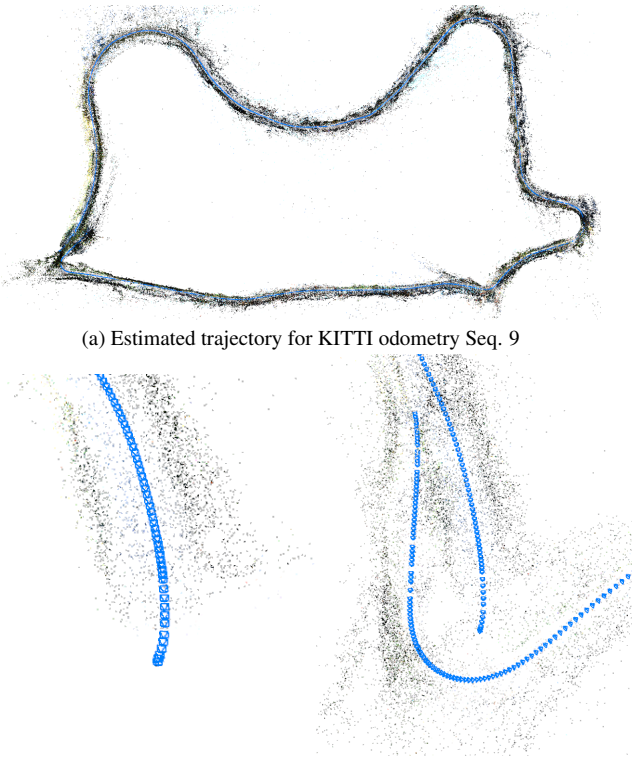
# Weakly Supervised End2End Deep Visual Odometry

Amin Abouee, Ashwanth Ravi, Lars Hinneburg, Mateusz Dziwulski,
Florian Ölsner, Jürgen Hess, Stefan Milz
Spleenlab GmbH, Germany
firstname.lastname@spleenlab.ai

Patrick Mäder
Technical University Ilmenau, Ilmenau, Germany
patrick.maeder@tu-ilmenau.de

## Abstract

*Visual odometry is an ill-posed problem and utilized in many robotics applications, especially automated driving for mapless navigation. Recent applications have shown that deep models outperform traditional approaches especially in localization accuracy and furthermore significantly reduce catastrophic failures. The disadvantage of most of these models is a strong dependence on high-quantity and high-quality ground truth data. However, accurate and dense depth ground truth data for real world datasets is difficult to obtain. As a result, deep models are often trained on synthetic data which introduces a domain gap. We present a weakly supervised approach to overcome this limitation. Our approach uses estimated optical flow for training that can be generated without the need for high-quality dense depth ground truth. Instead, it only requires ground truth poses and raw camera images for training. In the experiments, we show that our approach enables deep visual odometry to be efficiently trained on the target domain (real data) while achieving state-of-the-art performance on the KITTI dataset.*



(a) Estimated trajectory for KITTI odometry Seq. 9

(b) Initialization

(c) End of Trajectory

Figure 1. This figure presents a comprehensive analysis of our proposed method's performance on sequence 9 from KITTI odometry dataset. (a) displays the estimated trajectory and its sparse map, showcasing accurate localization and mapping throughout the sequence. (b) represents the initialization phase of our VO system and (c) illustrates the end of the trajectory, featuring a closed loop and demonstrating the high precision of our system, as indicated by the small size of the gap between the start and end points when compared to the length of the trajectory.

## 1. Introduction

Visual odometry (VO) is a crucial aspect of robotics that enables machines to measure the ego-motion of a camera and uses the relative motion between images to estimate the camera's global pose [14, 16]. The use of different sensors, including cameras, depth cameras, IMUs, and LiDAR sensors, has been widely explored in visual odometry estimation. Camera-based methods have emerged as a preferred choice due to their low cost, low power requirements, and the ability to provide useful complementary information compared to other sensors.

