PVO: Panoptic Visual Odometry

Weicai Ye\textsuperscript{1,2*}, Xinyue Lan\textsuperscript{1,2*}, Shuo Chen\textsuperscript{1,2}, Yuhang Ming\textsuperscript{3,4}, Xinyuan Yu\textsuperscript{1,2}, Hujun Bao\textsuperscript{1,2}, Zhaopeng Cui\textsuperscript{1}, Guofeng Zhang\textsuperscript{1,2†}

\textsuperscript{1}State Key Lab of CAD\&CG, Zhejiang University \textsuperscript{2}ZJU-SenseTime Joint Lab of 3D Vision \textsuperscript{3}School of Computer Science, Hangzhou Dianzi University \textsuperscript{4}VIL, University of Bristol

\{weicaiye, xinyuelan, chenshuo.eric, RickyYXY, baohujun, zhpcui, zhangguofeng\}@zju.edu.cn

yuhang.ming@hdu.edu.cn

https://zju3dv.github.io/pvo/

Figure 1. \textbf{Panoptic Visual Odometry.} PVO takes monocular video as input and outputs the panoptic 3D map while simultaneously localizing the camera itself with respect to the map.

Abstract

We present PVO, a novel panoptic visual odometry framework to achieve more comprehensive modeling of the scene motion, geometry, and panoptic segmentation information. Our PVO models visual odometry (VO) and video panoptic segmentation (VPS) in a unified view, which makes the two tasks mutually beneficial. Specifically, we introduce a panoptic update module into the VO Module with the guidance of image panoptic segmentation. This Panoptic-Enhanced VO Module can alleviate the impact of dynamic objects in the camera pose estimation with a panoptic-aware dynamic mask. On the other hand, the VO-Enhanced VPS Module also improves the segmentation accuracy by fusing the panoptic segmentation result of the current frame on the fly to the adjacent frames, using geometric information such as camera pose, depth, and optical flow obtained from the VO Module. These two modules contribute to each other through recurrent iterative optimization. Extensive experiments demonstrate that PVO outperforms state-of-the-art methods in both visual odometry and video panoptic segmentation tasks.

1. Introduction

Understanding the motion, geometry, and panoptic segmentation of the scene plays a crucial role in computer vision and robotics, with applications ranging from autonomous driving to augmented reality. In this work, we take a step toward solving this problem to achieve a more comprehensive modeling of the scene with monocular videos.

Two tasks have been proposed to address this problem, namely visual odometry (VO) and video panoptic segmentation (VPS). In particular, VO\textsuperscript{[9,11,36]} takes monocular videos as input and estimates the camera poses under the static scene assumption. To handle dynamic objects in the scene, some dynamic SLAM systems\textsuperscript{[2,43]} use instance segmentation network\textsuperscript{[14]} for segmentation and explicitly filter out certain classes of objects, which are potentially dynamic, such as pedestrians or vehicles. However, such approaches ignore the fact that potentially dynamic objects can actually be stationary in the scene, such as a parked vehicle. In contrast, VPS\textsuperscript{[17,42,49]} focuses on tracking individual instances in the scene across video frames given some initial panoptic segmentation results. Current VPS methods do not explicitly distinguish whether the object instance is moving or not. Although existing approaches broadly solve these two tasks independently, it is worth noticing that dy-
Dynamic objects in the scene can make both tasks challenging. Recognizing this relevance between the two tasks, some methods [5, 7, 19, 21] try to tackle both tasks simultaneously and train motion-semantics networks in a multi-task manner, shown in Fig. 2. However, the loss functions used in these approaches may contradict each other, thus leading to performance drops.

In this work, we propose a novel panoptic visual odometry (PVO) framework that tightly couples these two tasks using a unified view to model the scene comprehensively. Our insight is that VPS can adjust the weight of VO with panoptic segmentation information (the weights of the pixels of each instance should be correlated) and VO can convert the tracking and fusion of video panoptic segmentation from 2D to 3D. Inspired by the seminal Expectation-Maximization algorithm [26], recurrent iterative optimization strategy can make these two tasks mutually beneficial.

