PV-SLAM: Panoptic Visual SLAM with Loop Closure and Online Bundle Adjustment

Ashok Bandyopadhyay[1,2]
ban_ashok@vssc.gov.in,bashok@iitg.ac.in
Pranjal Baranwal[2]
pranjal.baranwal@iitg.ac.in
Arijit Sur[2]
arijit@iitg.ac.in
Rajeev UP[1]
up_rajeev@vssc.gov.in

[1] Vikram Sarabhai Space Centre, Indian Space Research Organisation, Thiruvananthapuram, Kerala, India

[2] Indian Institute of Technology Guwahati, Guwahati, Assam, India

Abstract

Visual Simultaneous Localization and Mapping (vSLAM), a variant of SLAM systems, is a navigation technique in which autonomous robots leverage visual data from camera to create a map of an unknown environment while concurrently determining their own position on that map. Panoptic segmentation-based Visual SLAM offers a more comprehensive, efficient, and robust solution for robot perception and scene understanding compared to approaches that rely solely on semantic or geometric information. In Visual SLAM, the panoptic segmentation simultaneously identifies and delineates all objects in an image, providing a detailed understanding of both their instances and semantic categories. It offers the potential to improve the robustness and accuracy of SLAM systems in complex and dynamic environments. Recent Panoptic segmentation based works on Visual SLAM ignore the problem of loop closure. Also, these methods use offline bundle adjustment, which can lead to drift errors, highlighting the need for online bundle adjustment. In this paper, we introduce a novel architecture that integrates loop closure and online bundle adjustment, expanding on the PVO model[22]. The results show that our model outperforms state-of-the-art methods in visual odometry tasks. The ablation study shows that our technique outperforms current state-of-the-art methods, exhibiting superior performance across majority of sequences.

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental capability of mobile robots exploring unknown environments without GPS, enabling them to map the surroundings and localize within this map concurrently. It can be categorized into different types based on the sensor used [4], such as visual SLAM (using cameras), LiDAR SLAM (using LiDAR sensors), and RGB-D SLAM (using depth cameras). Visual SLAM is advantageous due to its cost-effectiveness, ability to provide rich environmental information, and versatility in various environments and lighting conditions compared to other SLAM methods.

Visual SLAM techniques vary based on sensor type (monocular, stereo, RGB-D), feature extraction (feature-based, direct), optimization (keyframe-based, dense), and learning-based approaches, each offering unique advantages and disadvantages in terms of cost, accuracy, computational complexity, and adaptability. Monocular Visual SLAM is cost-effective but less accurate, Stereo SLAM provides depth information but requires calibration, RGB-D SLAM combines color and depth but has limited range, Feature-based SLAM is efficient but struggles in textureless areas, Direct SLAM offers dense reconstruction but is computationally intensive, and Learning-based SLAM adapts but requires significant data and resources. Panoptic segmentation is an advanced computer vision technique that simultaneously identifies and delineates all objects in an image, providing a detailed understanding of both their individual instances and semantic categories. It offers the potential to improve the robustness and accuracy of SLAM systems in complex and dynamic environments. Among the recent researches, Competitive Collaboration[14] is an unsupervised network framework. DROID-SLAM[16] uses feature and context encoders similar to RAFT[15] to construct a frame graph and applies dense bundle adjustment for drift error without explicit loop closure detection. PVO[22] extends DROID-SLAM by integrating video panoptic segmentation (VPS) and visual odometry (VO) modules. PVO ignores the problem of loop closure. These methods use offline bundle adjustment, which can lead to drift errors, highlighting the need for online bundle adjustment. We introduce a novel architecture that integrates loop closure and online bundle adjustment, expanding on the PVO model[22]. The results show that our model outperforms state-of-the-art methods in visual odometry tasks. The ablation study shows that our technique outperforms current state-of-the-art methods, exhibiting superior performance across majority of sequences. Following are our contributions:

Comprehensive modeling of the scene: We have introduced a new Video Panoptic Segmentation module to enhance Visual Odometry by incorporating panoptic segmentation results, creating panoptic-aware dynamic masks for a better understanding of scene.

