Fast Direct Stereo Visual SLAM

Jiawei Mo, Md Jahidul Islam, and Junaed Sattar

***Abstract*—We propose a novel approach for fast and accurate stereo visual Simultaneous Localization and Mapping (SLAM) independent of feature detection and matching. We extend monocular Direct Sparse Odometry (DSO) to a stereo system by optimizing the scale of the 3D points to minimize photometric error for the stereo configuration, which yields a computationally efficient and robust method compared to conventional stereo matching. We further extend it to a full SLAM system with loop closure to reduce accumulated errors. With the assumption of forward camera motion, we imitate a LiDAR scan using the 3D points obtained from the visual odometry and adapt a LiDAR descriptor for place recognition to facilitate more efficient detection of loop closures. Afterward, we estimate the relative pose using direct alignment by minimizing the photometric error for potential loop closures. Optionally, further improvement over direct alignment is achieved by using the Iterative Closest Point (ICP) algorithm. Lastly, we optimize a pose graph to improve SLAM accuracy globally. By avoiding feature detection or matching in our SLAM system, we ensure high computational efficiency and robustness. Thorough experimental validations on public datasets demonstrate its effectiveness compared to the state-of-the-art approaches.**

***Index Terms*—SLAM, vision-based navigation.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) has been an active research problem in robotics and computer vision over the past few decades [1], [2]. It deals with estimating a robot's instantaneous location by using onboard sensory measurements, *e.g.*, LiDAR (Light Detection and Ranging) sensors, cameras, and inertial measurement units (IMU). SLAM is particularly useful where GPS reception is weak such as indoor, urban, and underwater environments. Hence, it has been an essential component in AR/VR [3], autonomous driving [4], and GPS-denied robotics applications [5]. Among existing systems, visual SLAM [6] is of significant interest because cameras are low-cost passive sensors and thus consume less energy compared to active ones such as sonar or LiDAR. Autonomous mobile robots operating outdoors benefit greatly from the low power consumption of cameras in long-term deployments.

Visual SLAM systems can be categorized into *feature-based* methods and *direct* methods. The feature-based methods [7], [8] detect and match features across frames, and then estimate relative camera motion by minimizing the reprojection error; whereas direct methods [9], [10] estimate camera motion by minimizing photometric error directly without feature correspondences. The direct methods demonstrate higher accuracy and robustness over feature-based methods, especially in poorly-textured (less-textured or repetitively-textured) environments [11]. As the feature detection and matching algorithms are computationally expensive, sparse direct methods also have the potential to run much faster (*e.g.*, $\geq$ 300 FPS for SVO [11]). On the other hand, visual SLAM systems can also be categorized into monocular systems and multi-camera systems. Monocular systems [7]–[10] cannot estimate the metric *scale* of the environment which multi-camera systems are able to. Multi-camera systems usually achieve higher accuracy and robustness; among these, stereo systems [12]–[14] are particularly popular for their simplicity and accessibility.

Most existing stereo visual systems use the standard *stereo matching* algorithm [15] to solve the scale problem, which has two major shortcomings. First, finding stereo correspondences by individually searching along the respective epipolar lines is computationally expensive. Secondly, if multiple points look similar to the query point, it is challenging to pick the correct one; this happens when the texture is repetitive (*e.g.*, grass, sand). We address these two limitations in [16], where the 3D points in the monocular system are projected into the second camera and the scale problem is solved by minimizing the photometric error. We demonstrate that such direct *scale optimization* is computationally efficient and more robust to repetitive textures in the visual scene.

However, even with metric scale, the global camera pose inevitably deviates from the ground truth as the camera moves, because it is estimated by accumulating the relative camera motions incrementally. The *loop closure* brings non-local pose constraints to optimize poses globally to address this issue. The conventional bag-of-word (BoW) approach detects loop closures by matching features from the current view to the history. However, the BoW approach does not work out-of-the-box for direct SLAM systems since direct SLAM systems do not extract descriptors for features. Alternatively, we propose a *LiDAR descriptor-based* place recognition method [17] for urban driving scenarios. We assume the vehicle is moving in the forward direction so that we can accumulate 3D points from

Manuscript received July 29, 2021; accepted November 20, 2021. Date of publication December 10, 2021; date of current version December 16, 2021. This letter was recommended for publication by Associate Editor Teresa A. Vidal-Calleja and Editor Javier Civera upon evaluation of the reviewers' comments. This work was supported in part by the US National Science Foundation under Award IIS 1637875, in part by the University of Minnesota Doctoral Dissertation Fellowship, and in part by the MnRI Seed Grant. *(Corresponding author: Jiawei Mo.)*

Jiawei Mo and Junaed Sattar are with the Department of Computer Science and Engineering, Minnesota Robotics Institute (MnRI), University of Minnesota Twin Cities, Minneapolis, MN 55455 USA (e-mail: moxxx066@umn.edu; junaed@umn.edu).

Md Jahidul Islam was with the Department of Computer Science and Engineering, Minnesota Robotics Institute (MnRI), University of Minnesota Twin Cities, Minneapolis, MN 55455 USA (e-mail: jahid@ece.ufl.edu).