Our PVO consists of three modules, an image panoptic segmentation module, a Panoptic-Enhanced VO Module, and a VO-Enhanced VPS Module. Specifically, the panoptic segmentation module (see Sec. 3.1) takes in single images and outputs the image panoptic segmentation results, which are then fed into the Panoptic-Enhanced VO Module as initialization. Note that although we choose PanopticFPN [20], any segmentation model can be used in the panoptic segmentation module. In the Panoptic-Enhanced VO Module (see Sec. 3.2), we propose a panoptic update module to filter out the interference of dynamic objects and hence improve the accuracy of pose estimation in the dynamic scene. In the VO-Enhanced VPS Module (see Sec. 3.3), we introduce an online fusion mechanism to align the multi-resolution features of the current frame to the adjacent frames based on the estimated pose, depth, and optical flow. This online fusion mechanism can effectively solve the problem of multiple object occlusion. Experiments show that the recurrent iterative optimization strategy improves the performance of both VO and VPS. Overall, our contributions are summarized as four-fold.

- We present a novel Panoptic Visual Odometry (PVO) framework, which can unify VO and VPS tasks to model the scene comprehensively.
- A panoptic update module is introduced and incorporated into the Panoptic-Enhanced VO Module to improve pose estimation.
- An online fusion mechanism is proposed in the VO-Enhanced VPS Module, which helps to improve video panoptic segmentation.
- Extensive experiments demonstrate that the proposed PVO with recurrent iterative optimization is superior to state-of-the-art methods in both visual odometry and video panoptic segmentation tasks.

2. Related Work

2.1. Video Panoptic Segmentation

Video panoptic segmentation aims to generate consistent panoptic segmentation and track the instances to all pixels across video frames. A pioneer work, VPSNet [17] defines this novel task and proposes an instance-level tracking-based approach. SiamTrack [42] extends VPSNet by proposing a pixel-tube matching loss and a contrast loss to improve the discriminative power of instance embedding. VIP-Deeplab [30] presents a depth-aware VPS network by introducing additional depth information. While STEP [41] proposes to segment and track every pixel for video panoptic segmentation. HybridTracker [49] proposes to track instances from two perspectives: the feature space and the spatial location. Different from existing methods, we introduce a VO-Enhanced VPS Module, which exploits the camera pose, depth, and optical flow estimated from VO to track and fuse information from the current frame to the adjacent frames, and can handle occlusion.

2.2. SLAM and Visual Odometry

SLAM stands for simultaneous self-localization and map construction, and visual odometry, serving as the front end of SLAM, focuses on pose estimation. Modern SLAM systems roughly fall into two categories, geometry-based methods [8, 11, 27, 48], and learning-based methods [35, 37, 40, 54]. With the promising performance of supervised learning-based methods, unsupervised learning-based VO methods [31, 51, 52] have received much attention, but they do not perform as well as supervised ones. Some unsupervised methods [15, 47, 56] exploit multi-task learning with auxiliary tasks such as depth and optical flow to improve performance.

Recently, TartanVO [38] proposes to build a generalizable
learning-based VO and tests the system on a challenging SLAM dataset, TartanAir [39]. DROID-SLAM [34] proposes to iteratively update the camera pose and pixel-wise depth with a dense bundle adjustment layer and demonstrates superior performance. DeFlowSLAM [50] further proposes dual-flow representation and a self-supervised method to improve the performance of the conventional geometric-based SLAM, but they [1,2,10,25,29,32,45,53,55] mostly act on the stereo, RGBD, or LiDAR sequences. Instead, we introduce a panoptic update module and build the panoptic-enhanced VO on DROID-SLAM, and can work on monocular videos. Such a combination makes it possible to better understand of scene geometry and semantics, hence more robust to the dynamic objects in the scenes. Unlike other multi-task end-to-end models [19], our PVO has a recurrent iterative optimization strategy that prevents the tasks from jeopardizing each other.

3. Method

Given a monocular video, PVO aims for simultaneous localization and panoptic 3D mapping. Fig. 3 depicts the framework of the PVO model. It consists of three main modules: an image panoptic segmentation module, a Panoptic-Enhanced VO Module, and a VO-Enhanced VPS Module. The VO Module aims at estimating camera pose, depth, and optical flow, while the VPS Module outputs the corresponding video panoptic segmentation. The last two modules contribute to each other in a recurrent interactive manner.

