CTO-SLAM: Contour Tracking for Object-Level Robust 4D SLAM

Xiaohan Li[1], **Dong Liu**[1], **Jun Wu**[2*]

[1]Institute of Advanced Technology, University of Science and Technology of China
[2]Fudan University
li2xh@mail.ustc.edu.cn, dongeliu@ustc.edu.cn, wujun@fudan.edu.cn

Abstract

The demand for 4D (3D+time) SLAM system is increasingly urgent, especially for decision-making and scene understanding. However, most of the existing simultaneous localization and mapping (SLAM) systems primarily assume static environments. They fail to represent dynamic scenarios due to the challenge of establishing robust long-term spatiotemporal associations in dynamic object tracking. We address this limitation and propose CTO-SLAM, a monocular and RGB-D object-level 4D SLAM system to track moving objects and estimate their motion simultaneously. In this paper, we propose contour tracking, which introduces contour features to enhance the keypoint representation of dynamic objects and coupled with pixel tracking to achieve long-term robust object tracking. Based on contour tracking, we propose a novel sampling-based object pose initialization algorithm and the following adapted bundle adjustment (BA) optimization algorithm to estimate dynamic object poses with high accuracy. The CTO-SLAM system is verified on both KITTI and VKITTI datasets. The experimental results demonstrate that our system effectively addresses cumulative errors in long-term spatiotemporal association and hence obtains substantial improvements over the state-of-the-art systems. The source code is available at https://github.com/realXiaohan/CTO-SLAM.

Introduction

Simultaneous Localization and Mapping (SLAM) defines the problem of generating a map and estimating camera poses when robots enter unknown environment. Visual SLAM system is the one that relies solely on the on-board cameras. Because images can provide abundant information, visual SLAM has received extensive attention and developed rapidly over the past few decades. Most of the existing SLAM approaches typically operate under the assumption of a strictly static scene and usually consider dynamic objects as outliers (Mur-Artal and Tardós 2017; Engel, Koltun, and Cremers 2017). Thus, They lack enough ability to handle dynamic scenes. While the static assumption remains valid for the scene with controlled environments, it greatly restricts the applicability of SLAM in scenarios involving demanding settings, urban autonomous driving for example.

*Corresponding author.

Rich static structure is certainly beneficial for estimating ego-motion. However, having a comprehensive understanding of the dynamic environment is of paramount importance for meeting the evolving demands of emerging applications, such as AR/VR or autonomous driving. The predominant approach that tackles dynamic environments involves detecting and tracking dynamic objects apart from SLAM system. These methods deal with dynamic environments by utilizing traditional object tracking methods (Bârsan et al. 2018; Rosinol et al. 2020). The camera poses and dynamic object poses are then optimized in a unified bundle adjustment framework. Nevertheless, the accuracy of dynamic object motion estimation is relatively poor, which tends to degrade the accuracy of ego-motion estimation when jointly optimized. Some researchers have taken initial strides toward addressing the object tracking and visual SLAM together, which introduces additional complexity to the problem. Part of the systems are tailored to adapt to specific environments, leveraging different priors to constrain the solution space. However, priors required systems are hard to fulfill the needs of real world applications. A minor group of researchers tracks and estimates dynamic object motion with optical flow inside a feature-based SLAM (Zhang et al. 2020). This certainly provides a rather accurate object motion estimation in some scenarios but is sensitive to lighting changes and occlusion. Moreover, it fails to build a sparse map. In summary, due to the difficulty of establishing long-term data association, existing methods cannot achieve robust and accurate motion estimation for dynamic objects.

Inspired by the pixel tracking methods that track every pixel in images (Jiang et al. 2021; Li, Zhou, and Liu 2023), we seek to develop a long-term spatiotemporal association within SLAM. Since pixel tracking can effectively learn spatiotemporal information, it has advantages over optical flow or appearance-based methods in handling sparse points, withstanding occlusion, and providing robust correspondence for extended periods. However, pixel tracking lacks the ability to select robust pixels for tracking and to establish reliable 2D-3D correspondences, which is particularly important in dynamic SLAM. Because keypoints located at the object surface often share similar textures, we explore those situated at the object contour with semantic information to track, referred to as contour keypoints.

In this paper, we propose CTO-SLAM, a 4D visual SLAM

system that tracks and estimates camera motion and dynamic object motion simultaneously. Notably, our approach does not rely on any additional prior information for object motion estimation. In order to integrate dynamic objects and camera in a unified system, we present a dynamic object motion initialization algorithm to generate the poses when they are first observed. Then, we propose contour tracking to track contour keypoints with pixel tracking. In the backend, the structure of dynamic objects are optimized in a novel contour tracking based bundle adjustment optimization process. We verified the performance of our CTO-SLAM in KITTI and VKITTI datasets. The experiment results show the effectiveness and potential of our proposed methods, further reinforcing their value and applicability in practical applications. Finally, we explore the limitation of jointly optimizing dynamic object motion and camera ego-motion. In summary, our main contributions are listed as follows.

- The first 4D SLAM system that integrates contour tracking with sparse SLAM system to accurately track dynamic objects and create a strictly sparse map.
- The proposed sampling-based dynamic object motion initialization algorithm accurately provides poses when dynamic objects are first observed. The following contour tracking based bundle adjustment achieves robust and precise estimation of object motion in long duration.
- We discuss the limitation of classical bundle adjustment for jointly optimizing object motion and camera ego-motion thoroughly. Then, we propose dense then sparse method incorporating multi-objects tracking strategies.

