ParticleSfM: Exploiting Dense Point Trajectories for Localizing Moving Cameras in the Wild

Wang Zhao[1,3], Shaohui Liu[2,3], Hengkai Guo[3],
Wenping Wang[4], and Yong-Jin Liu[1*]

[1]Tsinghua University [2]ETH Zurich [3]ByteDance Inc. [4]Texas A&M University

Abstract. Estimating the pose of a moving camera from monocular video is a challenging problem, especially due to the presence of moving objects in dynamic environments, where the performance of existing camera pose estimation methods are susceptible to pixels that are not geometrically consistent. To tackle this challenge, we present a robust *dense indirect* structure-from-motion method for videos that is based on dense correspondence initialized from pairwise optical flow. Our key idea is to optimize long-range video correspondence as dense point trajectories and use it to learn robust estimation of motion segmentation. A novel neural network architecture is proposed for processing irregular point trajectory data. Camera poses are then estimated and optimized with global bundle adjustment over the portion of long-range point trajectories that are classified as static. Experiments on MPI Sintel dataset show that our system produces significantly more accurate camera trajectories compared to existing state-of-the-art methods. In addition, our method is able to retain reasonable accuracy of camera poses on fully static scenes, which consistently outperforms strong state-of-the-art dense correspondence based methods with end-to-end deep learning, demonstrating the potential of dense indirect methods based on optical flow and point trajectories. As the point trajectory representation is general, we further present results and comparisons on in-the-wild monocular videos with complex motion of dynamic objects. Code is available at https://github.com/bytedance/particle-sfm.

Keywords: Structure-from-Motion, Motion Segmentation, Video Correspondence, Visual Reconstruction

1 Introduction

Localizing moving cameras from monocular videos is a fundamental task in a variety of applications such as augmented reality and robotics. Many videos exhibit complex foreground motion from humans, vehicles and general moving objects, posing severe challenges for robust camera pose estimation. Traditional indirect SfM methods [21, 35, 48] are built on top of feature point detectors and descriptors. These methods rely on high-quality local features and perform

[*] Corresponding author.

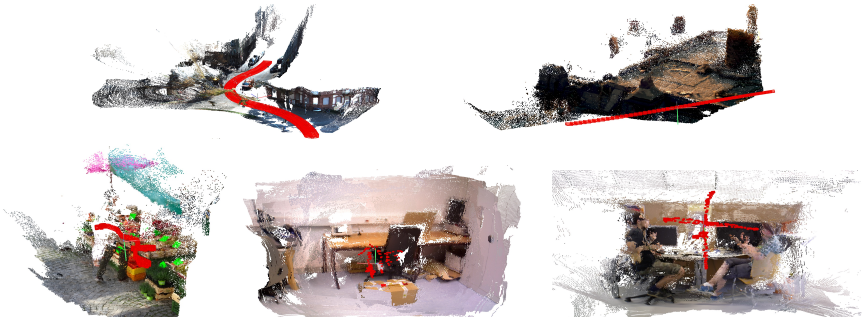


Fig. 1. We present a dense indirect method that is able to recover reliable camera trajectories from in-the-wild videos with complex object motion in dynamic scenes. (See Fig. 6 for qualitative comparisons with COLMAP [63])

non-linear optimization over the geometric reprojection error. Conversely, direct methods [19, 20, 49] track cameras by optimizing photometric error of the full image assuming consistent appearance across views. While both types of methods have produced compelling results, neither is robust against large object motion in dynamic environments, which is however ubiquitous in daily videos.

To mitigate the influence of moving objects, several existing monocular SLAM and SfM methods [5, 85, 89] attempt to focus on specific semantic classes of objects that are likely to move around, e.g. humans and cars. However, there are many general objects that can possibly move in the scene in practice (e.g. a chair carried by a human). And moreover, those "special" objects such as humans and cars are not necessarily moving in the videos, making these semantics-based methods limited. Recent methods employ end-to-end deep learning to implicitly deal with those complex motion patterns, placing focus on static parts with the aid of training data. However, the end-to-end learning on camera poses brings up limitation on the system generalization on in-the-wild daily videos.

We present a new *dense indirect* structure-from-motion system for videos that explicitly tackles the issues brought by general moving objects. Our method is based on dense correspondence initialized from pairwise optical flow. Inspired by the success of Particle video [61], our method exploits long-range video correspondence as dense point trajectories, which serves as an intermediate representation and provides abundant information for estimating motion segmentation and optimizing cross-view geometric consistency at global bundle adjustment.

Specifically, our method first connects and optimizes dense point trajectories using pairwise dense correspondence from optical flow. Then, we propose a specially designed network architecture to learn robust estimation of motion labels from point trajectories with variable lengths. Finally, we apply global bundle adjustment to estimate and optimize camera poses and maps over portions of each point trajectory that are classified as static. Since the point trajectories are high-level abstraction of the input monocular videos, training motion estimation solely on synthetic datasets such as FlyingThings3D [45] exhibits great generalization ability and enables our system to produce robust camera trajectories on general videos that contain complex and dense motion patterns (as shown in Fig. 1).

Experiments on MPI Sintel dataset [10] validate that our method significantly improves over state-of-the-art SfM pipelines on localizing moving cameras in dynamic scenes. In addition, on full static ScanNet dataset [13], our method is able to retain reasonable accuracy on the predicted camera trajectories, consistently outperforming strong dense correspondence based methods with end-to-end deep learning, such as DROID-SLAM [72]. We further present results and comparisons on in-the-wild monocular videos to demonstrate the improved robustness on dealing with complex object motion in dynamic environments.

2 Related Work

Dense Correspondence in Videos: Cross-image correspondences are conventionally built on local feature point detection and description [42, 60], and recent methods employ deep neural networks to improve local features [15, 18] and matching [62]. While the de-facto methods for localization and mapping operate on sparse feature points, dense pixel correspondences [23, 59, 67] have shown great potential especially on videos, thanks to the rapid developments of optical flow predictors. Early methods such as SIFT-Flow [40] achieve dense correspondences across different scenes, while recent advances [17, 29, 66] with deep learning regress the optical flow through differentiable warping and feature cost volumes. One notable recent work is RAFT [71] that introduces iterative recurrent refinement with strong cross-dataset generalization. While these optical flow methods have achieved remarkable performance on per-pixel accuracy, they intrinsically limit the correspondences to image pairs rather than the whole video data. Conversely, long-range video correspondence is early studied in Particle video [61] which first employs point trajectories to represent motion patterns in videos. However, the method is very computational expensive and practically not suitable for long videos. Sundaram et al. [69] further propose a fast parallel implementation of variational large displacement optical flow [9], and directly accumulate optical flow to get dense point trajectories. Inspired by these pioneer works, we present a method that sequentially tracks the optical flow and optimizes the pixel locations by exploiting path consistency to acquire reliable dense point trajectories.

Motion Segmentation: Motion segmentation aims to predict what is move for each image in a video sequence. Classical methods [65, 75] estimate the motion mask based on optical flow analysis, with follow-up works [8, 12] formulating joint optimization over optical flow and motion segmentation. Recent methods with deep neural networks [30, 73, 95] extract both appearance and optical flow features with a specially designed two-branch networks, or exploit semantic information [14] with optical flow based motion grouping. Self-supervised methods [81, 86] are also proposed to overcome the requirement of labeling data. To better parse the object motion, relative camera motion is estimated and incorporated in [6, 7, 37, 82]. However, these optical-flow based methods mostly suffer from temporal inconsistency and in-the-wild generalization to dynamic videos with different shapes and textures. On the other hand, point trajectories contain rich information for temporal object motion. There are a number of existing works [22, 31, 38, 50, 51, 64] that cluster the tracked point trajectories into motion segments with hand-crafted features and measurements. However, all theses clustering-based methods rely on heavy optimization and hand-crafted design, making them less scalable and general on in-the-wild video sequences. In this work, we combine the best of both worlds and introduce a specially designed network architecture for predicting trajectory attributes, specifically motion labels from irregular data of dense point trajectories.