3.1. Image Panoptic Segmentation

Image panoptic segmentation takes single images as input, and outputs the panoptic segmentation results of the images, which combines semantic segmentation and instance segmentation to model the instances of the image comprehensively. The output result is used to initialize video panoptic segmentation and then fed into the Panoptic-Enhanced VO Module (see Sec. 3.2). In our experiments, if not specifically indicated, we use the widely-used image panoptic segmentation network, PanopticFPN [20]. PanopticFPN is built on the backbone of ResNet \( f_\theta \) with weight \( \theta_e \) and extracts multi-scale features of image \( I_t \):

\[
z_t = f_{\theta_e}(I_t)
\]

(1)

It outputs the panoptic segmentation results using a decoder \( g_{\theta_d} \) with weights \( \theta_d \), consisting of semantic segmentation and instance segmentation. The panoptic segmentation results of each pixel \( p \) are:

\[
P_{s}(p|z_t) = g_{\theta_d}(p, z_t)
\]

(2)

The multi-scale features which are fed into the decoder are updated over time. In the beginning, the multi-scale features generated by the encoder are directly fed into the decoder (Fig. 3 blue part). In the later timesteps, these multi-scale features are updated with the online feature fusion module before being fed into the decoder (see Sec. 3.3).
3.2. Panoptic-Enhanced VO Module

In visual odometry, where dynamic scenes are ubiquitous, it is crucial to filter out the interference of dynamic objects. The front-end of DROID-SLAM [34] takes monocular video \( \{I_i\}_{i=0}^N \) as input and optimizes the residuals of camera pose \( \{G_i\}_{i=0}^N \in SE(3) \) and inverse depth \( d_i \in \mathbb{R}^{H \times W} \) by iteratively optimizing optical flow delta \( r_{ij} \in \mathbb{R}^{H \times W \times 2} \) with confidence \( w_{ij} \in \mathbb{R}^{H \times W \times 2} \). It does not consider that most backgrounds are static, foreground objects may be dynamic, and the weights of the pixels of each object should be correlated. The insight of the Panoptic-Enhanced VO Module (see Fig. 4) is to assist in obtaining better confidence estimation (see Fig. 7), by incorporating information from the panoptic segmentation. Thus, Panoptic-Enhanced VO can get more accurate camera poses. Next, we will briefly review the similar part (feature extraction and correlation) with DROID-SLAM, and focus on the sophisticated design of the panoptic update module.

3.2.1 Feature Extraction and Correlation

Feature Extraction. Similar to DROID-SLAM [34], the Panoptic-Enhanced VO Module borrows the key components of RAFT [33] to extract the features. We use two separate networks (a feature encoder and a context encoder) to extract the multi-scale features of each image, where the features from the feature encoder are exploited to construct 4D correlation volumes of pair images, and the features from the context encoder are injected into the panoptic update module (see Sec. 3.2.2). The structure of the feature encoder is similar to the backbone of the panoptic segmentation network, and they can be shared by a feature encoder. Note that for implementation convenience, we use different encoders.

Correlation Pyramid and Lookup. Similar to DROID-SLAM [34], we adopt a frame graph \( (\mathcal{V}, \mathcal{E}) \) to indicate the co-visibility between frames. For example, an edge \((i, j) \in \mathcal{E}) \) represents the two images \( I_i \) and \( I_j \) maintaining overlapped areas, and a 4D correlation volume can be constructed through dot product between the feature vectors of these two images:

\[
C^{ij} = \langle g_{\theta}(I_i), g_{\theta}(I_j) \rangle \tag{3}
\]

The average pooling layer is followed to gain the pyramid correlation. We use the same lookup operator defined in DROID-SLAM [34] to index the pyramid correlation volume values with bilinear interpolation. These correlation features are concatenated, resulting in the final feature vectors.

