PVO: Panoptic Visual Odometry

Weicai Ye[1,2*] Xinyue Lan[1,2*] Shuo Chen[1,2] Yuhang Ming[3,4] Xingyuan Yu[1,2] Hujun Bao[1,2]
Zhaopeng Cui[1] Guofeng Zhang[1,2†]

[1]State Key Lab of CAD&CG, Zhejiang University [2]ZJU-SenseTime Joint Lab of 3D Vision
[3]School of Computer Science, Hangzhou Dianzi University [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/

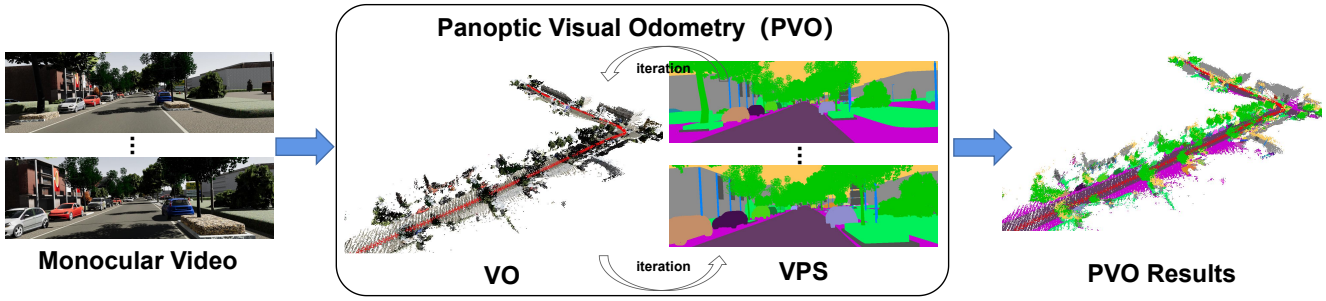


Figure 1. **Panoptic Visual Odometry.** PVO takes monocular video as input and outputs the panoptic 3D map while simultaneously localizing the camera itself with respect to the map.

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.

∗ indicates equal contribution. † indicates the corresponding author.

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 [9, 11, 38] 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 [2, 45] use instance segmentation network [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 [18, 44, 52] 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-

namic objects in the scene can make both tasks challenging. Recognizing this relevance between the two tasks, some methods [5, 7, 20, 22] 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 [28], 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 [21], 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.

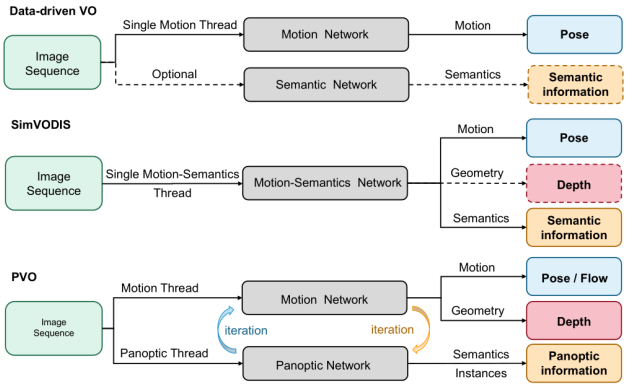


Figure 2. **Illustration.** Our PVO unifies visual odometry and video panoptic segmentation so that the two tasks can be mutually reinforced by iterative optimization. In contrast, methods such as SimVODIS [20] optimize motion and semantic information in a multi-task manner.

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 [18] defines this novel task and proposes an instance-level tracking-based approach. SiamTrack [44] extends VPSNet by proposing a pixel-tube matching loss and a contrast loss to improve the discriminative power of instance embedding. VIP-Deeplab [32] presents a depth-aware VPS network by introducing additional depth information. While STEP [43] proposes to segment and track every pixel for video panoptic segmentation. HybridTracker [52] 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, 29, 50], and learning-based methods [37, 39, 42, 57]. With the promising performance of supervised learning-based methods, unsupervised learning-based VO methods [33, 54, 55] have received much attention, but they do not perform as well as supervised ones. Some unsupervised methods [16, 49, 59] exploit multi-task learning with auxiliary tasks such as depth and optical flow to improve performance.

Recently, TartanVO [40] proposes to build a generalizable

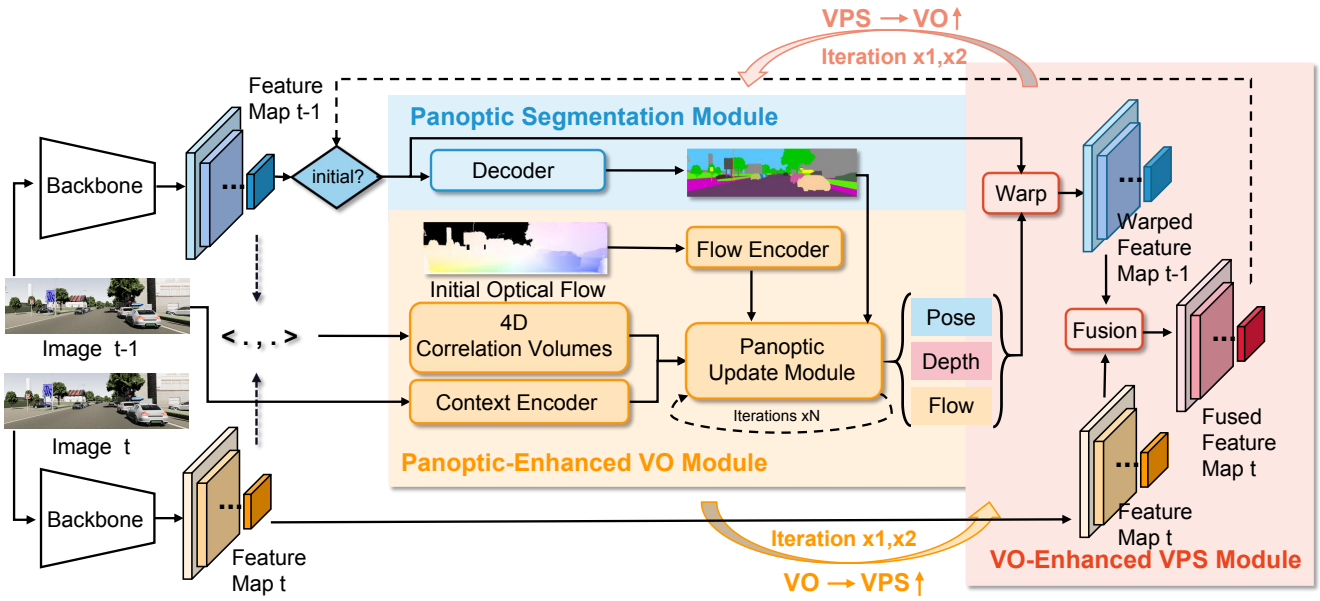


Figure 3. **Panoptic Visual Odometry Framework.** Our method consists of three modules, namely, an image panoptic segmentation module for system initialization (blue), a Panoptic-Enhanced VO Module (orange), and a VO-Enhanced VPS Module (red). The last two modules contribute to each other in a recurrent iterative manner.

