JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

PVO: Panoptic Visual Odometry

WED-AM-129

[Weicai Ye[1,2], Xinyue Lan[1,2]]Co-Authors, Shuo Chen[1,2], Yuhang Ming[3,4],

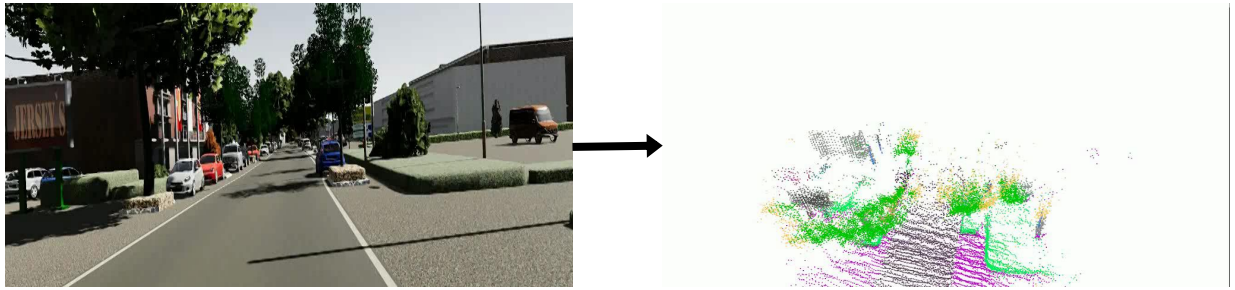
Xingyuan Yu[1,2], Hujun Bao[1,2], Zhaopeng Cui[1], Guofeng Zhang[1,2]*

[1]State Key Lab of CAD & CG, Zhejiang University, [2] ZJU-SenseTime Joint Lab of 3D Vision,

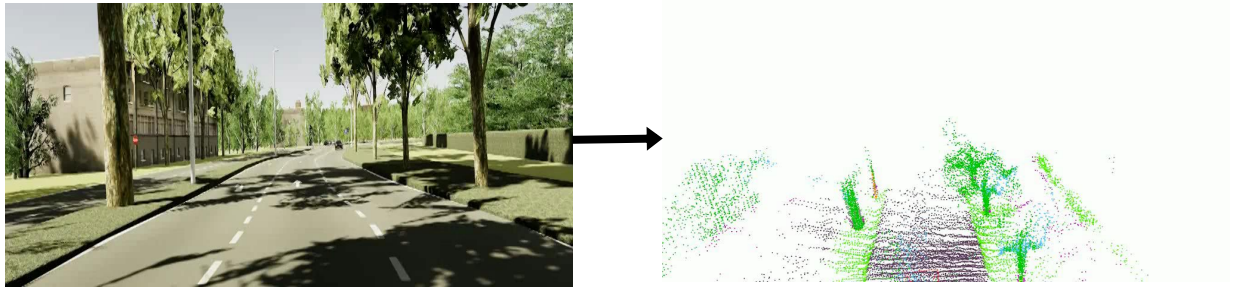
[3]School of Computer Science, Hangzhou Dianzi University, [4]Visual Information Laboratory, University of Bristol

What can we do with PVO?

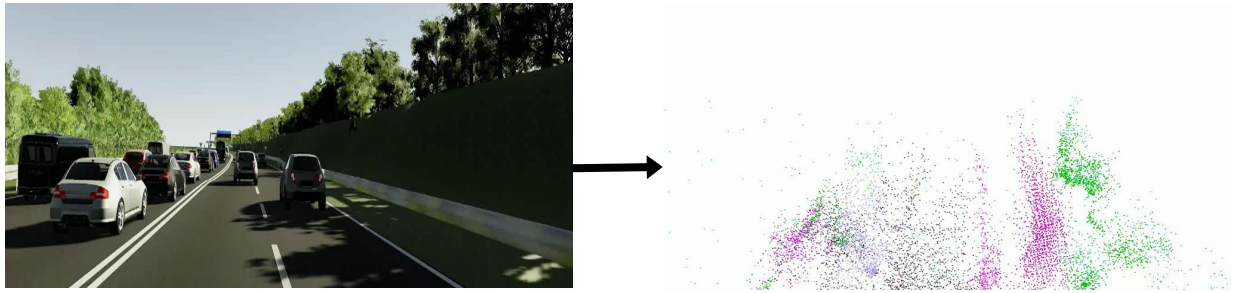
VKITTI2 Seq 01



VKITTI2 Seq 02



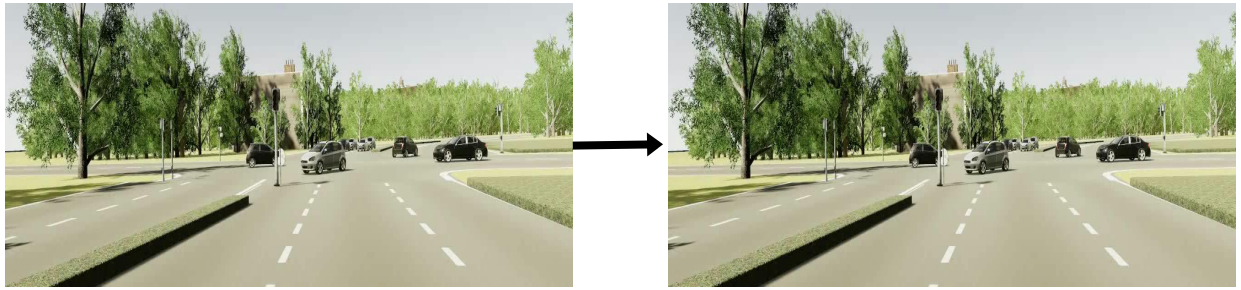
VKITTI2 Seq 20



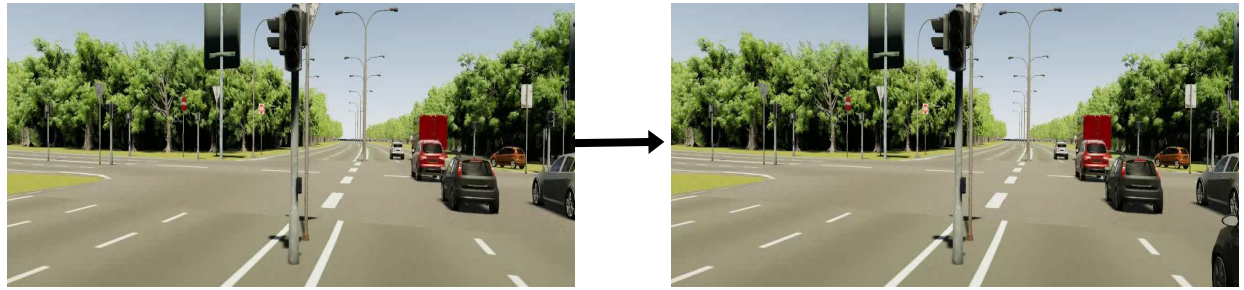
Monocular Videos Panoptic 3D map with Pose