SfM and SLAM: Traditional methods on structure-from-motion can be generally classified into indirect and direct methods. Indirect approaches [21, 35, 47, 48, 58, 63, 79, 88] rely on matched salient keypoints to determine geometric relationship for multi-view images. Conversely, direct methods [2, 19, 20, 49, 84] approximate gradients over dense photometric registration on the full image. While both trends of methods achieve great success in practice, they both suffer in dynamic scenes due to the large number of pixel outliers. Most related to us, recent literature attempts to exploits dense correspondences from optical flow. In particular, [57,94] employ two-view geometric consistency check to detect moving objects, and visual odometry systems are built by triangulating optical flow correspondences in [90,94]. VOLDOR [46] employs a probabilistic graphical model over optical flow to recover camera poses as hidden states. In TartanVO [76], optical flow is fed into an end-to-end network to directly predict camera poses. R-CVD [36] jointly optimizes depth and pose with flexible deformations and geometry-aware filtering. DROID-SLAM [72] implicitly learns to exclude dynamic objects by training on sythetic data with dynamic objects and employing deep bundle adjustment over keyframes. While our method also benefits from dense correspondences from pairwise optical flow, we do not employ end-to-end deep learning to encourage better generalization. Instead, we explicitly model dense point trajectories and perform bundle adjustment over these video correspondences.

Localizing Cameras in Dynamic Scenes: Localizing cameras in dynamic scenes is challenging due to violation of rigid scene assumption in multi-view geometry. Classical SfM and SLAM methods reject moving pixels as outliers through robust cost function [35] and RANSAC [48, 63], yet consistently fail under highly dynamic scenes with complex motion patterns. Beyond monocular SLAM, effective segmentation and tracking of dynamic objects can be achieved [1, 4, 27, 32, 39, 68, 78] with auxiliary depth data from stereo, RGB-D and LiDAR, which, however, is not generally available for in-the-wild captured videos. Thanks to the rapid development of deep learning on visual recognition, many works [3, 5, 85, 89, 93] tackle this problem by exploring the combination with object detection, semantic and instance segmentation. However, These methods are often restricted to pre-defined semantic classes. On the contrary, our method exploits the potential of long-range point trajectories in videos for robust motion label estimation.

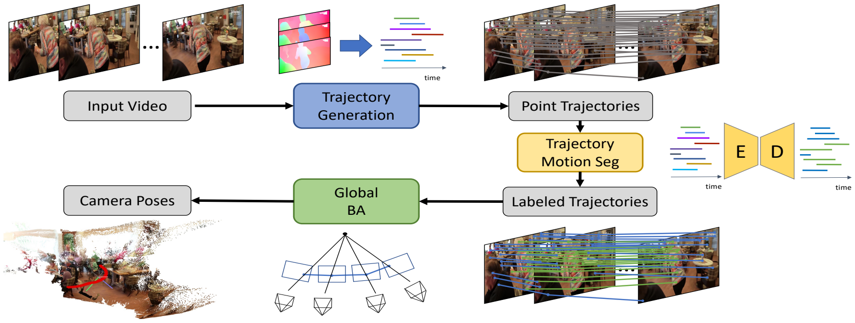


Fig. 2. Overview of our proposed system for localizing moving cameras. Given an input video, we first accumulate and optimize over pairwise optical flow to acquire high-quality dense point trajectories. Then, a specially designed network architecture is employed to process irregular point trajectory data to predict per-trajectory motion labels. Finally, the optimized dense point trajectories along with the motion labels are exploited for global bundle adjustment (BA) to optimize the final camera poses

3 Methods

The core idea of the proposed system is to exploit long-range video correspondences as dense point trajectories throughout the pipeline. Figure 2 shows an overview of the system. We first accumulate and optimize point trajectories sequentially in a sliding-window manner from optical flow. Then, those point trajectories are fed into the specially designed trajectory processing network to predict per-trajectory motion labels. Finally, we estimate initial camera poses, triangulate global maps from the portions of each trajectory that are classified as static, and perform bundle adjustment over those point tracks to optimize both camera poses and 3d points. As our method employs dense correspondence, the maps built from our system are denser and more complete than top sparse indirect methods such as COLMAP [63] with comparable running time needed.

3.1 Acquiring Dense Point Trajectories

We aim to acquire reliable point trajectories for motion estimation and global bundle adjustment. Following the practice of early literature [61, 69] that focus on long-range video correspondence, we start from pairwise optical flow and sequentially accumulate them into point trajectories. We use RAFT [71] as the base optical flow predictor.

For accumulation of dense point trajectories, given the current pixel location p_0 on image 0 (sized $H \times W$) and the optical flow $F_{0\rightarrow1} \in \mathbb{R}^{H\times W\times 2}$ from image 0 to image 1, the trajectory can be extended with: $p_1 = p_0 + F_{0\rightarrow1}(p_0)$. We continue tracking point trajectories until the point suffers from occlusion, which is determined by the forward-backward optical flow consistency check following common practice [69, 87]. This forward-backward consistency check not only deals with occlusion, but also filters out some erroneous optical flow, making the trajectories more reliable. To maintain dense point trajectories, for each accumulation step we generate new trajectories on the area that is not occupied by any trajectory on the current image. All trajectories are initialized at grid points. Following [69], a sub-sampling factor λ is employed to control the density of the trajectory by sub-sampling unoccupied pixels uniformly on 2D space, which helps balance the computational cost and trajectory density if needed.

The resulting point trajectories are dense and roughly correct as video correspondences. However, small errors from the optical flow accumulates across time, incurring the tracked pixel locations to gradually drift away from the true ones. Similar drifting errors often occur for tracking-based methods [33, 91] for accumulating sensor measurements. To fight against drifting, Particle video [61] attempts to perform a heavy optimization on pixel locations directly using appearance error on raw intensity images. Different from theirs, our method exploits path consistency that benefits from optical flow from non-adjacent image pairs.

Specifically, given consecutive frames I_0, I_1, I_2 and pairwise optical flows $F_{i\to j}$ from I_i to I_j, we first initialize the point trajectory p_1, p_2 on I_1 and I_2 sequentially with direct accumulation:

$$p_1' = p_0 + F_{0\to 1}(p_0), \quad p_2' = p_1' + F_{1\to 2}(p_1'). \tag{1}$$

Then, we compute stride-2 optical flow $F_{0\to 2}$ and optimizes p_1 and p_2 with respect to the following objectives:

$$\begin{aligned} L = (p_1 - p_1')^2 + &(p_2 - (p_0 + F_{0\to 2}(p_0)))^2 \\ &+ (p_2 - (p_1 + F_{1\to 2}(p_1)))^2 \end{aligned} \tag{2}$$

Gradients are numerically tractable with interpolation on the optical flow map $F_{1\to 2}$. Through the optimization we jointly adjust the pixel locations along the track to encourage consistency. The framework is easily extended for longer windows to exploit longer range of optical flow correspondences but we empirically find that adding a single stride-2 constraints already consistently improves the track quality by mitigating the drifting problem.

It is also worth noting that the success of path consistency formulation relies on the key assumption that the direct pairwise measurements are relatively more accurate than cumulative results when constructing point trajectories. This is generally true for accumulating sensor measurements for visual odometry, while in our case the assumption can be violated because if the pixel motion between two images is too significant the accuracy of the pairwise optical flow will degrade. Thus, it is relatively safe to keep a small window to avoid large degradation of the long-range optical flow. In practice, if $F_{0\to 2}$ does not pass forward-backward consistency check, we skip the optimization at the timestamp and keep the initial accumulated positions. The point trajectory is sequentially extended and optimized until occlusion is detected.