3.2.2 Panoptic Update Module

The Panoptic-Enhanced VO Module (see Fig. 4) which inherits from the front-end VO Module of DROID-SLAM, leverages the panoptic segmentation information to adjust the weight of VO. The flow information obtained by feeding the initial optical flow to the flow encoder and the 4D correlation volumes established from the two frames and the features acquired by the context encoder are fed to the GRU as intermediate variables, and then the three convolutional layers output a dynamic mask \( M_{d_{ij}} \in \mathbb{R}^{H \times W \times 2} \), a correlation confidence map \( w_{ij} \in \mathbb{R}^{H \times W \times 2} \) and a dense optical flow delta \( r_{ij} \in \mathbb{R}^{H \times W \times 2} \), respectively. We can adjust the dynamic mask to the panoptic-aware dynamic mask given the initialized panoptic segmentation. For understanding, we leave the notation unchanged. Especially, the stuff segmentation will be set as static, while the foreground objects with high dynamic probability will be set as dynamic. The confidence and panoptic-aware dynamic mask are passed through a panoptic-aware filter module to obtain the panoptic-aware confidence:

\[
w_{p_{ij}} = \text{sigmoid}(w_{ij} + (1 - M_{d_{ij}}) \cdot \eta) \tag{4}
\]

where \( \eta \) is set as 10 in our experiment.

The obtained flow delta \( r_{ij} \) adding the original optical flow is fed to the dense bundle adjustment (DBA) layer to optimize the residual of the inverse depth and the pose. The panoptic update module is iteratively optimized \( N \) times until convergence. Following DROID-SLAM [34], the pose residuals \( \Delta \xi^{(n)} \) are transformed on the SE3 manifold to update the current pose, while the residuals of depth and dynamic mask are added to the current depth and dynamic mask, respectively:

\[
G^{(n+1)} = \text{Exp}(\Delta \xi^{(n)}) \circ G^{(n)} \tag{5}
\]
An online fusion module will fuse the features of the current frame $t$ with the previous frame $t-1$. The dense correspondence field $p_{ij}$ for each edge $(i, j) \in E$ in the frame graph can be computed as follows:

$$p_{ij} = \Pi_i(G_{ij} \circ \Pi_i^{-1}(p_i, d_i)), \quad p_{ij} \in \mathbb{R}^{H \times W \times 2}, \quad G_{ij} = G_j \circ G_i^{-1}$$

(7)

where $\Pi_i$ is the camera model that reprojects 3D coordinate points to the image plane, while $\Pi_i^{-1}$ is the inverse function that projects the 2D coordinate grid $p_i$ and the inverse depth map $d$ to the 3D coordinate points. $G_{ij}$ represents the relative pose of the images $I_i$ and $I_j$. $p_{ij}$ is 2D coordinate grid when the coordinate of pixel $p_i$ is mapped to $j$ frame with the current estimated pose and depth. The corrected correspondence represents the sum of the predicted correspondence and the optical flow residuals, i.e. $p_{ij}^* = p_{ij} + r_{ij}$.

**DBA Layer.** We use the dense bundle adjustment layer (DBA) defined in DROID-SLAM [34] to map stream revisions to update the current estimated pixel-wise depths and poses. The cost function can be defined as follows:

$$E(G', d') = \sum_{(i,j) \in E} ||p_{ij}^* - \Pi_i(G_{ij}' \circ \Pi_i^{-1}(p_i, d_i))||^2_{\Sigma_{ij}}$$

(8)

$$\Sigma_{ij} = \text{diag}(w_{pij})$$

(9)

We use the Schur complement to solve this non-linear least squares problem, Eq. 8. The Gauss-Newton algorithm is exploited to update the residuals of the pose ($\Delta \Theta$), the depth, and the mask ($\Delta D$).

**3.3. VO-Enhanced VPS Module**

Video panoptic segmentation aims to obtain panoptic segmentation results for each frame and maintain the segmentation’s consistency between frames. To improve the segmentation accuracy and tracking accuracy, some methods such as FuseTrack [17] try to use optical flow information to fuse features and track them according to the similarity of features. These methods only come from a 2D perspective that may encounter occlusion or violent motion. We live in a 3D world where additional depth information can be used to model the scene better. Our VO-Enhanced VPS Module is based on this understanding and can better solve the mentioned problems.

Fig. 5 shows the VO-Enhanced VPS Module, which obtains the warped feature by warping the feature of the previous frame $t-1$ to the current frame $t$, using the depth, pose, and optical flow information obtained from visual odometry. An online fusion module will fuse the features of the current frame $t$ and the warped features to obtain the fused features.

