To appear in *Advanced Robotics*
Vol. 35, No. 5, March 2021, 1–14

FULL PAPER

Online Multi-Sensor Calibration Based on Moving Object Tracking

J. Peršić*, L. Petrović, I. Marković and I. Petrović

University of Zagreb, Faculty of Electrical Engineering and Computing,
Laboratory for Autonomous Systems and Mobile Robotics (LAMOR)

(*v1.0 released April 2020*)

Modern autonomous systems often fuse information from many different sensors to enhance their perception capabilities. For successful fusion, sensor calibration is necessary, while performing it online is crucial for long-term reliability. Contrary to currently common online approach of using ego-motion estimation, we propose an online calibration method based on detection and tracking of moving objects. Our motivation comes from the practical perspective that many perception sensors of an autonomous system are part of the pipeline for detection and tracking of moving objects. Thus, by using information already present in the system, our method provides resource inexpensive solution for the long-term reliability of the system. The method consists of a calibration-agnostic track to track association, computationally lightweight decalibration detection, and a graph-based rotation calibration. We tested the proposed method on a real-world dataset involving radar, lidar and camera sensors where it was able to detect decalibration after several seconds, while estimating rotation with 0.2° error from a 20 s long scenario.

Keywords: multi-sensor calibration, online calibration, moving object tracking, radar, lidar, camera

1. Introduction

Modern robotic systems such as autonomous vehicles (AV) usually operate in highly dynamic scenarios where the actions they take significantly impact the surrounding environment. In order to achieve autonomy, they have to reliably solve many complex tasks, such as environment perception, motion prediction, motion planning and control. Environment perception, as the first building block of the autonomy pipeline, provides input data for many complex components, such as simultaneous localization and mapping (SLAM), detection and tracking of moving objects (DATMO) and semantic scene understanding. To increase the accuracy and robustness of an autonomous system, environment perception is often based on fusion of information from multiple heterogeneous sensors, such as lidar, camera, radar, GNSS and IMU. Accurate sensor calibration is a prerequisite for successful sensor fusion.

The sensor calibration consists of finding the intrinsic, extrinsic and temporal parameters, i.e. parameters of individual sensor models, transformations between sensor coordinate frames and alignment of sensor clocks, respectively. There are numerous offline and online approaches to sensor calibration and they vary significantly based on the sensors involved. While the offline approaches rely on controlled environments or calibration targets to achieve accurate calibration, the online approaches use information from the environment during the regular system operation, thus enabling long term robustness of the autonomous system. In this paper, we focus on the online calibration methods which are applicable for lidar–camera–radar sensor systems.

*Corresponding author. Email: juraj.persic@fer.hr

The online calibration methods can be roughly divided into feature-based and motion-based methods. Feature-based methods rely on extracting informative structure from the environment to generate correspondences between the sensors. These methods are limited to camera–lidar calibration, since other existing sensors do not provide enough structural information. For instance, extrinsic camera–lidar calibration can be based on line features detected as intensity edges in the image and depth discontinuities in the point cloud [1, 2]. Alternatively, the intensity of signal returned by lidar was used in [3] to find extrinsic calibration by maximizing the mutual information between images from camera and projected intensity values measured by the lidar. Recently, Park et al. [4] proposed a method for extrinsic and temporal camera–lidar calibration based on 3D point features in the environment. When the sensors do not provide enough structural information (e.g. radar), online calibration can be solved by depending on either the ego-motion or motion of objects in the environment. The former come with the advantage that the sensors do not have to share a common field of view (FOV), while the latter also work with a static sensor systems. In [5] authors proposed an ego-motion based calibration suitable for camera–lidar calibration, while Kellner et al. [6] proposed a solution for radar odometry and alignment with the thrust axis of the vehicle. Furthermore, Kummerle et al. [7] proposed simultaneous calibration, localization and mapping framework which enables both extrinsic calibration and estimation of the robot kinematic parameters. Recently, Giamou et al. [8] proposed a solution for globally optimal ego-motion based calibration. Tracking-based methods have mostly been employed in static homogeneous sensor systems. To calibrate multiple stationary lidars, Quenzel et al. [9] relied on tracking of moving objects, while Glas et al. [10, 11] used human motion tracking. Human motion was also used for a stationary camera calibration [12, 13]. Considering tracking-based calibration of stationary heterogeneous sensors, Glas et al. [14] proposed a method for calibration of multiple 2D lidars and RGB-D cameras, while Schöller et al. [15] proposed a method for stationary camera–radar calibration.

Within the context of an AV, a sensor system consists of multiple lidars, cameras, radars and other sensors. While it is sufficient to use only a subset of sensors for accurate ego-motion estimation, DATMO is often performed using all the available exteroceptive sensors to provide a greater FOV coverage, robustness to adverse conditions and to increase the accuracy [16–18]. Several datasets have been recently made public by both the industry and the academia to emphasize importance and accelerate research on DATMO [17, 19–21]. In this paper, we leverage current state of the art in DATMO and propose an online calibration method based on it. Our motivation is to enable decalibration detection and recalibration based on the information which is already present in an autonomous system pipeline without adding significant computational overhead. To the best of the authors' knowledge, this is a first online calibration method that is based on heterogeneous sensor DATMO on a moving platform. In addition, while several target-based methods for calibration of radar–lidar–camera systems exists [22, 23], this is the first attempt to calibrate these sensors simultaneously in an online setting.