PVO Running Demo

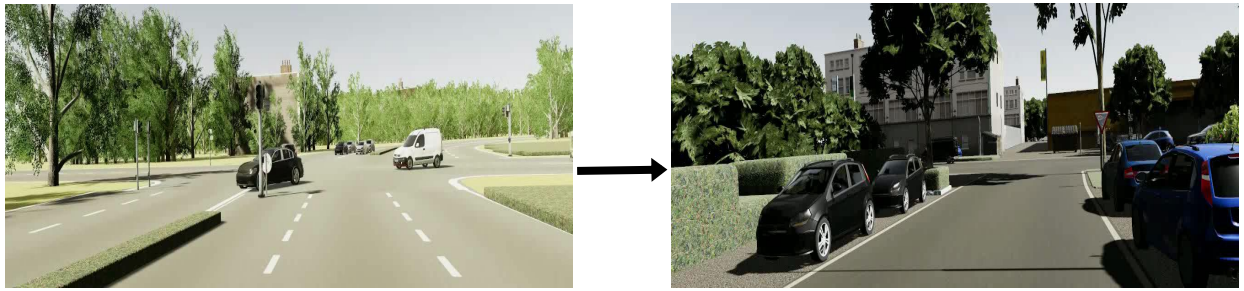
Motion Control



Copy & Paste A Car



Move A Car to New Scene



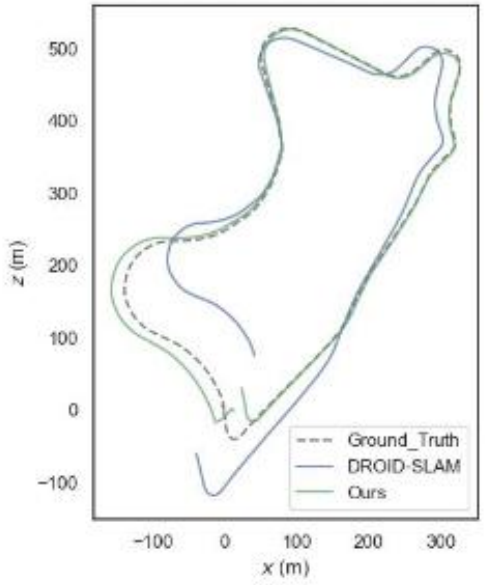
Original Video Edited Video

Video Editing

Related Work

Visual Odometry (VO)

- Input: monocular video
- Output: camera trajectory
- **should recognize** the dynamic objects of the video.



Video Panoptic Segmentation (VPS)

- Input: monocular video
- Output: consistent video segmentation
- **does not explicitly** distinguish dynamic or not

Problem: **without recognizing their relevance**

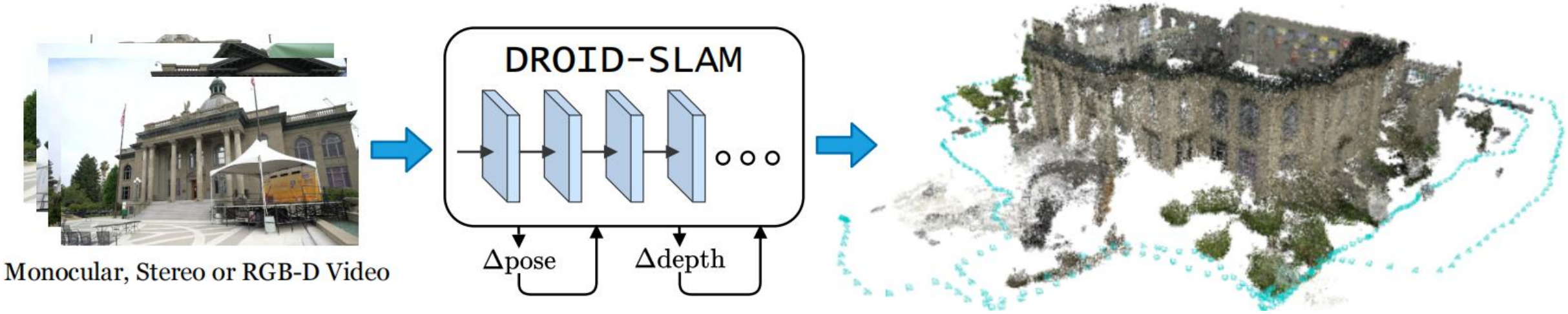
DROID-SLAM

Pros

- **Accurate**: more accurate than prior work
- **Robust**: few catastrophic failures
- **Generalizable**: trained on synthetic monular input, generalizes to real monular,stereo, RGB-D

Cons

- not good enough in **dynamic** scenes
- not robust in indoor scenes
- more memory consumption and loop-closure in long sequences



DROID-SLAM[Teed et al., 2021]

DROID-SLAM

- RAFT
  - Feature Extraction
  - Computing visual similarity
    - Correlation Pyramid

$$\mathbf{C}(g_\theta(I_1), g_\theta(I_2)) \in \mathbb{R}^{H\times W\times H\times W} \quad C_{ijkl} = \sum_h g_\theta(I_1)_{ijh} \cdot g_\theta(I_2)_{klh}$$

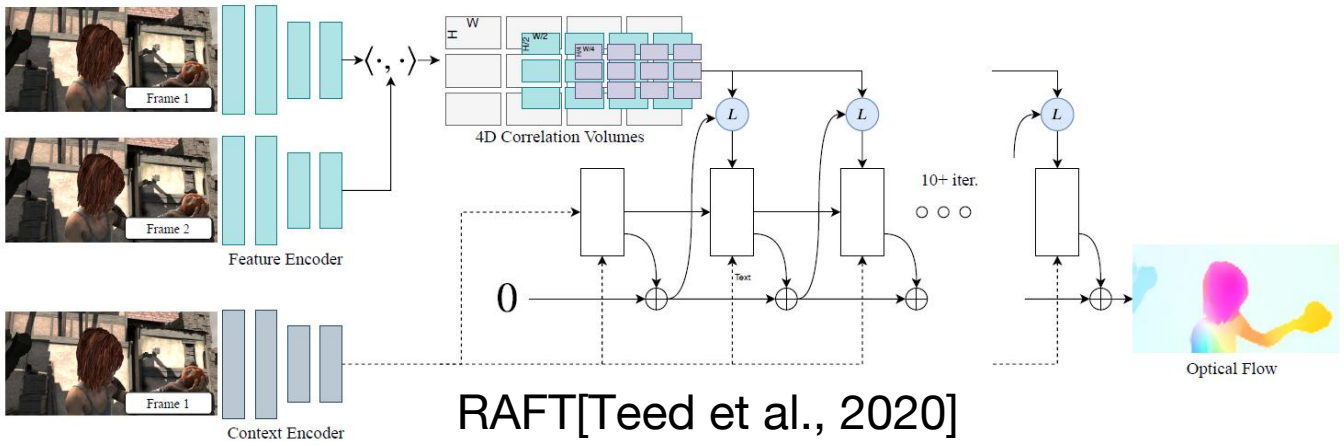
    - Correlation Lookup

$$\mathbf{x}' = (u + f^1(u), v + f^2(v)) \quad \mathcal{N}(\mathbf{x}')_r = \{\mathbf{x}' + \mathbf{dx} \mid \mathbf{dx} \in \mathbb{Z}^2, ||\mathbf{dx}||_1 \le r\}$$