Figure 5. VO-Enhanced VPS Module. VO-Enhanced VPS Module enables feature tracking and fusion of different frames using the pose, depth, and optical flow information obtained from Visual Odometry. An online fusion module is included to better cope with occlusion challenges. The video panoptic segmentation results will be fed into the Panoptic-Enhanced VO Module.

To keep the consistency of the video segmentation, we first feed the warped features $t-1$ (containing geometric motion information) and the fused feature map $t$ into the decoder to obtain the panoptic segmentation $t-1$ and $t$, respectively. Then a simple IoU-match module is used to obtain a consistent panoptic segmentation. This result will be fed into the Panoptic-Enhanced VO Module.

**VO-Aware Online Fusion.** The feature fusion network first concatenates the two features $z_{t-1}$ and $z_t$, and then passes through a convolutional layer with ReLU activations to obtain the fused features $z_t$. Inspired by NeuralBlox [24], we propose two loss functions for supervision to ensure that online feature fusion can be effective (see Tab. 5).

**Feature Alignment Loss [24].** We employ a feature alignment loss to minimize the distance between $z_t^*$ and $\hat{z}_t$ in latent space:

$$L_{fca} = ||z_t^* - \hat{z}_t||_1$$

(10)

where $z_t^*$ denotes the average feature of the same pixel warped from different images to the same image.

**Segmentation Consistent Loss.** Additionally, we add a segmentation loss that minimizes the logit differences of query pixels $p$ decoded using different features $z_t^*$ and $\hat{z}_t$:

$$L_{seg} = \sum_{p \in P} ||g_{\theta_q}(p, z_t^*) - g_{\theta_q}(p, \hat{z}_t)||_1$$

(11)
### Table 1. SLAM Comparison Results on KITTI (K) & Virtual KITTI (VK) Datasets with Metric: ATE[m]. X means system failure.

| Method      | K00  | K01  | K02  | K03  | K04  | K05  | K06  | K07  | K08  | K09  | K10  | VK01 | VK02 | VK06 | VK18 | VK20 |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| DynaSLAM    | 8.07 | 385.33 | 21.776 | 0.873 | 1.402 | 4.461 | 14.364 | 2.628 | 50.369 | 41.91 | 7.519 | 27.830 | X     | X    | X    | 2.807 |
| DROID-SLAM  | 4.86 | 95.45 | 18.81  | 0.893 | 16.03 | 42.786 | 16.34  | 46.4  | 1.091 | 0.025 | 0.113 | 1.156 | 8.285 |
| Ours        | 5.69 | 91.19 | 23.6   | 0.855 | 0.808 | 8.41  | 13.57  | 8.89  | 6.67  | 14.65 | 8.66  | 0.369 | 0.055 | 0.113 | 0.822 | 3.079 |

Table 2. Absolute Trajectory Error (ATE) Comparison on TUM-RGBD Dynamic Sequences. The best results are shown in bold. PVO achieves competitive and even best performance, outperforming DROID-SLAM in all sequences.

#### 3.4. Recurrent Iterative Optimization

We can optimize the proposed Panoptic-Enhanced VO Module and VO-Enhanced VPS Module in a recurrent iterative manner until convergence, which is inspired by the EM algorithm. Experimentally, it generally takes only two iterations for the loop to converge. Tab. 5 and Tab. 6 demonstrate that recurrent iterative optimization can boost the performance of both the VPS and VO Modules.

#### 3.5. Implementation Details

Implemented by PyTorch, PVO consists of three main modules: image panoptic segmentation, Panoptic-Enhanced VO Module, and VO-Enhanced VPS Module. We use three stages to train our network. Image panoptic segmentation is trained on Virtual KITTI [3] dataset as initialization. Following PanopticFCN, we adopt a multi-scale scaling policy during training. We optimize the network with an initial rate of 1e-4 on two GeForce RTX 3090 GPUs, where each mini-batch has eight images. The SGD optimizer is used with a weight decay of 1e-4 and momentum of 0.9. The training of the Panoptic-Enhanced VO Module follows DROID-SLAM [34], except that it additionally feeds the ground-truth panoptic segmentation results. Specifically, we trained this module on the Virtual KITTI dataset with two GeForce RTX-3090 GPUs for 80,000 steps, which took about two days. When training the VO-enhanced video panoptic segmentation module, we use the ground-truth depth, optical flow, and pose information as geometric priors to align the features, and fix the backbone of the trained single-image panoptic segmentation, and then train the fusion module only. The network is optimized with an initial learning rate of 1e-5 on one GeForce RTX 3090 GPU, where each batch has eight images. When the fusion network has largely converged, we add a segmentation consistency loss function to refine our VPS Module further.