In this work, we address the visual odometry (VO) prob-

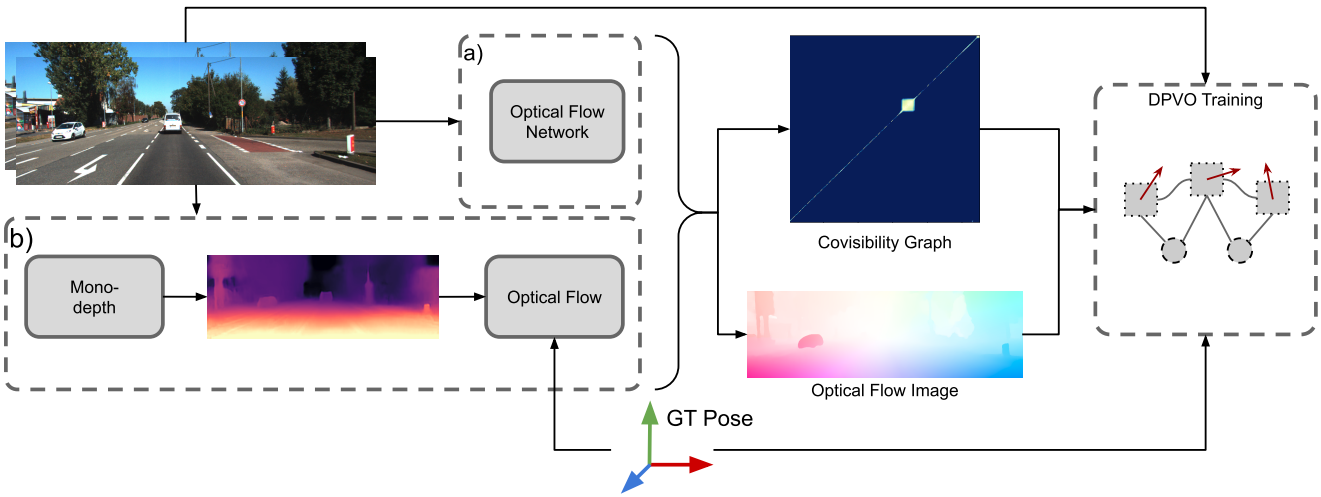



Figure 2. Overview of the system: Training of DPVO (right) requires optical flow, the co-visibility graph, ground truth poses as well as the camera images (top left) as input. Instead of relying on ground truth dense depth to pre-compute the optical flow we propose two extensions: a) use of a pre-trained optical flow estimator or b) utilization of a self-supervised monocular depth prediction network.

lem using a monocular camera, where the goal is to estimate the 6DoF motion of a moving camera. Geometric visual odometry is a traditional approach that estimates camera motion by matching features between consecutive frames and applying geometric constraints. However, it is limited by its dependence on accurate feature tracking and can struggle in low-texture environments. In recent years, deep learning-based methods have emerged as a promising alternative that can cope with challenging environments more effectively. Deep Patch Visual Odometry (DPVO) [22], a recently proposed supervised visual odometry system has shown improved accuracy and robustness over traditional methods on variety of data-sets. When benchmarked on synthetic and real-world datasets the method shows superior results on the training domain (synthetic data). However, DPVO's performance is highly dependent on ground truth pose and dense depth data which is hard to obtain for real world datasets. Training on simulated data as an alternative, however, introduces a domain gap and limits its effectiveness when applied to real-world environments.

In this work, we propose two weakly-supervised training techniques that enable efficient training of these networks on real-world datasets, eliminating the requirement for costly dense depth ground truth annotations. In addition, we introduce an innovative approach to reduce the runtime complexity of DPVOs preprocessing stage while maintaining nearly unchanged predictive performance. We validate our approach on the KITTI benchmark dataset (see Fig. 1 for an example result). Our experiments demonstrate that our proposed training approach allows the model to outperform its previous baseline and achieve state-of-the-art performance.(see Fig. 3)

## 2. Related Work

**Geometric Methods**: Geometric VO is a method for estimating the motion of a camera in 3D space based on geometric principles. In a classical geometric VO system [11, 16], the main components include feature detection, feature matching or tracking, motion estimation using triangulation, and local optimization through bundle adjustment. Additionally, a keyframe-based mechanism is often employed to enable a more reliable and traceable motion estimation over long periods of time. There are two main methods, namely indirect (feature-based) and direct methods. The feature-based method [7, 13] involves detecting and tracking discrete interest points across frames, while the direct method [4] solves an energy minimization problem based on the intensities or the feature warp error considering the entire image.

