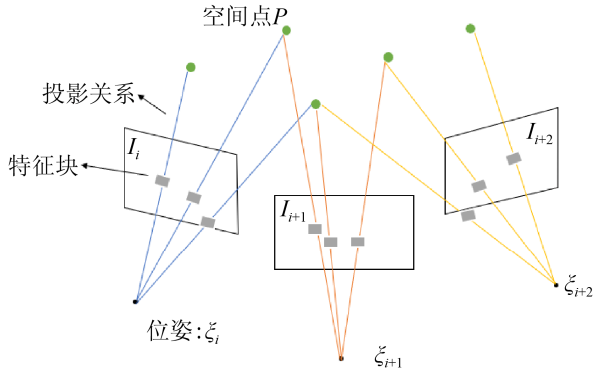

图 9 跟踪过程示意图
Fig. 9 Schematic diagram of tracking thread

图 9 中，设 $\boldsymbol{\xi}_i$ 为第 i 帧 I_i 的相对于世界坐标系的位姿；$\boldsymbol{T}_{i+1,i}$ 为 I_i 到 I_{i+1} 的位姿变换。通过最小化光度残差求得第 i 帧与第 i+1 帧的位姿变换为

$$\boldsymbol{T}_{i+1,i}=\arg\min_{\boldsymbol{T}_{i+1,i}}\frac{1}{2}\sum_{j\in R}\left\|\delta I(\boldsymbol{T}_{i+1,i},\boldsymbol{u}_j)\right\|^2 \tag{5}$$

由于估计的位姿转换矩阵误差导致投影块 $\boldsymbol{u}$ 在 I_{i+1} 帧中的位置不是真正的特征块所处的位置，所以需要进一步特征配准来找到地图中特征块 $\boldsymbol{u}$ 在 I_{i+1} 帧中应该出现的位置，根据 I_{i+1} 帧中对应位置的误差继续优化。根据光度不变性假设，特征块在以前参考帧 I_i 中的亮度应该和 I_{i+1} 帧中的亮度相似。所以重新构造残差，对特征块的位置 $\boldsymbol{u}$ 进行优化：

$$\boldsymbol{u}_j^{'}=\arg\min_{\boldsymbol{u}_j^{'}}\frac{1}{2}\left\|I_{i+1}(\boldsymbol{u}_j^{'})-\boldsymbol{A}_j\cdot I_i(\boldsymbol{u}_j)\right\|^2 \tag{6}$$

式中：$\boldsymbol{A}_j$ 为仿射矩阵；由于参考帧 I_i 在特征配准过程中不一定是上一帧，也可能是距离当前帧较远的关键帧，所以特征块的形状可能发生变化，使用仿射矩阵减少因形状变化带来的误差干扰。

由于第 2 步优化后的特征位置和第 1 步预测的特征位置存在差异，故继续构造新的优化目标函数，优化变量为相机位姿 $\boldsymbol{T}_{i+1,\mathrm{W}}$ 和 P 在世界中坐标系的坐标 $\boldsymbol{P}_{\mathrm{W}j}$：

$$\boldsymbol{T}_{i+1,\mathrm{W}}=\arg\min_{\boldsymbol{T}_{i+1,\mathrm{W}}}\frac{1}{2}\sum_{j\in R}\left\|\boldsymbol{u}_j-f(\boldsymbol{T}_{i+1,\mathrm{W}},\boldsymbol{P}_{\mathrm{W}j})\right\|^2 \tag{7}$$

建图线程的主要作用是将深度收敛的点反投影到地图中，维护一个局部稀疏地图。对于建图线程，给定一关键帧图像和它对应的位姿 $\{I_k,\boldsymbol{T}_{k,\mathrm{W}}\}$，建图线程估计三维点对应的未知的二维特征点的深度。特征的深度是用概率分布建模的，后续每个 $\{I_{k+1},\boldsymbol{T}_{k+1,\mathrm{W}}\}$ 观测都将被用来更新贝叶斯模型的分布，也被称为“深度滤波”。当模型深度分布的方差足够小的时候，说明深度收敛，可以将该点转化为三维点并插入地图中，立即参与运动估计。每一个深度滤波器都与一个参考关键帧相关联，对于新来的一帧，如果该帧为关键帧，则对关键帧提取特征点，作为种子点放进种子队列，使用当前帧的平均场景深度初始化深度滤波器；对于非关键帧，使用其上的深度信息更新所有种子点的深度概率分布。