Our method provides a full pipeline which includes: (i) DATMO algorithm for each sensor modality, (ii) track-to-track association based on a calibration invariant measure, (iii) efficient decalibration detection and (iv) a graph-based calibration handling multiple heterogeneous sensors simultaneously. We point out that our method estimates only rotational component of the extrinsic calibration, because translation is unobservable due to limited sensor accuracy and a bias in detections (e.g. radar might measure a metal rear axle, while lidar detections report center of a bounding box. Even the methods based on ego-motion would struggle estimating translation on an AV, because they require motion which excites at least two rotational axis [24]. However, in contrast to rotational decalibration, feasible translational decalibration would not have a significant impact on the system performance. For instance, if rotational decalibration existed, object detection fusion would experience a growing error in the position with the increase in object distance, while translational decalibration would only introduce a position error of equal value. Our method assumes that translational calibration is obtained using either target-based or sensor-specific methods. The presented approach was evaluated on the nuScenes dataset [17],

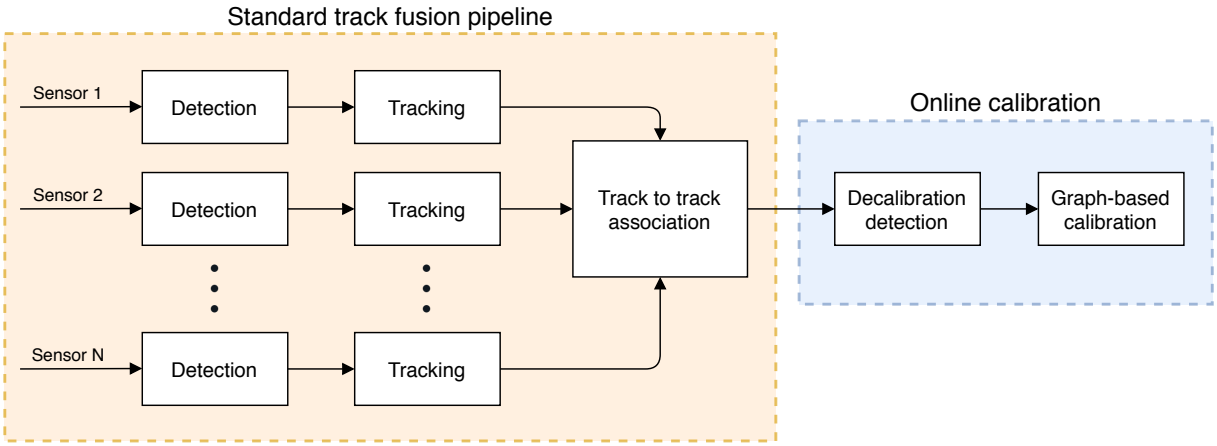


Figure 1. Illustration of the proposed pipeline for calibration based on DATMO. Detection, tracking and track to track association are commonly parts of a track fusion pipelines. However, our association criterion is oriented towards being calibration agnostic. Thereafter, we propose two new modules: decalibration detection and graph-based calibration.

but is not in any way limited to this specific sensor setup. However, for testing our method, it is currently the only dataset containing appropriate data from cameras, radars and a lidar, which are sensors in the focus of the proposed calibration method.

2. Proposed Method

In this section, we present each element of the method pipeline illustrated in Fig. 1. The pipeline starts with the object detection which is specific for each sensor. Afterwards, the detections are tracked with separate trackers for each sensor which slightly differ among sensor modalities to accommodate their specifics. The confirmed tracks of different sensors are then mutually associated using calibration invariant measures. Each aforementioned stage has built-in outlier filtering mechanisms to prevent degradation of the results of subsequent steps. With the associated tracks, we proceed to a computationally lightweight decalibration detection. Finally, if decalibration is detected, we proceed to the graph-based sensor calibration. The method handles asynchronous sensors by assuming temporal correspondence between sensor clocks is known and performing linear interpolation of the object positions. Throughout the paper, we use the following notation: world frame $\mathcal{F}_w$, ego-vehicle frame $\mathcal{F}_e$ and *i-th* sensor frame $\mathcal{F}_i$. For convenience, we choose one sensor to be aligned with $\mathcal{F}_e$. In the case of the nuScenes sensor setup, we chose the top lidar as it shares FOV segments with all the other sensors.

2.1 *Object detection*

The proposed pipeline starts with object detection performed for the each sensor individually. Automotive radars usually provide object detections obtained from proprietary algorithms performed locally on the sensor, while most can also provide tracked measurements. Obtaining the raw data is not possible due to low communication bandwidth of the CAN bus, typically used by these sensors. Nevertheless, radars provide a list of detected objects consisting of the following measured information: range, azimuth angle, range-rate, and radar cross-section (RCS). We use these detections and classify them as moving or stationary based on the range-rate. Furthermore, to avoid the need for extended target tracking where one target can generate multiple measurements, we perform clustering of close detections. These clusters are forwarded to the radar tracking module.

Contrary to the radar, lidar's and camera's raw data provides substantial information from which object detection is required. To extract detections from the lidar's point cloud, we used the

MEGVII network based on sparse 3D convolution proposed by Zhu et al. [25] which is currently the best performing method for object detection on the nuScenes challenge. The method works by accumulating 10 lidar sweeps into a single one to form a dense point cloud input, thus reducing the effective frame rate of the sensor by a factor of 10. As the output, the network provides 3D position of objects as well as their size, orientation, velocity, class and detection score. Finally, for the object detection from images, we rely on a state-of-the-art 3D object detection approach dubbed *CenterNet* [26]. The output of *CenterNet* is similar to the lidar detections output, except that the velocity information is not provided since detections are based on a single image. We used the network weights trained on the KITTI dataset and determined the range scale factor by comparing *CenterNet* detections to the *MEGVII* detections. At this stage, outlier filtering was based on the detection score threshold.

2.2 *Tracking of moving objects*

Tracking modules for individual sensors take detections from the previous step as inputs, associate them between different time frames and provide estimates of their states, which are later used as inputs for subsequent steps. Since tracking is sensor specific, we perform it in each respective coordinate frame $\mathcal{F}_i$. We adopt a similar single-hypothesis tracking strategy for all the sensors, following the nuScenes baseline approach [27]. Assigning detections to tracks is done by using a global nearest neighbor approach and the Hungarian algorithm which provides efficient assignment solution [28]. The assignment is tuned by setting a threshold which controls the likelihood of a detection being assigned to a track. The state estimation of individual tracks is provided by an Extended Kalman filter which uses a constant turn-rate and velocity motion model [29]. Thus, the state vector in the lidar and camera tracker is

$$\boldsymbol{x}_k = [x_k \; y_k \; z_k \; \dot{x}_k \; \dot{y}_k \; \dot{z}_k \; \omega_k]^T, \tag{1}$$

with the state transition defined as

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} 1 & 0 & 0 & \frac{\sin(\omega_k T)}{\omega_k} & -\frac{1-\cos(\omega_k T)}{\omega_k} & 0 & 0 \\ 0 & 1 & 0 & \frac{1-\cos(\omega_k T)}{\omega_k} & \frac{\sin(\omega_k T)}{\omega_k} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_k + \begin{pmatrix} \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} & 0 \\ T & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{w}, \tag{2}$$

where $\boldsymbol{w} = [w_x \; w_y \; w_z \; w_\omega]^T$ is white noise on acceleration and turn-rate, while T is sensor sampling time. Using object position measurements forms the measurement model defined as

$$\boldsymbol{y}_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}_k + \boldsymbol{v}, \tag{3}$$

where $\boldsymbol{v} = [v_x \; v_y \; v_z]^T$ is white measurement noise.