Digital Object Identifier 10.1109/LRA.2021.3133860

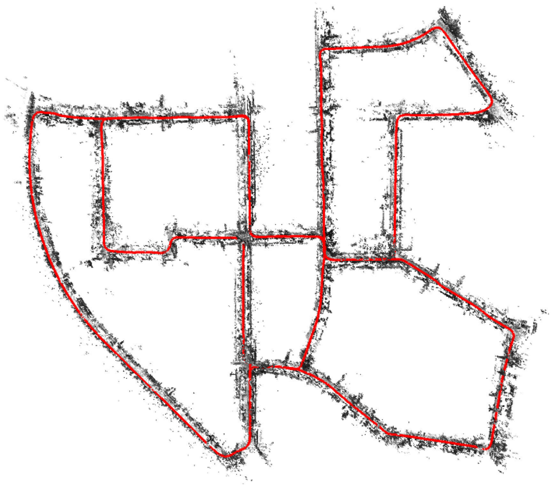

Fig. 1. The estimated trajectory and reconstructed environment by the proposed method on KITTI sequence 00.

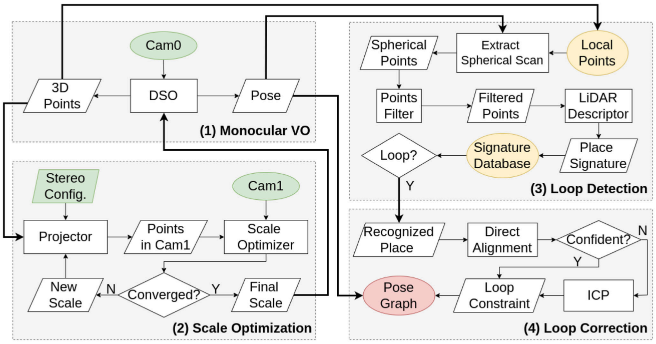

Fig. 2. An overview of DSV-SLAM: (1) starting from *Cam0*, the *Monocular VO* estimates camera poses and generates 3D points; (2) using the 3D points, *Scale Optimization* module estimates and maintains the scale of the VO; (3) *Loop Detection* module detects loop closures based on 3D points from VO; (4) for potential loop closures, *Loop Correction* module estimates the relative poses of loop closures and optimizes the poses globally.

the stereo direct SLAM system to imitate LiDAR scans, which are described by a LiDAR descriptor for place recognition. This facilitates significantly more efficient loop closure detection and ensures higher accuracy and robustness.

In this paper, we systematically combine scale optimization and the LiDAR descriptor-based place recognition approach into a fully-direct stereo SLAM system termed DSV-SLAM; we release an open-source implementation at https://github.com/IRVLab/direct_stereo_slam. We conduct thorough experiments to validate its state-of-the-art accuracy, superior computational efficiency, and robustness in visually challenging scenarios. DSV-SLAM demonstrates the feasibility of a full SLAM system without feature detection or matching. In DSV-SLAM, we adopt the state-of-the-art Direct Sparse Odometry (DSO) [10] to track camera poses and estimate 3D points. Then, we extend this to an efficient and accurate stereo visual odometry (VO) using scale optimization [16]. Subsequently, we use the LiDAR descriptor-based place recognition approach [17] to efficiently detect loop closures. The relative poses of potential loop closures are estimated by direct alignment and optionally, further refined by the Iterative Closest Point (ICP) method [18]. Finally, we compose and optimize a pose graph to further improve the SLAM accuracy globally. Fig. 1 shows the estimated trajectory and reconstructed environment by DSV-SLAM on sequence 00 of the KITTI dataset [19].

II. Related Work

Visual SLAM has been an active research problem in the robotics and computer vision literature over the last two decades. The early approaches relied on various filter-based estimation methods such as EKF-SLAM [20] and MSCKF [21]. Starting from PTAM [7], many popular approaches incorporate techniques borrowed from structure-from-motion [15] (*e.g.*, bundle adjustment) into optimization-based visual SLAM systems. The optimization-based visual SLAM systems can be categorized as either feature-based or direct methods depend on whether feature matching is used.

ORB-SLAM [8], [12], [22] is one of the most influential and established feature-based methods. In its stereo version [12], 3D points are triangulated from stereo matching and then tracked across frames. Subsequently, bundle adjustment is applied to jointly optimize the points and camera poses within a local sliding window by minimizing the reprojection error. In the back end, BoW is used for loop closure detection and relative pose estimation. Subsequently, an *essential graph* is optimized to improve the global accuracy. Global bundle adjustment is also executed to further improve the accuracy. Despite the gain in accuracy, it is computationally expensive.

DSO [10], [14], [23] is the current state-of-the-art for direct visual odometry. Wang *et al.* [14] extend DSO to a stereo system using stereo matching for depth initialization. To incorporate BoW into the DSO system for loop closure, Gao *et al.* [23] modify DSO's point selection strategy to flavor *trackable features* and compute descriptors for these features. However, stereo matching and feature detection and description are computationally expensive and lack robustness to poorly-textured environments.

As discussed in Sec. I, we proposed scale optimization [16] and LiDAR descriptor-based place recognition [17] as alternatives for stereo matching and BoW approach. They enable a fast and fully direct visual SLAM system, which we attempt to address in this paper.

III. Methodology

Fig. 2 illustrates an outline of the proposed system. There are four computational components: the monocular VO, the scale optimization module, the loop detection module, and the loop correction module.

III. Notations

We use ${}^{b}_{a}\mathbf{T} = [{}^{b}_{a}\mathbf{R}, {}^{b}_{a}\mathbf{t}] \in SE(3)$ to represent the **T**ransformation (**R**otation and **t**ranslation) from coordinate

a to coordinate b. We mark the stereo camera pair as Cam_0 and Cam_1. For Cam_k where $k \in \{0,1\}$, the corresponding image is I_k and the camera projection is denoted as Π_k. A 3D point is represented by $\Pi_0^{-1}(\mathbf{p}, d_\mathbf{p})$, where $\mathbf{p}$ and $d_\mathbf{p}$ are the pixel coordinates and (inverse) depth, respectively, which are back-projected into 3D space by Π_0^{-1}.