3.2 Trajectory-based Motion Segmentation

Moving objects, while being ubiquitous in daily videos, pose severe challenges for camera pose estimation as it violates geometric consistency across different

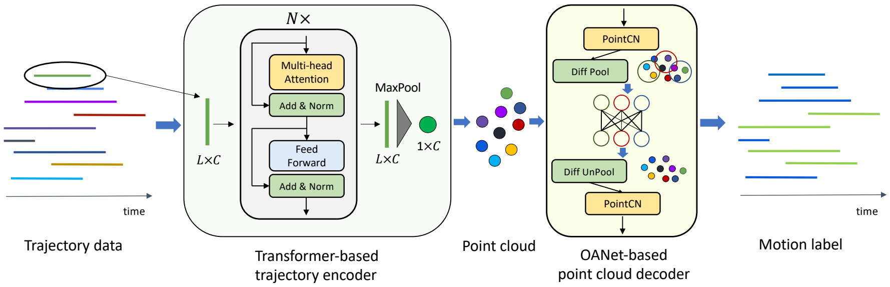


Fig. 3. Illustration of the trajectory-based motion segmentation network. To process irregular point trajectory data, we first employ a transformer-based encoder to extract features for each trajectory. Then, all trajectory features are considered as high-dimensional point cloud and fed into an OANet-based [92] decoder to predict per-trajectory motion labels. In practice, we split long-range point trajectories into segments and map the predicted motion labels for each segment back onto the pixels

timestamps, making it crucial to design strategies to filter out dynamic objects. One trend of commonly-used methods in dynamic SLAMs [5, 85, 89] is to utilize semantic segmentation model, e.g. Mask-RCNN [25] to get per-pixel semantic masks, and remove all potentially moving pixels according to its semantic labels, such as person, car, dog, etc. While these methods are effective under certain scenarios, they are intrinsically not general for segmenting moving objects, since it completely relies on pre-defined semantic classes and cannot distinguish moving or static objects within the same semantic category (e.g. moving and static cars). Conversely, another alternative is to use two-frame motion segmentation methods [73, 95], which are recently powered with various convolutional neural network (CNN) models. These methods provide true motion labels instead of semantic candidates. However, two-frame based CNN models suffer from severe temporal inconsistency, degraded estimation when input flows are noisy, and exhibit limited out-of-domain generalization on in-the-wild videos.

We propose to exploit the dense point trajectories we acquired for estimating motion labels. Trajectory-based methods are generally more robust and consistent compared to two-frame flow models for motion segmentation. Unfortunately, traditional cluster-based trajectory segmentation methods rely on heavy optimization and hand-crafted features, and are hard to scale with dense trajectories. To deal with such issues, we propose a novel neural network for processing trajectory data for prediction, which enables fast, robust and accurate dense point trajectory based motion segmentation. As shown in Fig. 3, our proposed network employs a encoder-decoder architecture. Specifically, the encoder directly consumes irregular trajectory data and embeds into high-dimensional feature space. Then, all the encoded trajectories are together fed into the decoder, which performs context-aware feature aggregations among trajectories to fuse both local and global information and finally regress the motion label.

As each trajectory has different start time, end time and length, the point trajectory data is highly irregular and hard to be directly processed with regular convolutional neural networks. Inspired by sequence modeling in natural language processing where language sequences are also irregular, we utilize the powerful transformer [74] model to extract features from trajectories. Built up with multi-head attention, transformer can effectively process sequential data, and is broadly used in language model, and recently extended to vision tasks [16]. Our input data for transformer encoder is N trajectories, and each trajectory includes a set of normalized pixel coordinates (u_i, v_i). We first cut and pad all the trajectories to the temporal window size L. After that, all trajectories have the shape of $(L, 2)$ with masks indicating where the pixel coordinates are zero-padded. To better exploit motion information, we augment the trajectory data $\{(u_i, v_i), i \in [0, L)\}$ with consecutive motion $(\mathit{\Delta} u_i, \mathit{\Delta} v_i) = (u_{i+1} - u_i, v_{i+1} - v_i)$. Furthermore, since the moving objects are much easier to be classified in 3d space, we integrate relative depth information from MiDaS [56] to disambiguate the motion segmentation from pure 2d pixel movements. The estimated relative depth is normalized to $(0, 1)$ and used to back-project the 2d pixels into 3d camera coordinates (x_i, y_i, z_i). We also include the 3d motion data $(\mathit{\Delta} x_i, \mathit{\Delta} y_i, \mathit{\Delta} z_i) = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)$ for the trajectory input. The final augmented trajectories have the shape of $(L, 10)$. This data is then embedded by two MLPs, resulting in intermediate features with shape (L, C), which is fed into the transformer module. The transformer consists of 4 blocks, each with multi-head attention and feed-forward layers to encode the temporal information of each trajectory. The output of the transformer module is encoded features also with shape (L, C). To get feature representation of the whole trajectory instead of each point, we further perform max-pooling over temporal dimension, resulting in $(1, C)$ feature vector for each trajectory.

To segment a trajectory as moving objects or static background, the model not only needs to extract motion pattern from each trajectory data, but also has to communicate and compare with other trajectories before making decisions, which is achieved by the decoder. All encoded trajectory features with shape (N, C) naturally forms a feature cloud in high-dimensional space. Thus, we build the decoder on top of carefully designed point cloud processing architectures.

Specifically, we choose OANet [92] as our backbone to benefit from its efficiency and well-designed local-global context mechanism. To capture the local feature context, the network first clusters input points by learning a soft assignment matrix (Diff Pool in Figure 3). Then, the clusters are spatially correlated to explore the global feature context. Next, the detailed context features of each point are recovered from embedded clusters through differentiable unpooling. And finally, several PointCN layers are applied, followed by sigmoid activation to get the binary prediction mask. We refer the reader to [92] for more details about differentiable pooling, unpooling and PointCN layer.

Our proposed trajectory motion segmentation network is fast, robust and general. It could process tens of thousands trajectories within a few seconds. By only trained on synthetic dataset *FlyingThings3D* [45], the network generalizes well across various scenarios including indoor, outdoor, synthetic movies and daily videos. Furthermore, the proposed network can also be easily extended to predict other trajectory attributes that benefit video understanding.

3.3 Global Bundle Adjustment over Point Trajectories

The optimized dense point trajectories along with the predicted motion labels can be jointly used for localizing moving cameras for monocular videos. Specifically, given a video sequence, we first accumulate and optimize optical flow to get long-range point trajectories. Then, the trajectory motion segmentation network predicts trajectory motion labels in a sliding-window manner with window size L. These motion labels are mapped back onto each pixel to get final per-point motion segmentation. In the map construction and global bundle adjustment we only consider the portions of each trajectory that have static labels.

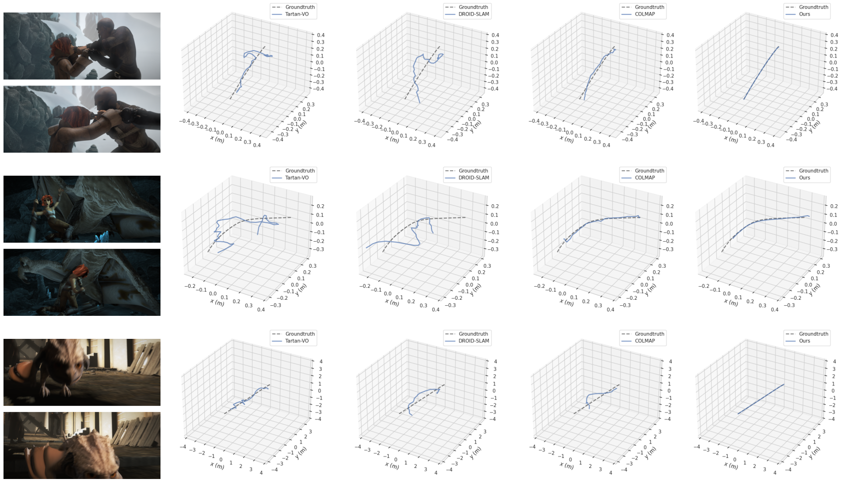
Since we have dense video correspondences, we can directly formulate non-linear geometric optimization over the point tracks. Inspired by [70, 79], we build a global SfM pipeline with dense point trajectories. Specifically, relative camera poses for neighboring views are first solved [24] with sampled correspondences from pairs of static pixels from the point trajectories with its motion labels. Then, rotation averaging [11] and translation averaging [52] are performed to get the initial camera pose estimations. Finally, global bundle adjustment is applied over the constructed point tracks at triangulation stage. Note that the point tracks are consistent with the original dense point trajectories since it comes from dense correspondences sampled from it. Thus, we achieve final pose refinement by making use of the static pixels along the trajectories without considering outlier pixels that are classified as parts of moving objects. As the trajectory processing network is well generalized, our method can recover reliable camera trajectories on in-the-wild daily videos with complex foreground motion.