Related Work

Based on the objectives in dealing with dynamic objects, research on dynamic SLAM is primarily categorized into three groups. The first and most extensively studied group aims to build an almost absolute static map for ensuring the accuracy of ego-motion estimation. Early works (Alcantarilla et al. 2012; Tan et al. 2013) explored dynamic objects and regarded them as outliers. More recent approaches seek to enrich the static structures by filtering and inpainting methods (Zhang, Fu, and Liu 2022; Li et al. 2022; Zhang et al. 2023). DynaSLAM (Xiao et al. 2019) utilizes deep learning models to identify and eliminate dynamic objects from the scene, then inpaint the background with multiview geometry. The above work can provide accurate static maps but fails to help robot navigation in complex dynamic scenes.

The second category is SLAM-MOT (Ballester et al. 2021; Ren et al. 2022), which utilizes moving objects tracking (MOT) method in an independent thread. (Wang et al. 2007) first developed a system that integrates a filtering-based SLAM with MOT to support navigation in dynamic scenes. Later, (Xu et al. 2019) fuses geometric and semantic information to model and track dynamic objects, leading to a dense mapping of indoor scenes.

The last yet growing category aims to tightly integrate both static structures and dynamic objects in a unified framework to achieve a better scene understanding. CubeSLAM (Yang and Scherer 2019) generated 3D bounding box proposals relying on 2D bounding boxes and vanishing points to estimate pose. Moreover, this system operates under the assumption that objects keep a constant velocity within a predetermined duration and utilizes geometric information of objects to refine pose estimation. ClusterSLAM (Huang et al. 2019) is a backend with a prior-free manner to track and estimate the motion of the dynamic objects. However, it is not a comprehensive SLAM system and the efficacy is intricately linked to the quality of data association. Later on, they developed ClusterVO (Huang et al. 2020) to make a full SLAM system, which incorporates a probability-based model for object points to address the challenges of segmentation inaccuracies. ClusterVO achieves commendable tracking results in both indoor and outdoor settings, but the accuracy of object motion estimation is rather unpromising. Recently, VDO-SLAM (Zhang et al. 2020) applied dense optical flow to optimize the points residing on dynamic objects and put cameras, objects, and points in a bundle adjustment framework. This work yields favorable outcomes, but its real-time implementation is hindered by the computational complexity and fails to create a strictly sparse map.

Beyond dynamic SLAM, research on object motion estimation plays a pivotal role in addressing the challenges of dynamic SLAM. Methods based on optical flow or scene flow fail to find long-term correspondence and cannot seamlessly integrate with feature-based SLAM. Recently, there has been significant progress in the development of pixel tracking to estimate the motion of points over extended durations. For the sake of efficiency and simplicity, these methods typically concentrate on a sparse selection of points and treat them as statistically independent entities. TAP-Vid (Doersch et al. 2022) addresses the problem of tracking any physical point in a video. The method computes a cost volume independently for each pair of frames, which introduces a straightforward baseline approach for this task.

In this paper, we delve into the spatiotemporal information of dynamic objects and propose a dynamic object-aware sparse visual SLAM system. It achieves robust ego-motion and object motion tracking, as well as consistent static and dynamic mapping in a novel SLAM formulation.

Notation

Coordinate Frames and Points

We denote T_{wc}^k and $T_{wo}^k \in SE(3)$ as camera pose and object pose in world space $\mathcal{W}$ at time k, where $k \in \mathcal{T}$ is the set of time steps. The calligraphic capital letters are used to represent sets of indices. We denote $P_{\mathcal{W}_i}^k$ as the homogeneous coordinates of the ith 3D world point at time k, where $P_{\mathcal{W}_i}^k \in \mathbb{E}^3$ and $i \in \mathcal{U}$ represents the set of 3D points in world coordinate $\mathcal{W}$. If an image is captured at time k, the pixel coordinate origin lies in the top left corner of the image, and the pixel $p_i^k \in \mathbb{E}^2$ projected from world coordinate is:

$$p_i^k = \pi(T_{wc}^k P_i^k) = K T_{wc_i}^k P_{\mathcal{W}_i}^k, \tag{1}$$

where $\pi(\cdot)$ is the projection function according to pinhole camera model and K is the intrinsics matrix of the camera.

Object and 3D Point Motions

Assume a keypoint $P_{\mathcal{W}_i}^{k-1}$ is detected on the dynamic object contour at time $k-1$, the pixel match at time k is de-

termined with the contour tracking and the corresponding 3D point is calculated with depthmap and K, denoted as $P^k_{\mathcal{W}_i}$. For contour keypoint, the motion between $P^{k-1}_{\mathcal{W}_i}$ and $P^k_{\mathcal{W}_i}$ includes both camera ego-motion and object motion $T^k_{wo} = T^k_{wc} \cdot (T^k_{oc})^{-1}$. The projection of contour keypoint from object coordinates to pixel coordinates is:

$$p^k_i = K(T^k_{wc})^{-1} T^k_{wo} P^k_{\mathcal{O}_i}, \tag{2}$$

where $\mathcal{O}$ denotes the object coordinates. A contour keypoint moves from time $k-1$ to time k and the projection from object coordinates to world coordinates is:

$$P^k_{\mathcal{W}_i} = T^k_{wo} P^k_{o_i} = T^{k-1}_{wo} T^{O_{k-1}}_{O_k} P^k_{O_i}. \tag{3}$$