#### 4. Experiments

For visual odometry, we conduct experiments on three datasets with dynamic scenes: Virtual KITTI, KITTI, and TUM RGBD dynamic sequences. Absolute Trajectory Error (ATE) is used for evaluation. For video panoptic segmentation, we use Video Panoptic Quality (VPQ) metric [17] on Cityscapes, and VIPER datasets. We further perform ablation studies on Virtual KITTI to analyze the design of our framework. Finally, we demonstrate the applicability of our PVO on video editing, shown in the supplementary materials.

#### 4.1. Visual Odometry

**VKITTI2.** Virtual KITTI dataset [3] consists of 5 sequences cloned from the KITTI tracking benchmark, which provides RGB, depth, class segmentation, instance segmentation, camera pose, flow, and scene flow data for each sequence. As shown in Tab. 6 and Fig. 6, our PVO outperforms DROID-SLAM by a large margin for most sequences and achieves competitive performance in sequence 02.

**Figure 6. Trajectory Comparison on KITTI and VKITTI2.** Our method performs better than DROID-SLAM, having better trajectory estimation results.

**KITTI.** KITTI [12] is a dataset capturing real-world traffic scenarios, ranging from freeways over rural areas to urban streets with plenty of static and dynamic objects. We applied the PVO model trained on the VKITTI2 [3] dataset
4.2. Video Panoptic Segmentation

We compare PVO with three instance-based video panoptic segmentation methods, namely VPSNet-Track, VPSNet-FuseTrack [18], and SiamTrack [42]. Built on the image panoptic segmentation model UPSNet [44], VPSNet-Track additionally adds MaskTrack head [46] to form the video panoptic segmentation model. VPSNet-FuseTrack based on VPSNet-Track additionally injects temporal feature aggregation and fusion. While SiamTrack finetunes VPSNet-Track with the pixel-tube matching loss [42] and the contrast loss and has slight performance improvement. VPSNet-FuseTrack is mainly compared because the code of SiamTrack is not available.

Cityscapes. We adopt the public train/val/test split of Cityscapes in VPS [17], where each video contains 30 consecutive frames, with the corresponding ground truth annotations for every five frames. Tab. 3 demonstrates that our method with PanopticFCN [22] outperforms the state-of-the-art method on the val dataset, achieving +1.6% VPQ higher than the VPSNet-Track. Compared with VPSNet-FuseTrack [17], our method has slight improvement and can keep consistent video segmentation, shown in the supplementary materials. The reason is that our VO Module only obtains 1/8 resolution optical flow and depth due to the limited memory.

VIPER. VIPER maintains plenty of high-quality panoptic video annotations, which is another video panoptic segmentation benchmark. We follow VPS [18] and adopt its public train/val split. We use 10 selected videos from day scenarios and the first 60 frames of each video are used for evaluation. Tab. 4 demonstrates that compared with VPSNet-FuseTrack, our method with PanopticFCN achieves much higher scores (+3.1 VPQ) on the VIPER dataset.

4.3. Ablation Study

VPS-Enhanced VO Module. In the Panoptic-Enhanced VO Module, we use DROID-SLAM [34] as our baseline. (VPS->VO) means the panoptic information prior was added to enhance the VO baseline. (VPS->VO x2) means that we can iteratively optimize the VO Module twice. (VPS->VO x3) means recurrent iterative optimization on the VO Module 3 times. Tab. 6 and Fig. 7 show the panoptic information can help improve the accuracy of DROID-SLAM on most of the highly dynamic VKITTI2 datasets. The recurrent iterative...
Table 5. **Ablation Study of VO-Enhanced VPS Module Variants on VKITTI2 Dataset.** Each cell contains VPQ / VPQTh / VPQSt scores. The best results are highlighted in boldface. Our method performs better than existing video panoptic segmentation methods.