Due to the lack of radar's elevation angle measurement, we drop the position and velocity in the z-direction, thus reducing the state vector for the radar tracker to

$$\boldsymbol{x}_k = [x_k \; y_k \; \dot{x}_k \; \dot{y}_k \; \omega_k]^T \tag{4}$$

with adjusted state transition function and measurement model.

Track management is based on the track history, i.e. the track is confirmed after N_{birth} consecutive reliable detections and removed after N_{coast} missing detections. All parameters are tuned for each sensor separately as they have significantly different frame rates and accuracies. Lastly, subparts of individual tracks that exhibit sudden changes in velocity are marked as unreliable and these time instants are excluded from the subsequent steps.

2.3 *Track-to-track association*

Track to track association has been previously studied and a common approach is based on the history and distance of track positions [30]. Contrary to the traditional approaches, we do not assume a perfect calibration, as decalibration could degrade the association. Thus, we observe two criteria for each track pair candidates through their common history: (i) mean of the velocity norm difference and (ii) mean of the position norm difference. The track pair has to satisfy both criteria and not surpass predefined thresholds. If multiple associations are possible, none of them are associated. This conservative approach helps in eliminating wrong association which would compromise the following calibration steps. However, the remaining tracks can be associated with more common association metrics (e.g. Euclidean or Mahalanobis distance) and used within a track fusion module. In our method we can use such a conservative approach and discard some track associations, since our goal is sensor calibration and not safety critical online DATMO for vehicle navigation.

The position norm is not truly calibration independent, as it is affected by both the measurement bias in the individual sensor and the translation between the sensors. Thus we use it in a loose way solely to distinguish between clearly distant tracks, i.e. we rely on the previously calibrated translational parameters and use a high threshold. On the other hand, velocity norm has already been used in a stationary system calibration for track association [14] as well as for frame-invariant temporal calibration of the sensors [31]. In a stationary scenario, it is trivial that velocity norm measured from different reference frames is equal. However, with a moving sensor platform which experiences both translational and rotational movement, this insight may not be that trivial. Namely, if a rigid body has non-zero angular velocity, different points on it will experience different translational velocities due to the lever arm. To state this more formally, we present the following proposition

Proposition 1. *Translational velocity norm of moving objects estimated from two reference frames $\mathcal{F}_1$ and $\mathcal{F}_2$ on the same rigid body is invariant to the transform between the frames and the motion of the rigid body.*

Proof. Let ${}^{w}\boldsymbol{p}_k$ be the position of the observed object at time k in the $\mathcal{F}_w$. Then, let ${}^{1}\boldsymbol{p}_k$ and ${}^{2}\boldsymbol{p}_k$ be the same position expressed in the sensor reference frames $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively:

$$ {}^{1}\boldsymbol{p}_k = {}^{1}_{w}\boldsymbol{R}_k \cdot {}^{w}\boldsymbol{p}_k + {}^{1}_{w}\boldsymbol{t}_k, \tag{5} $$

$$ {}^{2}\boldsymbol{p}_k = {}^{2}_{1}\boldsymbol{R} \cdot {}^{1}\boldsymbol{p}_k + {}^{2}_{1}\boldsymbol{t}, \tag{6} $$

where we express the motion of the rigid body $({}^{1}_{w}\boldsymbol{R}_k, {}^{1}_{w}\boldsymbol{t}_k)$ as time-varying $SE(3)$ transform, while the transform between sensors frames (i.e. calibration) is constant in time $({}^{2}_{1}\boldsymbol{R}, {}^{2}_{1}\boldsymbol{t})$. Let us now observe displacement of the moving object in the two sensor frames, ${}^{1}\delta p$ and ${}^{2}\delta p$, between two discrete time instances k and l:

$$ {}^{1}\delta\boldsymbol{p} = {}^{1}\boldsymbol{p}_k - {}^{1}\boldsymbol{p}_l, \tag{7} $$

$$ {}^{2}\delta\boldsymbol{p} = {}^{2}\boldsymbol{p}_k - {}^{2}\boldsymbol{p}_l = {}^{2}_{1}\boldsymbol{R}({}^{1}\boldsymbol{p}_k - {}^{1}\boldsymbol{p}_l) = {}^{2}_{1}\boldsymbol{R}\,{}^{1}\delta\boldsymbol{p}. \tag{8} $$

Since the rotation matrix is orthogonal, the norm of displacement is equal, i.e. $||^2\delta\boldsymbol{p}|| = ||^2_1\boldsymbol{R}^1\delta\boldsymbol{p}|| = ||^1\delta\boldsymbol{p}||$. Thus, the translational velocity norm is also equal because it is simply the ratio of the above displacements over the time difference $k - l$. □ □

2.4 *Decalibration detection*

In a standard track fusion pipeline, track associations from the previous step are commonly used in object state estimate fusion. However, fusion depends on the accuracy of sensor calibration which can change over time due to disturbances. Thus, we propose a computationally inexpensive decalibration detection method, which is based on the data already present in the system. Similarly to the strategy presented by Deray et al. [32], we adopt a window-based approach for decalibration detection, but tailor the criterion we observe to accommodate the tracking-based scenario.