Since most of the dynamic objects are rigid in autonomous driving, cars for example, the contour keypoints remain unchanged in object coordinates. With this constrain, we have:

$$P^k_{\mathcal{O}_i} = (T^k_{wo})^{-1} P^k_{\mathcal{W}_i} = (T^{k-1}_{wo})^{-1} P^{k-1}_{\mathcal{W}_i}. \tag{4}$$

Combine equations 3 and 4, the projection of a contour keypoint from object coordinates to world coordinates between time $k-1$ to time k is:

$$\begin{aligned} P^k_{\mathcal{W}_i} &= T^{k-1}_{wo} T^{k-1}_o T^k_o P^k_{\mathcal{O}_i} \\ &= T^{k-1}_{wo} T^{k-1}_o T^k_o (T^{k-1}_{wo})^{-1} P^{k-1}_{\mathcal{W}_i}. \end{aligned} \tag{5}$$

Equation 5 serves as the cornerstone of our dynamic object motion estimation. It articulates the transformation of the rigid object pose, solely concerning the points residing on the object contour. Remarkably, this equation obviates the requirement to incorporate the object's 3D pose as a random variable during the estimation process. In the context of our paper, we denote $\mathsf{M}^k_{k-1} := T^{k-1}_{wo} T^{k-1}_o T^k_o (T^{k-1}_{wo})^{-1}$ as the motion of object in global reference frame.

In this section, we present a novel contour tracking based 4D SLAM system that effectively captures the movements of both the camera and dynamic objects, while accurately reconstructing the static and dynamic elements of the environment. A comprehensive illustration of the entire system is in Fig. 1. Similar to feature-based SLAM, the proposed system comprises three primary components: image preprocessing, tracking, and mapping. Through the integration of these components, our proposed system achieves robust estimation of camera motion and object motions over a long-term period, enabling a detailed understanding of the spatial structures and temporal dynamics of the environment. CTO-SLAM takes monocular or RGB-D image streams as input. Specifically, we implement a depthNet (Casser et al. 2019) to generate depthmap for monocular setup.

Pre-processing

Instance-level semantic segmentation plays a crucial role in segmenting and identifying potentially movable objects within a scene. This semantic understanding serves as a valuable prior in separating points belonging to static and potentially moving objects. By aligning the masks with intersection over union (IoU), the system is able to determine which instance represents the truly dynamic object, enhancing our ability to track and monitor each object's motion. Furthermore, the masks provide a precise boundary delineating the object's body-frame. It ensures robust tracking of contour keypoints, allowing for more accurate and reliable tracking results.

Ego-motion Tracking

The core contribution is the proposed novel tracking method that can robustly track and estimate the motion of both static background and dynamic objects simultaneously. To sum up, there are two different tracking threads in this module. One is the ego-motion tracking thread with sub-modules of dense prediction, feature detection and pose estimation. The other is the object motion tracking thread including sub-modules of dynamic object pose initialization, contour tracking and object motion estimation. This part first explains the camera tracking thread in detail.

Dense prediction The accuracy of camera pose estimation is essential for the object pose estimation according to equation 5. To achieve a more accurate camera pose estimation, we first modify a dense optical flow based pose estimation network called DROID-SLAM (Teed and Deng 2021) to generate the coarse pose guess after pre-processing. The coarse guess performs as prior which is then optimized in feature-based SLAM bundle adjustment.

Camera Pose Estimation Our system develops upon the widely-used ORB-SLAM II (Mur-Artal and Tardós 2017). After we carefully segment the absolute dynamic object, the camera pose guess is refined from all detected 3D-2D static point correspondences. The rest process including tracking and optimization is similar to ORB-SLAM II which we present the details in supplementary material.

Mapping

The mapping thread first builds a global map that maintains the poses and keypoints of both static and dynamic structures. The local map is extracted from the global map with a sliding window. Both the global and local maps are updated via a batch optimization process. Moreover, the maintenance of the global map is depicted in Fig. 1 through the blue arrows. It links the global map with various components including camera pose, dynamic poses, static keypoints and contour keypoints from the tracking thread. With the robot exploring dynamic environment, the map keeps updating with time and creates a 4D map.

Object Motion Tracking

In this part, we provide a comprehensive introduction for the object motion tracking thread which includes three modules, dynamic pose initialization, object contour tracking and dynamic pose estimation. Firstly, dynamic object pose initialization estimates the 6DoF pose when it is first seen, which provides a robust foundation for the following tracking process. Then, contour tracking is applied to track contour keypoints. We extract keypoints associated with semantic information, light and wheel for example, in the edge of dynamic objects. Next, the pixel tracking method is applied to establish a robust and effective spatiotemporal association in 2D