High Accuracy:We have introduced a vision transformer-based loop closure module that corrects accumulated errors by recognizing and aligning revisited locations in the map. Additionally, we have substituted the traditional bundle adjustment module with an innovative online bundle adjustment to optimize camera trajectories in real-time, enhancing system responsiveness and adaptability.

High Adaptability:We incorporated a novel panoptic update module that utilizes panoptic segmentation to improve confidence maps of 3D point clouds to handle dynamic objects.

High Robustness: We compared loop closure detection techniques, particularly evaluating the effectiveness of Deit-base and Deit-base-distilled methods by Touvron et al.[17]

Remaining part of the paper is arranged as follows:-Section 2 provides the background and reviews related work, Section 3 presents the proposed scheme, Section 4 describes experimental setup, Section 5 presents the results, Section 6 shows the ablation study, and Section 7 concludes the paper.

2 Related Work

Visual SLAM architecture consists of front-end, back-end, loop closure, and bundle adjustment. Front-end processes sensor data, extracting features, tracking them across frames, and estimating the camera's motion and a local map. Back-end refines this information, optimizing the camera trajectory and map using bundle adjustment. Bundle adjustment is a nonlinear optimization technique that refines 3D reconstruction models by minimizing differences be-

tween predicted and observed feature positions in multiple images. Loop closure recognizes previously visited locations, enabling the system to correct drift and enhance map accuracy. These modules collaborate iteratively to enhance system's accuracy and robustness, enabling real-time map construction and maintenance.

Challenges in deep learning-based Visual SLAM include the demand for extensive labeled data, computational complexity, and ensuring robustness to diverse environmental conditions and sensor noise. There are many SLAM algorithms, each with its own limitations. ORB-SLAM[13] and ORB-SLAM2[12] classify dynamic object points as outliers. Kinectfusion[7], ORB-SLAM, and ORB-SLAM2 are designed under the assumption of a stationary environment. ParticleSfM[24] focuses on camera pose estimation in SfM. DynaSLAM[1] combines Mask-RCNN and multiview geometry to manage both known and unknown moving objects, yet it is computationally intensive and best suited for offline use. D3VO[20] is critiqued for its sensitivity to lighting changes. EffiScene[8] employs regular 2D images as input and relies on photometric error as its primary loss function. The competitive collaboration[14] scheme excels at separating independent moving objects but struggles with static objects or objects moving at the same speed as the background. DeFlowSLAM[21] introduces a dual-flow representation and a self-supervised method to enhance performance, yet it struggles with high computational demands. DROID-SLAM[16] conducts recurrent iterative optimizations of camera poses and depth maps using a dense bundle adjustment layer, but it incurs high computational costs. NeRF-based GO-SLAM[23] tries to achieve real-time global optimization of poses and 3D reconstruction. VPSNet[9] introduces a novel task and proposes an instance-level tracking-based approach, representing a pioneering effort in the field. SiamTrack[19] builds on VPSNet by introducing a pixel-tube matching loss and a contrast loss to enhance the distinguishing ability of instance embedding. STEP[18] suggests segmenting and tracking each pixel for video panoptic segmentation.

Diverging from existing approaches, we present a VO-Enhanced VPS Module that incorporates camera pose, depth, and optical flow estimated from VO to track and integrate information from the current frame to neighboring frames, effectively addressing occlusion challenges. Our model includes loop closure, a critical component of SLAM that detects revisited locations to correct localization errors and enhance map consistency. We have implemented online Bundle adjustment which is essential to refine camera poses and 3D structure, improving map accuracy by minimizing reprojection errors.

3 Proposed Scheme

3.1 Network Architecture

The figure 1 shows the overall network architecture. The notations used are listed in table 1 The algorithm 1 represents the overall algorithm. Key components are given below.