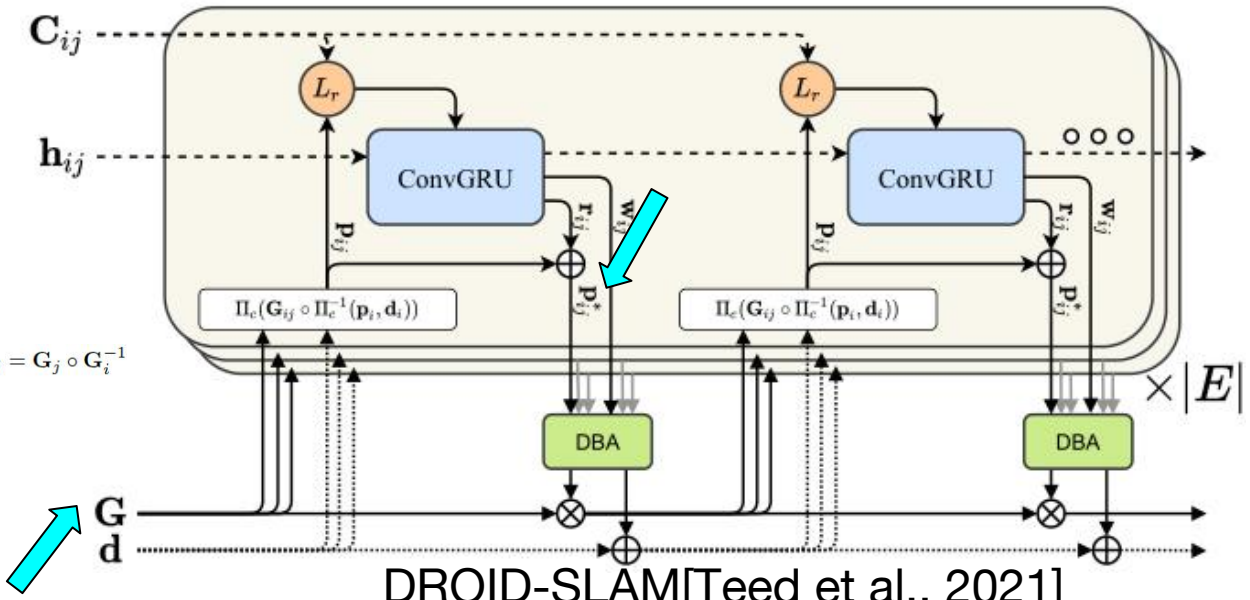
  - Iterative Updates

$$\mathbf{f}_{k+1} = \Delta\mathbf{f} + \mathbf{f}_{k+1} \quad \mathbf{f}_k \to \mathbf{f}^*$$

  - Supervision $\mathcal{L} = \sum_{i=1}^{N} \gamma^{N-i} ||\mathbf{f}_{gt} - \mathbf{f}_i||_1$
- Updates Operator
  - correspondence: $\mathbf{p}_{ij} = \Pi_c(\mathbf{G}_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}_i)), \quad \mathbf{p}_{ij} \in \mathbb{R}^{H\times W\times 2} \quad \mathbf{G}_{ij} = \mathbf{G}_j \circ \mathbf{G}_i^{-1}$
  - update: $\mathbf{p}^*_{ij} = \mathbf{r}_{ij} + \mathbf{p}_{ij} \quad \mathbf{w}_{ij} \in \mathbb{R}^{H\times W\times 2}_+$
  - dense BA: $\mathbf{E}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \|\mathbf{p}^*_{ij} - \Pi_c(\mathbf{G}'_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}'_i))\|^2_{\Sigma_{ij}} \quad \Sigma_{ij} = \text{diag}\, \mathbf{w}_{ij}$
  - trainning loss: pose, flow, depth loss



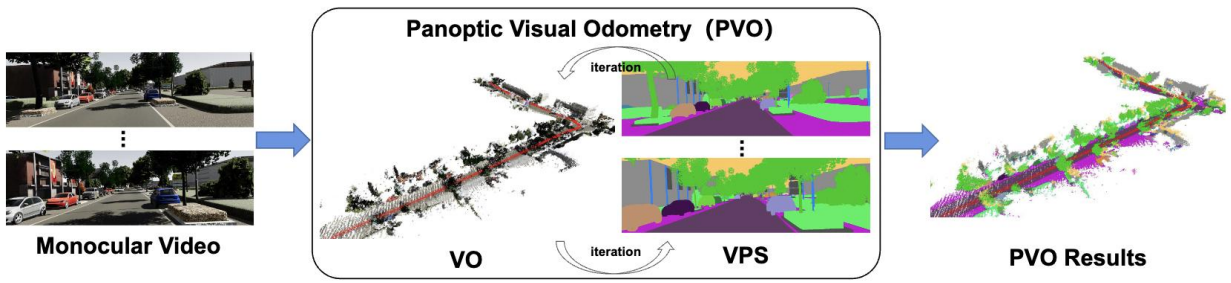
RAFT[Teed et al., 2020]



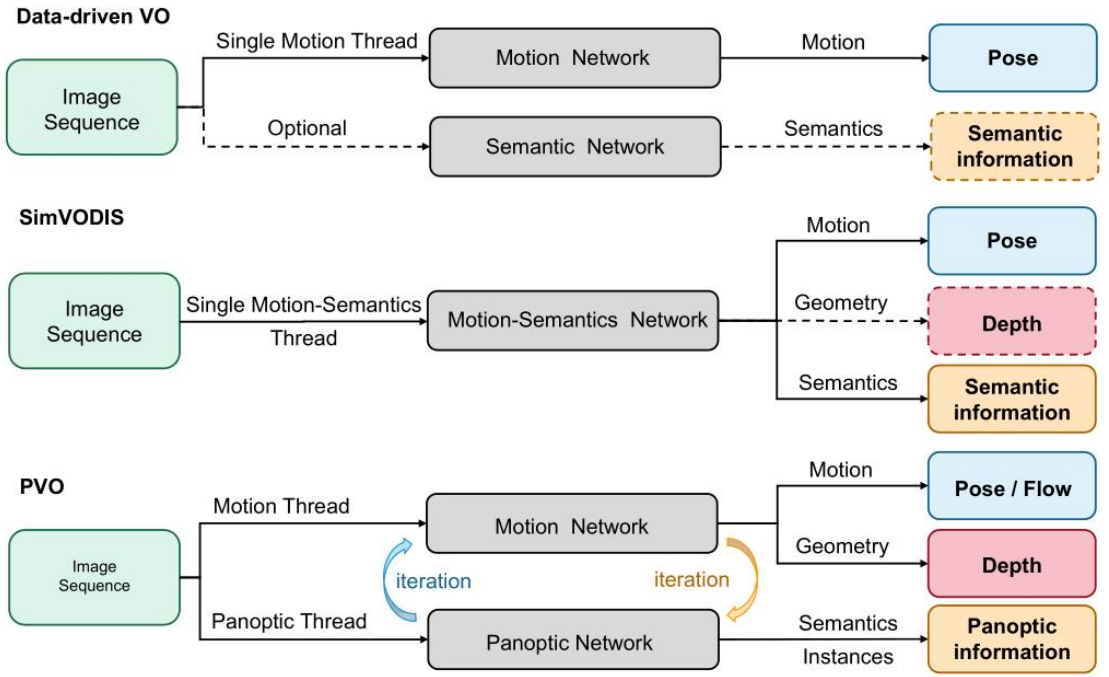
DROID-SLAM[Teed et al., 2021]

- **Problem: dynamic objects lead to ambiguity in optical flow estimation**
- **No correlation constraint on the weights of the pixels of each instance.**