| Methods on VKITTI2 | Temporal window size | VPQ |
|-------------------|----------------------|-----|
|                   | k = 0                |     |
| VPS baseline      | 58.24 / 60.11 / 57.93|     |
| VPS baseline + w/fusion | 59.16 / 67.00 / 54.91|     |
| Ours (VO->VPS + w/o fusion) | 58.24 / 60.11 / 57.93| 55.67 / 54.44 / 56.28 |
| Ours (VO->VPS + w/fusion + w/o fca loss) | 58.51 / 64.07 / 56.97| 54.29 / 59.01 / 55.53 |
| Ours (VO->VPS + w/fusion + w/o seg loss) | 58.73 / 65.05 / 56.95| 54.51 / 59.34 / 54.89 |
| Ours (VO->VPS)    | 59.18 / 67.00 / 54.94| 54.94 / 57.77 / 54.15 |
| Ours (VO->VPS + w/o depth ) x2 | 59.17 / 66.87 / 56.95| 54.72 / 56.46 / 54.22 |
| Ours (VO->VPS) x2 | 59.18 / 67.00 / 54.94| 54.94 / 57.66 / 54.17 |

Table 6. **Ablation Study of Panoptic-Enhanced VO Module Results on VKITTI2 Dataset.** Our method outperforms DROID-SLAM on most of the highly dynamic VKITTI2 datasets, and the accuracy of the pose estimation is significantly improved and slightly slowed down after recurrent iterative optimization.

| Monocular | 01 | 02 | 06 | 18 | 20 | Avg |
|-----------|----|----|----|----|----|-----|
| DROID-SLAM [34] | 0.049 | 0.025 | 0.113 | 1.156 | 8.285 | 2.134 |
| Ours (VPS->VO w/o filter) | 0.384 | 0.061 | 0.116 | 0.936 | 5.375 | 1.374 |
| Ours (VPS->VO) | 0.374 | 0.057 | 0.113 | 0.960 | 3.487 | 0.998 |
| Ours (VPS->VO x2) | 0.371 | 0.057 | 0.113 | 0.954 | 3.135 | 0.926 |
| Ours (VPS->VO x3) | 0.369 | 0.055 | 0.113 | 0.822 | 3.079 | 0.888 |
| DROID-SLAM's runtime (FPS) | 5.73 | 12.67 | 19.96 | 7.08 | 10.20 | 11.13 |
| Ours' runtime (FPS) | 4.45 | 9.69 | 14.52 | 6.22 | 8.10 | 8.60 |

**Figure 7. Panoptic-Aware Confidence.** We visualize the confidence of the PVO model vs. DROID-SLAM. We can see that with panoptic information, the panoptic weights can better remove the dynamic interference and keep the static features for solving the camera pose. The black color indicates that the confidence tends to be close to 0.

flow information from RAFT [33] for inter-frame tracking. This is set as VPS baseline. (VPS baseline + w/fusion) means we additionally fuse the feature with the flow estimation. (VO->VPS + w/o fusion) means that we use additional depth, pose, and other information on top of the baseline. (VO->VPS) means we additionally fuse the feature. (VO->VPS x2) means that we use the recurrent iterative optimization module to enhance the VPS results further. As shown in Tab. 5 and in the supplementary materials, the VO-Enhanced VPS Module is effective in improving segmentation accuracy and tracking consistency.

**Online Fusion in VO-Enhanced VPS Module.** To validate the effectiveness of the proposed Feature Alignment Loss (fea loss) and Segmentation Consistent Loss (seg loss), the methods are followed: (VO->VPS + w/fusion + w/o fca loss) means we train the online fusion module without Feature Alignment Loss. (VO->VPS + w/fusion + w/o seg loss) means that we train the online fusion module without Segmentation Consistent Loss. Tab. 5 demonstrates the effectiveness of these two loss function.

### 5. Conclusion

We have presented a novel panoptic visual odometry method, which models the VO and the VPS in a unified view, enabling the two tasks to facilitate each other. The panoptic update module can help improve the pose estimation, while the online fusion module helps improve the panoptic segmentation. Extensive experiments demonstrate that our PVO outperforms state-of-the-art methods in both tasks.

**Limitations.** The main limitation is that PVO is built on DROID-SLAM and panoptic segmentation, which makes the network heavy and requires much memory. Although PVO can perform robustly in dynamic scenes, it ignores the problem of loop closure when the camera returns to the previous position. Exploring a low-cost and efficient SLAM system with loop closure is our future work.