5.2 单目深度估计与 SVO 融合

首先，在 SVO 初始化的过程中增加更新尺度操作。SVO 进行三角化与归一化的同时，对关键帧进行深度估计，得到真实尺度，并使用该尺度更新归一化时的尺度。然后，通过建图线程继续创建后续点。建图线程通过 2 个过程创建新的地图点：初始化新地图点和使用深度过滤器更新地图点的深度不确定性。由 5.1 节可知，SVO 的做法是以当前帧的平均场景深度作为深度滤波器种子点的均值。由于本身环境深度就是一个归一化后的状态，没有具体尺度。根据文献[7]对于深度初值与滤波器的理解，本文采用单目深度估计为深度滤波器提供带有尺度信息的深度图，并且只对关键帧上的特征点估计深度，对于非关键帧，仍使用 SVO 深度滤波器的过程，不对其进行深度估计，直接使用非关键帧的信息更新深度滤波器。流程如图 10 所示。

当特征点的深度收敛，即它的不确定性低于阈值，就把它插入地图中，用于跟踪。假设参考帧 I_i 中特征块的位置已知，它的深度值在[$Z_{\min}$, $Z_{\max}$]之间，根据这 2 个端点深度值对应到当前帧中即是一条极线段(图 11 I_{i+1} 帧中蓝色位置)。在极线上进行特征搜索与匹配，如果极线段很短(小于 2 个像素)，直接使用特征配准中得到的特征块位置；如果极限段很长，首先在极线上分段采样，对采样的多个特

征块与参考帧中的特征块匹配并评分，选取得分最高的特征块，然后使用特征配准方法再次精确该特征点的位置，最后使用三角化计算深度。由于多帧存在共视点，可以不断优化 3D 空间点的深度并减弱外点的干扰(极线搜索匹配错误的情况)，采用贝叶斯概率模型对深度值进行更新。

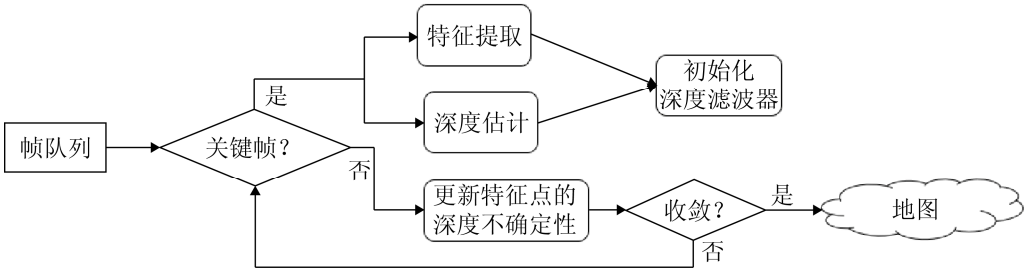

图 10　深度估计融合视觉里程计流程示意图
Fig. 10　Flow diagram of depth estimation fusion visual odometry

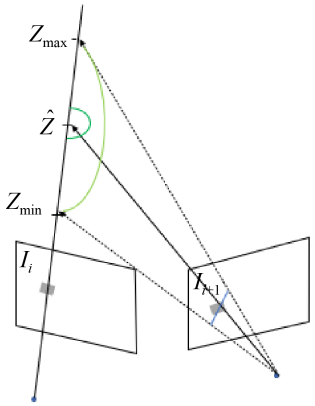

图 11　极线搜索区域
Fig. 11　Polar line search area

具体地，使用高斯分布和均匀分布联合建模深度滤波器：

$$p(x\mid Z,\pi)=$$
$$\pi N(x\mid Z,r^2)+(1-\pi)U(x\mid Z_{\min},Z_{\max}) \tag{8}$$

式中：π 表示 x 是有效测量值的概率；r^2 为逆深度的方差。

对同一个种子点的一系列测量值$\{x_1,x_2,\cdots,x_n\}$，假设这些测量值是独立的，那么求 Z,π 的最直观的方法是最大似然估计。但是根据文献[14]可知，最大似然估计容易被局部极大值干扰，所以采用最大后验概率求解，并用高斯 Beta 分布来近似：

$$q(Z,\pi\mid a_n,b_n,u_n,\sigma_n)=$$
$$\mathrm{Beta}(\pi\mid a_n,b_n)\cdot N(Z\mid u_n,\sigma_n^2) \tag{9}$$