4 Experiments

4.1 Implementation Details

We use the pre-trained RAFT [71] model on *FlyingThings3D* dataset [9]. The sub-sampling factor λ of point trajectory is set to 2 for all the experiments to balance the reconstruction density and computation time. For trajectory motion segmentation network, we use 4 heads for multi-head attention and 64 dimension for feed-forward layer. As for OANet, we set the cluster number to 100 and the layer number to 8. We train the proposed network on the training split of *FlyingThings3D* dataset [45]. We implement the network in PyTorch [53] and train it using Adam optimizer [34] with learning rate 1e-4 for 30 epochs. At inference, we use a window size L of 10 by default.

We test our system on Sintel [10], ScanNet [13], and in the wild video sequences from DAVIS [54]. For ScanNet dataset, we evaluate our method on the first 20 scenes in the test split. Since the whole video sequence is very long and not suitable for evaluating time-consuming offline methods like COLMAP [63], we only take the first 1500 frames and down-sample with stride 3 for each scene, resulting in a roughly 10 FPS video. Since all the methods use monocular video as input, we first scale and align all the output camera trajectories with respect to groundtruth, and then calculate commonly used pose metrics: RMSE of absolute trajectory error, translation and rotation part of relative pose error.



Sample frames Tartan-VO [76] DROID-SLAM [72] COLMAP [63] Ours

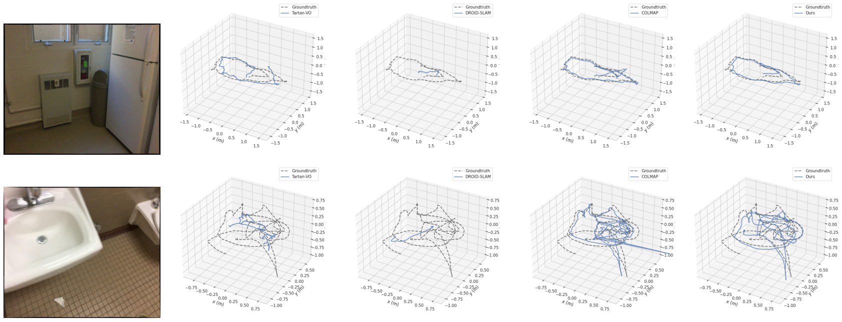
Fig. 4. Qualitative results of moving camera localization on MPI Sintel dataset [10]

Table 1. Quantitative evaluation on MPI Sintel dataset [10]. COLMAP [63] is additionally compared on its successful subset. Metrics are averaged across sequences

	Methods	ATE (m)	RPE trans (m)	RPE rot (deg)
COLMAP subset	COLMAP [63]	0.145	0.035	0.550
	MAT [95] + [63]	0.069	0.024	0.726
	Mask-RCNN [25] + [63]	0.109	0.039	0.605
	Ours	**0.019**	**0.005**	**0.124**
Full set	COLMAP [63]	X	X	X
	R-CVD [36]	0.360	0.154	3.443
	Tartan-VO [76]	0.290	0.092	1.303
	DROID-SLAM [72]	0.175	0.084	1.912
	Ours	**0.129**	**0.031**	**0.535**

4.2 Evaluation on MPI Sintel dataset

MPI Sintel dataset [10] contains 23 synthetic sequences of highly dynamic scenes. We remove sequences that are not valid to evaluate monocular camera pose (e.g. static cameras, perfectly straight line), resulting in a total of 14 sequences for comparison. We compare our system with both feature based indirect SfM method COLMAP [63] and state-of-the-art deep learning methods [36, 72, 76]. The results are summarized in Table 1. We also provide comparisons with representative SLAM methods ORB-SLAM [48] and DynaSLAM [5] in the supplementary materials. Since COLMAP fails in 5 of 14 sequences, we perform comparison on the subset of other 9 sequences. Furthermore, we set up baselines by extracting motion masks from state-of-the-art methods MAT [95] and Mask-RCNN [25] to augment COLMAP [63], where no feature points are extracted in the dynamic region. For Mask-RCNN [25], all the pixels that belong to potentially dynamic



Sample frame Tartan-VO [76] DROID-SLAM [72] COLMAP [63] Ours

Fig. 5. Qualitative results of moving camera localization on ScanNet dataset [13]

objects (person, vehicle, animals) are considered dynamic. As shown in Table 1, while explicit motion removal improves the perforamance, our method outperforms compared baselines by a largin margin thanks to the advantages of long-range point trajectories. For learning-based methods, Tartan-VO [76] and DROID-SLAM [72] are both trained on large-scale dataset TartanAir [77] and have demonstrated strong generalization ability across different datasets. However, they struggle on dynamic scenes and fail to predict reliable camera trajectories. Some qualitative examples are shown in Figure 4, where our system produces accurate camera poses over highly dynamic sequences.

4.3 Evaluation on ScanNet

Table 2. Quantitative evaluation on ScanNet [13]. COLMAP [63] is additionally compared on its successful subset. Metrics are averaged across sequences

	Methods	ATE (m)	RPE trans (m)	RPE rot (deg)
COLMAP subset	COLMAP [63]	**0.171**	0.064	2.900
	Ours	0.319	**0.017**	**0.632**
Full set	COLMAP [63]	X	X	X
	R-CVD [36]	0.468	0.065	7.626
	Tartan-VO [76]	0.353	0.045	2.620
	DROID-SLAM [72]	0.687	0.038	3.117
	Ours	**0.349**	**0.024**	**0.924**

To study the generalization of the proposed dense indirect SfM system, we further test our method on fully static indoor dataset ScanNet [13]. Some sequences in ScanNet contain inf values in groundtruth poses and we exclude them from evaluation, resulting in out of 20 test sequences. The results are shown in Table 2. Our method consistently improve dense baselines and provide reasonable camera trajectories with complex camera motions, as shown in Figure 5. As for

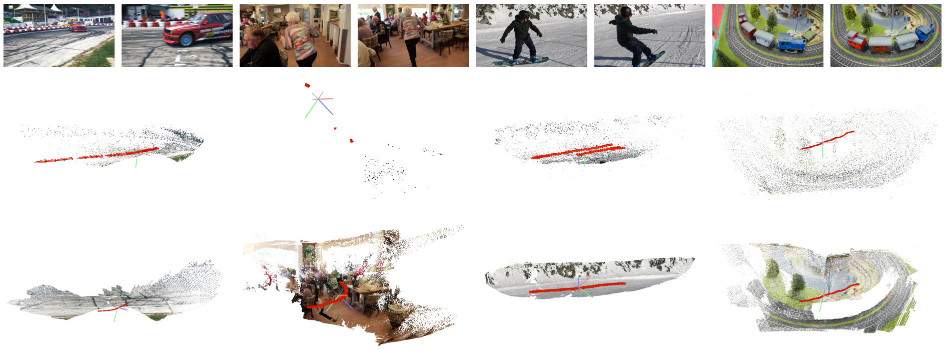


Fig. 6. Top row: COLMAP [63]. **Bottom row:** Ours. Qualitative comparison on in-the-wild videos from DAVIS [54]. Our dense indirect system produces robust camera trajectory and denser maps for videos with complex foreground motion

feature based method, COLMAP fails in 3 sequences out of 17, while our dense indirect system succeeds on all 17 sequences. We further present comparisons on the successful subset of COLMAP. Our method is slightly behind on ATE, but better on RPEs. This is possibly due to the degraded quality of indoor optical flow and the missing loop closure in our global bundle adjustment.

4.4 Qualitative evaluation on in-the-wild videos

As our system is general, we further present results and comparisons with COLMAP [63] on in-the-wild monocular videos with complex motion of dynamic objects in Figure 6. Videos are taken from DAVIS [54]. While COLMAP sometimes fails to predict reasonable camera poses (see the first and second sample of Figure 6), our method generalizes well in the wild, and is more robust to dynamic objects. Furthermore, our dense indirect solution is able to build significantly denser 3d reconstructions, which demonstrates the potential of indirect methods for recovering fine-grained 3D geometry.