A. Monocular VO

As mentioned, we choose a direct method over feature-based methods for its accuracy, computational efficiency, and robustness in poorly-textured environments. The current state-of-the-art direct VO method is DSO [10], which works by minimizing the photometric error defined over a sliding window $\mathcal{F}$ of keyframes and points as

$$E = \sum_{i\in\mathcal{F}} \sum_{\mathbf{p}\in\mathcal{P}_i} \sum_{j\in obs(\mathbf{p})} E_{\mathbf{p}j} \tag{1}$$

$$E_{\mathbf{p}j} = \sum_{\mathbf{p}\in\mathcal{N}_\mathbf{p}} w_\mathbf{p} ||(I_0[\mathbf{p}'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}}(I_0[\mathbf{p}] - b_i)||_\gamma, \tag{2}$$

$$\mathbf{p}' = \Pi_0({}^j_i\mathbf{T}\Pi_0^{-1}(\mathbf{p}, d_\mathbf{p})). \tag{3}$$

That is, for each point $\mathbf{p} \in \mathcal{P}_i$ in keyframe $i \in \mathcal{F}$, if it is observed by keyframe j, then $E_{\mathbf{p}j}$ denotes the associated photometric error. $E_{\mathbf{p}j}$ defined in Eq. 2 is essentially the pixel intensity difference between a point $\mathbf{p}$ in keyframe i and its projection $\mathbf{p}'$ in keyframe j as defined in Eq. 3; the affine brightness terms ($a_{i/j}$, $b_{i/j}$), exposure times $t_{i/j}$, pixel pattern $\mathcal{N}_p$, weight w_p, and Huber norm $||\cdot||_\gamma$ are included for photometric robustness. Please refer to [10] for more details. It is worth mentioning that any monocular VO (preferably direct VO) method can be used here instead of DSO due to our modular system design.

B. Scale Optimization

As DSO is monocular VO, the scale is unobservable and tends to drift as time goes on. Stereo VO systems solve this problem by bringing the *metric distance* between cameras into the odometry system. As discussed, stereo matching is the conventional way to extend monocular VO to stereo VO but it is computationally expensive and it does not fit well into direct VO. Hence, we adopt scale optimization [16] in the proposed system to balance robustness and efficiency.

The main idea of scale optimization is to project the points from monocular VO on Cam_0 to Cam_1 and find the optimal scale that minimizes the photometric error defined as:

$$E = \sum_{\mathbf{p}\in\mathcal{P}} w_\mathbf{p} ||I_1[\mathbf{p}'] - I_0[\mathbf{p}]||_\gamma, \tag{4}$$

$$\mathbf{p}' = \Pi_1({}^1_0\mathbf{R} \cdot s\Pi_0^{-1}(\mathbf{p}, d_\mathbf{p}) + {}^1_0\mathbf{t}). \tag{5}$$

For each 3D point $\Pi_0^{-1}(\mathbf{p}, d_\mathbf{p})$, it is re-scaled in Cam_0 frame by current scale s and then projected into Cam_1 by $[{}^1_0\mathbf{R}, {}^1_0\mathbf{t}]$ and Π_1, which are known from the stereo calibration. The photometric error E in Eq. 4 is then defined as the pixel intensity difference between the original point $\mathbf{p}$ in I_0 and its projection $\mathbf{p}'$ in I_1. An example of such scale optimization is illustrated in Fig. 3. Eq 4 is an analogous formulation of Eq. 2 with two simplifications. First, there is no affine brightness parameters or exposure times. It is feasible as validated in the experiments of [16] because a stereo camera is usually hardware synchronized and triggered. Secondly, the photometric error is calculated using a single pixel instead of all pixels in a pattern $\mathcal{N}_p$ (as in Eq. 2) since the points remain fixed here. Consequently, the scale s is the only *free* parameter to optimize. These simplifications facilitate a computationally efficient optimization process.

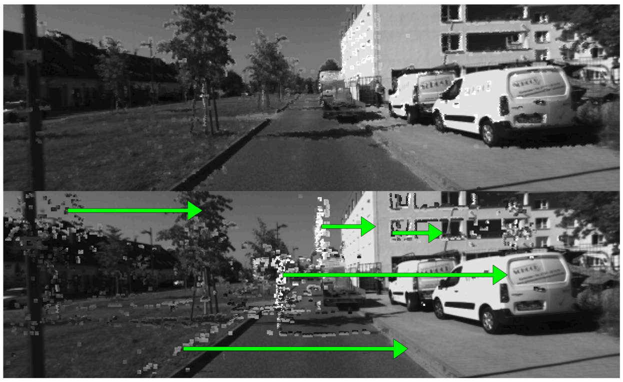

Fig. 3. An example of scale optimization on sequence 06 of the KITTI dataset. The top image is the projection with the optimal scale, where the projection well overlaps with the image; the bottom image is the projection with an incorrect scale ($0.1\times$optimal scale), the green arrows indicate the locations of the correct projections.

Since we do not have prior knowledge of the scale when the system starts, we run scale optimization with a series of initial guesses ranging from 0.1 to 50 (empirically chosen) to initialize the scale. Following scale optimization, DSO is adjusted correspondingly by re-scaling the *Pose* and *3D points*. For the consistency of DSO, we only re-scale the pose of the most recently created keyframe and reset its evaluation point; we do not re-scale the other keyframes because of the First Estimate Jacobians [24], [25], but their scale will be optimized heuristically. As a result, the metric scale of DSO is estimated and maintained by scale optimization alone. The resulting stereo VO is computationally efficient and remains fully direct requiring no feature extraction or matching.