式中：a_n, b_n 为 Beta 分布中的参数；u_n, σ_n^2 为高斯分布中的均值和方差。对于 a_n，b_n 的增量贝叶斯更新使用文献[14]中的方法，更新迭代公式为

$$q(Z,\pi\mid a_n,b_n,u_n,\sigma_n^2)\approx$$
$$p(x_n\mid Z,\pi)\cdot q(Z,\pi\mid a_{n-1},b_{n-1},u_{n-1},\sigma_{n-1}^2) \tag{10}$$

一旦 σ_n^2 低于阈值，则表示该深度值已收敛。

6　实验验证与分析

6.1　室内单目深度估计算法的测试

本文使用 pytorch 框架搭建神经网络模型，数据集为自构建的低视角数据集，训练样本 5 700 张，验证和测试样本 1 425 张。模型训练平台为 NVIDIA GTX1080Ti，测试平台为 NVIDIA Jetson TX2 与 Jetson Nano。预训练模型采用 NYUDepthV2 数据集训练过的 Mobilenet 网络，抽取其中特征提取部分的权重作为编码器初始化权重，解码器权重使用 Kaiming 初始化，训练轮次 40，梯度下降采用 SGD，动量 0.9，权重衰减 0.001，初始学习率为 0.01。精度评估采用相对误差在 25%以内的预测像素的百分比 δ_1 与 RMSE 均方根误差。测试效果如图 12 所示，预测效果基本符合深度分布。为了对比不同编码器对整个网络模型效果的影响，本文也对 ResNet50 特征提取作为编码器进行了测试，模型在测试集上的测试精度与均方误差如表 1 所示，实验表明使用 MobileNet 作为编码器可以获得更好的精度。模型推理整体性能如表 2 所示。

从表 2 可以看出，TX2 GPU 平台上单目深度

DOI: 10.16182/j.issn1004731x.joss.21-FZ0863

估计推理速度完全超实时，在 Nano GPU 平台上，推理速度接近实时，功耗能控制在较低的水平。

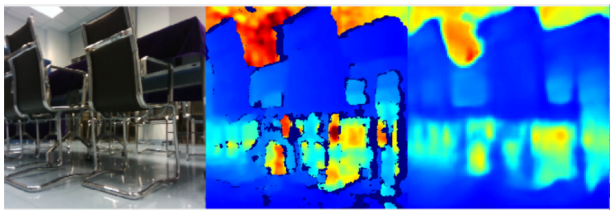

图 12 原图–真实深度图–预测深度图
Fig. 12 Original-Truth-Predicted

表 1 模型精度与方差
Tab. 1 Model accuracy and variance

网络结构	δ_1	*RMSE*
MobileNet+解码器	0.720	0.752
Resnet50+解码器	0.704	0.760

表 2 单目深度估计模型在 TX2 和 Nano 上的性能对比
Tab. 2 Performance comparison of monocular depth estimation model on TX2 and Nano

设备模式	推理时间/ms	最大帧速/fps	平均功率/W
TX2 GPU	6.68	159.0	8.360
TX2 CPU	36.17	27.6	5.870
Nano GPU	50.07	20.0	5.090
Nano CPU	133.40	7.5	2.989

6.2 融合后的单目视觉里程计的测试

测试平台为搭载了 Jetson Nano 的模型车(包括控制板)，传感器为单目摄像头，分辨率为 640×480@30fps，如图 13 所示。测试场景为室内，摄像头视角类似于自构建数据集的低视角，有利于得到更高的深度估计精度，如图 14 所示。

由于显示设备需要与模型车通过网线互联，不利于小车的运动，所以本文首先在地面为运动路径做好标记，然后使用笔记本电脑控制模型车摄像头录制不同运动视频，最后将视频放到 Nvidia Jetson Nano 上计算。如图 15 所示。最后使用 evo 工具计算轨迹信息。

本文设置 2 条真实运动轨迹，分别是直线和折线，实验结果如表 3 所示。从轨迹的真实距离与算法估计的距离来看，本文算法能够成功进行轨迹跟踪与定位，且误差控制在极小的范围内。