learning-based VO and tests the system on a challenging SLAM dataset, TartanAir [41]. DROID-SLAM [36] proposes to iteratively update the camera pose and pixel-wise depth with a dense bundle adjustment layer and demonstrates superior performance. DeFlowSLAM [53] further proposes dual-flow representation and a self-supervised method to improve the performance of the SLAM system in dynamic scenes. To tackle the challenge of dynamic scenes, dynamic SLAM systems [4, 13] usually leverage semantic information as constraints [24] or prior to improve the performance of the conventional geometric-based SLAM, but they [1, 2, 10, 27, 31, 34, 47, 56, 58] 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 [20], 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 [21]. PanopticFPN is built on the backbone of ResNet f_{θ_e} with weight θ_e and extracts multi-scale features of image I_t:

$$\mathbf{z}_t = f_{\theta_e}(I_t) \tag{1}$$

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

$$P_s(\mathbf{p}|\mathbf{z}_t) = g_{\theta_d}(\mathbf{p}, \mathbf{z}_t) \tag{2}$$

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 [36] takes monocular video $\{\mathbf{I}_t\}_{t=0}^{N}$ as input and optimizes the residuals of camera pose $\{\mathbf{G}_t\}_{t=0}^{N} \in SE(3)$ and inverse depth $\mathbf{d}_t \in \mathbb{R}_{+}^{H\times W}$ by iteratively optimizing optical flow delta $\mathbf{r}_{ij} \in \mathbb{R}^{H\times W\times 2}$ with confidence $\mathbf{w}_{ij} \in \mathbb{R}^{H\times W\times 2}$. 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 [36], the Panoptic-Enhanced VO Module borrows the key components of RAFT [35] 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 use a shared encoder. Note that for implementation convenience, we use different encoders.

Correlation Pyramid and Lookup. Similar to DROID-SLAM [36], we adopt a frame graph $(\mathscr{V},\mathscr{E})$ to indicate the co-visibility between frames. For example, an edge $(i,j) \in \mathscr{E}$ 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 [36] 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

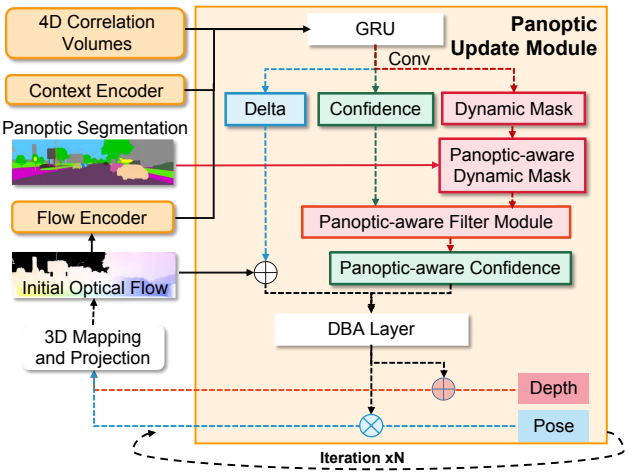


Figure 4. **Panoptic-Enhanced VO Module.** The Panoptic-Enhanced VO Module mainly feeds the 4D correlation volumes, the context information from the context encoder, and the flow information into the panoptic update module. The panoptic update module iterates N times to obtain better depth, pose, and optical flow estimation. The panoptic segmentation information is used to adjust the correlation weight and the optical flow is initialized as 0 and iteratively updated with the DBA layer.

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 $\mathbf{M}_{\mathbf{d}ij} \in \mathbb{R}^{H\times W\times 2}$, a correlation confidence map $\mathbf{w}_{ij} \in \mathbb{R}^{H\times W\times 2}$ and a dense optical flow delta $\mathbf{r}_{ij} \in \mathbb{R}^{H\times W\times 2}$, 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:

$$\mathbf{w}_{\mathbf{p}ij} = \text{sigmoid}(\mathbf{w}_{ij} + (1 - \mathbf{M}_{\mathbf{d}ij}) \cdot \eta) \tag{4}$$

where η is set as 10 in our experiment.

The obtained flow delta $\mathbf{r}_{ij}$ 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 [36], 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:

$$\mathbf{G}^{(n+1)} = \text{Exp}(\Delta\xi^{(n)}) \circ \mathbf{G}^{(n)} \tag{5}$$

$$\Theta^{(n+1)} = \Delta\Theta^{(n)} + \Theta^{(n)}, \Theta \in \{\mathbf{d}, \mathbf{M_d}\} \tag{6}$$

Correspondence. We first use the current pose and depth estimates at each iteration to search for the correspondence. Refer to DROID-SLAM [36], for each pixel coordinates $\mathbf{p}_i \in \mathbb{R}^{H\times W\times 2}$ in frame i, the dense correspondence field $\mathbf{p}_{ij}$ for each edge $(i,j) \in \mathscr{E}$ in the frame graph can be computed as follows:

$$\mathbf{p}_{ij} = \Pi_c(\mathbf{G}_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}_i)),\ \mathbf{p}_{ij} \in \mathbb{R}^{H\times W\times 2},\ \mathbf{G}_{ij} = \mathbf{G}_j \circ \mathbf{G}_i^{-1} \tag{7}$$

where Π_c is the camera model that reprojects 3D coordinate points to the image plane, while Π_c^{-1} is the inverse function that projects the 2D coordinate grid $\mathbf{p}_i$ and the inverse depth map $\mathbf{d}$ to the 3D coordinate points. $\mathbf{G}_{ij}$ represents the relative pose of the images I_i and I_j. $\mathbf{p}_{ij}$ is 2D coordinate grid when the coordinate of pixel $\mathbf{p}_i$ 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. $\mathbf{p}^*_{ij} = \mathbf{p}_{ij} + \mathbf{r}_{ij}$.

DBA Layer. We use the dense bundle adjustment layer (DBA) defined in DROID-SLAM [36] to map stream revisions to update the current estimated pixel-wise depths and poses. The cost function can be defined as follows:

$$\mathbf{E}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathscr{E}} \left\| \mathbf{p}^*_{ij} - \Pi_c(\mathbf{G}'_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}'_i)) \right\|^2_{\Sigma_{ij}} \tag{8}$$

$$\Sigma_{ij} = \operatorname{diag} \mathbf{w}_{\mathbf{p}_{ij}} \tag{9}$$

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 ($\Delta\xi$), the depth, and the mask ($\Delta\Theta$).

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 [18] 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.