C. Loop Detection

For VO, camera pose drift is inevitable because it is estimated by accumulating local camera motions. To compensate for this error, loop closure brings non-local pose constraints for global pose optimization. BoW [26], [27] is the conventional loop closure approach, but it does not fit well into direct methods for reasons discussed previously.

We propose an alternative approach in [17] that fits naturally into direct SLAM. Instead of 2D features, we focus on the 3D structure for place recognition. We adapt LiDAR descriptors on the 3D points from stereo VO to describe a place. However, the 3D points from VO are distributed in a *frustum* due to the narrow field-of-view of cameras. The pose of the frustum changes with that of the camera, which is not desired for place recognition.

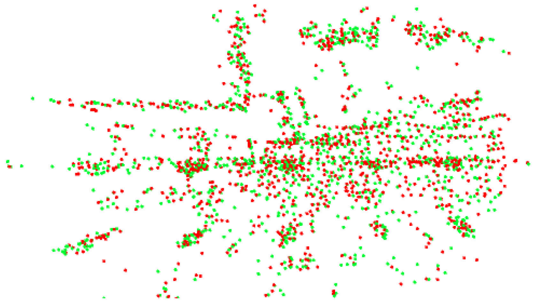

Fig. 5. When direct alignment fails, ICP finds the optimal pose that aligns the imitated LiDAR scans of the recognized place (red) and the current place (green).

Our solution to this is illustrated in Fig. 2(3); with the assumption that camera motion is predominantly in the forward direction, we propose to locally accumulate 3D points from VO to get a set of *Local Points*, then generate a set of *Spherical Points* around the current *Pose* to imitate a LiDAR scan. This is feasible because VO is locally accurate. For efficiency, we use *Points Filter* to remove redundant points. The *Filtered Points* make up the final imitated LiDAR scan (*e.g.*, Fig. 5). To describe the imitated LiDAR scan, we prefer global LiDAR descriptors over local ones mainly for two reasons. First, generating and matching global LiDAR descriptors is usually faster than local ones. Secondly, the imitated LiDAR scan is neither as consistent nor dense as a real LiDAR scan, which is not ideal for local LiDAR descriptors. We are able to use global LiDAR descriptors because the 3D points generated by the proposed stereo VO (DSO with scale optimization) have a metric scale. In [17], we validated that Scan Context [28] is accurate and efficient for datasets recorded in urban areas. Hence, we use Scan Context as our LiDAR descriptor and focus on urban driving scenarios.

The main idea of Scan Context is to use height distribution in urban areas (*e.g.*, buildings) to describe the point cloud generated by a LiDAR. The original Scan Context aligns the point cloud with respect to the gravity axis measured by IMU. Since we do not wish to bring additional sensors (*i.e.*, IMU) to our visual SLAM system, we align the point cloud using PCA [29] instead. After alignment, the horizontal plane (the most significant PCA plane in our case) is divided into multiple bins based on *radius* and *azimuth*. The maximal height in each bin is concatenated to form a signature for the current place. The authors of Scan Context also propose to use ring-key [28] for fast preliminary search before Scan Context, which encodes the occupancy ratio in each ring determined by radius. An illustration is given in Fig. 4.

In our system, for each keyframe from the stereo VO, we imitate a LiDAR scan by the proposed method and generate its place signature using our modified Scan Context descriptor. Then we search for potential loop closure in the *Signature Database*. We first search by ring-key, which is fast but less discriminative, so we select the top three place candidates for Scan Context to make the final decision.

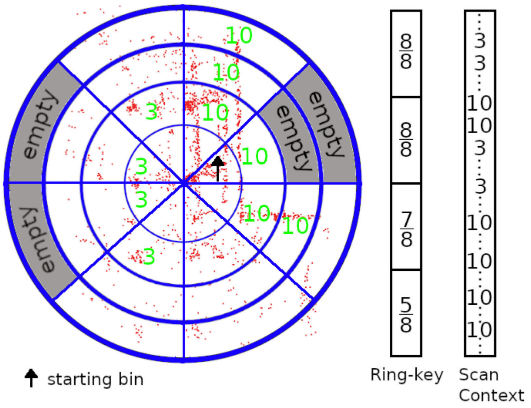

Fig. 4. A simplified illustration of ring-key and Scan Context descriptor on the imitated LiDAR scan near the place in Fig. 3. We assume the heights of buildings and trees are 10 meters and 3 meters, respectively (for this illustration only).

D. Relative Pose Estimation

As Fig. 2(4) shows, for each *Recognized Place*, we try to estimate a *Loop Constraint* (*i.e.*, the relative pose) between the current place and recognized place. This is achieved by *direct alignment* as done in DSO tracking based on the following equations:

$$E = \sum_{\mathbf{p}\in\mathcal{P}} w_{\mathbf{p}} ||(I_0^c[p'] - b_c) - \frac{t_c e^{a_c}}{t_r e^{a_r}}(I_0^r[\mathbf{p}] - b_r)||_\gamma, \tag{6}$$

$$p' = \Pi_0({}^c_r\mathbf{T}\Pi_0^{-1}(\mathbf{p}, d_{\mathbf{p}})). \tag{7}$$

Here, I_0^c and I_0^r are the **c**urrent and **r**ecognized frames, respectively. We are estimating ${}^c_r\mathbf{T}$, the relative pose from recognized frame to current frame, which is initialized by the PCA alignment in *Loop Detection*. The other variables are same as the ones in Eqs. 2 and 3. We specifically project points from the recognized frame to the current frame for memory efficiency, because we only need to store the sparse points instead of the entire image for the recognized frame.