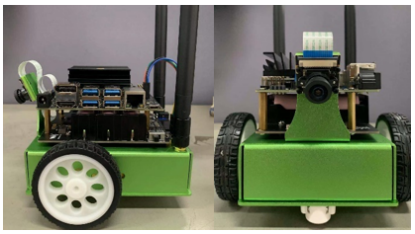

图 13 测试平台图
Fig. 13 Test platform

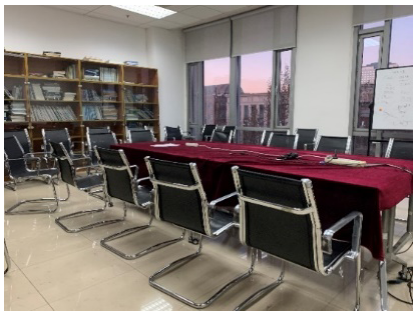

图 14 测试环境图
Fig. 14 Test environment

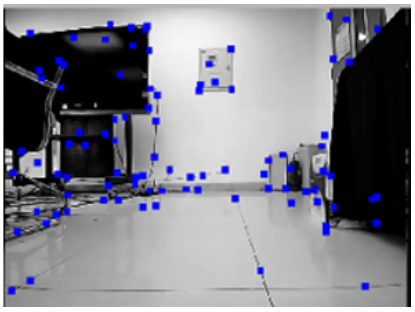

图 15 测试过程中回传的图像
Fig. 15 Images returned during test

表 3 本文算法在不同轨迹下的测试结果
Tab. 3 Test results on different trajectories

轨迹	真实距离/m	测试距离/m	误差/%
直线	4.50	4.38	2.60
折线	5.89	5.96	1.12

将本文算法分别与 SVO, SVO2.0[15] (无 IMU), ORB_SLAM2[16]单目对比，实验环境为真实室内环境，实验平台均为 Jetson Nano。实验结果如图 16, 17 所示。可以看到本文算法能够完整跟踪模型车运动，而 SVO2.0(无 IMU)算法在初始化时出现跟踪丢失的情况(红色标注部分)，ORB_SALM2 单目初

始化失败，未完成轨迹跟踪。轨迹对比图如图 18 所示，绿色表示 SVO2.0 的轨迹，蓝色表示本文算法的轨迹，虚线表示真实轨迹。轨迹长度与误差如表 4 所示。

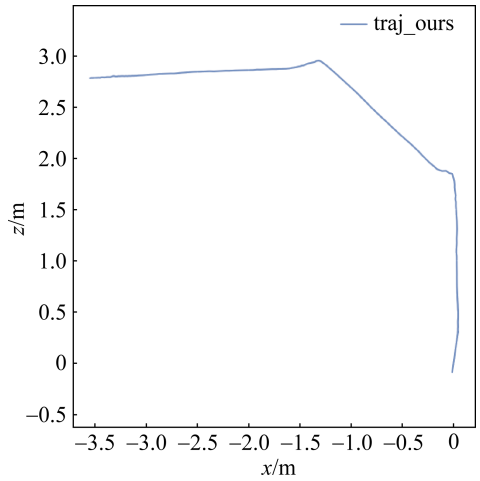

图 16 基于单目深度估计的视觉里程计轨迹结果(ours)
Fig. 16 Results of visual odometry based on monocular depths estimation

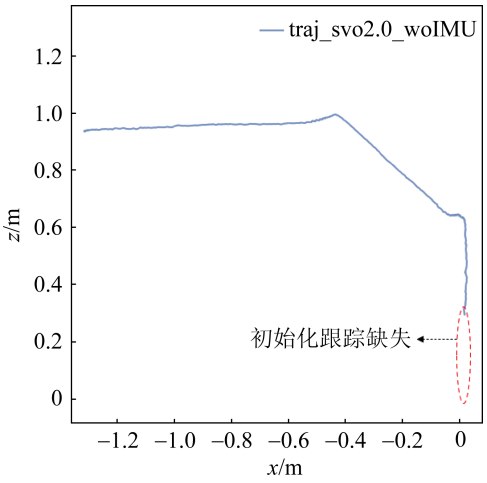

图 17 SVO2.0 (w/o IMU)轨迹结果
Fig. 17 Results of SVO2.0(w/o IMU)

从表 4 可以看出，SVO2.0 在无 IMU 参与的情况下，计算的轨迹是没有尺度的。另外，SVO 在初始化时匹配的特征点过少，导致跟踪丢失，未完成整个轨迹的运行。ORB_SLAM2 单目在初始化时虽然能够匹配到足够的特征点，但这些特征点质量不好，误匹配的情况广泛存在，所以无法跟踪完成整个轨迹。而本文算法使用了单目深度估计，相当于为单目直接提供了深度信息，能够利用该信息更精确地确定待匹配的特征点的 3D 位置，使深度滤波器的迭代不发散，能匹配到足够的特征点，并保持跟踪。