At the time instant t_k we form sets of corresponding track positions ${}^{i,j}\mathcal{S}_w = ({}^e\boldsymbol{x}_i, {}^e\boldsymbol{x}_j)$ that fall within the time window of length T_w ($t \in (t_k - T_w, t_k)$) for each sensor pair, where ${}^e\boldsymbol{x}_i$ and ${}^e\boldsymbol{x}_j$ represent stacked object positions obtained by i-th and j-th sensor, respectively. The positions are transformed from individual sensor frames $\mathcal{F}_i$ and $\mathcal{F}_j$ into the common reference frame $\mathcal{F}_e$ using the current calibration parameters. In the ideal case, the position should coincide, but due to the inevitable bias in the sensor measurements and the decalibration, in practice the error is always non-zero. To distinguish the error caused by bias from the decalibration error, we use an efficient closed-form solution for *orthogonal Procrustes problem* to obtain pairwise sensor calibrations [33]. Based on the ${}^{i,j}\mathcal{S}_w$, we form a 3×3 data matrix:

$$\boldsymbol{H} = ({}^e\boldsymbol{x}_i - {}^e\bar{\boldsymbol{x}}_i)({}^e\boldsymbol{x}_j - {}^e\bar{\boldsymbol{x}}_j)^T, \tag{9}$$

where ${}^e\bar{\boldsymbol{x}}_i$ and ${}^e\bar{\boldsymbol{x}}_j$ are means of corresponding sets. The rotation ${}^j_i\boldsymbol{R}$ can be found using the singular-value decomposition (SVD):

$$[U, S, V] = \mathrm{SVD}(H), \tag{10}$$

$${}^j_i\boldsymbol{R} = VU^T. \tag{11}$$

Since the ${}^j_i\boldsymbol{R}$ should be an identity matrix in the ideal case, we define the decalibration criterion for the time instant t_k as an angle of rotation in the angle-axis representation by

$$J_k = \arccos\left(\frac{\mathrm{Tr}({}^j_i\boldsymbol{R}) - 1}{2}\right). \tag{12}$$

When the criterion (12) surpasses a predefined threshold, the system proceeds to the complete graph-based sensor calibration. The magnitude of the minimal decalibration that can be detected is limited by the predefined threshold and the horizon defined with the T_w. Longer horizon enables detection of smaller calibration changes, but with slower convergence.

2.5 *Graph-based extrinsic calibration*

The last step of the pipeline estimates the extrinsic parameters when the system detects decalibration. As previously mentioned, we handle only rotational decalibration due to the limited accuracy and the bias in the measurements. Since we are dealing with more than two sensors, pairwise calibration would produce inconsistent transformations among the sensors. Thus, we rely

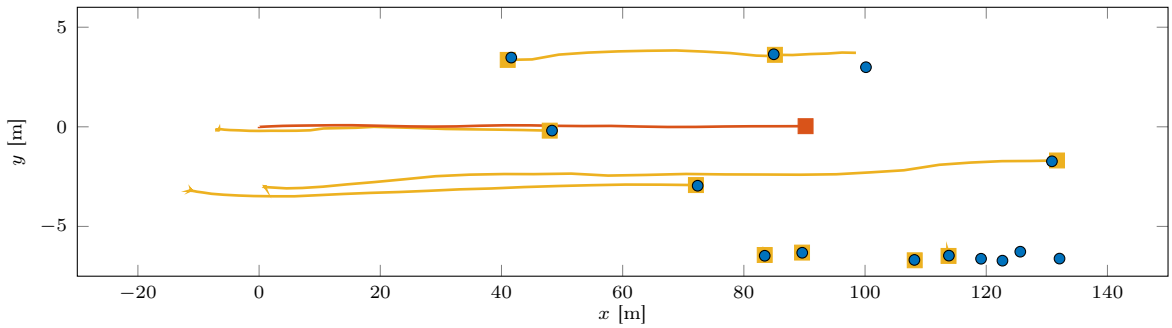


Figure 2. Illustration of the used scene showing ego-vehicle trajectory (red), lidar detections at $t = 19s$ (blue) and history of lidar tracks (yellow).

on the graph-based optimization presented in [34]. However, to ensure and speed up the convergence, we use the results of the previous step as an initialization. In the graph-based multi-sensor calibration paradigm, one sensor is chosen as an anchor and aligned with the $\mathcal{F}_e$ for convenience. We then search for the poses of other sensors with respect to the anchor sensor by minimizing the following criterion:

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \sum_{i,j}^{i \neq j} \sum_{k=1}^{N_{ij}} \boldsymbol{e}_{i,j,k}^{T} \cdot \boldsymbol{\Omega}_{i,j,k} \cdot \boldsymbol{e}_{i,j,k} \tag{13}$$

$$\boldsymbol{e}_{i,j,k} = {}^{i}\boldsymbol{p}_{i,k} - ({}^{i}_{j}\boldsymbol{R}(\boldsymbol{\phi}) {}^{j}\boldsymbol{p}_{j,k} + {}^{i}\boldsymbol{t}_{j}) \tag{14}$$

where ϕ is a set of non-anchor sensor rotation parametrizations and N_{ij} is the number of corresponding measurements between the $i-th$ and $j-th$ sensor. To enable integration of the noise from both sensors, we follow the total least squares approach presented in [35] and define the noise model as:

$$\boldsymbol{\Omega}_{i,j,k} = ({}^{i}_{j}\boldsymbol{R}(\boldsymbol{\phi}) V[{}^{j}\boldsymbol{p}_{j,k}] {}^{i}_{j}\boldsymbol{R}^{T}(\boldsymbol{\phi}) + V[{}^{i}\boldsymbol{p}_{i,k}])^{-1} \tag{15}$$

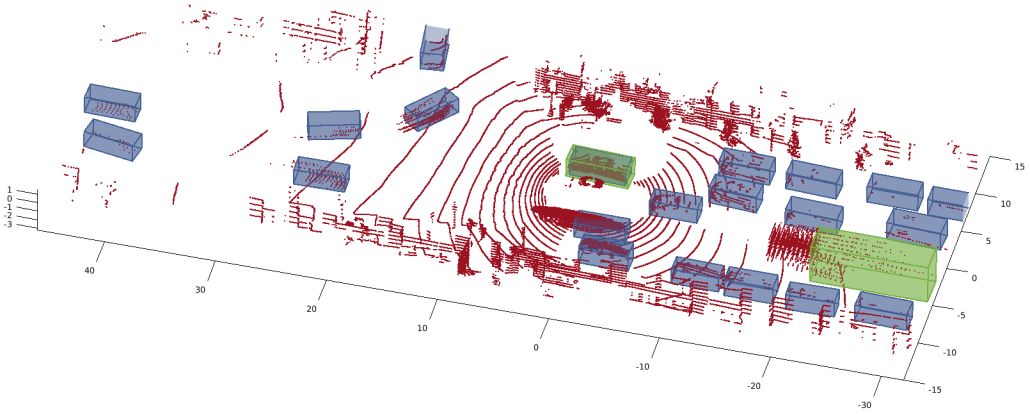
where $V[\cdot]$ is an observation covariance matrix of the zero-mean Gaussian noise. Additionally, if a sensor does not have a direct link with the anchor sensor, we obtain ${}^{i}_{j}\boldsymbol{R}$ by multiplying the corresponding series of rotation matrices to obtain the final rotation between the *i-th* and *j-th* sensor. This approach enables the estimation of all parameters with a single optimization, while ensuring consistency between sensor transforms.