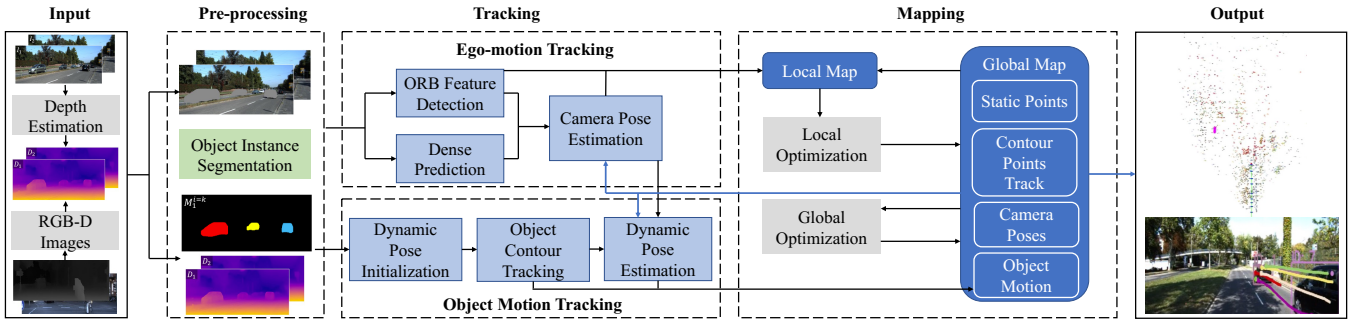

Figure 1: Overview of the proposed CTO-SLAM. This system first performs instance segmentation and locates the real dynamic objects. The static background is first used to estimate a prior by employing a PoseNet. The camera pose prior is then refined by tracking the static keypoints. Object motions are initialized when they are first observed and then the incremental poses are estimated from contour tracking. Finally, object poses are optimized in a novel contour tracking based bundle adjustment. The system outputs camera poses, static structure, tracks of dynamic objects, and estimates of their motions over time.

image plane over a long period. Notably, CTO-SLAM system is the first to apply pixel tracking in SLAM area based on our current knowledge. Finally, a novel contour tracking based bundle adjustment is designed to optimize the pose from the above two modules within a sliding window.

Dynamic Pose Initialization

Dynamic object pose initialization is crucial for accurately positioning objects in the global map when they are first observed. Most of the existing methods initialize dynamic object poses with learning-based ways. However, learning-based ways are limited in generation and hard to adapt to complex autonomous driving scenarios. Hence, we propose object pose initialization to robustly estimate a 6DoF pose when the dynamic object is first seen.

The pipeline of object pose initialization is depicted in Fig. 2. Since almost all the dynamic objects in autonomous driving scenarios lie on the ground, it is reasonable to assign pitch and roll angle to zero. Then, the mission for the dynamic object pose initialization is to find the best yaw angle. Thus, we first fit the ground using RANSAC ground fitting algorithm and generate the ground equation, denoted as $n_1x + n_2y + n_3z + d = 0$, where $\vec{n} = [n_1, n_2, n_3]$ is the normal vector of the fitted plane and d is the distance. With the ground equation, we adjust the current camera pose to ensure that the ground is horizontally aligned with world coordinates and has a 0 height. While the ground is established, we first project the dynamic object's mask onto world coordinates and sample the yaw angle discretely across a two-dimensional projection plane, ranging from 0 to 180 degrees. For each sample, a bounding box is calculated with minimal area that encompasses the entire 2D projection region. This bounding box is then utilized as the top view of the cuboid structure, enabling the estimation of the cuboid's parameters $[tx, ty, yaw, l, w]$. Subsequently, the cuboid is oriented to enclose all 3D points along the yaw direction, thereby determining the remaining two parameters $[tz, h]$. Finally, the initial pose of dynamic object is generated.

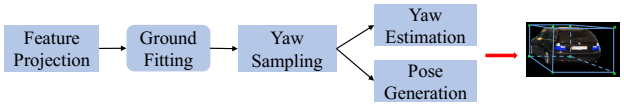

Figure 2: Pipeline of dynamic object pose initialization.

Dynamic Object Tracker

Mask Propagation Strategies Ideally, for each detected object in frame k, the labels of all its points should be aligned with the labels of their correspondences in frame $k-1$. However, the association is affected by the noise, image boundaries and occlusions. To overcome this, we assign all the points with the label that appears most in their correspondences and propagate the masks if they surpass a predetermined threshold of 2D IoU. Moreover, if the most frequent label in the last frame is 0, it means that the object starts to move, appears in the scene at the boundary, or reappears from occlusion. In this case, the object is assigned a new label and re-detect contour points for tracking.

Object Motion Estimation Objects normally appear as small portions in the scene, which is hard to get sufficient sparse features for robust and accurate tracking with feature-based SLAM system. Thanks to the proposed contour tracking, CTO-SLAM is able to set a robust long-term spatiotemporal association and hence track the contour keypoints with a high accuracy. The pipeline of our dynamic object motion estimation is depicted as Fig. 3. Different from camera pose estimation, contour tracking is first utilized to track multi-objects with a small amount number of contour keypoints. A

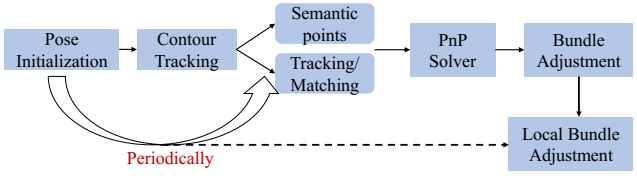

Figure 3: Pipeline of our dynamic object motion estimation.