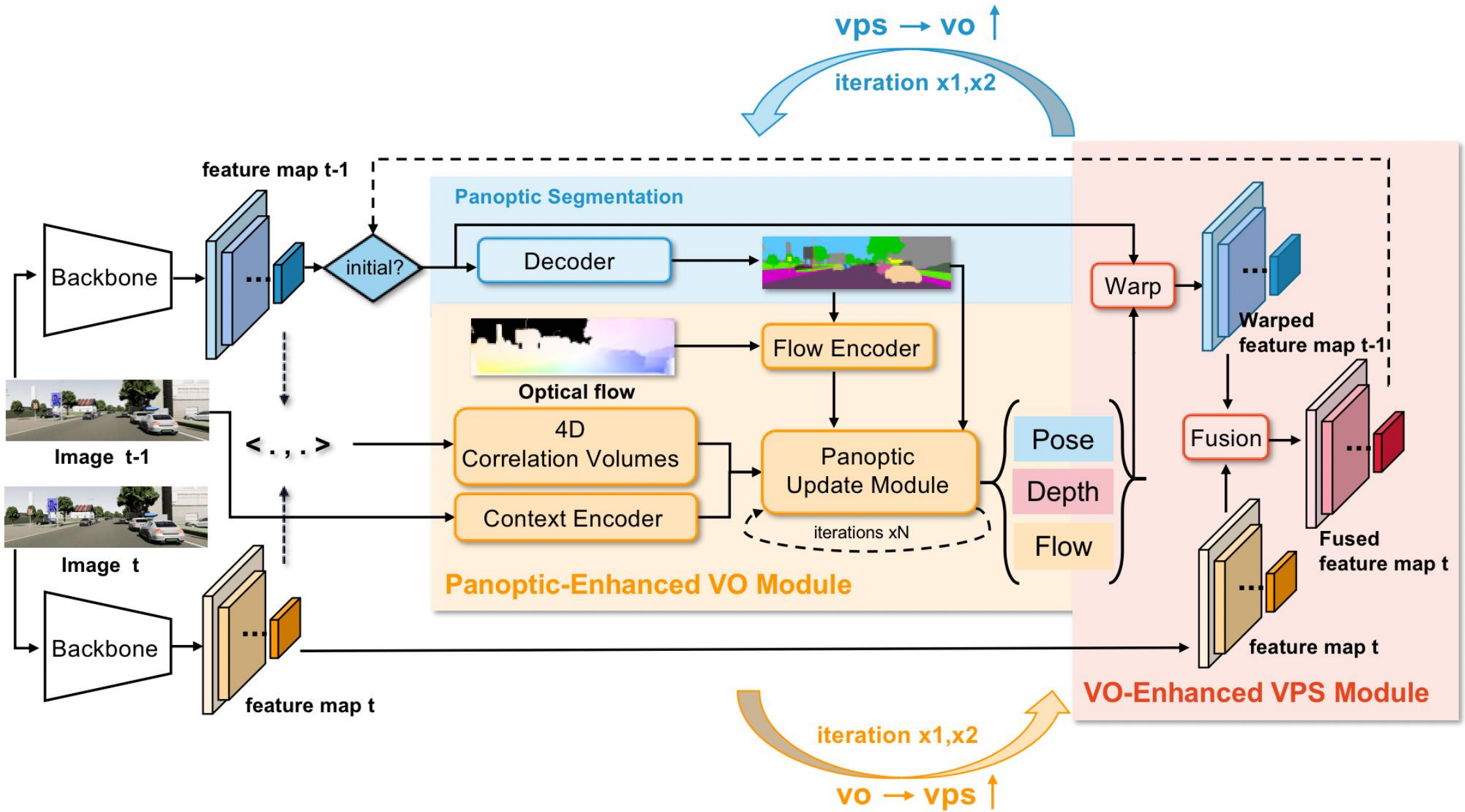
Other Hybrid Methods vs. PVO

- motion-semantics network:
  - loss functions may contradict each other.
- PVO:
  - **Unify VPS and VO** to model the scene comprehensively
  - 

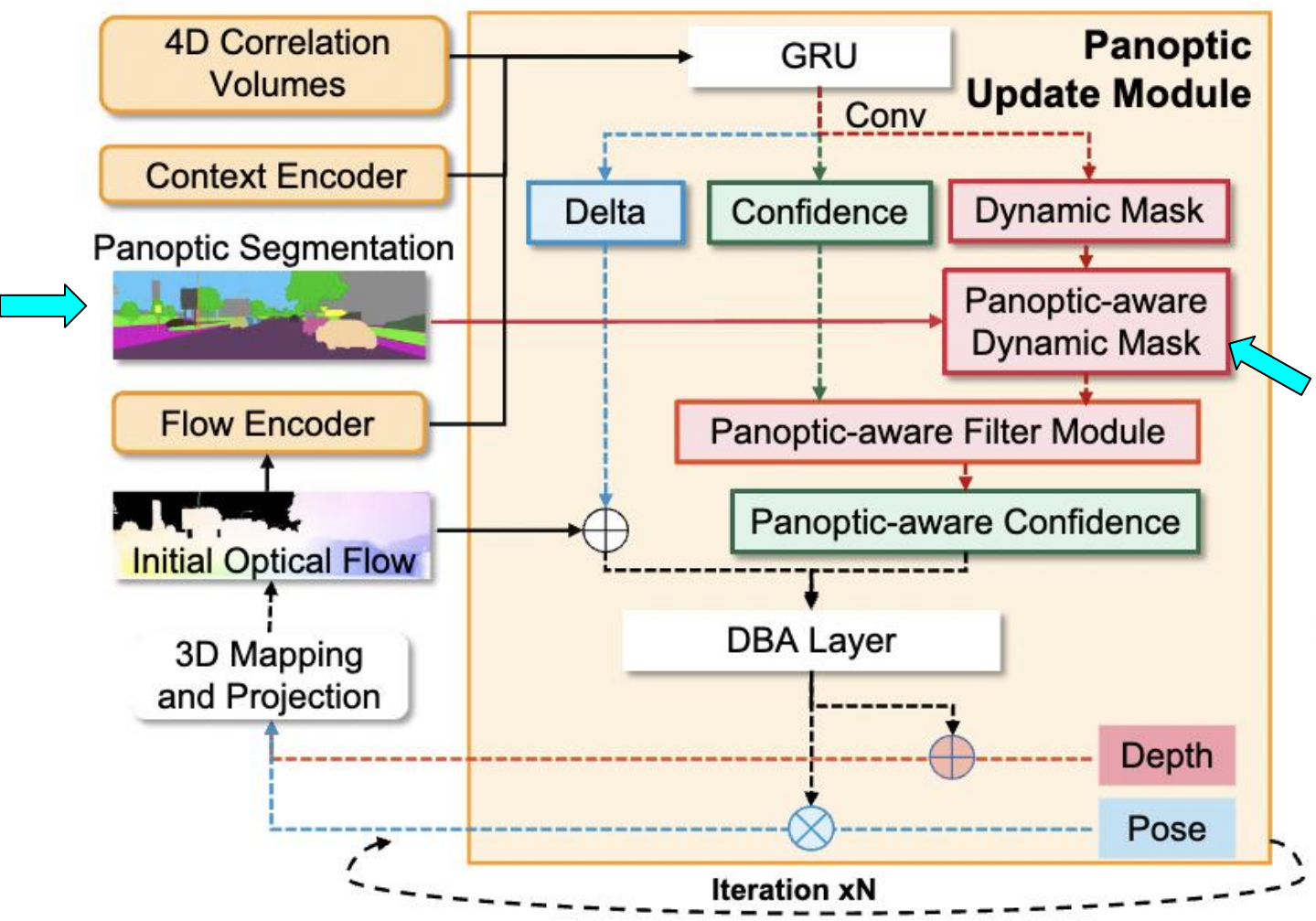
  - Mutually beneficial through **recurrent iterative optimization**
  - VPS helps VO: adjust weight for stuff and thing
  - VO helps VPS: tracking and fusion **from 2D to 3D**



PVO Pipeline



Panoptic-Enhanced VO Module



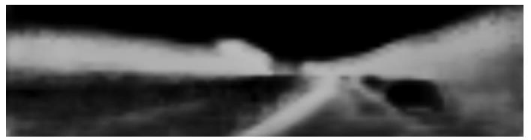
- Panoptic-Aware Dynamic Mask
  - stuff-->static
  - thing with high dynamic probability --> dynamic (each instance)

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

- Panoptic-Aware Confidence
  - remove dynamic interference
  - keep the static feature