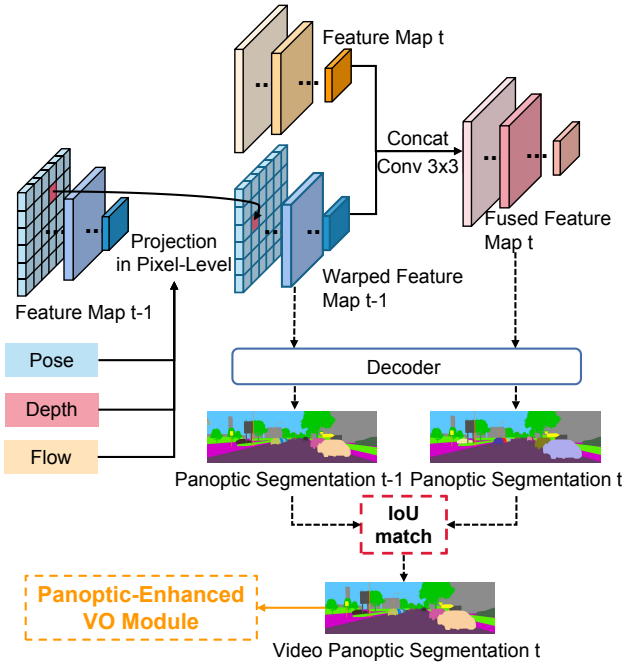


Figure 5. **VO-Enhanced VPS Module.** VO-Enhanced VPS Module enables feature tracking and fusion of different frames using the pose, depth, and optical flow information obtained from Visual Odometry. An online fusion module is included to better cope with occlusion challenges. The video panoptic segmentation result 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 $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$, and then passes through a convolutional layer with ReLU activations to obtain the fused features $\hat{\mathbf{z}}_t$. Inspired by NeuralBlox [25], we propose two loss functions for supervision to ensure that online feature fusion can be effective (see Tab. 5).

Feature Alignment Loss [25]. We employ a feature alignment loss to minimize the distance between $\mathbf{z_t}^*$ and $\hat{\mathbf{z}}_\mathbf{t}$ in latent space:

$$\mathscr{L}_{fea} = \left\| \mathbf{z_t}^* - \hat{\mathbf{z}}_\mathbf{t} \right\|_1 \tag{10}$$

where $\mathbf{z_t}^*$ 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 $\mathbf{p}$ decoded using different features $\mathbf{z_t}^*$ and $\hat{\mathbf{z}}_\mathbf{t}$:

$$\mathscr{L}_{seg} = \sum_{\mathbf{p}\in\mathbb{P}} \left\| g_{\theta_d}(\mathbf{p}, \mathbf{z_t}^*) - g_{\theta_d}(\mathbf{p}, \hat{\mathbf{z}}_\mathbf{t}) \right\|_1 \tag{11}$$

Method	K00	K01	K02	K03	K04	K05	K06	K07	K08	K09	K10	VK01	VK02	VK06	VK18	VK20
DynaSLAM [2]	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 [36]	**4.86**	95.45	**18.81**	0.893	0.816	16.03	42.786	27.402	16.34	46.4	11.308	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 1. **SLAM Comparison Results on KITTI (K) & Virtual KITTI (VK) Datasets with Metric: ATE[m].** X means system failure.

Sequences		Trans. RMSE of trajectory alignment [m]				
		DVO SLAM [17]	ORB-SLAM2 [30]	PointCorr [6]	DROID-SLAM [36]	Ours
slightly dynamic	fr2/desk-person	0.104	**0.006**	0.008	0.017	0.013
	fr3/sitting-static	0.012	0.008	0.010	0.007	**0.006**
	fr3/sitting-xyz	0.242	0.010	**0.009**	0.016	0.014
	fr3/sitting-rpy	0.176	0.025	**0.023**	0.029	0.027
	fr3/sitting-halfsphere	0.220	0.025	0.024	0.026	**0.022**
highly dynamic	fr3/walking-static	0.752	0.408	0.011	0.016	**0.007**
	fr3/walking-xyz	1.383	0.722	0.087	0.019	**0.018**
	fr3/walking-rpy	1.292	0.805	0.161	0.059	**0.056**
	fr3/walking-halfsphere	1.014	0.723	**0.035**	0.312	0.221

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 in 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 [36], 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 [18] on Cityscapes and VIPER datasets. We further perform ablation studies to analyze the design of our framework. Finally, we demonstrate the applicability of our PVO on video editing, as shown in Sec. B 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.

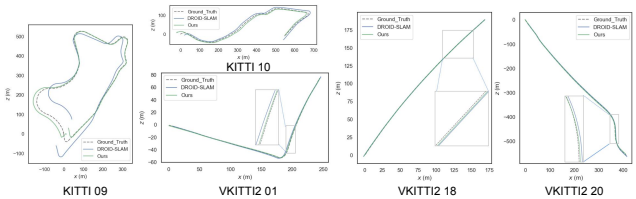


Figure 6. **Trajectory Comparison on KITTI and VKITTI2.** Our method performs better than DROID-SLAM, having better trajectory estimation results.

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 to the KITTI [12] sequences. As shown in Fig. 6 (KITTI

Methods on **Cityscapes-VPS *val***	Temporal window size				VPQ	FPS
	k = 0	k = 5	k = 10	k = 15		
VPSNet-Track	63.1 / 56.4 / 68.0	56.1 / 44.1 / 64.9	53.1 / 39.0 / 63.4	51.3 / 35.4 / 62.9	55.9 / 43.7 / 64.8	4.5
VPSNet-FuseTrack	64.5 / 58.1 / 69.1	57.4 / 45.2 / 66.4	54.1 / 39.5 / 64.7	52.2 / 36.0 / 64.0	57.2 / 44.7 / 66.6	1.3
SiamTrack	64.6 / 58.3 / 69.1	57.6 / 45.6 / 66.6	54.2 / 39.2 / 65.2	52.7 / 36.7 / 64.6	57.3 / 44.7 / 66.4	4.5
PanopticFCN [23] + Ours	**65.6** / 60.0 / 69.7	**57.8** / 45.7 / 66.6	54.3 / 39.5 / 65.1	52.1 / 35.4 / 64.3	**57.5** / 45.1 / 66.4	5.1
VPSNet-FuseTrack + Ours	65.0 / 59.0 / 69.4	57.6 / 45.0 / 66.7	**54.4** / 39.1 / 65.6	**52.8** / 35.8 / 65.2	**57.5** / 44.7 / 66.7	1.1

Table 3. **Video Panoptic Segmentation Comparison Results on Cityscapes-VPS Validation Dataset with VO-Enhanced VPS Module Variants.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores. The best results are highlighted in boldface. Our method generally outperforms VPSNet-FuseTrack [19] and SiamTrack [44].