simple but effective PnP (Perspective-n-Point) algorithm (Lu 2018) is then applied to give an initial guess of the object motion. PnP is a classical way to solve the pose when 2D-3D point correspondences are known between consecutive frames. Specifically, P3P only requires three 2D-3D correspondences to calculate the motion. Benefiting from the effective spatiotemporal association of our contour tracking, the estimation of dynamic object motion requires a minimum of four 2D-3D correspondences, as an extra correspondence to verify the calculated pose. In our paper, we uniformly extract ten contour keypoints for an object and make a robust guess with RANSAC. Moreover, a cost function based on re-projection error is constructed to optimize the object motion estimation. When the initial guess is generated from PnP Solver, the re-projection error of a contour keypoint in frame I_k is:

$$\begin{aligned} e_i({}_{O_{k-1}}T_{O_k}) &= \tilde{p}_i^k - \pi(T_{wo}^{k-1}({}_{O_{k-1}}T_{O_k})P_{O_i}^k) \\ &= \tilde{p}_i^k - \pi(T_{wo}^k P_{O_i}^k), \end{aligned} \tag{6}$$

where $T_{wo}^k \in SE(3)$. Parameterize ${}^{\mathcal{O}}\xi_k \in se(3)$ with Lie-algebra. The least squares cost function is expressed as:

$${}^{\mathcal{O}}\xi_k^{*\vee} = \underset{{}^{\mathcal{O}}\xi_k^{*\vee}}{\arg\min} \sum_i^m \mathsf{H}(\cdot)(e_i^\top({}^{\mathcal{O}}\xi_k)\Sigma_{ik}^{-1}e_i({}^{\mathcal{O}}\xi_k)), \tag{7}$$

where m is the number of contour keypoint correspondences. The motion M_{k-1}^k can be recovered afterwards. Due to occlusion, the number of contour points gradually decreases and leads to tracking failure. To ensure the robustness of P3P, new contour keypoints are detected and added to the map when previous contour keypoints are occluded.

Object Motion Optimization SLAM problem is usually transformed into a factor graph optimization, aiming to enhance the accuracy of both camera and object motions. In the object tracking thread, we design a contour tracking based factor graph framework to further refine the object poses over a period of time. To represent the dynamic scene, the camera pose is first fixed and object motion is optimized with a sliding window. Besides, the object motion initialization algorithm is periodically called to correct and align the ground over every 5 keyframes. Finally, all contour keypoints, camera poses, object poses and sampling-based poses are integrated to form a factor graph and optimized within it. Because object poses generated by heuristic methods typically exhibit higher accuracy (Henein et al. 2020), we increase the weight of sampling-based poses in factor graph for better optimization.

Experiments

To assess the performance of CTO-SLAM, we first evaluate the ego-motion estimation process, then analyze the performance of multi-object tracking and finally discuss the limitation of jointly optimizing ego-motion and dynamic motion. The experiment is performed on two datasets: Virtual KITTI dataset (VKITTI) (Cabon, Murray, and Humenberger 2020) and KITTI Tracking dataset (Geiger et al. 2013). Both datasets are for outdoor scenarios and rich in dynamic objects, lighting changes and occlusions. To gauge the effectiveness of our CTO-SLAM, we compare its results with some sota methods, including DynaSLAM (Bescos et al. 2018), ORB-SLAM II (Mur-Artal and Tardós 2017) and DynaSLAM II (Bescos et al. 2021), allowing us to gain insights into the strengths and weaknesses of CTO-SLAM with these existing approaches. To address the non-deterministic nature of the proposed system, particularly in processes like RANSAC, each sequence is executed five times, and the median is taken as the result.

Datasets

The KITTI Tracking dataset serves as a valuable resource for evaluating CTO-SLAM. The VKITTI dataset is derived from the KITTI tracking benchmark. Because the lighting intensity remains relatively constant in the VKITTI dataset and there is an accurate distinction between static backgrounds and dynamic objects, we utilize this dataset to explore the effectiveness of jointly optimizing camera poses and dynamic object poses. All these two datasets provide groundtruth for both camera motion and dynamic object motions. Moreover, for the error metric, it is the same with (Sturm et al. 2012).

Ego-Motion Estimation

We test our dense then sparse method on KITTI Tracking datasets. Fig. 4 provides the qualitative evaluation of KITTI Tracking datasets and Tab. 1 provides the quantitative evaluation. In addition, we compare our method with the sota feature-based SLAM system including ORB-SLAM2, DynaSLAM and DynaSLAM II in ego-motion estimation. Results of DynaSLAM and DynaSLAM II are obtained directly from their paper. From the result, it demonstrates that the proposed CTO-SLAM gains competitive accuracy over the compared methods. Specifically, our CTO-SLAM shows an average accuracy improvement of 20% compared to the three methods in relative pose error (RPE). For absolute trajectory error (ATE), CTO-SLAM has an average improvement of 55% compared with ORB-SLAM2, 28% with DynaSLAM and 32% with DynaSLAM II respectively.

Dynamic Object Motion Estimation

We present the evaluation of our proposed dynamic object motion estimation. The quantitative results are shown in Tab. 2. Due to the excessively large errors in CubeSLAM, which render it devoid of comparative value, we solely choose DynaSLAM II to compare with. From the result, it is evident that the CTO-SLAM exhibits an average accuracy improvement of 33% compared to DynaSLAM II in ATE, indicating that our CTO-SLAM system performs better than the state-of-the-art SLAM system in dynamic environments.