3. Experimental Results

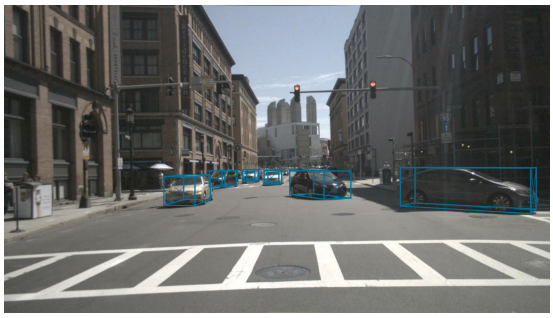
To validate the proposed method we used real world data provided with the nuScenes dataset [17]. Important details on the dataset, sensor setup and the scenario are given in Sec. 3.1, while Sec. 3.2 presents the results for each step of the calibration pipeline with greater attention on the introduced novelties related to calibration (Sec. 2.3 - 2.5).

3.1 *Experimental setup*

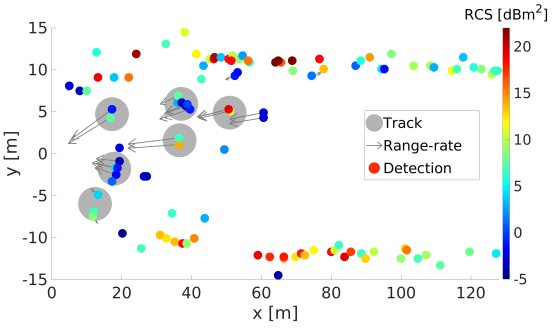
The nuScenes dataset consists of 1000 scenes that are 20 s long and collected with a vehicle driven through Boston and Singapore. The vehicle is equipped with a roof-mounted 3D lidar, 5 radars and 6 cameras. Each sensor modality has 360° coverage with small overlap of the sensors



(a) Lidar



(b) Camera



(c) Radar

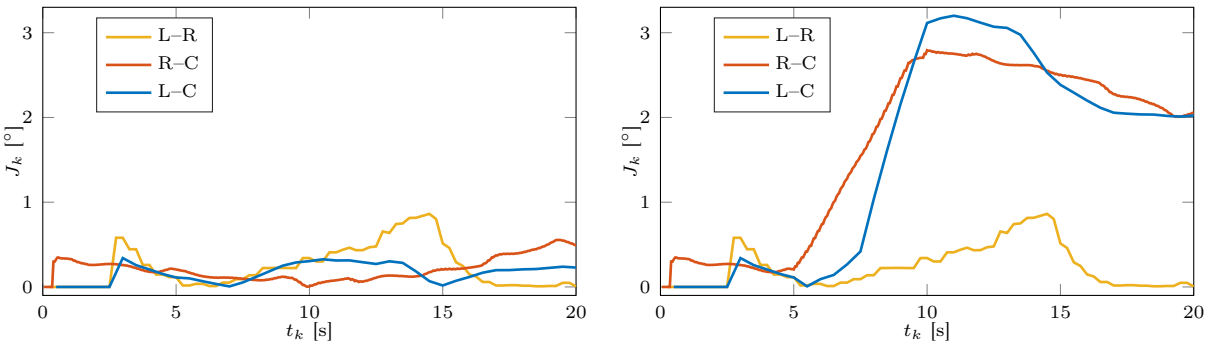
Figure 3. Vehicle detection using lidar, camera and radar. Lidar pointcloud consists of 10 consecutive sweeps, while blue and green boxes represent car and truck detections, respectively. Radar detections are colored with radar cross-section and show range rate, while gray circles represent confirmed tracks.

within the same modality. For clarity, in this experiment we focus only on measurements from the top lidar, front radar and front camera which all share a common FOV. The radar works at 13 Hz, camera 12 Hz, while the lidar provides point clouds with 20 Hz with the effective frame rate reduced to 2 Hz due to the object detector. The intrinsic and extrinsic calibration of all the sensors is obtained using several methods and calibration targets and is provided with the dataset. We considered it as a ground truth in the assessment of our method. Furthermore, we used isotropic and identical noise models for all the sensors involved.

In the following section, we present the results for *scene 343* from the dataset, because it contains variety of motions (cf. Fig. 2). The scene is from the test subpart of the dataset for which the data annotations are not provided. The ego vehicle is stationary during the first 5 s, while afterward it accelerates and reaches a speed of around 40 km/h. Through the scene, total of 17 moving vehicles are driving in both the same and the opposite direction, while some of them make turns. In addition, the scene contains 8 stationary vehicles in the detectable area for all the sensors.

3.2 *Results*

We present the results sequentially for all the steps of the method as they progress through the pipeline illustrated in Fig. 1. The starting point of the pipeline is object detection using individual sensors illustrated by Fig. 3. Object detection using lidar and camera provided reliable results for the range of up to 50 m, both for moving and stationary vehicles. Rare false negatives did not cause significant challenges for the subsequent steps. In comparison to the camera, lidar provided significantly more detections with frequent false positives which we successfully filtered by setting a threshold on their detection scores. In addition, Fig. 3(a) illustrates how the *MEGVII*



(a) Without decalibration

(b) Decalibration at $t_k = 5$

Figure 4. Test of the decalibration criterion J_k for each sensor pair through the scene with horizon $T_w = 5s$. Fig. 4(a) show the case with correct calibration, while the Fig. 4(b) shows an example of introducing 3° error in camera yaw angle. Significant increase in the criteria for pairs involving camera clearly indicates its decalibration.

network occasionally detects and classifies the same object as both car and truck, visible as blue and green box next to the ego-vehicle. However, these ambiguities were easily handled by the filtering within the tracking algorithm. In contrast to the other sensors, the radar provided many false positives and multiple detections of the same vehicles. We were able to extract only the moving vehicles based on the range rate, because it was difficult to differentiate stationary vehicles from the close-by surrounding buildings as they had the same range rate. Figure 3(c) illustrates radar detections colored with RCS, measured range-rate and confirmed radar tracks. It is clear that RCS is not a reliable measure for vehicle classification as it varies significantly across different vehicles due to their orientation, construction and other factors. On the other hand, the strongest reflections belong to the infrastructure and buildings, but the limited resolution prevents extracting fine structural information. Vehicles at closer range are usually detected as multiple objects, which was handled by the clustering algorithm. The range-rate provided useful information for classification of moving object, but the Fig. 3(c) illustrates how cross-traffic vehicles impose greater challenge because their range-rate is closer to zero.