Although Eqs. 6 and 7 look similar to the error terms in DSO (*i.e.*, Eqs. 1-3), there are only two keyframes (*i.e.*, recognized frame and current frame) in this optimization rather than a sliding window in DSO, and hence, fewer points and constraints; additionally, the illumination, occlusion, and even the scene can vary drastically for loop closures. Consequently, direct alignment alone is not robust enough for loop closure. For robustness, we execute ICP [18] to align the imitated LiDAR scans when direct alignment is not confident (Eqs. 6-7 converge to a large photometric error). An example of ICP is shown in Fig. 5. ICP is particularly robust when the visual appearance changes drastically. Although it is computationally more expensive than direct alignment, the initial relative pose from PCA in *Loop Detection* is reasonably accurate and facilitates a fast convergence. Alternatively, the pose can be estimated by direct alignment and ICP jointly [30] for improved accuracy and robustness.

Finally, the *Pose Graph* composed of the consecutive keyframes and loop closures is optimized to improve the pose accuracy globally. Although not implemented yet, global bundle adjustment can be done using the 3D points association from either direct alignment or ICP algorithm for improved map consistency.

IV. Experimental Evaluation

To evaluate the accuracy and computational efficiency of DSV-SLAM system, we include several variants of DSO for internal comparison. In particular, we compare the scale optimization in DSV-SLAM with the stereo matching approach adopted in the Stereo DSO[1] [14]. We also compare the performance of our LiDAR descriptor-based place recognition module to the conventional BoW approach used in LDSO [23]. Externally, we include the performance evaluations of stereo ORB-SLAM2 [12] for accuracy and efficiency comparison. Since the Scan Context used in this system is designed for the urban driving scenario, we mainly focus on two publicly available datasets: the KITTI visual odometry dataset [19] and the Malaga dataset [31]. Our experiments are conducted on an Intel$^{\text{TM}}$ i7-8750H platform having a 2.2 GHz CPU with six cores and 16 GB RAM. We use the *default* setting of DSO with 2000 points located in 5-7 keyframes in the sliding window for optimization (*i.e.*, in Eq. 1-3). Moreover, when imitating a LiDAR scan for loop detection, we set the LiDAR range (*i.e.*, the radius of *Spherical Points* in Fig. 2(3)) to 40 meters. In the current implementation, scale optimization runs sequentially in the main DSO thread while the loop closure parts (detection, estimation, and pose optimization) run in a separate thread. Due to the inherent randomness in DSO and ORB-SLAM2, we run each algorithm five times and compute the average when calculating accuracy and efficiency.

A. Evaluation on the KITTI Dataset

The KITTI dataset contains 22 sequences of stereo images. The ground truths for the first 11 sequences are publicly available; while the ground truths for the rest are reserved for ranking VO algorithms. We focus on the first 11 sequences for complete evaluations.

1) Accuracy: To compute accuracy, we align the estimated trajectory to the ground truth and compute the root mean square error of the trajectory as the absolute trajectory error (ATE). Since the DSO and LDSO are monocular systems and unaware of scale, the alignment is based on $Sim3$; Stereo DSO, (stereo) ORB-SLAM2, and DSV-SLAM are aligned with $SE3$. Since the pose graph consists of keyframes only, the comparisons are based on keyframes.

Table I reports the accuracy of the state-of-the-art visual SLAM systems on the KITTI dataset. Our results for LDSO and ORB-SLAM2 agree with the results reported in [23] and [12], respectively. The ATEs of Stereo DSO are calculated with the trajectories provided by [14] (they do not provide code); we also report the results using the 3 rd party implementation in parenthesis.

TABLE I
Comparisons for Accuracy Based on Absolute Trajectory Error (ATE) in *Meters* on the KITTI Dataset. Results With Loop Closures Are Marked With Asterisk (*). For Stereo DSO, the Results Are 'Official Results (3 rd Implementation)'; for DSV-SLAM, the Results Are 'Loop Enabled (No Loop)'

Seq.	DSO ($Sim3$)	LDSO ($Sim3$)	Stereo DSO	DSV-SLAM	ORB-SLAM2
00	102.21	6.51*	3.73 (4.19)	2.59* (5.27)	**1.19***
01	122.67	7.32	**4.07** (5.64)	5.73	9.59
02	101.48	15.08*	6.99 (4.56)	**4.18*** (5.61)	5.35*
03	1.94	2.07	1.22 (1.16)	2.47	**0.64**
04	0.82	0.95	0.83 (0.82)	1.09	**0.19**
05	46.63	4.48*	1.99 (2.36)	3.83* (3.92)	**0.72***
06	54.21	11.64*	1.78 (23.57)	0.80* (1.04)	**0.75***
07	14.60	13.60*	1.15 (2.97)	4.37	**0.48***
08	95.83	100.02	**2.70** (3.26)	4.77	3.23
09	58.31	60.37	3.63 (2.95)	5.32	**2.85**
10	10.94	13.71	**0.77** (1.04)	1.78	1.05

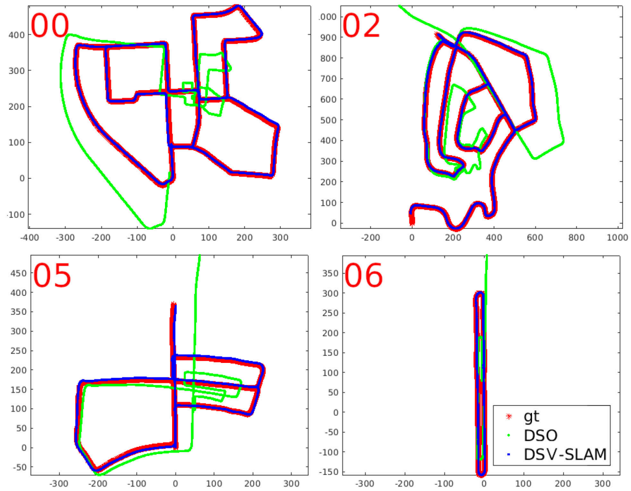

Fig. 6. The trajectories estimated by DSO (green), DSV-SLAM (blue), and the ground truth (red) on KITTI sequence 00, 02, 05, and 06. With scale optimization and loop closure, the improvement of DSV-SLAM over DSO is significant.