4.5 Ablation Study

We ablate different components of our system on MPI Sintel dataset [10]. Table 3 shows the results. Without trajectory motion segmentation, all candidate correspondences, including those on moving objects, are used for map construction and global bundle adjustment. Results show that this leads to significantly degraded accuracy of recovered poses, indicating the important of removing dynamic pixels. Moreover, by optimizing point trajectories with path consistency, the mean endpoint error of trajectories is reduced by around 10%, which consequently improves camera localization (the last two rows). To further validate the necessity of dense point trajectories, we introduce two feature-based baselines that are built on SIFT [42], where one is also integrated with MAT [95] for removing moving pixels. As shown in Table 3, these two methods sometimes fail on sequences, and

Table 3. Results of ablation studies on MPI Sintel dataset [10]. We report metrics for all methods on both the full set (left) and the successful subset for SIFT + Global BA (right). "Optim" denotes trajectory optimization and "Seg" denotes trajectory-based motion segmentation and its use for global bundle adjustment

Methods	ATE (m)	RPE trans (m)	RPE rot (deg)
SIFT + Global BA	X / 0.060	X / 0.042	X / 0.635
SIFT + MAT [95] + Global BA	X / 0.054	X / 0.055	X / 0.621
Traj + Global BA	X / 0.071	X / 0.041	X / 0.969
Traj + Optim + Global BA	X / 0.072	X / 0.042	X / 0.929
Traj + Seg + Global BA	0.146 / 0.046	0.039 / 0.015	0.567 / 0.212
Traj + Optim + Seg + Global BA	**0.129** / **0.042**	**0.031** / **0.013**	**0.535** / **0.199**

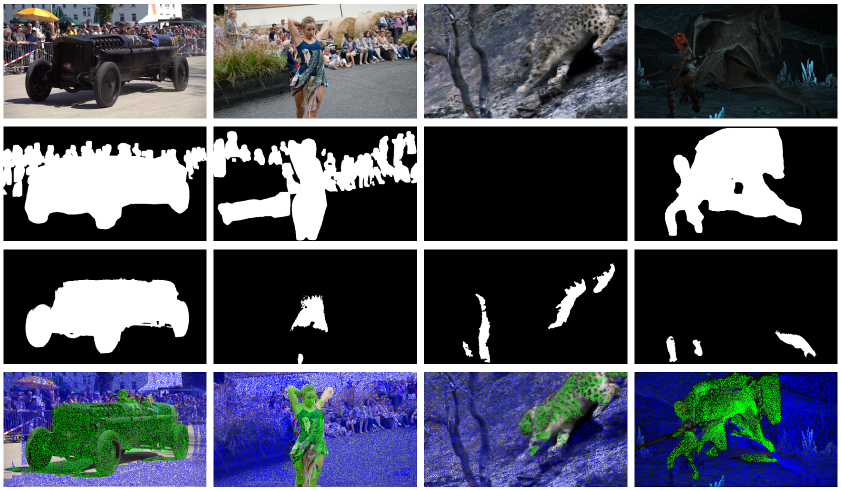


Fig. 7. Qualitative results on motion segmentation. **From top to bottom:** sample image, Mask-RCNN [25], MAT [95] and our trajectory-based motion segmentation

are behind on accuracy when tested on their successful subset compared to our system, demonstrating the effectiveness of dense point trajectories.

To study the effects of different components in the trajectory motion segmentation network, we perform ablation studies on MPI Sintel dataset [10] by evaluating segmentation w.r.t groundtruth motion masks. Since trajectory motion segmentation methods only offer predictions for trajectories, we map the motion labels into corresponding pixel locations and evaluate all methods only on these pixels. Table 4 shows the results. We compare our method with state-of-the-art supervised motion segmentation method [95] that is trained on DAVIS [54] and YouTube-VOS [80]. Our method, while only trained on synthetic FlyingThings3D dataset [45], achieves better motion segmentation even without depth information. Figure 7 shows some qualitative results, where our method predicts reliable motion masks for moving pixels. To further support the novel design of the network, we introduce a baseline that utilize two-branch CNN networks with ResNet-50 [26] to aggregate motion, appearance, and optionally also depth

Table 4. Evaluation and ablation studies for motion segmentation on MPI Sintel dataset [10]. We compare our method with several state-of-the-arts and ablate different components in our design on trajectory-based motion segmentation

Methods	mIoU (%)	Precision	Recall	F1-score
MAT [95]	47.5	**0.82**	0.54	0.56
COS [43]	55.0	0.67	**0.77**	0.65
MotionGrouping [81]	16.2	0.64	0.19	0.25
AMD [41]	31.5	0.42	0.62	0.45
Two-branch CNN w/o depth	29.2	0.54	0.49	0.39
Two-branch CNN	33.7	0.62	0.50	0.44
Ours with MLP encoder	54.8	0.67	0.73	0.66
Ours with PointNet decoder	46.3	0.65	0.67	0.58
Ours w/o depth	54.6	0.72	0.70	0.65
Ours	**60.6**	0.79	0.74	**0.72**

information. However, results indicate that this two-branch network exhibits low generalization ability when trained on FlyingThings3D dataset. Finally, we also ablate the architecture design by substituting the encoder with vanilla MLPs and the decoder with PointNet [55]. Results clearly show the advantages of using attention-based encoder and local-global context-aware decoder in our system.

Runtime analysis. For a 50-frame video, our system runs within 690s on average, including all the I/O time of each module. Specifically, optical flow extraction takes around 57s, the acquisition of dense point trajectories takes 200s, the trajectory motion segmentation takes around 65s, and the final global BA takes 368s. For comparison, COLMAP [63] takes 410s on average. Note that it is computationally unaffordable to run COLMAP with dense correspondences from the acquired dense point trajectories, while our method achieves dense correspondence with reasonable overhead.

5 Conclusions

In this work, we present a general *dense indirect* system for localizing moving cameras from in-the-wild videos. The key to the success of our method is to optimize and exploit long-range dense video correspondences as point trajectories, which are used for motion analysis and global bundle adjustment. A specially designed trajectory-based motion segmentation network is proposed to process the irregular point trajectory data. Experimental results show that our method can recover reliable camera trajectories from in-the-wild videos with complex motion patterns. Possible future directions include advanced optimization for long-range point trajectories and integration of loop closure for pose optimization.

Acknowledgements We thank anonymous reviewers for their valuable feedback. This work was supported by the Natural Science Foundation of China (61725204) and Tsinghua University Initiative Scientific Research Program.

A More Implementation Details

Point Trajectory. We use the pretrained *raft-things* model for RAFT [71] in all our experiments. It is trained on FlyingChairs and FlyingThings3D [45]. For optical flow forward-backward consistency check, we use the threshold of 1px for MPI Sintel dataset [10], and 3px for ScanNet dataset [13].

Motion Segmentation. For training trajectory motion segmentation network, we first prepare the trajectory data of FlyingThings3D dataset [45]. The dataset consists of over 2000 training scenes, and each scene contains 10 video frames, together with groundtruth optical flow, camera parameters and depth maps. We run our dense point trajectory generation algorithm to track trajectories from groundtruth optical flows, and then calculate trajectory groundtruth motion labels by comparing optical flow with rigid flow from depths and camera poses. We infer the relative depth information by pretrained MiDaS [56] model. For all experiments, we use the pretrained *midas-v21* model. During training, we directly take all the point trajectories from 10 video frames and output the per-trajectory motion label. Weighted binary cross-entropy loss is then applied.

Global Bundle Adjustment (BA). The implementation of our pipeline is mainly based on the Theia SfM system [70]. With the dense correspondences sampled from the point trajectories, we first compute two-view geometry [24] between valid image pairs and decompose the relative poses. In particular, the view pairs with very few or extremely noisy correspondences are detected with geometric verification [63] and ignored in the subsequent stages. Then, L1-IRLS rotation averaging [11] is applied to estimate the global orientations from the relative rotations among those valid pairs. After filtering outlier pairs with large errors on the relative rotations, we solve for the relative translations with the global rotations and apply LUD translation averaging [52] to get global translations. With these initial global poses, we incrementally triangulate 2D observations as in [63] and perform bundle adjustment on all the poses and 3D points to get the final output camera poses. Note that since we triangulate over the correspondences directly sampled from the dense point trajectories, each constraint in the bundle adjustment exactly corresponds to a part of the original point trajectory with geometric filtering, enabling effective global bundle adjustment over the input trajectory observations.