The previously described detections were used in the subsequent tracking step for each sensor. Counting only tracks longer than 2 s, radar extracted 18 (105 s), lidar 25 (177 s) and camera 18 (84 s) tracks with total duration given in the parentheses. A visual of both detections and tracks for each sensor is available in the accompanying video[1].

To test the track association, which is the first step of the pipeline, we hand-labeled the ground truth track associations for each sensor pair by carefully observing the measurement data. The proposed method did not produce any false positive associations, while the success rate for each sensor pair was as follows: lidar–radar 93%; lidar–camera 94%; radar–camera 94%. An average time for two tracks to be associated after the tracking has started with both sensors was 1.5 s for every sensor combination. In addition, we note that introducing decalibration did not lead to any noticeable difference in results.

The following step, the decalibration detection, was tested in two scenarios as illustrated in Fig. 4. The two scenarios examine the temporal evolution of the change detection criterion J_k using the same horizon $T_w = 5s$. The scenario with correct calibration throughout the scene (Fig. 4(a)) shows that the criterion for each sensor pair is below 1° throughout the scene. The criterion varies mostly due to the number of correspondences between the sensors where the first part of the scene contains more moving objects than the second. In the second scenario (Fig. 4(b)), we introduced an artificial decalibration of 3° in the yaw angle of the camera frame with respect to ego frame at the time instant $t_d = 5$ s. We can notice a significant increase in the criterion for the sensor pairs involving the camera, while the criterion for lidar–radar remained the same. Thus, besides detecting system decalibration, we were also able to assess which sensor

[1] https://youtu.be/MgqIs-d6hRM

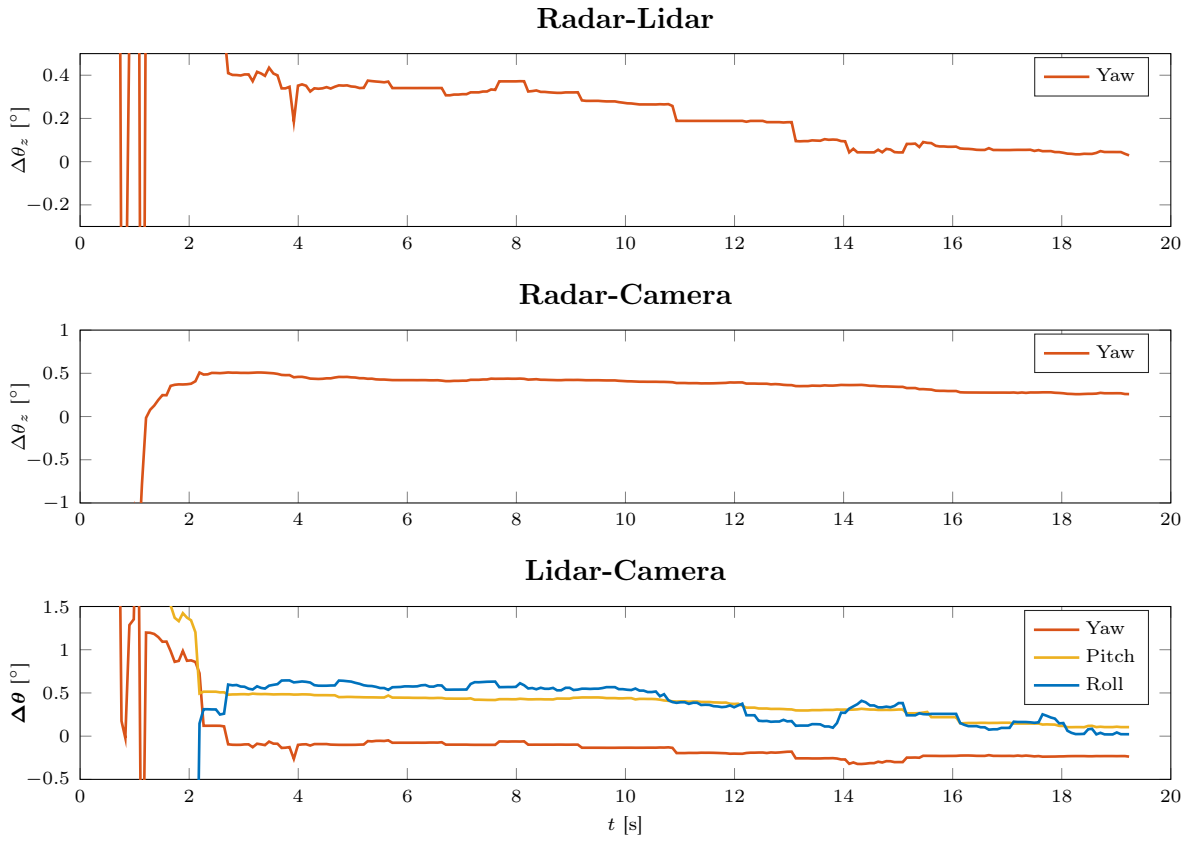


Figure 5. The plots show temporal evolution of the rotation calibration results by using available correspondences until the time $t = 20\,\mathrm{s}$. The errors are reported in Euler angles as the difference between the estimated and ground truth parameters obtained by target-based methods before the experiment. The results for sensor combination involving radar do not show pitch and roll as radar's missing elevation angle measurements prevent their accurate calibration.

changed its orientation by simply comparing the sensor-pairwise criteria.