**Supervised Methods**: With the development of deep neural networks, approaches based on end-to-end learning have been proposed to solve the VO problem [22, 26, 27, 30, 31]. These methods rely on supervised loss functions using ground truth data like pose, depth or optical flow to estimate the camera's relative 6DoF pose from two consecutive image frames. Recent approaches use CNNs to jointly predict scene depth and the camera pose by exploiting the geometric relationship between structure and motion [19, 20, 35].

**Self-Supervised Methods**: There also exists a growing interest in self-supervised training for VO approaches since they reduce the amount of necessary annotated training data [2, 9, 15, 25, 34, 36]. These algorithms rely on photometric consistency between adjacent frames as their primary supervisory signal. Although they attain good performance for single-view depth estimation, the performance of ego-

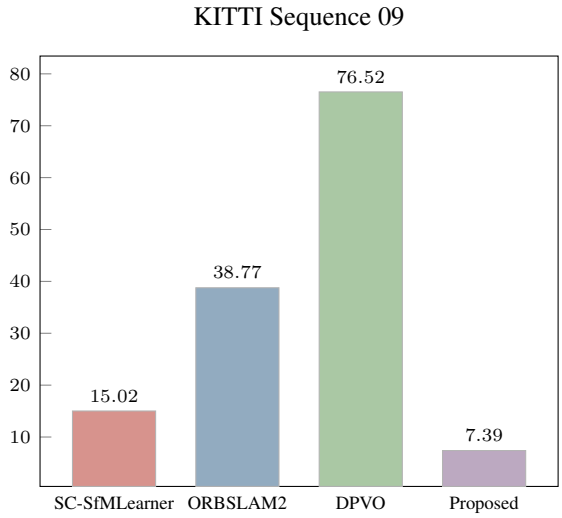



Figure 3. Absolute Trajectory Error (ATE) of our proposed method on KITTI Odometry Seq. 09. Our method outperforms the other state of the art VO methods.

motion estimation is still substantially lower compares to standard VO approaches. Bian *et al*. [2] argue that pose networks cannot offer complete camera trajectories over lengthy sequences due to the uneven scale of per-frame estimations. As a result, they suggest a geometry consistency requirement.

In this paper, we present a novel weak flow supervision approach for VO that predicts camera pose and sparse patch depth given a sequence of input images. We base our proposed model on DPVO, a supervised learning approach that combines the strengths of both classical techniques and deep learning to improve performance and robustness in challenging scenarios.

## 3. Method

In this section, we briefly explain the main features of DPVO before describing our approach.

### 3.1. Preliminaries: DPVO

DPVO operates on a sequential input of images $I_t$ and estimates the camera pose $T \in SE(3)$ as well as the inverse depth of $m$ patches of size $3 \times 3$ for each image $t$. Both camera pose and inverse depths are updated iteratively as new frames are processed. For training, a frame graph $G = (V, E)$ is constructed to indicate co-visibility between frames, where the nodes are input images and the edges $(i, j) \in E$ imply that the images $I_i$ and $I_j$ have overlapped views. The VO algorithm iteratively runs in three steps:

1. **Feature and Patch Extraction**: Similar to RAFT [21], the feature and patch extraction uses two residual networks with 4 blocks to extract features for matching and context awareness on $1/4$ of the original input image resolution. In the next step $m$ patch positions are randomly sampled from both feature and context maps and stored in the frame and patch feature maps for subsequent use.
2. **Update Operator**: This operator is used for joint pose and patch optimization. The proposed operator optimizes both, the poses and patches using correlation features and message passing. First, for each edge $(i, j)$ in the patch graph, the correlation features are computed by projecting patches from frame $i$ into frame $j$ at two pyramid levels of resolution. Then, a 1D temporal convolution is applied to propagate information along each patch trajectory, followed by global message passing layers. Finally, two multilayer perceptrons are used to predict two parameters for each edge $(i, j)$ in the pose graph: a 2D flow vector $\delta_{ij}$, providing information on how to update the patch center's reprojection in two dimensions and a confidence weight map $\Sigma_{ij}$ to update the patch depths.
3. **Differentiable Bundle Adjustment**: In this step, two iterations of the Gauss-Newton method are applied to the patch graph using a fixed window size. Both camera poses as well as the inverse depth components of the patches are jointly optimized while keeping the pixel coordinates constant. To make the decomposition process more efficient, the Schur complement trick is used to backpropagate gradients through the Gauss-Newton iterations.