B Comparisons with SLAM methods

We provide the quantitative comparison results on Sintel dataset with representative monocular SLAM methods ORB-SLAM [48] and DynaSLAM [5]. ORB-SLAM [48] is the most popular feature-based monocular SLAM method which utilizes robust front-end tracking and local bundle-adjustment to achieve accurate camera localization. Built on top of ORB-SLAM, DynaSLAM [5] further introduces semantics to remove potentially dynamic objects to improve the robustness. Since ORB-SLAM and DynaSLAM only provide key-frame poses, we compared the localization accuracy solely on their key-frames. For Sintel dataset, both ORB-SLAM and DynaSLAM **consistently fail in 8 of 14 sequences**, and we summarize the results from other 6 successful sequences in Table 5. Our method surpasses both of them by a large margin even on their successful subset. Furthermore, ORB-SLAM and DynaSLAM **fail on all 17 ScanNet sequences**, probably due to large motion blur and poorly textured regions, while our method consistently provides reasonable camera poses.

Table 5. Evaluation on the successful subset (6 out of 14 sequences) of ORB-SLAM / DynaSLAM on MPI Sintel dataset.

Methods	ATE (m)	RPE trans (m)	RPE rot (deg)
ORB-SLAM	0.042	0.022	0.402
Ours	**0.009**	**0.006**	**0.101**
DynaSLAM	0.020	0.019	0.359
Ours	**0.007**	**0.005**	**0.090**

C Per-scene Results on MPI Sintel and ScanNet

We show the per-scene comparison results of MPI Sintel [10] and ScanNet [13] dataset in Table 6 and Table 7. For MPI Sintel, our method achieves the best performance on most sequences, demonstrating the advantage of the proposed system in dynamic scenarios. For fully static indoor dataset ScanNet, our method retain comparable performance with COLMAP [63], slightly behind on ATE and better on RPEs. COLMAP globally matches the feature points between image pairs, thus naturally has the ability of loop closure, while our method lacks as point trajectories are accumulated sequentially. Although our method could achieve good relative pose estimations, the trajectory error will be accumulated without loop closure. This is possibly the main reason why our method is worse than COLMAP in ATE but better in RPEs. In the future, we aim to implement the loop closure inside our system by matching trajectories across frames.

D Additional Visualization

We show more visualizations about trajectory motion segmentation and camera localization in Figure 8. Sequences are from 3DPW [44], Youtube-VOS [80], GOT-10K [28] and BANMO [83] dataset. See the attached video for better experience.

Table 6. Per-scene results on MPI Sintel dataset [10].

	Metrics	COLMAP [63]	MAT [95] + [63]	Mask-RCNN [25] + [63]	R-CVD [36]	Tartan-VO [76]	DROID-SLAM [72]	Ours
3*alley_2	ATE (m)	0.072	**0.002**	0.072	0.026	0.062	0.057	**0.002**
	RPE trans (m)	0.039	0.0009	0.038	0.056	0.049	0.035	**0.0007**
	RPE rot (deg)	0.678	0.014	0.679	0.821	0.856	1.047	**0.009**
3*ambush_4	ATE (m)	0.030	0.174	**0.029**	0.171	0.100	0.104	0.068
	RPE trans (m)	0.032	0.046	0.045	0.048	0.038	0.035	**0.027**
	RPE rot (deg)	0.377	3.425	0.541	3.025	1.320	1.385	**0.473**
3*ambush_5	ATE (m)	0.028	0.090	0.004	0.230	0.098	0.112	**0.002**
	RPE trans (m)	0.015	0.036	0.005	0.046	0.037	0.029	**0.001**
	RPE rot (deg)	0.607	0.817	0.204	4.105	1.107	1.580	**0.055**
3*ambush_6	ATE (m)	X	X	X	**0.199**	0.205	0.289	0.269
	RPE trans (m)	X	X	X	0.112	0.107	**0.078**	0.081
	RPE rot (deg)	X	X	X	4.147	4.293	4.596	**0.951**
3*cave_2	ATE (m)	X	X	X	**0.596**	1.167	0.351	0.961
	RPE trans (m)	X	X	X	0.171	**0.131**	0.172	0.142
	RPE rot (deg)	X	X	X	7.508	4.112	5.489	**3.678**
3*cave_4	ATE (m)	0.051	0.049	**0.044**	0.179	0.120	0.155	0.068
	RPE trans (m)	0.013	0.028	0.040	0.087	0.039	0.035	**0.012**
	RPE rot (deg)	0.451	0.700	0.600	2.040	1.327	2.710	**0.409**
3*market_2	ATE (m)	X	X	X	0.032	0.068	0.011	**0.003**
	RPE trans (m)	X	X	X	0.018	0.007	**0.006**	0.010
	RPE rot (deg)	X	X	X	0.141	0.090	**0.036**	0.041
3*market_5	ATE (m)	1.105	0.284	0.816	1.213	1.158	0.912	**0.012**
	RPE trans (m)	0.210	0.095	0.212	0.762	0.294	0.293	**0.006**
	RPE rot (deg)	2.232	0.055	2.380	1.863	1.100	3.334	**0.034**
3*market_6	ATE (m)	X	X	X	0.248	0.260	0.057	**0.018**
	RPE trans (m)	X	X	X	0.214	0.110	0.037	**0.007**
	RPE rot (deg)	X	X	X	0.817	1.287	1.296	**0.120**
3*shaman_3	ATE (m)	0.012	0.006	0.009	0.054	0.008	0.001	**0.0005**
	RPE trans (m)	0.003	0.005	0.007	0.023	0.006	0.002	**0.0004**
	RPE rot (deg)	0.537	0.978	0.977	0.718	0.185	0.199	**0.072**
3*sleeping_1	ATE (m)	**0.008**	0.013	**0.008**	0.029	0.017	0.011	**0.008**
	RPE trans (m)	**0.001**	0.006	**0.001**	0.019	0.011	0.006	**0.001**
	RPE rot (deg)	0.053	0.530	0.046	0.668	0.344	0.479	**0.042**
3*sleeping_2	ATE (m)	**0.0002**	**0.0002**	**0.0002**	0.043	0.013	0.005	0.0007
	RPE trans (m)	**0.0002**	**0.0002**	**0.0002**	0.049	0.022	0.0177	**0.0002**
	RPE rot (deg)	0.008	**0.006**	0.007	0.446	0.267	0.139	**0.006**
3*temple_2	ATE (m)	**0.004**	0.006	0.006	1.245	0.447	0.073	0.011
	RPE trans (m)	0.003	**0.002**	**0.002**	0.394	0.324	0.348	**0.002**
	RPE rot (deg)	0.012	**0.007**	0.013	1.318	0.789	1.298	0.019
3*temple_3	ATE (m)	X	X	X	0.769	0.331	**0.310**	0.381
	RPE trans (m)	X	X	X	0.161	0.105	**0.093**	0.149
	RPE rot (deg)	X	X	X	20.592	**1.166**	3.230	1.583

Table 7. Per-scene results on ScanNet dataset [13].