Methods on **VIPER**	Temporal window size				VPQ	FPS
	k = 0	k = 5	k = 10	k = 15		
VPSNet-Track	48.1 / 38.0 / 57.1	49.3 / 45.6 / 53.7	45.9 / 37.9 / 52.7	43.2 / 33.6 / 51.6	46.6 / 38.8 / 53.8	5.1
VPSNet-FuseTrack	49.8 / 40.3 / 57.7	51.6 / 49.0 / 53.8	47.2 / 40.4 / 52.8	45.1 / 36.5 / 52.3	48.4 / 41.6 / 53.2	1.6
SiamTrack	51.1 / 42.3 / 58.5	**53.4** / 51.9 / 54.6	49.2 / 44.1 / 53.5	47.2 / 40.3 / 52.9	50.2 / 44.7 / 55.0	5.1
PanopticFCN + Ours	**54.6** / 50.3 / 57.9	51.7 / 44.5 / 57.3	**50.5** / 41.8 / 57.2	**49.1** / 38.9 / 56.9	**51.5** / 43.9 / 57.3	3.6

Table 4. **Video Panoptic Segmentation Comparison Results on VIPER with VO-Enhance VPS Variants.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores. The best results are highlighted in boldface. Our method generally outperforms VPSNet-FuseTrack [19] and SiamTrack [44].

09 and 10 sequences), the pose estimation error of PVO is only half that of DROID-SLAM, which proves the good generalization ability of PVO. Tab. 1 shows the complete SLAM comparison results on KITTI and VKITTI datasets, where PVO outperforms DROID-SLAM and DynaSLAM by a large margin in most scenarios. Note that we use the code of DynaSLAM, which is a classic SLAM system with instance segmentation. DynaSLAM falls into the catastrophic system failure in the VKITTI2 02, 06, and 18 sequences.

TUM-RGBD. TUM RGBD is a dataset capturing indoor scenes with a handheld camera. We choose the dynamic sequences of the TUM RGBD dataset to show the effectiveness of our method. We compare PVO with DROID-SLAM and three state-of-the-art dynamic RGB-D SLAM systems, namely DVO SLAM [17], ORB-SLAM2 [30] and PointCorr [6]. Note that PVO and DROID-SLAM only use monocular RGB videos. Tab. 2 demonstrates that PVO outperforms DROID-SLAM in all scenes. Compared to the conventional RGB-D SLAM systems, our method also performs better in most of the scenes.

4.2. Video Panoptic Segmentation

We compare PVO with three instance-based video panoptic segmentation methods, namely VPSNet-Track, VPSNet-FuseTrack [19], and SiamTrack [44]. Built on the image panoptic segmentation model UPSNet [46], VPSNet-Track additionally adds MaskTrack head [48] 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 [44] 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 [18], 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 [23] outperforms the state-of-the-art method on the val dataset, achieving **+1.6% VPQ** higher than the VPSNet-Track. Compared with VPSNet-FuseTrack [18], our method has slight improvement and can keep consistent video segmentation, shown in Fig. A4 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 [19] 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 [36] 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 optimization can further improve the results.

Methods on **VKITTI2**	Temporal window size				VPQ
	k = 0	k = 5	k = 10	k = 15	
VPS baseline	58.24 / 60.11 / 57.93	55.50 / 53.78 / 56.28	54.13 / 50.29 / 55.53	53.65 / 48.53 / 55.46	54.90 / 51.95 / 56.05
VPS baseline + w/fusion	59.16 / 67.00 / 56.91	56.27 / 60.98 / 54.96	54.96 / 57.74 / 54.18	54.58 / 55.97 / 54.19	55.81 / 59.23 / 54.85
Ours (VO->VPS + w/o fusion)	58.24 / 60.11 / 57.93	55.67 / 54.44 / 56.28	54.29 / 50.91 / 55.53	53.83 / 49.22 / 55.46	55.04 / 52.48 / 56.05
Ours (VO->VPS + w/fusion + w/o fea loss)	58.51 / 64.07 / 56.97	55.62 / 58.53 / 54.86	54.29 / 55.15 / 54.13	53.94 / 53.40 / 54.19	55.14 / 56.62 / 54.81
Ours (VO->VPS + w/fusion + w/o seg loss)	58.73 / 65.05 / 56.95	55.83 / 59.34 / 54.89	54.51 / 56.01 / 54.15	54.15 / 54.26 / 54.19	55.37 / 57.49 / 54.82
Ours (VO->VPS)	59.18 / 67.00 / 56.94	56.25 / 61.00 / 54.93	54.94 / 57.77 / 54.15	54.57 / 56.01 / 54.17	55.80 / 59.25 / 54.83
Ours (VO->VPS + w/o depth) x2	59.17 / 66.87 / 56.95	56.39 / 61.45 / 56.25	55.04 / 58.15 / 54.15	54.72 / 56.46 / 54.22	55.89 / 59.57 / 54.83
Ours (VO->VPS) x2	**59.18** / 67.00 / 56.94	**56.42** / 61.67 / 54.93	**55.10** / 58.40 / 54.15	**54.84** / 56.67 / 54.17	**55.94** / 59.77 / 54.83

Table 5. **Ablation Study of VO-Enhanced VPS Module Variants on VKITTI2 Dataset.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores. The best results are highlighted in boldface. Our method performs better than existing video panoptic segmentation methods.

Monocular	01	02	06	18	20	Avg
DROID-SLAM [36]	1.091	**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

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.

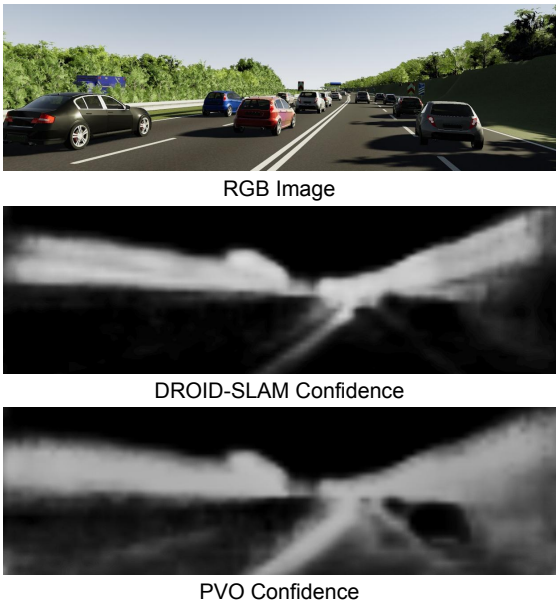


Figure 7. **Panoptic-Aware Confidence.** We visualize the confidence of the PVO model vs. DROID-SLAM. We can see that with panoptic information, the panoptic weights can better remove the dynamic interference and keep the static features for solving the camera pose. The black color indicates that the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

VO-Enhanced VPS Module. To evaluate whether VO helps VPS, we first use PanopticFPN [21] to get the panoptic segmentation results for each frame, and then use the optical flow information from RAFT [35] 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 Fig. A3 in the supp. 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 fea loss) means that 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. Weicai 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. 3