RGB Image


DROID-SLAM Confidence


PVO Confidence

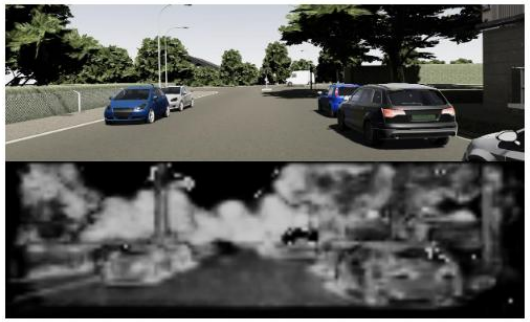
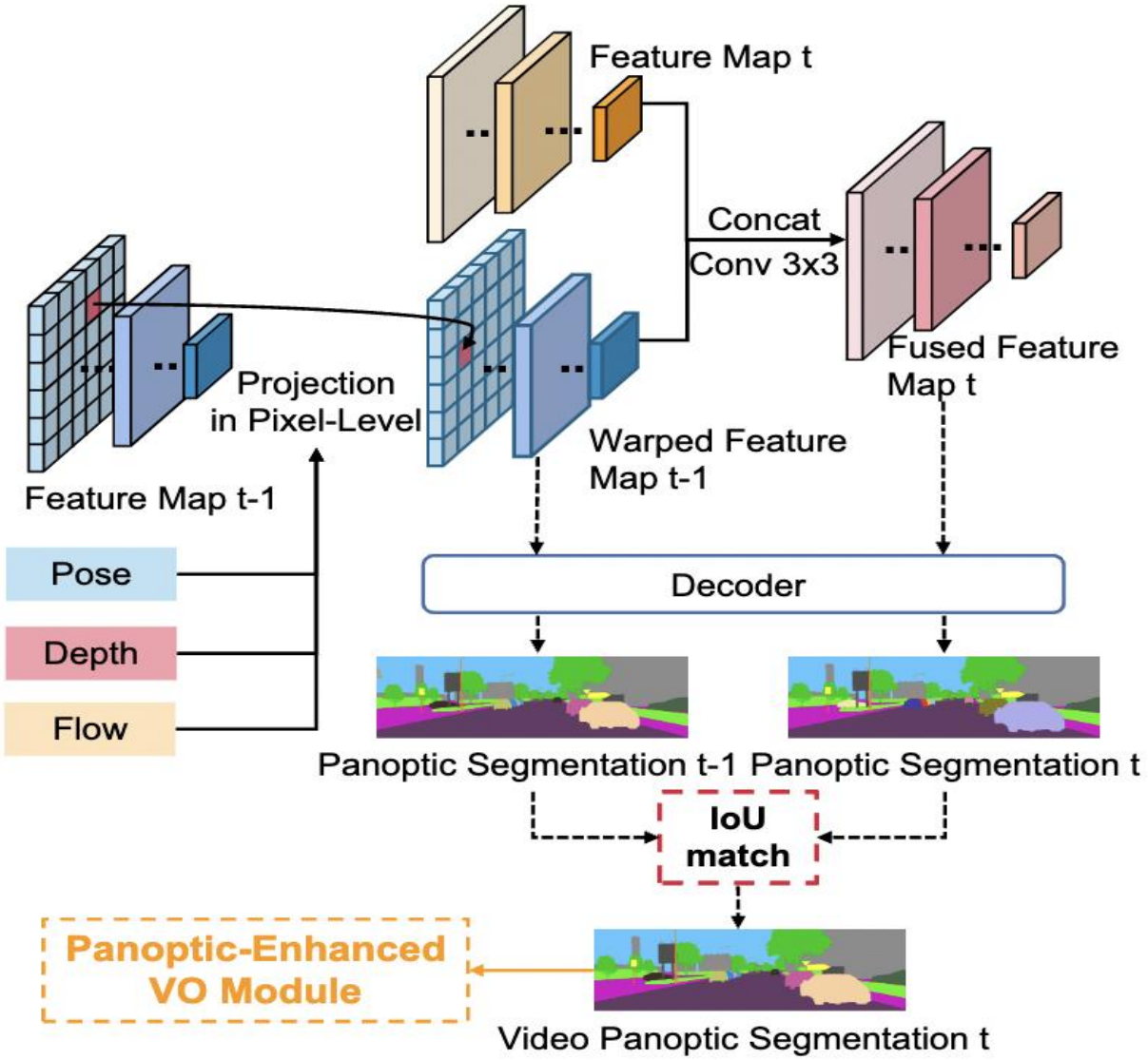


Figure 1. **Dynamic Probability of Parked Cars.**

VO-Enhanced VPS Module



- VO-Aware Online Fusion

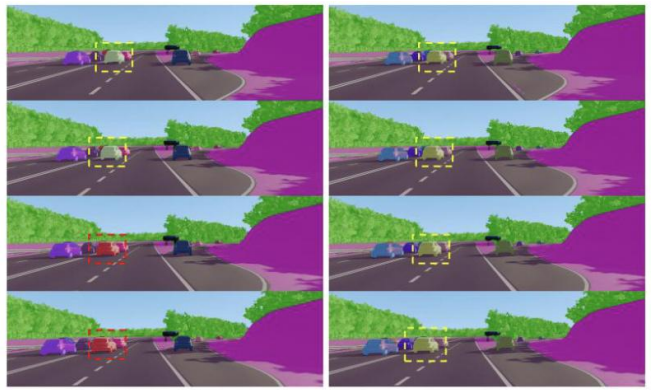
Feature Alignment Loss [24]. We employ a feature alignment loss to minimize the distance between $\mathbf{z_t}^*$ and $\hat{\mathbf{z}}_\mathbf{t}$ in latent space:

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

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 $\hat{\mathbf{z}}_\mathbf{t}$:

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



VPS Baseline VO-Enhanced VPS

Robust in occlusion case

SOTA VO Results

VKITTI2

- Robust in Dynamic Scene

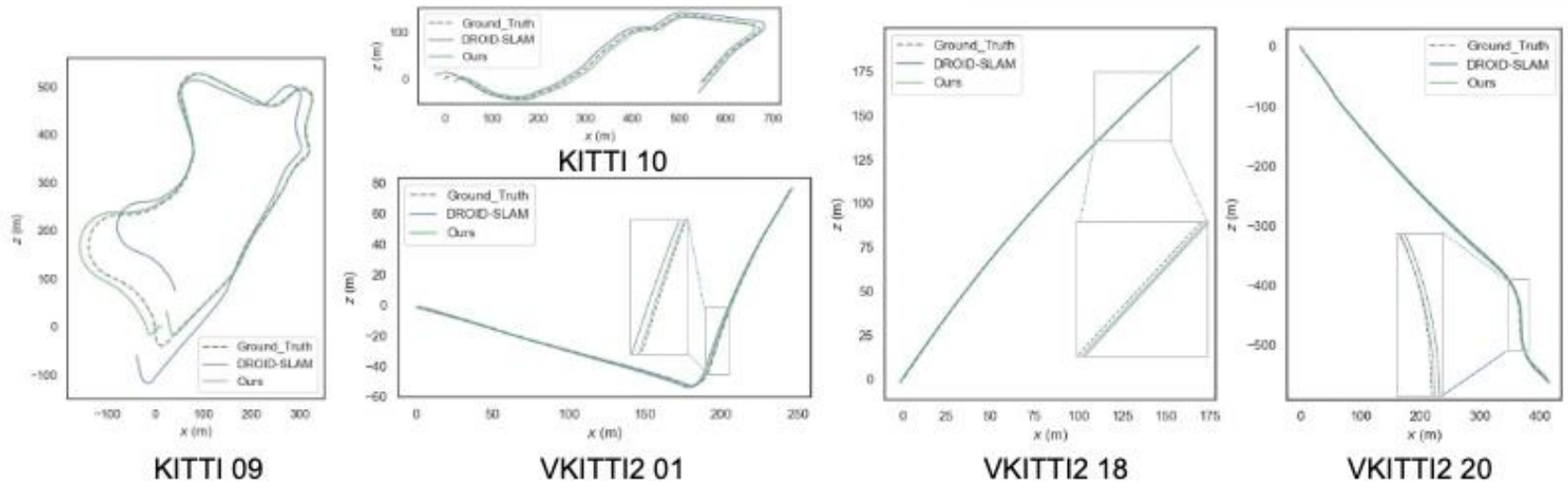
Monocular	01	02	06	18	20	Avg
DROID-SLAM [34]	1.091	**0.025**	0.113	1.156	8.285	2.134
Ours (VPS->VO w/o filter)	0.384	0.061	0.116	0.936	5.375	1.374
Ours (VPS->VO)	0.374	0.057	0.113	0.960	3.487	0.998
Ours (VPS->VO x2)	0.371	0.057	0.113	0.954	3.135	0.926
Ours (VPS->VO x3)	**0.369**	0.055	**0.113**	**0.822**	**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