Ablation Study

Contour Tracking Compared to traditional methods that track appearance-based feature points, pixel tracking methods excel in finding long-term spatiotemporal associations and recovering from occlusion. To illustrate the exceptional accuracy of matching dynamic points with contour tracking, we compare the keypoint tracking trajectories over an extended time in 2D image plane with SIFT (Ng and Henikoff

Method	ORB-SLAM2		DynaSLAM		DynaSLAM II		CTO-SLAM	
Sequence	RPE(m/s)	ATE(m)	RPE(m/s)	ATE(m)	RPE(m/s)	ATE(m)	RPE(m/s)	ATE(m)
00	0.04	1.32	0.04	1.35	0.04	1.29	**0.03**	**1.18**
01	**0.05**	1.95	**0.05**	2.42	**0.05**	2.31	**0.05**	**1.20**
02	**0.04**	0.95	**0.04**	1.04	**0.04**	0.91	**0.04**	**0.83**
03	0.07	0.74	0.07	0.78	0.06	0.69	**0.04**	**0.48**
04	**0.07**	1.44	**0.07**	1.52	**0.07**	1.42	**0.07**	**1.08**
05	0.06	1.23	0.06	1.22	0.06	1.34	**0.02**	**0.10**
06	**0.02**	0.19	**0.02**	0.19	**0.02**	0.19	**0.02**	**0.17**
07	**0.05**	2.47	**0.05**	2.69	**0.05**	3.10	**0.05**	**1.52**
08	0.08	1.40	0.08	1.29	0.10	1.68	**0.07**	**1.18**
09	**0.06**	4.00	**0.06**	3.55	**0.06**	5.02	**0.06**	**3.31**
10	**0.07**	1.68	**0.07**	1.84	**0.07**	**1.30**	0.08	2.11
11	**0.04**	0.97	**0.04**	1.05	**0.04**	1.03	**0.04**	**0.86**
13	**0.04**	1.18	**0.04**	1.18	**0.04**	**1.10**	0.05	1.34
14	**0.03**	0.13	**0.03**	0.13	**0.03**	**0.12**	0.04	0.23
18	**0.05**	0.89	**0.05**	1.00	**0.05**	1.09	**0.05**	**0.70**
19	0.05	2.31	0.05	2.35	0.05	2.25	**0.02**	**0.31**
20	0.11	16.80	**0.05**	**1.10**	0.07	1.36	0.06	1.19
Average	0.05	2.33	0.05	1.45	0.05	1.54	**0.04**	**1.05**

Table 1: Comparison versus ORB-SLAM II, DynaSLAM and DynaSLAM II for ego-motion estimation on KITTI Tracking datasets. Bold numbers indicate the better result.

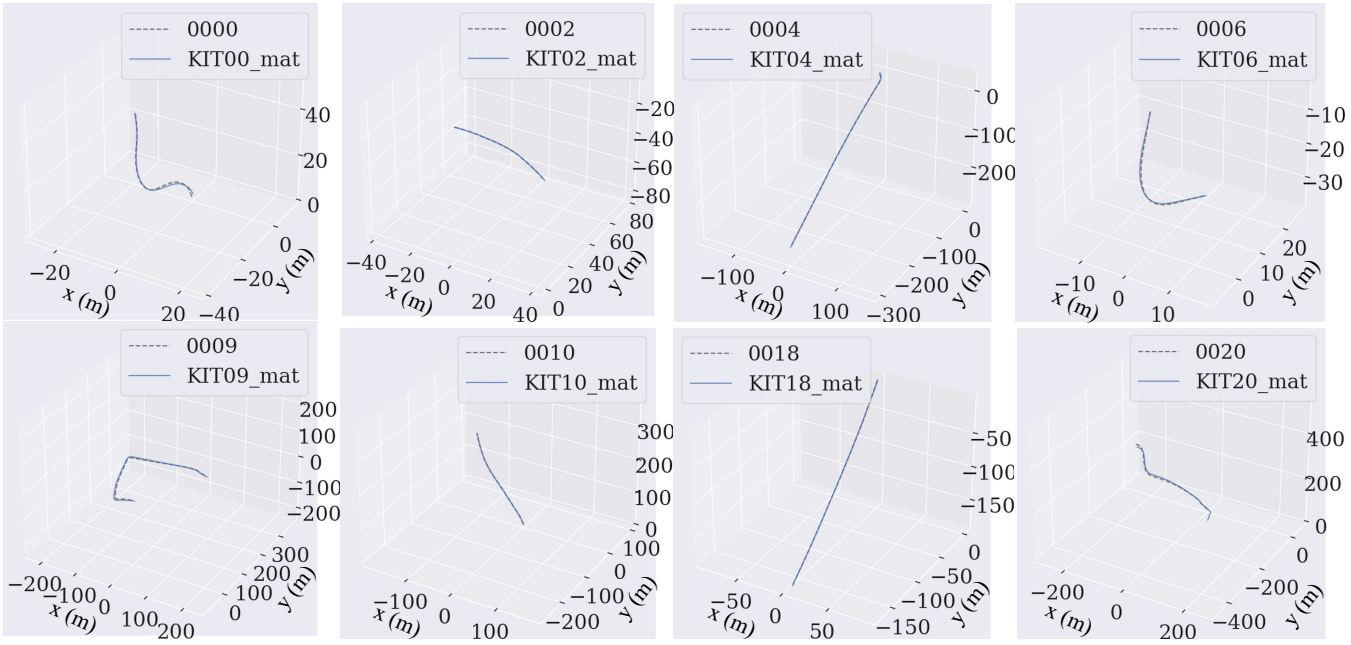

Figure 4: Qualitative results of sequence from KITTI Tracking dataset in ego-motion estimation. The gray dashed line represents groundtruth and the blue line represents the trajectory estimated from the proposed dense then sparse method.