[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. 1, 3, 6

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. 6

[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. 3

[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. 2

[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. 6, 7

[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. 2

[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. 2

[9] Jakob Engel, Jurgen 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. 1

[10] Yingchun Fan, Qichi Zhang, Yuliang Tang, Shaofen Liu, and Hong Han. Blitz-SLAM: A Semantic SLAM in Dynamic Environments. *Pattern Recognition*, 121:108225, 2022. 3

[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. 1, 2

[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. 6

[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. 3

[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. 1

[15] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video Propagation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 451–461, 2017. 13

[16] 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. 2

[17] 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. 6, 7

[18] 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. 1, 2, 5, 6, 7, 13

[19] 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. 7

[20] 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. 2, 3

[21] 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. 2, 3, 8

[22] Jianfeng Li, Junqiao Zhao, Shuangfu Song, and Tiantian Feng. Unsupervised Joint Learning of Depth, Optical Flow, Egomotion from Video. *arXiv preprint arXiv:2105.14520*, 2021. 2

[23] 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. 7

[24] Konstantinos-Nektarios Lianos, Johannes L Schonberger, Marc Pollefeys, and Torsten Sattler. VSO: Visual Semantic Odometry. In *Proceedings of the European Conference on Computer Vision*, pages 234–250, 2018. 3

[25] 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. 5

[26] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimatte: Associating Objects and Their Effects in Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4507–4515, 2021. 16

[27] 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. 3

[28] T.K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. 2

[29] 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. 2

[30] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 6, 7

[31] Wenyan Ou, Jiaming Zhang, Kunyu Peng, Kailun Yang, Gerhard Jaworek, Karin Müller, and Rainer Stiefelhagen. Indoor Navigation Assistance for Visually Impaired People via Dynamic SLAM and Panoptic Segmentation with an RGB-D Sensor. In *Proceedings of International Conference Computers Helping People with Special Needs*, pages 160–168. Springer, 2022. 3

[32] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-DeepLab: Learning Visual Perception with Depth-Aware Video Panoptic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3997–4008, 2021. 2

[33] 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*, June 2019. 2

[34] Martin Runz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 10–20, 2018. 3

[35] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020. 4, 8

[36] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in Neural Information Processing Systems*, 2021. 3, 4, 5, 6, 7, 8, 12

[37] S. Vijayanarasimhan, S. Ricco, C. Schmidy, R. Sukthankar, and K. Fragkiadaki. SfM-Net: Learning of Structure and Motion from Video. In *arXiv:1704.07804*, 2017. 2

[38] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *IEEE International Conference on Robotics and Automation*, pages 2043–2050. IEEE, 2017. 1

[39] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks. *The International Journal of Robotics Research*, 37(4-5):513–542, 2018. 2

[40] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. TartanVO: A Generalizable Learning-based VO. *arXiv preprint arXiv:2011.00359*, 2020. 2

[41] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A Dataset to Push the Limits of Visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4909–4916, 2020. 3

[42] Xiangwei Wang, Daniel Maturana, Shichao Yang, Wenshan Wang, Qijun Chen, and Sebastian Scherer. Improving Learning-based Ego-motion Estimation with Homomorphism-based Losses and Drift Correction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 970–976. IEEE, 2019. 2

[43] Mark Weber, Jun Xie, Maxwell D. Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, Aljosa Osep, Laura Leal-Taixé, and Liang-Chieh Chen. STEP: Segmenting and Tracking Every Pixel. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 2

[44] 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. 1, 2, 7

[45] Linhui Xiao, Jinge Wang, Xiaosong Qiu, Zheng Rong, and Xudong Zou. Dynamic-SLAM: Semantic Monocular Visual Localization and Mapping based on Deep Learning in Dynamic Environment. *Robotics and Autonomous Systems*, 117:1–16, 2019. 1

[46] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A Unified Panoptic Segmentation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 7

[47] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM. In *International Conference on Robotics and Automation*, pages 5231–5237, 2019. 3

[48] Linjie Yang, Yuchen Fan, and Ning Xu. Video Instance Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019. 7

[49] 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*, June 2020. 2

[50] Xingrui Yang, Yuhang Ming, Zhaopeng Cui, and Andrew Calway. FD-SLAM: 3-D Reconstruction Using Features and Dense Matching. In *International Conference on Robotics and Automation*, pages 8040–8046, 2022. 2

[51] Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis. *arXiv preprint arXiv:2210.00647*, 2022. 16

[52] Weicai Ye, Xinyue Lan, Ge Su, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Hybrid Tracker with Pixel and Instance for Video Panoptic Segmentation. *arXiv preprint arXiv:2203.01217*, 2022. 1, 2

[53] Weicai Ye, Xingyuan Yu, Xinyue Lan, Yuhang Ming, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. De-

FlowSLAM: Self-Supervised Scene Motion Decomposition for Dynamic Dense SLAM. *arXiv preprint arXiv:2207.08794*, 2022. 3

[54] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[55] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 2

[56] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. VDO-SLAM: A Visual Dynamic Object-Aware SLAM System. *arXiv preprint arXiv:2005.11052*, 2020. 3

[57] 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. 2

[58] 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. 3

[59] 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. 2

PVO: Panoptic Visual Odometry

Supplementary Material

In this supplementary document, we provide more experiment results (Sec. A), such as the ablation study of our method. We further demonstrate the applicability of our method in video editing (Sec. B) and discuss the limitation (Sec. C) of PVO in video editing. We also provide the supplementary video which demonstrates the qualitative results of our method and the video editing effects.

A. Experiments Results

A.1. Ablation Study of Panoptic-Enhanced VO Module

In our Panoptic-Enhanced VO Module, unlike DROID-SLAM [36], we adjust the confidence by incorporating information from the panoptic segmentation. The dynamic mask is adjusted to the panoptic-aware dynamic mask given the initialized panoptic segmentation. Panoptic segmentation treats trees and buildings as stuff (i.e., the background is static), people and cars, etc. as things (i.e., the foreground). So the foreground objects with a high probability of motion are set to dynamic. We show an example of waiting for a traffic light in Fig. A1, where the white color indicates that the parked cars are static. We find the dynamic threshold set as 0.5 may achieve the best results, shown in Tab. A1. The reason is that when the dynamic threshold is small, too many static pixel points may be removed, while the dynamic threshold is too large, and small movements may be ignored. The confidence and panoptic-aware dynamic mask are passed through a panoptic-aware filter module to obtain the panoptic-aware confidence. As shown in Tab. A1, the panoptic-aware filter module can help improve the estimation of camera pose.

We show the qualitative results of the panoptic 3D maps produced by our method, shown in Fig. A2. The supplementary video also shows how our method works.