Initial Panoptic Segmentation: This module takes an image as input and produces panoptic segmentation, which merges the semantic segmentation and instance segmentation of the image. The output result is fed into the VO module and the VPS module. This module uses PanopticFPN[10], which uses ResNet $f\theta_e$ as the backbone. It extracts multiscale features of image I_t.

$$z_t = f_{\theta_e}(I_t) \tag{1}$$

The results are then produced using a decoder g_{θ_d} with weights as θ_d.The panoptic results

Not.	Description	Not.	Description
V, E	Frame Graph	g_θ	feature vector
I_t	Image	G_t	Camera pose
d_t	Inverse Depth	τ_c	Camera Parameters
$\Delta\varepsilon^k$	Pose Update	Δd^k	Depth update
MSA	Multihead Self-Attention	MLP	Multilayer Perceptron
T_k	Processed embedded patch		

Table 1: Notations.

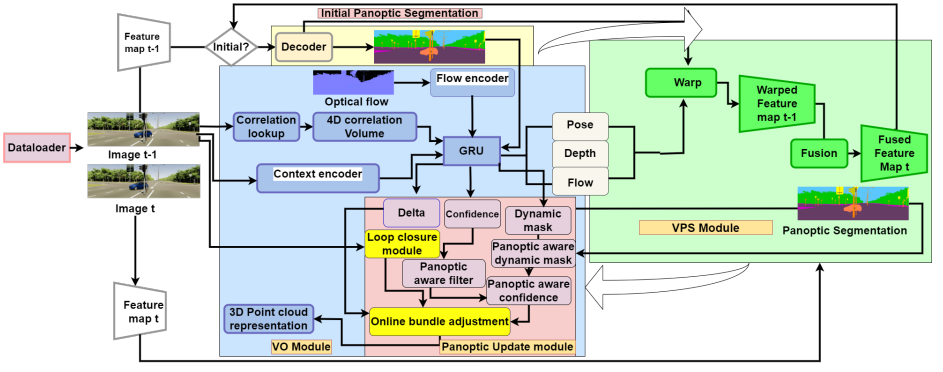


Figure 1: Overall Architecture of the proposed model

for a pixel p can be written as the following:

$$P_s(p|z_t) = g_{\theta_d}(p, z_t) \tag{2}$$

The VO and VPS modules further improve the panoptic segmentation results.

Visual Odometry Module: It has the following segments:

Feature Extraction: The feature extraction part adopts key components from the RAFT[15] network. Two separate networks have been used for the feature extraction: the feature encoder and the context encoder. The feature vector creates $4D$ correlation volumes while the context vector is embedded into the panoptic update module. The structure of the feature encoder is similar to that of the panoptic segmentation network.

Correlation Volume. A graph (V, E) is created between the frames to specify the co-visibility between frames, where an edge between two frames means that they share some area in common. We then form a $4D$ correlation volume using feature vectors $g_\theta I_i$ and $g_\theta I_j$. The correlation between 2 frames is defined as follows:

$$C_{ij} = \langle g_\theta I_i, g_\theta I_j \rangle \tag{3}$$

Panoptic Update Module: This module is shown in pink color in figure 1. The update module incorporates the results from the VPS module to adjust the weights. The update module is a 3×3 convolutional GRU unit. This module is similar to [22]. The GRU has a flow encoder, context encoder and $4D$ correlation volume. The GRU unit then computes optical flow delta r_{ij}, confidence map of correlation w_{ij}, dynamic mask M_{dij}, flow updates, $\Delta\varepsilon$ and Δd. For calculating panoptic aware confidence:

$$w_{pij} = sigmoid(w_{ij} + (1 - M_{dij}) \cdot \eta) \tag{4}$$

Algorithm 1 Complete algorithm for our Panoptic Segmentation based Visual SLAM

Require: Frame: Images acquisition
/*Initial Panoptic Segmentation*/
while Frame I_i **do**
  $P_i \leftarrow PanopticFPN(I_i)$
end while
/*VO module*/
while Frame I_i **do**
  $g_{\theta i} \leftarrow FeatureVector$; $c_\theta \leftarrow ContextVector$; $Pose, Flow, Depth \leftarrow Backbone(I_i)$;
  while j such that i, j share common areas **do**
    $C_{ij} \leftarrow \langle g_\theta I_i, g_\theta I_j \rangle$
  end while
  $F_i \leftarrow Flow$; $Delta, Confidence, DynamicMask \leftarrow GRU(C_i, F_i, P_i)$;
  $PanopticAwareDynamicMask \leftarrow Sigmoid(DynamicMask, P_i)$
  if LoopDetection(i) = True **then**
    Add neighbour edges in the frame graph i,j-1
  end if
  $BundleAdjustment(I_i, PanopticAwareConfidence)$; $Genferate3DPointCloud(I_i)$;
end while
/*VPS module*/
while Frame t **do**
  $WarpedFeaturest \leftarrow Fusion(Depth, Flow, Pose, FeatureMapt)$
end while

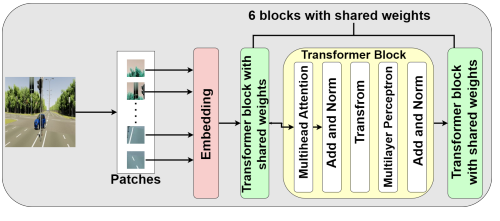


Figure 2: Loop closure module*(Refer Yellow coloured box in Fig 1).*

where η=10. Then, updates are applied to the current depth and pose estimates.

$$G^{(n+1)} = exp(\Delta\varepsilon^{(n)}) \circ G^{(n)}, \tag{5}$$

$\Delta\varepsilon^{(n)}$ are pose residuals transformed using the SE3 manifold to update the current pose.

$$d^{(n+1)} = \Delta d^{(n)} + d^{(n)} \quad ; \quad M_d^{(n+1)} = \Delta M_d^{(n)} + M_d^{(n)} \tag{6}$$

Correspondence. At each iteration, pose estimate and current depth are used to estimate the correspondence of the grid of pixels pi in frame i. The correspondence is:

$$p_{ij} = \tau_c(G_{ij} \circ \tau_c^{-1}(pi, di)) \tag{7}$$

Here, camera mode τ_c maps the corresponding 3D points to the image, and τ_c^{-1} is the inverse transformation that projects the pixel grid pi and depths d to the 3D point cloud.

Loop Closure(LC): The loop closure module(fig 2) starts by first patching the image into several parts. Then, each patch x^k is fed into the transformer block. The transformer then creates a feature vector out of it. Then, we can recognise loop closure when two frames have a similarity score higher than a preset threshold. We adopt the same procedure described in TT-LCD[5]. The similarity score of frames i and j is defined as where P_i represents the feature vector obtained by the transformer block[5]:

$$Sim(i,j) = \frac{P_i \cdot P_j}{\|P_i\|\|P_j\|} \tag{8}$$

Similar to[5], the feature extraction of the module can be illustrated as follows:

$$T_0 = [xclass; x_1 Emb; x_2 Emb; \ldots; x_m Emb] + Emb_{pos} \tag{9}$$

$$T'_k = MSA(LN(T_{k-1})) + T_{k-1}, k = 1, \ldots, K \tag{10}$$

$$T_k = MLP(LN(T'_k)) + T'_k, k = 1, \ldots, K \tag{11}$$

$$y = PCA(LN(T^0_K)) \tag{12}$$

. The Multi-head Self-Attention module(MSA)[5] can be represented as:

$$MSA(h) = Linear(Concate(h_1, .., h_k, ..., h_n)) \quad ; \quad h_k = softmax(\frac{Q_k K_k^T}{\sqrt{d}})V_k \tag{13}$$

Here, LN is the layer norm, and Qk, Kk, and Vk denote the queries, keys and values generated by linear projections respectively, and d is the dimension of patch embedding. After realising the loop closure event between i and j frames, we add edges i,j+1 and i,j-1 to the frame graph to enrich the 3D point cloud. We currently use DEIT-based-distilled model[17] by Facebook research to get feature vectors out of the images. These extracted features are used for the loop closure detection using the cosine similarity score.