TUM RGBD Dynamic

- Outperform DROID-SLAM

Sequences		Trans. RMSE of trajectory alignment [m]				
		DVO SLAM [16]	ORB-SLAM2 [28]	PointCorr [6]	DROID-SLAM [34]	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

- Trajectory Comparison



KITTI 09 | KITTI 10 | VKITTI2 01 | VKITTI2 18 | VKITTI2 20

SOTA VPS Results

- Ablation study on VKITTI2

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

- VPS on Cityscapes, VIPER

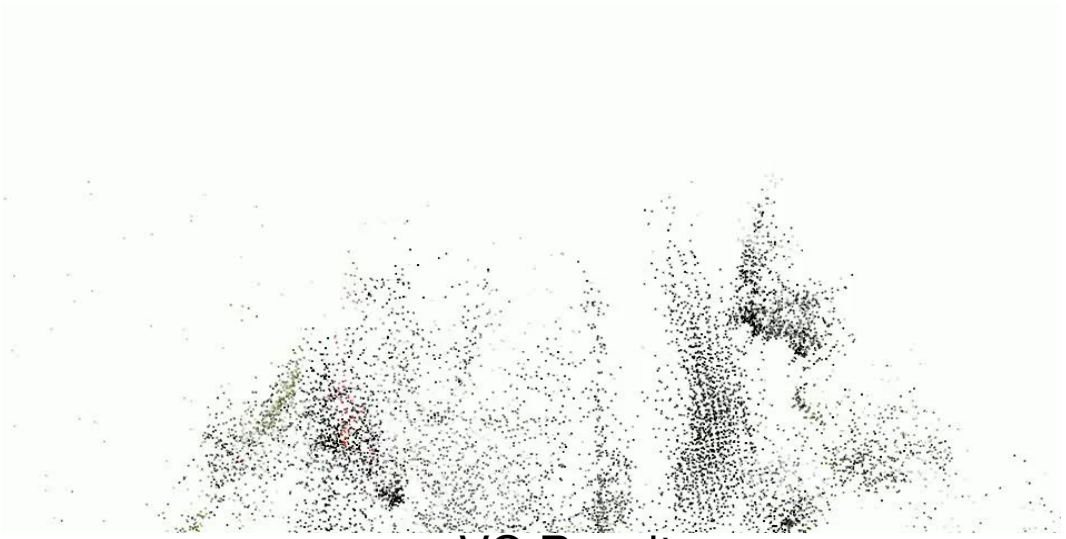
Methods on **Cityscapes-VPS *val***	Temporal window size				VPQ	FPS
	k = 0	k = 5	k = 10	k = 15		
VPSNet-Track	63.1 / 56.4 / 68.0	56.1 / 44.1 / 64.9	53.1 / 39.0 / 63.4	51.3 / 35.4 / 62.9	55.9 / 43.7 / 64.8	4.5
VPSNet-FuseTrack	64.5 / 58.1 / 69.1	57.4 / 45.2 / 66.4	54.1 / 39.5 / 64.7	52.2 / 36.0 / 64.0	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 [22] + 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

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 [22] + 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