Sequence	ObjectId (class)	DynaSLAM II	Ours
0003	1 (car)	0.69	**0.47**
0011	0 (car)	1.05	**0.66**
	35 (car)	1.25	**0.30**
0018	3 (car)	1.13	**0.93**
0019	63 (car)	0.86	**0.37**
0020	0 (car)	0.56	**0.42**
	12 (car)	1.18	**1.17**
	122 (car)	0.87	**0.80**
Average		0.95	**0.64**

Table 2: Evaluation of dynamic object motion on KITTI Tracking datasets. Bold numbers indicate the better results.

2003) and ORB (Rublee et al. 2011) which are widely used in keypoints matching. The qualitative result is depicted in Fig. 5. Upon analyzing the trajectories with multi-object tracking, it is evident that the SIFT and ORB encounter certain incorrect matches, especially with occlusion (black car in the back). This impacts significantly on the accuracy of motion estimation. However, the trajectory of contour tracking based matches exhibits a higher level of accuracy. These trajectories closely align with the actual motion of dynamic objects within the 2D image plane.

Limitation of Bundle Adjustment with Objects Most SLAM systems assume that ego-motion estimation could benefit from jointly optimizing static background and dynamic objects with bundle adjustment. However, the optimized camera ego-motion is highly related to the accuracy of object motion. Due to the dynamic objects have more complex motion and are hard to establish long-term data association, the accuracy of dynamic motion estimation with feature-based methods is rather lower than that of camera pose estimation. Thus, the lower accuracy of pose estimation for moving objects will unavoidably degrade the accuracy of camera pose estimation when bundle adjustment optimization is applied. In this paper, we design several experiments to explore the limitations of dynamic objects. The experimental settings are categorized based on the richness of static structures, ranging from minimal to maximal. Specifically, we utilized images that contain both foreground and background elements (**Full**), images that solely consist of static backgrounds (**BK**), and images where dynamic foreground objects are inpainted into static backgrounds (**IN**) using E2FGVI (Li et al. 2022). The quantity results of camera motion estimation are shown in Tab. 3. The experimental results indicate that as the static structure is enriched, the accuracy of camera motion estimation improves significantly.

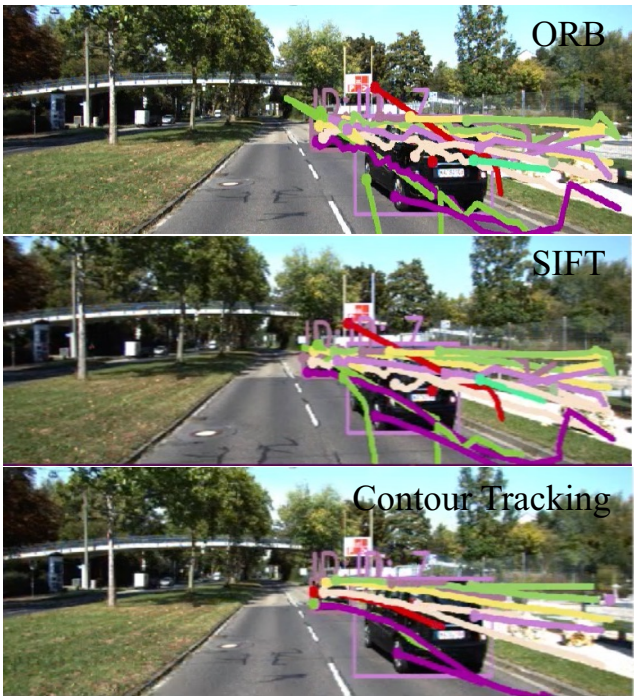

Figure 5: Trajectories of the multi-object tracking with different methods. The top comes from ORB, the middle comes from SIFT and the bottom comes from contour tracking.

Metric	RPE(m/s)		ATE(m)	
Sequence	mean	rmse	mean	rmse
01-Full	0.00160	0.00210	0.11860	0.13200
02-Full	0.00040	0.00061	0.00700	0.00930
06-Full	0.00500	0.00673	0.03660	0.04700
18-Full	0.00200	0.00280	0.08838	0.10220
01-BK	0.00077	0.00099	0.05340	0.05890
02-BK	0.00046	0.00072	0.00945	0.01220
06-BK	0.00575	0.00719	0.03231	0.03498
18-BK	0.00222	0.00312	0.09625	0.11154
01-IN	**0.00033**	**0.00039**	**0.01600**	**0.00175**
02-IN	**0.00035**	**0.00057**	**0.00685**	**0.00893**
06-IN	**0.00015**	**0.00018**	**0.00058**	**0.00092**
18-IN	**0.00185**	**0.00258**	**0.08139**	**0.09390**

Table 3: Results of ego-motion estimation on VKITTI datasets with respect to different richness of static structures. Bold numbers indicate the better results.