Online Bundle Adjustment We modified Bundle Adjustment(BA) as specified in [16]. DROID-SLAM[16] does bundle adjustment after the end of camera tracking, which may result in accumulation of drift errors. So, we use an online version of bundle adjustment, which applies bundle adjustment as soon as a new frame is encountered. We also apply a loop bundle adjustment with relaxed parameters to realize a loop closure.

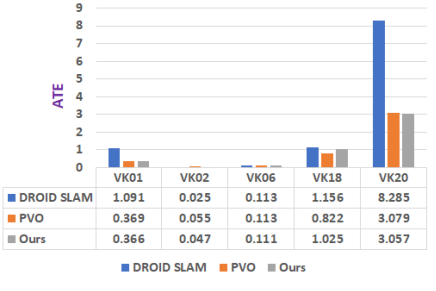


Figure 3: Virtual KITTI Dataset result

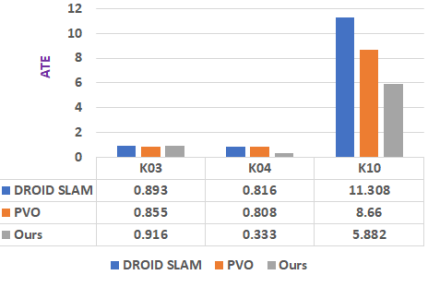
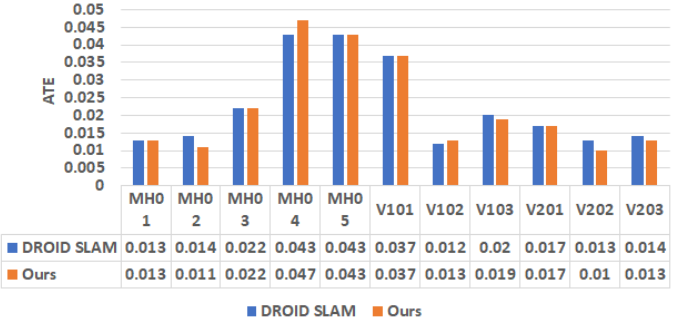


Figure 4: KITTI Dataset result

Video Panoptic Segmentation Module: Similar to PVO, Video panoptic segmentation tries to segment the incoming frames while maintaining the consistency of segmentation



	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203
DROID SLAM	0.013	0.014	0.022	0.043	0.043	0.037	0.012	0.02	0.017	0.013	0.014
Ours	0.013	0.011	0.022	0.047	0.043	0.037	0.013	0.019	0.017	0.01	0.013

Figure 5: EuRoC Dataset result

among the images. The VPS module (as shown in fig 1 by green part) obtains the warped feature of the frame t-1 and frame t by incorporating the flow, depth and pose information obtained using the VO module. For the segmentation to be consistent, the warped features of t-1 and fused feature map t are fed into the decoder to obtain the panoptic segmentation of t-1 and t. An IOU(intersection over union) match is performed between the two frames to obtain a consistent panoptic segmentation.

Algorithm1 processes each frame with segmentation, feature extraction, and depth computation, refines dynamic masks for moving objects using a GRU-based module, and employs loop detection and bundle adjustment for accurate mapping.

3.2 Cost Functions

1. **Loop Closure Module**: We use cosine similarity score (eq 14) as loss function for this module. In eq 14, P_i and P_j are feature vectors for the frames i and j respectively. A high cosine similarity score indicates that the two keyframes are likely to be from the same or nearby locations, which can be indicative of a loop closure.

2. **Bundle Adjustment Module**: We use following loss function [16] for this module. Eqn 14 outlines the objective of determining an updated G'and depth d'to ensure that reprojected points align with the updated correspondence pij*, following the predictions of the update operator.

$$E(G',d') = \sum_{(i,j)\in(V,E)} \|p_i j* - \tau_c(G'_{ij} \circ \tau_c^{-1}(p_i, d'_i)\|^2_{\Sigma_{ij}} \quad ; \quad \Sigma_{ij} = diagw_{pij} \tag{14}$$

4 Experimental Details

Dataset: We used Virtual Kitti, Kitti and EuRoC datasets[11] for evaluating our model.