Finally, we tested the extrinsic calibration by iteratively adding the available correspondences through the scene and running the calibration. Temporal evolution of the estimated calibration error is shown in the Fig. 5. The results are presented as a difference of ground truth and estimated pairwise rotation expressed in Euler angles for intuitive assessment. For the calibration involving radar, we observe only the yaw angle. Namely, even for the target-based methods [23], the accuracy of the remaining angles is limited due to the lack of elevation measurements. We can notice a consistent convergence of the error towards zero as more correspondences are added with the following final errors: lidar–radar yaw $\Delta\theta_z = 0.03°$; radar–camera yaw $\Delta\theta_z = -0.26°$; lidar–camera yaw, pitch and roll $\Delta\Theta = (-0.24, 0.10, 0.02)°$. Additionally, the analysis provided good guidelines for determining the minimal horizon in the previous step. For this particular scene, at least 2 s are necessary to reduce the error down to 0.5°. Furthermore, to test the influence of graph-based optimization, we performed pairwise calibration for all three sensor combinations using the data from the whole scene. Compared to the ground truth calibration, errors for sensor pairs were: lidar–radar yaw $\Delta\theta_z = 0.02°$; radar–camera yaw $\Delta\theta_z = -0.42°$; lidar–camera yaw, pitch and roll $\Delta\Theta = (-0.20, 0.22, -0.28)°$. While the magnitude of the error is similar to the joint graph optimization, pairwise calibration violates the consistency of the solution. Namely, rotational error of closing the loop, i.e., evaluating $\Delta R = {}^{l}_{c}R \cdot {}^{c}_{r}R \cdot {}^{r}_{l}R$, resulted with an error expressed in yaw, pitch and roll angles $\Delta\Theta = (-0.19, -0.18, -0.34)°$, while the problem of rotational error due to loop closing does not exist in the graph-based approaches.

3.3 *Comparison with odometry-based calibration*

To compare our method with an online calibration approach based on ego-motion, we tested the *SRRG* method proposed in [36]. The *SRRG* method is based on odometry constraints and can estimate vehicle odometry and extrinsic and temporal parameters of multiple sensors. However, we limit the comparison to lidar–camera calibration as these sensors can provide reliable 6DoF estimates of the vehicle ego-motion. For the lidar ego-motion, we used results of the map-based localization provided with the nuScenes dataset [17]. In the nuScenes dataset, only images from monocular cameras are available which prevents ego-motion estimation with correct scale, which is needed by *SRRG*. Therefore, we coupled the front camera images with the on-board IMU to obtain 6DoF odometry using the *Rovio* toolbox [37].

The chosen test scene presented significant challenges for the ego-motion methods because it included forward-only vehicle motion, which is usually the most common driving mode of vehicles. Namely, to achieve full observability, such methods usually require non-planar movement and excitation of at least two rotational axes [24]. This conclusion is also confirmed by our results, where translation in all the three axes and roll could not be estimated. For example, with a small perturbation in the initial calibration guess, the error in translation parameters reached 18 m, while the roll angle error reached 173°. This is not surprising, since these parameters are unobservable, but nevertheless they should not be estimated because they affect estimation accuracy of the observable parameters. Specifically, when estimating all 6DoF, error distributions of the yaw and pitch angles were $\mathcal{N}(-0.24°, 0.14°)$ and $\mathcal{N}(0.33°, 0.16°)$, respectively. On the other hand, when we locked the estimation of translational parameters by setting a prior to the ground truth values, we noticed a significant decrease in the standard deviation of the yaw and pitch angle error distributions $\mathcal{N}(-0.31°, 0.014°)$ and $\mathcal{N}(0.11°, 0.04°)$, respectively. However, the roll angle error was still significant with distribution $\mathcal{N}(24.11°, 0.961°)$. These results show that the proposed method yilded similar accuracy in the yaw and pitch angles as the odometry-based method, with an additional benefit of being able to estimate the roll angle on a dataset with forward-only motion.

4. Conclusion

In this paper we have proposed an online multi-sensor calibration method based on detection and tracking of moving objects. To the best of the authors' knowledge, this is the first method which calibrates radar–camera–lidar sensor system on a moving platform without relying on a known target. We proposed a complete pipeline for track based fusion which does not assume a constant and known sensor calibration. Proposed track to track association is based on a criterion resistant to decalibration, which is then followed by a decalibration detection relying on the information already present in the system without imposing significant computational burden. Finally, pairwise calibration provided by the decalibration detection module is used as an initialization for the final graph-based optimization which refines the results and provides consistent transformation across multiple frames. We validated the method on real world data from the nuScenes dataset which provides radar, lidar and camera measurements collected with a vehicle driving through an urban environment. The method was able to perform track association for calibration with high success rate and without wrong associations. Furthermore, it was able to detect decalibration within several seconds. Due to limited accuracy in position measurements, the method is currently limited to rotation calibration only. Nevertheless, it was able to estimate rotation parameters with an approximate error of 0.2° from a 20 s long scene.

For future work, we pan to explore the possibility of using the motion of moving objects for temporal calibration as well. We believe it could improve fusion since not all sensors, e.g. radar, can always be hardware synchronized. Additionally, a statistical analysis of individual sensor noises using the whole nuScenes and other datasets could further improve the results of the proposed method.

Acknowledgment

This work has been supported by the European Regional Development Fund under the grants KK.01.2.1.01.0022 (SafeTRAM) and KK.01.1.1.01.0009 (DATACROSS).