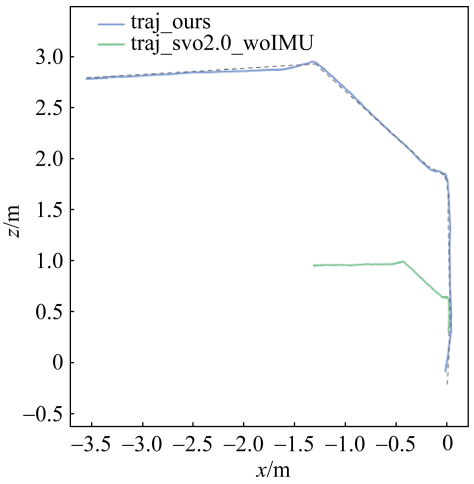

图 18 轨迹估计对比
Fig. 18 Trajectory estimation correlation

表 4 几种单目视觉里程计的轨迹跟踪对比(室内)
Tab. 4 Trajectory tracking comparison of several monocular odometers(indoor)

算法	真实距离/m	测试距离/m	误差/%	平均功率/W
SVO	5.89	0.67 (lost)		
SVO2.0 (w/oIMU)	5.89	2.011	68	4.2
ORB_SLAM2 单目	5.89	1.05 (lost)		
Ours	**5.89**	**5.96**	**1.19**	**4.6**

7 结论

本文针对微小型机器人的自身体积和载荷对视觉里程计传感器以及功耗的限制问题，引入了基于神经网络的单目深度估计技术，通过搭建卷积神经网络从单张图像中预测深度信息，并对神经网络模型进行结构优化以加快计算速度，降低功耗。最后将深度估计与单目视觉里程计融合，部署到 Jetson Nano 设备上，实现了一种可用于微型机器人的具有深度尺度的低功耗单目视觉里程计。

参考文献：

[1] Laina I, Rupprecht C, Belagiannis V, et al. Deeper Depth

Prediction with Fully Convolutional Residual Networks[C]// Fourth International Conference on 3d Vision. Piscataway, NY: IEEE, 2016.

[2] Godard C, Mac A O, Brostow G. Unsupervised Monocular Depth Estimation with Left-right Consistency[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NY: IEEE, 2016.

[3] Engel J, Koltun V, Cremers D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence (S1939-3539), 2018, 40(3): 611-625.

[4] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics (S1941-0468), 2017, 31(5): 1147-1163.

[5] Christian F, Matia P, Davide S. SVO: Fast Semi-direct Monocular Visual Odometry[C]// IEEE International Conference on Robotics and Automation. Piscataway, NY: IEEE, 2014.

[6] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based Visual–inertial Odometry Using Nonlinear Optimization[J]. International Journal of Robotics Research (S0278-3649), 2014, 34(3): 314-334.

[7] Loo S Y, Amiri A J, Mashohor S, et al. CNN-SVO: Improving the Mapping in Semi-Direct Visual Odometry Using Single-Image Depth Prediction[C]// 2019 International Conference on Robotics and Automation (ICRA). Piscataway, NY: IEEE, 2019.

[8] Silberman N, Hoiem D, Kohli P, et al. R.: Indoor Segmentation and Support Inference from RGBD Images[C]// Berlin, German: ECCV, 2012.

[9] Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NY: IEEE, 2017.

[10] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. arXiv:1704.04861, 2017.

[11] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NY: IEEE, 2016.

[12] Jin H, Han M, Dong W, et al. From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation[C]// IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NY: IEEE, 2019.

[13] Yang T J, Howard A, Chen B, et al. NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications[C]// Computer Vision – ECCV 2018. Munich, Germany: ECCV, 2018.

[14] George V A, Carlos H B. Video-based, Real-time Multi-view Stereo[J]. Image & Vision Computing (S0262-8856), 2011, 29(7): 434-441.

[15] Forster C, Zhang Z, Gassner M, et al. SVO: Semi-direct Visual Odometry for Monocular and Muticameras Systems[J]. IEEE Transactions on Robotics (S1941-0468), 2017, 33(2): 249-265.

[16] Mur-Artal R, Juan Tardós D. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras[C]// Piscataway, NY: IEEE Trans. Robot, 2017.