With DSO being monocular VO, its ATEs are large due to the drifting scale, especially on long sequences like 00, 02, and 08. For LDSO, the ATEs decrease drastically compared to DSO on sequences with loop closures (*i.e.*, 00, 02, 05, 06, and 07). The scale drifting problem is solved by all the stereo systems. Overall, ORB-SLAM2 performs the best on KITTI dataset, possibly due to the maturity of feature-based methods and the comprehensive system design (*e.g.*, global bundle adjustment). For Stereo DSO and DSV-SLAM, although the results on some sequences (*e.g.*, 04) are not as good as ORB-SLAM2, they achieve competitive accuracy on more than half of the sequences. The fast vehicle movement with low camera frame-rate (10 Hz) in KITTI dataset are not ideal for direct methods (*i.e.*, DSO).

As the results suggest, the accuracy of DSV-SLAM is comparable to the state-of-the-art visual SLAM systems. With loop closure, the accuracy of DSV-SLAM is further improved on sequences 00, 02, 05, and 06. Fig. 6 shows the trajectories estimated by DSV-SLAM. Since our stereo VO with scale optimization is already very accurate, the improvement with

[1] Since no official release of Stereo DSO is available, we use a third-party implementation in *https://github.com/JingeTu/StereoDSO*

TABLE II

COMPARISONS FOR RUN-TIME (*MEAN* × *EXECUTION COUNT*) ON THE KITTI DATASET. [S.DSO: STEREO DSO; O.-SLAM2: ORB-SLAM2; SM: STEREO MATCHING; SO: SCALE OPTIMIZATION; SC: SCAN CONTEXT; RK: RING-KEY; D: DIRECT ALIGNMENT; I: ICP]

Time(ms)	Sequence 06 (short)					Sequence 00 (comprehensive)				
Methods	DSO	LDSO	S.DSO	DSV-SLAM	O.-SLAM2	DSO	LDSO	S.DSO	DSV-SLAM	O.-SLAM2
Point Sel.	4.45	7.16	**4.35**	4.43	22.8	**4.72**	8.07	**4.72**	4.77	21.2
SM/SO	-	-	10.5	**2.09**	17.6	-	-	10.8	**2.19**	18.7
BoW/SC	-	1.71×828	-	$\mathbf{0.46} \times 625$	7.44×506	-	1.89×3461	-	$\mathbf{0.49} \times 2321$	7.37×1380
Loop Det.	-	0.33×828	-	$\mathbf{0.05} \times 625^{RK}$ $\mathbf{0.01} \times 370^{SC}$	6.95×504	-	1.74×3461	-	$\mathbf{0.06} \times 2321^{RK}$ $\mathbf{0.01} \times 1915^{SC}$	8.70×1378
Loop Est.	-	$\mathbf{0.44} \times 453$	-	0.68×50^{D} 8.25×14^{I}	0.77×258	-	$\mathbf{0.81} \times 2058$	-	0.96×175^{D} 8.10×100^{I}	1.19×367
Pose Opt.	-	200×28	-	$\mathbf{35.9} \times 43.4$	589×1	-	1796×34	-	$\mathbf{125} \times 110$	917×4
Full BA	-	-	-	-	9600×1	-	-	-	-	4793×4
Loop #	-	37.0	-	$35.4^{D} + 8.0^{I}$	1.0	-	277.6	-	$75^{D} + 35^{I}$	4.0

loop closure is not as drastic as LDSO over DSO. However, unlike LDSO and ORB-SLAM2, DSV-SLAM does not capture the loop closure in sequence 07. This is because the overlapped trajectory is too short to accumulate *Local Points* and imitate LiDAR scans for place recognition, whereas BoW works on a single frame.

2) Efficiency: We investigate the efficiency of each computational component and report the results of a short sequence (06) and a comprehensive sequence (00) in Table II.

To enable BoW, the *point selection* in LDSO is tuned to prefer features for cross-frame matching, and then a descriptor is extracted for each feature. Consequently, the time spent on point selection is increased compared to DSO. However, the point selection in Stereo DSO and DSV-SLAM is as fast as in DSO. We find that scale optimization (SO) in DSV-SLAM is much faster than the stereo matching (SM) in Stereo DSO and ORB-SLAM2. The stereo matching in ORB-SLAM2 is based on feature descriptors and it is the slowest. On the contrary, in Stereo DSO, the points are projected to the stereo frame and the correspondences are searched around that projection, which is potentially the reason for its faster performance. Nevertheless, scale optimization offers the fastest run-time.

For loop closure, generating BoW in LDSO is slower than generating the Scan Context (SC) descriptor in DSV-SLAM. Detecting loop closure using BoW is also slower than using the hierarchical search method (*i.e.*, ring-key and Scan Context descriptor) in DSV-SLAM. For loop pose estimation, direct alignment in DSV-SLAM is marginally slower than the PnP method [15] used in LDSO. Although the ICP in DSV-SLAM is much slower, it is executed only when the direct alignment is not confident, which happens less frequently for simple tests (06). Moreover, the ratio of accepted / proposed loop closures in DSV-SLAM ($\frac{43.4}{50}$ and $\frac{110}{175}$) is much higher than in LDSO ($\frac{37}{453}$ and $\frac{277.6}{2058}$). This indicates that our LiDAR descriptor-based place recognition approach in DSV-SLAM achieves higher precision over the BoW approach (refer to [17] for more detailed validation). Consequently, the time saved by DSV-SLAM on point selection and loop detection is more significant than the loss on loop pose estimation.