A.2. Ablation Study of VO-Enhanced VPS Module

Qualitative results in Fig. A3 shows our method can cope with occlusion better on VKITTI2 dataset. Fig. A4 demonstrates our method keeps consistent video panoptic segmentation on Cityscape-VPS dataset, compared with VPSNet-FuseTrack.

B. Video Editing Applications with PVO

In this section, we show the applicability of video editing with PVO, as shown in Fig. B5. We can obtain rich 2D and 3D information from panoptic visual odometry, which can be utilized in video editing.

Fig. B8 illustrates how we can perform consistent video editing using panoptic visual odometry. Firstly, we feed the original video frames from t to t+n into the PVO network. The VO-Enhanced VPS Module and VPS-Enhanced VO Module will get the panoptic segmentation result and optical flow estimation, depth, and pose information for each frame. In addition, the motion of dynamic objects can be decomposed into the dynamic field and static field of the camera. Similarly, the above operation in the new scene can get the whole scene modeling information. We can first select one instance of the original video, then obtain the motion of the target in the new scene by merging the static field of the new scene and the dynamic field of the selected object of the original video, together with additional information such as depth checks and occlusion completion, to complete the video effect of inserting the object into the new scene. In this

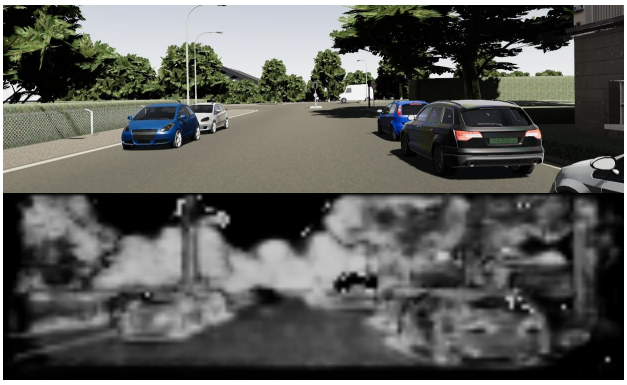


Figure A1. **Dynamic Probability of Parked Cars.** The black color indicates that the confidence tends to be close to 0.

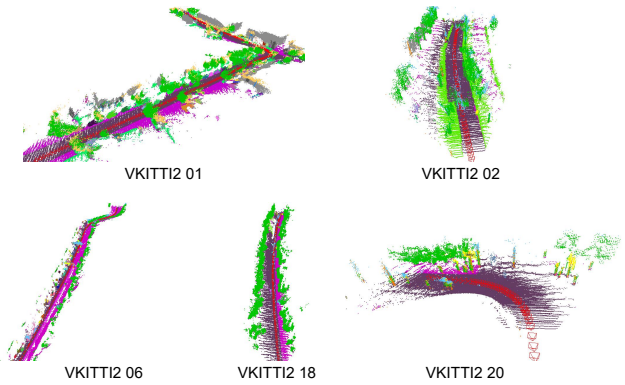


Figure A2. **Qualitative Results of Panoptic 3D Map Produced by PVO on Virtual KITTI Dataset.** We show the panoptic 3D map produced by our method. The red triangles indicate the camera pose, and different colors indicate different instances.

Monocular	vkitti01	vkitti02	vkitti06	vkitti18	vkitti20	Avg
DROID-SLAM	1.091	**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**
Ours (VPS->VO x3) threshold=0.1	0.377	0.052	**0.112**	0.950	3.240	0.946
Ours (VPS->VO x3) threshold=0.3	0.374	0.054	0.113	0.946	3.107	0.919
Ours (VPS->VO x3) threshold=0.5	0.369	0.055	0.113	0.822	3.079	0.888
Ours (VPS->VO x3) threshold=0.7	0.384	0.059	0.114	0.863	22.993	4.883
Ours (VPS->VO x3) threshold=0.9	1.348	0.065	0.119	0.885	17.337	3.951

Table A1. **Ablation Study of Panoptic-Enhanced VO Module on Virtual KITTI2 Dataset.** Panoptic-Enhanced VO Module outperforms DROID-SLAM on most of the highly dynamic VKITTI2 datasets, and the accuracy of the pose estimation is significantly improved after recurrent iterative optimization. The dynamic threshold set as 0.5 can achieve the best performance.

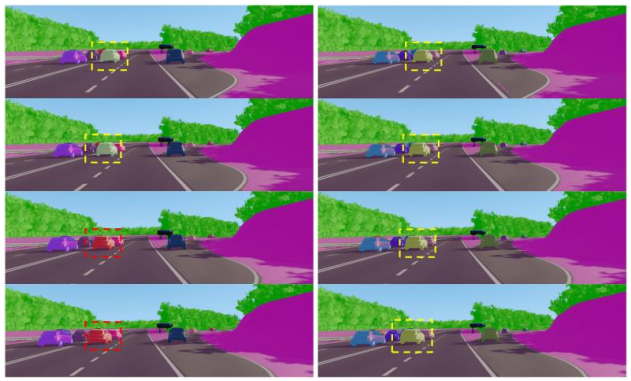


Figure A3. **Comparison Results of Our VO-Enhanced VPS Module with VPS Baseline on VKITTI2 Dataset**. Our method keeps the consistent video segmentation for it is better to cope with occlusion. Different colors indicate tracking failure.

Figure A4. **Comparison Results of Our method with VPSNet-FuseTrack [18] on Cityscape-VPS Val Dataset.** Compared with VPSNet-FuseTrack, our method can keep consistent video segmentation. Different colors indicate tracking failure.

way, we can perform several vivid video effects, including motion control, replication, deletion, and instance interaction. Note that when the initial segmentation is incomplete, with PVO, we can first fill in the occluded parts from multiple views, thus ensuring the integrity of the object, as shown in Fig. B9.

B.1. Ablation Study of Video Editing

We perform an ablation study of PVO in video editing compared with the existing method.

Baseline: We use Video Propagation Network [15] to perform video editing for motion control. The baseline method is generally simple to manipulate objects without considering occlusion, but it doesn't look realistic.