	Metrics	COLMAP [63]	R-CVD [36]	Tartan-VO [76]	DROID-SLAM [72]	Ours
3***scene0707_00**	ATE (m)	**0.147**	0.442	0.418	0.978	0.199
	RPE trans (m)	0.059	0.092	0.065	0.043	**0.020**
	RPE rot (deg)	0.803	7.301	2.914	3.530	**0.574**
3***scene0709_00**	ATE (m)	**0.143**	0.437	0.202	0.872	0.220
	RPE trans (m)	0.067	0.088	0.063	0.052	**0.015**
	RPE rot (deg)	0.780	6.852	2.698	3.187	**0.523**
3***scene0710_00**	ATE (m)	**0.073**	0.429	0.306	0.631	0.247
	RPE trans (m)	0.016	0.039	0.027	0.030	**0.012**
	RPE rot (deg)	**0.371**	4.412	1.961	2.153	0.424
3***scene0712_00**	ATE (m)	**0.051**	0.183	0.514	0.639	0.232
	RPE trans (m)	**0.016**	0.021	0.025	0.017	**0.016**
	RPE rot (deg)	**0.383**	3.807	1.943	2.221	0.619
3***scene0713_00**	ATE (m)	**0.204**	0.472	0.515	0.616	0.309
	RPE trans (m)	0.124	0.090	0.047	0.047	**0.024**
	RPE rot (deg)	9.536	18.216	3.437	4.127	**1.287**
3***scene0714_00**	ATE (m)	0.891	0.644	0.389	0.916	**0.372**
	RPE trans (m)	0.215	0.075	0.064	0.045	**0.019**
	RPE rot (deg)	9.190	7.498	2.630	3.485	**0.418**
3***scene0715_00**	ATE (m)	0.267	**0.230**	0.239	0.511	0.341
	RPE trans (m)	0.156	0.039	0.049	0.039	**0.026**
	RPE rot (deg)	15.059	8.835	2.930	3.524	**0.611**
3***scene0717_00**	ATE (m)	**0.091**	0.324	0.508	0.782	0.252
	RPE trans (m)	0.040	0.050	0.058	0.040	**0.022**
	RPE rot (deg)	0.586	7.267	3.006	3.453	**0.555**
3***scene0718_00**	ATE (m)	X	0.350	**0.111**	0.385	0.295
	RPE trans (m)	X	0.080	0.065	0.066	**0.039**
	RPE rot (deg)	X	12.460	3.837	5.189	**0.844**
3***scene0719_00**	ATE (m)	**0.051**	0.373	0.171	0.657	0.268
	RPE trans (m)	0.019	0.041	0.044	0.031	**0.012**
	RPE rot (deg)	**0.330**	6.919	2.423	3.380	0.401
3***scene0720_00**	ATE (m)	**0.133**	0.390	0.331	0.389	0.815
	RPE trans (m)	**0.040**	0.034	0.030	0.027	0.021
	RPE rot (deg)	0.900	**0.668**	2.002	1.915	0.875
3***scene0721_00**	ATE (m)	X	0.521	**0.259**	1.345	0.625
	RPE trans (m)	X	0.054	**0.042**	0.070	0.110
	RPE rot (deg)	X	6.439	**2.022**	2.260	5.304
3***scene0722_00**	ATE (m)	**0.050**	0.427	0.319	0.486	0.467
	RPE trans (m)	0.027	0.041	0.042	0.031	**0.019**
	RPE rot (deg)	**0.444**	8.193	2.943	3.523	0.489
3***scene0723_00**	ATE (m)	**0.139**	0.766	0.483	0.521	0.220
	RPE trans (m)	0.031	0.079	0.036	0.028	**0.015**
	RPE rot (deg)	0.796	5.675	2.201	2.304	**0.603**
3***scene0724_00**	ATE (m)	**0.062**	0.647	0.429	0.702	0.332
	RPE trans (m)	0.040	0.090	0.036	0.027	**0.013**
	RPE rot (deg)	**0.854**	5.857	2.796	3.218	1.022
3***scene0725_00**	ATE (m)	X	0.714	0.550	0.882	**0.548**
	RPE trans (m)	X	0.075	0.036	0.027	**0.013**
	RPE rot (deg)	X	9.665	2.272	2.612	**0.708**
3***scene0726_00**	ATE (m)	**0.100**	0.474	0.258	0.380	0.193
	RPE trans (m)	0.051	0.055	0.043	0.035	**0.011**
	RPE rot (deg)	0.563	6.567	2.524	2.904	**0.458**

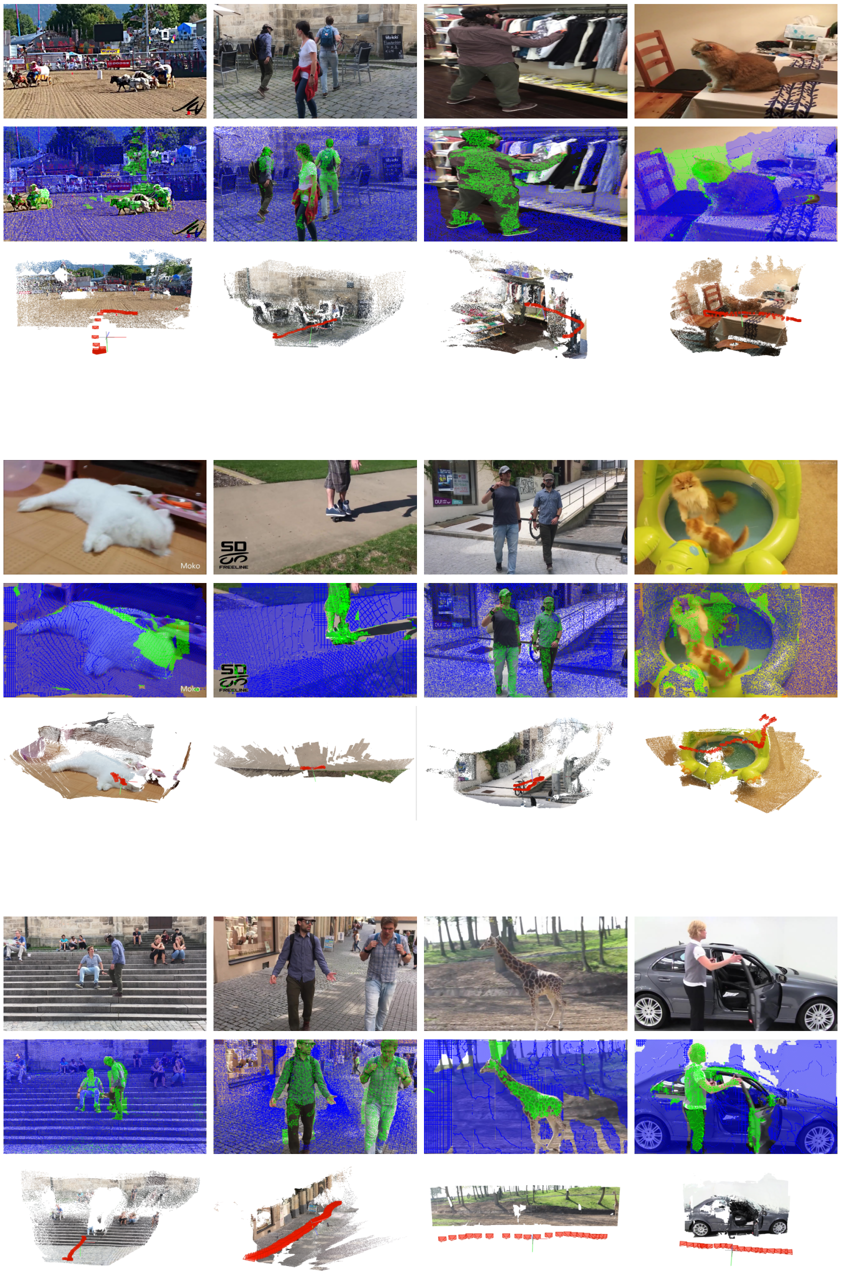


Fig. 8. Visualization of trajectory motion segmentation and camera localization of in-the-wild videos. **Moving pixels from point trajectories are colored in green and static background pixels are in blue. Free space with no colored pixels indicates that there are no trajectory points due to occlusion or large flow forward-backward consistency error.**