The original DPVO model was trained on the synthetic TartanAir dataset [28]. For this dataset perfect pose and dense depth data is available, but it only contains rendered images that lack realism. To increase variance of the data, trajectories with random frame gaps are sampled from the sequences, under the condition that subsequent frames have sufficient view overlap. This requires an upfront construction of a co-visibility graph for each sequence. The nodes of this graph represent the image frames and the edges represent the degree of co-visibility between the two frames, which is expressed by the optical flow magnitude between them. In other words, the co-visibility graph captures the amount of overlap between frames. Fractions of the adjacency matrices for such co-visibility graphs are visualized in Figure 5.

For each pair of frames, the optical flow is computed by reprojecting pixels from frame $i$ into frame $j$ using ground truth dense depth and poses provided by the TartanAir dataset. At training time the loss function is supervised in two ways:

1. **Pose supervision**: The predicted poses are directly compared to the ground truth poses. This is applied after the differential bundle adjustment step.
2. **Flow supervision**: The displacement of corresponding

patches between pairs of frames can be seen as a sparse optical flow prediction. This predicted optical flow is compared to the ground truth optical flow which is derived from the ground truth depth in the co-visibility graph construction.

## 3.2. Our Contribution

The baseline DPVO model is solely trained on synthetic data. Teed *et al*. [22] show that their model achieves competitive results on a real indoor drone dataset. However, the performance of DPVO on the KITTI odometry benchmark suite is inferior (see Table 1). This is likely due to the significant domain gap between TartanAir and real world automotive scenes. For closing this gap, it is necessary to train on target domain data. The drawback of the DPVO training approach is its dependence on dense and accurate ground truth depth for all camera frames which is needed for the optical flow computation. This information is hard to obtain for large scale realistic datasets. Sensors that measure distances directly like LiDARs are usually much sparser than the cameras and have limited field of views and ranges. Ground truth camera poses on the other hand can be gathered much easier, e.g. by fusing GNSS, IMU and wheel odometry measurements [1, 8, 10, 32].

In the following, we present two approaches, i.e. monocular depth estimation and direct optical flow estimation, that do not require any ground truth depth information. They enable training of DPVO at scale on real world datasets using weak flow supervision. A graphical overview of the entire system and both methods is given in Figure 2.

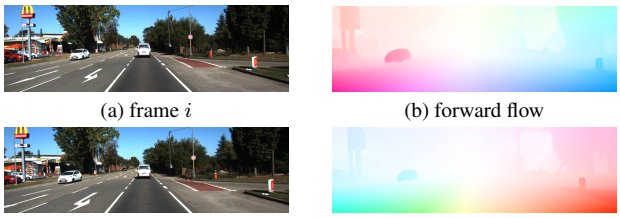
### 3.2.1 Monocular Depth Estimation

This approach corresponds to path *b)* in Figure 2. It replaces the missing ground truth depth with a prediction from a monocular depth estimation network. Recent research [2, 9] has shown that these networks can be trained in a self-supervised manner using nothing but sequences of calibrated images which renders this approach applicable to any kind of dataset.

To estimate the depth maps more efficiently, we downscale the input images to $1/4$ of the resolution. Furthermore, we filter and remove points that are projected in close proximity to the camera. Optical flow is then computed following the same procedure as in the baseline training approach, utilizing the ground truth poses.