Conclusion

We propose an object-level 4D SLAM system incorporating a unique method for tracking object contours and optimizing among cameras, objects, and 3D points. The foundational contour tracking module establishes robust long-term spatiotemporal associations. It excels in scenarios with high dynamics and effectively complements feature-based SLAM systems (ORB-SLAM II). The introduced contour tracking based optimization framework tightly integrates cameras, objects, static keypoints and contour keypoints, achieving high-precision localization of dynamic objects. Experiments reveal the exemplary accuracy of CTO-SLAM in both camera ego-motion and object motion estimation.

Acknowledgments

This work was supported by National Key R&D Program of China under Grant 2020YFA0711400, National Natural

Science Foundation of China under Grants 61931014 and U21A20452, the Fundamental Research Funds for the Central Universities under No. WK3490000006, and the Key-Area Research and Development Program of Guangdong Province under Grant 2018B010115002.

References

Alcantarilla, P. F.; Yebes, J. J.; Almazán, J.; and Bergasa, L. M. 2012. 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*, 1290–1297. IEEE.

Ballester, I.; Fontán, A.; Civera, J.; Strobl, K. H.; and Triebel, R. 2021. DOT: Dynamic object tracking for visual SLAM. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11705–11711. IEEE.

Bârsan, I. A.; Liu, P.; Pollefeys, M.; and Geiger, A. 2018. Robust dense mapping for large-scale dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 7510–7517. IEEE.

Bescos, B.; Campos, C.; Tardós, J. D.; and Neira, J. 2021. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM. *IEEE robotics and automation letters*, 6(3): 5191–5198.

Bescos, B.; Fácil, J. M.; Civera, J.; and Neira, J. 2018. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4): 4076–4083.

Cabon, Y.; Murray, N.; and Humenberger, M. 2020. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*.

Casser, V.; Pirk, S.; Mahjourian, R.; and Angelova, A. 2019. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 8001–8008.

Doersch, C.; Gupta, A.; Markeeva, L.; Recasens, A.; Smaira, L.; Aytar, Y.; Carreira, J.; Zisserman, A.; and Yang, Y. 2022. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35: 13610–13626.

Engel, J.; Koltun, V.; and Cremers, D. 2017. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3): 611–625.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

Henein, M.; Zhang, J.; Mahony, R.; and Ila, V. 2020. Dynamic SLAM: The need for speed. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2123–2129. IEEE.

Huang, J.; Yang, S.; Mu, T.-J.; and Hu, S.-M. 2020. 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*, 2168–2177.

Huang, J.; Yang, S.; Zhao, Z.; Lai, Y.-K.; and Hu, S.-M. 2019. Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5875–5884.

Jiang, W.; Trulls, E.; Hosang, J.; Tagliasacchi, A.; and Yi, K. M. 2021. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6207–6217.

Li, R.; Zhou, S.; and Liu, D. 2023. Learning Fine-Grained Features for Pixel-wise Video Correspondences. *arXiv preprint arXiv:2308.03040*.

Li, Z.; Lu, C.-Z.; Qin, J.; Guo, C.-L.; and Cheng, M.-M. 2022. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17562–17571.

Lu, X. X. 2018. A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics: Conference Series*, volume 1087, 052009. IOP Publishing.

Mur-Artal, R.; and Tardós, J. D. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5): 1255–1262.

Ng, P. C.; and Henikoff, S. 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13): 3812–3814.

Ren, Y.; Xu, B.; Choi, C. L.; and Leutenegger, S. 2022. Visual-inertial multi-instance dynamic SLAM with object-level relocalisation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11055–11062. IEEE.

Rosinol, A.; Gupta, A.; Abate, M.; Shi, J.; and Carlone, L. 2020. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*.

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, 2564–2571. Ieee.

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; and Cremers, D. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580. IEEE.

Tan, W.; Liu, H.; Dong, Z.; Zhang, G.; and Bao, H. 2013. Robust monocular SLAM in dynamic environments. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 209–218. IEEE.

Teed, Z.; and Deng, J. 2021. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34: 16558–16569.

Wang, C.-C.; Thorpe, C.; Thrun, S.; Hebert, M.; and Durrant-Whyte, H. 2007. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9): 889–916.

Xiao, L.; Wang, J.; Qiu, X.; Rong, Z.; and Zou, X. 2019. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems*, 117: 1–16.

Xu, B.; Li, W.; Tzoumanikas, D.; Bloesch, M.; Davison, A.; and Leutenegger, S. 2019. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, 5231–5237. IEEE.

Yang, S.; and Scherer, S. 2019. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4): 925–938.

Zhang, J.; Henein, M.; Mahony, R.; and Ila, V. 2020. VDO-SLAM: a visual dynamic object-aware SLAM system. *arXiv preprint arXiv:2005.11052*.

Zhang, K.; Fu, J.; and Liu, D. 2022. Flow-Guided Transformer for Video Inpainting. In *European Conference on Computer Vision*, 74–90. Springer.

Zhang, K.; Peng, J.; Fu, J.; and Liu, D. 2023. Exploiting Optical Flow Guidance for Transformer-Based Video Inpainting. *arXiv preprint arXiv:2301.10048*.