Qualitative Results

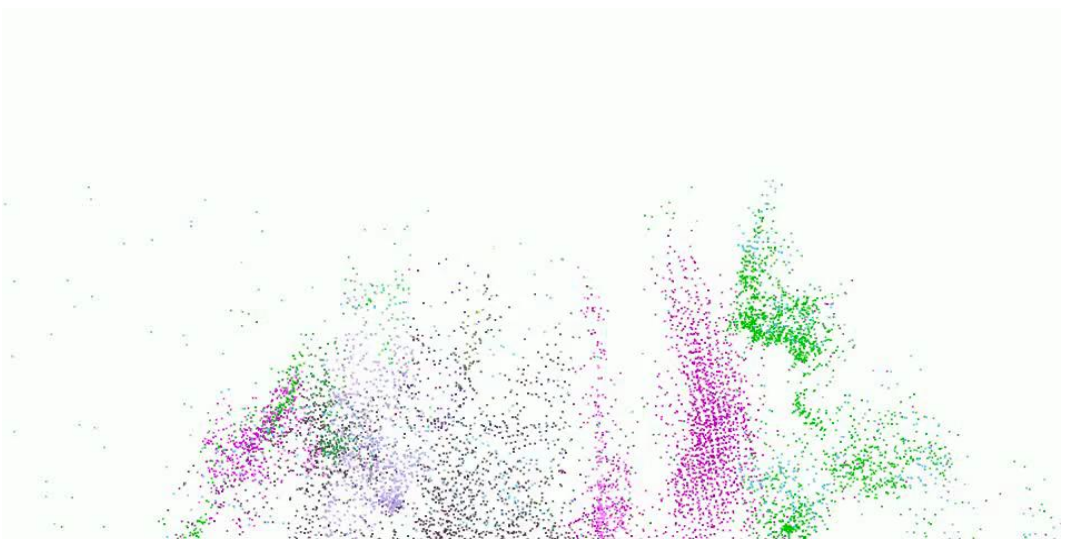
VKITTI2 Seq 01

Monocular Video



VO Result

VPS Result



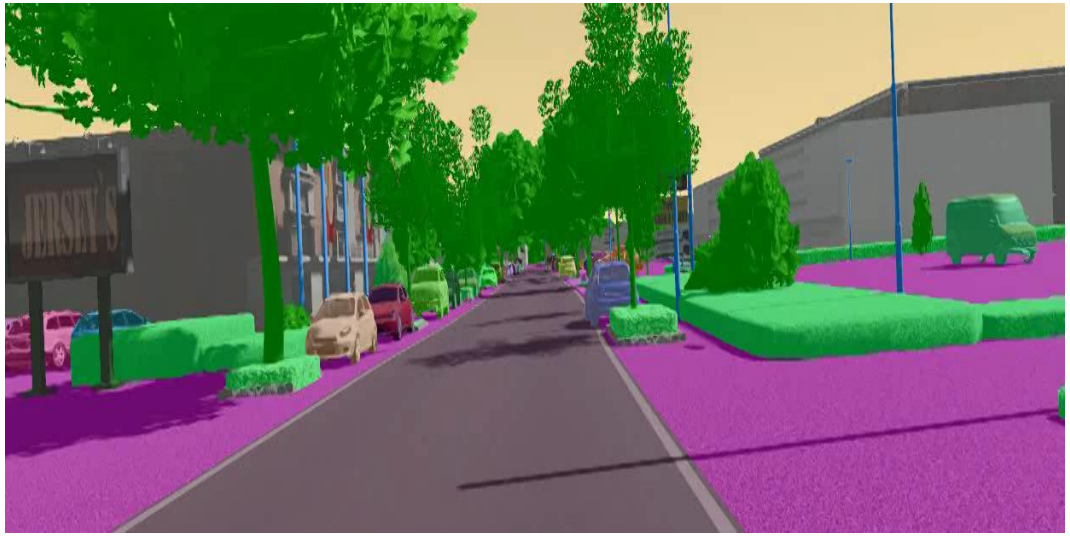
Final PVO Result

Qualitative Results

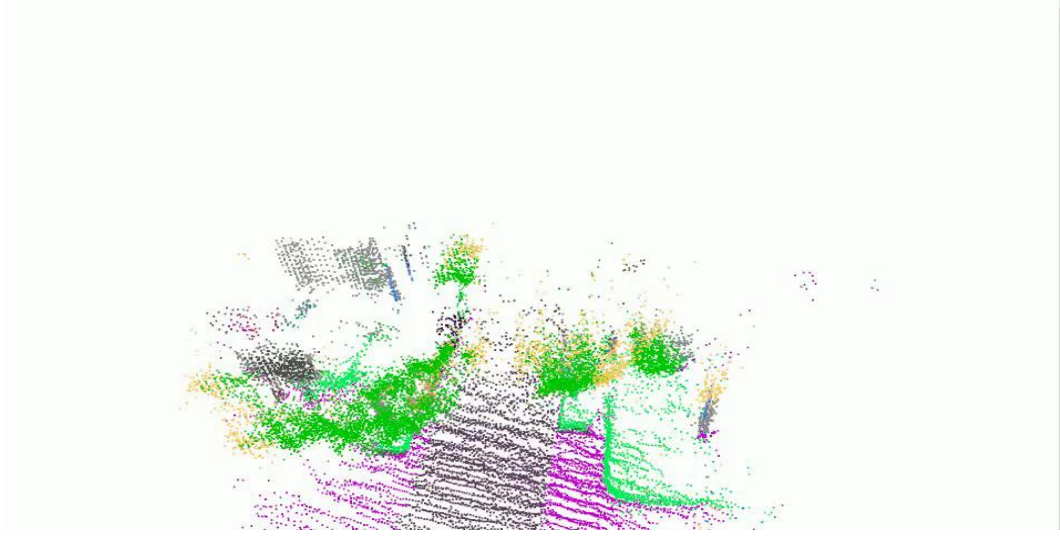
VKITTI2 Seq 20

Monocular Video

VO Result

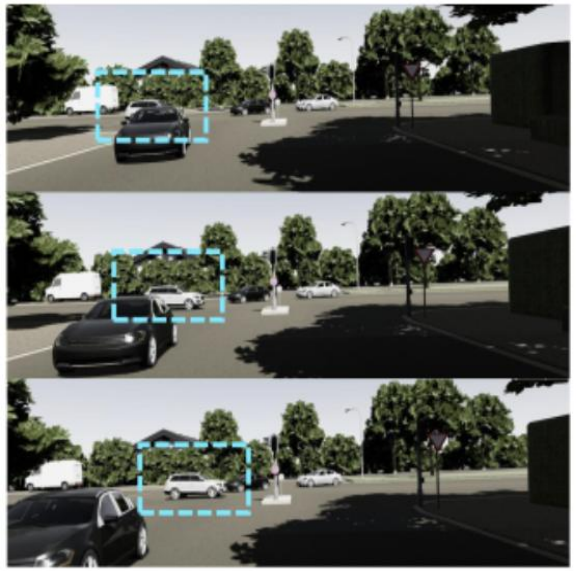


VPS Result

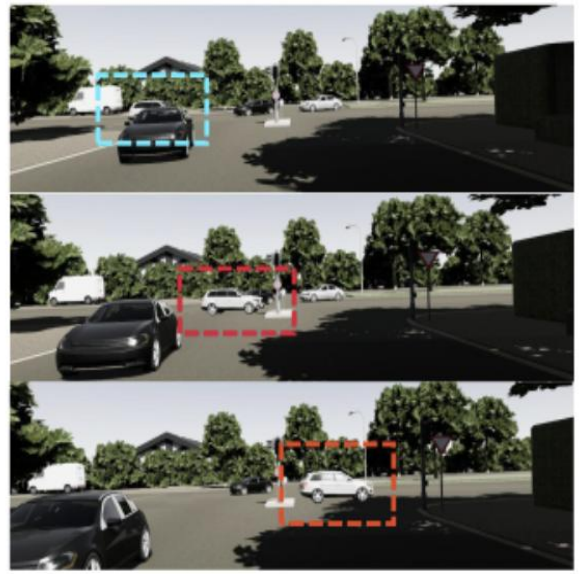


Final PVO Result

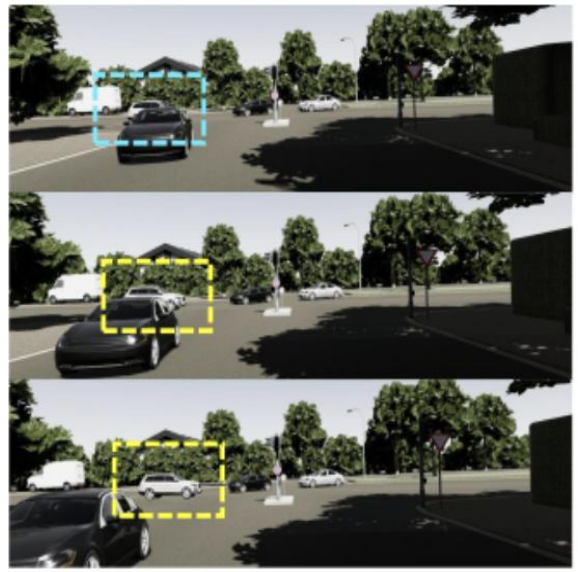
Video Editing Applications



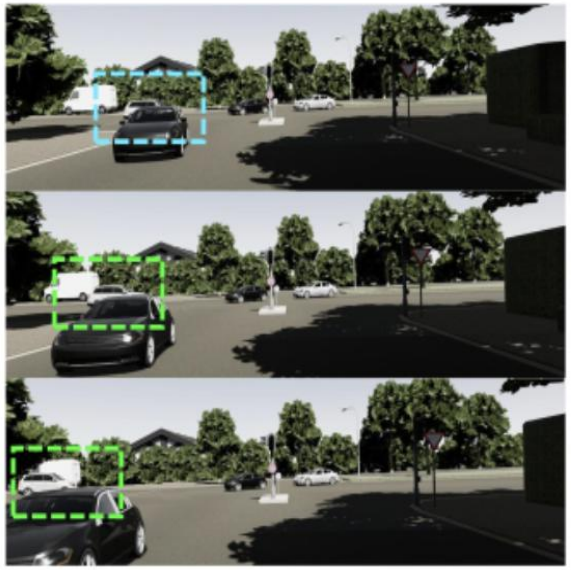
(a) Original video



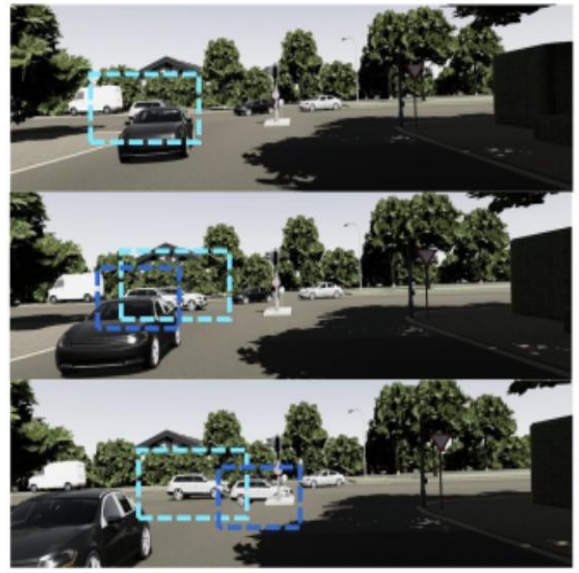
(b) Speed up



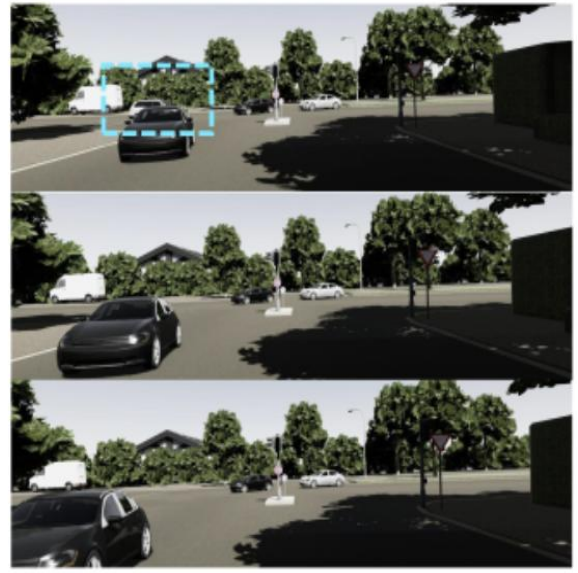
(c) Slow down



(d) Reverse

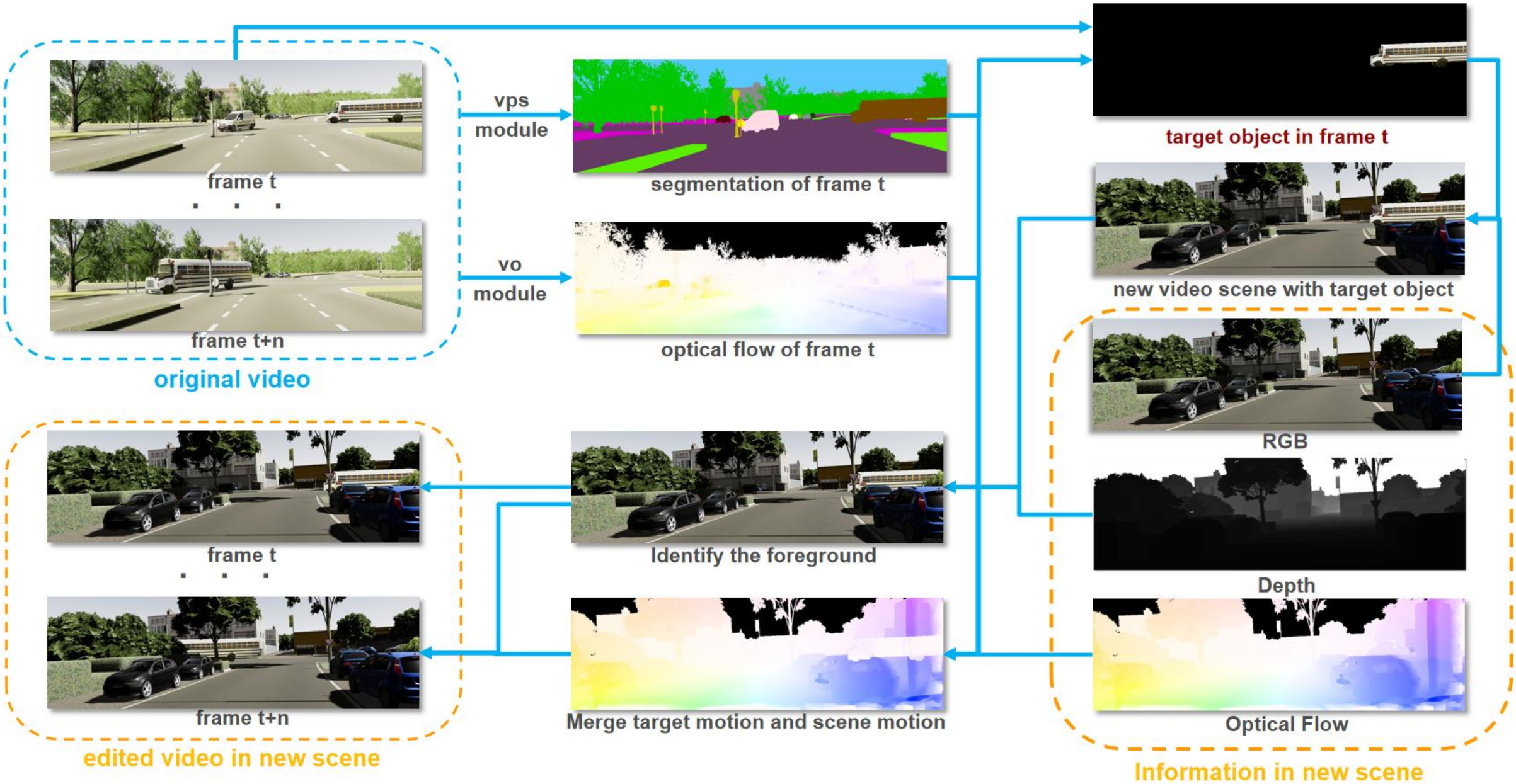


(e) Copy & paste



(f) Delete

Video editing pipeline for motion control with PVO