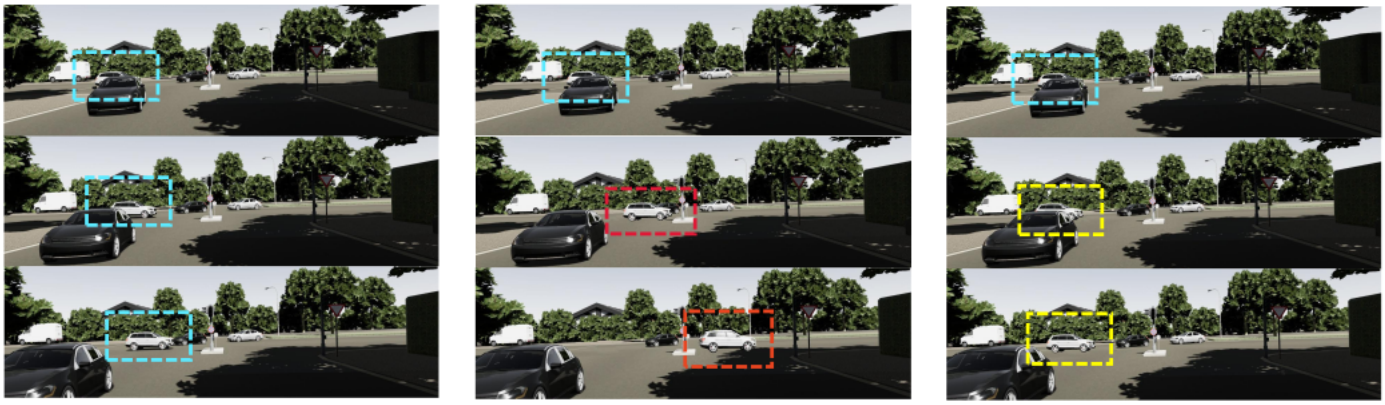
Ours (PVO): The PVO method can better model scene segmentation and motion geometry information and achieve better object manipulation results in occluded scenes, shown in Fig. B10.

B.2. Motion Control

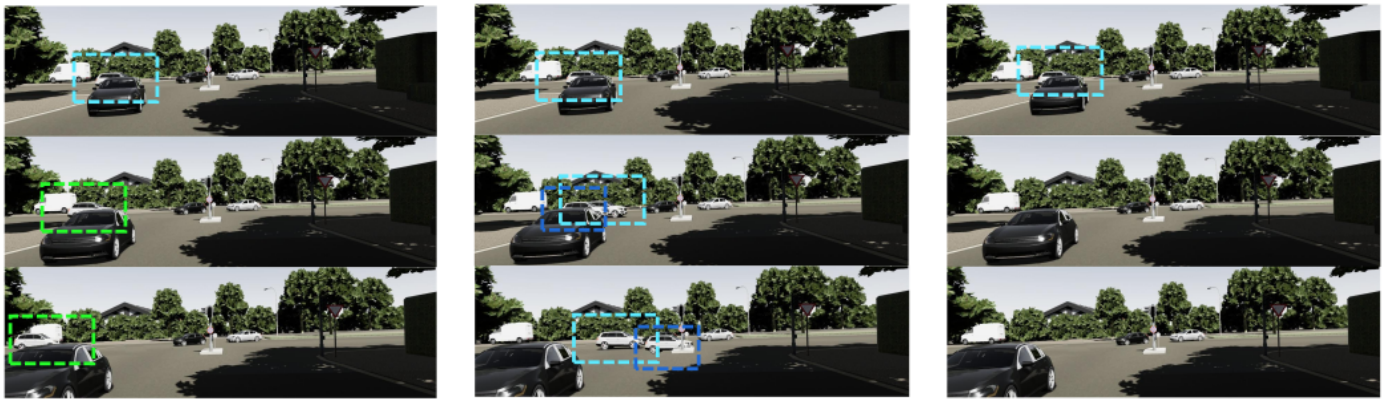
As shown in Fig. B6, we can insert moving objects into the new scene and also directly manipulate the motion patterns of the moving objects of the original video, such as acceleration, deceleration, pause, and rewind. We can also apply PVO to natural scenes such as Cityscapes for motion control, shown in Fig. B7 which shows the generalization of Panoptic Visual Odometry.

B.3. Single vs Multi Instance Interaction

Since PVO provides more comprehensive information about the motion and panoptic segmentation of the scene for the cases such as occlusion, where adjacent frames can



(a) Original video (b) Speed up (c) Slow down



(d) Reverse (e) Copy & paste (f) Delete

Figure B5. **Panoptic Visual Odometry (PVO) can Support Many Video Editing Effects of Motion Control.** With PVO, we can manipulate the white car in the original video with different motions and keep the overall consistency of the video. (a) The car in the original video; (b) Speed the car up; (c) Slow the car down, (d) Put the car in reverse, (e) Copy the new similar car keeping the similar motion, (f) Delete the car. The cyan box indicates the original motion pattern, and the other colors indicate our motion manipulation effect.

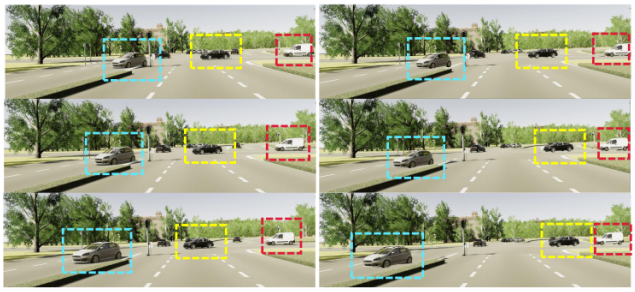


Figure B6. **Multi-Instance Motion Control.** PVO allows a more comprehensive modeling of the scene's motion, panoptic segmentation, and geometric information. PVO can support different motion manipulation of multiple moving objects, even if the camera is also moving. The blue box indicates accelerating the car, the yellow box indicates reversing, and the red box indicates decelerating the car.

be complemented, we can better perform the interaction between different instances, as shown in Fig. B9.

Figure B7. **Generalization Results of Motion Control on Cityscapes Dataset.** PVO demonstrates generalizability in natural scenes.