1. **Virtual Kitti**[3]: This valuable dataset consists of 5 sequences cloned from kitti. Its synthetic nature allows for scalability and variability.

2. **EuRoC**[2]: This dataset consists of synchronised stereo images, IMU (Inertial Measurement Unit) readings, and ground truth poses collected from a micro aerial vehicle (MAV) flying indoor and outdoor scenarios. The dataset includes various challenging

Model	VK01	VK02	VK06	VK18	VK20
DROID-SLAM	1.091	**0.025**	0.113	1.156	8.285
Ours(VPS→VO x1)	0.366	0.048	0.111	1.034	3.264
Ours(VPS→VO x2)	0.366	0.047	0.111	1.026	3.151
Ours(VPS→VO x3)	**0.366**	0.047	**0.111**	**1.025**	**3.057**

Table 2: **Ablation study of VO module**: These results show how the recurrent update of VO → VPS module affect the results and its comparison with DROID-SLAM.

Model	VK01	VK02	VK06	VK18	VK20
PVO(VPS→VO)	0.374	0.057	0.113	0.960	3.487
PVO(VPS→VO x2)	0.371	0.057	0.113	0.954	3.135
PVO(VPS→VO x3)	0.369	0.055	0.113	**0.822**	3.079
Ours(VPS→VO x1)	0.366	0.048	0.111	1.034	3.264
Ours(VPS→VO x2)	0.366	0.047	0.111	1.026	3.151
Ours(VPS→VO x3)	**0.366**	**0.047**	**0.111**	1.025	**3.057**

Table 3: **Ablation study of VO module**: These results show how the recurrent update of VO → VPS module affect the results and its comparison with PVO.

environments, such as offices, corridors, and urban areas, providing diverse conditions for testing algorithms. It consists of 11 sequences in total.

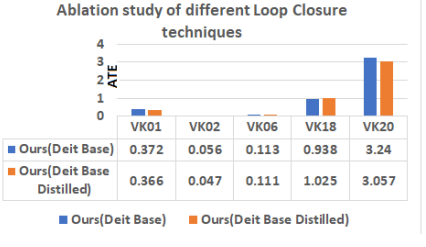


Figure 6: Loop Closure Ablation

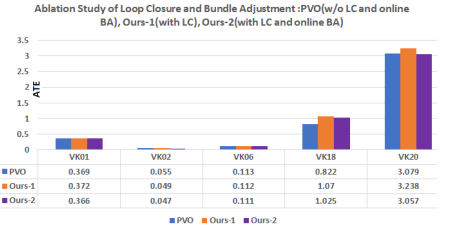


Figure 7: LC and Online BA Ablation

3. **KITTI**[6]: This dataset captures real-world traffic scenarios with plenty of static and dynamic objects. We have selected K03, K04 and K10 as these sequences are diverse in terms of scene complexity, lighting conditions, and weather conditions.

Training Details: The training of the VO module and VPS module are done as specified in PVO. We currently use a pre-trained image transformer named DEIT(Data Efficient Image Transformers)[17] (base distilled variant) for loop closure detection.
Evaluation Metrics: Absolute Trajectory Error(ATE) is used as evaluation metric. ATE quantifies the difference between the estimated and ground truth trajectories in SLAM systems, providing a measure of the overall accuracy of the estimated trajectory.

5 Results

We have evaluated the proposed model with different datasets. The results are shown in Fig 3, Fig 4 and Fig 5. We have achieved better performance in majority of the sequences across the datasets. Our trajectories(Fig 8) are very close to the ground truth. Plots for all virtual Kitti sequences, K10 of the Kitti dataset, and MH02, V103, and V202 of EuRoC are shown.

1. **Virtual Kitti**: Our method exhibits competitive performance(Fig3.) across various sequences of the KITTI dataset compared to PVO and DROID-SLAM. Our approach achieves superior or comparable ATE scores in most cases. Our model under-performs PVO only in VK18 and DROID-SLAM in VK02.