### 3.2.2 Direct Optical Flow Estimation

Training DPVO requires dense depth only during the precomputation of the pairwise optical flow maps. The second approach (illustrated as path *b)* in Figure 2) directly computes the optical flow between image pairs without the need to estimate depth first. We propose to use a pre-trained neural network that takes two consecutive frames as input and outputs a dense optical flow field. These networks are usually trained supervised but generalize very well across domains. To improve accuracy, we implemented forward and backward consistency checks to eliminate occluded pixels (similar to [12]).



(a) frame $i$ (b) forward flow

(c) frame $i+1$ (d) backward flow

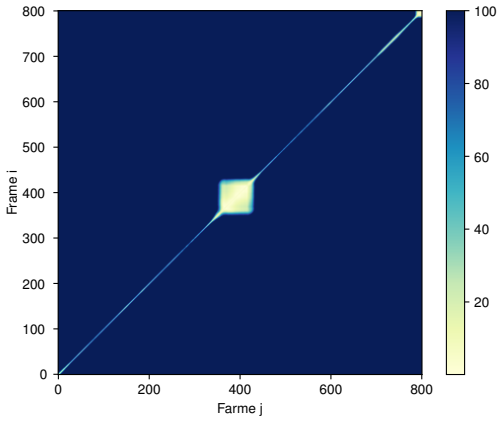
Figure 4. Forward and backward flows are derived from two consecutive images.

### 3.2.3 Efficient Co-Visibility Graph Construction

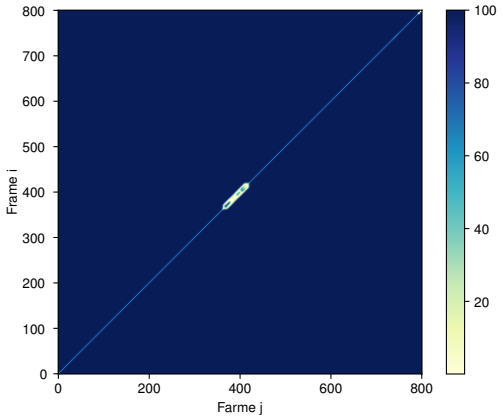
Constructing the co-visibility graph requires computing the optical flow between all image pairs in a sequence and thus has complexity $O(n^2)$ where $n$ is the number of frames. This becomes computationally intensive for extended sequences or datasets with a high frame rate. For instance, even if computing optical flow for a single frame pair takes only 10 milliseconds, processing a 10-minute recording at 25 frames per second (FPS) would require over 26 days (using a single thread). However, after constructing the co-visibility graph, the majority of optical flow maps are discarded (shown as the dark blue area in Figure 5) due to insufficient overlap. Especially in the automotive domain, only frames with a minimal time difference typically have overlapping field of views. Consequently, loop closures characterized by low optical flow magnitude are exceptionally rare. We propose a *neighboring* approach that computes the flow only for pairs of frames that are within a certain graph distance $n$ from each other $|i-j| < n$. Edges between all other frame pairs in the co-visibility graph are directly discarded. For the aforementioned example, this would reduce the runtime of the to less than 3 minutes. When comparing the co-visiblity graphs of the *full* approach and the *neighboring* approach on the same sequence it becomes apparent that their difference is negligible (see Figure 5).

# 4. Experiments

We evaluate our method on the KITTI dataset [5, 6], a widely-used automotive benchmarking dataset in computer vision. The KITTI dataset is a large-scale outdoor driving dataset that includes various splits for different tasks.




(a) Full approach (baseline) with $O(n^2)$ complexity




(b) Neighbouring approach (ours) with $O(n)$ complexity

Figure 5. Visualizations of the adjacency matrices of co-visibility graphs created using the *full* and the *neighbouring* approach on KITTI sequence 07. Note that only small fractions of the full matrices are shown. The large bright square in sub-figure (a) originates from a period where the car moving slow or not at all, which adds only little value to the training anyway.

Specifically, we focus on the odometry dataset, which consist of 11 driving sequences with publicly available ground truth camera poses. These sequences include long sequences with and without loop closures. Following the approach of [36], we train our networks on sequences 00 to 08 and evaluate on sequence 09 and 10.