Besides, LDSO spends more time on loop pose optimization; other than consecutive keyframes and loop closures, LDSO also brings the connection between each keyframe and the very first keyframe to the pose graph for accuracy and robustness. Lastly, the loop closure module of ORB-SLAM2 is much slower overall due to its complex mechanisms to improve accuracy and robustness. For instance, ORB-SLAM2 searches the lowest score in its covisibility graph and compares it to the candidate score for loop detection; a loop candidate is accepted only when three consistent and consecutive loop candidates are found in the *covisibility* graph. Such conservative approaches incur considerable computational overhead.

B. Evaluation on the Malaga Dataset

To further validate the proposed DSV-SLAM system, we evaluate its performance on the Malaga dataset [31]. It is more challenging than the KITTI dataset because it consists of various test cases with adverse visual conditions. A few challenging scenarios with poor visibility and direct sunlight are shown in Fig. 7. In the evaluation, we focus on sequences with loop closures (*i.e.*, sequence 05, 06, 07, 08, and 10) for testing. Since there is only GPS data available as ground truth, rather than conducting quantitative analyses, we show a qualitative performance comparison in Fig. 7. Our observations from the experimental results are listed below:

- Overall, the scales of the trajectories by both DSV-SLAM and ORB-SLAM2 are slightly inaccurate. We suspect the potential reason is that the structures might be too far for the short-baseline (12 cm) stereo camera used in the Malaga dataset.
- In sequence 05, the DSO tracking in DSV-SLAM struggles at the roundabout (see the red rectangle in Fig. 7) due to direct sunlight. Scale optimization also fails several times. However, the shape of DSV-SLAM's trajectory is still more accurate than ORB-SLAM2.
- In sequence 06, the DSO tracking in DSV-SLAM also fails due to traffic and pedestrians when the vehicle stops for about 40 seconds (see the blue rectangle in Fig. 7). It takes a few seconds to recover tracking, which leads to an inconsistent trajectory estimation by DSV-SLAM without loop closure (denoted by the green trajectory). However, loop closure finds the failure point and eventually corrects the trajectory. ORB-SLAM2 is slightly better with a more accurate scale.
- In sequence 07, the orientation of the trajectory estimated by ORB-SLAM2 is slightly off, whereas the scale of DSV-SLAM is slightly off.

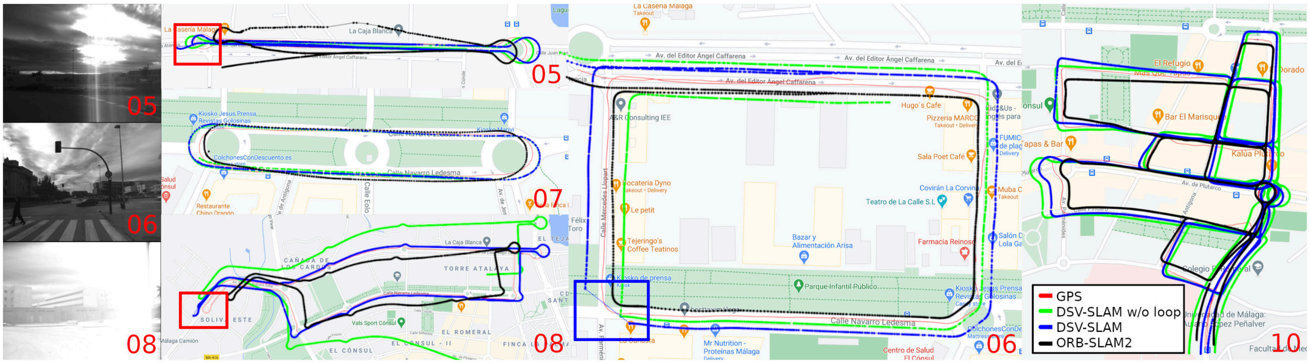

Fig. 7. Results on Malaga dataset. The blue rectangle in the sequence 06 shows where the vehicle stops for about 40 seconds and the underlying DSO in DSV-SLAM loses tracking due to traffic and pedestrians. DSO tracking is also lost in the red rectangles in the sequence 05 and 08 due to direct sunlight. Nevertheless, loop closure significantly improves the accuracy of DSV-SLAM in these challenging scenarios.

TABLE III
COMPARISONS FOR RUN-TIME (*MEAN MILLISECOND* × *EXECUTION COUNT*) ON SEQUENCE 10 OF THE MALAGA DATASET. [SM: STEREO MATCHING; SO: SCALE OPTIMIZATION; SC: SCAN CONTEXT; RK: RING-KEY; D: DIRECT ALIGNMENT; I: ICP]

	DSV-SLAM	ORB-SLAM2
Select Point	$\mathbf{7.50} \times 5053$	28.7×17310
SM/SO	$\mathbf{4.06} \times 5050$	24.6×17310
BoW/SC Gen.	$\mathbf{0.64} \times 3890$	7.76×3072
Loop Det.	$\mathbf{0.07} \times 3890^{RK} + \mathbf{0.04} \times 3369^{SC}$	15.3×3070
Loop Est.	$1.91 \times 854^{D} + 11.98 \times 755^{I}$	$\mathbf{2.09} \times 1129$
Pose Opt.	$\mathbf{164.2} \times 215$	4202×11
Full BA	-	90578×11
FPS	$\mathbf{27.5}$	14.0
Loop Count	$99^{D} + 116.4^{I}$	11.6