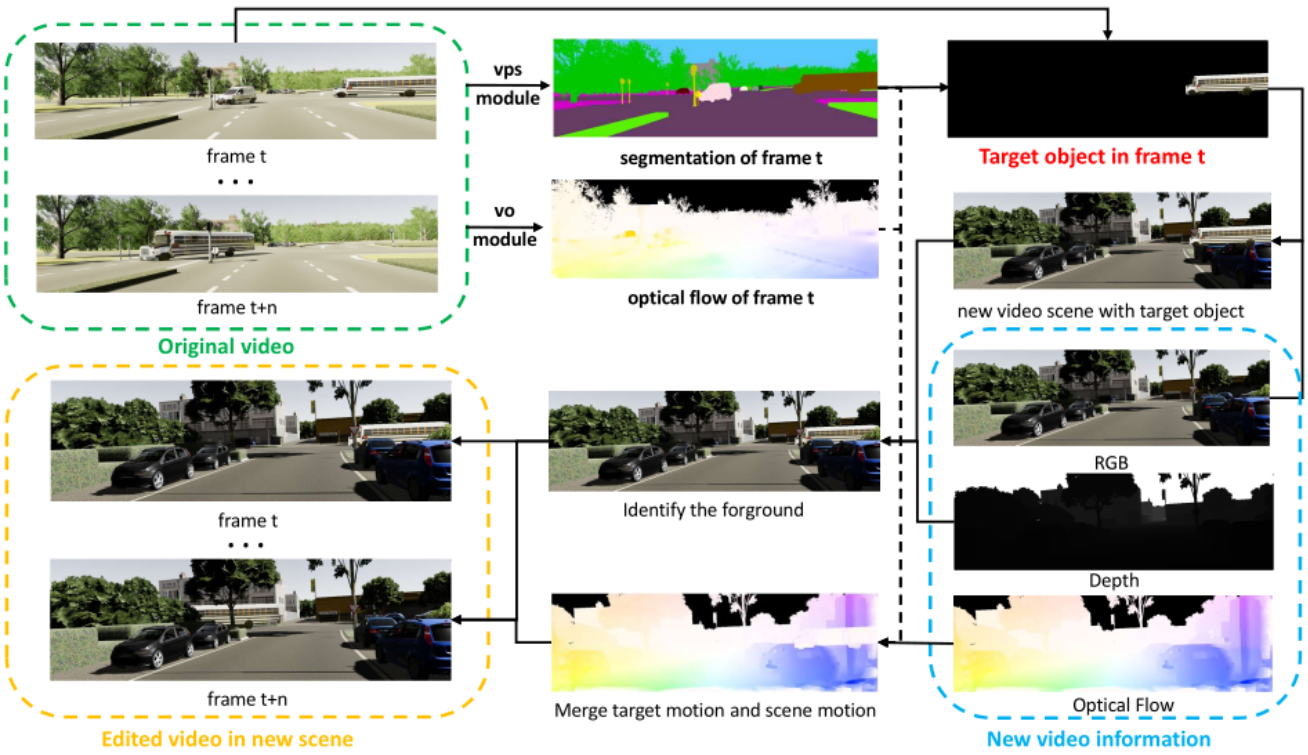
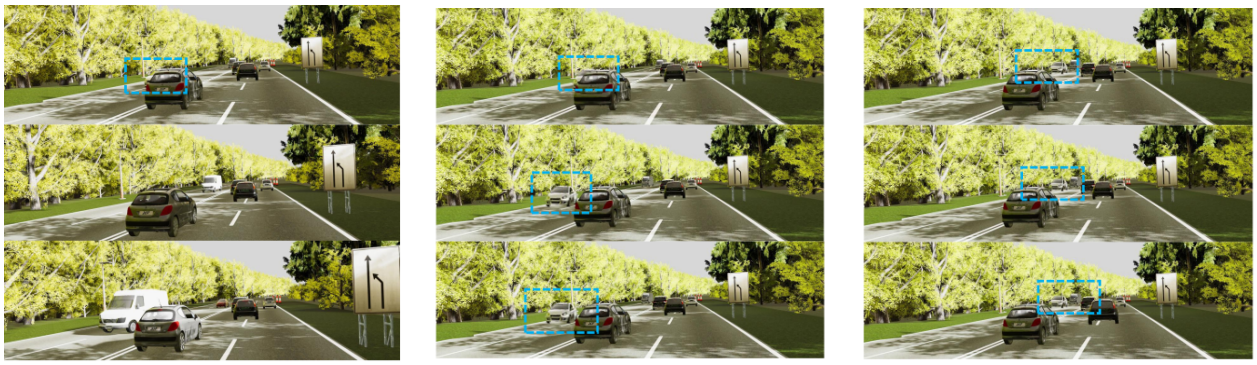


Figure B8. **Video Editing Pipeline with Panoptic Visual Odometry.** PVO comprehensively models the panoptic segmentation and motion information of the entire scene. The motion of dynamic objects can be decomposed into static fields and dynamic fields. We can select an instance and directly manipulate its static fields and dynamic fields to generate a new video. Moreover, the original moving objects can be inserted into the new scene. Some of the occlusion completion and depth checks are taken into account to create a more realistic editing effect.



(a) Original video (b) Pulling cars left and right (c) Pulling cars back and forth

Figure B9. **Multi-Object Occlusion Interaction.** PVO can model the motion, panoptic segmentation, and geometric information of the scene more comprehensively. PVO can support completing multiple mutually occluding moving objects. From left to right, they are a: the original video, b: the occluded part can be restored by pulling the car left and right. c: the occluded part can be restored by pulling the car back and forth.

B.4. Copy and Paste

It's also interesting to copy objects running in other lanes into an empty lane, shown in Fig. B11.

B.5. Delete

If a single lane is overloaded, we can also perform the operation of removing the running vehicle, shown in Fig. B12.

Figure B10. **Ablation study of Video Editing.** Copy the moving objects from the original video to the current moving video. From left to right: the original video, the edited results of the baseline method, and the edited results of the PVO method. PVO can model the complete scene information, such as depth, pose, optical flow, panoptic segmentation, etc. Compared to the baseline method, which edits the scene only by segmentation and optical flow, the results of PVO are more realistic.

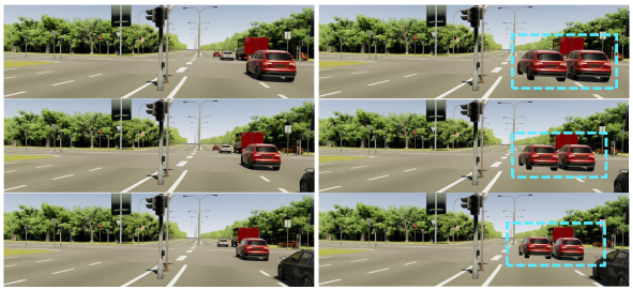


Figure B11. **Copy & Paste.** We can replicate the same moving vehicle in a new lane and keep the video consistent, leveraging the proposed PVO method.

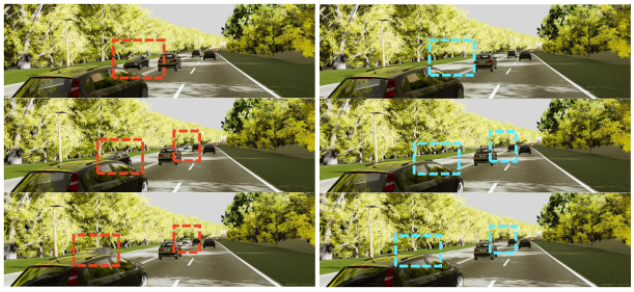


Figure B12. **Delete.** If a single lane is overloaded, we can also perform the operation of removing the moving vehicle using the proposed PVO method. From left to right: the original video, the edited result of removing the car.

C. Discussion

Although PVO can model the panoptic segmentation and motion information of the scene well and support video editing effects such as manipulating the motion patterns of objects. However, it does not take into account the intrinsic physical information of the scene [51], such as lighting, materials, shading, etc., so it cannot make a completely realistic video. In addition, effects [26] related to the objects themselves, such as shadows, cannot be modeled. Fully modeling the movement, panoptic segmentation, effects, and physical properties of the scene is an issue worth exploring. Furthermore, we can explore the application of PVO to autonomous driving simulations to test the robustness of autonomous driving systems by manipulating the motion of objects. We leave this as future work.