References

[1] Levinson J, Thrun S. Automatic Online Calibration of Cameras and Lasers. In: Robotics: Science and systems (rss). 2013.
[2] Moghadam P, Bosse M, Zlot R. Line-based extrinsic calibration of range and image sensors. In: Ieee international conference on robotics and automation (icra). 2013. p. 3685–3691.
[3] Pandey G, McBride JR, Savarese S, Eustice RM. Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information. Journal of Field Robotics. 2015;32(5):696–722.
[4] Park C, Moghadam P, Kim S, Sridharan S, Fookes C. Spatiotemporal Camera-LiDAR Calibration: A Targetless and Structureless Approach. IEEE Robotics and Automation Letters. 2020;5(2):1556 – 1563. 2001.06175.
[5] Taylor Z, Nieto J. Motion-Based Calibration of Multimodal Sensor Extrinsics and Timing Offset Estimation. IEEE Transactions on Robotics. 2016;32(5):1215–1229.
[6] Kellner D, Barjenbruch M, Dietmayer K, Klappstein J, Dickmann J. Joint radar alignment and odometry calibration. In: International conference on information fusion. 2015. p. 366–374.
[7] Kümmerle R, Grisetti G, Burgard W. Simultaneous Parameter Calibration , Localization , and Mapping. Advanced Robotics. 2012;26(17):2021–2041.
[8] Giamou M, Ma Z, Peretroukhin V, Kelly J. Certifiably Globally Optimal Extrinsic Calibration From Per-Sensor Egomotion. IEEE Robotics and Automation Letters. 2019;4(2):367–374. 1809.03554.
[9] Quenzel J, Papenberg N, Behnke S. Robust extrinsic calibration of multiple stationary laser range finders. In: Ieee international conference on automation science and engineering (case). 2016. p. 1332–1339.
[10] Glas DF, Miyashita T, Ishiguro H, Hagita N. Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. In: Ieee/rsj international conference on intelligent robots and systems (iros). 2010. p. 2938–2945.
[11] Glas DF, Ferreri F, Miyashita T, Ishiguro H. Automatic calibration of laser range finder positions for pedestrian tracking based on social group detections. Advanced Robotics. 2014;28(9):573–588.
[12] Tang Z, Lin YS, Lee KH, Hwang JN, Chuang JH, Fang Z. Camera self-calibration from tracking of moving persons. In: International conference on pattern recognition (icpr). 2017. p. 265–270.
[13] Jung J, Yoon I, Lee S, Paik J. Object Detection and Tracking-Based Camera Calibration for Normalized Human Height Estimation. Journal of Sensors. 2016;.
[14] Glas DF, Brscic D, Miyashita T, Hagita N. SNAPCAT-3D: Calibrating networks of 3D range sensors for pedestrian tracking. In: Ieee international conference on robotics and automation (icra). 2015. p. 712–719.
[15] Schöller C, Schnettler M, Krämmer A, Hinz G, Bakovic M, Güzet M, Knoll A. Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems. In: 2019 ieee intelligent transportation systems conference (itsc). 2019. p. 3934–3941. 1904.08743.
[16] Ćesić J, Marković I, Cvišić I, Petrović I. Radar and stereo vision fusion for multitarget tracking on the special Euclidean group. Robotics and Autonomous Systems. 2016;83:338–348.
[17] Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O. nuScenes: A multimodal dataset for autonomous driving. arXiv preprint. 2019;(March). 1903.11027.
[18] Wang X, Xu L, Sun H, Xin J, Zheng N. On-Road Vehicle Detection and Tracking Using MMW Radar and Monovision Fusion. IEEE Transactions on Intelligent Transportation Systems. 2016;17(7):2075–2084.
[19] Pitropov M, Garcia D, Rebello J, Smart M, Wang C, Czarnecki K, Waslander S. Canadian Adverse Driving Conditions Dataset. 2020;2001.10117.
[20] Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, Vasudevan V, Han W, Ngiam J, Zhao H, Timofeev A, Ettinger S, Krivokon M, Gao A, Joshi A, YuZhang, Shlens J, Chen Z, Anguelov D. Scalability in Perception for Autonomous Driving: An Open Dataset Benchmark. 2019;1912.04838.

[21] Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Wang D, Carr P, Lucey S, Ramanan D, Hays J. Argoverse: 3D Tracking and Forecasting with Rich Maps. 2019;:8748–87571911.02620.
[22] Domhof J, Kooij JFP, Gavrila DM. A Multi-Sensor Extrinsic Calibration Tool for Lidar , Camera and Radar. In: Ieee international conference on robotics and automation (icra). 2019. p. 1–7.
[23] Peršić J, Marković I, Petrović I. Extrinsic 6DoF calibration of a radar–LiDAR–camera system enhanced by radar cross section estimates evaluation. Robotics and Autonomous Systems. 2019;114.
[24] Brookshire J, Teller S. Extrinsic Calibration from Per-Sensor Egomotion. In: Robotics: Science and systems (rss). 2012.
[25] Zhu B, Jiang Z, Zhou X, Li Z, Yu G. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. arXiv preprint. 2019;:1–81908.09492.
[26] Zhou X, Wang D, Krähenbühl P. Objects as Points. arXiv preprint. 2019;1904.07850.
[27] Weng X, Kitani K. A Baseline for 3D Multi-Object Tracking. arXiv preprint. 2019;1907.03961.
[28] Kuhn HW. The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly. 1955;2(5):83–97.
[29] Rong Li X, Jilkov V. Survey of Maneuvering Target Tracking. Part I. Dynamic Models. IEEE Transactions on Aerospace and Electronic Systems. 2003;39(4):1333–1364.
[30] Houenou A, Bonnifait P, Cherfaoui V. A Track-To-Track Association Method for Automotive Perception Systems. In: Intelligent vehicles symposium (iv). 2012. p. 704–710.
[31] Peršić J, Petrović L, Marković I, Petrović I. Spatio-Temporal Multisensor Calibration Based on Gaussian Processes Moving Object Tracking. arXiv preprint. 2019;1904.04187.
[32] Deray J, Sola J, Andrade-Cetto J. Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. In: European conference on mobile robots (ecmr). 2019. p. 1–6.
[33] Larusso A, Eggert D, Fisher R. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In: The british machine vision conference. 1995. p. 237–246.
[34] Owens JL, Osteen PR, Daniilidis K. MSG-cal: Multi-sensor graph-based calibration. In: Ieee/rsj international conference on intelligent robots and systems (iros). 2015. p. 3660–3667.
[35] Estépar RSJ, Brun A, Westin CF. Robust Generalized Total Least Squares. In: International conference on medical image computing and computer-assisted intervention (miccai). 2004. p. 234–241.
[36] Corte BD, Andreasson H, Stoyanov T, Grisetti G. Unified Motion-Based Calibration of Mobile Multi-Sensor Platforms with Time Delay Estimation. IEEE Robotics and Automation Letters. 2019; 4(2):902–909.
[37] Bloesch M, Burri M, Omari S, Hutter M, Siegwart R. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. International Journal of Robotics Research. 2017;36(10):1053–1072.