Copy the moving objects from original video to new moving video

Original Video

Baseline Edited Video

PVO Edited Video

Multi-Instance Motion Control

Original Video

PVO Edited Video

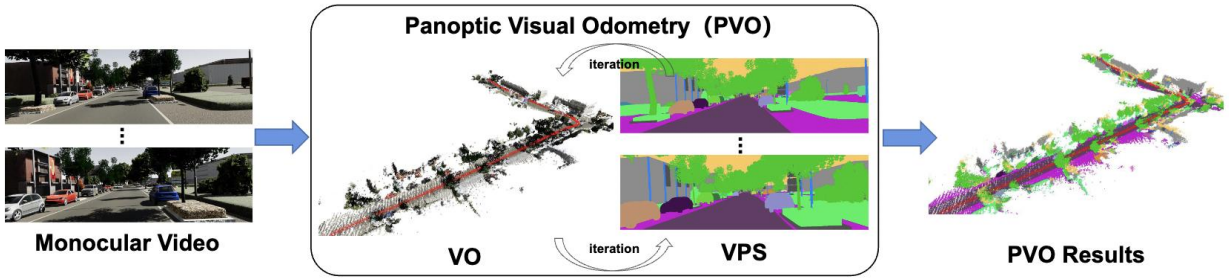
Copy and Paste a car

Original Video

PVO Edited Video

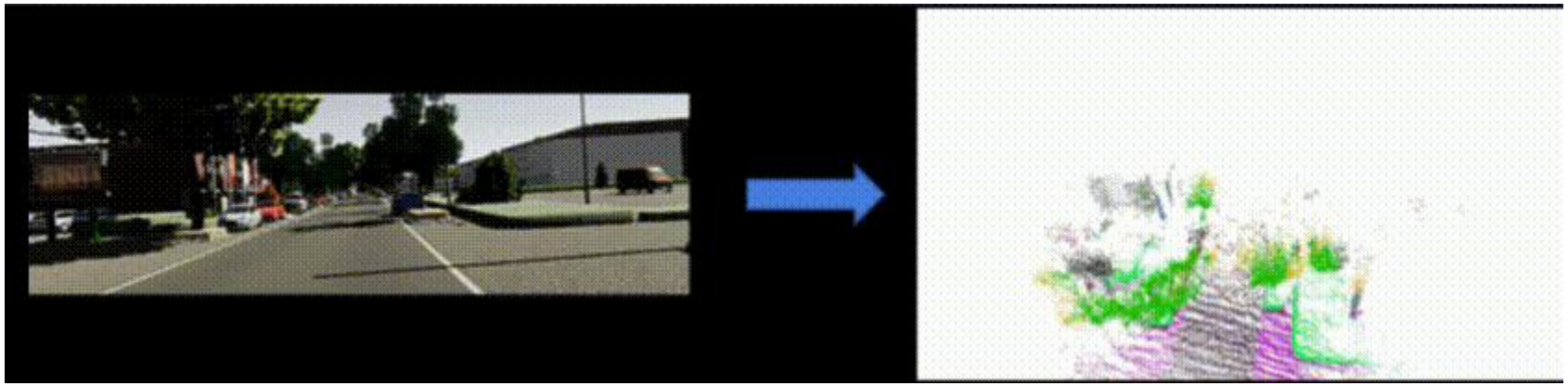
Take-home message

- PVO can **unify VO and VPS**, and make them mutually reinforcing by **recurrent iterative optimization**



- PVO can perform robustly than DROID-SLAM in **dynamic** scenes
- PVO can be applied to video editing for **motion control**.

- Future work:
  - **Loop closure** in SLAM
  - **Low-cost** SLAM
  - Apply PVO to **AutoDriving Simulation**.



Thank you for your watching!

code: https://github.com/zju3dv/pvo

homepage: zju3dv.github.io/pvo/