2. **EuRoC**: Our model outperforms or matches DROID-SLAM in 9 of the 11 sequences in this dataset(Fig 5). However, it can't beat DROID-SLAM in MH04 and V102 seq.

3. **Kitti**: We have evaluated on K03,K04 and K10 sequences (Fig 4). Here, our model outperforms both models, DROID-SLAM and PVO, in 2 of the sequences. However, it fails to do so in K03, but the results in this sequence are also close enough.

6 Ablation Study

1. **Ablation Study for VO module** We compare the performance (tables 2, 3) of a model with another method across multiple iterations of VPS $\rightarrow$ VO cycles. It consistently demonstrates the superiority of the discussed model over PVO from the first iteration.

2. **Ablation study for loop closure models.** Figure 6 summarizes the comparison of loop closure performance between two models: Deit-base and Deit-base-distilled.

3. **Ablation study for Impact of loop closure and bundle adjustment modules**

   Our analysis investigates the influence of our loop closure and online bundle adjustment technique on the Virtual KITTI dataset. The results are consolidated in Table 7. It is observed that while the incorporation of loop closure enhances performance in certain sequences, the effect is notably augmented when coupled with online bundle adjustment. This suggests a synergistic relationship between loop closure and online bundle adjustment, further improving the dataset's outcomes.

7 Conclusion

In this paper we have proposed a novel Visual SLAM architecture that fuses Panoptic Segmentation with bundle adjustment and loop closure and presents a powerful framework for constructing detailed and semantically meaningful maps of the environment, enabling advanced robotic perception and navigation capabilities. The experimental results reveals that proposed PV-SLAM model outperforms SOTA schemes for three datasets namely Virtual Kitti, EuroC and KITTI. The ablation study indicates that adding loop closure and online bundle adjustment improves the results. However, our results did not exhibit improvement in specific instances, such as VK18. This can be attributed to the fact that certain images, despite lacking sufficient similarity, were erroneously matched as loop closure instances.