### 4.1. Training Details

We train our networks from scratch using the PyTorch framework. The training procedure involved 80K iterations using a single NVIDIA RTX A6000 GPU with a batch size of one. We applied the *AdamW* optimizer, with an initial learning rate of $8e-5$. The learning rate was gradually reduced during training. Standard augmentation techniques, including resizing and color jittering, to the KITTI images with size $320 \times 1024$. During the training phase each data sequence utilized the first six frames to initialize the system, followed by the incremental addition of seven frames. The update operator was unrolled 18 times throughout the training process.

### 4.2. Pre-Trained Models

**Optical Flow**: Numerous deep learning methods have been proposed for optical flow estimation. Our approach is largely independent of the used optical flow estimator. For efficient training, we selected *GMFlow* [29] due to its combination of speed and accuracy. Specifically, we used the *mixdata* model which was trained on several public datasets covering differing domains and is recommended for in-the-wild use cases. The supervision of this optical flow network introduces a weak supervision in our approach.

**SIDE**: For single image depth estimation, we utilize *FeatDepth* [18] trained on the Eigen split of KITTI raw dataset [3] with an image resolution $1024 \times 320$ and outputting at half the resolution. Due to the self-supervision, no other data than sequences of raw input images was required for training.

### 4.3. Evaluation Metric

We use standard evaluation criteria to analyze monocular camera pose estimation methods. These include the Absolute Trajectory Error (ATE) for assessing the root-mean-square error between predicted and ground truth poses, and the Relative Pose Error (RPE) for evaluating frame-to-frame relative pose accuracy. Since monocular methods lack a scaling factor to match the real-world scale, we perform scaling and alignment using 7 Degrees of Freedom (7DoF) optimization [24] during the evaluation. This ensures that the predicted camera poses are accurately evaluated against the ground truth poses with respect to the real-world scale.

### 4.4. KITTI Odometry

**Ablation study**: We conduct an ablation study on the KITTI Odometry dataset validation sequences 09 and 10 to evaluate our design decisions. Specifically, we analyze the following variations:

1. **ours (OF+N)**: Direct estimation of optical flow using GMFlow using the *neighboring* approach for constructing the co-visibility graph.
2. **ours (SD+N)**: Indirect computation of optical flow using SIDE using the *neighboring* approach for constructing the co-visibility graph.
3. **ours (SD+F)**: Similar to *SD+N* but employing the *full* approach for constructing the co-visibility graph, i.e., computing optical flow between all pairs.

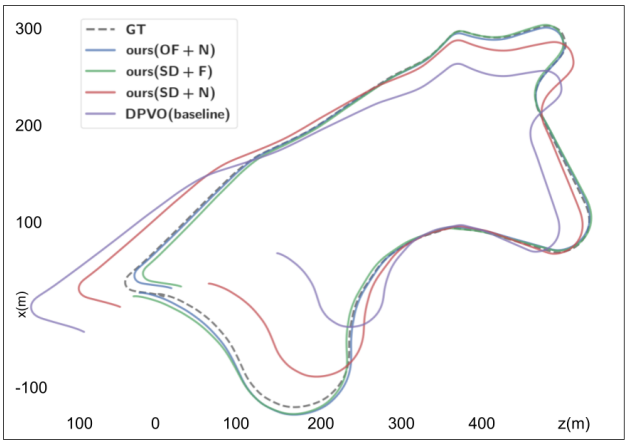
| Method | 09 | | | 10 | | |
|---|---|---|---|---|---|---|
| | ATE | RPE (m) | RPE (°) | ATE | RPE (m) | RPE (°) |
| **Deep Learning** | | | | | | |
| DPVO [22] | 76.52 | 0.28 | 1.33 | 12.10 | 0.08 | 1.10 |
| SfM-Learner [36] | 26.93 | 0.103 | 0.159 | 24.09 | 0.118 | 0.171 |
| SC-SfMLearner [2] | 15.02 | 0.095 | 0.102 | 20.19 | 0.105 | 0.107 |
| Depth-VO-Feat [34] | 52.12 | 0.164 | 0.233 | 24.70 | 0.159 | 0.246 |
| MonoDepth2 [9] | 55.47 | - | - | 20.46 | - | - |
| DeepV2D [20] | 79.06 | - | - | 48.49 | - | - |
| DeepMatchVO [17] | 27.08 | - | - | 24.44 | - | - |
| CC [15] | 29.00 | - | - | 13.77 | - | - |
| GeoNet [33] | 158.45 | - | - | 43.04 | - | - |
| **Geometric** | | | | | | |
| DSO [4] | 52.23 | - | - | 11.09 | - | - |
| ORB-SLAM2 (w/o LC) [13] | 38.77 | 0.128 | 0.061 | **5.42** | 0.045 | 0.065 |
| VISO2 [7] | 52.62 | 0.284 | 0.125 | 57.25 | 0.442 | 0.154 |
| **Proposed** | | | | | | |
| Our(OF+N) | **7.39** | 0.13 | 0.32 | <u>7.18</u> | 0.07 | 0.10 |
| Our(SD+F) | <u>10.84</u> | 0.11 | 0.33 | 11.47 | 0.09 | 0.10 |
| Our(SD+N) | 33.92 | 0.18 | 0.34 | 9.92 | 0.08 | 0.11 |