- In sequence 08, the DSO tracking in DSV-SLAM fails at the red rectangle due to the sudden brightness change. Consequently, the trajectory of DSV-SLAM without loop closure is off; nevertheless, it can re-localize itself with loop closure when the vehicle comes back to the start location. For ORB-SLAM2, the scale of its trajectory is noticeably smaller than the ground truth.
- Lastly, sequence 10 is a long run with various straights and turns as well as loop closures, which tests visual SLAM algorithms comprehensively. The trajectory generated by DSV-SLAM is slightly more accurate than ORB-SLAM2. We also notice that the distance between the start and the end of the trajectory is considerably reduced by the loop closure (from the green trajectory to the blue one).

Overall, we find that DSV-SLAM's accuracy is comparable and often better than ORB-SLAM2 on the Malaga dataset. However, DSV-SLAM is computationally more efficient with significant margins as presented in Table III. ICP is executed more frequently on the Malaga dataset than on the KITTI dataset since direct alignment is vulnerable to brightness change.

C. Evaluation on the RobotCar Dataset

RobotCar dataset [32] is recorded in different seasons throughout the year, which we used to validate the robustness of the LiDAR descriptor-based place recognition approach against visual appearance changes in [17]. Snapshots are given in Fig. 8. We demonstrate the preliminary result of DSV-SLAM on the RobotCar dataset in Fig. 9, where we first play the sequence '2015-05-19-14-06-38' (*run1*), and then we "kidnap" the robot to sequence '2015-08-13-16-02-58'(*run2*). As Fig. 9 shows, the DSO scale gets larger throughout; the drifting scale is fixed with scale optimization (see the green trajectory); with loop closures, the robot eventually re-localize itself and bring the two runs together (see the blue trajectory). We also run ORB-SLAM2 using the same setting; however, its tracking fails consistently.

Fig. 8. Snapshots of the RobotCar dataset. There are many visual appearance differences including trees and foliage, traffic, pedestrians, and varying brightness.

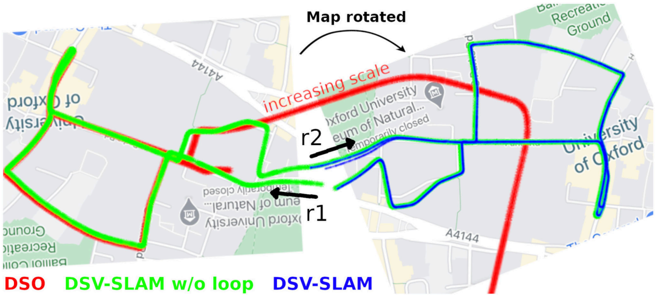

Fig. 9. The trajectories estimated by DSO, DSV-SLAM with and without loop closure on RobotCar dataset. While DSO has an increasing scale issue, DSV-SLAM estimates the scale accurately and consistently, and it is able to re-localize the robot after being kidnapped using loop closures. [r1: start of run1; r2: start of run2 where the robot is kidnapped to].

V. Conclusion

In this paper, we propose the first fully-direct visual SLAM system for driving scenarios, demonstrating the feasibility of a full SLAM system without feature detection or matching. We first extend the monocular DSO to a stereo system using scale optimization; then we integrate a LiDAR descriptor-based place recognition approach to detect loop closures; for potential loop closures, we use direct alignment to estimate the relative pose, which is bolstered by ICP when the direct alignment is not confident. Validation on public datasets demonstrates that the proposed system achieves considerably better computational efficiency while offering comparable accuracy and improved robustness in challenging scenarios. For future work, we will look into eliminating the forward-moving camera assumption when imitating LiDAR scans to expand our potential use cases. We also intend to extend the system into a stereo-visual-inertial system by integrating IMU measurements to further improve robustness.

References

[1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[2] S. Thrun, "Simultaneous localization and mapping," in *Robot. Cogn. Approaches Spatial Mapping*, Springer, 2007, pp. 13–41.

[3] O. G. Grasa, E. Bernal, S. Casado, I. Gil, and J. Montiel, "Visual SLAM for handheld monocular endoscope," *IEEE Trans. Med. Imag.*, vol. 33, no. 1, pp. 135–146, Jan. 2014.

[4] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[5] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, 2011.

[6] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 55–81, 2015.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Computer Society, 2007, pp. 1–10.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[9] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 834–849.

[10] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[11] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[12] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[13] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1935–1942.

[14] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3903–3911.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[16] J. Mo and J. Sattar, "Extending monocular visual odometry to stereo camera systems by scale optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6921–6927.

[17] J. Mo and J. Sattar, "A fast and robust place recognition approach for stereo visual odometry using LiDAR descriptors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5893–5900.

[18] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms Data Struct.*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[20] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[21] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565–3572.

[22] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[23] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2198–2204.

[24] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A first-estimates jacobian EKF for improving SLAM consistency," in *Exp. Robot,*. Springer, 2009, pp. 373–382.

[25] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[26] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1470–1478.

[27] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[28] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4802–4809.

[29] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2010, pp. 356–369.

[30] C. Park, S. Kim, P. Moghadam, J. Guo, S. Sridharan, and C. Fookes, "Robust photogeometric localization over time for map-centric loop closure," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1768–1775, Apr. 2019.

[31] J. Blancoclaraco, F. Morenoduenas, and J. Gonzalezjimenez, "The málaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 207–214, 2014.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.