### 6. Acknowledgements

This work was partially supported by NSF of China (No. 61932003) and ZJU-SenseTime Joint Lab of 3D Vision. Weici Ye was partially supported by China Scholarship Council (No. 202206320316).
References

[1] Berta Bescos, Carlos Campos, Juan D Tardós, and José Neira. DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM. IEEE Robotics and Automation Letters, 6(3):5191–5198, 2021.
[2] 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.
[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. arXiv preprint arXiv:2001.10773, 2020.
[4] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient Lidar-Based Semantic SLAM. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4530–4537. IEEE, 2019.
[5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint Learning for Video Object Segmentation and Optical Flow. In Proceedings of the IEEE International Conference on Computer Vision, pages 686–695, 2017.
[6] Weichen Dai, Yu Zhang, Ping Li, Zheng Fang, and Sebastian Scherer. RGB-D SLAM in Dynamic Environments using Point Correlations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(1):373–389, 2020.
[7] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every Frame Counts: Joint Learning of Video Segmentation and Optical Flow. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 10713–10720, 2020.
[8] J. Engel, T. Schops, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the European Conference on Computer Vision, 2014.
[9] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In Proceedings of the International Conference on Computer Vision, pages 1449–1456, 2013.
[10] Yingchun Fan, Qichi Zhang, Yuliang Fan, Shaofen Liu, and Hong Han. Blitz-SLAM: A Semantic SLAM in Dynamic Environments. Pattern Recognition, 121:108225, 2022.
[11] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In International Conference on Robotics and Automation, pages 15–22. IEEE, 2014.
[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision Meets Robotics: The KITTI Dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.
[13] Mathieu Gonzalez, Eric Marchand, Amine Kacete, and Jerome Royan. S3LAM: Structured Scene SLAM. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 6389–6395. IEEE, 2022.
[14] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pages 2961–2969, 2017.
[15] 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.
[16] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust Odometry Estimation for RGB-D Cameras. In IEEE International Conference on Robotics and Automation, pages 3748–3754. IEEE, 2013.
[17] 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.
[18] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video Panoptic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[19] Ue-Hwan Kim, Se-Ho Kim, and Jong-Hwan Kim. SimVODIS: Simultaneous Visual Odometry, Object Detection, and Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(1):428–441, 2020.
[20] 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.
[21] Jianfeng Li, Junqiao Zhao, Shuangfu Song, and Tianfeng Feng. Unsupervised Joint Learning of Depth, Optical Flow, Egomotion from Video. arXiv preprint arXiv:2105.14520, 2021.
[22] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully Convolutional Networks for Panoptic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 214–223, 2021.
[23] Konstantinos-Nektarios Lianos, Johannes L Schönberger, Marc Pollefeys, and Torsten Sattler. VSO: Visual Semantic Odometry. In Proceedings of the European Conference on Computer Vision, pages 234–250, 2018.
[24] Stefan Lionar, Lukas Schmid, Cesar Cadena, Roland Siegwart, and Andrei Cramariuc. NeuralBlox: Real-Time Neural Representation Fusion for Robust Volumetric Mapping. In Proceedings of the International Conference on 3D Vision, pages 1279–1289. IEEE, 2021.
[25] Yuhang Ming, Xingrui Yang, and Andrew Calway. Object-Augmented RGB-D SLAM for Wide-Disparity Relocalisation. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2203–2209, 2021.
[26] T.K. Moon. The Expectation-Maximization Algorithm. IEEE Signal Processing Magazine, 13(6):47–60, 1996.
[27] 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.
[28] Raul Mur-Artal and Juan D Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D.
[53] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. VDO-SLAM: A Visual Dynamic Object-Aware SLAM System. arXiv preprint arXiv:2005.11052, 2020.

[54] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1851–1858, 2017.

[55] Hu Zhu, Chen Yao, Zheng Zhu, Zhengtao Liu, and Zhenzhong Jia. Fusing Panoptic Segmentation and Geometry Information for Robust Visual SLAM in Dynamic Environments. In IEEE 18th International Conference on Automation Science and Engineering, pages 1648–1653, 2022.

[56] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency. In Proceedings of the European Conference on Computer Vision, pages 36–53, 2018.