Table 1. Quantitative result on KITTI Odometry Seq. 09-10. The best result is printed **bold** and second best is <u>underlined</u>.

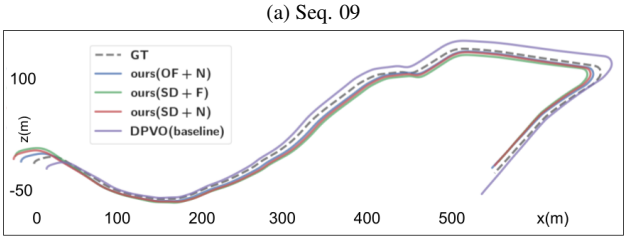
We do not evaluate the variant where optical flow is estimated directly and the *full* co-visibility due to its high runtime, rendering it impractical for our purposes.
The KITTI dataset only contains sparse ground truth depth data [23] which is common for real-world datasets. As a result, DPVO cannot be trained directly on the KITTI dataset. Despite the domain gap, we included the baseline DPVO model, trained on the TartanAir dataset, into the evaluation.

Quantitative results are summarized in Table 1 and a qualitative comparison of the trajectories is depicted in Figure 6. All variants of our weakly-supervised method outperform the DPVO baseline model by a significant margin. The *OF+N* approach achieves a lower ATEs than the *SD+F* and *SD+N* on both validation sequences. Using the *SD+F* improves performance on sequence 09 but not on 10. It shows that the *full* co-visibility approach adds only little benefit while being much more expensive.

**Comparison with the state-of-the-art**: For evaluation, we selected a number of state-of-the-art techniques for comparison. We compare our results to the geometric monocular version of ORB-SLAM2 without loop closure [13], DSO [4] and VISO2 [7], supervised learning method [20], and the self-supervised methods [2, 9, 15, 17, 33, 34, 36]. Because ORB-SLAM2 experiences tracking failure or unsuccessful initialization on occasion, we executed ORB-SLAM2 three times and present the result with the minimum trajectory error. As can be seen from the table 1 and Figure 7, we demonstrate that our proposed methods outperform pure deep learning methods that rely on *PoseCNN* for camera motion estimation by a large margin in ATE metrics. Furthermore, we show that our approach also outperforms well-known geometric methods in sequence 09.




(a) Seq. 09




(b) Seq. 10

Figure 6. Ablation study: comparing variants of our method with the DPVO baseline model: trajectories for KITTI sequences 09 (top) and 10 (bottom).

In sequence 10, while the ORB-SLAM method performs best, both of our proposed methods using the neighboring approach, show comparable results. In addition, our approaches did not experience tracking or initialization failures and significantly reduced translation drift over long sequences. As expected, our approach outperforms the baseline DPVO, as it allowed us to bridge the domain gap from the original dataset.