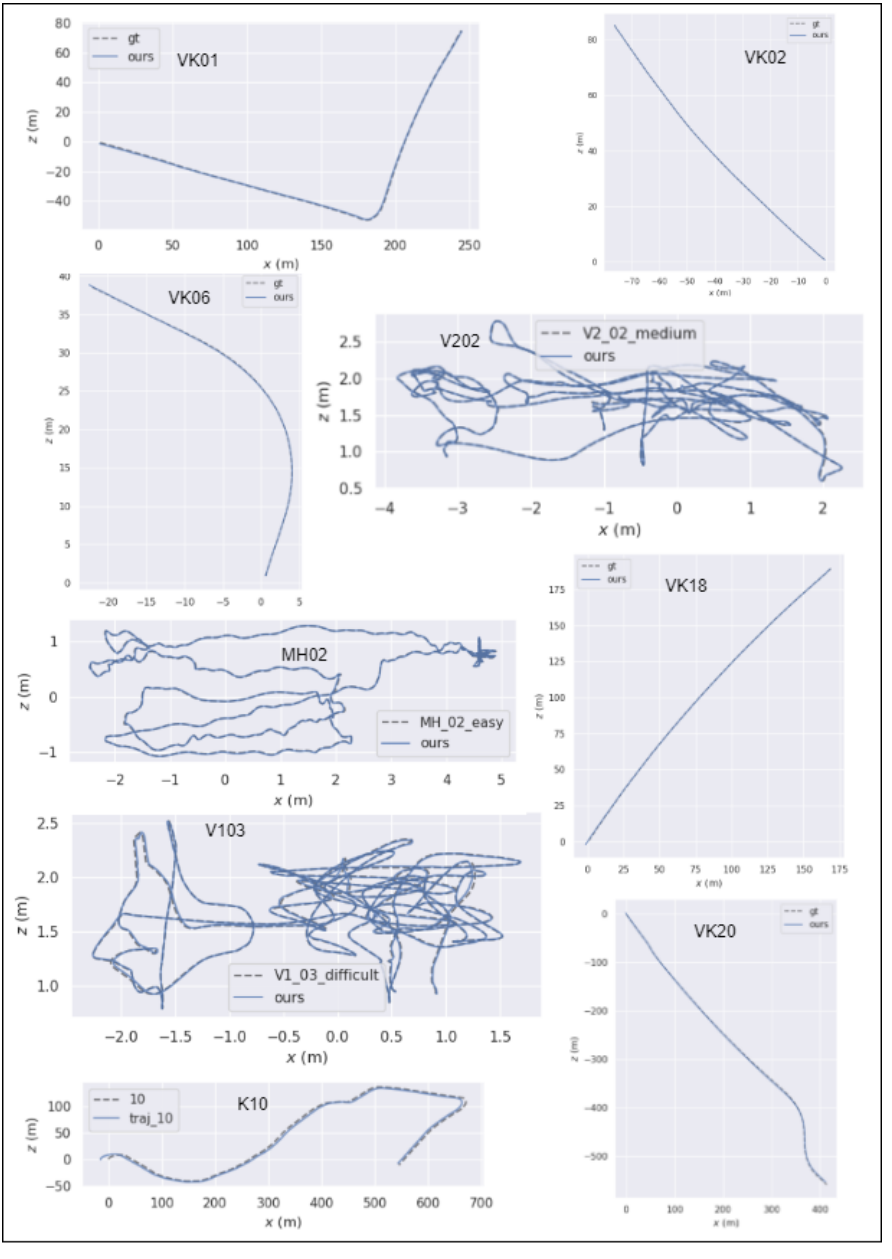


Figure 8: Generated trajectories of Kitti, Virtual Kitti and EuRoC dataset various sequences. The trajectories for Kitti and Virtual Kitti are taken in outdoor environments, while EuRoC is in indoor environment. This plot shows how close our results are to the ground truth.

Consequently, this misidentification resulted in subpar outcomes, showing the importance of ensuring accurate matching criteria. In the future, we plan to test the proposed model on additional datasets to demonstrate its versatility.

References

[1] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4): 4076–4083, 2018.

[2] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

[4] Weifeng Chen, Guangtao Shang, Aihong Ji, Chengjun Zhou, Xiyang Wang, Chonghui Xu, Zhenxiong Li, and Kai Hu. An overview on visual slam: From tradition to semantic. *Remote Sensing*, 14(13):3010, 2022.

[5] Chenchen Ding, Hongwei Ren, Zhiru Guo, Minjie Bi, Changhai Man, Tingting Wang, Shuwei Li, Shaobo Luo, Rumin Zhang, and Hao Yu. Tt-lcd: Tensorized-transformer based loop closure detection for robotic visual slam on edge. In *2023 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 166–172. IEEE, 2023.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[8] Yang Jiao, Trac D Tran, and Guangming Shi. Effiscene: Efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5538–5547, 2021.

[9] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020.

[10] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.

[11] Yuanzhi Liu, Yujia Fu, Fengdong Chen, Bart Goossens, Wei Tao, and Hui Zhao. Simultaneous localization and mapping related datasets: A comprehensive survey. *arXiv preprint arXiv:2102.04036*, 2021.

[12] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras. *IEEE Transactions on Robotics*, 33(5): 1255–1262, oct 2017. doi: 10.1109/tro.2017.2705103. URL https://doi.org/10.1109%2Ftro.2017.2705103.

[13] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5): 1147–1163, 2015.

[14] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019.

[15] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[16] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[17] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.

[18] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

[19] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2705–2714, 2021.

[20] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1281–1292, 2020.

[21] Weicai Ye, Xingyuan Yu, Xinyue Lan, Yuhang Ming, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Deflowslam: Self-supervised scene motion decomposition for dynamic dense slam. *arXiv preprint arXiv:2207.08794*, 2022.

[22] Weicai Ye, Xinyue Lan, Shuo Chen, Yuhang Ming, Xingyuan Yu, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pvo: Panoptic visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2023.

[23] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023.

[24] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*, pages 523–542. Springer, 2022.