References

1. Alcantarilla, P.F., Yebes, J.J., Almazán, J., Bergasa, L.M.: On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In: 2012 IEEE International Conference on Robotics and Automation. pp. 1290–1297. IEEE (2012)
2. Alismail, H., Browning, B., Lucey, S.: Photometric bundle adjustment for vision-based slam. In: Asian Conference on Computer Vision. pp. 324–341. Springer (2016)
3. Ballester, I., Fontan, A., Civera, J., Strobl, K.H., Triebel, R.: Dot: dynamic object tracking for visual slam. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 11705–11711. IEEE (2021)
4. Bârsan, I.A., Liu, P., Pollefeys, M., Geiger, A.: Robust dense mapping for large-scale dynamic environments. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 7510–7517. IEEE (2018)
5. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. IEEE Robotics and Automation Letters **3**(4), 4076–4083 (2018)
6. Bideau, P., Learned-Miller, E.: It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In: European Conference on Computer Vision. pp. 433–449. Springer (2016)
7. Bideau, P., Menon, R.R., Learned-Miller, E.: Moa-net: self-supervised motion segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)
8. Brox, T., Bruhn, A., Weickert, J.: Variational motion segmentation with level sets. In: European Conference on Computer Vision. pp. 471–483. Springer (2006)
9. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE transactions on pattern analysis and machine intelligence **33**(3), 500–513 (2010)
10. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European conference on computer vision. pp. 611–625. Springer (2012)
11. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 521–528 (2013)
12. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. International Journal of Computer Vision **62**(3), 249–265 (2005)
13. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
14. Dave, A., Tokmakov, P., Ramanan, D.: Towards segmenting anything that moves. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
15. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

17. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
18. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint description and detection of local features. In: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition. pp. 8092–8101 (2019)
19. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **40**(3), 611–625 (2017)
20. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014)
21. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 15–22. IEEE (2014)
22. Fragkiadaki, K., Zhang, G., Shi, J.: Video segmentation by tracing discontinuities in a trajectory embedding. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1846–1853. IEEE (2012)
23. Germain, H., Bourmaud, G., Lepetit, V.: S2dnet: learning image features for accurate sparse-to-dense matching. In: European Conference on Computer Vision. pp. 626–643. Springer (2020)
24. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Transactions on pattern analysis and machine intelligence **19**(6), 580–593 (1997)
25. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
27. Huang, J., Yang, S., Mu, T.J., Hu, S.M.: Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2168–2177 (2020)
28. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(5), 1562–1577 (2019)
29. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
30. Jain, S.D., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). pp. 2117–2126. IEEE (2017)
31. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: Proceedings of the IEEE international conference on computer vision. pp. 3271–3279 (2015)
32. Kim, D.H., Kim, J.H.: Effective background model-based rgb-d dense visual odometry in a dynamic environment. IEEE Transactions on Robotics **32**(6), 1565–1573 (2016)
33. Kim, P., Coltin, B., Kim, H.J.: Low-drift visual odometry in structured environments by decoupling rotational and translational motion. In: 2018 IEEE international conference on Robotics and automation (ICRA). pp. 7247–7253. IEEE (2018)

34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
35. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. pp. 225–234. IEEE (2007)
36. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1611–1621 (2021)
37. Lamdouar, H., Yang, C., Xie, W., Zisserman, A.: Betrayed by motion: Camouflaged object discovery via motion segmentation. In: Proceedings of the Asian Conference on Computer Vision (2020)
38. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR 2011. pp. 3369–3376. IEEE (2011)
39. Li, S., Lee, D.: Rgb-d slam in dynamic environments using static point weighting. IEEE Robotics and Automation Letters **2**(4), 2263–2270 (2017)
40. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. IEEE transactions on pattern analysis and machine intelligence **33**(5), 978–994 (2010)
41. Liu, R., Wu, Z., Yu, S., Lin, S.: The emergence of objectness: Learning zero-shot segmentation from videos. In: NeurIPS (2021)
42. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
43. Lu, X., Wang, W., Shen, J., Crandall, D., Luo, J.: Zero-shot video object segmentation with co-attention siamese networks. T-PAMI (2020)
44. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 601–617 (2018)
45. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
46. Min, Z., Yang, Y., Dunn, E.: Voldor: Visual odometry from log-logistic dense optical flow residuals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4898–4909 (2020)
47. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3248–3255 (2013)
48. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
49. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: 2011 international conference on computer vision. pp. 2320–2327. IEEE (2011)
50. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 614–621. IEEE (2012)
51. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE transactions on pattern analysis and machine intelligence **36**(6), 1187–1200 (2013)

52. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2674–2683 (2015)
53. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
54. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016)
55. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
56. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019)
57. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: 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. pp. 12240–12249 (2019)
58. Resch, B., Lensch, H., Wang, O., Pollefeys, M., Sorkine-Hornung, A.: Scalable structure from motion for densely sampled videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3936–3944 (2015)
59. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. Advances in neural information processing systems **31** (2018)
60. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)
61. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International Journal of Computer Vision **80**(1), 72–91 (2008)
62. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
63. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
64. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 1219–1225. IEEE (2009)
65. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: Sixth international conference on computer vision (IEEE Cat. No. 98CH36271). pp. 1154–1160. IEEE (1998)
66. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
67. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
68. Sun, Y., Liu, M., Meng, M.Q.H.: Improving rgb-d slam in dynamic environments: A motion removal approach. Robotics and Autonomous Systems **89**, 110–122 (2017)

69. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by gpu-accelerated large displacement optical flow. In: European conference on computer vision. pp. 438–451. Springer (2010)
70. Sweeney, C.: Theia multiview geometry library: Tutorial & reference. `http://theia-sfm.org`
71. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
72. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in Neural Information Processing Systems **34** (2021)
73. Tokmakov, P., Schmid, C., Alahari, K.: Learning to segment moving objects. International Journal of Computer Vision **127**(3), 282–301 (2019)
74. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
75. Wang, J.Y., Adelson, E.H.: Representing moving images with layers. IEEE transactions on image processing **3**(5), 625–638 (1994)
76. Wang, W., Hu, Y., Scherer, S.: Tartanvo: A generalizable learning-based vo. arXiv preprint arXiv:2011.00359 (2020)
77. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4909–4916. IEEE (2020)
78. Wang, Y., Huang, S.: Motion segmentation based robust rgb-d slam. In: Proceeding of the 11th World Congress on Intelligent Control and Automation. pp. 3122–3127. IEEE (2014)
79. Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: European conference on computer vision. pp. 61–75. Springer (2014)
80. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 585–601 (2018)
81. Yang, C., Lamdouar, H., Lu, E., Zisserman, A., Xie, W.: Self-supervised video object segmentation by motion grouping. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7177–7188 (2021)
82. Yang, G., Ramanan, D.: Learning to segment rigid motions from two frames. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1266–1275 (2021)
83. Yang, G., Vo, M., Natalia, N., Ramanan, D., Andrea, V., Hanbyul, J.: Banmo: Building animatable 3d neural models from many casual videos. arXiv preprint arXiv:2112.12761 (2021)
84. Yang, S., Scherer, S.: Direct monocular odometry using points and lines. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 3871–3877. IEEE (2017)
85. Yang, S., Scherer, S.: Cubeslam: Monocular 3-d object slam. IEEE Transactions on Robotics **35**(4), 925–938 (2019)
86. Yang, Y., Loquercio, A., Scaramuzza, D., Soatto, S.: Unsupervised moving object detection via contextual information separation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 879–888 (2019)
87. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1983–1992 (2018)

88. Yokozuka, M., Oishi, S., Thompson, S., Banno, A.: Vitamin-e: Visual tracking and mapping with extremely dense feature points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9641–9650 (2019)
89. Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q., Fei, Q.: Ds-slam: A semantic visual slam towards dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1168–1174. IEEE (2018)
90. Zhan, H., Weerasekera, C.S., Bian, J.W., Reid, I.: Visual odometry revisited: What should be learnt? In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 4203–4210. IEEE (2020)
91. Zhang, J., Singh, S.: Visual-lidar odometry and mapping: Low-drift, robust, and fast. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 2174–2181. IEEE (2015)
92. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5845–5854 (2019)
93. Zhang, J., Henein, M., Mahony, R., Ila, V.: Vdo-slam: a visual dynamic object-aware slam system. arXiv preprint arXiv:2005.11052 (2020)
94. Zhao, W., Liu, S., Shu, Y., Liu, Y.J.: Towards better generalization: Joint depth-pose learning without posenet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9151–9161 (2020)
95. Zhou, T., Wang, S., Zhou, Y., Yao, Y., Li, J., Shao, L.: Motion-attentive transition for zero-shot video object segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13066–13073 (2020)