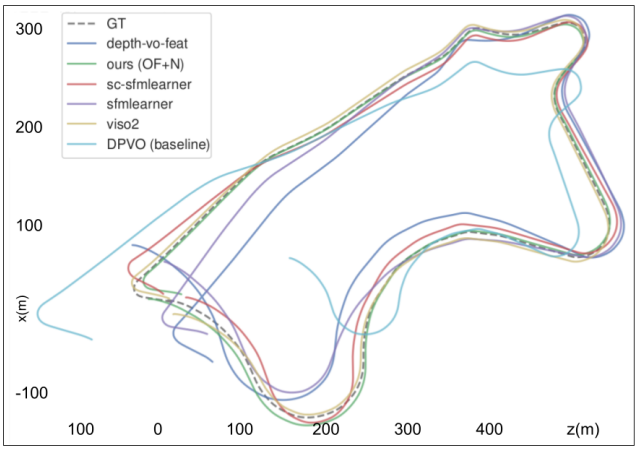



Figure 7. Trajectories Comparison for Sequence 09. We compare our best approach against some state-of-the-art approaches

### 4.5. Inference Time

In addition to the experiments of Teed *et al*. we tested the DPVO runtime performance on three different devices: On a high-end NVIDIA GeForce RTX 3080 Laptop GPU we achieved a frame rate of approximately 28 fps. On low-power systems we get 9 fps on the NVIDIA Jetson AGX Orin 64GB and 4 fps on the NVIDIA Jetson Xavier NX 8GB. It shows that the system is real-time capable, making it a practical and efficient solution for a wide range of robotics applications that require on-board accurate visual odometry for mapless navigation such as autonomous vehicles and drones.

## 5. Conclusions

We proposed a weakly supervised approach for deep visual odometry that overcomes the limitation of requiring high-quality dense depth ground truth data. By leveraging optical flow generated from raw camera images and poses as ground truth, our approach achieves state-of-the-art performance on the KITTI dataset, outperforming traditional methods and reducing catastrophic failures. Our method has potential applications in robotics, especially in automated driving for mapless navigation, where accurate and dense depth ground truth data is challenging to acquire. The proposed approach reduces the dependence on high-quality ground truth data, making it a practical and efficient solution for visual odometry in real-world scenarios.

## References


[1] P Aggarwal, Z Syed, and N El-Sheimy. Hybrid extended particle filter (hepf) for integrated civilian navigation system. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 984–992. IEEE, 2008. 4

[2] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019. 2, 3, 4, 6

[3] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 5

[4] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2, 6

[5] 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. 4

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 4

[7] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 963–968. Ieee, 2011. 2, 6

[8] Audrey Giremus, J-Y Tourneret, and Petar M Djuric. An improved regularized particle filter for gps/ins integration. In *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, 2005.*, pages 1013–1017. IEEE, 2005. 4

[9] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 2, 4, 6

[10] Mohinder S Grewal, Lawrence R Weill, and Angus P Andrews. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007. 4

[11] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 2

[12] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[13] 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. 2, 6

[14] David Nistér, Oleg Naroditsky, and James Bergen. Visual

odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004. 1

[15] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 2, 6

[16] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 1, 2

[17] Tianwei Shen, Zixin Luo, Lei Zhou, Hanyu Deng, Runze Zhang, Tian Fang, and Long Quan. Beyond photometric loss for self-supervised ego-motion estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6359–6365. IEEE, 2019. 6

[18] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, pages 572–588. Springer, 2020. 5

[19] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 2

[20] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 2, 6

[21] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3

[22] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *arXiv preprint arXiv:2208.04726*, 2022. 2, 4, 6

[23] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017. 6

[24] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991. 5

[25] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5555–5564, 2019. 2

[26] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2043–2050. IEEE, 2017. 2

[27] 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

[28] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 3

[29] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 5

[30] Fei Xue, Qiuyuan Wang, Xin Wang, Wei Dong, Junqiu Wang, and Hongbin Zha. Guided feature selection for deep visual odometry. In *Asian Conference on Computer Vision*, pages 293–308. Springer, 2018. 2

[31] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8575–8583, 2019. 2

[32] Yudan Yi and Dorota A Grejner-Brzezinska. Tightly-coupled gps/ins integration using unscented kalman filter and particle filter. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pages 2182–2191, 2006. 4

[33] 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. 6

[34] 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, 6

[35] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018. 2

[36] 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, 5, 6